BigFix
Installation Guide

Special notice

Before using this information and the product it supports, read the information in Notices (on page 684).

Edition notice

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

Contents

Special notice.....2

Edition notice.....3

Chapter 1. Introduction.....1

V11 Overview.....2

What is new in BigFix 11 Platform.....6

Architectural components overview.....43

Chapter 2. BigFix Platform Unicode Support Overview.....46

Masthead encoding parameters.....47

Top-down data flow: from the BigFix server to the clients.....47

Bottom-up data flow: from BigFix clients to BigFix server.....48

Unicode support requirements and limitations.....49

Reading and writing files in the specific encodings.....53

Background information.....54

Reading file inspectors.....54

Writing file with the encode command.....57

Reading and writing files with encode.....58

Managing actions on clients with different local encoding.....59

Chapter 3. Sample deployment scenarios.....61

Basic deployment.....61

Main Office with Fast-WAN Satellites.....64

Disaster Server Architecture.....66

Efficient relay setup.....68

Hub and spoke.....70

Remote Citrix / Terminal Services Configuration....73
Chapter 4. Requirements and assumptions....76
Server requirements....76
Console requirements....77
Client requirements....78
Database requirements....81
Security requirements....91
Network configuration requirements....93
Assumptions....94
Chapter 5. Types of installation....96
Evaluation installation....96
Production installation....97
Moving from evaluation installation to production installation....97
Chapter 6. Managing licenses....100
Creating the License Authorization File....103
Licensing Assistance....103
Extending the license entitlements....103
Distributing the Updated License and Masthead....105
Distributing the masthead from the Windows server to clients....106
Distributing the masthead from the Linux server to the clients....110
Chapter 7. Before installing....111
Configuring a Local Firewall....111
Modifying port numbers....111
Understanding the server components....112
Chapter 8. Installing on Windows systems....114

Installing and configuring SQL Server 2025 DB....114
Installing and configuring DB on Cloud....115
Downloading BigFix....118
Performing an evaluation installation....119
Requesting a license certificate and creating the masthead....131
Installing the components....141
Managing the Server ID limit....142
Installing the Windows primary server....143
Running the BigFix Diagnostics tool....161
Installing the Client on Windows....166
Installing the Console....166
Installing Web Reports Standalone....167
Installing the WebUI Standalone....172
Removing the BigFix components from Windows....177
DSA on Windows....179
Installing Additional Windows Servers (DSA)....179
Authenticating Additional Servers....186
Uninstalling a Windows replication server....189
Chapter 9. Installing on Linux systems....190
Installing and configuring DB2....190
Installing and configuring MS SQL....192
Installing and configuring DB on Cloud....194
Downloading BigFix....197
Performing an evaluation installation with DB2....197
Performing an evaluation installation with MS SQL....201

Installation Command Options..... 203
Installing the components..... 204
Installing the Server..... 204
Installing Web Reports Standalone with DB2..... 229
Installing Web Reports Standalone with MS SQL..... 234
Installing the WebUI Standalone with DB2..... 238
Installing the WebUI Standalone with MS SQL..... 241
Verifying Server Installation..... 244
Silent installation..... 246
Installing the Client on Linux..... 269
Installing the Console..... 269
Installation Folder Structure..... 270
Managing the BigFix Services..... 273
Changing the DB2 port..... 275
Removing the BigFix components from Linux..... 276
DSA on Linux..... 278
Installing Additional Linux Servers (DSA)..... 278
Authenticating Additional Servers (DSA)..... 280
Uninstalling a Linux replication server..... 281
Chapter 10. Installing the clients..... 283
Using the Client Deploy Tool..... 283
Target prerequisites..... 283
Client Deploy Tool wizard..... 285
Client Deploy Tool Fixlet..... 292
Client Deploy Tool standalone..... 296

Troubleshooting the client deployment..... 301
Log files..... 305
Uploading the target logs to the server..... 307
Limitations..... 308
Installing the Client on AIX..... 308
AIX Fixlet Content..... 310
Installing the Client on Linux..... 310
Alma Linux Installation Instructions..... 311
Amazon Linux Installation Instructions..... 312
CentOS Installation Instructions..... 313
Oracle Linux Installation Instructions..... 315
Raspbian Installation Instructions..... 316
Red Hat Installation Instructions..... 317
Rocky Linux Installation Instructions..... 324
SUSE Linux Enterprise (64-bit) Installation Instructions..... 326
OpenSUSE Leap (64-bit) Installation Instructions..... 327
Ubuntu/Debian (64-bit) Installation Instructions..... 329
Installing the Client on Mac..... 330
Mac Fixlet Content..... 331
Installing the Client on Solaris 11..... 332
Installing the Client on Windows..... 335
Installing the Client using the .exe setup..... 336
Installing the Client using the .msi setup..... 338
Embedding in a Common Build..... 344
Avoiding duplicates when a Client is restored..... 346

Enabling encryption on Clients..... 348
Chapter 11. BigFix Administration Tool..... 350
BESAdmin Windows GUI..... 350
Masthead Management..... 351
System Options..... 351
Advanced Options..... 352
Replication..... 353
Encryption..... 354
Security..... 356
Computer Remover..... 358
Audit Trail Cleaner..... 364
Health and Cleanup..... 370
Property ID Mapper..... 374
BESAdmin Windows Command Line..... 376
Working with TLS cipher lists..... 419
BESAdmin Linux Command Line..... 422
Working with TLS cipher lists..... 464
Logging Cleanup Tasks Activities..... 467
Chapter 12. Post-installation configuration steps..... 469
Post-installation steps..... 469
Starting and stopping the BigFix server..... 473
Subscribing to content sites..... 475
Changing the database password..... 476
Chapter 13. Managing relays..... 479
Relay requirements and recommendations..... 480

Setting up a relay..... 481
Installing and upgrading a relay from the command line..... 482
Assigning relays to clients..... 483
Assigning relay at client installation time.....483
Manually assigning relays to existing clients.....488
Automatically assigning relays at client installation time..... 489
Automatically assigning relays to existing clients..... 489
Using relay affiliation.....490
Notes about automatic relay assignment..... 492
Adjusting the BigFix Server and Relays.....492
Assigning a relay when the server is unreachable..... 494
Setting up internet relays.....494
Viewing which relay is assigned to a client..... 499
Viewing the relay chain on the client..... 499
Chapter 14. Relay Health.....504
Chapter 15. Introduction to Tiny Core Linux - BigFix Virtual Relay..... 511
Architectural overview.....511
Phase 1 - Configuring the Tiny Core Linux virtual machine..... 513
Downloading the ISO image..... 513
Creating a virtual machine.....513
Installing Tiny Core Linux on the virtual machine..... 514
Phase 2 - Preparing the BigFix Virtual Relay template.....521
Template setup and customization from a local folder.....523
Template setup and customization from the network.....526
Creating the BigFix Virtual Relay template..... 531

Phase 3 - Configuring the BigFix Virtual Relay instance..... 531
Deploying a new BigFix Virtual Relay instance by using Auto-Deployment..... 531
Manually deploying a new BigFix Virtual Relay instance..... 533
Maintenance..... 536
Troubleshooting..... 539
Limitations..... 539
VMware and Open VM tools..... 539
Auto-Deployment..... 541
Modifying the Virtual Relay template..... 542
Chapter 16. Setting up a proxy connection..... 545
Enabling client polling..... 550
Connecting the console to the server through a proxy..... 551
Setting a proxy connection on the server..... 552
Setting up a proxy connection on a relay..... 557
Setting up a proxy connection on a client..... 561
Best practices to consider when defining a proxy connection..... 565
Troubleshooting proxy connection..... 567
Error Unable to get site content (failed to pass sha1 hash value checks in BESRelay.log..... 567
Error Unexpected HTTP response: 503 Service Unavailable in GatherDB.log..... 568
How to check if the proxy configuration is correct..... 568
New site version cannot be gathered..... 570
Chapter 17. Running backup and restore..... 572
On Windows systems..... 573
Server Backup..... 573

Server Recovery..... 576
Verifying restore results..... 579
DSA Recovery..... 580
On Linux systems..... 584
Server Backup..... 584
Server Recovery..... 587
Verifying restore results..... 591
Enabling the DB2 database online backup..... 592
Automatic DB2 databases backup upon upgrade..... 592
DSA Recovery..... 594
Chapter 18. HA and DR configurations..... 597
Introduction to HA and DR..... 597
Configuring HA and DR on Windows..... 602
HA and DR limitations..... 619
Chapter 19. Upgrading..... 620
Upgrade paths to BigFix 11..... 620
Before upgrading..... 621
Upgrade prerequisite checks..... 622
Upgrade steps..... 624
Automatic upgrade..... 625
Manual upgrade on Windows..... 626
Upgrading the installation generator and the primary server..... 627
Upgrading a BigFix Server with a remote database..... 627
Upgrading the secondary server..... 630
Upgrading the Console..... 631

Upgrading the remote WebUI..... 631
Upgrading the Relays..... 632
Upgrading the Clients..... 632
Upgrading the remote Web Reports server..... 632
Manual upgrade on Linux..... 633
Upgrading the server..... 634
Upgrading the Console..... 636
Upgrading the relays..... 636
Upgrading the clients..... 636
Upgrading the Web Reports and WebUI standalone servers..... 637
Rollback..... 637
Chapter 20. SQL Server parallelism optimization..... 638
Chapter 21. Known limitations and workarounds..... 647
Appendix A. Logging paths..... 650
Appendix B. Uninstalling the BigFix client..... 658
Uninstalling the BigFix Client on AIX..... 658
Uninstalling the BigFix Client on Linux..... 660
Uninstalling the BigFix Client on Solaris..... 662
Uninstalling the BigFix Client on Mac..... 663
Appendix C. Glossary..... 664
Appendix D. Support..... 679
Appendix E. Accessibility features for BigFix..... 680
Notices..... 684
Index.....

Chapter 1. Introduction

BigFix aims to solve the increasingly complex problem of keeping your critical systems updated, compatible, and free of security issues. It uses patented Fixlet technology to identify vulnerable computers in your enterprise. With just a few mouse-clicks you can remediate them across your entire network from a central console.

Fixlets are powerful, flexible, and easily customized. Using Fixlet technology, you can:

- Analyze vulnerabilities (patched or insecure configurations)
- Easily and automatically remediate all your networked endpoints
- Establish and enforce configuration policies across your entire network
- Distribute and update software packages
- View, modify, and audit properties of your networked client computers

Fixlet technology allows you to analyze the status of configurations, vulnerabilities, and inventories across your entire enterprise and then enforce policies automatically in near realtime. In addition, administrators can create or customize their own Fixlet solutions and tasks to suit their specific network needs.

BigFix is easy to install and has built-in public and private-key encryption technology to ensure the authenticity of Fixlets and actions. It grants you maximum power as the administrator, with a minimal impact on network traffic and computer resources. BigFix can handle hundreds of thousands of computers in networks spanning the globe.

When installed, you can easily keep your networked computers correctly configured, updated, and patched, all from a central console. You can track the progress of each computer as updates or configuration policies are applied, making it easy to see the level of compliance across your entire enterprise. In addition to downloads and security patches, you can also examine your managed computers by specific attributes, allowing you to group them for action deployments, ongoing policies, or asset management. You can log the results to keep an audit trail and chart your overall activity with a convenient web-based reporting program.

V11 Overview

BigFix Platform 11 introduces the following major changes in the security area:

- OpenSSL has been upgraded to 3.1.1
- TLS 1.3 protocol for HTTPS communications is now supported
- SHA-384 is now available as cryptographic hash function for all digital signatures.

With BigFix Platform 11:

- SHA-256 and TLS 1.2 are the minimum security levels used for digital signatures and HTTPS communications respectively
- SHA-384-only and/or TLS 1.3-only modes can be optionally enforced
- SHA-1 and SHA-256 are still used for validation of downloaded files in prefetch statements
- BigFix 10.0.7 is the minimum version supporting the upgrade of the BigFix Server components to Version 11.

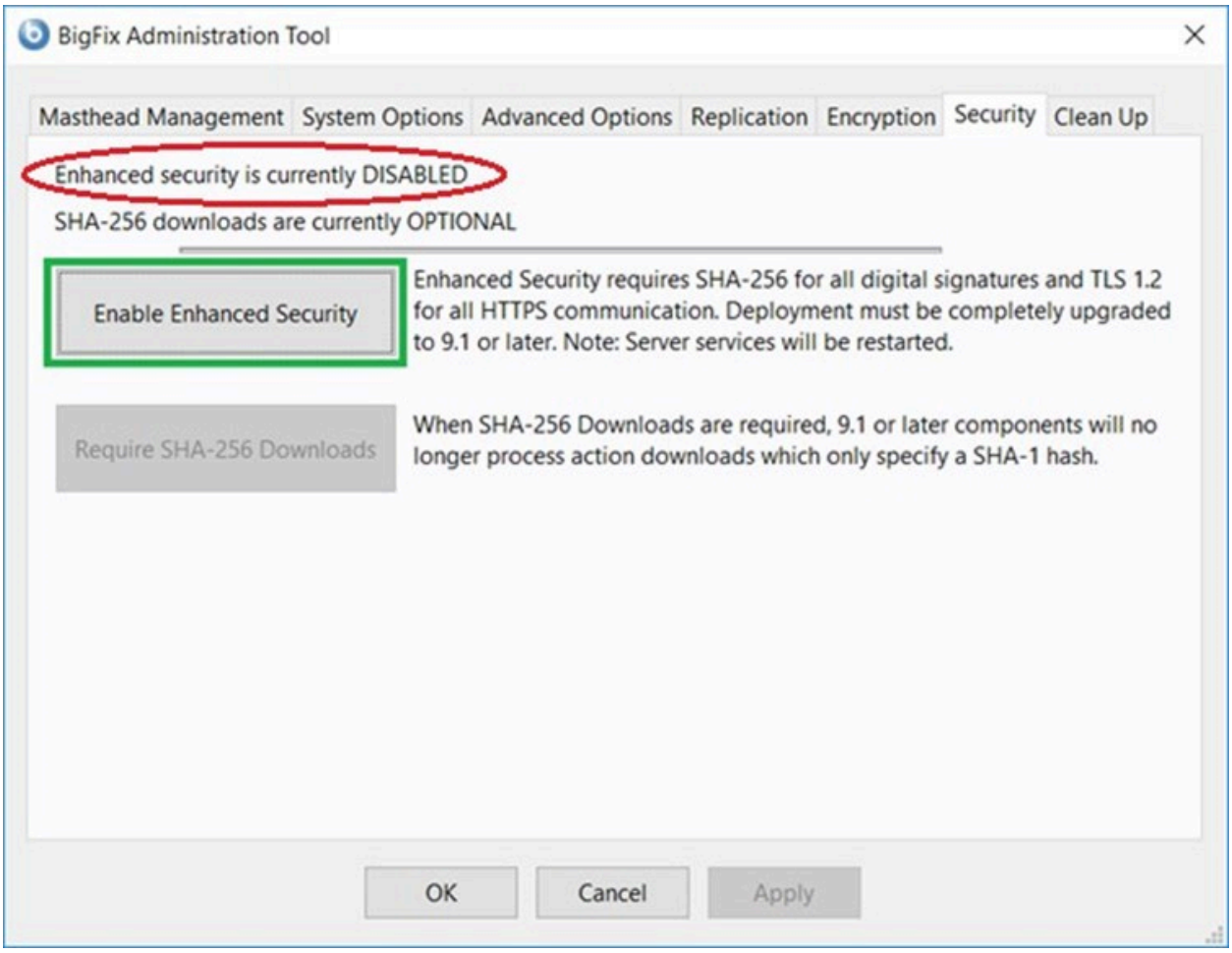
Considerations on the upgrade of the BigFix Platform to v11

Compared to previous BigFix versions, there are no changes regarding the manual and Fixlet upgrade options and procedures. The only main difference is related to the upgrade of the BigFix Server for which the following prerequisites must be satisfied:

- BigFix 10.0.7 is the minimum version supporting the upgrade of the BigFix server components to Version 11.
- “Enhanced Security” must be enabled (Security Configuration Scenarios).

Note: Plan to enable the "Enhanced Security" well in advance of the upgrade because the feature takes time to propagate through the deployment. For detailed information and guidance, please read carefully Guidance on Enabling Enhanced Security prior to enabling "Enhanced Security".

On Windows, you can enable the "Enhanced Security" using the relevant button of the graphical BigFix Administration Tool, as shown in the following picture:



On Linux, run the following command:

```
/opt/BESServer/bin/BESAdmin.sh -securitysettings -enableEnhancedSecurity
-sitePvkLocation=<path+license.pvk> [ -sitePvkPassword=<password> ]
```

By enabling "Enhanced Security", objects within the BigFix Server database will be re-signed. Before proceeding with the upgrade, it is recommended to check for invalid data by running the BESAdmin Command Line with the findinvalidsignatures option as described in:

- findinvalidsignatures *(on page 433)* on Linux platforms
- findinvalidsignatures *(on page 388)* on Windows platforms.

Note: The update Fixlets might take longer than usual to be reported as relevant due to the change in major version of the BigFix Server.

Results of the upgrade to BigFix v11

After you have started upgrading to BigFix 11, even if your deployment is not all at Version 11 yet, you will get the following benefits:

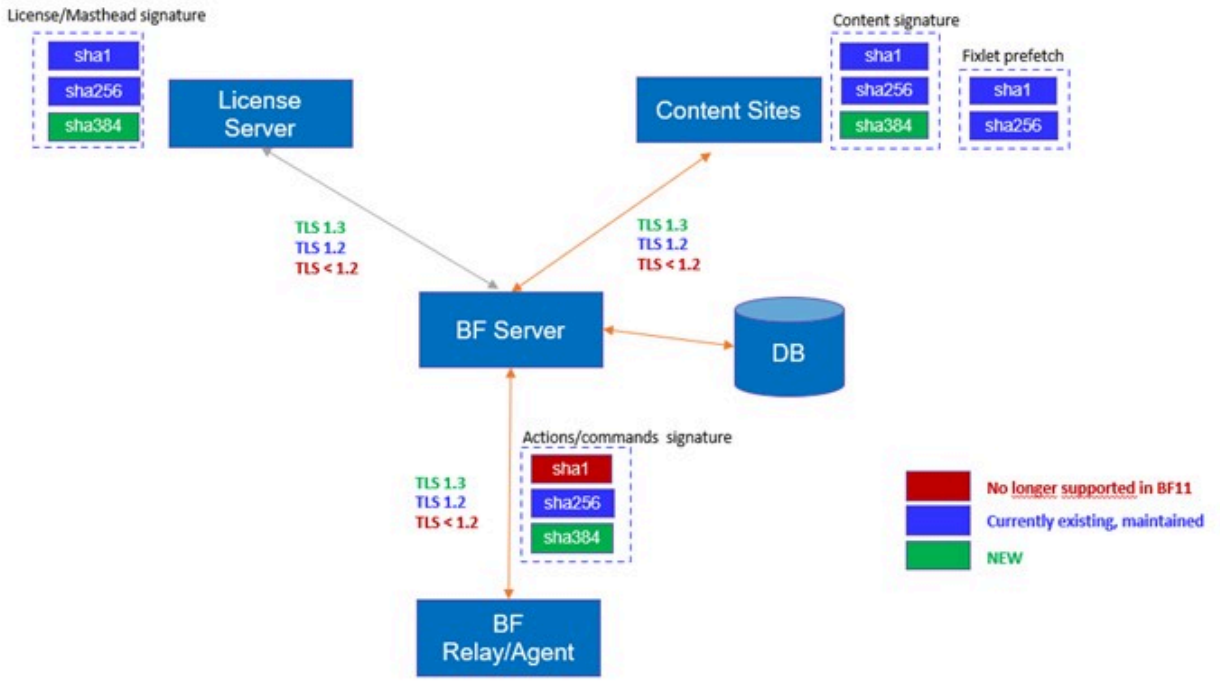
- TLS handshakes will use SHA-384 certificates whenever they are available. After upgrading the BigFix Server to v11, SHA-384 certificates will be available to:
  - All Agents/Relays at Version 11
  - Agents and Relays at 10.0 patch 7 (or later patch) that have had their certificates refreshed, either automatically for expiration of the 13-month validity period, or manually with the “client certificate refresh” actionscript command (see Client certificate)
  - All Agents and Relays installed after upgrading the BigFix Server to Version 11, regardless of their version.
- Content generated by MOs and NMOs like sites, Fixlets, analyses, actions, etc. will be signed with both SHA-256 and SHA-384. To validate such content, BigFix components at v11 will rely on both, while BigFix components at earlier versions will rely on the SHA-256 signature.
- The BigFix Server will validate external site content using either the SHA-256 or the SHA-384 signature.
- The BigFix Server will process Agent reports that are signed using either SHA-256 or SHA-384.
- BigFix Platform 11 will support both TLS 1.2 and TLS 1.3 (with TLS 1.3 being always the first attempted option), while it will no longer support TLS 1.1 or below in ANY scenario requiring HTTPS.

Note: Because of the above, BigFix Platform components at Version 11 will no longer be backward compatible with BigFix components at version 9.0 or lower.

Note: SHA-1 and SHA-256 are still used for validation of downloaded files in prefetch statements.

The communications with the BigFix Database are not affected by the version change but depend on the database configuration.

The following picture provides a summary of what is described above.



Enforcing SHA-384-only for signatures

You can optionally require the usage of SHA-384 as the only allowed cryptographic hash function for all digital signatures.

Note: This requires the whole deployment to be at Version 11. BigFix components at earlier versions will be no longer manageable (e.g. Agents and Relays will no longer be able to gather, validate and process content propagating from the BigFix Server).

Enforcement can be done with the BigFix Administration tool (Security *(on page 356)*) and has the following consequences:

- Any content generated by MOs and NMOs like sites, Fixlets, analyses, actions, etc. will be signed only with SHA-384. Therefore, only Agents and Relays at Version 11 will be able to validate and process it.
- The BigFix Server will validate external site content relying only on the SHA-384 signature.
- The BigFix Server will only process Agent reports that are signed using SHA-384.

Enforcing TLS 1.3-only for HTTPS communications

You can optionally require the usage of TLS 1.3 as the only allowed protocol for HTTPS communications.

Note: This requires the whole deployment to be at Version 11. BigFix components at earlier versions will be no longer manageable (e.g. Agents and Relays will no longer be able to communicate via HTTPS with BigFix components of Version 11).

Enforcement can be done with the BigFix Administration tool (Security *(on page 356)*) and will affect the following scenarios:

- Internal HTTPS communications between BigFix components
- HTTPS communications where a BigFix component plays as HTTPS Server (e.g. a REST API client connecting to the BigFix Server, or a browser connecting to the Relay Diagnostics page)
- External site gathers performed by the BigFix Server.

Libraries upgraded in V11

For a full list of libraries upgraded in BigFix Platform V11, see What is new in BigFix 11 Platform (on page 6).

Updated system requirement matrix in V11

In BigFix Platform 11 the system requirement matrix has been updated. For more information, see BigFix Support Matrix.

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.

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.

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.
- **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 and Configuring FIPS 140-3 on the BigFix Server. 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 *(on page 350)*, BESAdmin Windows Command Line *(on page 376)* and BESAdmin Linux Command Line *(on page 422)*.

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.
- **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, BESAdmin Windows Command Line *(on page 376)* and BESAdmin Linux Command Line *(on page 422)*.
- **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.

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 and Peer to peer mode.
- **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, 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.
- **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.

- **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, Microsoft Azure Plugin Commands, Google Cloud Platform Plugin Commands and VMware Plugin Commands.

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 *(on page 370)*, BESAdmin Windows Command Line *(on page 376)* and BESAdmin Linux Command Line *(on page 422)*.

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.

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.

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.

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, BESAdmin Windows Command Line (on page 376) and BESAdmin Linux Command Line (on page 422).

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 (on page 597).

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 and Peer to peer mode.

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.

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.

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.

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 (on page 376).

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 (on page 376), BESAdmin Linux Command Line (on page 422), Computer Remover (on page 358) and Audit Trail Cleaner (on page 364).

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 (on page 376) and BESAdmin Linux Command Line (on page 422).

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.

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.

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.

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.

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.

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 (on page 376) and BESAdmin Linux Command Line (on page 422).

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.

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.

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 (on page 364).

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 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 (on page 273).

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.

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, Configuring cloud plugins, VMware Asset Discovery Plugin Inspectors and VMware Plugin Commands.

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 (on page 212), Installing and configuring MS SQL (on page 192), Installing Web Reports Standalone with MS SQL (on page 234) and Installing the WebUI Standalone with MS SQL (on page 241).

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 (on page 376).

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
- 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 (on page 2).

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 (on page 2).

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 (on page 2).

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.

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

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 (on page 161) and Database requirements (on page 81).

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 (on page 81) 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

Architectural components overview

The BigFix system encloses different components. Main components are:

BigFix Agents:

They are installed on every computer that you want to manage using BigFix. A computer on which the BigFix agent is installed, is also referred to as client. Clients access a collection of Fixlets that detects security exposures, incorrect configurations, and other vulnerabilities. The client can implement corrective actions received from the console through the server. The BigFix client runs undetected by users and uses a minimum of system resources.

BigFix also allows the administrator to respond to screen prompts for those actions that require user input. BigFix clients can encrypt their upstream communications, protecting sensitive information. To acknowledge all the Operating Systems that the BigFix Client software supports, please refer to the BigFix Support Matrix.

BigFix Servers:

Offer a collection of interacting services, including application services, a web server, and a database server, forming the heart of the BigFix system. They coordinate the flow of information to and from individual computers and store the results in the BigFix database. The BigFix server components operate quietly in the background, without any direct intervention from the administrator. BigFix servers also include a built-in **Web Reporting** module to allow authorized users to connect through a web browser to view all the information about computers, vulnerabilities, actions, and more. BigFix supports multiple servers, adding a robust redundancy to the system.

To acknowledge all the Operating Systems that the BigFix Server software supports, please refer to the BigFix Support Matrix.

Note: On Windows, as of BigFix V10, Server and Web Reports components support only 64-bit architecture.

BigFix Relays:

Increase the efficiency of the system. Instead of forcing each networked computer to directly access the BigFix server, relays spread the load. Hundreds to thousands of BigFix clients can point to a single BigFix Relay for downloads, which in turn makes only a single request to the server. BigFix Relays can connect also to other relays, further increasing efficiency. A BigFix Relay need not be a dedicated computer.

To acknowledge all the Operating Systems that the BigFix Relay software supports, please refer to the BigFix Support Matrix.

As soon as you install a BigFix Relay, the clients in your network can automatically discover and connect to them.

BigFix Consoles:

Join all these components together to provide a system-wide view of all the computers in your network, along with their vulnerabilities and suggested remedies. The BigFix Console allows an authorized user to quickly and simply distribute fixes to each computer that needs them without impacting any other computers in the network.

To acknowledge all the Operating Systems that the BigFix Console software supports, please refer to the BigFix Support Matrix.

Note: The computer where the Console runs requires network access to the BigFix server.

Note: On Windows, as of BigFix V10, the Console component supports only the 64-bit architecture.

BigFix Client Deploy Tool:

Allows you to install Windows, UNIX and Mac target computers in an easy way.

BigFix Plugin Portal:

Helps you manage cloud devices as well as modern devices enrolled in your BigFix deployment.

BigFix WebUI:

A cross-platform user interface that lets you perform actions on devices, perform deployments and manage reports.

Chapter 2. BigFix Platform Unicode Support Overview

BigFix Platform V11 gathers data from BigFix clients deployed with different code pages and languages. It encodes all data into UTF-8 format, and reports it back to the BigFix server.

This capability is useful when your environment has clients with different code pages and the client reports contain non-ASCII characters. To achieve this result, the masthead file has been modified. After it is propagated to the clients and the new values become active, all the client reports containing non-ASCII characters are displayed correctly on the BigFix console, even if the reports come from clients with code page different from the BigFix server code page. The BigFix console works also if installed on a system with local encoding different from the BigFix server encoding by getting the BigFix server encoding and the report encoding from the masthead.

Starting from BigFix Version 9.5.13, to gather sites that contain non-ASCII fixlets on Linux computers, you must enable the _BESGather_Download_AllowNoStopTranscoding configuration setting. For details, look up the setting in List of settings and detailed descriptions.

To understand how this feature works, read about the concepts of FXF encoding and report encoding.

FXF encoding

Affects the top-down data flow, that is, the encoding of the data that flows from the BigFix server to the BigFix clients (such as actions, site subscriptions, computer groups, and more).

Report encoding

Affects the bottom-up data flow, that is, the encoding of data that flows from the BigFix clients to the BigFix server.

Important: Before upgrading BigFix server to V11, it is required that it is at Version 10 Patch 7 later.

Note: The language in which the Web Reports component is displayed is determined by the browser settings. For example if your browser language is Japanese, then you will see Web Reports in Japanese.

Masthead encoding parameters

The masthead file contains the following two new parameters related to Unicode:

```
x-bes-fxf-charset: codepage_data_from_server_to_clients (for
 example:Windows_1252)
x-bes-report-charset: utf-8
```

where:

x-bes-fxf-charset

Represents the **FXF encoding** and affects the encoding of the data that is sent from the BigFix server to the BigFix clients (such as actions, site subscriptions, computer groups, and more). Its value is determined by the code page of the system where the BigFix server component is installed, and, for both Windows and Linux systems, can assume one of the values of the Code Pages supported by Windows.

x-bes-report-charset

Represents the **report encoding** and affects how BigFix clients encode their reports before sending them to the BigFix server. Its value is always **utf-8**.

Top-down data flow: from the BigFix server to the clients

The encoding capability when sending data from the BigFix server to BigFix clients is the same as BigFix older versions; UTF-8 support of the top-down data flow is not available

in BigFix. The top-down data flow (such as actions, site subscriptions, computer groups, custom sites, and more) works only if you use non-ASCII data that belongs to the FXF encoding.

For example, if you try to create a custom site whose name contains characters different from the FXF encoding (suppose FXF encoding is Windows 1252, and the name of the site is `Site_albêrto`, where character `ê` is specific to the Windows 1250 encoding), the BigFix console displays an error message. The same error occurs when trying to create this custom site using the REST API or Command-Line Interface (CLI).

On BigFix servers running on Windows systems the installer sets the FXF encoding (`x-bes-fxf-charset`) in the masthead without any user interaction.

On BigFix servers running on Linux systems the installer shows the `x-bes-fxf-charset` setting to the user, proposing a default value and allowing the user to confirm or change it. You can also change it by using the `ENCODE_VALUE` response key in the silent installations.

Bottom-up data flow: from BigFix clients to BigFix server

BigFix Platform V11 uses UTF-8 as the standard encoding system to send reports from BigFix clients to BigFix server. The `x-bes-report-charset` parameter is set to **utf-8** in the masthead and its value cannot be changed.

In this way, any text character is efficiently represented and handled, regardless of the language, application, or platform that you are working on.

After upgrading a BigFix server to V11, the updated masthead with the new `x-bes-fxf-charset` and `x-bes-report-charset` parameters is propagated through the whole deployment.

If the value of the `x-bes-fxf-charset` in the updated masthead is different from the value of the FXF encoding that the client was using before the upgrade, then the `BESClient` process must be restarted for the new value to become effective.



Note: After upgrading BigFix to V11, the `filldb` process on clients might fail if the `filldb` buffer directory contains reports that use an unexpected report encoding

value, for example `ReportEncoding: hp-roman8`. If so, clean up the `filldb` buffer directory containing the reports and then restart `filldb`.

Unicode support requirements and limitations

This is a list of BigFix Platform V11 requirements and limitations when using non-ASCII characters:

- **BigFix installation:**
  - The BigFix components (server, relays, WebReports) must be installed on systems whose hostname contains only ASCII characters.
  - The installation paths of the components must contain only ASCII characters.
  - The License Key Password cannot contain double quotes and cannot be longer than 35 characters.
- **BigFix upgrade to V11:**
  - You can upgrade BigFix server to V11 only manually.
  - After the upgrade, passwords containing non-ASCII characters are corrupted and users can no longer log in. In this case, before using the product, you must reset this type of password. When resetting the corrupted password, you can choose to use the same password or a new password that can have non-ASCII characters.
  - You might encounter issues upgrading a BigFix client having settings whose names or values contain non-ASCII characters. If custom settings with these names or values exist, check they are still valid after upgrading to V11.
  - After the upgrade, non-ASCII characters input in action message tabs might be garbled on Client user interfaces because of missing character sets, when the BigFix deployment encoding code page is different from the client code page. As a workaround, you can edit the `/usr/share/fonts/liberation/fonts.dir` file to remove references to fonts that do not exist, and correct the font count at the beginning of the file to match those actually found. It must also be noted that including the `xorg-x11-fonts-misc` font package might provide the missing character sets for certain localized installations.

- After the upgrade of the BigFix Server to V11, if you have settings with non-ASCII characters on clients earlier than V11, you might need to reset them.
- After upgrading BigFix to V11, the `filldb` process on clients might fail if the `filldb` buffer directory contains reports that use an unexpected report encoding value, for example `ReportEncoding: hp-roman8`. If so, clean up the `filldb` buffer directory containing the reports and then restart `filldb`.
- The License Key Password cannot contain double quotes and cannot be longer than 35 characters.

- **BigFix proxy agent:**

  Subscription to computer groups containing special characters in the name is not correctly evaluated by devices managed by the BigFix proxy agent. Avoid special characters in the computer group names.

- **BigFix interfaces:**
  - The BigFix interface generally allows only characters belonging to the FXF encoding (that is the value assigned to the `x-bes-fxf-charset`). For example, if a user tries to create a custom site whose name contains characters other than FXF encoding (suppose `x-bes-fxf-charset` parameter is Windows 1252, and the name of the site is `Site_albêrto`, where character ê is specific to the Windows 1250 encoding), the BigFix console displays an error message. The same error occurs when trying to create the custom site via REST API or Command-Line Interface (CLI). This rule does not apply to Operator Name and Fixlet Description that allow non-ASCII characters.
  - Operators with non-ASCII characters in the user name might have problems to log in through REST API. It is recommended that your REST API client uses the UTF-8 encoding format.
  - The BigFix APIs, including server, client, dashboard, and Web Reports, support only UTF-8 encoding.
  - Situations in which non-ASCII data is used might change behavior. For example, manually percent encoded data in an action script might be decoded assuming a different encoding than in previous client versions, resulting in a different action operation.

- **Files:**

  - The name of a custom site can contain only characters belonging to the FXF encoding (that is the value assigned to the `x-bes-fxf-charset` parameter). When a user creates a new custom site, BigFix console and REST APIs will block any input names that contain characters outside that set. If a client of the deployment has a local OS encoding other than the FXF, a local client user might not be able to display the name of the Subscribe or Unsubscribe files correctly.
  - Files that are added to custom sites must be ASCII named.

- **Linux and the Windows HTTP servers:**

  Both the Linux and the Windows HTTP servers now accept both raw UTF-8 encoded URLs as well as percent encoded UTF-8 URLs .

- **Client Relevance:**

  Client relevance supports characters in FXF encoding (the value assigned to the `x-bes-fxf-charset` parameter). Session relevance supports Unicode characters because it uses UTF-8 encoding with all interfaces (that is.BigFix console, WebReports, REST API, SOAP API). The following session relevance inspector, which allows you to get the FXF encoding of a BigFix Server is available:

  ```
  fxf character set of <bes server>: string
  ```

- **Key exchange:**

  The BigFix client can include passwords with ASCII characters only in its key exchange with the authenticating relay.

- **Log files:**
  - Product log files are always UTF-8 encoded. On platforms other than Windows, log paths and file names must contain ASCII characters only.
  - If you change the name or the path of the client log in the **_BESClient_EMsg_File** setting, to avoid character display problems, ensure that you use names that have only ASCII characters.

- **DB2 Administrative user password:**

  You can use only ASCII characters when setting the DB2 Administrative user password.

- **Download and upload:**

- Dynamic download of URL having non-ASCII characters fails on Windows clients. The URLs in the file should be percent encoded UTF-8 to avoid this kind of issue.
  For example the URL:

  ```
  /tmp/dòwnlòàd
  ```

  should be represented as:

  ```
  /tmp/d%C3%B2wnl%C3%B2%C3%A0d
  ```

- From Macintosh clients V9.2 or earlier, you cannot upload files with file name containing non-ASCII characters that are not contained in the server code page.
- You cannot download a file that contains non-ASCII characters in its name.
- You cannot use non-ASCII characters in file names to be downloaded.

- **Command prompt property:**

  Command prompt property whose font is set to Raster Fonts on the BigFix server workstation might generate problems in displaying characters. It is recommended to set the font to a value different from Raster Fonts.
- **Earlier clients:**

  A client BigFix V9.2 or earlier working on a code page different from the FXF code page might not support the file archive operation. When the name of an archived file has a character that does not exist on the FXF code page, the file is not available on the target directory after archive now.
- **Server system locale:**

  Do not change the system locale on the BigFix server to avoid compromising the server functionalities.
- **Web Reports server:**

  Web Reports cannot aggregate data sources that have different FXF encodings. For example, you cannot aggregate a Windows-1252 data source with a Shift-JIS data source within the same instance of a Web Reports server.
- **Fixlet Debugger (QnA) tool and QnA command line:**

- The Fixlet Debugger graphical breakdown window can only render 256 unique text characters. Relevance text characters outside the supported range will be replaced by an 'X' with inverted foreground and background colors.
- If you want to use non-ASCII characters in the QnA tool, ensure that the font in the command prompt properties of your workstation is set to a value compatible with non-ASCII characters, otherwise characters might not display correctly. Alternatively, use only ASCII characters.

- **Client settings:**

  Setting names on BigFix V10 UNIX and Linux clients cannot contain any of these three characters: "]" (right square bracket), "\" (backslash), "=" (equals) If you use any of these characters in a setting name, the Add Setting will appear to succeed but the setting will be either missing or corrupt. It is also possible that neighboring settings might be affected adversely.
- **BES Support:**
  - Fixlets are not properly displayed when BigFix was installed using the Korean language. In particular, the Take Action drop-down list is blank. As a workaround, either reinstall BigFix by selecting a language different from Korean and create a new masthead, or create a custom site and copy the tasks/Fixlets/ analyses from the BES Support site to the custom site.
  - If the task **2283-WARNING: BES Client has local codepage limiting content** is relevant for a client, the local codepage of the client has some limitations and might be able to understand only ASCII as the codepages are not in common with the codepage of the server. A simple restart of the client service could address the issue. Because on some platforms the default installation is only available in ASCII (AIX for example) , as a workaround you should install the Unicode language functionality.

Reading and writing files in the specific encodings

BigFix agent uses UTF-8 as the internal representation of strings.

When data is written to a file or read from a file, text is transcoded from the local encoding to UTF-8 or viceversa, unless the file is already UTF-8. Using the encoding inspector, you can

now specify to read and write a file in a specific encoding. You can use the inspector in an action or in a relevance expression:

- To read a file in a specific encoding you can specify the encode in the `file content`, `file line` and `file section` inspectors.
- To write a file in a specific encoding you can specify the `action uses file` encoding command along with the `appendfile` and `createfile until` commands.

Background information

When files are written, by default, the BigFix agent assumes that the encoding is the local one.

On a client with local encoding different from UTF-8 (like Windows), text data is transcoded from UTF-8 to the local encoding. Using the `encode` inspector you can now specify what encoding must be used to read and write files.

When files are read, if the Byte Order Mark (BOM) is set, file encoding is determined based on the BOM. Otherwise, the content of file is assumed to be in the local encoding.



Note: Inspectors and the Action Script language no longer handle binary file contents, because file is transcoded to or from UTF-8.

Reading file inspectors

You can use the inspector object `encoding` to specify an encoding to be used to read file in a relevance expression.

If you don't specify any encoding, the files are read in the local encoding. The `encoding` object is used to read a `file` as the following:

```
file "filename" of encoding "encoding"
```

The *`encoding`* might be any name which ICU can recognize, such as `ISO-8859-1`, `Shift_JIS`, and `UTF-8`.

Using this `encoding` object, you can affect the behaviors and results of relevance expressions using the following objects:

- file content
- file line
- file section

Here some simple examples:

```
(content of file "c:\aaa\bbb.txt" of encoding "Shift_JIS")
    contains "??"
    # Return if the word "??" is found in the file "c:\aaa\bbb.txt" that
      is written in Shift_JIS

    line 3 of file "eee.txt" of folder "/ccc/ddd" of encoding
"Windows-1252"
    # Return the third line of the file "/ccc/ddd/eee.log" in Windows-1252

    lines of file "/fff/ggg.txt" of encoding "UTF8"
    Return the lines of the file "/fff/ggg.txt" in UTF8

    lines of file "/hhh/iii.txt" of encoding "ISO-8859-1"
    Return the lines of the file "/hhh/iii.txt" in ISO-8859-1
```

You can use the `encoding` object by adding it after the keywords listed below to create `file` objects:

- file
- folder
- download file
- download folder
- find file <string> of <folder>
- x32 file (Windows only)
- x32 folder (Windows only)

- x64 file (Windows only)
- x64 folder (Windows only)
- native file (Windows only)
- native folder (Windows only)
- symlink (Unix only)
- hfs file (Mac only)
- posix file (Mac only)
- hfs folder (Mac only)
- posix folder (Mac only)

The `encoding` object cannot be used with creation methods for Mac's special folders such as `apple extras folder`, or `application support folder`. For such folders, you can uses the `folder` object by specifying their paths.



Note: If you try to open a file with an encoding using the `encoding` object and the file has a BOM, the file is opened in the encoding indicating the BOM; that is, the specified encoding is ignored.



Note: If, for whichever reason, the BOM of the file does not reflect the encoding of its content, the file line inspector fails with the `U_INVALID_CHAR_FOUND` error.



Important:

The `file` objects must be evaluated as a property of the `encoding` object during its creation. You cannot specify any encoding to `file` objects which are already created in the relevance expression:

```
(file "aaa.txt" of folder "c:\test") of encoding "Windows-1252"
    # Not work. The encoding will be ignored.
```

In this relevance expression, the file `C:\test\aaa.txt` is read in the local encoding, not the Windows-1252 encoding, because a file object representing `C:\test`

`\aaa.txt` is created first with the expression enclosed parenthesis and the subsequent `encoding` expression is ignored.

In the following expression the file `C:\test\aaa.txt` is read in the Windows-1252 encoding:

```
file "aaa.txt" of folder "c:\test" of encoding "Windows-1252"
```

Writing file with the encode command

You can use the Action Script command `action uses file encoding` to specify the encoding in which to write files when using the appendfile and createfile until commands.

The encoding is effective until another encoding is specified. If you do not use the `action uses file encoding` command, the `appendfile` and `createfile` commands create files in the local encoding.

The command syntax is:

```
action uses file encoding encoding [ NoBOM ]
```

The encoding might be any name which ICU can recognize, such as `ISO-8859-1`, `Shift_JIS`, and `UTF-8`. After created, the `file` objects can be used as regular `file` objects and you can apply any operations applicable to text files.

To turn off the encoding change and reuse the local encoding, you can set the encoding keyword to `local`.

If any of the UTF encodings (UTF-8, UTF-16, or UTF-32) is specified as the value of encoding, the file to be created will have a BOM (Byte Order Mark) at the head of it. If the client local encoding is UTF-8 and no encoding is specified in an action, files to be created with the action will be written in UTF-8 without BOM.

To suppress adding any BOM, you can use the option `NoBOM` (case-insensitive) following the value of encoding. The `NoBOM` option is effective only with any UTF encodings (UTF-8, UTF-16, and UTF-32), and it is ignored if it is used with any other encoding name.

The following action creates a files using the Windows-1253 (Greek) encoding:

```
delete "{(client folder of current site as string) & "/__appendfile"}"
action uses file encoding Windows-1253
appendfile Κόκκινο ου?ανό τη ν?χτα
delete C:\encode_test.txt
move __appendfile C:\encode_test.txt
```

The following action creates two files the first using the Windows-1253 (Greek) encoding, the second using the local encoding:

```
delete "{(client folder of current site as string) & "/__appendfile"}"
appendfile Following lines contains Greek language strings
action uses file encoding Windows-1253
appendfile Κόκκινο ου?ανό τη ν?χτα
move __appendfile C:\Greek_test.txt
// switch to local encode
delete "{(client folder of current site as string) & "/__appendfile"}"
appendfile Following lines contains English strings
action uses file encoding local
appendfile  Am I writing a local US strings now !
delete C:\tmp\local_test.txt
move __appendfile C:\tmp\local_test.txt
```

The following action creates a file using the UTF-8 encoding without a BOM:

```
delete "{(client folder of current site as string) & "/__appendfile"}"
action uses file encoding UTF-8 noBOM
appendfile Hello world !!
delete /tmp/encode_test.txt
move __appendfile /tmp/encode_test.txt
```

Reading and writing files with encode

Using the encode inspectors, you can also read from and write to files, having different encoding.

The following example shows an action that reads the first line of a file having Windows-1253 (Greek) encoding, and writes it into a file having Windows-1252 (English) encoding:

```
delete "{(client folder of current site as string) & "/__appendfile"}"
action uses file encoding Windows-1253
appendfile
{
line 1 of file "/tmp/Greek.txt" of encoding "Windows-1253"
}
delete "/tmp/encode.txt"
move __appendfile /tmp/encode.txt
```

Managing actions on clients with different local encoding

BigFix Platform introduces the possibility to specify the names of files and folders of UNIX clients in any encoding, even if it is different from the encoding used by the BigFix server.

You can do this by specifying the corresponding hexadecimal representation (string) of the file and folder names, in a BigFix action. A set of commands that use binary strings is now available for this purpose. For example, to create a new folder having the Japanese name "`ハロ`" in UTF-8 encoding, you can submit an action script from the BigFix console by specifying the corresponding hexadecimal value "`e3838fe383ad`" as follows:

```
binary name folder create “e3838fe383ad"
```

With BigFix 9.5.5 or earlier, you can only use characters in FXF character set (character set matches to your BigFix server locale) to create files, and characters in local character set when retrieving names.

With BigFix 9.5.6, you can specify binary names and folders but not all binary actions are fully supported.

With BigFix 9.5.7 and later versions, all binary actions are fully supported.

In addition to the **binary name folder create** command used in the example, depending on the actions to be completed on the client, you can use a set of commands that are documented on BigFix Developer site: https://developer.bigfix.com/relevance/reference/binary_string.html.

Chapter 3. Sample deployment scenarios

The following deployment scenarios illustrate some basic configurations taken from actual case studies.

Your organization might look similar to one of the examples below, depending on the size of your network, the various bandwidth restrictions between clusters and the number of relays and servers. The main constraint is not CPU power, but bandwidth.

Pay careful attention to the relay distribution in each scenario. Relays provide a dramatic improvement in bandwidth and should be thoughtfully deployed, especially in those situations with low-speed communications. Relays are generally most efficient in fairly flat hierarchies. A top-level relay directly eases the pressure on the server, and a layer under that helps to distribute the load. However, hierarchies greater than two tiers deep might be counterproductive and must be carefully deployed. Multiple tiers are generally only necessary when you have more than 50 relays. In such a case, the top tier relays would be deployed on dedicated servers that would service from 50-200 second-tier relays. The following examples help you deploy the most efficient network layout.

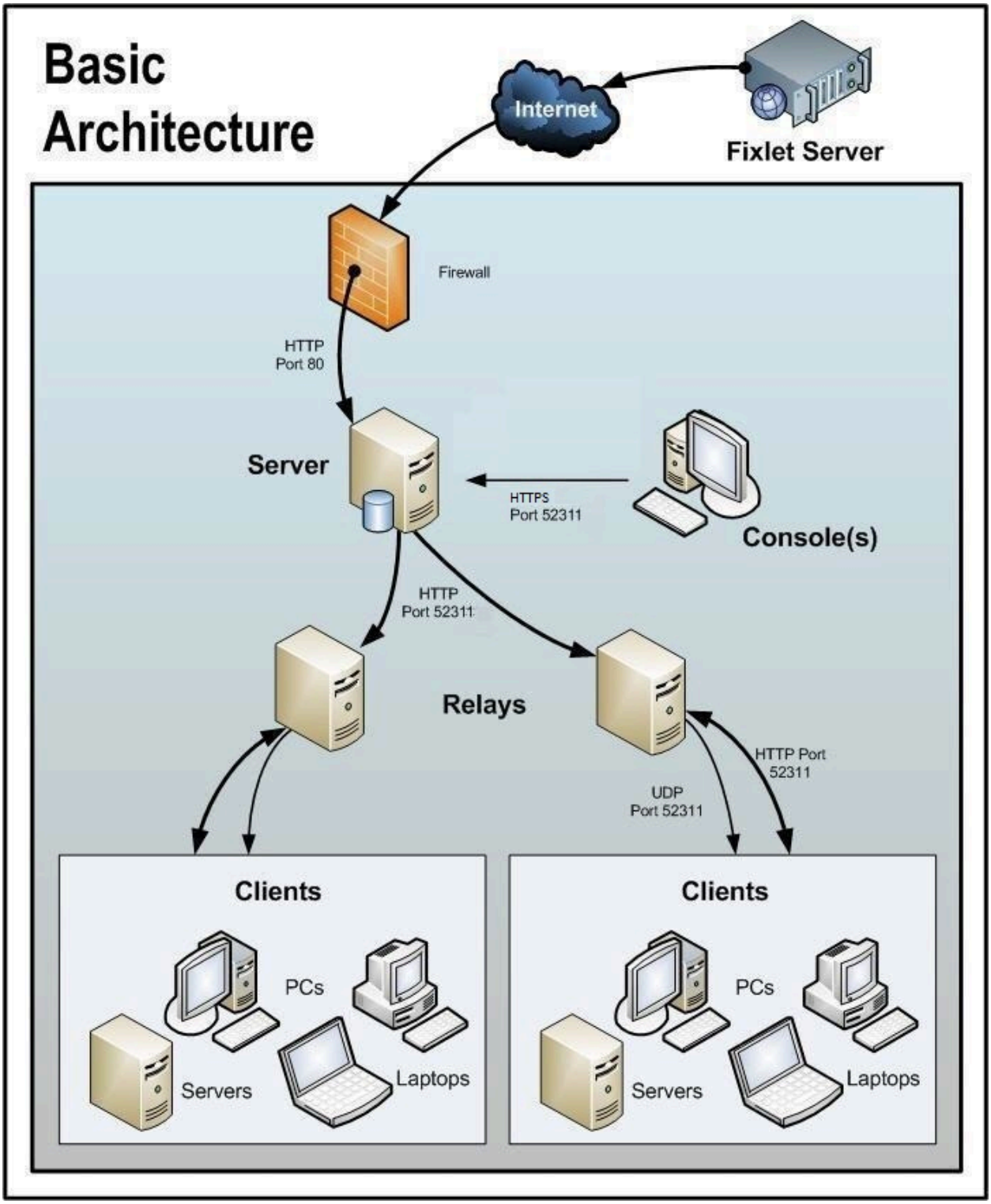
Note that additional servers can also add robustness to a network, by spreading the load and supplying redundancy. Using redundant servers allows failback and failover to be automated, providing minimal data loss, even in serious circumstances.

With the correct deployment of servers and relays, networks of any size can be accommodated. Beyond the examples shown here, your HCL support technician can help you with other configurations.

Basic deployment

A simplified BigFix deployment, that points out the basic hierarchy and the ports used to connect the components, is shown in the following diagram.

There is at least one server that gathers Fixlets from the Internet where they can be viewed by the console operator and distributed to the relays. Each client inspects its local computer environment and reports any relevant Fixlets back to the relay, which compresses the data and passes it back up to the servers.


Basic Architecture
Internet
Fixlet Server
Firewall
HTTP
Port 80
Server
HTTPS
Port 52311
Console(s)
HTTP
Port 52311
Relays
HTTP Port
52311
UDP
Port 52311
Clients
PCs
Servers
Laptops
Clients
PCs
Servers
Laptops

The BigFix console oversees all this activity. It connects to the Servers and periodically updates its displays to reflect changes or new knowledge about your network.

The BigFix console operator can then target actions to the appropriate computers to fix vulnerabilities, apply configuration policies, deploy software, and so on. The progress of the actions can be followed in near realtime as they spread to all the relevant computers and, one by one, address these critical issues.

This diagram labels all the default ports used by BigFix, so that you can see which ports need to be open and where. These ports were selected to avoid conflict, but if you are currently using any of these ports, they can be customized upon installation.

Note: The arrows in the diagram illustrate the flow of information throughout the enterprise. The arrows from the Fixlet server to the servers represent the flow of Fixlets into your network. Clients gather Fixlets and action information from relays. They then send small amounts of information back to the servers through the relays. The UDP packets from the relay to the clients are small packets sent to each client to inform them that there is new information to be gathered. The UDP messages are not strictly necessary for BigFix to work correctly. View the network traffic article at the BigFix support site, or ask your support technician for more details.

Note the following about the diagram:

- Port 80 is used to collect Fixlet messages over the Internet from Fixlet providers such as HCL.
- A dedicated port (defaulting to 52311) is used for HTTP communications between servers, relays, and Clients.
- A dedicated port (defaulting to 52311) is used for HTTPS communications between servers and Consoles.
- Relays are used to share the server load. This diagram only shows two relays, but you can use dozens or even hundreds of relays in a similar flat hierarchy. Typically a Relay is deployed for every 500-1,000 computers.

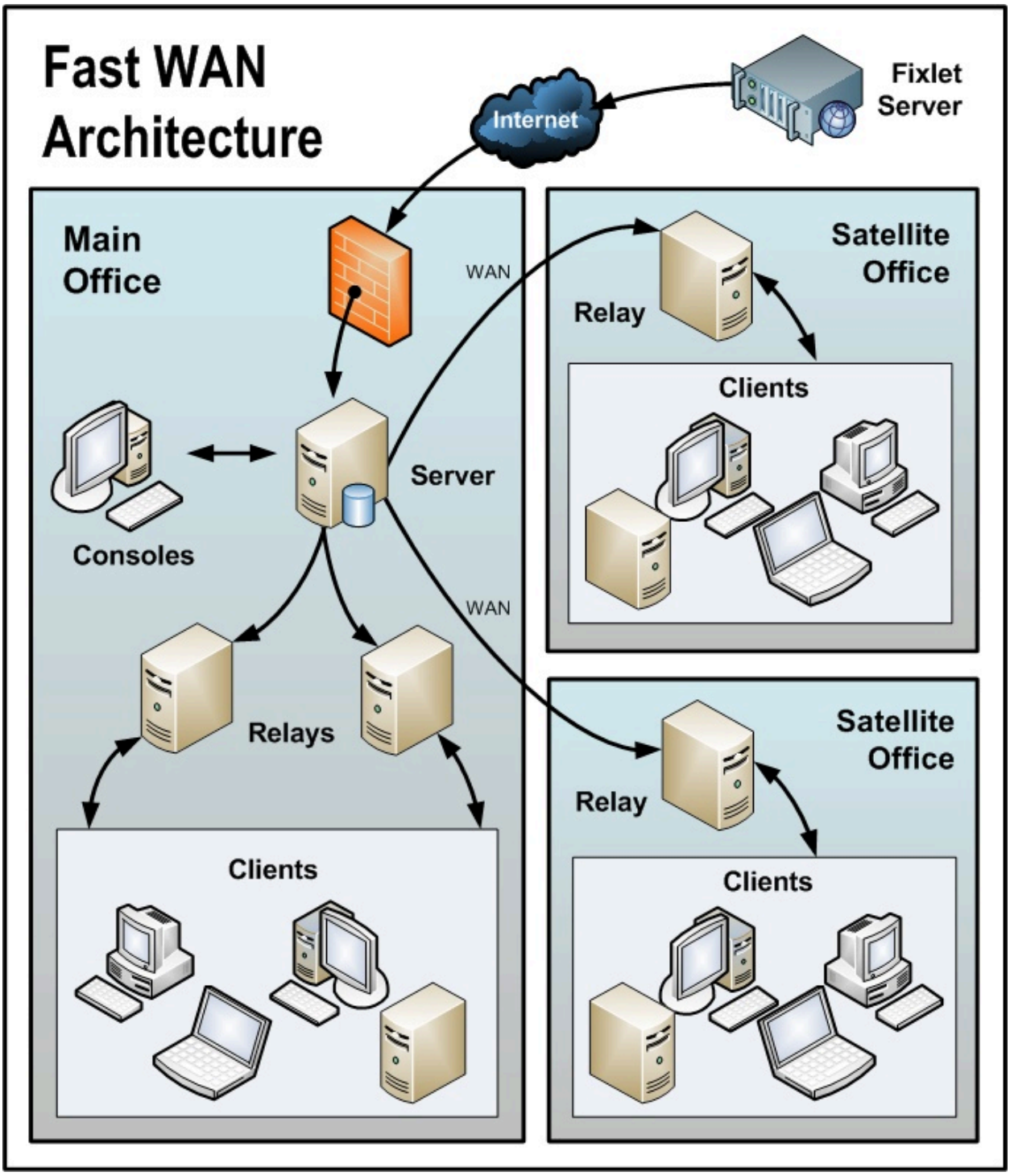
- The BigFix relays can also take advantage of a UDP port to alert the Clients about updates, but this is not strictly necessary.
- The BigFix Clients are typically PCs or Workstations, but can include other servers, dockable laptops, and more. Any device that can benefit from patches and updates is a candidate to include in the deployment.

BigFix has far greater flexibility and potential than this simple case suggests. It is capable of overseeing hundreds of thousands of computers, even if they are spread out around the world. The next scenarios build on this basic deployment.

Main Office with Fast-WAN Satellites

This configuration is common in many universities, government organizations, and smaller companies with only a few geographical locations.

This type of deployment is relatively easy to set up and administer because there are no (or very few) slow WAN pipes to consider.



Note the following about the diagram:

- In this configuration, the relays are used both to relieve the server and to distribute the communications, optimizing the bandwidth.
- This scenario has large WAN pipes, so office relays can communicate directly with the main server. A thin WAN could force a change in the layout of the relays (see the scenarios above and below).
- The more relays in the environment, the faster the downloads and response rates.
- Because of the nature of this network, when the clients are set to **Automatically Locate Best relay**, many of the relays are the same distance away. In this scenario, the clients automatically load-balance themselves amongst all the relays that are nearby.
- For this high-speed LAN, a relatively flat hierarchy is recommended, with all relays reporting directly to the main server. Any extra levels in the hierarchy would only introduce unnecessary latency. However, if there were over 50-100 relays in this environment, another level of relays should be considered.

Disaster Server Architecture

Companies with sensitive or high availability needs might want to deploy multiple, fully-redundant servers to maintain continuous operation even in the event of serious disruptions. BigFix includes the important ability to add multiple, fully redundant servers: a feature called Disaster Server Architecture (DSA).

Each server maintains a replica of the BigFix database and can be positioned anywhere in the world. In the case of a network fracture, these servers continue to provide uninterrupted service to the local network. As soon as the connection is reestablished, the servers automatically reconnect and sync up. The BigFix relays and clients are also capable of successfully recovering from such a disconnect. DSA provides the following capabilities:

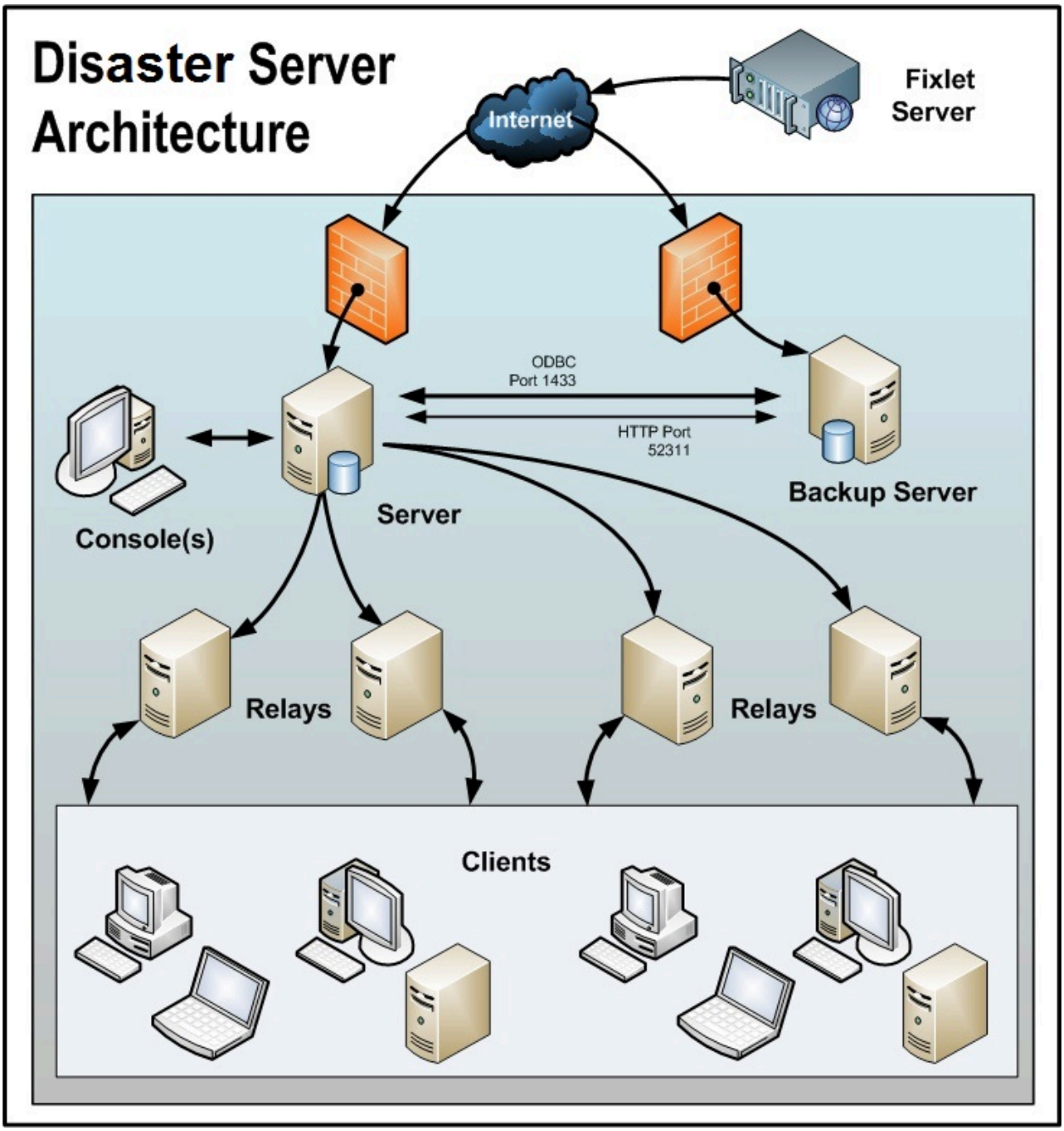
- Continued service availability on both sides of a network split (automatic failover).
- Continued availability in the event of a server outage.
- Distribution of console database load during normal operation.
- Automatic failback upon reconnecting.

To take advantage of this function, you need one or more additional servers with a capability at least equal to your primary server. All the BigFix servers in your deployment must run the

same version of SQL Server. If your existing Server is running SQL 2016, your new servers must run SQL 2016 as well.

For more information about using server redundancy, see Using multiple servers (DSA).

Multiple servers also help to distribute the load and create a more efficient deployment. Here is a simple diagram of how multiple servers might be set up to provide redundancy:



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.

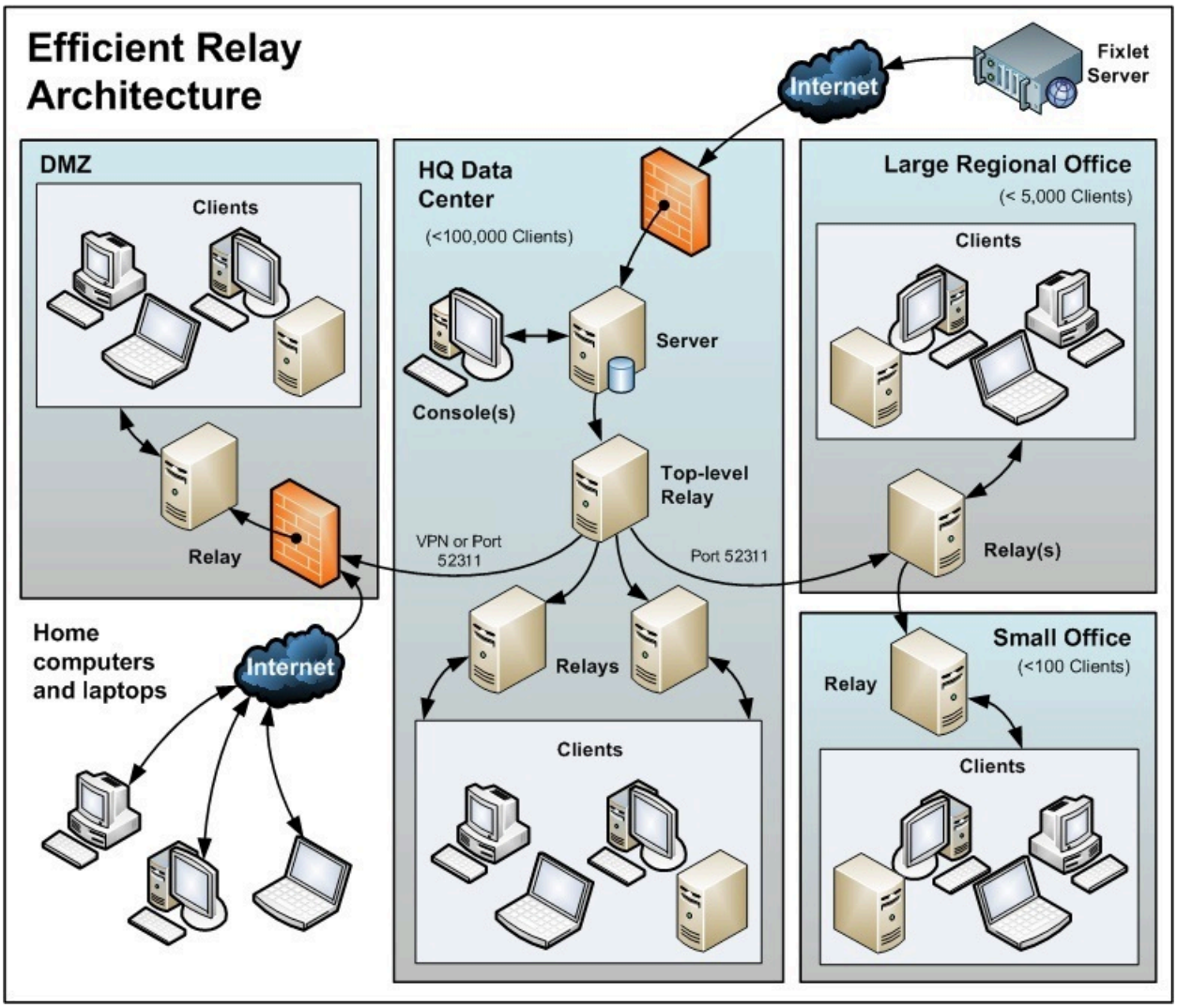
Note the following about the diagram:

- The BigFix servers are connected by a fast WAN, allowing them to synchronize several times per hour.
- The servers need both an ODBC and an HTTP link to operate and replicate properly.
- There is a primary server with an ID of 0 (zero). It is the first server that you install, and it is the default server for running the BigFix Administration Tool.
- For the sake of clarity, this is a minimal configuration. A more realistic deployment would have a top-level relay and other WAN connections to regional offices.
- The BigFix servers and relays are configured so that control can be automatically routed around a server outage (planned or otherwise), and upon failover reconnection, the databases are automatically merged.
- The BigFix servers communicate on a regular schedule to replicate their data. You can review the current status and adjust the replication interval through BigFix Administration > Replication. For the best possible performance, these pipes should be FAT.
- This diagram only shows two servers, but the same basic architecture would apply to each additional server. With multiple servers, a shortest-path algorithm is used to guide the replication.
- 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 on failover, precedence goes to the version on the server with the lowest server ID.

Efficient relay setup

To increase efficiency and reduce latency, this company has set up a hierarchy of relays to help relieve the server load.

Each relay they add takes an extra burden off the server for both patch downloads and data uploads. Setting up relays is easy, and the clients can be set to automatically find the closest relay, further simplifying administration.



Note the following about the diagram:

- There is a dedicated server computer known as the Top-Level relay that is used to take the load off the server computer.
- All relays are manually configured to point to either the top level relay or to another relay that is closer. The general rule for configuring relays is that you want as

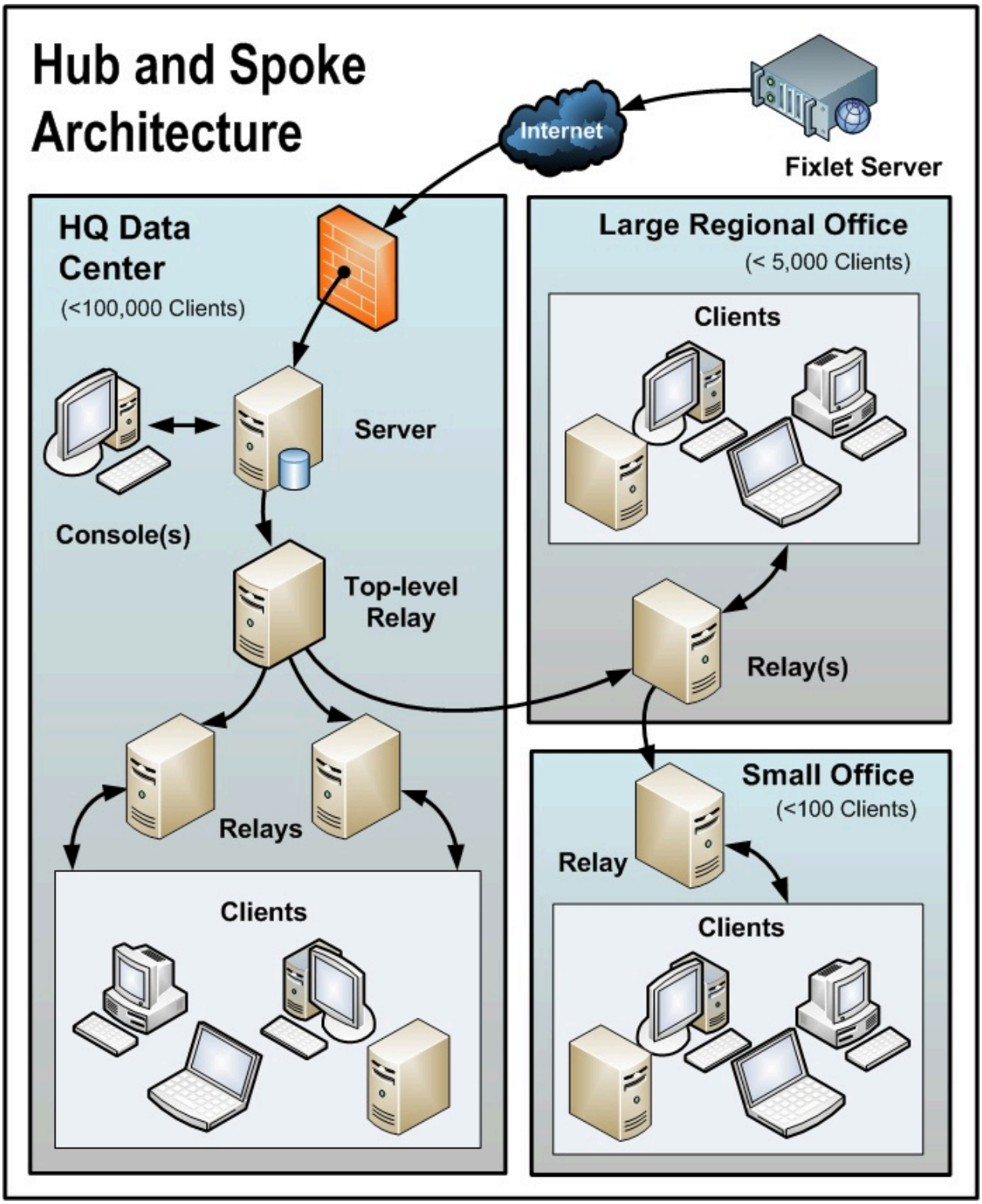
few levels as possible to the relays unless there is a bandwidth bottleneck. Communications over thin pipes should be relay to relay. The top-level relay relieves the server, and the secondary relay allows a single download to be distributed over hundreds of clients.

- There is a relay in the DMZ set up with a special trust relationship with the server. This relay allows clients in the DMZ or on the public Internet to be managed by BigFix. The DMZ places a security firewall between the relay and the set of home computers and laptops reporting in from the Internet. In the DMZ scenario, an office connected by a VPN can share the content from a relay or server.
- In the direct connection, home PCs and laptops can connect directly to the Internet for content from Fixlet servers through their own private firewalls. For the sake of clarity, these extra connections might not be shown in all diagrams, but they are generally present in most deployments.
- This diagram shows a single relay in the large regional office. However, for offices with more than a few hundred clients, there will typically be multiple relays to effectively distribute the load.
- As a general rule, you should deploy at least one relay per 500-1000 clients to maximize the efficiency of the relay.

Hub and spoke

This scenario involves a main data center, a small number of large regional offices, and many small regional offices.

This configuration is common in large international organizations. The BigFix clients are installed on computers in offices all around the world. Many of these locations have slow WAN connections (8 kbps-512 kbps), but there are many offices with faster WAN connections (1mbps-45mbps).


Hub and Spoke Architecture
Internet
Fixlet Server
HQ Data Center
(<100,000 Clients)
Server
Console(s)
Top-level Relay
Relays
Clients
Large Regional Office
(< 5,000 Clients)
Clients
Relay(s)
Small Office
(<100 Clients)
Relay
Clients

Often these locations are configured in a hub-and-spoke arrangement. This scenario builds on the previous one, but the hub-and-spoke configuration permits more levels in the relay hierarchy.

Note the following about the diagram:

- In this scenario, the relays are carefully deployed at the proper junctions within the WAN to optimize bandwidth. Poor placement of relays can adversely impact your network performance.
- It is vital that at least one relay is installed in every location with a slow WAN connection. Often a company already has a server in just such a location, acting as a file server, print server, AV distribution server, SMS distribution server or domain controller, or any other computer. The BigFix relay is usually installed on these existing computers.
- To provide redundancy in a typical office, more than one relay should be installed. If a relay fails for any reason (powered down, disconnected from the network, and so on.), its attached clients can then automatically switch over to a different relay. A redundant relay is less important in very small offices because fewer computers are affected by the failure of a relay.
- When the clients are set to **Automatically Locate Best Relay**, they will choose the closest one. If any relay fails, the clients automatically seek out another relay. You should monitor the relay configuration after the initial automated setup (and periodically after that) to ensure that the clients are pointing to appropriate locations. Talk to your support technician for more details about how to protect against overloading WAN pipes with BigFix data.
- Bandwidth throttling at the relay level is very helpful in this configuration. The BigFix relays are set up to download slowly across the WAN pipes so as not to saturate the slow links.
- Instead of pointing to the main server, the relays are configured to point to the top level relay. This frees up the server to couple more tightly to the console and improves reporting efficiency.

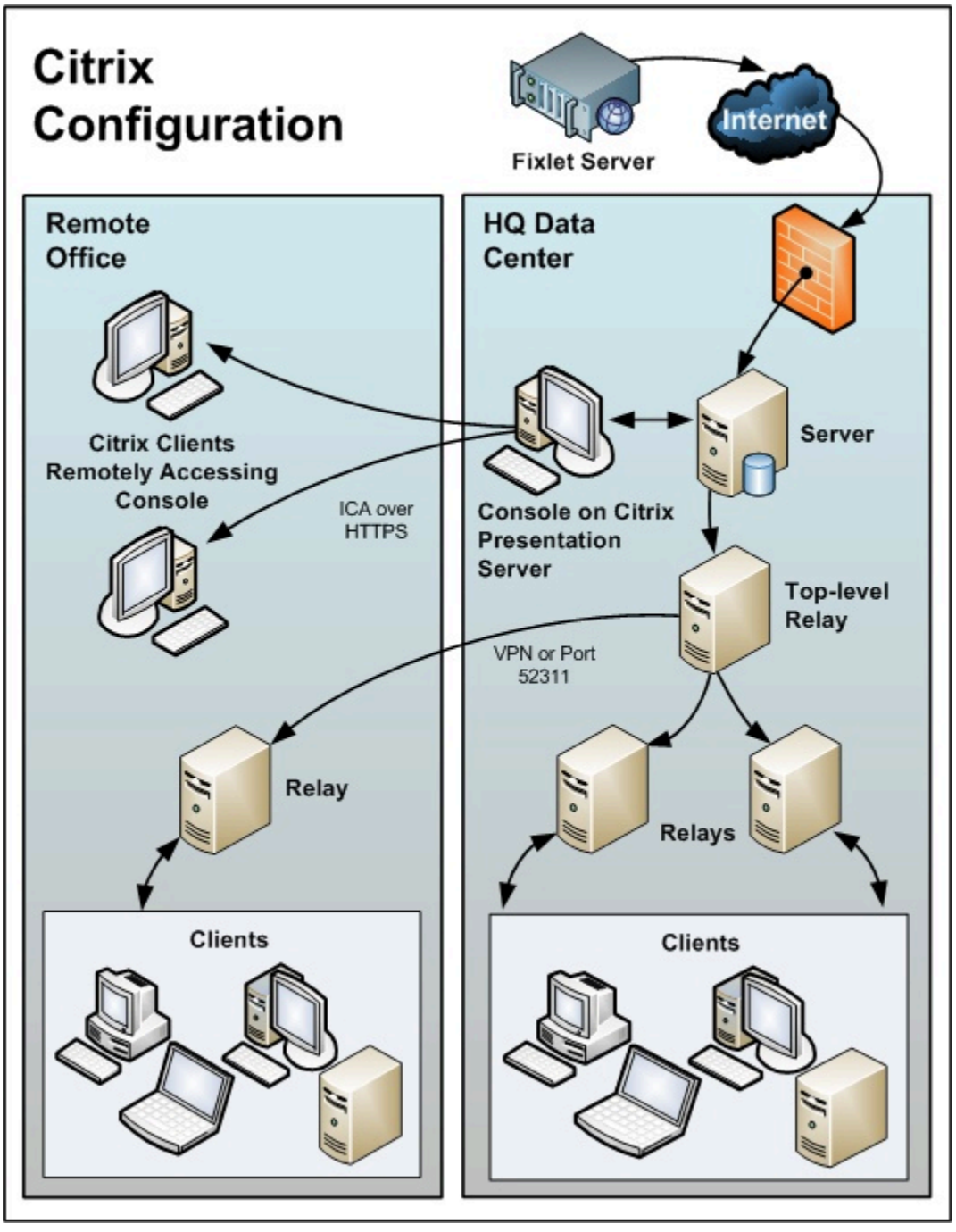
The BigFix relays are configured to manually create the optimal hierarchy. The hierarchy has three levels (from the top down):

1. The top-level relay that connects directly to the server.
2. The regional office relays that connect to the top-level relay.
3. Multiple branch office relays that connect to specified regional office relays.

Remote Citrix / Terminal Services Configuration

Although BigFix can efficiently deliver content even over slow connections, the console itself is data-intensive and can overwhelm a link slower than 256 kbps. Adding more Clients further increases the lag time.

However, you can access the console remotely from a Citrix, Windows Terminal Server, VNC or Dameware-style presentation server and realize excellent performance. Here is what this configuration looks like:



Note the following about the diagram:

- In the main office, the console is set up on a computer that is close to the server for fast data collection. This is your Presentation server.
- You must create user accounts for each remote user. These users can then access the console quickly because the time-critical data loading is done at the main office over a fast link.
- Your remote connection can be over HTTPS to improve security.
- Note that running a console from a Presentation server containing the private key is inherently less secure than if the key is stored on a removable drive.
- You might be able to benefit from load-balancing software to spread the remote accesses across multiple servers.
- The main bottleneck for a console running on Citrix is memory size. If the console runs out of memory, its performance decreases sharply. A good technique to determine the memory requirement is to open the console as a Master Operator. Check the memory used: this indicates the maximum memory requirement per user. Then log in as a typical operator and use this as your average memory requirement. If your Citrix server can support all concurrent users with the maximum memory then a single box suffices. If not, then use the average memory requirement per user to determine how many extra Citrix servers you might need.
- The second constraint is CPU power. During refreshes, the console works best with a full CPU core. This means the Presentation server will be optimized with one CPU core running the console for each concurrent user.
- The final concern is disk space for the console cache. You can understand the size of the cache by looking at an example on your local computer: C:\Documents and Settings\<USERNAME>\Local Settings\Application Data\BigFix\Enterprise Console \BES_bfenterprise. There should be enough disk space to provide one cache file for each console operator.

Chapter 4. Requirements and assumptions

BigFix runs efficiently using minimal server, network, and client resources.

The hardware required by the server and the console depends on the number of computers that are administered and the total number of consoles. The distributed architecture of BigFix allows a single server to support hundreds of thousands of computers.

Server requirements

To find the latest information about the server requirements, see BigFix Support Matrix.

Notes about Windows operating systems:

- Only the 64-bit architecture is supported for installing the BigFix server and Web Reports components on Windows systems.
- The Windows firewall can be either turned off or configured to open the following two ports:
  - Port `52311` for UDP and TCP/IP
  - Port `8083` for Web Reports and TCP/IP
- Windows BigFix servers cannot be migrated to Linux BigFix servers.

Notes about Linux operating systems:

- Red Hat packages and all their prerequisites required by BigFix Linux server and Web Reports components:
  - `cyrus-sasl-lib.x86_64`
  - `libstdc++.x86_64`
  - `pam.x86_64`
  - `krb5-libs.x86_64` (minimum version 1.15)
  - `unixODBC.x86_64` (for installations with DB2 only)
  - `msodbcsql17.x86_64` (for installations with MS SQL only)
  - `fontconfig.x86_64` (Web Reports only)
  - `libXext.x86_64` (Web Reports only)
  - `libXrender.x86_64` (Web Reports only)

- `libpng12.x86_64` (RHEL 7 and 8 only)
- `libpng15.x86_64` (RHEL 9 only)

Installation or upgrade of the Server components on Linux installs the unixODBC RPM package which is a prerequisite for installations with DB2.

Installation of the Server components on Linux installs the msodbcsql17 RPM package and, if missing, its repository. The package is a prerequisite for installations with MS SQL.

If the package manager repository is not configured on the Linux system, you must manually install the unixODBC or the msodbcsql17 RPM package prior to running the installation or upgrade.

Notes about Linux operating systems with DB2:

- IBM DB2 11.5 / 11.5.4 / 11.5.5 / 11.5.6 / 11.5.7 / 11.5.8 / 11.5.9. For information about how to install DB2 server on Red Hat Enterprise Linux Server 64-bit, see Installing and configuring DB2 *(on page 190)*.
- Red Hat Enterprise Linux 8 x86 64-bit requires IBM DB2 11.5.4.

Notes about Linux operating systems with MS SQL:

- MS SQL Server 2019 / 2022. For information about how to install MS SQL server on Red Hat Enterprise Linux Server 64-bit, see Installing and configuring MS SQL *(on page 192)*.

Disk space and other requirements

For details, see BigFix Performance & Capacity Planning Resources.

Console requirements

For details about the console requirements, see BigFix Support Matrix.

The BigFix console can be installed on a laptop or any moderately-powerful computer. However, as the number of computers that you are managing with the console increases, you might need a more powerful computer.

The BigFix console also requires a high bandwidth connection (LAN speeds work best) to the server due to the amount of data that needs to be transferred to the console. If you need to remotely connect to the server across a slow bandwidth connection, it is recommended that you use a remote control connection to a computer (such as a Citrix server or Terminal Services computer) with a high-speed connection to the Server.

Contact your support technician for more information about console scaling requirements.

Note: The console is the primary interface to BigFix and manages a great deal of information about the clients. If the console computers are underpowered or on a slow connection, it can adversely impact performance.

Client requirements

To find the latest information about the client requirements, see BigFix Support Matrix.

If you are using BigFix Inventory, before storing scan uploads, see Hardware requirements for the client.

AIX

Starting from Patch 1, on AIX systems, ensure that your AIX 7.1 TL5, 7.2 TL5 or 7.3 installation satisfies the following prerequisite conditions of packages installed:

- xlC.rte >= version 16.1.0.10
- if libc++.rte version < 17.1.0.0, then libc++.rte must be >= version 16.1.0.10
- if libc++.rte >= 17.1.0.0, then libc++.rte, libc++abi.rte and libunwind.rte must be >= 17.1.1.4

These filesets are contained in the latest fix pack released by IBM.

Windows

On Windows systems, ensure that the BESClient service runs as the SYSTEM account.

Linux

On Red Hat Enterprise Linux™ 7 or later, ensure that you have installed the Athena library (libXaw package) before installing the client.

On Red Hat Enterprise Linux™ 9, ensure that you have installed the initscripts package before installing the client.

On Oracle Enterprise Linux™ 9, ensure that you have installed the initscripts package before installing the client.

On Rocky Linux™ 9, ensure that you have installed the initscripts package before installing the client.

On SUSE Linux Enterprise Server (SLES) 11 PPC64, ensure that you have installed the rpm-32bit package before installing the client.

On SUSE Linux Enterprise Server (SLES) 15 x86_64, ensure that you have installed the insserv-compat rpm package before installing the client.

On OpenSUSE Leap 15.4 x86-64, ensure that you have installed the sysvinit-tools rpm package before installing the client.

Mac

On Mac OS Mojave Version 10.14 or later, some default security settings restrict access to certain folders in the user's library which in turn might affect the behavior of custom content.

Inspectors interacting with files and directories associated with the following artifacts are impacted:

- Location
- Contacts
- Photos
- Calendar
- Reminders
- Camera
- Microphone

- Mail database
- Message history
- Safari data
- Time Machine backups
- iTunes device backups
- Locations and routines
- System cookies

To avoid access-related issues, do the following steps:

1. Go to the **Privacy** pane of **Security & Privacy** preferences.
2. Select **Full Disk Access**.
3. Add the BESAgent application.

Disk space requirements

Ensure that your targets have enough disk space before you start installing the BigFix client. The following table displays the disk space requirements for installing the BigFix client on the different operating systems.

OS	Directory	Space required	Description
Windows	C:\Program Files (x86)\BigFix Enterprise\BES Client	50 MB	Client Data and Installation Directories
Linux Intel & zLinuz	/var/opt/BESClient	20 MB	Client Data Directory
	/opt/BESClient	75 MB	Client Installation Directory
Linux PPC	/var/opt/BESClient	20 MB	Client Data Directory
	/opt/BESClient	100 MB	Client Installation Directory

OS	Directory	Space required	Description
AIX	/var/opt/BESClient	20 MB	Client Data Directory
	/opt/BESClient	115 MB	Client Installation Directory
Solaris	/var/opt/BESClient	20 MB	Client Data Directory
	/opt/BESClient	85 MB	Client Installation Directory
Mac	/Library/Application Support/Bigfix	45 MB	Client Data Directory
	/Library/BESAgent	50 MB	Client Installation Directory

Database requirements

The database stores all the data retrieved from the clients. Before installing the BigFix server, ensure that the database requirements are met.

A pre-upgrade check Fixlet is available to perform a set of checks to verify if the BigFix server can be successfully upgraded to V11. A log file is created in the BigFix server directory containing details about the executed steps. If these checks fail, a `preupgrade-Version 11 <datetime>.err` log file is created in the BigFix server directory. If these checks end successfully, a `preupgrade-Version 11.out` log file is created in the BigFix server directory. X is the modification level. This task is not relevant only when all the checks are completed successfully.

- The BigFix server on Windows systems supports the following configurations:
  - Local or remote Microsoft SQL Server 2014, 2016, 2017, 2019, 2022 or 2025 or the following remote databases:
    - Google Cloud SQL for Microsoft SQL Server 2022 database
    - Azure SQL Managed Instance for Microsoft SQL Server 2022 database
    - Amazon RDS for Microsoft SQL Server 2022 database.
    - Azure SQL Managed Instance for Microsoft SQL Server 2025 database
    - Google Cloud SQL for Microsoft SQL Server 2025 database

    The following table shows details about the new databases supported on BigFix Server on Windows systems:

Database	Product minimum version	Notes
Microsoft SQL Server 2019 or 2022 (Enterprise and Standard are recommended)	11.0.0	
Microsoft SQL Server 2025	11.0.6	
Google Cloud SQL for Microsoft SQL Server 2022	11.0.2	▪ Only the SQL Server authentication is supported. ▪ DSA configurations are not supported.

Database	Product minimum version	Notes
Google Cloud SQL for Microsoft SQL Server 2025	11.0.6	▪ Only the SQL Server authentication is supported. ▪ DSA configurations are not supported.
Amazon RDS for Microsoft SQL Server 2022	11.0.4	▪ Only the SQL Server authentication is supported. ▪ DSA configurations are not supported. ▪ **Important note:** On Amazon RDS for Microsoft SQL Server 2022, the SQL Server authentication user must be granted the server roles: processadmin, public, and setupadmin. The same user must also have db_owner privileges on the

Database	Product minimum version	Notes
		BFEnterprise and BESReporting databases.
Azure SQL Managed Instance for Microsoft SQL Server 2022	11.0.4	▪ Only the SQL Server authentication is supported. ▪ DSA configurations are not supported.
Azure SQL Managed Instance for Microsoft SQL Server 2025 database	11.0.6	▪ Only the SQL Server authentication is supported. ▪ DSA configurations are not supported.

Note: When installing BigFix in a production environment, Microsoft SQL Server Enterprise and Standard are recommended. The Express edition should never be used for production deployments. Refer to Microsoft documentation to choose the best database for your deployment.

Important: When installing or upgrading BigFix, the user account performing the installation or upgrade must have `sysadmin` server role in SQL Server. When working with a remote instance of SQL Server, the AD domain service account must have dbo rights, that is `db_owner` on the BFEnterprise and BESReporting databases. Start with `sysadmin` server

role to perform the installation or upgrade, then back off the privileges to `db_owner` after the product is up and running.

Important: When working with SQL Server, ensure that you satisfy the following prerequisites for the Microsoft SQL database collation:

- The database collation must be case insensitive.
- The database collation at the server, database and column level must be set to the same value.

The database collation can be used unless it is mentioned in the Known limitations and workarounds (on page 647) page.

To verify the collation, run the following SQL Server queries:

At the SQL server instance level

```
SELECT ServerProperty('Collation')
```

At the database level

```
SELECT
DatabasePropertyEx('BFEnterprise','Collation')

SELECT
DatabasePropertyEx('BESReporting','Collation')
```

At the column level in the BFEnterprise and BESReporting databases

```
SELECT C.name, O.name,
    C.collation_name
from BFEnterprise.sys.columns C,
    BFEnterprise.sys.all_objects O
where C.collation_name is not NULL
and C.object_id = O.object_id
and (O.schema_id = (SELECT SCHEMA_ID('dbo'))
or O.schema_id = (SELECT SCHEMA_ID('webui')))
```

```
SELECT C.name, O.name,
    C.collation_name
from BESReporting.sys.columns C,
    BESReporting.sys.all_objects O
where C.collation_name is not NULL
and C.object_id = O.object_id
and (O.schema_id = (SELECT SCHEMA_ID('dbo')))
```

Important: The database compatibility level for BFEnterprise and BESReporting must be at least 120.

Note: The database passwords must not contain the characters '{' or '}' , otherwise the installation might fail.

- Microsoft SQL Server 2019 or 2022 on Windows can be configured to leverage Microsoft Extended Protection for Authentication. On Microsoft SQL Server 2022 and 2025 the Strict Encryption is not supported.
- The BigFix server on Red Hat Enterprise Linux systems supports the following configurations with **MS SQL** as deployed database:
  - Starting from Patch 1, Microsoft SQL Server 2019 or 2022 (Enterprise and Standard are recommended) can be installed on BigFix Server Linux Red Hat 9.
  - Starting from Patch 2, BigFix Server supports Google Cloud SQL for Microsoft SQL Server 2022 database.
  - Starting from Patch 4, BigFix Server supports:
    - Azure SQL Managed Instance for Microsoft SQL Server 2022 database
    - Amazon RDS for Microsoft SQL Server 2022 database.
  - Starting from Patch 6, BigFix Server supports Azure SQL Managed Instance for Microsoft SQL Server 2025 and Google Cloud SQL for Microsoft SQL Server 2025 databases.

The following table shows details about the new databases supported on BigFix Server on Red Hat Enterprise Linux systems:

Database	Product minimum version	Notes
Microsoft SQL Server 2019 or 2022 (Enterprise and Standard are recommended)	11.0.1	◦ Only the SQL Server authentication is supported. ◦ DSA configurations are not supported. ◦ Microsoft SQL Server 2019 or 2022 on Linux can be configured to leverage Microsoft Encrypted Connection.
Google Cloud SQL for Microsoft SQL Server 2022	11.0.2	◦ Only the SQL Server authentication is supported. ◦ DSA configurations are not supported.
Google Cloud SQL for Microsoft SQL Server 2025	11.0.6	◦ Only the SQL Server authentication is supported. ◦ DSA configurations are not supported.

Database	Product minimum version	Notes
Amazon RDS for Microsoft SQL Server 2022	11.0.4	◦ Only the SQL Server authentication is supported. ◦ DSA configurations are not supported. ◦ **Important note:** On Amazon RDS for Microsoft SQL Server 2022, the SQL Server authentication user must be granted the server roles: processadmin, public, and setupadmin. The same user must also have db_owner privileges on the BFEnterprise and BESReporting databases.
Azure SQL Managed Instance for Microsoft SQL Server 2022	11.0.4	◦ Only the SQL Server authentication is supported. ◦ DSA configurations are not supported.

Database	Product minimum version	Notes
Azure SQL Managed Instance for Microsoft SQL Server 2025 database	11.0.6	◦ Only the SQL Server authentication is supported. ◦ DSA configurations are not supported.

- The BigFix server on Red Hat Enterprise Linux systems supports the following configurations with **DB2** as deployed database:
  - If the DB2 server is installed locally:
    - DB2 V11.5 GA / 11.5.4 / 11.5.5 / 11.5.6 / 11.5.7 / 11.5.8 / 11.5.9 Standard Edition 64-bit. These DB2 Versions are supported starting from BigFix Server Version 11.0.0.
    - DB2 V12.1.3 Standard Edition 64-bit. This DB2 Version is supported for BigFix Server Version 11.0.6.

      Note: This DB2 version (12.1.3) is not compatible with Red Hat Enterprise Linux 10.

    - DB2 V12.1.4 Standard Edition 64-bit. This DB2 Version is supported for BigFix Server Version 11.0.6.
  - If the DB2 server is installed remotely: IBM Data Server Client V11.5.

To check if you have a server or a client installed and to verify the DB2 edition, you can run the `db2licm -l` command. On the computer where the DB2 server is installed, you receive a detailed report, if only the client is installed you receive an empty report. To check which DB2 version is installed, run the `db2level` command.

Note: After installing a deployment, the DB2 instance name used to connect to the database can no longer be modified. The default name is `db2inst1`.

Note:

- The DB2 instance name with which BigFix connects to both BigFix Server and Web Reports databases must have the following privileges:
  - The SYSADM privilege during the phase of installation.
  - The DBADM privilege during the phase of upgrade.
  - The DATAACCESS privilege at run time.
- Ensure that you meet the following required database configurations needed to install/upgrade BigFix:
  - LOGFILSIZ must have a minimum value of 10240.
  - LOGPRIMARY must have a minimum value of 10.
  - LOGSECOND must have a minimum value of 100.
  - AUTO_REORG must be set to ON.
- Ensure that you you have stopped all BigFix services and you have closed all connections to the BFENT database. To verify if you have successfully done these operations, run the following command:

  ```
  db2 list applications for db BFENT
  ```

  It should return the following output:

  ```
  SQL1611W No data was returned by Database System Monitor.
  ```
- The DB2 instance names used to install the BES Root Server cannot contain the following special characters: blanks, tabs `\t`, returns `\n` and `; & | " ' < >`
- The database passwords cannot contain the following characters: blanks, tabs `\t`, returns `\n` and `; & | " ' < > %`
- To successfully install the BigFix server on Linux systems, ensure that you unset the DB2 registry variable DB2_COMPATIBILITY_VECTOR. This variable should be set to null.

Note: Do *not* change the name of BigFix databases; if you do, upgrades might fail.

For more information about the supported database versions, see BigFix Support Matrix.

Security requirements

The system authenticates all Fixlets and actions using secure public-key infrastructure (PKI) signatures. PKI uses public/private key pairs to ensure authenticity.

Before installing BigFix, you must run the Installer on Windows and the script `install.sh` on Linux to generate your own **private key** and then apply to HCL for a signed certificate containing your **public key**. Your private key (which only exists on your computer and is unknown to anyone else, including HCL) is encrypted by a password of your choosing, so if someone steals it, they still need to know your password to be able to use it. Nevertheless, guard it well. **Anyone who has the private key and password for your site, access to the server, and a database login will be able to apply any action to your Client computers.**

Treat your private key just like the physical key to your company front door. Do not leave it lying around on a shared disk. Instead, store it on a removable disk or a secured location and **do not lose it**. In the physical world, if you lose your master key you have to change all the locks in the building. Similarly, if you lose your digital key, you will need to do a migration to a new authorization key or a fresh installation of the entire system (including all the Clients). It is not unreasonable to store a backup copy of your site level key files in a secured safe deposit box.

During the installation process a server signing key is created and stored as a file on the server machine. Whenever operators issue an action, it is digitally signed by the server signing key, and the client will only trust actions that are signed by that key. Since clients will trust any action signed by the server signing key, it is important to protect the server signing key file. To protect the server signing key file, administrator access to the server machine must be restricted.

Fixlets are also digitally-signed. The Fixlet site author signs each message with a key that can be traced back to the BigFix root for authentication. This signature must match the Fixlet sites masthead, which is placed in the Client install folder when subscribing to the site. This procedure prevents spoofing and man-in-the-middle attacks, and guarantees that the Fixlets you receive are from the original certified author.

There are a few other security-related issues to address before installing BigFix in your organization:

- Make sure the server computer is running Windows Server 2008+ 64 bit with the latest Service Pack available from Microsoft.
- Make sure that the SQL Server is secured with the latest security-related patches.
- Make sure that TCP/IP and UDP on the specified port (default value is `52311` for all the components, included the console) is completely unblocked at all internal routers and internal firewalls.
- Verify that your external router forbids inbound and outbound traffic on the specified port (default value is `52311` for all the components) so that BigFix-related traffic will be unable to flow into or out of your network.

  You can administer roaming laptops by putting an authenticating relay in your DMZ. For additional details, see Authenticating relays and servers.
- Verify with your network administrator that you can allow the server to access the Internet via port **80**.The BES Root Server service on Windows and the `beserver` service on Linux access the Internet and by default they run as the SYSTEM account on Windows and as root on Linux.

  Note: In your environment, if you reach the Internet through a proxy configure the connection as described in Setting up a proxy connection *(on page 545)*. If you have firewall restrictions, see Configuring a Local Firewall *(on page 111)*.

  To maintain a physical disconnect from the Internet, see Downloading files in air-gapped environments.
- Secure the server computers and the SQL database using company or industry-wide standards. Contact your network administrator or database administrator for more information.

Note: Certain rare lock-down procedures might cause the servers to function incorrectly. Contact your HCL software support if you have any specific questions about lock-down procedures.

Secure the client computers by using company- or industry-wide standards; applying the Principle of Least Privilege (PoLP) is recommended.

For best results on Windows systems, do the following:

- Keep the UAC feature enabled always.
- Avoid seting up user accounts with local administrative privileges.
- Ensure restircted access to the system directory paths (for example, Windows, System32, Program Files (x86), Program Files). Prevent local users from accessing these locations.

Network configuration requirements

The following network configuration is recommended for security and performance reasons:

- All internal network communication is on one specified port (`52311` is the default port for all the components, including the console) to allow for simplicity and flexibility of deployment. TCP/IP and UDP on this port must be completely unblocked at all internal routers and internal firewalls (you can optionally disable UDP, but that might negatively affect performance).
- The BigFix server should connect to the network at 100 mbps or higher.
- Consoles should have high speed connections to the BigFix server (100 mbps or higher)
- The BigFix client must be installed on the BigFix server machine.

These networking recommendations are typically easy to satisfy for most organizations maintaining a moderate security posture. For information about larger installations, see Deployment Scenarios (on page 61).

The BigFix Server requirements and performance can also be affected by other factors in addition to the number of clients. These factors include:

The number of console operators

Multiple console operators can connect to the servers at the same time to manage subsets of the networked computers. Some deployments can have hundreds of operators. If you plan to have more than 30 operators, you might want to have a more powerful Server to support the additional load.

Relays

Use to lighten the load on the servers by accepting connections from clients and then forwarding the data to a server. In most deployments, very few clients report directly to the main Server.



Note: To improve performance, you can connect from 500 to 1000 clients to each relay and use a parent child relay configuration.

The number and type of Retrieved Properties and Analyses

Custom-Retrieved properties and analyses can provide extremely useful data, but if custom properties are poorly implemented or overused, they can also create undue load on the system by requiring too much bandwidth or too many client resources. For example, it would be unwise to create a custom-retrieved property that returned the names of every file on every computer, due to the load on the client computers and the network.

Assumptions

The process of getting the BigFix up and running varies, depending on your network environment and your security policies.

This guide focuses on a standard deployment, which applies to workgroups and to enterprises within a single administrative domain. For the sake of readability and generality, this guide assumes these restrictions:

- BigFix servers can make connections to the Internet on port 80. The BigFix server can be set up to use a proxy, which is a common configuration. For more information see Setting up a proxy connection *(on page 545)*. Alternatively, an air-gap can be

used to physically separate the BigFix server from the Internet Fixlet server. For more information, see Downloading files in air-gapped environments.

- Each BigFix server must have access to the SQL server, located locally on the server machine or remotely on a separate SQL Server.
- Each console operator can make an HTTP connection to the BigFix server.
- Each BigFix client computer in the network must be able to make an HTTP connection to a server or relay on the specified port (the default port is 52311, but any available port can be used).
- Each console in the network must be able to make an HTTPS connection to a server on the same port as the clients (default value is `52311`).
- The BigFix components (server, relays, WebReports) must be installed on systems whose hostname contains only ASCII characters.
- The installation paths of BigFix components must contain only ASCII characters.

Some enterprises might not meet one or more of these conditions, but can still deploy BigFix in their environments.

For more information, see Sample deployment scenarios (on page 61).

If your network configuration does not match any of the described scenarios, contact a support technician for more options.

The initial deployment of a minimal BigFix system (server, console, and a few clients) takes about one hour to complete.

When you are ready to install the full system, pay extra attention to the sections in this document on client and relay deployment, to ensure an efficient rollout.

Several steps in the BigFix installation depend on the completion of prior steps. For this reason, it is recommended that you follow this guide in the order presented.

Chapter 5. Types of installation

Before you install the product, decide if you want to do an evaluation or production installation.

If you choose the evaluation installation, you do not need to buy a license. You can install a trial BigFix Server on your own and use it for a period of 30 days.

If you choose the production installation, you must purchase a license. When you receive your BigFix license authorization file, you are ready to create a personalized **action site masthead** that, in turn, allows you to install and use BigFix.

The masthead includes URLs for the Server CGI programs and other site information in a signed MIME file. The masthead is central to accessing and authenticating your action site and is linked to the hostname or IP address of the server machine.

Evaluation installation

When you run the evaluation installation, you do not need to buy any license files from HCL. The evaluation installation configures BigFix as a Try and Buy product, so you can try it first and buy a license later.

The trial product has the following limitations:

- A license expiration of 30 days
- A license allocation of 1000 clients
- A predefined set of allowed sites. You can enable the desired sites after completing the evaluation installation.

When you install BigFix in evaluation mode, the installation wizard collects the information required to request an evaluation BigFix license to HCL, and creates an evaluation masthead that entitles your organization to use BigFix. You must specify the correct data which is needed if you want to convert your server evaluation license into a production license.

You must generate the license on a machine with network connectivity. You can then use this evaluation license the same way you would use a production license.

If you are generating an evaluation license for a server that is located in an airgapped environment, stop the current installation and copy the generated `license.crt` and `license.pvk` files from the local machine to the isolated server. Restart the installation on that server by choosing type "Production" and selecting the files you copied previously.

After an Evaluation installation, a user named `EvaluationUser` is created to log on both the BigFix console and BigFix Web Reports.

If you need an evaluation time longer than 30 days, contact the sales support to arrange a limited production license.

It is possible to convert the evaluation installation to a production installation. For additional information see Moving from evaluation installation to production installation (on page 97).

Note: The evaluation installation does not support the enhanced security option. For more information about this feature, see Security Configuration Scenarios.

Note: A supported version of Microsoft SQL Server must be installed on the system prior to launching the evaluation installation.

Production installation

If you purchased a license from HCL, you can perform a production installation.

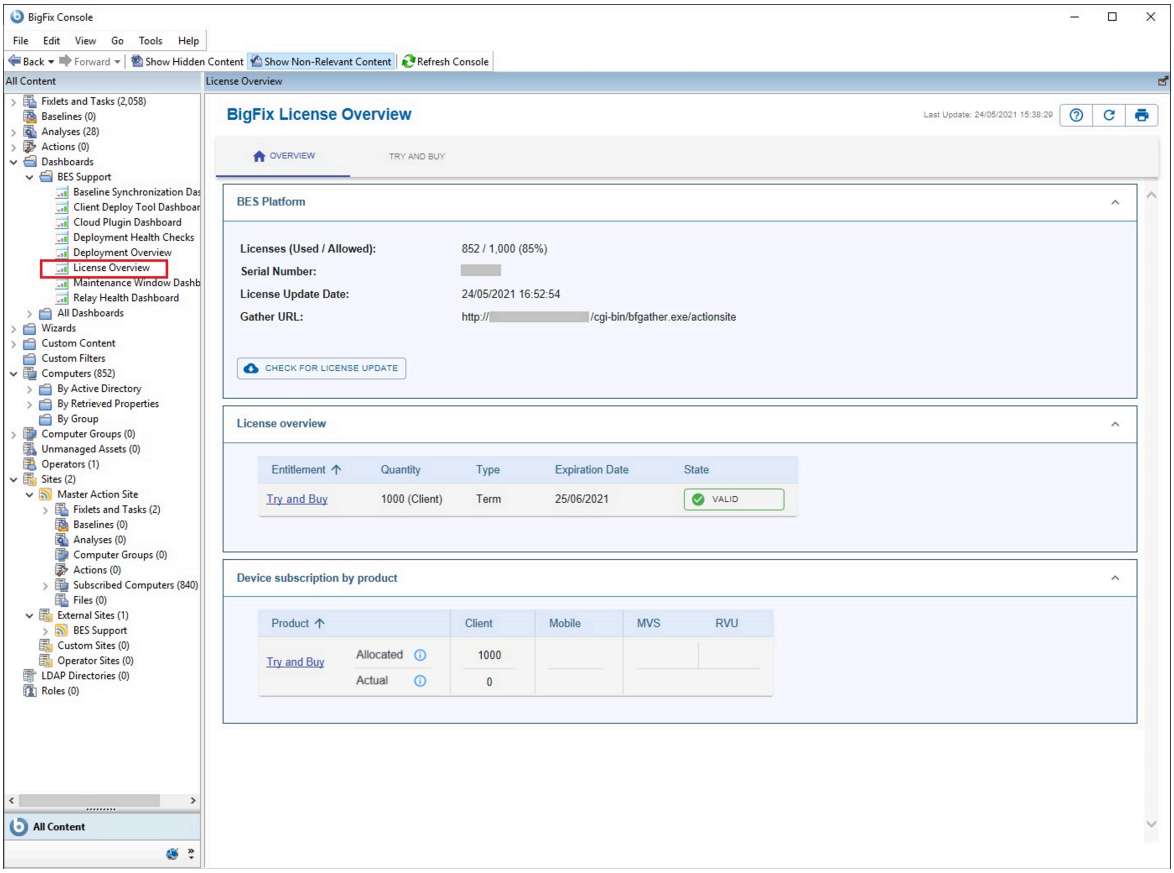
After the production installation, a user (default name is `BFAdmin`) is created to logon to the BigFix Console and BigFix Web Reports.

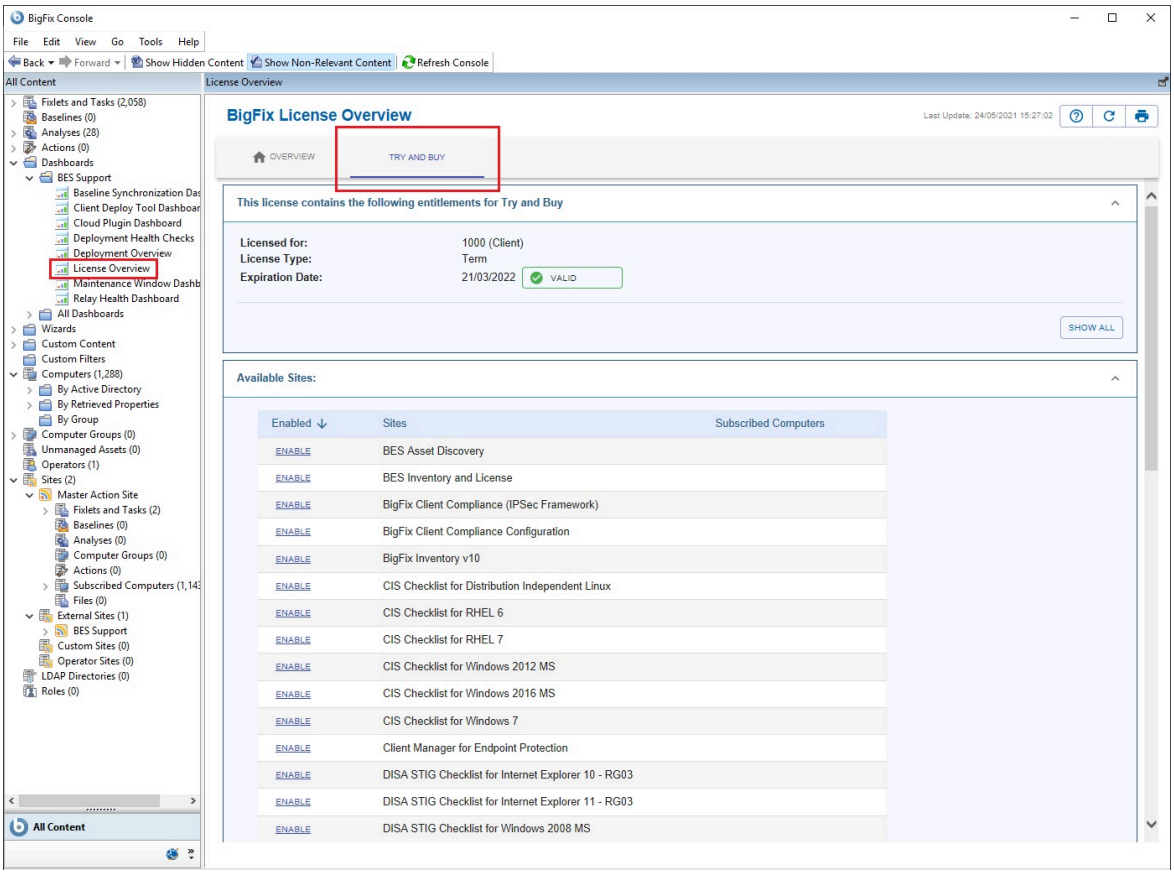
Moving from evaluation installation to production installation

You can install and configure a BigFix environment in evaluation mode and later decide to upgrade the environment to production by converting the evaluation license to production without reinstalling the entire environment.

After you install in evaluation mode you can decide to:

- Let the evaluation license expire. When the evaluation license expires the environment will stop working.
- Upgrade the evaluation license to production without reinstalling the environment from scratch. You must ask for a license update from evaluation to production. From the BigFix License Overview dashboard, click **Check for license update**:





Run the BESAdmin to propagate the change. You can also wait for the BigFix server periodically checks of license changes. When license updates are available, it processes and propagates them to the environment.

When the masthead is propagated, you can enable all the sites that are included in the production license you have bought. All the sites that were included in the evaluation license but are not included in the production license you have acquired are disabled.

Chapter 6. Managing licenses

You must obtain a license key before you can install and use BigFix.

Your license is composed of two files:

- Your public key file: `license.crt`
- Your private key file: `license.pvk` protected by a password

The following table lists the tasks that are required to purchase, generate, and manage your license keys.

Task	Description
Check the product license require-ments	It is important to understand the license requirements of the system you want to protect. A license lets you install the BigFix client on a specified number of computers.
Purchase a license	You must purchase a license in the following situations: • You want to purchase BigFix. • Your trialware license expired. • Your paid license expired. • Your license is over-deployed and an updated license.crt is required for the increased license count purchase. • Your upgrade license expired. Within a few hours of your purchase you receive two emails. One email is sent from HCL as confirmation of your purchase. Another email contains instructions about how to access My HCL Software. These emails are sent to the technical contact associated with the HCL Customer Number for the account.

Get the license authorization file	To get your product license you must have an authorization file from the My HCL Software site. See Creating the License Authorization File *(on page 103)*.
Generate your license files during installation:	During the installation of the Server, after you specify the license authorization file, you generate the `license.pvk` file, which is your private key file. You also request and get the `license.crt` file, which is your public key file. These two files together complete your license. See Requesting the license files (on page 131) on Windows.

- Create the private key file
- Request and get the license certificate
- Generate the

masthead file	
Back up your license files	Store your `license.crt` (public key) file with your existing `license.pvk` (private key) file. Keep these two keys together and create a backup copy in a secure location. Create also a backup copy of the site admin password that you used to encrypt the private key file, and store it in a secure location. Only in this way are you in complete control of your license keys. Backing up your license files preserves the license files in case the database or the computer hard disk is damaged. In particular the `license.pvk` file is the part of your key files that needs to stay private. The `license.crt` file is your public key file and must be combined with your private key file to complete your license. You can open the license files in a text editor to review their contents.
Check license status and distribute the new license and masthead files	You can see the notifications about expired license and other license issues for the license that you imported into the console. See Distributing the Updated License and Masthead (on page 105).

This is a summary of the steps to perform to get your license key files:

1. Purchase a license.
2. Get an authorization file from the My HCL Software site.
3. Start the BigFix installation and enter the authorization file when requested to get the `license.crt` file. At the end of the process both the public key and private key license files are generated together with the masthead file. This file contains configuration,

license, and security information, including URLs that point to where trusted Fixlet content is available. It is used for installing DSA servers and is distributed to all the clients using that server.

For detailed steps, you can read the Knowledge Base article Managing BigFix licenses in the My HCL Software Portal.

Creating the License Authorization File

To create your license authorization file (`.BESLicenseAuthorization`), containing deployment and licensing information and used during the installation to create your license files, access the **My HCL Software** portal.

This portal hosts a software download, license key delivery and management service that allows you to obtain and manage the license keys you need to use the product.

To create the authorization file perform the following steps:

1. Access My HCL Software
2. Enter your email address and the password you received together with the instructions about how to access My HCL Software
3. For each product in the list specify the allocated client quantity. If you leave 0, you cannot install the related product.

Licensing Assistance

For specific problems with your license such as license expiration date, entitlement counts, or lost authorization files, open a Licensing case in the HCL Customer Support Portal at https://support.hcl-software.com/csm

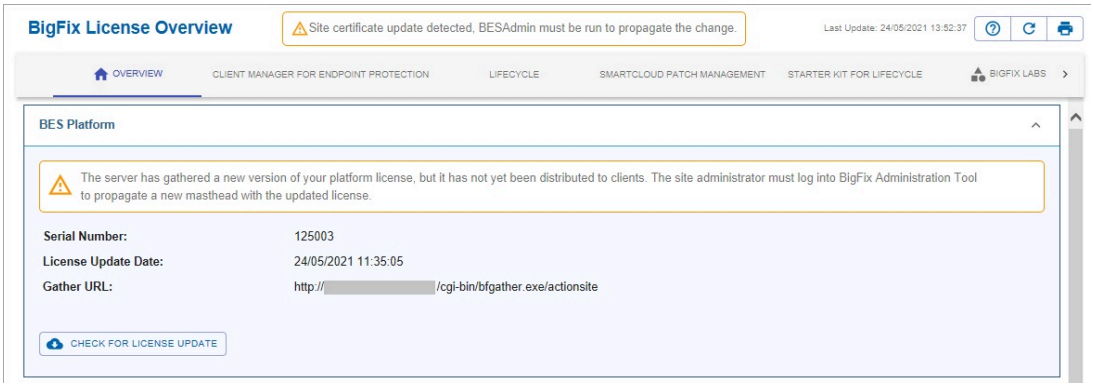
Extending the license entitlements

How to purchase extra license entitlements.

To use BigFix Fixlet sites, you must first purchase extra entitlements. The entitlements can be added to your current license. After your license is updated, you can enable the Fixlet sites in the BigFix Platform.

Procedure

1. Contact the Sales team to purchase entitlements for the BigFix Fixlet sites.
2. Check the serial number of your current BigFix license:
   a. Log in to the BigFix console.
   b. In the bottom-left corner, click **BigFix Management**.
   c. In the navigation tree, click **License Overview**.
   d. Your serial number is displayed in the BigFix Platform window.



3. In the BES Platform window that you copied the serial number from, click **Check for license update**. If there is a change to your license, the following message is displayed: `Site certificate update detected, BESAdmin must be run to propagate the change.`
4. Go to the BigFix server and run the Administration Tool to update the license:

   On Linux operating systems

   a. Go to `/opt/BESServer/bin`.
   b. Run the following command:

   ```
   ./BESAdmin.sh -syncmastheadandlicense -sitePvkLocation
   =path_to_license.pvk
   ```

   On Windows operating systems

a. Go to `C:\Program Files (x86)\BigFix Enterprise\BES Server`.
b. Run `BESAdmin.exe`.
c. When prompted, provide the path to the site signing key (`license.pvk`), and enter the password.
d. In the Masthead Management tab, click **OK**.

Results

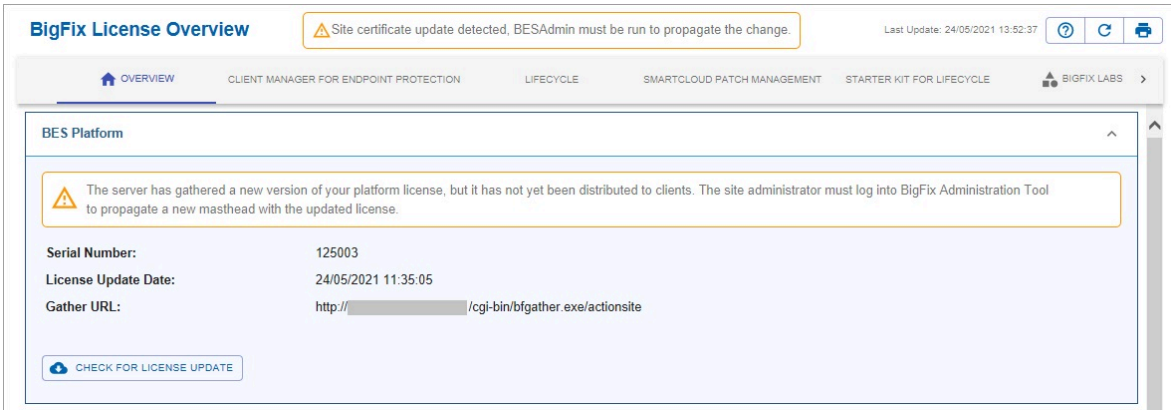
You updated your license and propagated the change to your endpoints. It might take several minutes until the BigFix console displays the updated status. If the status does not change, restart the console.

Distributing the Updated License and Masthead

If you are connected to Internet, the message of a new license ready to be distributed to the clients together with the masthead is displayed in the License Overview dashboard after an automatic periodic gather or a manual check.

To force your server to check immediately run the following steps:

1. Open the BigFix console.
2. go to the **BigFix Management** domain.
3. click the **License Overview** node
4. Click **Check for license update**. You might receive a notification that BigFix deployment has gathered an update to your license (a new `license.crt` file).



Note: This message might appear either because HCL needs to update the license or because you requested an update of your license. If you requested an update of your license, you receive a new `license.crt` file, that you must save on your server computer.

To distribute the updated license, resign the masthead and the objects in the database with both SHA-256 and SHA-384 signatures, run the Administration Tool (`./BESAdmin.sh` on Linux as super user).

If you are in an air-gapped environment the update of the license is not processed automatically. You can retrieve the license from the HCL site by using the AirgapTool utility. After importing it, you are notified from the License dashboard that a license update is ready to be distributed. You must run the Administration tool (`./BESAdmin.sh` on Linux) to distribute the updated license, and to resign the masthead and the database objects.

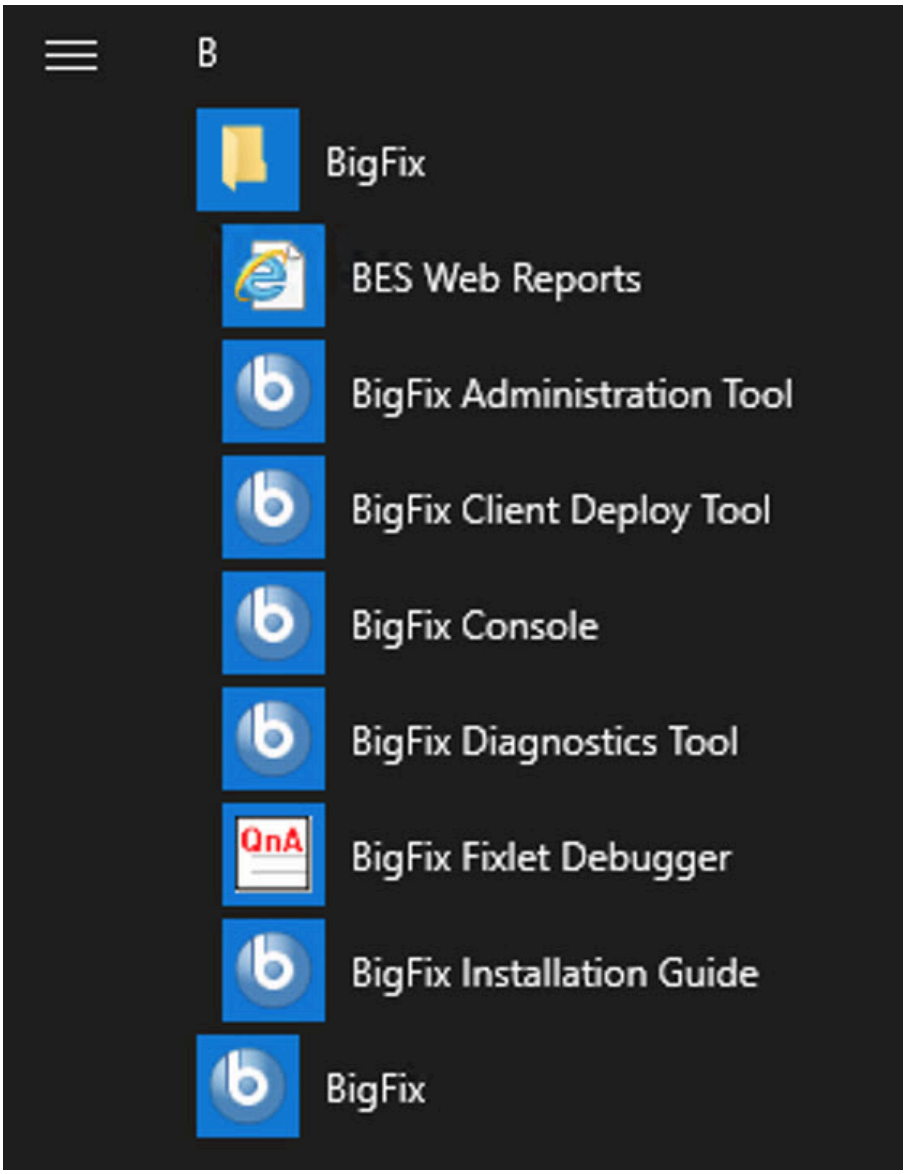
For more information about how to distribute the masthead on the clients see Distributing the masthead from the Windows server to clients (on page 106) and Distributing the masthead from the Linux server to the clients (on page 110).

Distributing the masthead from the Windows server to clients

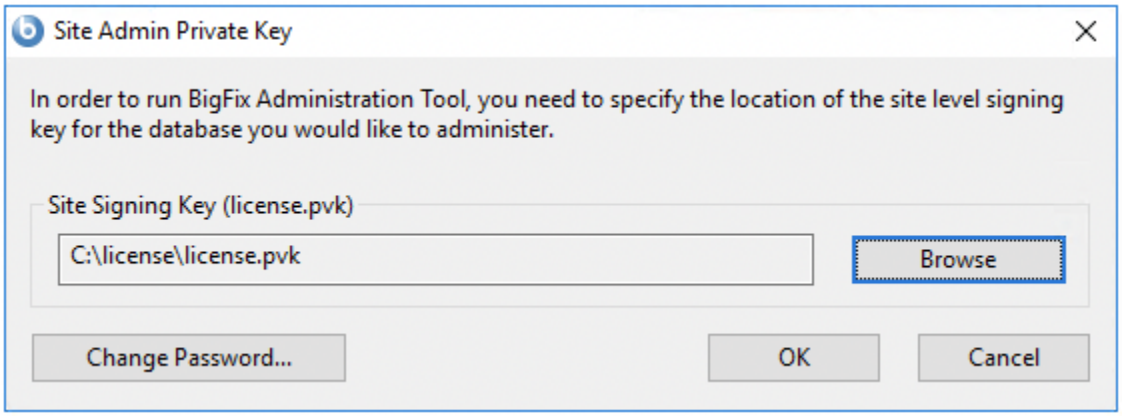
From an BigFix Windows server, you can distribute a new masthead file with an updated license certificate, that extends your license, seat count, or entitlements to the clients.

Perform the following steps:

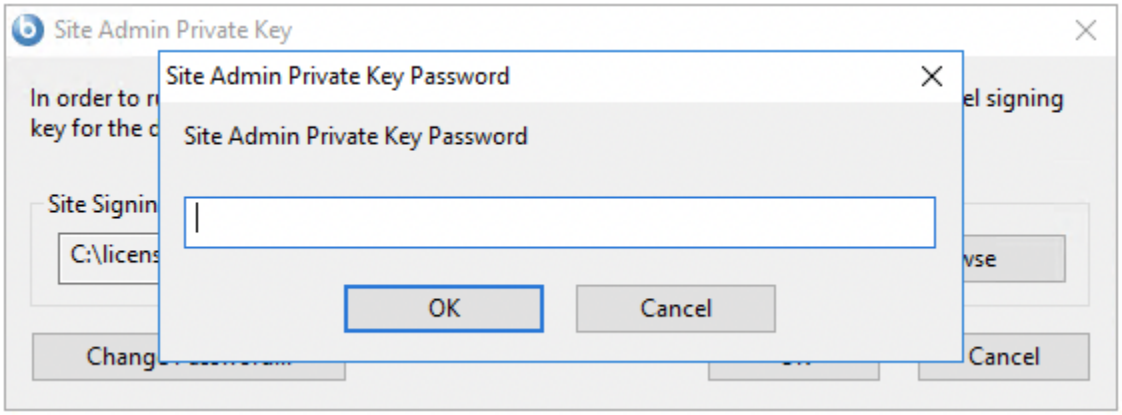
1. Open the Administration Tool by selecting **Start > BigFix > BigFix Administration Tool**. After you log in, the installation Admin account distributes the masthead to the clients.



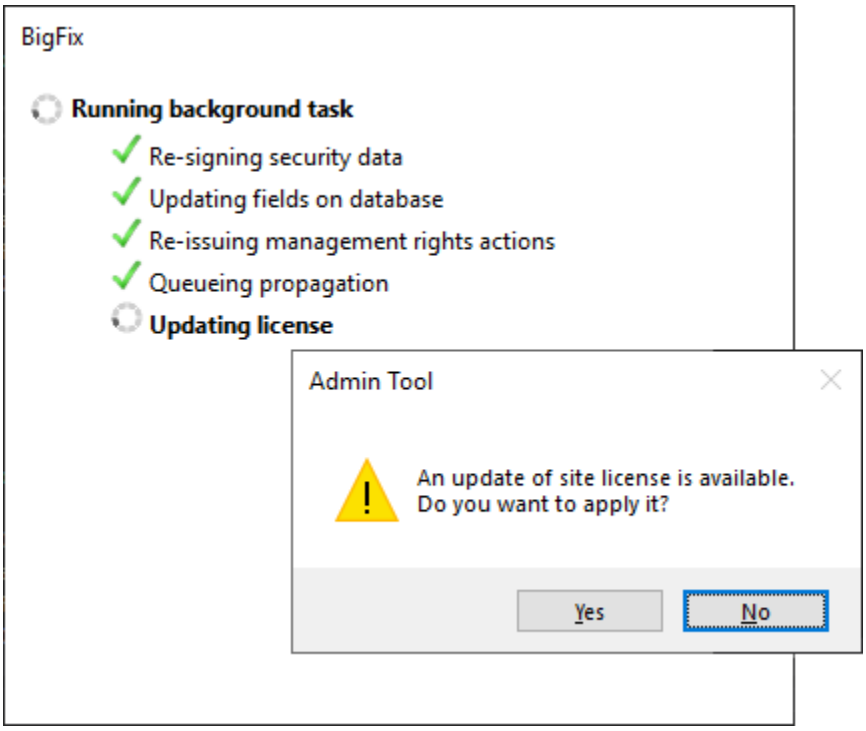
2. Choose your `license.pvk` file.



3. Enter your master (site level) password



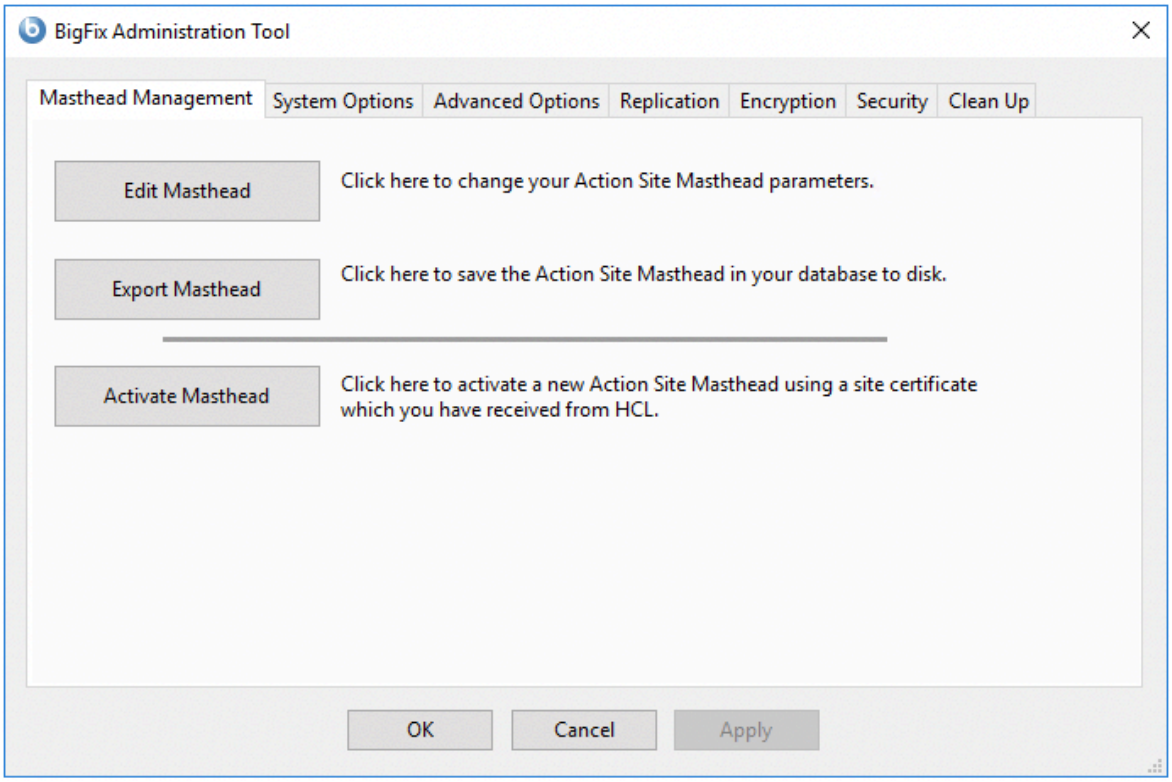
4. Confirm or not if you want to apply the license update



From a BigFix Windows server, you can distribute the updated license, resign the masthead and the database objects to the clients, also by running the following command:

```
.\BESAdmin.exe /syncmastheadandlicense /
sitePvkLocation=<path+license.pvk>
/sitePvkPassword=<password>
```

5. In **Masthead Management**, click **OK**.



As soon as the clients receive the new masthead, they receive the updated license information.

Distributing the masthead from the Linux server to the clients

From a BigFix Linux server, you can distribute the updated license, resign the masthead and the database objects to the clients, by running the following command as super user:

```
./BESAdmin.sh -syncmastheadandlicense -sitePvkLocation=<path+license.pvk>
-sitePvkPassword=<password>
```

Chapter 7. Before installing

Before running the installation make sure that you read the following topics and run the requested activities if needed.

Configuring a Local Firewall

If you have defined an active firewall on the computer where you are installing the BigFix server, you can decide to configure this firewall during the BigFix server installation in one of the following ways:

- During an interactive installation, the installation programs detects if a local firewall is active and you can specify if you want to configure it for the BigFix server.
- During a silent installation, you can set `CONF_FIREWALL=YES` in the response file to require the firewall configuration. For more information, see Silent installation *(on page 246)*.

When you specify to configure the firewall, open the following two ports:

- Port `52311` for UDP and TCP/IP
- Port `8083` for Web Reports and TCP/IP

Modifying port numbers

By default, the server uses port **52311** to communicate with the clients, but you can choose any port number (although you should avoid the reserved ports between 1 to 1024 because of potential conflicts and difficulty managing network traffic).

Your choice of the server port number is factored into the generation of the masthead, which specifies URLs for the action, registration, reporting, and mirror servers. As a consequence, you must finalize your port number **before installation**.

Consoles use port **52311** to connect to the server.

Understanding the server components

The BigFix server responds to messages and requests from the relay, client, and console computers using a variety of components.

To better understand what the server does, read the descriptions of some of the components.

Client Registration Component

When the client is installed on a new computer, it registers itself with the client registration component of the server and the client is given a unique ID. If the computer IP address changes, the client automatically registers the new IP address with the client registration component.

Post Results Server Component

When a client detects that a Fixlet has become relevant, it reports to the Post Results server component using an HTTP POST operation. It identifies the relevant Fixlet together with the registered ID of the client computer. This information is passed on to the BigFix database through the FillDB service and then becomes viewable in the console. Other state changes are also periodically reported by the clients to the server directly or through relays.

Gather Server Component

This component watches for changes in Fixlet content for all the Fixlet sites to which you are subscribed. It downloads these changes to the server and makes them available to the GatherDB component.

FillDB Component

This component posts client results into the database.

GatherDB Component

This component gathers and stores Fixlet downloads from the Internet into the database.

Download Mirror Server Component

The Download Mirror Server component hosts Fixlet site data for the relays and clients. This component functions as a simplified download server for BigFix traffic.

Chapter 8. Installing on Windows systems

Now that you understand the terms and the administrative roles, you are ready to get authorized and install the programs.

Because BigFix is powerful, you might want to limit access to trusted, authorized personnel only. The product depends on a central repository of Fixlet actions called the **Action site**, which uses public/private key encryption to protect against spoofing and other unauthorized usage. To get started, you need authorization from HCL by getting a **License Authorization** file, which will have a name like `CompanyName.BESLicenseAuthorization`.



Note: The Administrator privileges are required to perform the installation of the server components.

The Installer program collects further information about your deployment and then creates a file called the **action site masthead**. This file establishes a chain of authority from the BigFix root all the way down to the Console operators in your organization. The masthead combines configuration information (IP addresses, ports, and so on) and license information (how many Clients are authorized and for how long) together with a public key that is used to verify the digital signatures. To create and maintain the digital signature keys and masthead, you use the **BigFix Installer**, which you can download from HCL.

Installing and configuring SQL Server 2025 DB

Starting from BigFix Version 11 **Patch 6**, the Windows Server installation introduces the support of **Windows Server 2019 (or later)** x86 64-bit with **Microsoft SQL Server 2025** database.

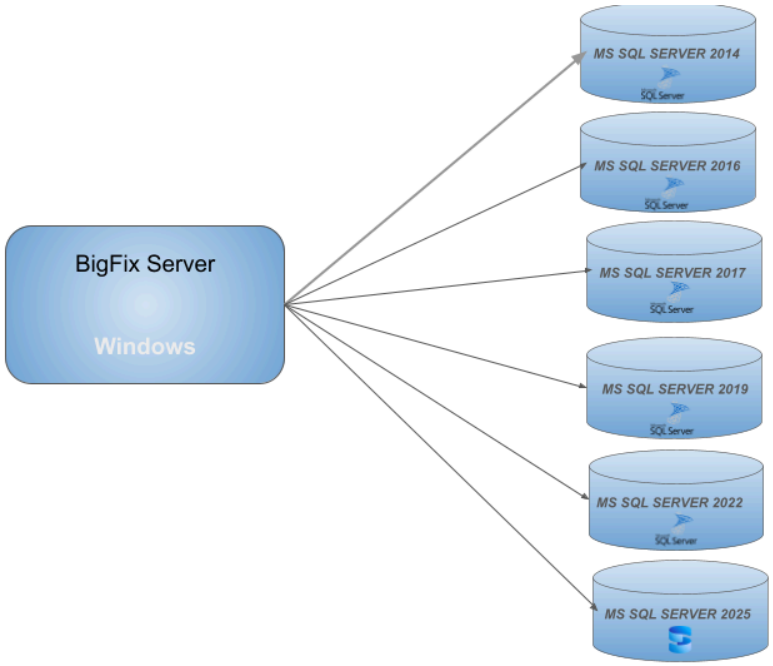
BigFix Platform Version 11 GA supported the following configurations:

- Microsoft SQL Server 2014 installed on Windows.
- Microsoft SQL Server 2016 installed on Windows.
- Microsoft SQL Server 2017 installed on Windows.
- Microsoft SQL Server 2019 installed on Windows.
- Microsoft SQL Server 2022 installed on Windows.

Note that the Enterprise and Standard editions are recommended.

Starting from BigFix Version 11 Patch 6, the Windows Server installation introduces the support of Windows Server x86 64-bit with Microsoft SQL Server 2025 database.

Note that the Enterprise and Standard editions are recommended.



On Microsoft SQL Server 2022 and 2025 the Strict Encryption is not supported.

Installing and configuring DB on Cloud

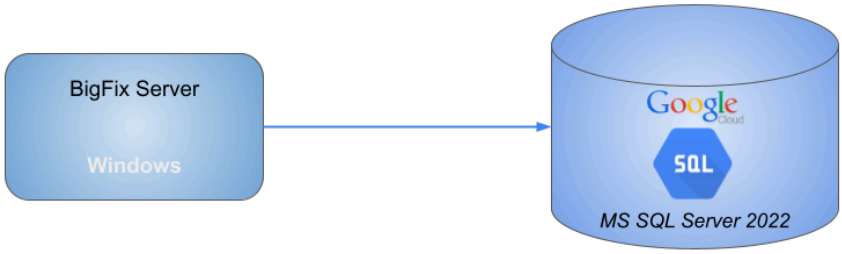
Starting from BigFix Version 11 Patch 2, the Windows Server installation introduces the support of Windows (2019 or later) with Google Cloud SQL for Microsoft SQL Server 2022 database.

Starting from BigFix Version 11 Patch 4, the Windows Server installation introduces the support of Windows (2019 or later) with:

- Azure SQL Managed Instance for Microsoft SQL Server 2022 database
- Amazon RDS for Microsoft SQL Server 2022 database.

Starting from BigFix Version 11 Patch 6, the Windows Server installation introduces the support of Windows (2019 or later) with Azure SQL Managed Instance for Microsoft SQL Server 2025 and Google Cloud SQL for Microsoft SQL Server 2025 databases.

BigFix Platform Version 11 Patch 2 supports the configuration shown below:

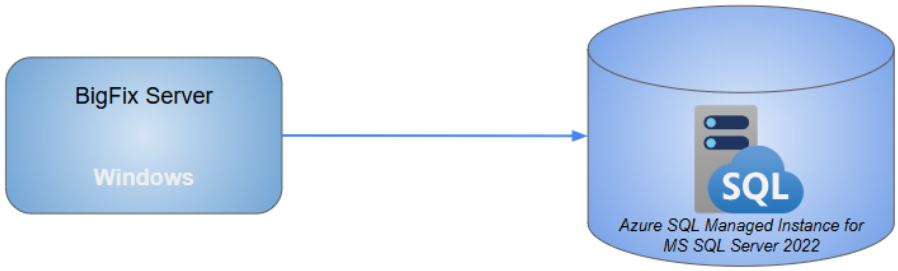


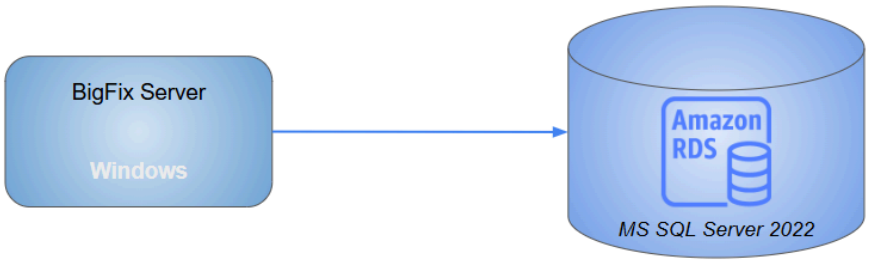
Note that the Enterprise and Standard editions are recommended.

Note: Only the SQL Authentication is supported on the BigFix Windows Server with Google Cloud SQL for Microsoft SQL Server 2022 database.

Note: DSA configurations are not supported with Google Cloud SQL for Microsoft SQL Server 2022 database.

BigFix Platform Version 11 Patch 4 supports the configurations shown below:





Note that the Enterprise and Standard editions are recommended.

Note: Only the SQL Authentication is supported on the BigFix Windows Server with:

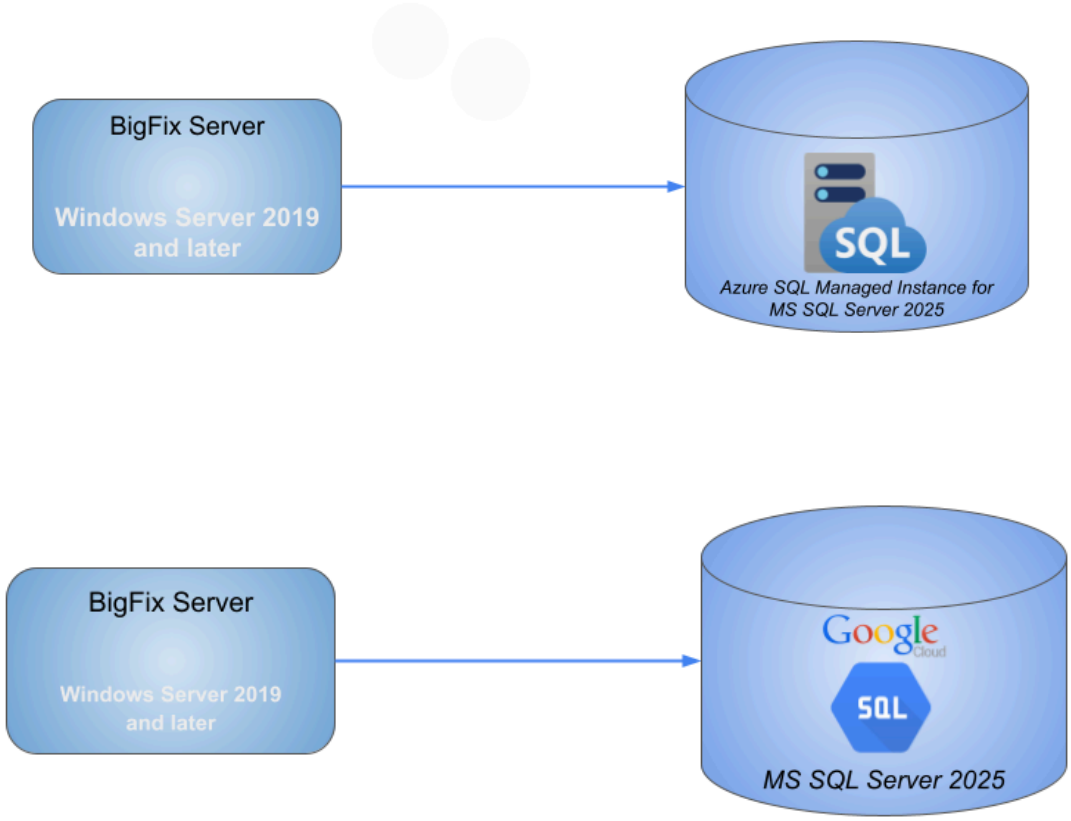
- Azure SQL Managed Instance for Microsoft SQL Server 2022 database
- Amazon RDS for Microsoft SQL Server 2022 database.

Note: DSA configurations are not supported with:

- Azure SQL Managed Instance for Microsoft SQL Server 2022 database
- Amazon RDS for Microsoft SQL Server 2022 database.

Note: When installing Amazon RDS for Microsoft SQL Server 2022 database, the installer is not able to automatically configure MaxDoP and CTFP parameter values for best performance. The installer, if required, suggests the recommended parameter values to be configured in order to optimize the database performance. These values can be set by the Operator on the Amazon RDS Console.

BigFix Platform Version 11 Patch 6 supports the configurations shown below:



Note that the Enterprise and Standard editions are recommended.

Note: Only the SQL Authentication is supported on the BigFix Windows Server with Azure SQL Managed Instance for Microsoft SQL Server 2025 and Google Cloud SQL for Microsoft SQL Server 2025 databases.

Note: DSA configurations are not supported with Azure SQL Managed Instance for Microsoft SQL Server 2025 and Google Cloud SQL for Microsoft SQL Server 2025 databases.

Downloading BigFix

Download BigFix from the HCL License & Delivery Portal (Flexnet).

You can download BigFix also from the support site at http://support.bigfix.com/bes/install/downloadbes.html.

To install the server component, download the following e-images from HCL License & Delivery Portal:

Table 1. Software required for installing BigFix Server Version 11

Software Name	Image
BigFix Platform Install V11.0 for Multiplatform Multilingual	`HCL_BigFix_v11.0.x_Win_Lnx_Install.zip`

To extract the BigFix Windows server installation files, perform the following steps:

1. Copy the BigFix Server zip file `HCL_BigFix_v11.0.x_Win_Lnx_Install.zip` to your Windows Server.
2. Expand the zip file using the following command:

   ```
   unzip "HCL_BigFix_v11.0.x_Win_Lnx_Install.zip"
   ```

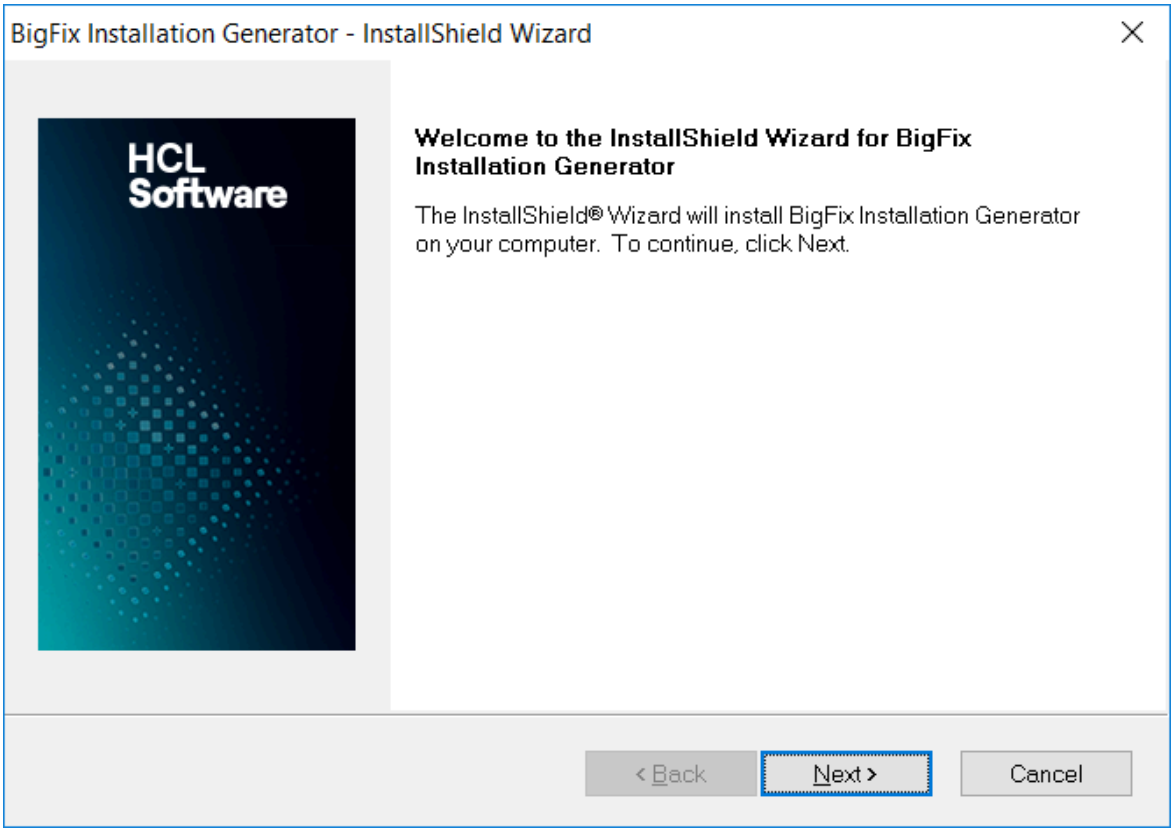
   You can find the `setup.exe` file to install the Windows Server.

Performing an evaluation installation

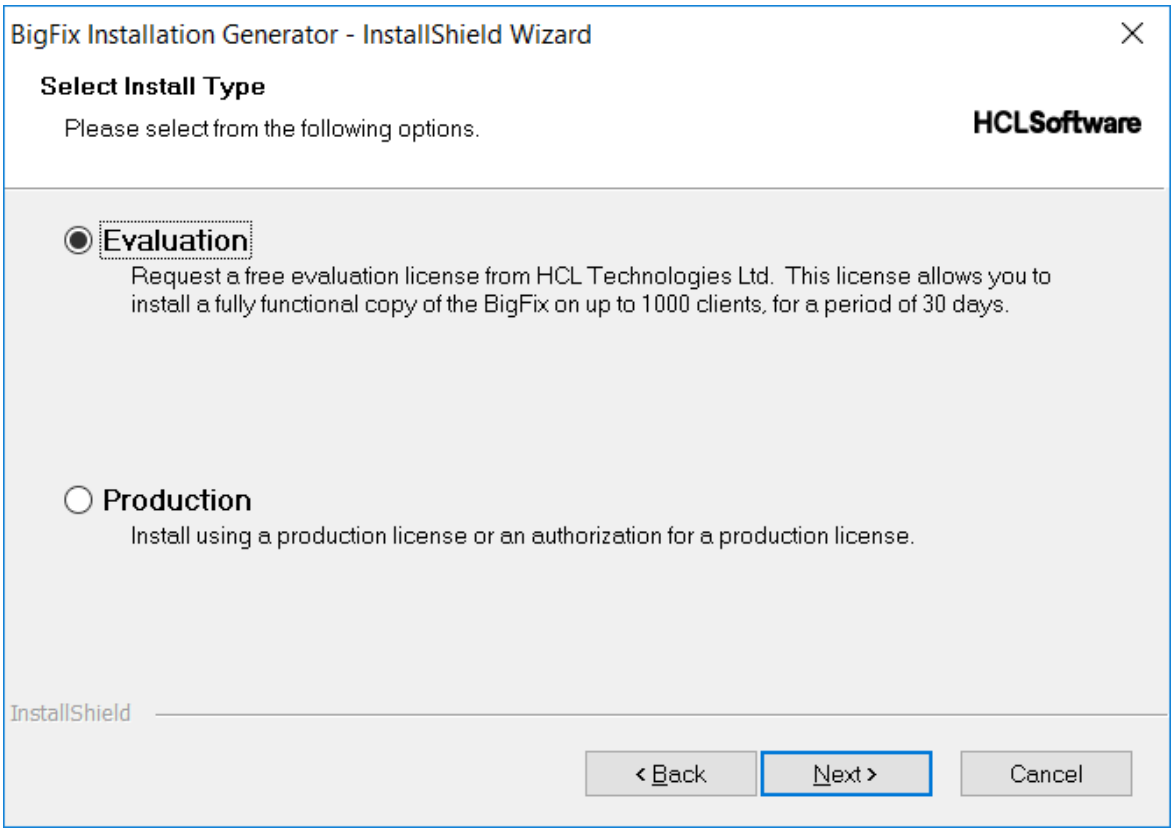
When performing a fresh installation of BigFix Server Version 11, you can either perform an evaluation installation or a production installation.

To install a BigFix server with an evaluation license, perform the following steps:

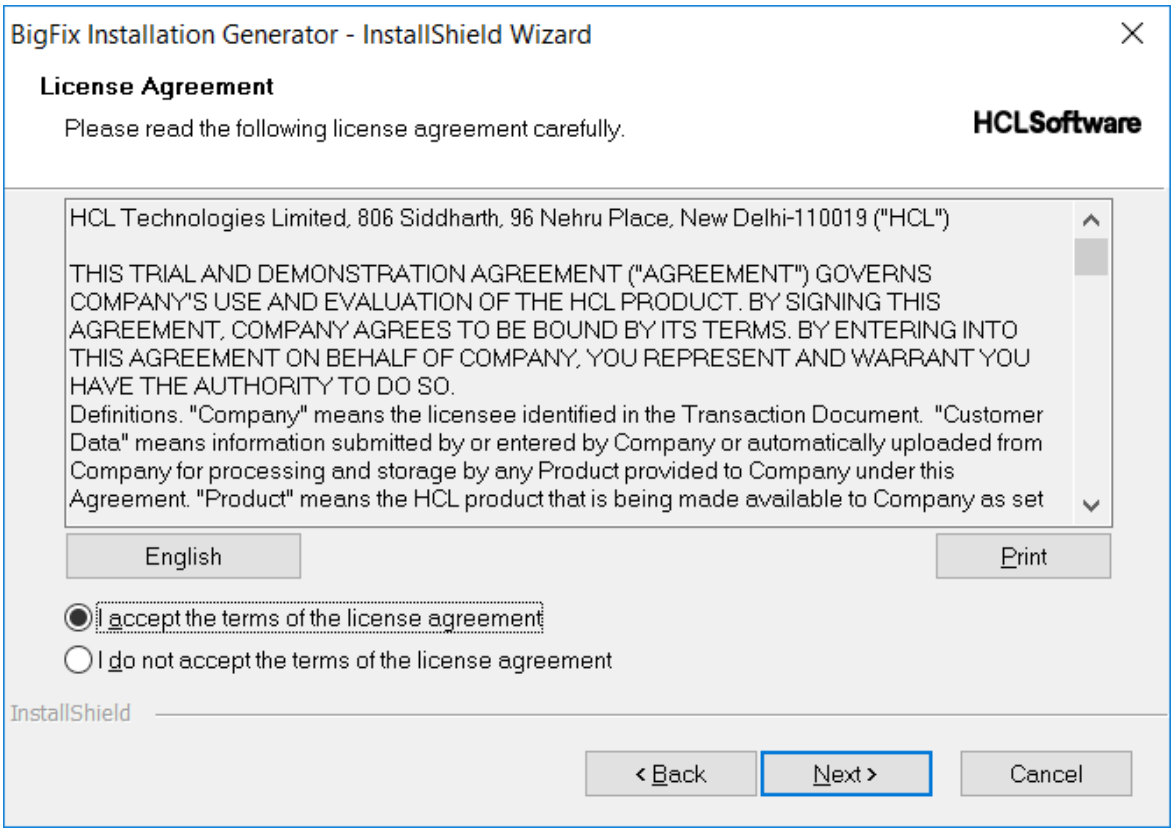
1. On the computer where you want to install the BigFix server, run the BigFix Installation Generator.



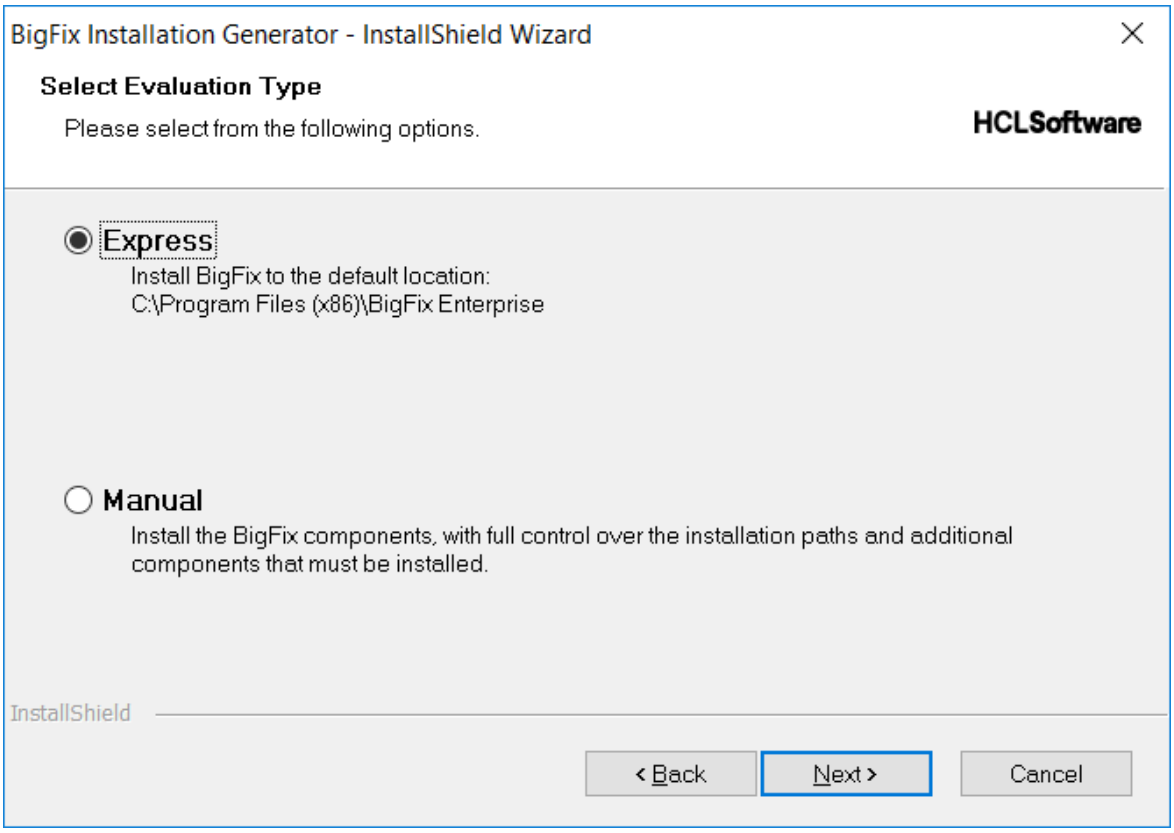
2. Select **Evaluation**.



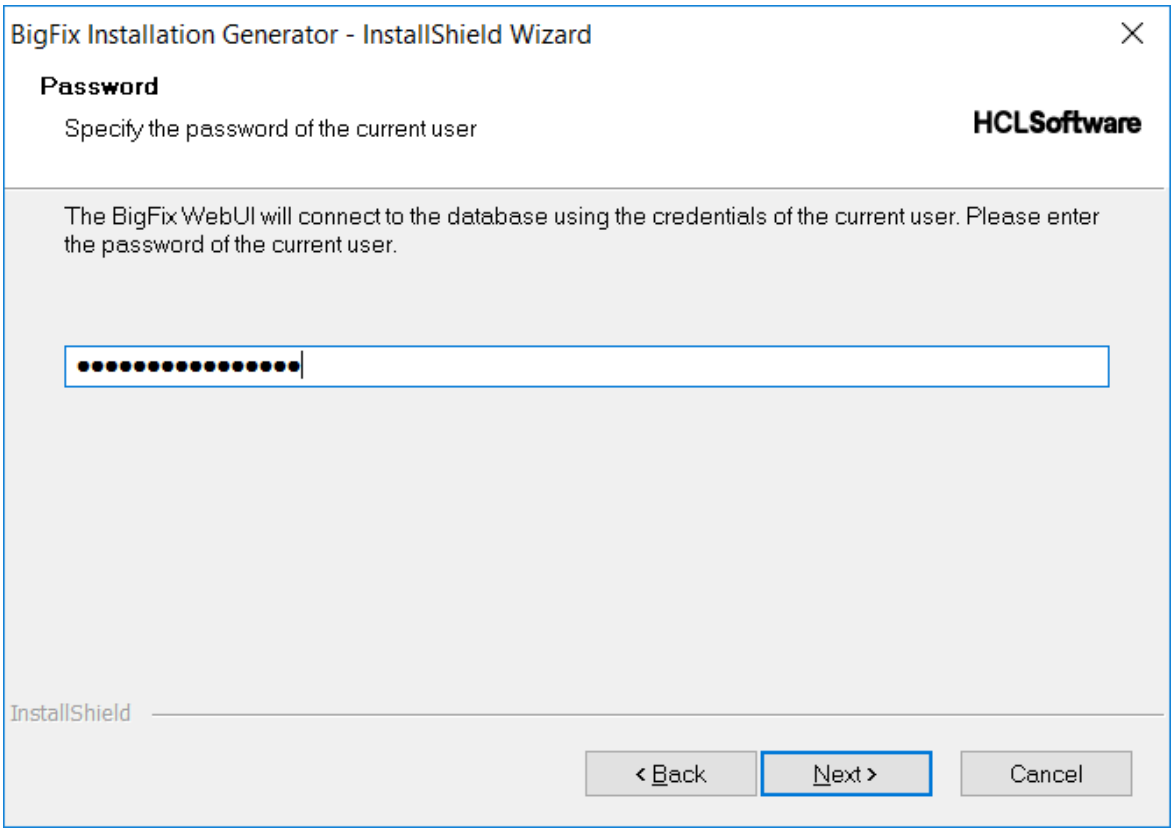
3. Accept the terms in the license agreement.



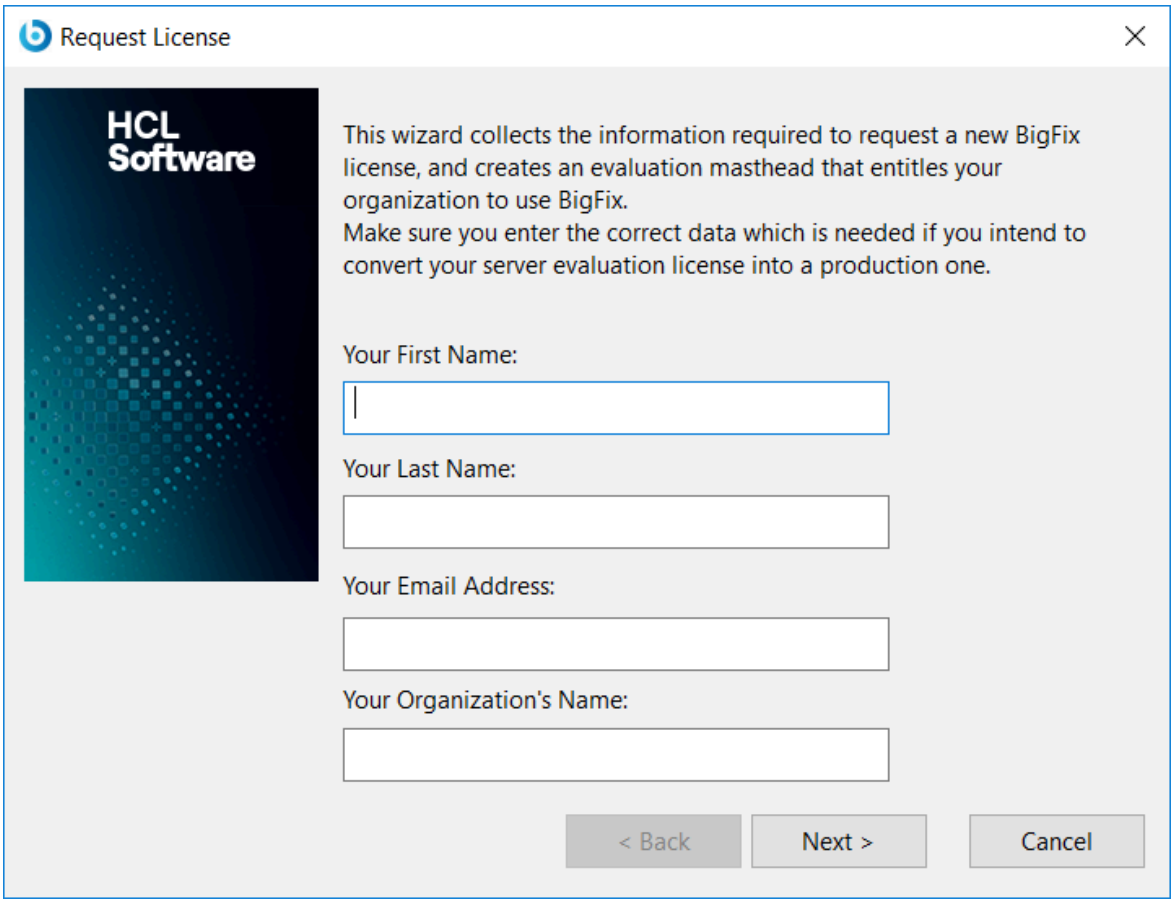
4. Select **Express** if you want to install BigFix to the default location.



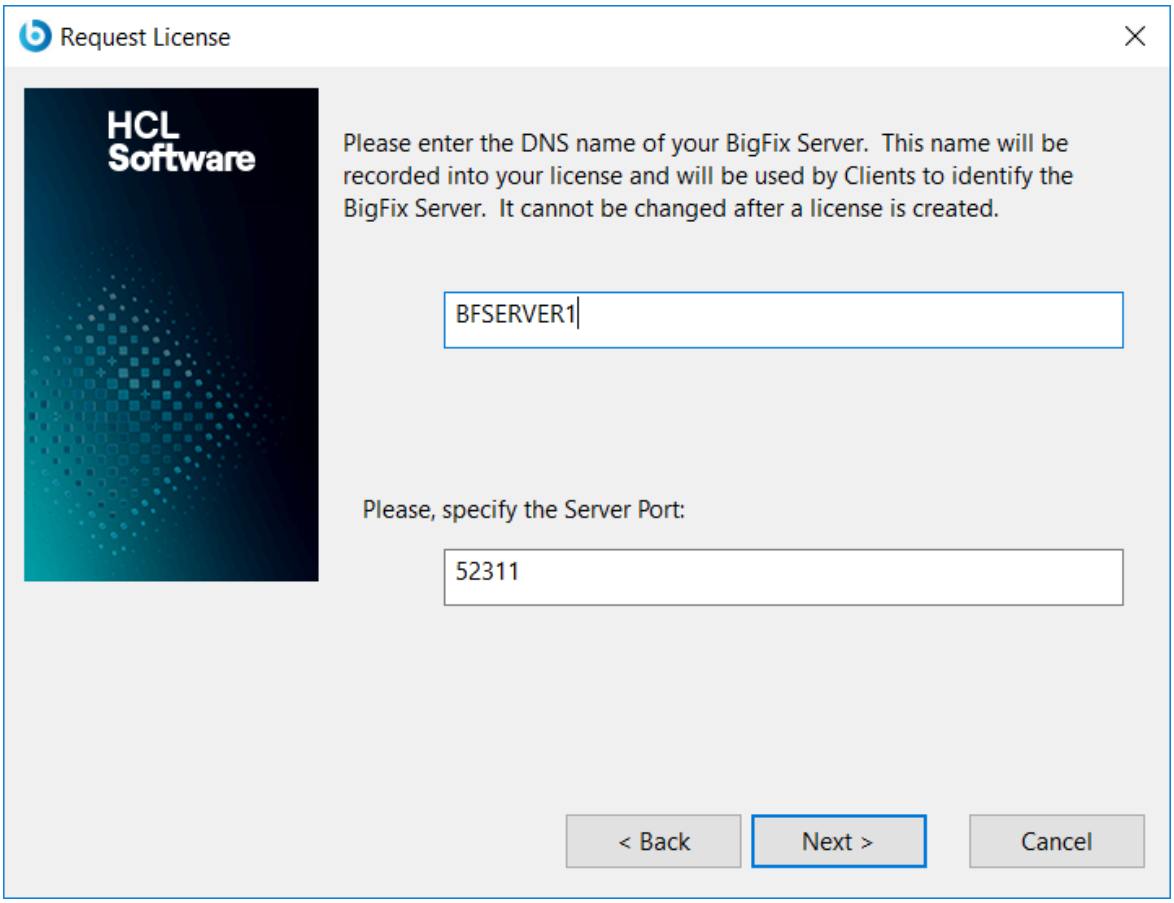
5. After choosing the **Express** option, specify the password for the current user.



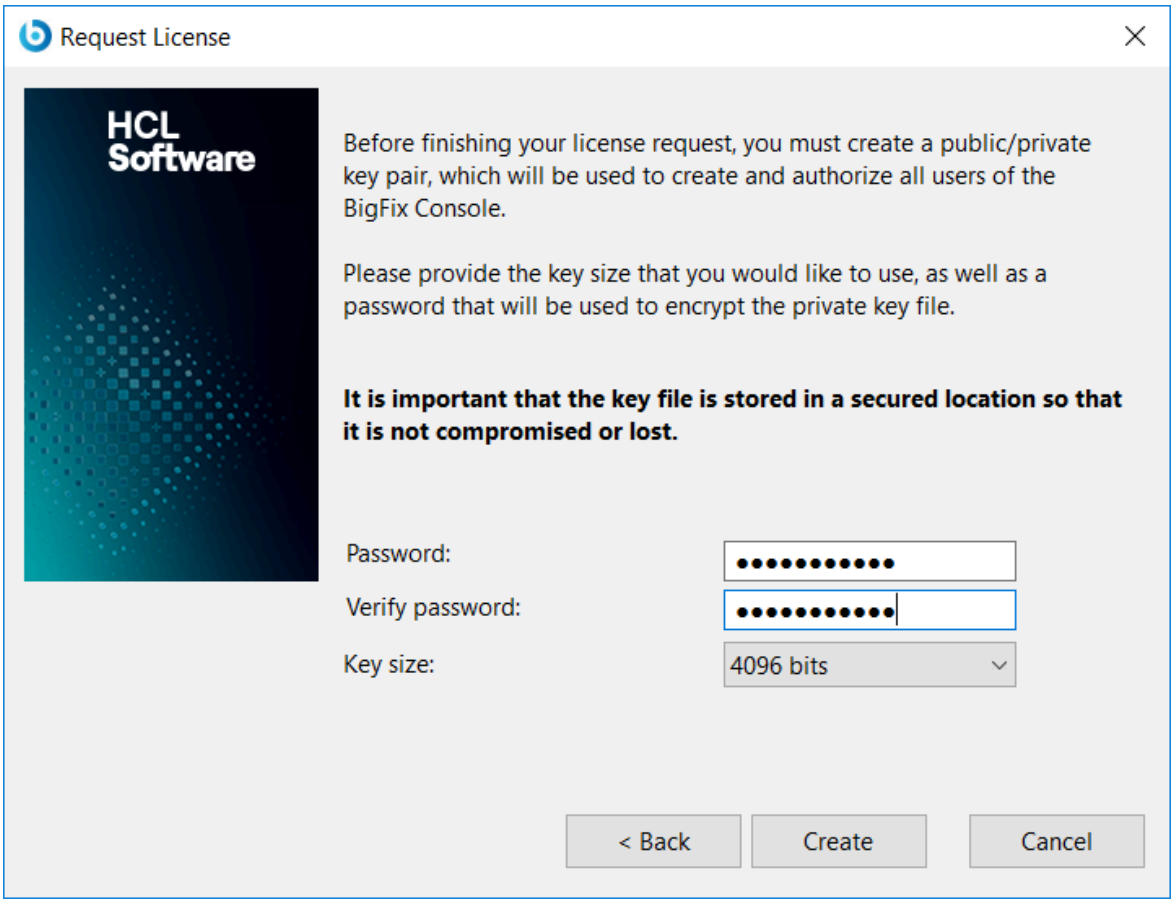
6. Request an evaluation license certificate file. To obtain it, you must enter the following data like in the example below:



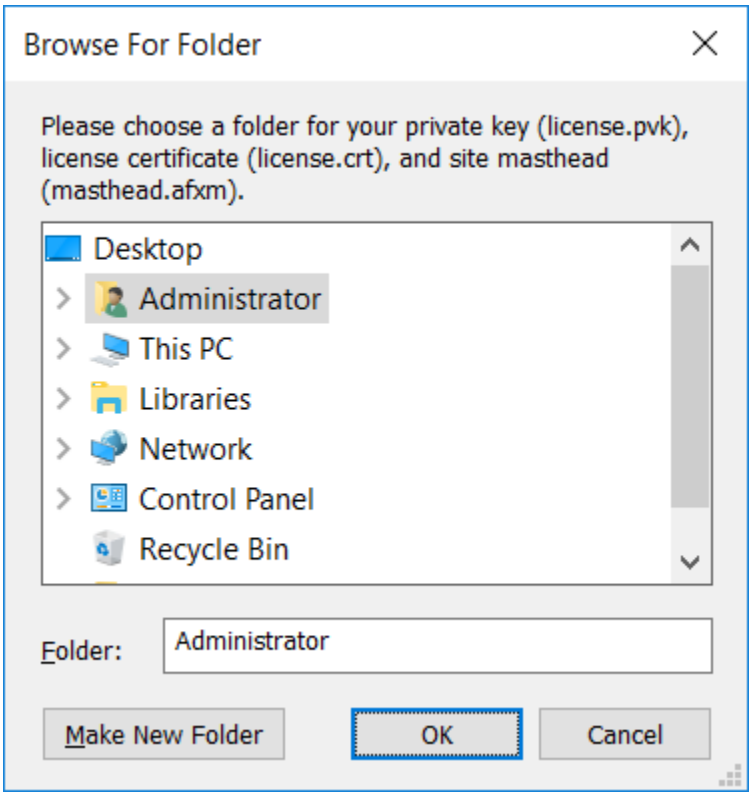
7. Provide the domain name of the server and port on which you are going to install (isolated server), or leave the local domain name if you are installing BigFix on the local server.



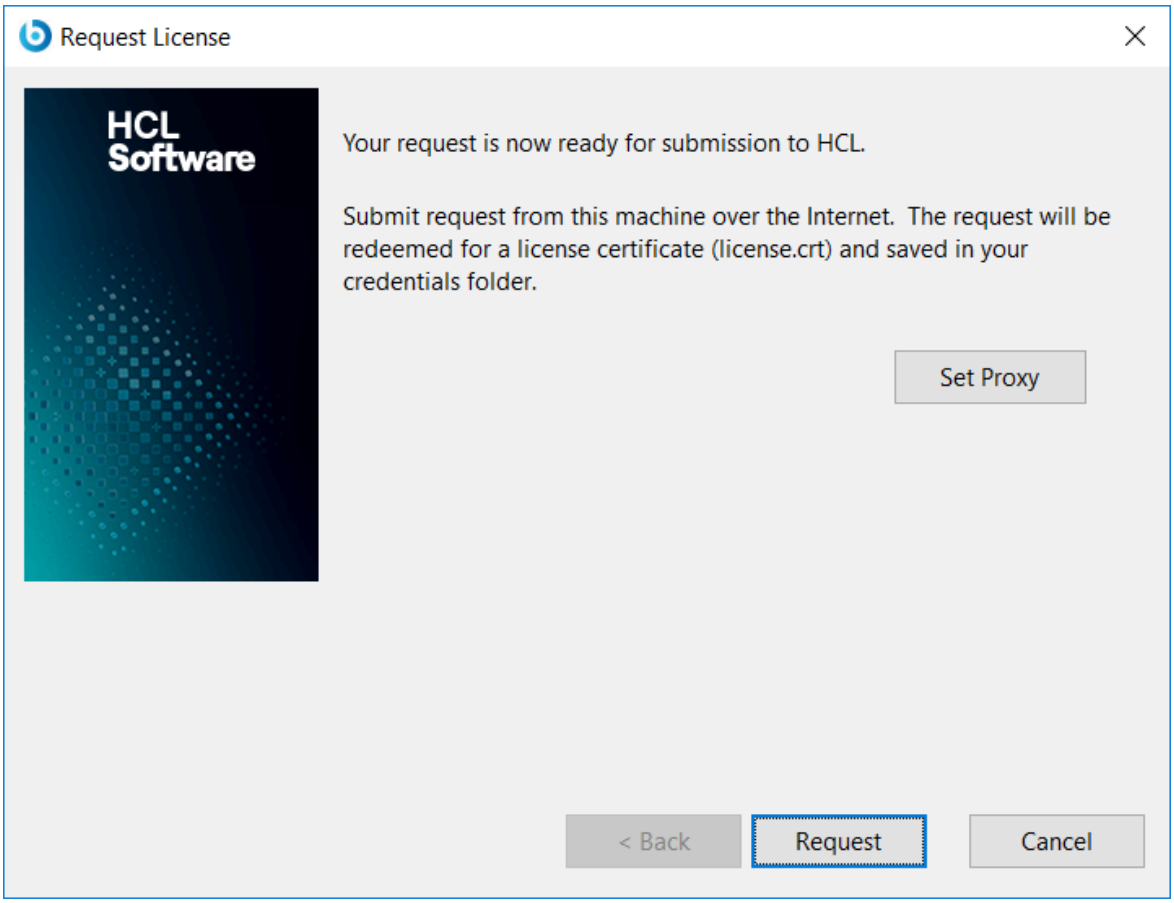
8. Create a public/private key pair and specify a password that will be used to encrypt the private key file.



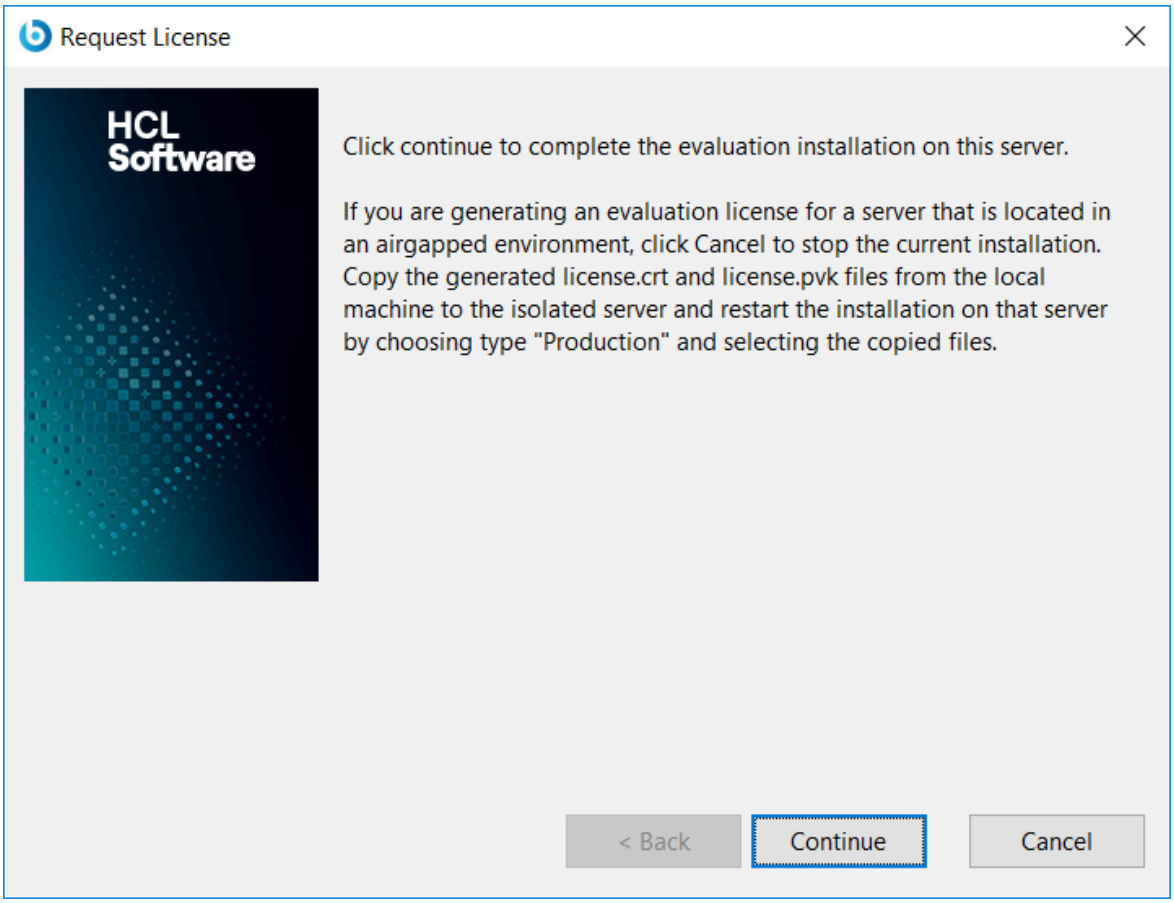
9. Specify the folder where your private key and license certificate will be created.



10. (Optional) Click **Set Proxy** if you use a proxy to connect to Internet.
11. Click **Request** to proceed with the request.



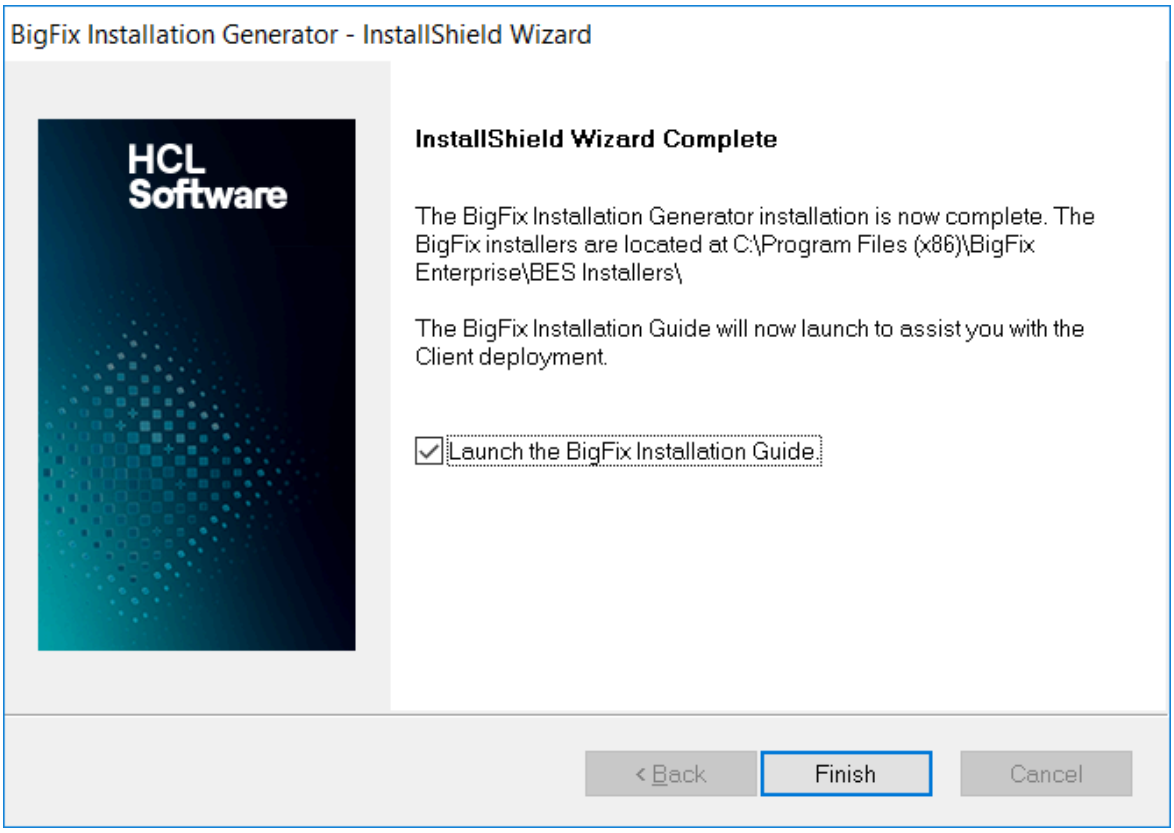
12. Click **Continue** to proceed with the evaluation installation on this local server.



Note: If you want to perform the evaluation installation on an isolated server (server located in an airgapped environment), in this panel click **Cancel** and perform the following actions:

a. Copy the license files, generated for the isolated server, from the current local machine to the isolated server.
b. On the isolated server, launch again the BigFix Installation Generator.
c. Perform the server installation by selecting:
   - **Production** as installation type
   - **I want to install with a production license I already have** as installation option.
d. Provide the location of the license files generated for the isolated server.

13. Click **Finish** to close the BigFix Installation Generator - InstallShield Wizard.

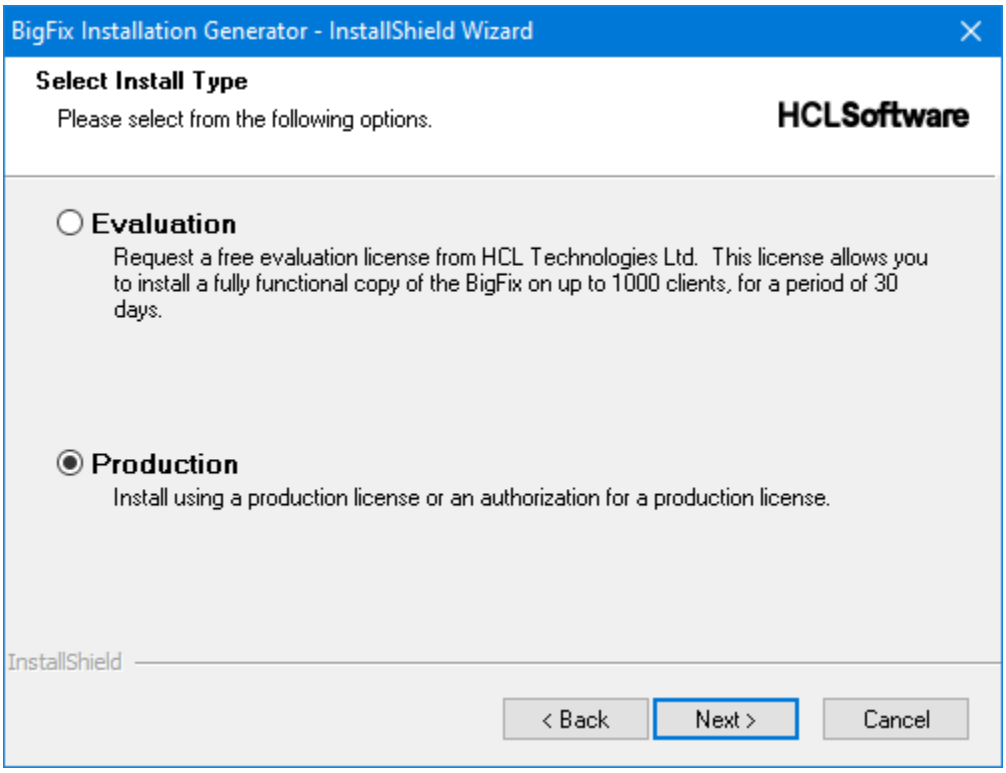


Requesting a license certificate and creating the masthead

Before you perform the steps below, you must have purchased a license and obtained a BigFix license authorization file (`*.BESLicenseAuthorization`) using your My HCL Software account or, in the case of a Proof-of-Concept evaluation, that was provided to you by your HCL Technical Sales Representative.

When you have your license authorization file, you are ready to request a license certificate and then create a personalized **site masthead** that, in turn, allows you to install and use BigFix. The masthead includes URLs for the Server CGI programs and other site information in a signed MIME file. The masthead is central to accessing and authenticating your action site. To create the masthead and activate your site, follow these steps:

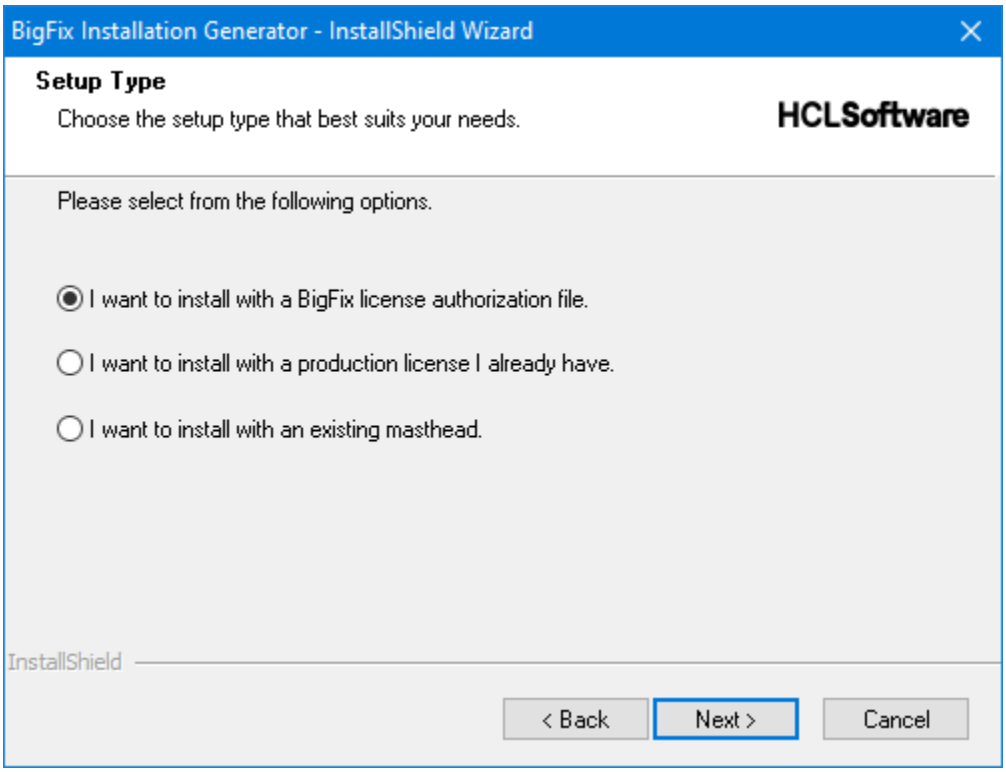
1. Run the BigFix installer `BigFix-BES-11.0.exe`. When prompted, choose **Production** installation:



   Accept the Software License Agreement. On the welcome screen, click **Next**.

   Note: If you choose the **Evaluation** installation, consider that this type of installation does not support the enhanced security option. For more information about this feature, see Security Configuration Scenarios.

2. After reading and accepting the License Agreement, select the first option **I want to install with a BigFix license authorization file**, to create your Private Key and Masthead.



3. Enter the location of your license authorization file, which has a name such as `CompanyName.BESLicenseAuthorization`.
4. Specify a **DNS name** or **IP address** for your BigFix server and click **Next**. The name that you enter in this field is recorded in the license and used by clients to identify the BigFix server.

   Note: Enter a DNS name, such as `bes.companyname.com`, because of its flexibility when changing server computers and doing advanced network configurations. This name is recorded into your license certificate and is used by clients to identify the BigFix server. After your license certificate is created, the DNS name cannot be changed. To change the DNS name, you must request a new license certificate, which requires a completely new installation.

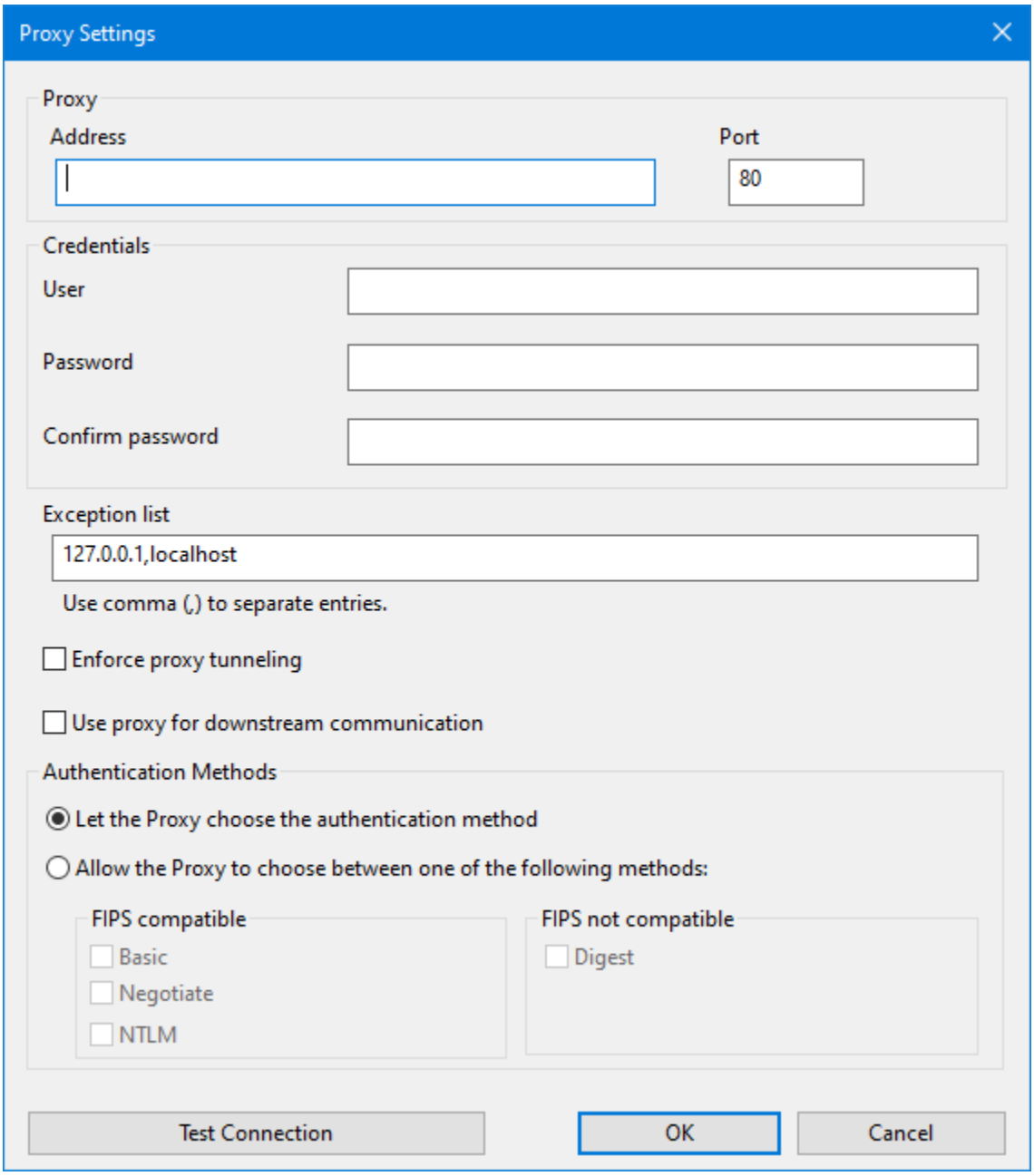
5. Type a site credential **password** to allow you to create a site admin key for your deployment. Type your password twice (for verification), and specify a key size (from 2K to 4K bits) for encrypting the private key file. Make a backup copy of this password

and store it in a secure location. Click **Create**. In this way you generate a private/public key pair used to create and authorize all the BigFix users.

6. Save your private key (`license.pvk`) file from the **Browse for Folder** dialog in a folder with secure permissions or on a removable drive, such as a PGPDisk or a USB drive. Click **OK**.

   Important: If you lose the private key file or the site credential password, a new license certificate needs to be created which requires a completely new installation. In addition, anyone with the private key file and password have full control over all computers with the BigFix clients installed, so ensure that you keep the private key file and password secured.

7. You are requested to send the request file to HCL for license verification. If you have internet connectivity, choose the option to submit your request over the internet. In this case, a request file is sent to HCL for license verification. This request consists of your original authorization file, your server DNS name and your public key, all packaged into a single file.
8. If you select to submit the request over the Internet and your enterprise uses a proxy to access the Internet, click **Set Proxy**. The **Proxy Settings** panel opens. In this panel you can configure the proxy connection.



9. Specify:
    - The hostname or IP Address and, optionally, the port number to communicate with the proxy machine.
    - The credentials of the user defined on the proxy machine that must be used when establishing the connection.

- The comma-separated list of hostnames, subdomains, IP addresses that identify systems in the BigFix topology that must not be reached thru the proxy. By default, BigFix V9.5 prevents diverting internal communications towards the proxy. If you set a value in this field, you overwrite the default behavior. To ensure that internal communications are not directed to the proxy, add `localhost, 127.0.0.1, yourdomain.com, IP_Address` to the list of exceptions specified in this field.
- Whether or not the proxy is enforced to attempt tunneling. By default the proxy does not attempt tunneling.
- The authentication method to use when establishing the communication. You can either let the proxy choose the authentication method or you can impose to use specific authentication methods.



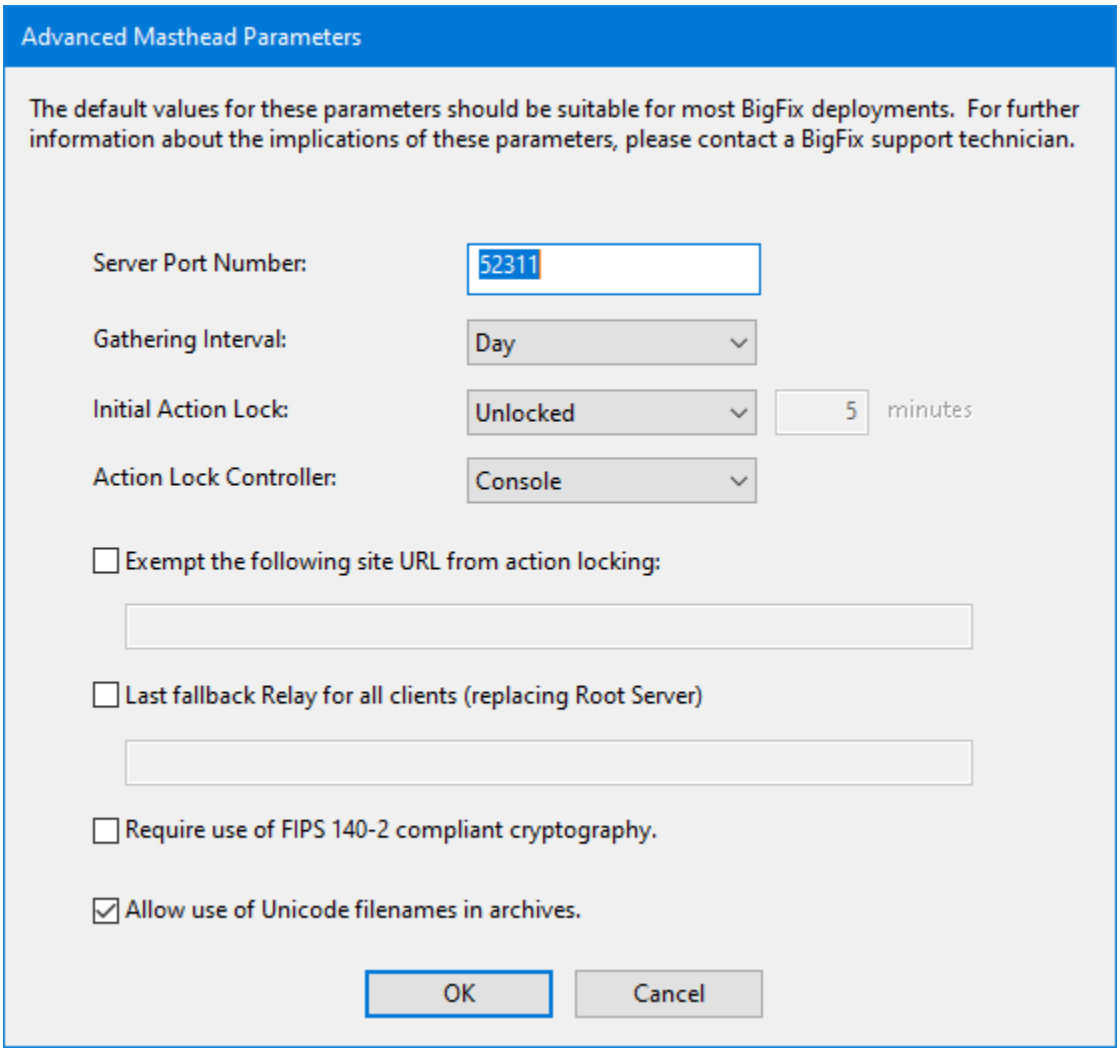
Note: If you want to enable FIPS mode, select an authentication method other than `digest`.

You can click **Test Connection** to verify if the connection with the proxy that you configured can be successfully established. For more information about the values and the syntax to use in these input fields, see Setting a proxy connection on the server *(on page 552)*.

Click **OK** save the settings and return to the **Request License** panel.

10. Click **Request**. The Wizard retrieves your license certificate (`license.crt`) from the BigFix License server.

    Alternatively, if you are on an airgap without internet connectivity, choose the option to save the request as a file named `request.BESLicenseRequest`. Copy the file to a machine with internet connectivity and submit your request to the URL of the BigFix website shown in the installer. The page provides you with a `license.crt` file. Copy the file back to the installation computer and import it into the installer.
11. From the **Request License** dialog, click **Create** to create the masthead file.
12. 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 set 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. If you do decide to change this number *after* deploying the clients, BigFix will not work correctly. For

additional information, see Modifying port numbers *(on page 111)*. see *Modifying port numbers*.

Note: Do not use port number 52314 for the network communication between the BigFix components because it is reserved for proxy agents.

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.

Note: You can specify only one site URL and it must begin with `http://`.

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. By default, the client continues in non-FIPS mode if it fails to correctly enter FIPS, which might be a problem with certain legacy operating systems. Be aware that checking this box can add a few seconds to the client startup time.

For more information see FIPS 140-2 cryptography in the BigFix environmentFIPS 140-2 cryptography in the BigFix environment in the Configuration Guide.



Note: Enabling FIPS mode might prevent 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`.

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.

Click **OK** when you are finished.

13. Choose the folder in which to install the BigFix component installers. The BigFix Installation Guide wizard is launched to lead you through the installation of the BigFix components.

    Note: This step creates the installers for the BigFix client, BigFix console, and BigFix server, but does not install the components.

Note: The private key (`license.pvk`) authorizes the creation and rotation of server signing keys, which are trusted by all agents. This key is *not* sent to HCL during the license certificate creation process, and must be carefully protected. Create also a backup copy of the credential password that you used to encrypt the private key file, and store it in a secure location. To reinstall the server on your workstation, you must reuse the stored BigFix credentials. If you lose the private key file or the site credential password, a new license certificate needs to be created which requires a completely new installation.

Installing the components

You have now created a private key, requested and received a certificate, used the certificate to create a masthead, and then generated the various installation components, including the **BigFix Installation Guide**.

When the components have been saved, the **BigFix Installation Guide** automatically launches. You can also run it at any time by selecting it from the Start Menu.

To install the three major components of BigFix (server, console, and client), follow these steps:

1. If it is not already running, launch the Installation Guide (**Start > Programs > BigFix > BigFix Installation Guide**).
2. A dialog box opens, prompting you to select a component to install. Click the links on the left, in order from top to bottom, to install the BigFix components. You can also Browse Install Folders. The component installers includes:

- Install Server
- Install Console
- Install Clients
- Install WebUI

3. The BigFix server, console, clients and WebUI all have their own installers. Follow the instructions for each, as described in the following sections.



Note: All BigFix Platform components and their pertinent processes are signed by the **HCL America Inc.** signer, as displayed by the **Properties**, in the **Digital Signatures** tab, of the related .EXE files.

Managing the Server ID limit

The Server ID is the identifier of the BigFix server.

In a single instance server deployment, the Server ID is 00. If there are other servers in the deployment that are configured for DSA replication with the primary server,they will assume the following values: 01, 02, 03, etc..

With BigFix 9.5.10, the maximum number of Server IDs you can use is 32 (from 00 to 31). Consequently, the number of allowed DSA servers decreases from 256 to 32.

The following checks are run:

- In the server upgrade scenario, a pre-check is run to ensure that the DSA servers that are already configured are not more than 32, otherwise an error message to reduce the Server IDs is issued. To solve the problem, contact HCL Customer Support.
- When installing a new DSA server, an additional check ensures that the upper limit for the Server IDs does not exceed a given threshold, otherwise a warning is logged in `BESAdminDebugOut.txt`. The same check inhibits or prevents the new DSA server installation if the limit is reached.
- When you replicate the database using the Replication tab in the BigFix Admin tool, the BESAdmin command or the RESTAPI, if the Server ID is between 26 and 31, a warning is logged in the `BESAdminDebugOut.txt` and `BESRelay.log` respectively.

Installing the Windows primary server

The BigFix server is the heart of the system. It runs on a server-class computer on your network, which must have direct Internet access as well as direct access to all the client computers in your network.

Ensure that your server meets the requirements outlined in Detailed system requirements.

Important: Ensure that the user that logs in to install the BigFix server has `sysadmin` rights for the MSSQL Server to create the database and its tables.

Note: If you removed Microsoft SQL Server from your system where you plan to install BigFix, ensure that all the Microsoft SQL components are correctly deleted before running the installation.

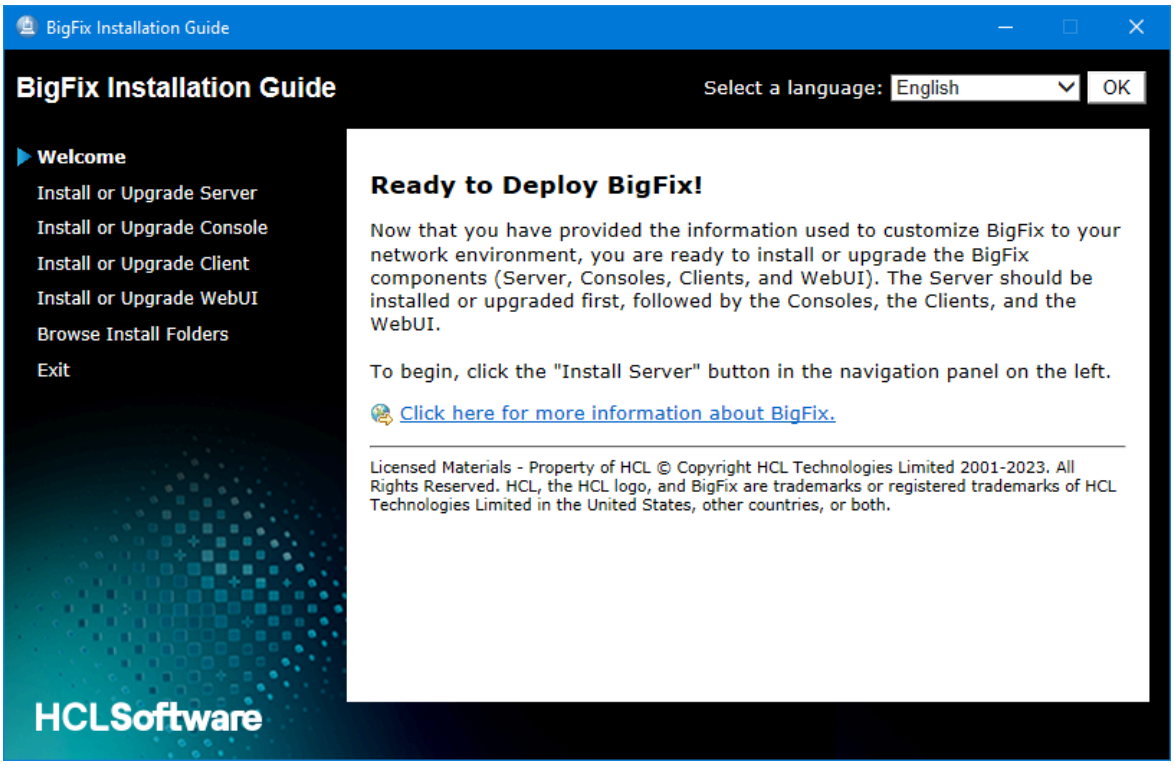
Note: The version of Microsoft SQL Server installed with BigFix is SQL Server Evaluation, which is a fully-functional version for a limited time (180 days).

The default installation path for the BigFix components is `%PROGRAM FILES%\BigFix Enterprise\BES Server`. If you want to change this customizable path, remember that it must be made only by ASCII characters.

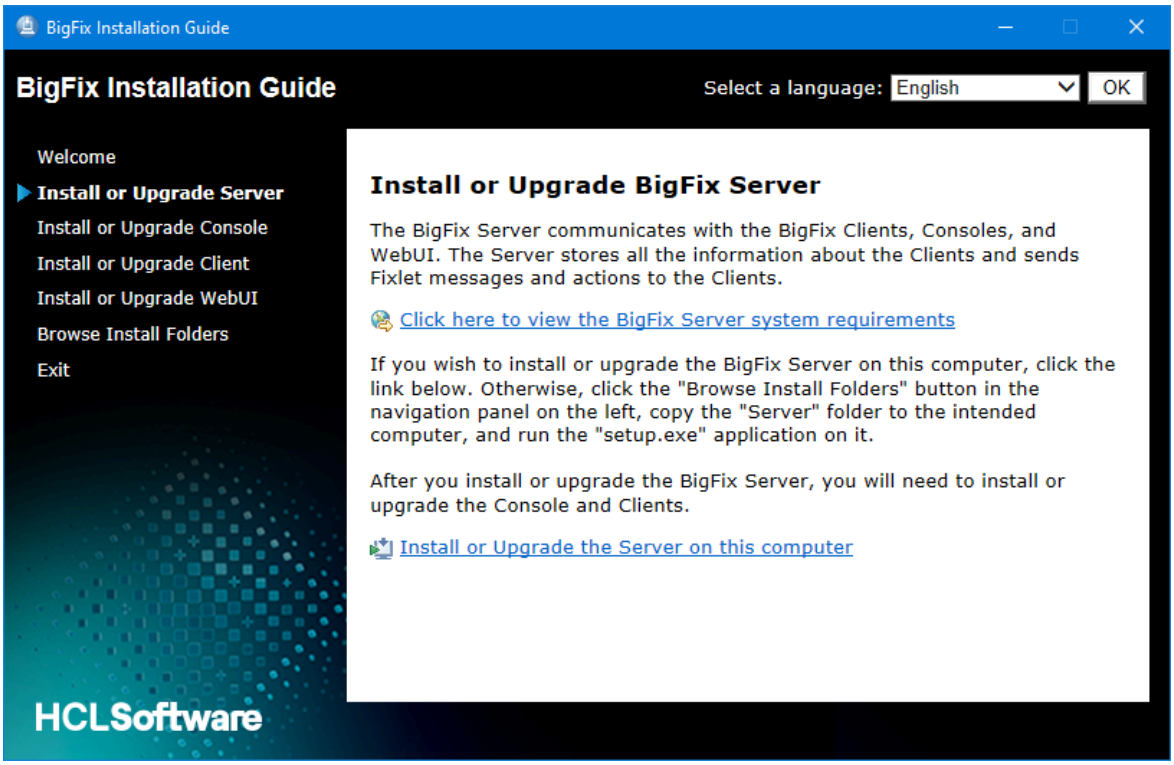
Note: On Windows the BigFix V11 server and Web Reports components support only 64 bit architecture. For information about the complete list of operating systems supported, see Detailed system requirements.

To install the server, follow these steps:

1. If you have not already done so, run the Installation Guide (**Start > Programs > BigFix > BigFix Installation Guide**). A new panel opens.



2. Click **Install or Upgrade Server**:



Click **Install or Upgrade the Server on this computer** to install the server locally.

If you want to install the server on a different computer run the following steps:

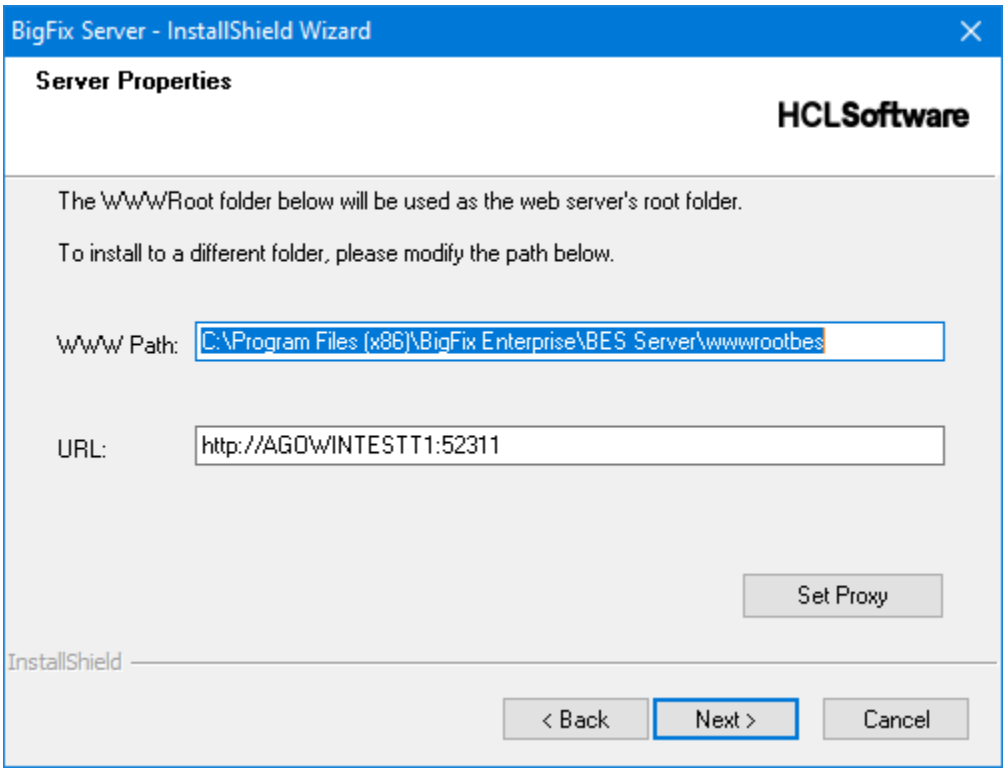
a. Click **Browse Install Folders**.

b. Copy the Server folder to the target computer.

c. On the target computer double-click `setup.exe` to launch the installer.

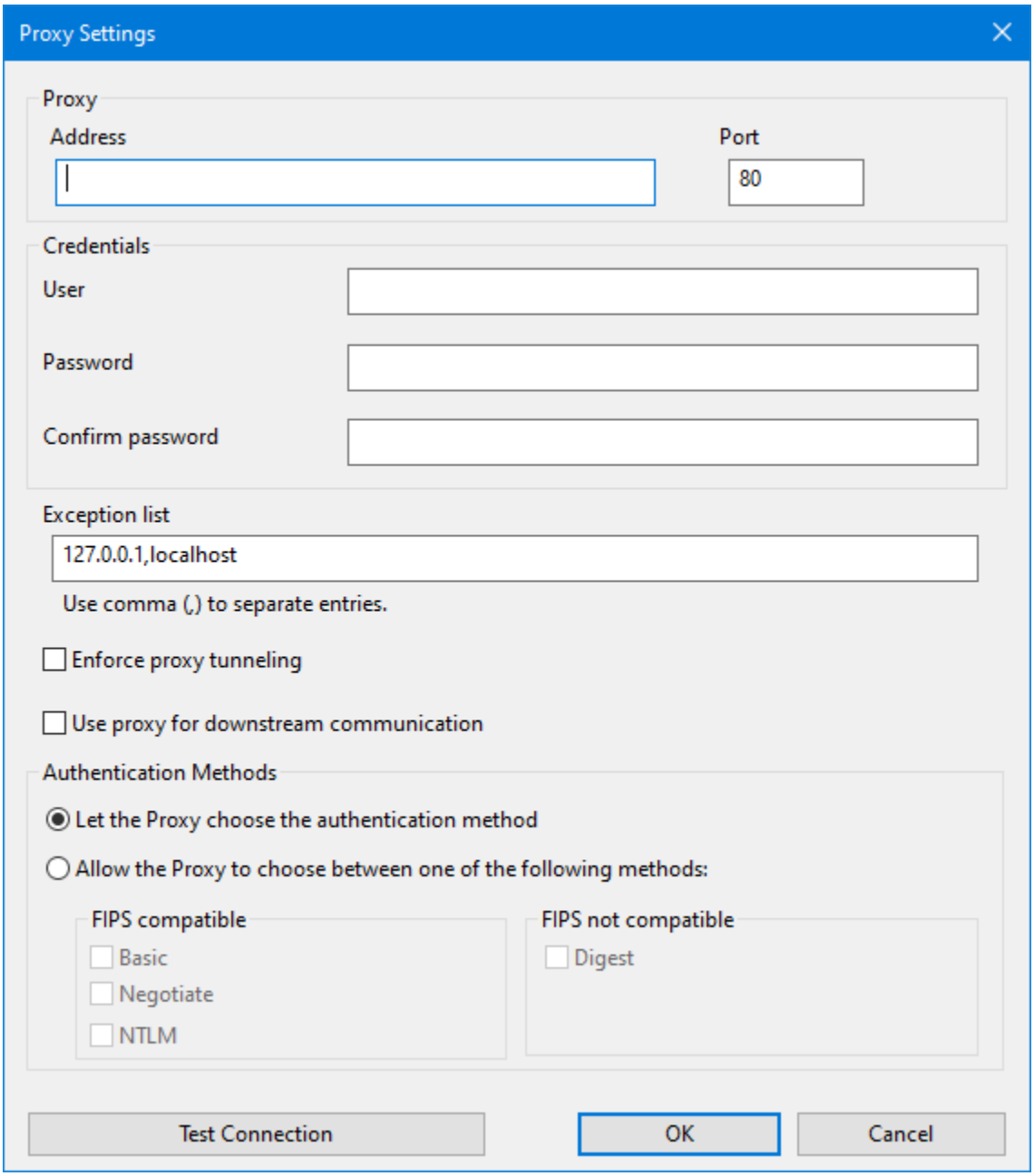
3. On the welcome page, click **Next**.
4. Select the features that you want to install and click **Next.**.
5. After reading the **License Agreement**, click **Yes** to accept it and continue.
6. A dialog displays a list of the Server components about to be installed. In general, accept the default components and click **Next**.
7. A dialog prompts you to choose a **Single or Master Database** or a **Replicated Database**. Click the first button to create a **Master** database for later replication or if you only need a **Single** database in your deployment. Click the second button to create a **Replica** of an existing Master. If this is your initial installation, click the first button. Click **Next**.

8. A dialog prompts you to choose if you want to **Use Local Database** or **Use Remote Database**. If you want to use another computer to host the BigFix Database, it must have a SQL Server already installed. The most common choice is to use the local database. If you are installing BigFix with a remote database, see Installing a server with remote database *(on page 153)*.
9. The installer prompts you for a destination for the Server components. The default location is `%PROGRAM FILES%\BigFix Enterprise\BES Server`, but you can specify a different location by clicking the **Browse** button. If you want to change this customizable path, remember that it must be made only by ASCII characters. When you have chosen the destination, click **Next**.
10. The Server Properties dialog prompts you to enter a location for the Server web root folder (if different from the default). This is where downloaded files for the Clients will be stored. The default URL is also available for editing, if you want to change it.



Note: No other application can be listening on the BigFix port or errors will occur. Do not use port number 52314 for the network communication between the BigFix components because it is reserved for proxy agents.

11. In the Server Properties dialog, click **Set Proxy** if a proxy must be used to communicate over the internet to external content sites or to BigFix subnetworks. The **Proxy Settings** panel opens. In this panel you can configure the proxy connection.



12. Specify:
    - The host name or IP Address and, optionally, the port number to communicate with the proxy machine.
    - The credentials of the user defined on the proxy machine that must be used when establishing the connection.
    - The comma-separated list of host names, subdomains, IP addresses that identify systems in the BigFix topology that must not be reached thru the proxy. By default, BigFix prevents diverting internal communications towards the proxy. If you set a value in this field, you overwrite the default behavior. To ensure that internal communications are not directed to the proxy, add `localhost, 127.0.0.1, yourdomain.com, IP_Address` to the list of exceptions specified in this field.
    - Whether or not the proxy is enforced to attempt tunneling. By default the proxy does not attempt tunneling.
    - The authentication method to use when establishing the communication. You can either let the proxy choose the authentication method or you can impose to use specific authentication methods.

    Note: If you plan to enable FIPS mode, ensure that the proxy configuration is set up to use an authentication method other than `digest`.

    Click **Test Connection** to verify if the connection with the proxy that you configured can be successfully established.

    For more information about the values and the syntax to use in these input fields, see Setting a proxy connection on the server (on page 552).

    Click **OK** to proceed with the next step.

    Note: The proxy configuration specified at this step is saved in the server configuration file `BESServer.config` and it is used also at runtime.

13. The Web Reports Properties dialog prompts you to enter:

- A location for the Web Reports web root folder (`WWWRoot`), if different from the default.
- The port number to use. The default value is `8083`. During the installation, the Web Reports component is configured to use the HTTPS protocol. If you want, you can modify the Web reports configuration, after the installation completes successfully.
- The user installing the Web Reports component. You can choose to create the user during the installation or to use an already existing user. The default user is *LocalSystem*.

14. The Server installer opens a window displaying the selected installation parameters of the components to be installed. Click **Next** to continue the installation.
15. The program prompts you to locate your `license.pvk` file. Click the **Browse** button to locate the file. Enter your password to initialize the database and click **OK** to continue.
16. After the database has been initialized you are prompted to enter your initial username and password for the BigFix console. This is the account used to log in to the console the first time. It is a fully privileged master operator account.
17. The BigFix server installation is now complete. Ensure that the box labeled Run the BigFix Diagnostic Tool is unchecked and then click **Finish** to exit the wizard.

    Note:

    If you select to run the diagnostic tools at this stage, some steps are likely to fail (for example, you have not installed a client yet). However, the services and web reports should be running correctly.

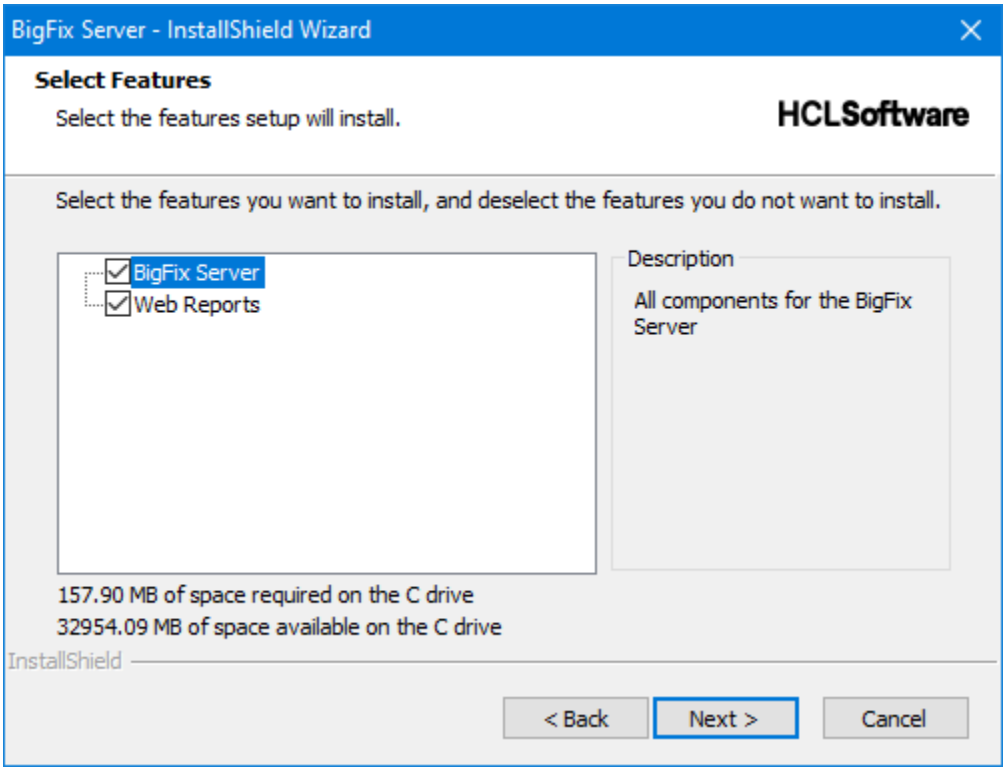
18. Follow the instructions listed in Installing the clients *(on page 283)* to install the BigFix Client locally on the same Windows system where you installed the Server.
19. On the Windows desktop select **Start > Run the BigFix Diagnostic Tool**. The BigFix Diagnostic Tool tabs show the results of the verification run in your environment. For more information about this tool, see Running the BigFix Diagnostics tool *(on page 161)*.

Installing a server with local database

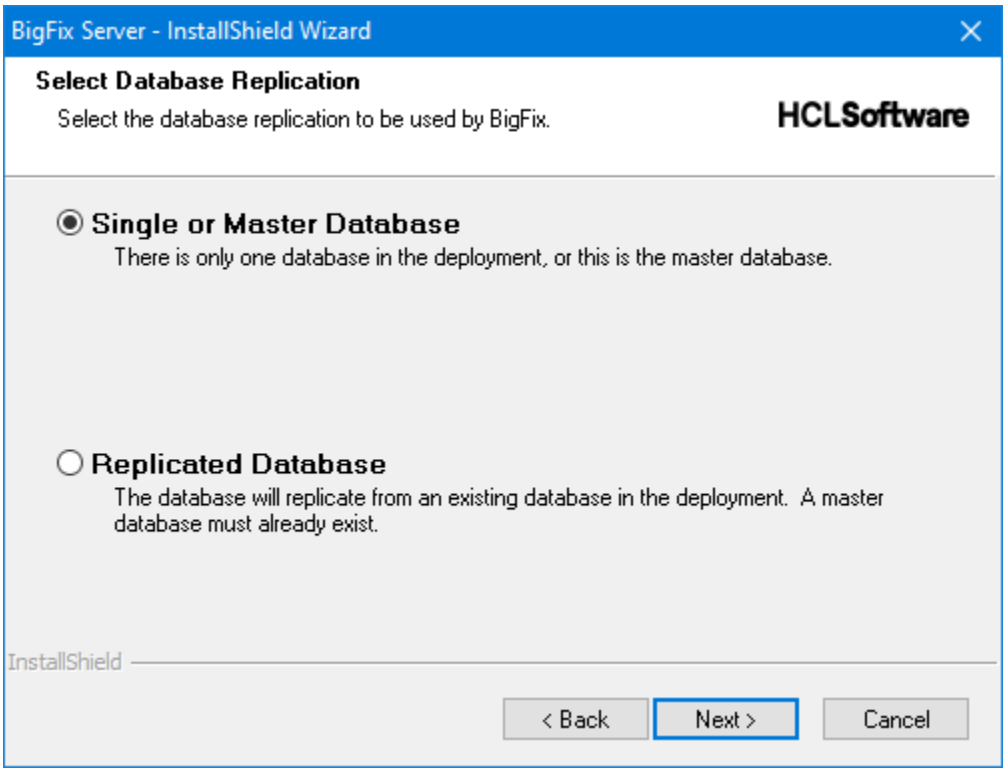
When performing a fresh installation of BigFix Server Version 11, you can either perform an evaluation installation or a production installation.

To install a BigFix server with a production license, perform the following steps:

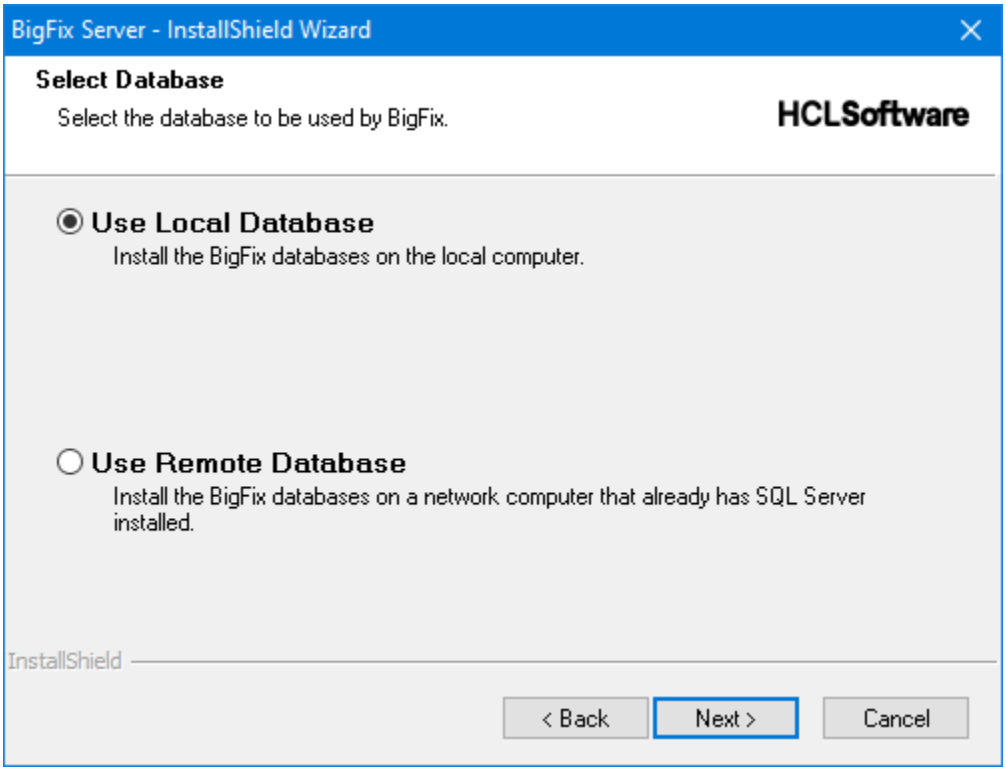
1. On the computer where you want to install the BigFix server, run the BigFix Server - InstallShield Wizard.
2. Select the features that you want to install.



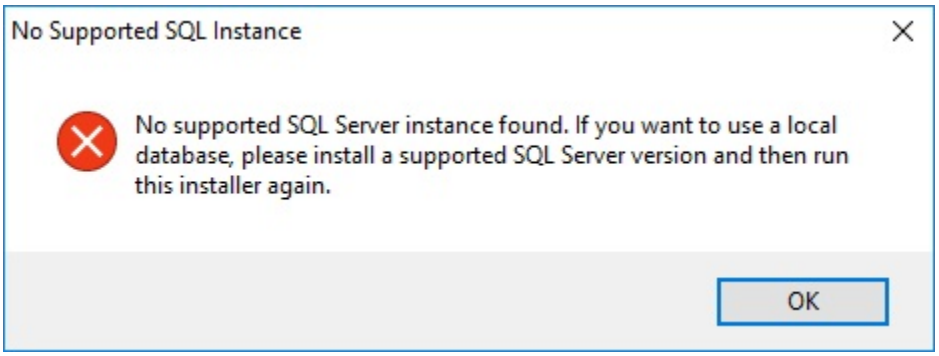
3. During the server installation, select **Single or Master Database** as database replication.



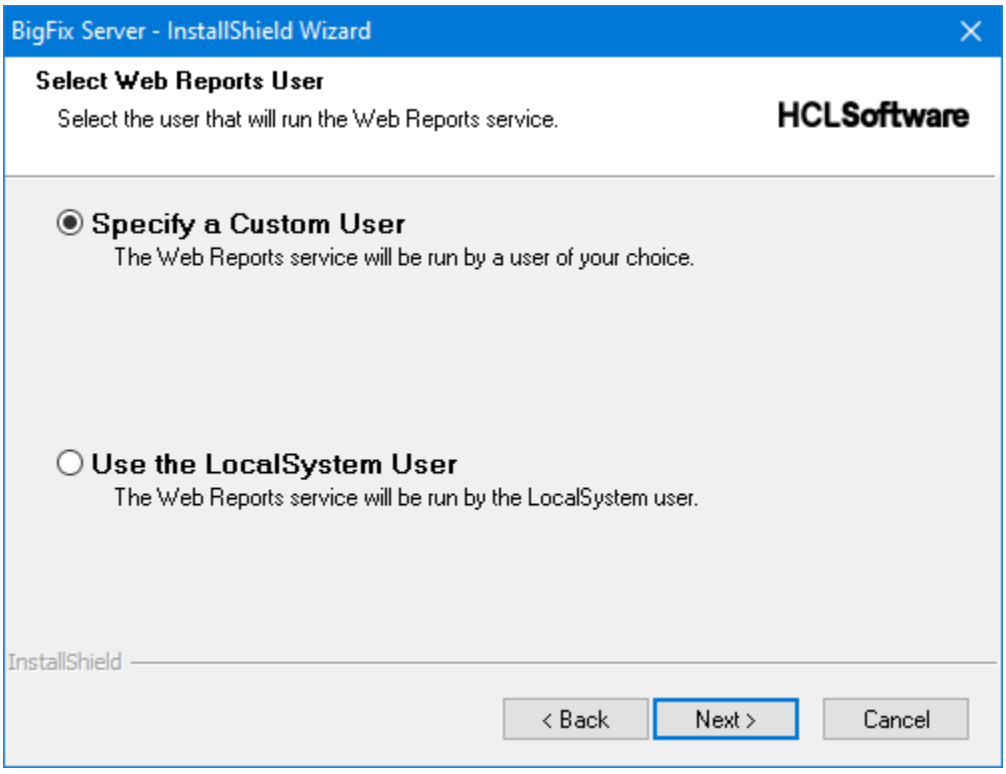
4. Select **Use Local Database** as the type of database.



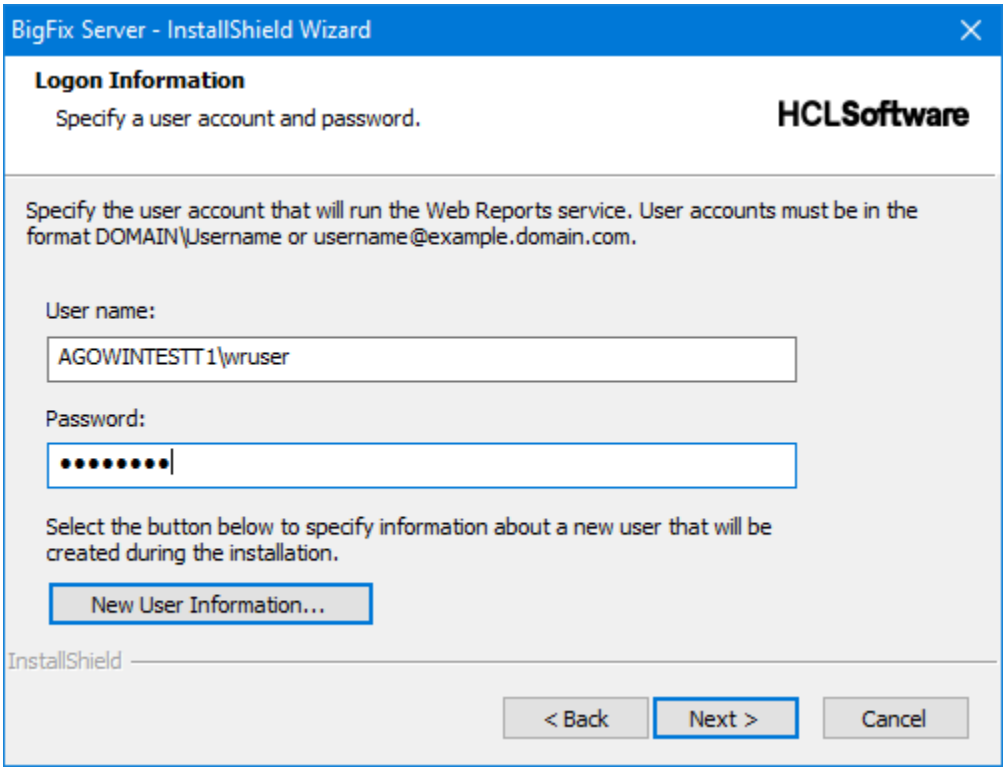
5. After choosing the **Use Local Database** option, if no supported SQL Server instance is found on the local computer, the followig error message is displayed:



6. The set up shows you the destination folder where it installs:
    a. The BigFix Server.
    b. The Web server wwwroot folder.
    c. The Web Reports wwwroot folder.
7. Specify the type of user that will run the Web Reports services, either a custom user or the LocalSystem user.



8. If you choose to specify a custom user, you can enter the credentials of an existing Windows user, or enter the information needed to create a new user during the installation:



9. Review all the installation information displayed and click **Next**.

Installing a server with remote database

Prerequisites to install with a remote database.

Before installing a BigFix server with a remote database, ensure that:

- You install the BigFix Server as a user with SA privileges.
- The SQL Server Browser is running.
- The Windows Authentication or the SQL Server Authentication is enabled.

Creating a new database user

After creating a database instance on the machine where the Microsoft SQL Server is installed, if you do not want to use the SA user for the database connection, you must create a new user with the `sysadmin` server role.

To create a new user for a specific database instance, for example `BIGFIX`, perform the following steps:

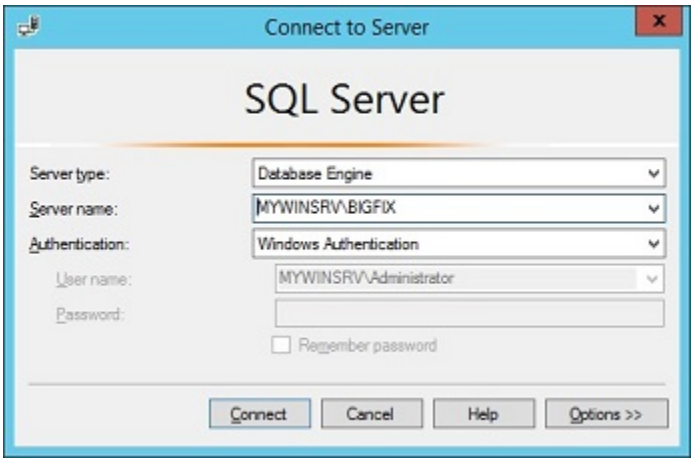
1. Start the Microsoft SQL Server Management Studio.
2. In the Connect to Server panel, specify the following parameters:

   Server Type

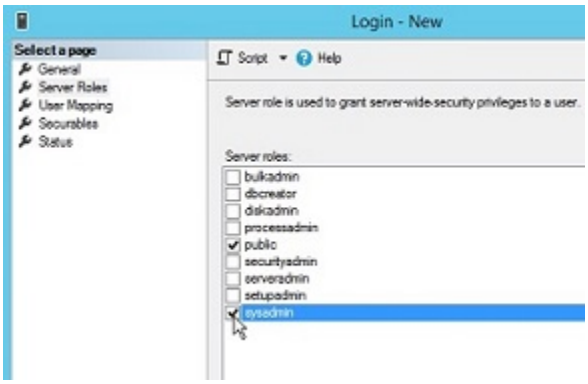
   Database Engine

   Server Name

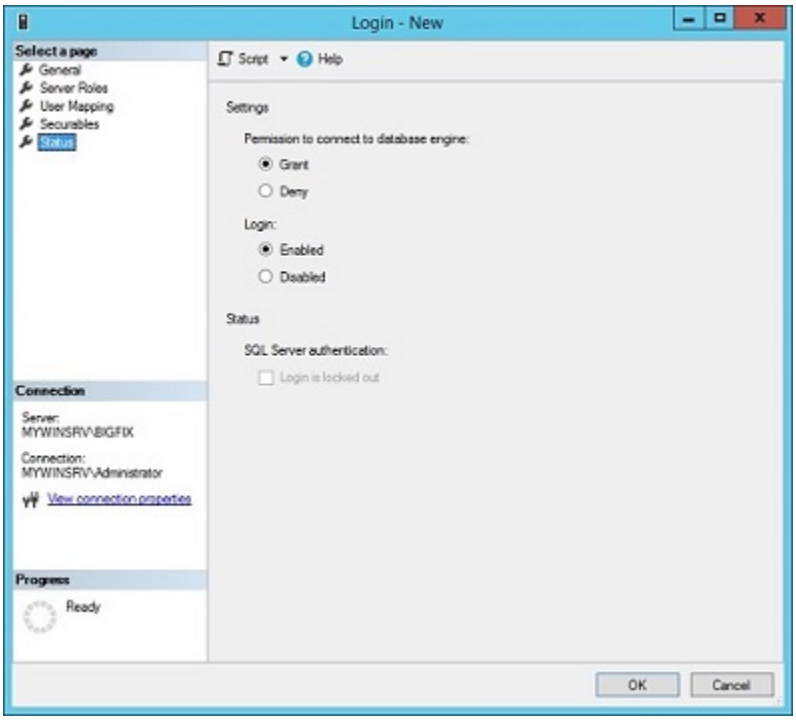
   `<DB_HOSTNAME>\<INSTANCE_NAME>` If the server host name is `MYWINSRV`, and the instance name is `BIGFIX`, the server name is: `MYWINSRV\BIGFIX`.



3. From the portfolio, select **Security -> Login -> New Login**.
4. In the **General** tab, specify the User Name and the credential for SQL Server Authentication and click **OK**.
5. In the **Server Roles** tab, select **sysadmin** and click **OK**.



6. In the **Status** tab, ensure that the following selections are displayed and click **OK**.

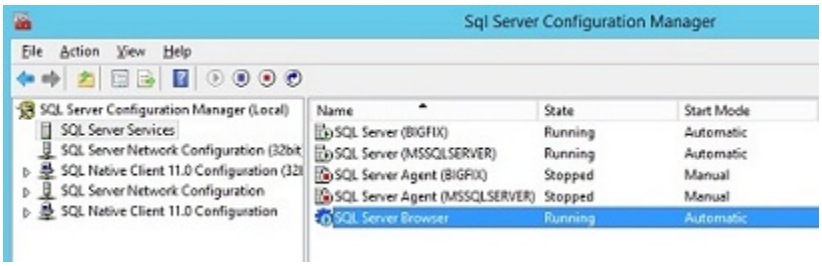


Starting the SQL Server Browser

On the computer where the Microsoft SQL Server is installed, ensure that the SQL Server Browser is running.

Perform the following steps:

1. Start the **SQL Server Configuration Manager**.
2. Select **SQL Server Services** and start the SQL Server Browser if it is not running:

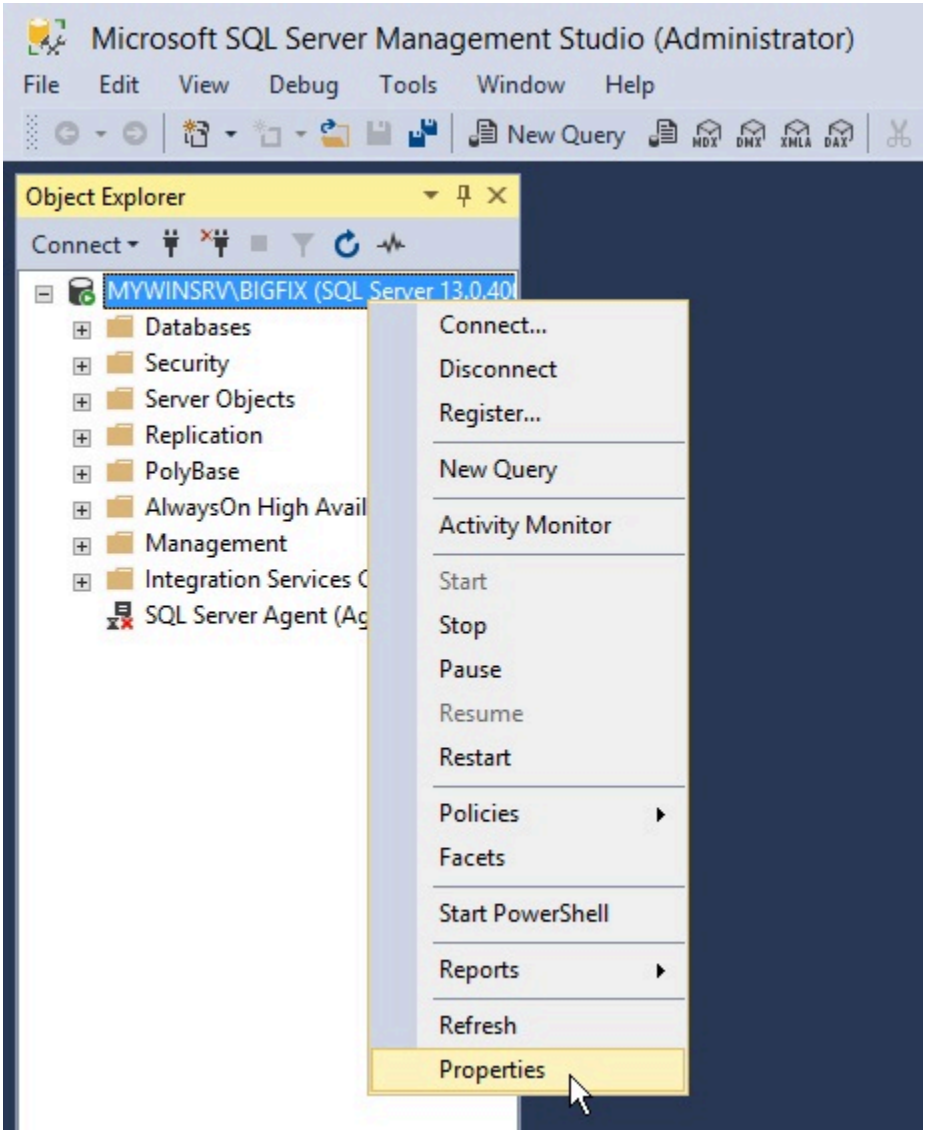


Enabling the SQL Server Authentication Mode

On the computer where the Microsoft SQL Server is installed, ensure that the SQL Server Authentication Mode is enabled.

Perform the following steps:

1. Start the Microsoft SQL Server Management Studio.
2. Select the database instance.
3. Select **Properties > Security**.



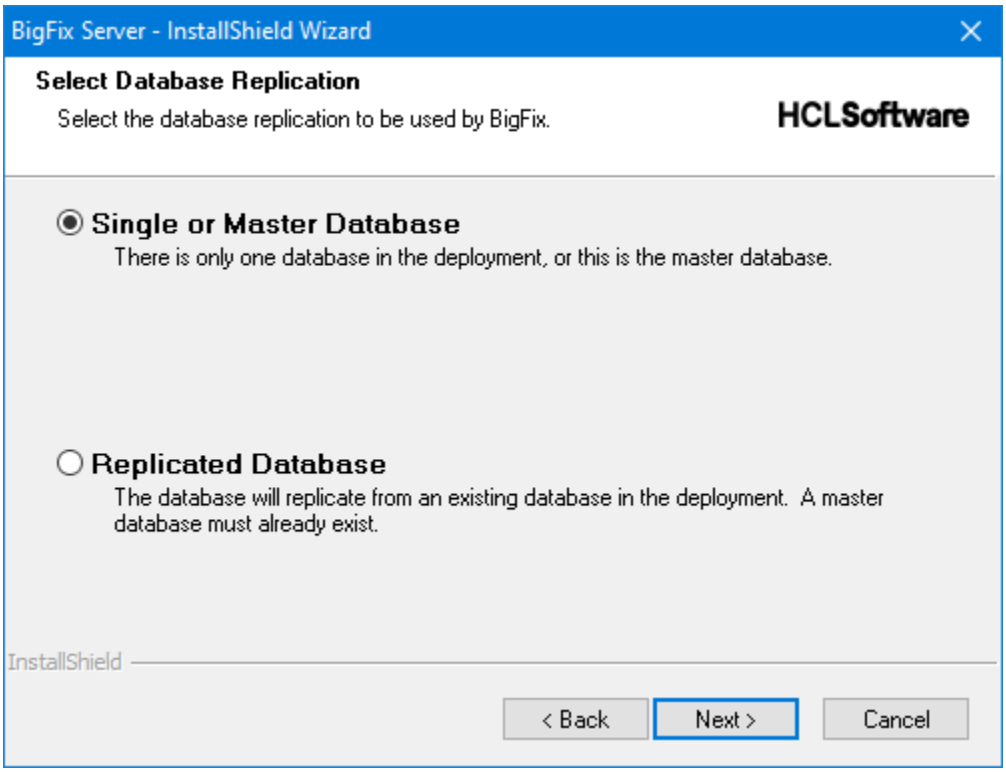
4. Verify that **SQL Server and Windows Authentication mode** is selected.

Installing a server with a remote database

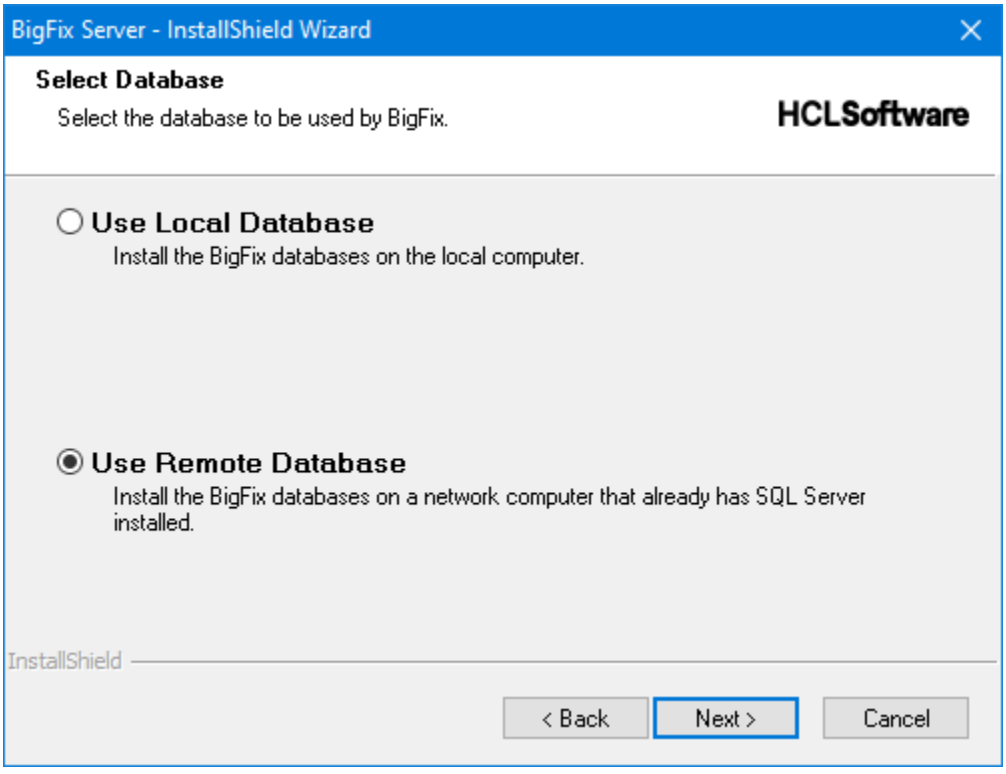
Procedure to install a BigFix server with a remote database.

Perform the following steps:

1. On the computer where you want to install the BigFix server, run the installation.
2. During the server installation, select **Single or Master Database** as database replication.



3. Select **Use Remote Database** as the type of database.



4. In the Database Server window, click **Browse** and select the database server instance you want to use.

   Note:

   During the installation on Windows Servers, when the installer must connect to the remote SQL Server database engine, the computer might display a message related to the Computer Browser Service:

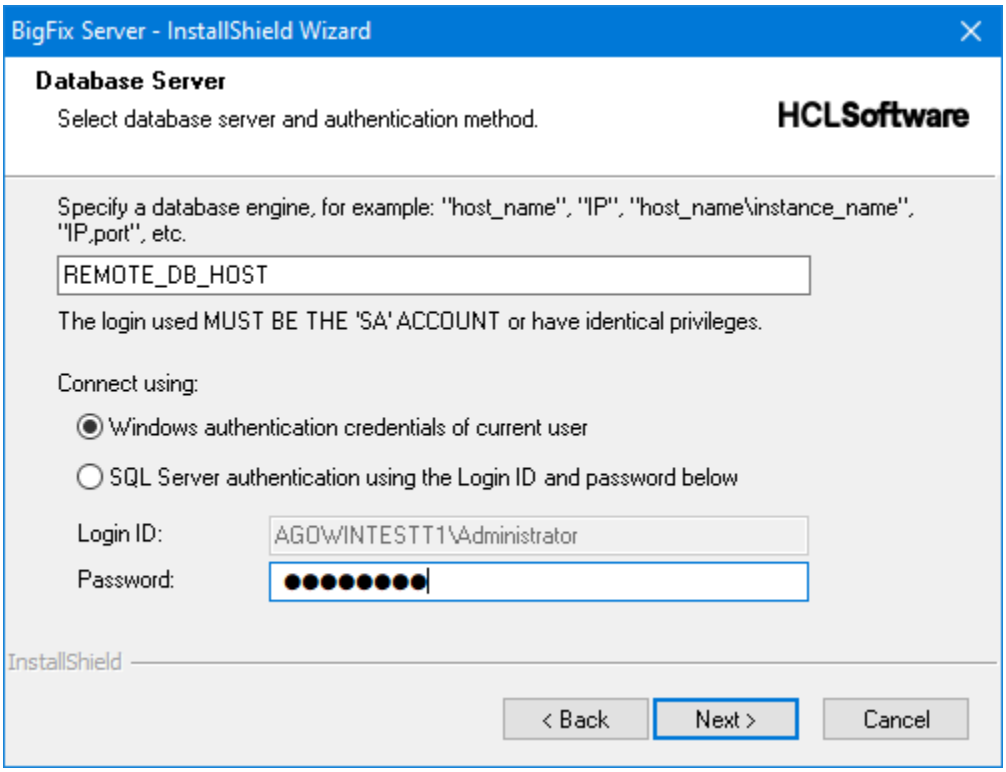
   ```
   Computer Browser Error with Windows Authentication
   ```

   To correct this issue, enable the file and printer sharing option by following these steps:

   a. Go to **Control Panel > Network and Sharing Center**.
   b. Click Change advanced sharing settings.
   c. Click the down arrow next to the network you want to enable file and printer sharing for.
   d. Select **Turn on file and printer sharing**.
   e. Save the changes.

5. Select the authentication method. If you select **SQL Server Authentication using the login ID and password below**, provide the credentials of the user with SA privileges.

   Note: These credentials are stored in clear text in the Windows registry but they get obfuscated after the first time the service using these credentials is started again.



The database is created on the remote computer where the Microsoft SQL Server is installed. On the computer where the BigFix Server is installed, the registry is updated with the database authentication credentials.

Note: If you choose to connect to a remote SQL Server database using the Windows authentication, ensure that you:

- Specify a Windows domain user that can access both the local and the remote computer.
- Specify a Windows domain user that has sysadmin privileges on the database engine, at least during this installation and during the following upgrade.

Moreover, if you choose the Windows authentication, the user chosen to access the remote database will run the BigFix Server services and also the Web Reports server service (if present on the same computer).

Installing a server with remote database deployed in a docker container

From BigFix V10 Patch 2, you can install the Windows server with a remote database using a Microsoft SQL Server Docker image.

The official Docker images developed and supported by Microsoft are available at: https://hub.docker.com/_/microsoft-mssql-server.

The supported configurations are available at: Technical support policy for Microsoft SQL Server.

BigFix Platform supports only Docker images based on Ubuntu 16.04 platform or, in case of Version SQL Server 2022, Ubuntu 20.04 LTS platform.

For more details about the Microsoft SQL Server deployment inside a a docker container, refer to the following Microsoft documentation pages, depending on the version you need:

- Quickstart: Run SQL Server container images with Docker (Version SQL Server 2017 available starting from V10 Patch 2)
- Quickstart: Run SQL Server container images with Docker (Version SQL Server 2019 available starting from V10 Patch 2)
- Quickstart: Run SQL Server container images with Docker (Version SQL Server 2022 available starting from V10 Patch 10)

Note: The Windows authentication described in Installing a server with a remote database *(on page 157)* is not supported.

Running the BigFix Diagnostics tool

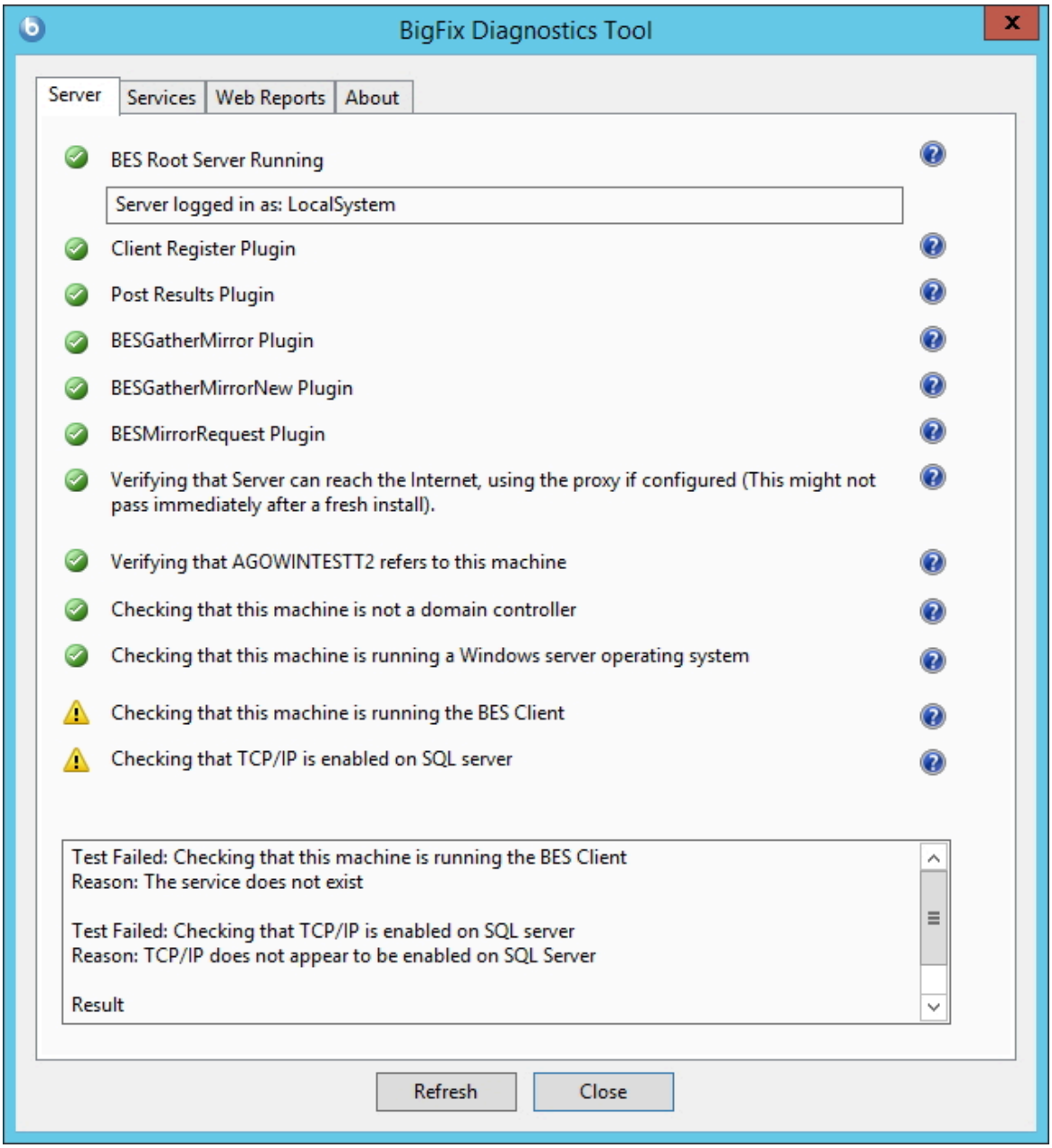
The BigFix Diagnostics tool verifies that the server components are working correctly.

It identifies components that are incorrectly configured or non-functional and displays the results. To run the diagnostics, follow these steps:

1. If you have just installed the Server, the Diagnostics Tool should already be running. Otherwise, log on to the Server as an administrator and launch the program.

   Start > Programs > BigFix > BigFix Diagnostics Tool.

   The program analyzes the server components and creates a report.
2. For more in-depth information, click the **Full Interface** . The BigFix Diagnostic control panel is displayed. This window has tabs corresponding to the categories of server diagnostics, including **Services** and **Web Reports**.

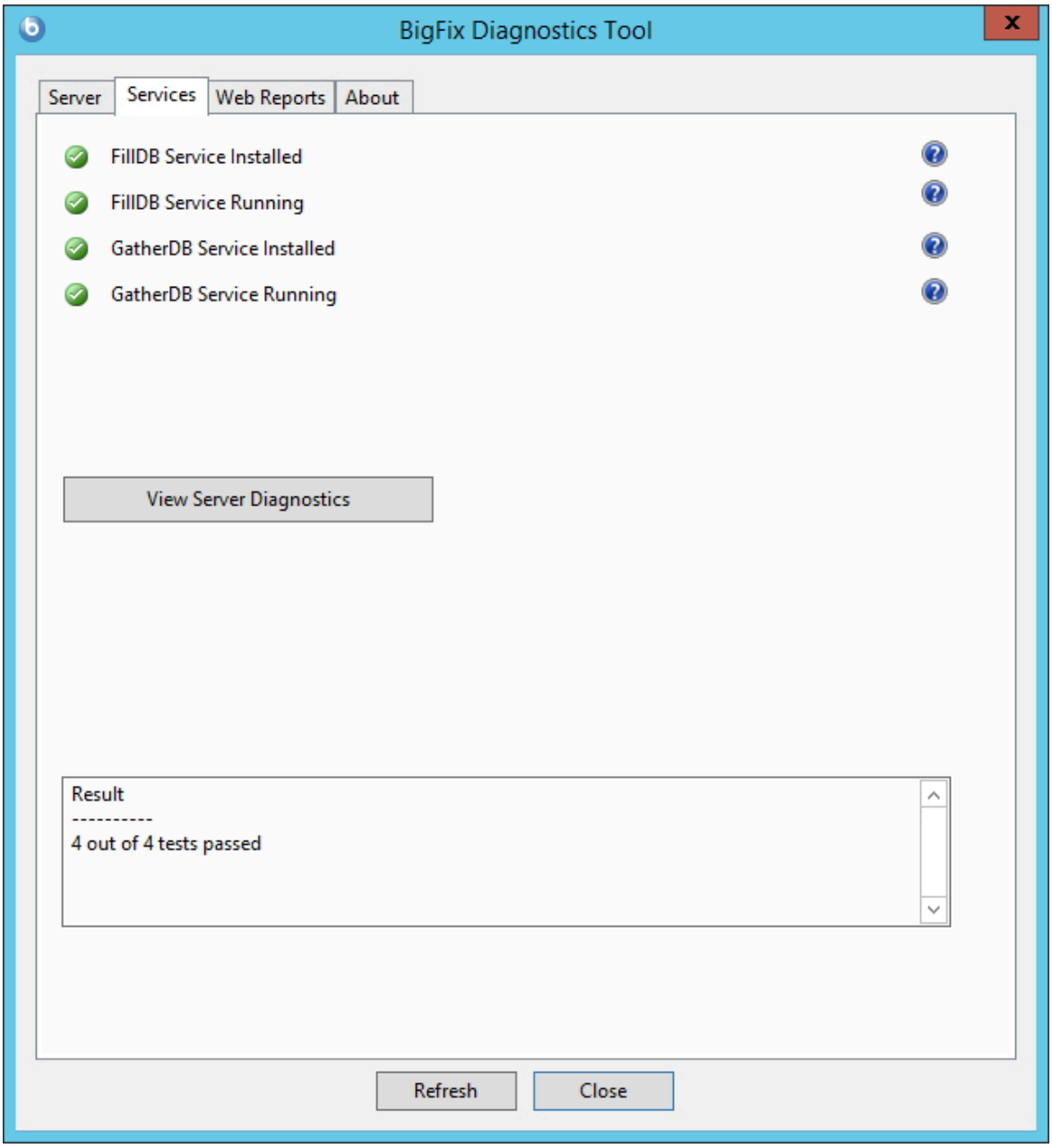




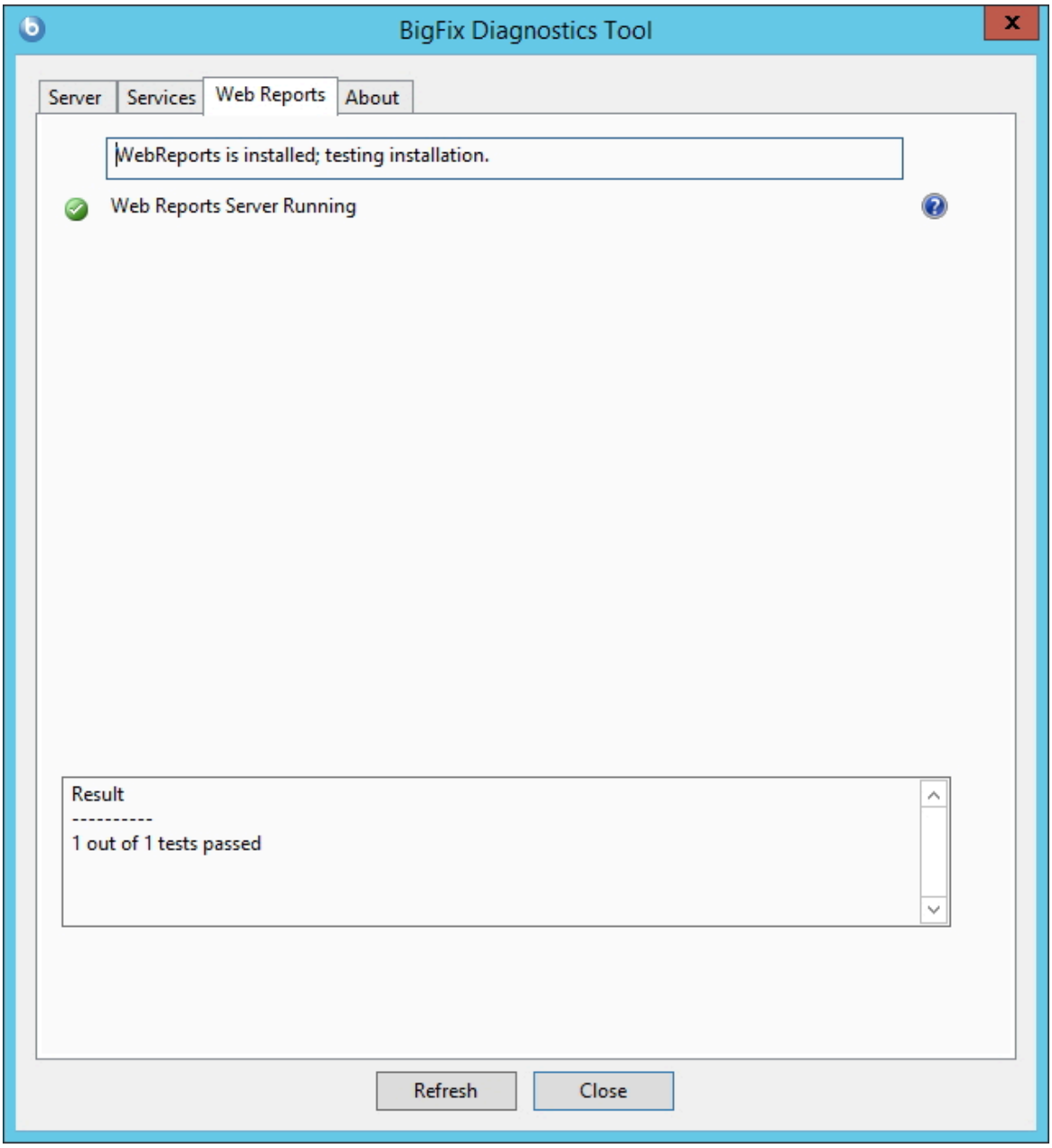
Note: If the message `Verifying that the BESGather service can reach the Internet` is displayed after a fresh install and you have a proxy, ensure that you configured it as described in Setting up a proxy connection *(on page 545)*.

If you have not yet installed the client, a warning light is shown. It becomes green as soon as you install the client.

3. In the **Services** tab check if the database and gathering services are correctly installed and running.



If a red light is glowing next to an item, it indicates a failure of that component. You must address the stated problem before you can be sure that the Server is functioning correctly. Similarly, there is a tab to diagnose the **Web Reports** server.



4. To find out more information, click the question mark button to the right of any item. These buttons link to knowledge-base articles at the BigFix Support Site.
5. If all the buttons are glowing green, click **Close** to exit the Diagnostic.

Note: If the Server computer is a member of a domain, but you are logged in as a local user, the Diagnostics Tool will sometimes erroneously report that permissions

are incorrect. If you see that your permissions tests are incorrectly failing, you can safely ignore the diagnostics warnings.

Installing the Client on Windows

For more details about how to install the Clients, see section Installing the clients (on page 283).

Installing the Console

The BigFix Console lets the operator monitor and fix problems on all managed computers across the network.

It can be installed on any computer that can make a network connection via HTTPS port 52311 to the server. Except in testing or evaluation environments, do not run the console on the server computer itself due to the performance and security implications of having the publisher key credentials on a computer that is running a database or web server.

To install the **BigFix Console on a Windows BigFix Server machine**, follow these steps:

1. Run the Installation Guide (**Start > Programs > BigFix > BigFix Installation Guide**).
2. On the BigFix Installation Guide panel, click **Install or Upgrade Console**.
3. When prompted, enter the installation location for the Console. The default location is `C:\Program Files (x86)\BigFix Enterprise\BES Installers\Console`. To choose another destination, click **Browse** and navigate to the desired location. Click **Next** to continue.
4. After the files are installed, click **Finish** to complete the installation. You can now choose to launch the Console, or continue to the next section to install the Clients.

To install the **BigFix Console on a Windows machine with the BigFix Client**, use the Fixlet "Install BigFix Console (Version X)" available in the BES Support site. For example, to install the BigFix Console Version 11 GA, use the Fixlet named "Install BigFix Console (Version 11.0.0)".

To install the **BigFix Console on a Windows machine without the BigFix Client**, follow these steps:

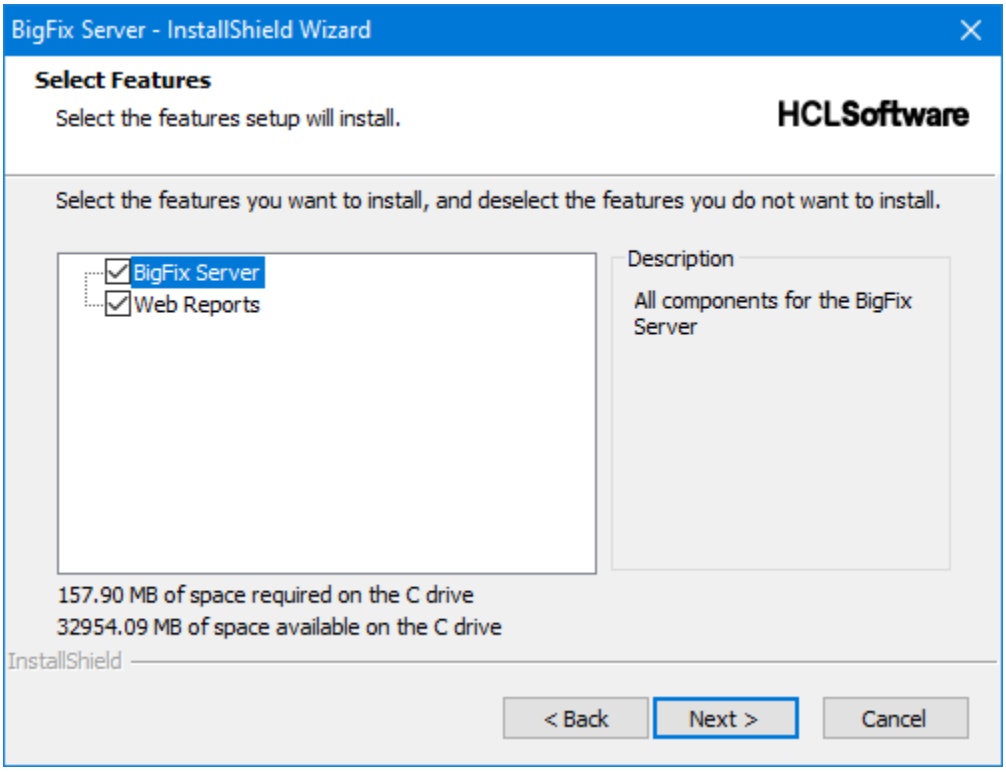
1. From the Windows BigFix Server machine `C:\Program Files (x86)\BigFix Enterprise\BES Installers` directory, copy the "Console" folder on the machine where you intend to install the Console.
2. Replace the masthead.afxm file in the "Console" folder with an updated version. The updated version can be taken from the BigFix Server directory `C:\Program Files (x86)\BigFix Enterprise\BES Server` and it is named `ActionSite.afxm`, or it can be downloaded from the `http://servername:port/masthead/masthead.afxm` site.
3. Run the setup.exe, located in the "Console" folder, to start the installation of the BigFix Console.

For more details about using the BigFix Console, see the BigFix Console Users Guide.

Installing Web Reports Standalone

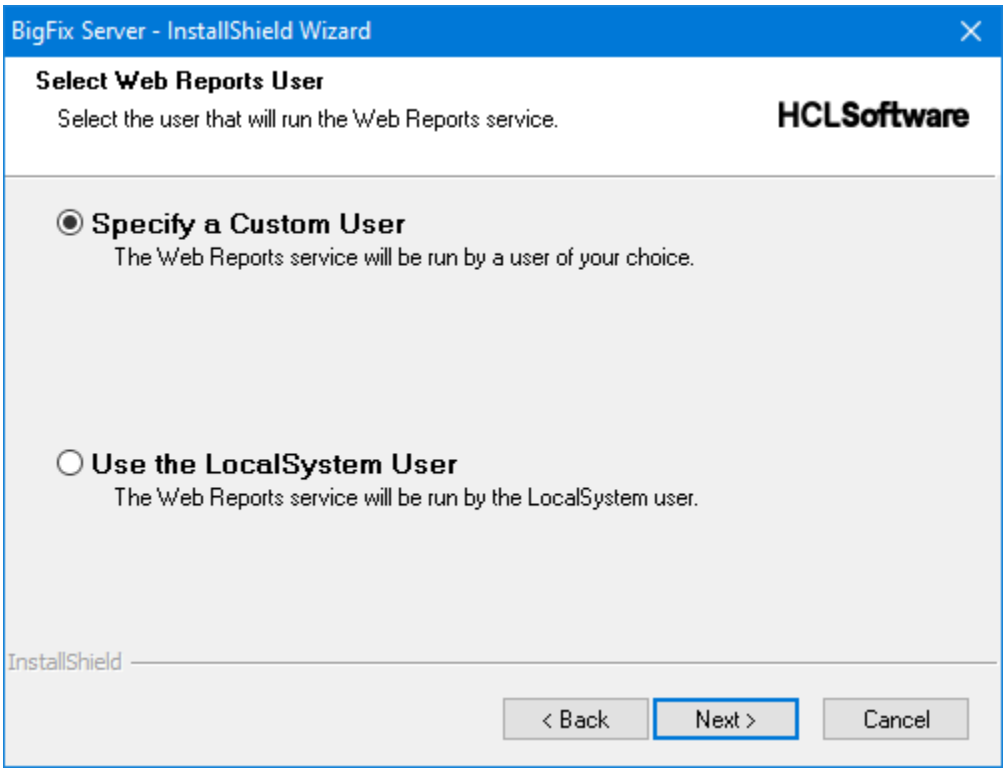
By default, the Web Reports component is installed along with the BigFix server.

However, you can choose not to install this component by deselecting the related check box in the following dialog:

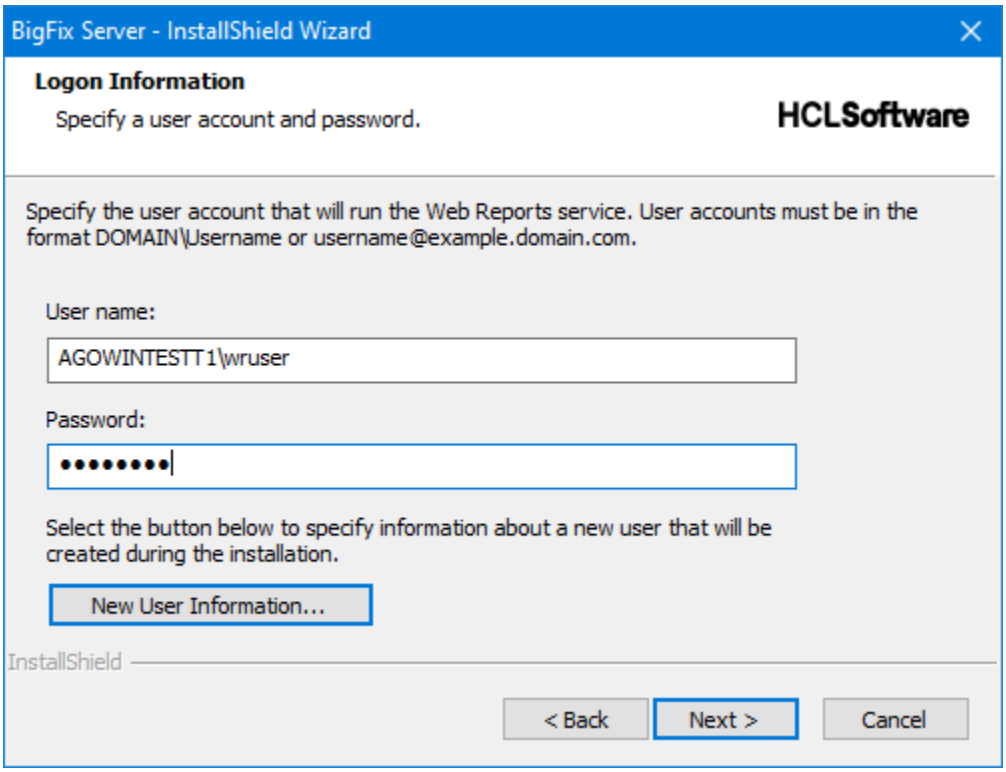


You can install it later only on a different system by running the following steps:

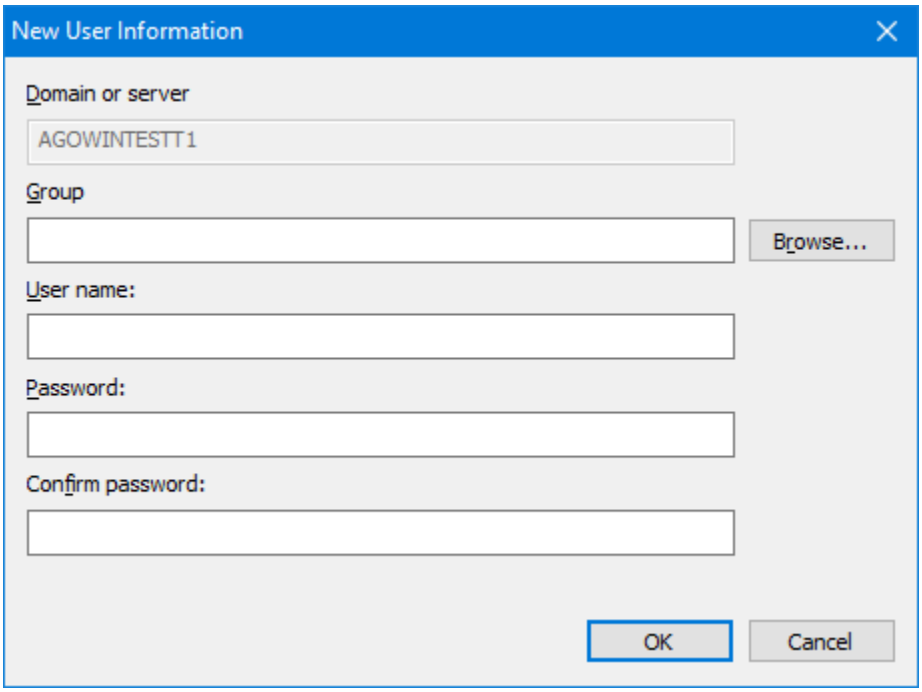
1. Copy the `BigFix Enterprise\BESInstallers\Server` directory from your BigFix server to the system where you want to install the stand-alone Web Reports server.
2. Replace the masthead.afxm file in the "Server" folder with an updated version. The updated version can be taken from the BigFix Server directory `C:\Program Files (x86)\BigFix Enterprise\BES Server` and it is named `ActionSite.afxm`, or it can be downloaded from the `http://servername:port/masthead/masthead.afxm` site.
3. From the Server directory, run the setup.exe program to start the server installer.
4. In the **Select Features** dialog, select to install only the Web Reports component.
5. Complete by choosing the database options that applies to your configuration. If you select **Use Remote Database**, complete the following configuration steps:
    a. On the **Database Server** window select the desired authentication method. If you choose Windows authentication, you must later on modify the Web Reports service logon to use a Windows authenticated user logon.
    b. On the **Select Features** dialog, ensure that only the **Web Reports** check box is selected.
    c. Choose the appropriate **Destination Location**.
    d. Choose where the Web Reports server will have its root directory and click **Next**.
6. Starting from V9.5.3, you can specify the type of user that will run the Web Reports services in this dialog:


BigFix Server - InstallShield Wizard
Select Web Reports User
Select the user that will run the Web Reports service.
HCLSoftware
Specify a Custom User
The Web Reports service will be run by a user of your choice.
Use the LocalSystem User
The Web Reports service will be run by the LocalSystem user.
InstallShield
< Back
Next >
Cancel

a. If you choose to specify the LocalSystem User, you continue with step 7 and review the installation parameters. If you choose to specify a custom user, you see the following dialog:



where you can specify an existing local user. This user does not need to have any particular rights or to belong to a group with particular rights to be used as a Web Reports installation user. Enter manually the user name and password or click the Browse button to select the user. If you want to create a new user, click **New User Information**, and enter the required information in this dialog:



Note: If you are upgrading from earlier versions, the Web Reports service user remains the same as before the upgrade.

7. Review the installation parameters and click **Next** to trigger the installation.
8. Specify the database login for the server components and authentication method and click **Next**.
9. Configure a data source to specify the BigFix server from where the data are collected. To create or modify a data source, see the procedure in Datasource Settings.

Note: If you run a fresh installation of Web Reports V9.5.2 or later, the HTTPS configuration is automatically enabled on port 8083. After the installation completes successfully, you can switch to the HTTP configuration by changing the value of the `_WebReports_HTTPServer_UseSSLFlag` setting to 0. For more information, see Customizing HTTPS on Web Reports.

Installing the WebUI Standalone

Use this procedure to install the WebUI on BigFix Version 11.

Prerequisites: Before running this procedure, ensure that you installed the BigFix client.

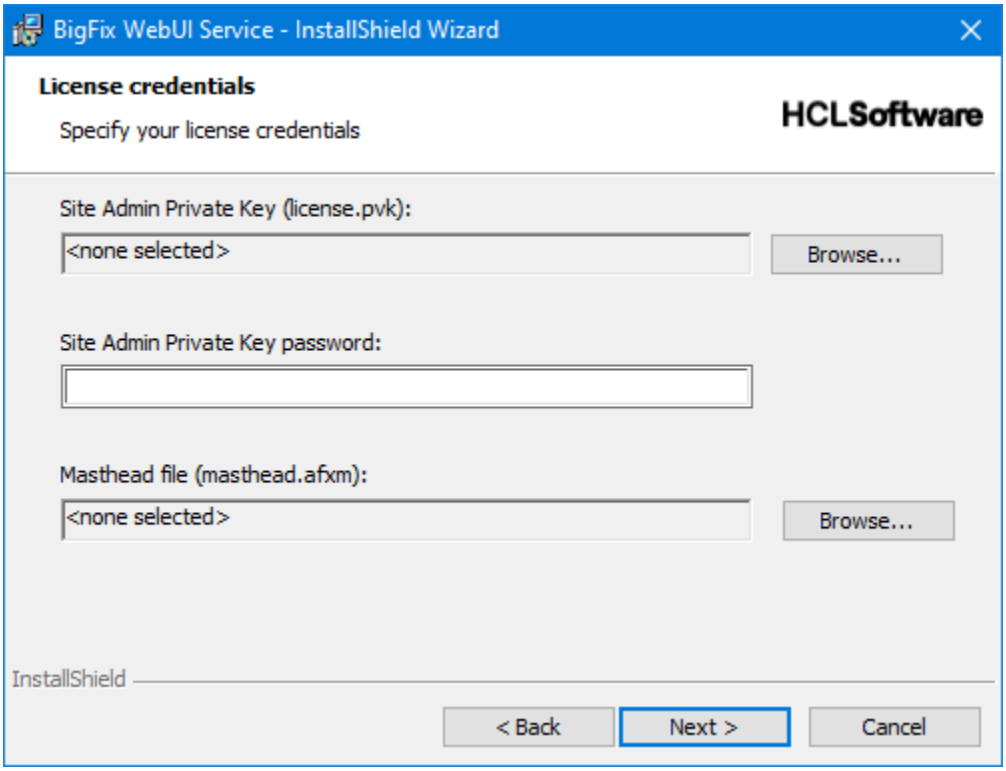
Note: You can install the WebUI component either locally on your BigFix server or on a different client computer of your environment. Only one WebUI installation is supported in your BigFix environment.

Note: To install the WebUI component on a different computer, you can use the Fixlet named **Install BigFix WebUI Service (Version 11)**. For more details, see Installation Procedure.

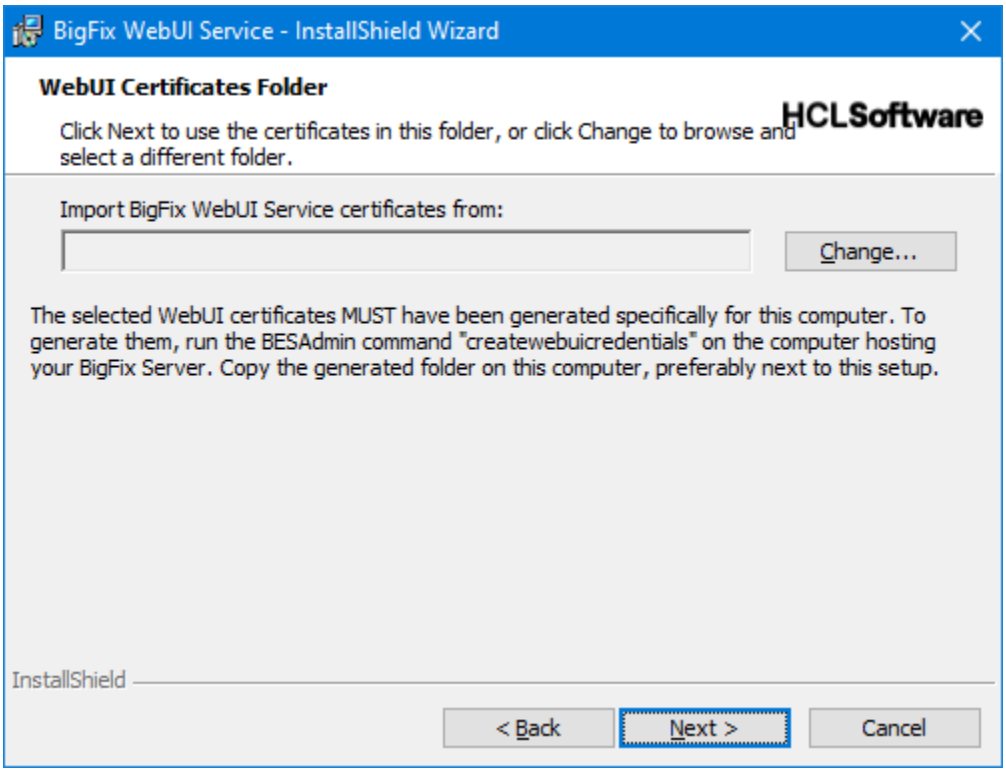
To manually install the BigFix WebUI component on BigFix Version 11, perform the following steps:

1. If you are installing the WebUI component on a remote client computer of your environment and not on your BigFix server, ensure that you copy from the server to the remote client computer:
   - The `setup.exe` file located in the `BigFix Enterprise\BES Installers\WebUI\` server directory.
   - The certificates that you have generated for the client computer where you will install the WebUI by running the `BESAdmin.exe /createwebuicredentials` command on the server. For more details about the command, see BESAdmin Windows Command Line *(on page 376)*.
2. Run the `setup.exe` file located in the `BigFix Enterprise\BES Installers\WebUI\` directory to launch the InstallShield Wizard for the BigFix WebUI.
3. If you are installing the WebUI component on the same computer where you installed your BigFix server (local WebUI), you will see the following dialog. Otherwise, if you are installing a remote WebUI, skip this step and move to step 4.

   Provide the location of the license.pvk file and the corresponding password. Specify the location of your masthead file. Click Next, skip the next step and move to step 5.



4. If you are installing the WebUI component on a computer where the BigFix server is not installed (remote WebUI), you will see the WebUI Certificates Folder dialog.

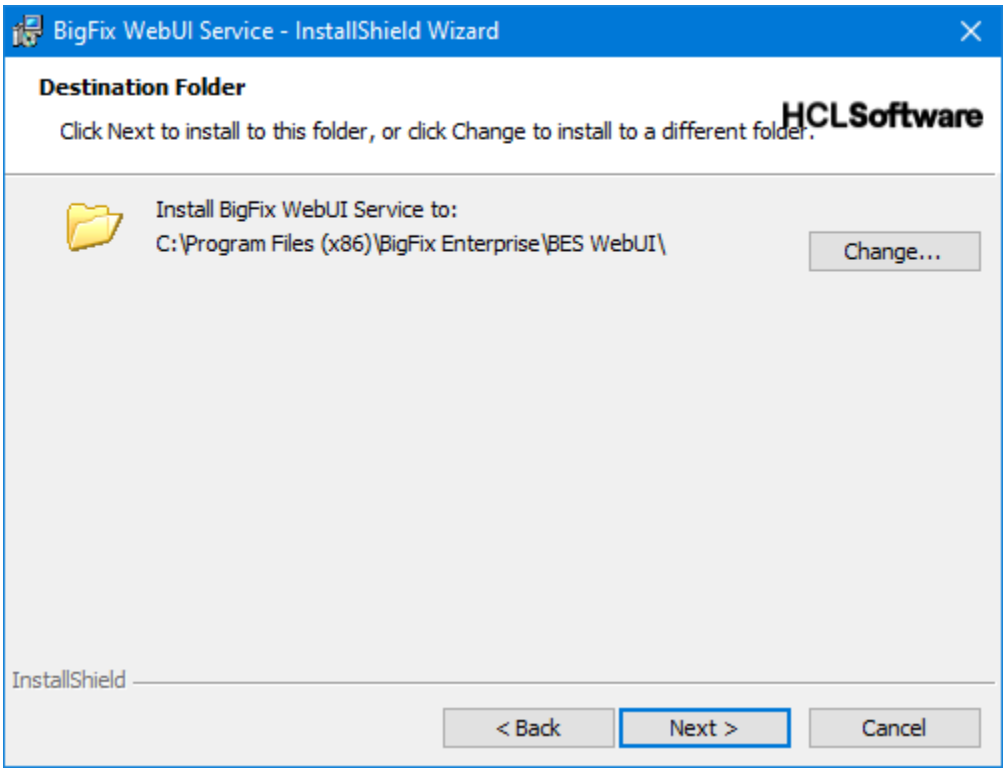


If you have a folder named **cert_*hostname*** in the same directory where the `setup.exe` file of the WebUI component is located, this folder will be the default directory used by the installer. Otherwise, browse and select the folder containing the following files:

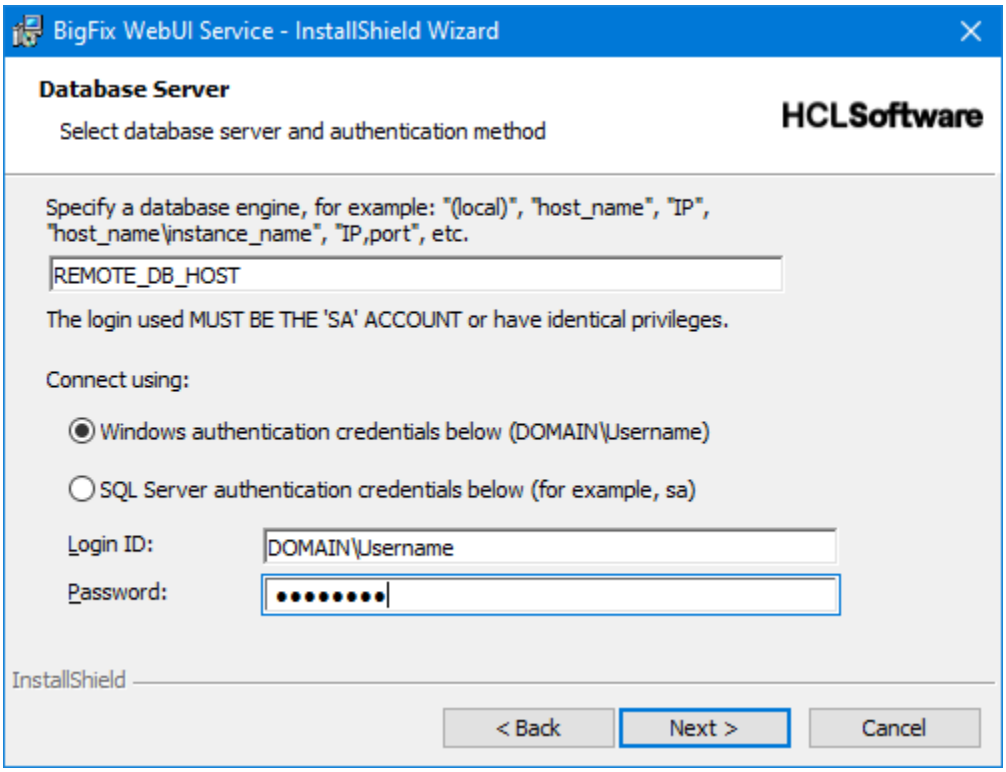
- ca.crt
- auth_cert.crt
- auth_key.key

Note: The certificates must have been generated for the computer where you are installing the WebUI. To generate the certificates, run the `BESAdmin.exe /createwebuicredentials` command. For more details about the command, see BESAdmin Windows Command Line *(on page 376)*.

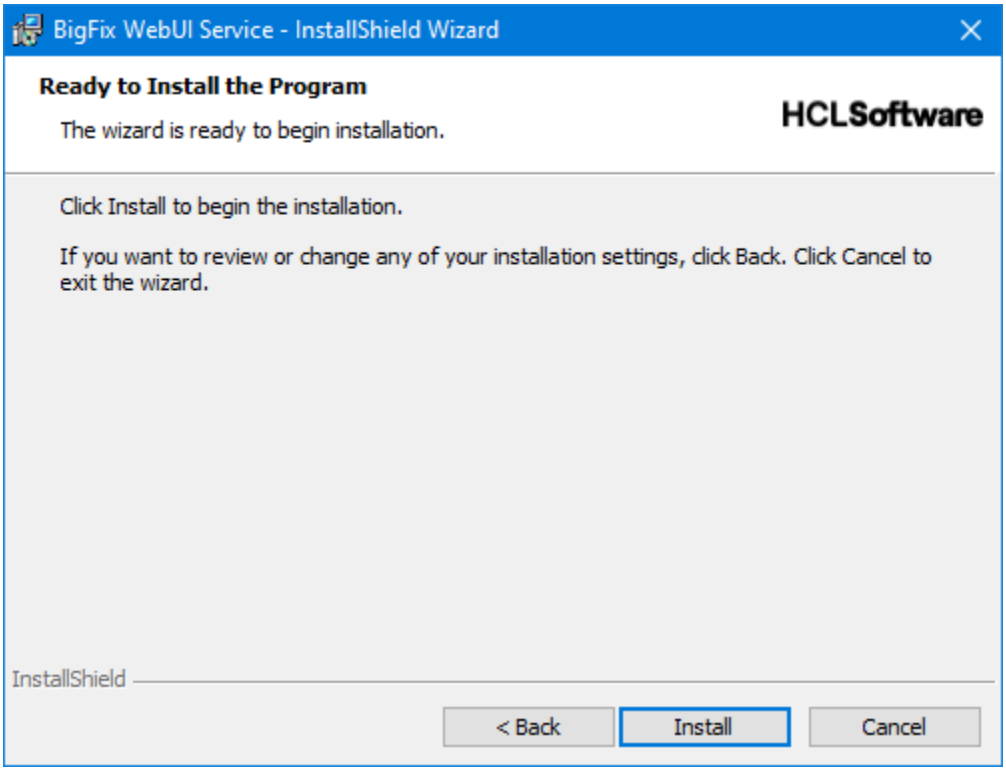
5. Browse and select the WebUI installation folder, if different from the default folder.



6. Specify the BigFix database server and the related authentication method. You can connect to the database by using either the Windows authentication or the SQL Server authentication. For the WebUI to function properly, you must specify a user with sysadmin privileges.



7. Click **Install** to proceed with the BigFix WebUI installation.



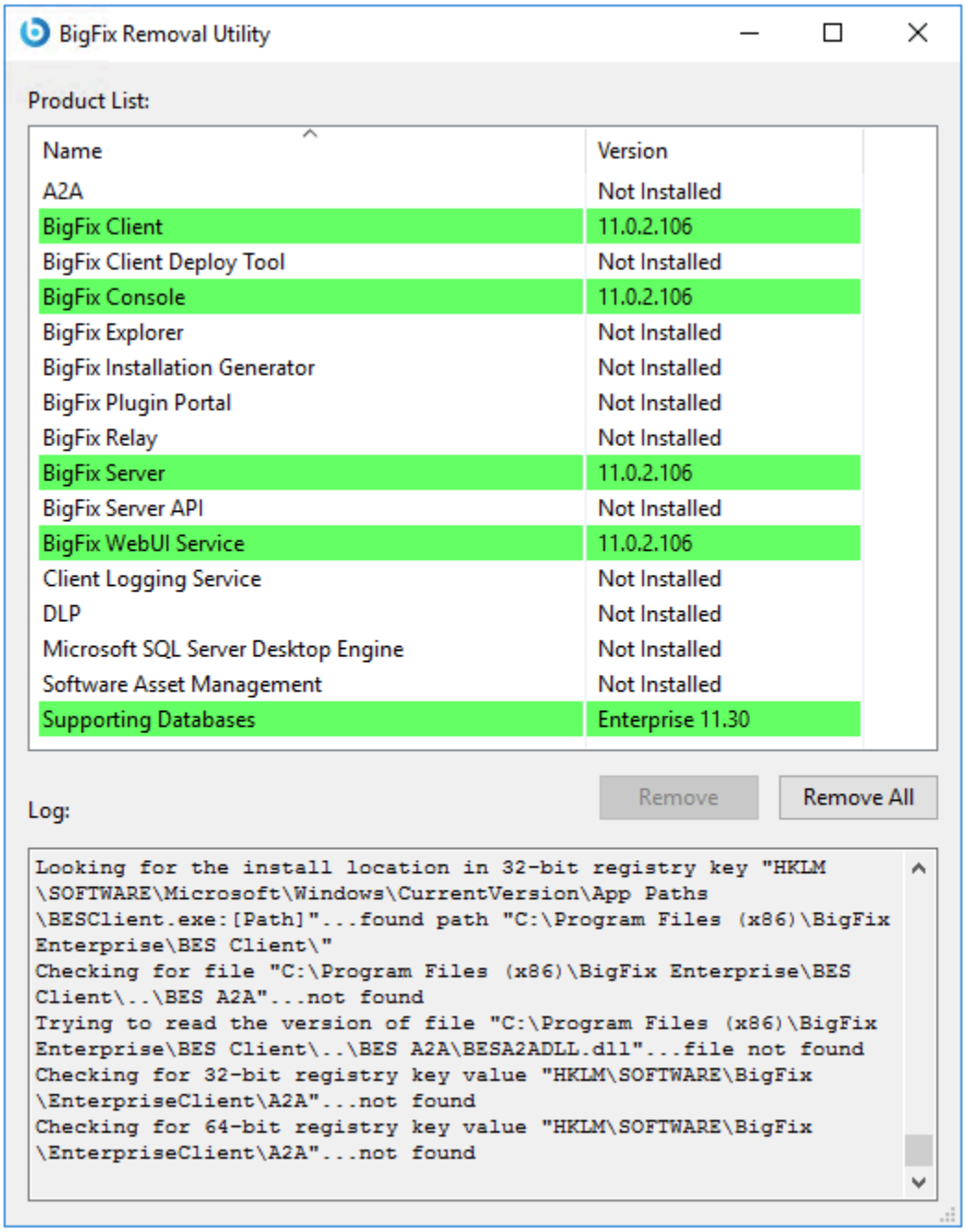
8. The installation process might take several minutes. When the installation completes, click **Finish** to exit the wizard.

Removing the BigFix components from Windows

If you have one or more BigFix components installed on a computer, you can remove all of them at once, or one at a time.

To uninstall one or more BigFix components installed on a local Windows system, run the following steps:

1. Go to the BigFix Download Center Platform Release page.
2. Open the page of the latest release and download the **BESRemove.exe** utility (listed under the **Utilities** section). The latest version of the tool can remove any previous version.
3. Double-click **BESRemove.exe** to run the utility.



4. Select the components that you want to uninstall and then click **Remove**, or click **Remove All** to remove from the system all the BigFix components installed.
5. If BESRemove did not remove your databases, delete them manually.

Note: This step only applies if you are using an on-premise SQL Server (local or remote). In case your databases are on Cloud, follow the specific Cloud provider guidelines to remove your databases.

If your BigFix databases are remote, BESRemove cannot remove them. There could be other reasons why BESRemove may not be able to detect and remove your BigFix databases. In those cases, you can delete them manually. For example, you can use SQL Server Management Studio to connect to the instance hosting your BigFix databases and run the following commands:

```
sqlcmd -U sa -S localhost,1433
USE master;
ALTER DATABASE BFEnterprise SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
DROP DATABASE BFEnterprise;
GO
ALTER DATABASE BESReporting SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
DROP DATABASE BESReporting;
GO
```

DSA on Windows

Installing Additional Windows Servers (DSA)

Before proceeding with this section, determine your authentication method and complete the appropriate steps in .

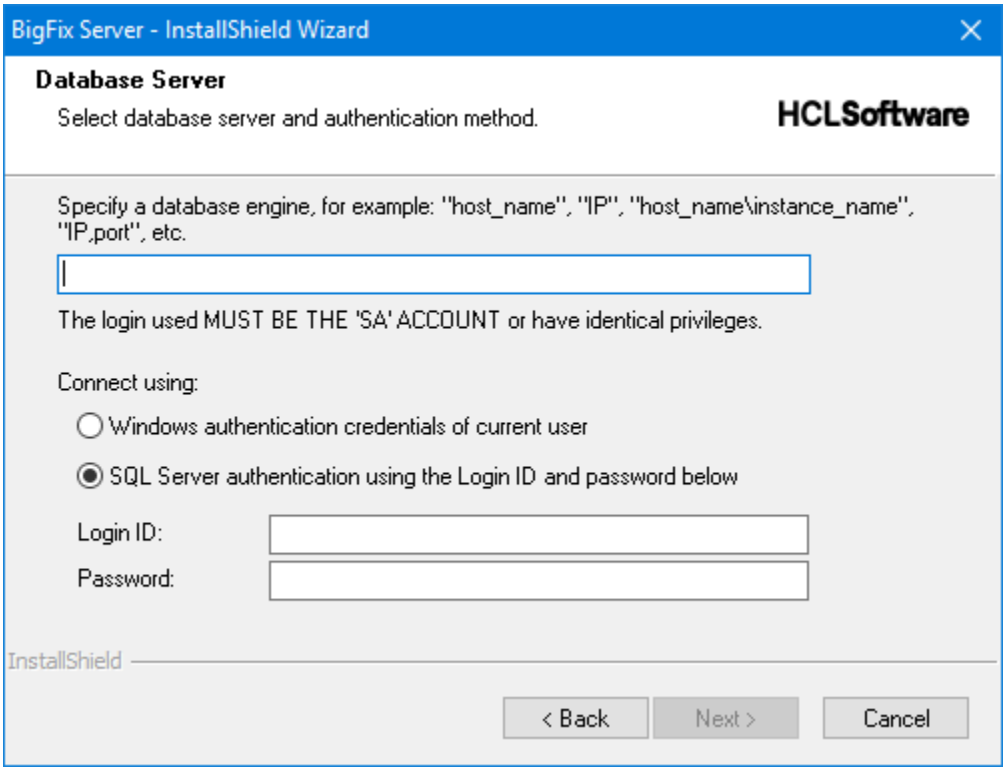
For each additional server that you want to add to your deployment, make sure it can communicate with the other servers, and then follow these steps:

1. Download the BigFix Server installer having the same version as the one installed on the master server or copy it from the `BigFix Enterprise\BESInstallers\Server` directory of your BigFix Server to the system where you want to install the additional server.
2. Replace the masthead.afxm file in the "Server" folder with an updated version. The updated version can be taken from the BigFix Server directory `C:\Program Files (x86)\BigFix Enterprise\BES Server` and it is named `ActionSite.afxm`, or it can be downloaded from the `http://servername:port/masthead/masthead.afxm` site.
3. Copy the `license.pvk` file from the master server to each computer where you intend to install an additional DSA Server.
4. Each DSA Server must have **its own** SQL Server database engine, either local or remote. Ensure that each server uses the same SQL Server version. Do not use the same database engine to store the databases of two different DSA servers. Each DSA Server must be able to access its own database engine and also the database engines of the other DSA Servers.
5. Use the same authentication method to access all the SQL Server database engines, either *Windows authentication* or *SQL Server authentication*. If you chose the Windows authentication method, use the same domain user to access all your database engines. If you chose the SQL Server authentication method, use the same user name and password. Ensure that your database user has sysadmin privileges on all database engines.
6. If you are extracting the server installer from the Installation Generator, select **Production Deployment**, and **I want to install with an existing masthead**. Specify the `masthead.afxm` file from the master server.
7. On the dialog of the server installer, choose a combination of components that includes the BigFix Server. Do not install the WebUI component on the secondary DSA servers.
8. On the **Select Database Replication** dialog of the server installer, select **Replicated Database**.

The following steps of the installation procedure differ depending on the BigFix installer version.

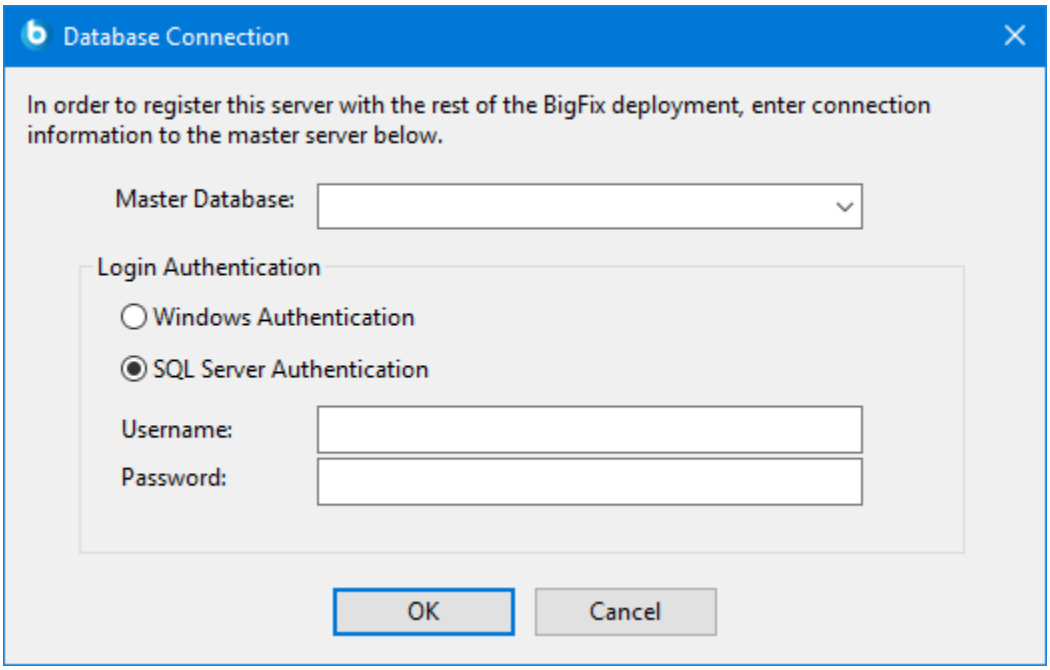
Database configuration steps for 11.0.5 and earlier

1. In the **Select Database** dialog, you can choose the **Local Database** option to host the secondary server's own database locally. When choosing this option, the LocalSystem user will be used to access the local database through Windows authentication. However, this user only exists locally, so you will have to use SQL Server authentication to access the primary server's database. Alternatively, you can choose "Remote Database" and specify the local computer as the host, or an actual remote computer.
2. If you chose **Local Database**, skip this step. In the **Database Server** dialog, enter the hostname or IP address of the computer that will host your secondary server's database, and the credentials of an account with sysadmin permissions on the database instance. Consider that, if you choose Windows authentication, you will have to use the same user to access the primary server's database.



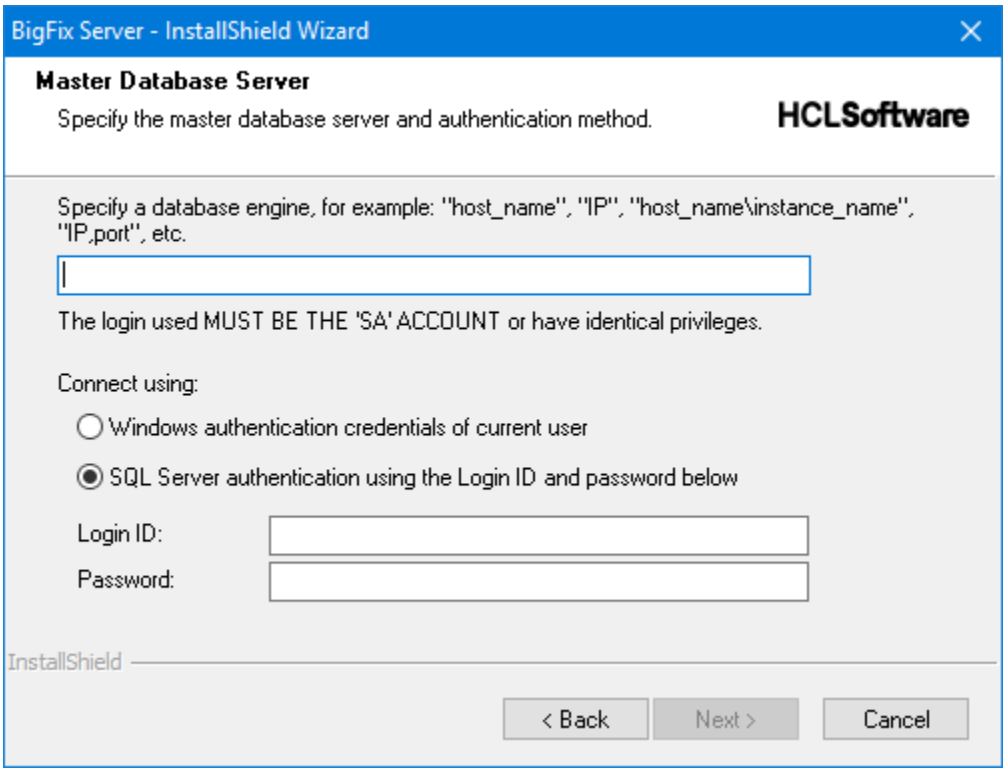
3. Proceed through the dialogs until the installation operations begin. At one point during the setup process, the **Database Connection** dialog will pop up. Enter the hostname or IP address of the computer hosting the database of your primary server. If you chose

the **Local Database** option for the secondary server's database, you must use SQL Server authentication to access the primary server's database. If you chose to access the secondary server's database using Windows authentication, you will have to use the same user to access the primary server's database.

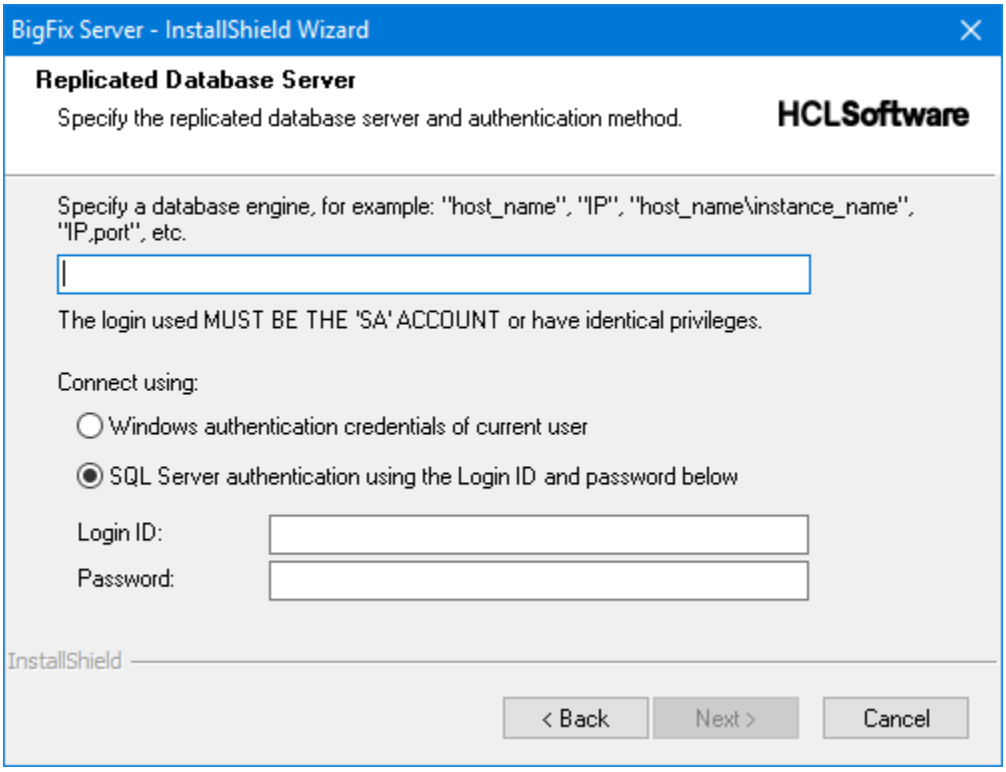


Database configuration steps for 11.0.6 and newer

1. In the **Master Database Server** dialog, enter the hostname or IP address of the computer hosting the database of your primary server's database, and the credentials of an account with db_owner permissions on the BFEnterprise database. If you choose Windows authentication, you will have to use the same user to access the secondary server's database.


BigFix Server - InstallShield Wizard
Master Database Server
Specify the master database server and authentication method.
HCLSoftware
Specify a database engine, for example: "host_name", "IP", "host_name\instance_name", "IP,port", etc.
The login used MUST BE THE 'SA' ACCOUNT or have identical privileges.
Connect using:
Windows authentication credentials of current user
SQL Server authentication using the Login ID and password below
Login ID:
Password:
InstallShield
< Back
Next >
Cancel

2. In the **Replicated Database Server** dialog, enter the hostname or IP address of the computer that will host your secondary server's database, and the credentials of an account with sysadmin permissions on the database instance.



Last installation steps

1. The BigFix Administration Tool displays a pop-up dialog containing an error message in red, which describes a failed connection to the database that the server is trying to replicate from. Ignore this error message and click **OK**.
2. If you select the check box **Run the BigFix Diagnostic Tool** on the last installation dialog, after you click **Finish**, the BigFix Server Diagnostics displays a pop-up dialog containing several test failures. Ignore this dialog and click **Close**.

Post installation steps

1. Depending on the authentication method used to access the SQL Server database engine:
   - If you are using Windows Authentication, ensure that the user running the FillDB service on all DSA servers of your environment is the same Windows domain user. This user must have access to all database engines used by your DSA servers. If needed, change the Log On settings of the FillDB service on all your DSA servers accordingly and then restart the service.
   - If you are using SQL Authentication, stop the FillDB service on all your DSA servers, open the registry key `HKLM\Software\Wow6432Node\BigFix\Enterprise Server\FillDB` and add the following string values to all your DSA servers (if they are missing):

     ```
     ReplicationDatabase = BFEnterprise
     ReplicationUser = <login name>
     ReplicationPassword = <password>
     ```

     Then, restart the FillDB service. **Note:** For version 11.0.6 and newer, those keys are automatically created on the secondary server, but you still need to manually create them on the primary.
2. After installing a new DSA server in an environment where the custom CA is already present, you must also install the custom CA (using the `setcustomca` BESAdmin command) there for the new server to work.
3. On the master server, run the resign security data command by using the BigFix Administration tool.

   ```
   .\BESAdmin.exe /resignsecuritydata
   ```

   For additional information on the command, see BESAdmin Windows Command Line *(on page 376)*.
4. On the newly-installed server, run the BigFix Administration Tool and select the **Replication** tab to see the current list of servers and their replication status. You might need to wait for the next replication period before all servers show a successful connection. Select the newly-installed server from the drop-down menu, and verify

in the table that it has successfully connected to the primary server. Then select the primary server from the same menu, and verify that it is connected to the new server.

Note: The initial replication can take several minutes to hours, depending on the size of your database. Wait for the replication to complete before taking any actions from a console connected to the secondary DSA server. Moreover, the replication process might get interrupted. If you experience this problem, you can discuss it with your HCL Software Support.

5. Verify that all servers have successfully completed their replication.
6. The replication server window shows you the server configuration for your current deployment. By default, your newly-installed server is configured to replicate directly from the primary server's database every 5 minutes. This time interval can be changed to a larger value.

Authenticating Additional Servers

Multiple servers can provide a higher level of service for your BigFix installation.

If you choose to add Disaster Server Architecture (DSA) to your installation, you will be able to recover from network and systems failures automatically while continuing to provide local service. To take advantage of this function, you must have one or more additional servers with a capability at least equal to your primary server. Because of the extra expense and installation involved, you should carefully think through your needs before committing to DSA.

You must first decide how you want your servers to communicate with each other. There are three inter-server authentication options: the first two are flavors of NT and the third is SQL. Because it is more secure, NT Authentication is recommended. You cannot mix and match; all servers must use the same authorization.

Using NT Authentication with domain users and user groups

With this method, each server uses the specified domain user or a member of the specified user group to access all the other servers in the deployment.

To authenticate your servers using domain users and user groups, follow these steps:

1. Create a service account user or user group in your domain. For a user group, add authorized domain users to your servers. You might need to have domain administration privileges to do this.
2. On the Master Server, use SQL Server Management Studio to create a login for the domain service account user or user group, with a default database of **BFEnterprise**, and give this login System Admin (sa) authority or the DBO (DataBase Owner) role on the BFEnterprise and master databases.
3. On the Master Server, change the **LogOn** settings for the FillDB, BES Root, and Web Reports services to the domain user or member of the user group created in step 2, and restart the services.

Note: After you complete the installation of the BigFix server and begin to use Product sites, you might install additional components such as the **BES Server Plugin Service** and **BES NMAP Unmanaged Asset Importer**. Both these services have their **LogOn** settings set for the NT user for Remote Database access.

Using NT Authentication with domain computer groups

With this method, each server is added to a specified domain computer group and each server accepts logins from members of that domain group.

To authenticate your servers using domain computer groups, follow these steps:

1. Create a Global Security Group in your domain containing your chosen servers. You might need to have domain administration privileges to do this.
2. After creating the group, each server must be rebooted to update its domain credentials.
3. On the Master Server, use SQL Server Management Studio to create a login for the domain group, with a default database of BFEnterprise, and give this login System Admin (sa) authority or the DBO (DataBase Owner) role on the BFEnterprise and master databases.

Using SQL Authentication

With this method, each server is given a login name and password, and is configured to accept the login names and passwords of all other servers in the deployment.

The password for this account typed in clear text is obfuscated under the `HKLM` branch of the registry on each server, after the restart of the FillDB service.

To authenticate your servers using SQL authentication, follow these steps:

1. Choose a single login name (for example, `besserverlogin`), and a single password to be used by all servers in your deployment for inter-server authentication.
2. On the Master server, use SQL Server Management Studio to create a SQL Server login with this name. Choose SQL Server Authentication as the authentication option and specify the password. Change the default database to `BFEnterprise` and assign the sysadmin server role to the new user, or map it to the role of `db_owner` on the `BFEnterprise` and master databases.
3. On the master server, add the following string values under the `HKLM\Software\Wow6432Node\BigFix\Enterprise Server\FillDB` key:

   ```
   ReplicationUser = <login name>
   ReplicationPassword = <password>
   ReplicationPort = <SQL_port>
   ```

4. Restart the `FillDB` service.

Note:

This choice must be made on a deployment-wide basis; you cannot mix domain-authenticated servers with SQL-authenticated servers.
`ReplicationUser`, `ReplicationPassword`, and `ReplicationPort` must be uniquely defined in all the server registries of your DSA environment.
All BigFix servers in your deployment must be running the same version of SQL server.

Uninstalling a Windows replication server

To uninstall a replication server, call the database-stored procedure **delete_replication_server**, which removes the specified ID from the replication set. Be careful not to delete the wrong server, or you might lock yourself out.

The details of this procedure are beyond the scope of this guide, but basically you must log in to the database with SQL Server Management Studio. You can call the procedure with something like:

```
exec BFEnterprise.dbo.delete_replication_server n
```

where *n* is the identifier of the server to delete.

The steps involved in completely deleting the server are beyond the scope of this guide, but the full procedure is available in How to remove a secondary DSA server from BigFix Administration Tool.

Chapter 9. Installing on Linux systems

After understanding the terms and the administrative roles, you are ready to actually get authorized and install the programs.

Because BigFix is powerful, you might want to limit access to trusted, authorized personnel only. The program depends on a central repository of Fixlet actions called the **Action site**, which uses public/private key encryption to protect against spoofing and other unauthorized usage. To get started, you need authorization from HCL by getting a **License Authorization** file, which will have a name like `CompanyName.BESLicenseAuthorization`.

Note: The root privileges are required to perform the installation of the server components. The 'sudo' utility cannot be used.

The installation program collects further information about your deployment and then creates a file called the **action site masthead**. This file establishes a chain of authority from the BigFix root all the way down to the Console operators in your organization. The masthead combines configuration information (IP addresses, ports, and so on) and license information (how many Clients are authorized and for how long) together with a public key used to verify the digital signatures.

Installing and configuring DB2

BigFix 11 Server installation on Linux has IBM DB2 as a prerequisite. It must be installed before the BigFix Server is installed.

While earlier versions of certain BigFix products included DB2 entitlement and installer, this is no longer the case with BigFix 11. Because of this, if you want to install your BigFix 11 Server on Linux, and you do not already own a DB2 license, you must procure one with IBM.

DB2 versions supported by BigFix are available at Database requirements (on page 81).

DB2 can be installed on the local workstation where you want to install the BigFix Server or on a remote workstation.

For information about how to install and verify DB2 server installation on Red Hat Enterprise Linux server 64-bit, see *DB2 servers and IBM data server clients*. Before installing the BigFix Server ensure that the DB2 has been installed and started as follows:

- If the DB2 is installed locally:
  1. Ensure that the DB2 instance is up and running and that the DB2 administrative server is started. If you configured DB2 to use the default user names, run the following commands:

     ```
     su - db2inst1
     db2start
     exit
     ```

  2. You can also verify that the DB2 instance is running by checking that the `db2sysc` process is active using the following command:

     ```
     ps -ef | grep db2sysc
     ```

- If the DB2 is installed remotely:
  1. Install a DB2 client on the workstation from where you run the BigFix server installation. The actual product you need to install is IBM Data Server Client 11.5 and its product identifier is "db2client". The port of the remote DB2 database must be reachable by the workstation where the installation is running. No additional DB2 configurations (such as the catalog of the remote database) are required.

     Note:
     To use a remote database for BigFix, ensure you provide the following information in the installation procedure:
     a. The remote DB2 node
     b. The DB2 port number
     c. The user name of the local DB2 instance owner for the remote DB2 client and the remote DB2 server.

Important: The DB2 instance names used to install the BES Root Server cannot contain the following special characters:

blanks, tabs (`\t`), returns (`\n`) and `; & | " ' < >`.

For the DB2 password rules, see Database requirements (on page 81).

To install the DB2 client you can run the installation wizard or the silent installation with a response file. For additional details, see Installation methods for IBM data server clients.

2. On the remote DB2, ensure that the DB2 instance is up and running and that the DB2 administrative server is started. If you configured DB2 to use the default user names, run the following commands:

```
su - db2inst1
db2start
exit
```

For information about database requirements, see Installation requirements for DB2 database products and Database requirements (on page 81).

Installing and configuring MS SQL

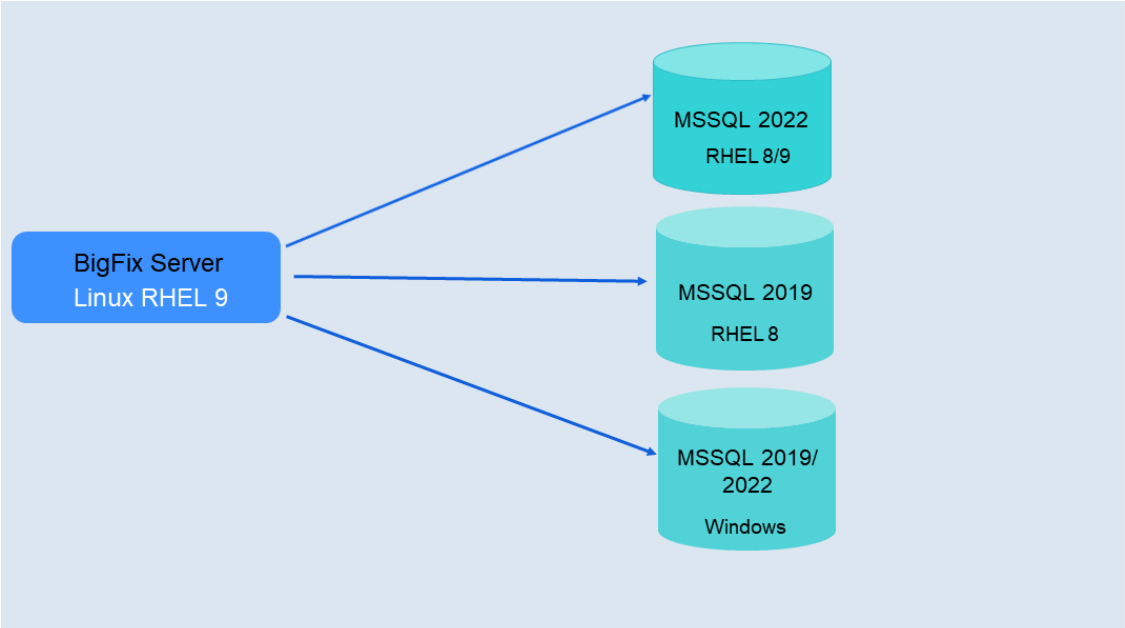
Starting from BigFix Version 11 Patch 1, the Linux Server installation introduces the support of Red Hat 9 x86 64-bit with Microsoft SQL Server database.

Starting from BigFix Version 11 Patch 2, the Linux Server installation introduces the support of Red Hat 9 x86 64-bit with Google Cloud SQL for Microsoft SQL Server 2022 database.

Starting from BigFix Version 11 Patch 4, the Linux Server installation introduces the support of Red Hat 9 x86 64-bit with:

- Azure SQL Managed Instance for Microsoft SQL Server 2022 database
- Amazon RDS for Microsoft SQL Server 2022 database.

BigFix Platform Version 11 Patch 1 supports the configurations shown below:



- Microsoft SQL Server 2022 installed on Red Hat Enterprise Linux (RHEL) 8 and 9
- Microsoft SQL Server 2019 installed on Red Hat Enterprise Linux (RHEL) 8
- Microsoft SQL Server 2019 and 2022 installed on Windows.
- Google Cloud SQL for Microsoft SQL Server 2022 database on Windows at Installing and configuring DB on Cloud *(on page 115)* and on Linux at Installing and configuring DB on Cloud *(on page 194)*.
- Azure SQL Managed Instance for Microsoft SQL Server 2022 database on Windows at Installing and configuring DB on Cloud *(on page 115)* and on Linux at Installing and configuring DB on Cloud *(on page 194)*.
- Amazon RDS for Microsoft SQL Server 2022 database on Windows at Installing and configuring DB on Cloud *(on page 115)* and on Linux at Installing and configuring DB on Cloud *(on page 194)*.

Note that the Enterprise and Standard editions are recommended.

Note: Only the SQL Authentication is supported on the BigFix Linux Server with the Microsoft SQL database.

Note: DSA configurations are not supported on the BigFix Linux Server with the Microsoft SQL database.

Installing and configuring DB on Cloud

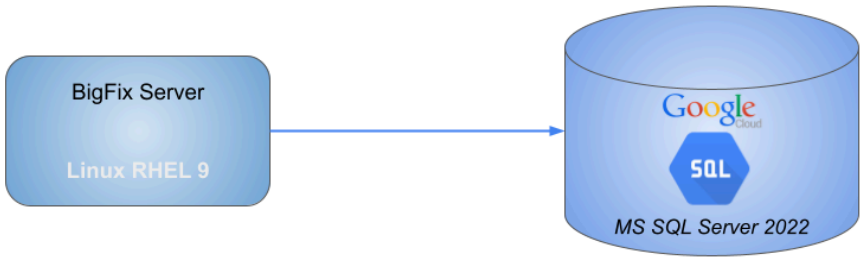
Starting from BigFix Version 11 Patch 2, the Linux Server installation introduces the support of Red Hat 9 x86 64-bit with Google Cloud SQL for Microsoft SQL Server 2022 database.

Starting from BigFix Version 11 Patch 4, the Linux Server installation introduces the support of Red Hat 9 x86 64-bit with:

- Azure SQL Managed Instance for Microsoft SQL Server 2022 database
- Amazon RDS for Microsoft SQL Server 2022 database.

Starting from BigFix Version 11 Patch 6, the Linux Server installation introduces the support of Red Hat 9 x86 64-bit with Azure SQL Managed Instance for Microsoft SQL Server 2025 and Google Cloud SQL for Microsoft SQL Server 2025 databases.

BigFix Platform Version 11 Patch 2 supports the configuration shown below:

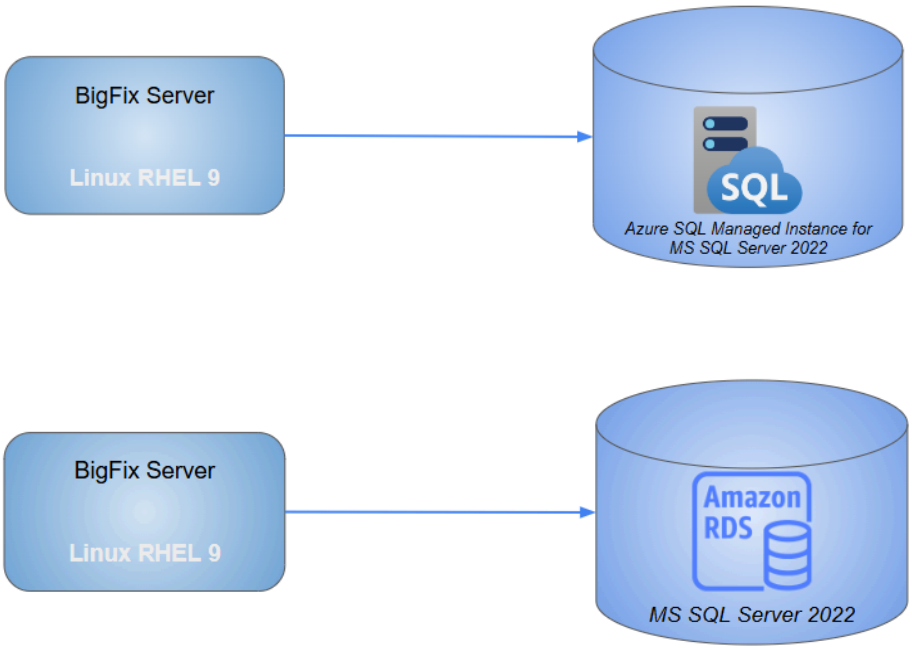


Note that the Enterprise and Standard editions are recommended.

Note: Only the SQL Authentication is supported on the BigFix Linux Server with Google Cloud SQL for Microsoft SQL Server 2022 database.

Note: DSA configurations are not supported with Google Cloud SQL for Microsoft SQL Server 2022 database.

BigFix Platform Version 11 Patch 4 supports the configurations shown below:



Note that the Enterprise and Standard editions are recommended.

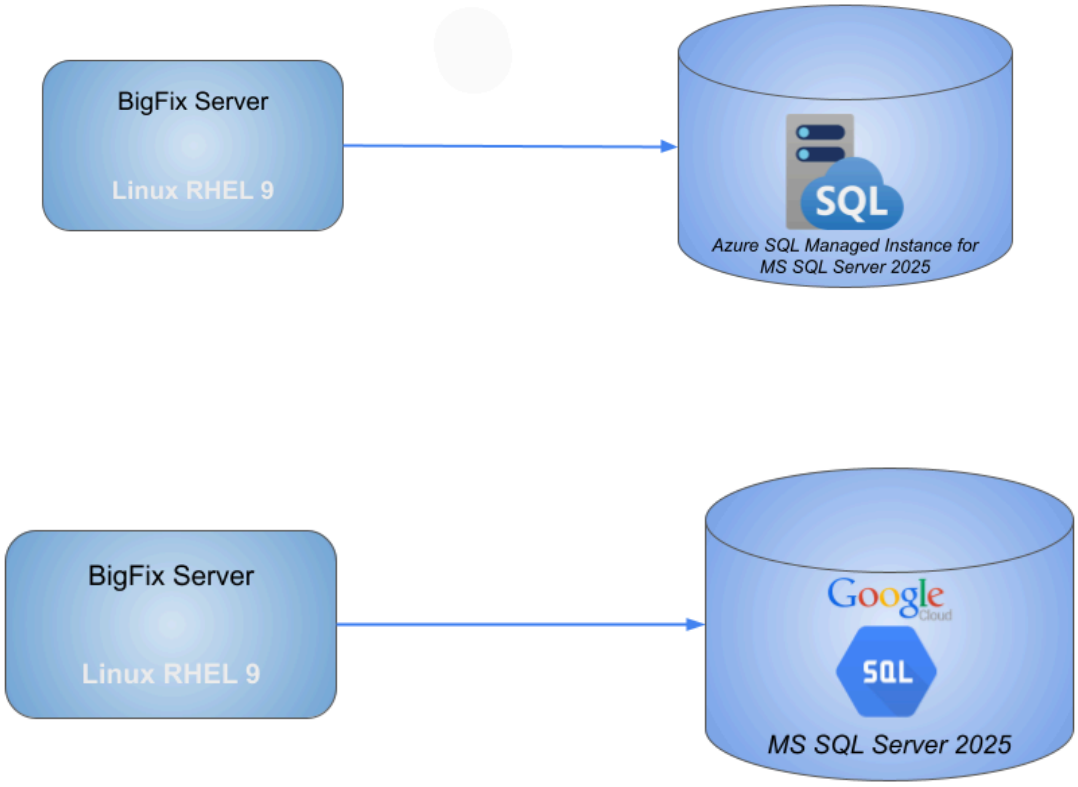
Note: Only the SQL Authentication is supported on the BigFix Linux Server with:

- Azure SQL Managed Instance for Microsoft SQL Server 2022 database
- Amazon RDS for Microsoft SQL Server 2022 database.

Note: DSA configurations are not supported with:

- Azure SQL Managed Instance for Microsoft SQL Server 2022 database
- Amazon RDS for Microsoft SQL Server 2022 database.

BigFix Platform Version 11 Patch 6 supports the configuration shown below:



Note that the Enterprise and Standard editions are recommended.

Note: Only the SQL Authentication is supported on the BigFix Linux Server with Azure SQL Managed Instance for Microsoft SQL Server 2025 and Google Cloud SQL for Microsoft SQL Server 2025 databases.

Note: DSA configurations are not supported with Azure SQL Managed Instance for Microsoft SQL Server 2025 and Google Cloud SQL for Microsoft SQL Server 2025 databases.

Downloading BigFix

Download BigFix from HCL License & Delivery Portal (Flexnet).

You can download BigFix also from the support site at http://support.bigfix.com/bes/install/downloadbes.html.

To install the server component, download one of the following files from HCL License & Delivery Portal:

Table 2. Parts required for installing BigFix Server Version 11 Patch 1

Software Name	Image
BigFix Platform Install V11.0.1 for Multiplatform Multilingual	`HCL_BigFix_v11.0.1.104_Win_Lnx_Install.zip`
BigFix Platform Install V11.0.1 with DB2	`ServerInstaller_11.0.1.104-db2-rhe7.x86_64.tgz`
BigFix Platform Install V11.0.1 with MS SQL	`ServerInstaller_11.0.1.104-mssql-rhe7.x86_64.tgz`

To extract the BigFix Linux Server installation files, perform the following steps:

1. Copy the compressed file on your Linux Server.
2. Extract the compressed file.

Performing an evaluation installation with DB2

To install a BigFix server with an evaluation license on Linux with DB2, perform the following steps:

1. On the computer where you want to install the BigFix server, from the shell where you extract the contents of the `.tgz` server installer, enter the following command:

   ```
   ./install.sh
   ```

2. To install the Evaluation version, enter `1`:

```
Select the type of installation
[1]  Evaluation: Request a free evaluation license from HCL
This license allows you to install a fully functional copy of HCL
 BigFix on up to 1000 clients,
for a period of 30 days.
[2]  Production: Install using a production license or an
 authorization for a production license.
Choose one of the options above or press <Enter> to accept the default
 value: [1]
```

3. After reading the License Agreement, enter `1` to accept it and continue.
4. Enter your First Name, Last Name, Email address and Organization's Name to create the digital signature.
5. Enter the name with which your server is registered to the DNS. It will be used by the BigFix clients to identify the BigFix server. It cannot be changed after a license is created.
6. Enter the server identification port to use for all communication by the BigFix components or press <Enter> to accept the default value which is 52311.
7. Enter the Web Reports server HTTPS port number or press <Enter> to accept the default value which is 8083.
8. Enter the WebUI HTTPS port number or press <Enter> to accept the default value which is 443.
9. Enter the WebUI HTTP redirect port number or press <Enter> to accept the default value which is 80.
10. To configure the firewall, enter `1`:

```
The firewall of the operating system is active on the local server.
To enable the communication using the specified ports you can:
[1]  Configure the firewall now
[2]  Configure the firewall later
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```

11. **With DB2 already installed**

    - Enter the name of the local DB2 instance that you want to use or press <Enter> to accept the default value which is `db2inst1`
    - Enter the user name of the local DB2 instance owner that you want to use or press <Enter> to accept the default value which is `db2inst1`

    Without DB2 already installed

    Note: The following installer option to install the database supports DB2 11.5 only.

    - To install DB2, press <Enter> to accept the default value which is `1`:

      ```
      The installer does not detect if DB2 is installed on the system.
      Specify which option corresponds to your installation:
      [1]  DB2 is not installed, install it.
      [2]  DB2 is installed, use the installed instance.
      [3]  Exit from the installation.
      Choose one of the options above or press <Enter> to accept the
       default value: [1]
      ```

      DB2 is installed with the following default settings:

      ```
      DB2 instance owner: db2inst1
      DB2 fenced user: db2fenc1
      DB2 communication port: 50000
      DB2 installation directory: /opt/ibm/db2/V11.5
      ```

    - If you want to use different values for these settings, specify them in the installation response file or CLI options. Otherwise, press <Enter> to accept the default value:

      ```
      [1]  Proceed to install DB2.
      ```

    - If one or more DB2 prerequisite packages were not installed, you can ignore the warning messages and proceed with the installation (not recommended)

or install the missing packages using the YUM repository, if configured (recommended)

- Enter the location of the DB2 setup file or press <Enter> to accept the default value which is `../server/db2setup`

12. Enter the password of the DB2 administrative user. This password will also be used to log in to the BigFix database and to digitally sign all actions taken from the BigFix console.
13. Choose the key size that you want to use or press <Enter> to accept the default value which is 2:

```
Key size level
Provide the key size that you want to use:
[1]  'Min' level (2048 bits)
[2]  'Max' level (4096 bits)
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```

14. Choose a folder for your private key (`license.pvk`), the license certificate (`license.crt`), and the site masthead (`masthead.afxm`) or press <Enter> to accept the default value which is `./license`.
15. Choose if you want to use the proxy to access the internet or press <Enter> to accept the default value which is 2:

```
Proxy usage
[1]  Use the proxy to access the internet
[2]  Do not use the proxy
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```

16. Choose the value of the encoding that will be used for the content (FXF Encoding) or press <Enter> to accept the default value which is in the following example:

```
[8]   Western European languages ( Latin 1 )  - [1252]
```

17. Specify the DB2 port number or press <Enter> to accept the default value which is 50000.

After performing these steps, the evaluation license was generated successfully. If you chose to generate a response file named `response.txt`, verify its content after the installation.

Performing an evaluation installation with MS SQL

To install a BigFix server with an evaluation license on Linux with MS SQL, perform the following steps:

1. On the computer where you want to install the BigFix server, from the shell where you extract the contents of the `.tgz` server installer, enter the following command:

   ```
   ./install.sh
   ```

2. To install the Evaluation version, enter `1`:

   ```
   Select the type of installation
   [1]  Evaluation: Request a free evaluation license from HCL
   This license allows you to install a fully functional copy of HCL
    BigFix on up to 1000 clients,
   for a period of 30 days.
   [2]  Production: Install using a production license or an
    authorization for a production license.
   Choose one of the options above or press <Enter> to accept the default
    value: [1]
   ```

3. After reading the License Agreement, enter `1` to accept it and continue.
4. Enter your First Name, Last Name, Email address and Organization's Name to create the digital signature.
5. Enter the name with which your server is registered to the DNS. It will be used by the BigFix clients to identify the BigFix server. It cannot be changed after a license is created.
6. Enter the server identification port to use for all communication by the BigFix components or press <Enter> to accept the default value which is 52311.
7. Enter the Web Reports server HTTPS port number or press <Enter> to accept the default value which is 8083.

8. Enter the WebUI HTTPS port number or press <Enter> to accept the default value which is 443.
9. Enter the WebUI HTTP redirect port number or press <Enter> to accept the default value which is 80.
10. To configure the firewall, enter `1`:

```
The firewall of the operating system is active on the local server.
To enable the communication using the specified ports you can:
[1]  Configure the firewall now
[2]  Configure the firewall later
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```

11. Choose the key size that you want to use or press <Enter> to accept the default value which is 2:

```
Key size level
Provide the key size that you want to use:
[1]  'Min' level (2048 bits)
[2]  'Max' level (4096 bits)
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```

12. Specify the local SQL Server administrative user:

```
Specify the user name of the local SQL Server instance owner that you
 want to use
or press <Enter> to accept the default value: sa
```

13. Specify the local SQL Server administrative user password:

```
Specify the password of the local SQL Server administrative user:
password
Enter the password again for verification:
password
```

14. Choose a folder for your private key (`license.pvk`), the license certificate (`license.crt`), and the site masthead (`masthead.afxm`) or press <Enter> to accept the default value which is `./license`.
15. Choose if you want to use the proxy to access the internet or press <Enter> to accept the default value which is 2:

```
Proxy usage
[1]  Use the proxy to access the internet
[2]  Do not use the proxy
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```

16. Choose the value of the encoding that will be used for the content (FXF Encoding) or press <Enter> to accept the default value which is in the following example:

```
[8]   Western European languages ( Latin 1 )  - [1252]
```

17. Specify the local SQL Server port:

```
Specify the local SQL Server port number or press <Enter> to accept
 the default value:
1433
```

After performing these steps, the evaluation license was generated successfully. If you chose to generate a response file named `response.txt`, verify its content after the installation.

Installation Command Options

You can run the Production or Evaluation installation in interactive or silent mode. You can use a single installer for different components.

The full command to run any type of installation is the following:

```
./install.sh [ -f <input_response_file> ] [ -g <output_response_file> ]
 [ -upgrade ]
```

```
[ -reuseDb ] [ -opt <key_name1>=<key_value1> ] [ -opt
 <key_name2>=<key_value2> ] ...
```

where:

`-f <input_response_file>`

Specifies the full path and file name of the response file to use.

`-g <output_response_file>`

Generates a response file.

`-upgrade`

Runs the script to upgrade all the components.

`-reuseDb`

Allows you to use an existing database. If during the disaster recovery the installation program finds `BFENT` or `BESREPOR` databases, it uses them.

`-opt <key_name>=<key_value>`

Allows you to override at runtime a value assigned to a key in the response file.

Installing the components

Installing the Server

To install the BigFix Server with DB2 database, see Installing the Server with DB2 (on page 205).

To install the BigFix Server with MS SQL database , see Installing the Server with MS SQL (on page 212).

Starting from Patch 2, BigFix Server on Windows Server (2019 or later) and on Linux Red Hat 9 systems supports Google Cloud SQL for Microsoft SQL Server 2022 database. Only the SQL Authentication is supported. DSA configurations are not supported with Google Cloud SQL for Microsoft SQL Server 2022 database.

Starting from Patch 4, BigFix Server on Windows Server (2019 or later) and on Linux Red Hat 9 systems supports:

- Azure SQL Managed Instance for Microsoft SQL Server 2022 database
- Amazon RDS for Microsoft SQL Server 2022 database.

Only the SQL Authentication is supported. DSA configurations are not supported.

Installing the Server with DB2

Before running the installation, to ensure you have all the prerequisites, see Server requirements (on page 76).

Note: The installation program installs all prerequisites using Yum. For information about how to configure Yum and Yum repositories see Configuring Yum and Yum Repositories.

To install the BigFix Server in your production environment, perform the following steps:

1. From the shell where you extract the server package, move to the installation directory, `ServerInstaller_11.0.xxx-rhe7.x86_64` and enter the following command:

   ```
   ./install.sh
   ```

   If you want to generate a response file for future unattended installations, add the -g option followed by the path where to store the response file, as follows:

   ```
   ./install.sh -g response.txt
   ```

2. To install the Production, enter `2`.

   Note: If you enter `1` to run the evaluation installation, consider that this type of installation does not support the enhanced security option. For more information about this feature, see Security Configuration Scenarios.

3. After reading the License Agreement, enter `1` to accept it and continue.
4. Select `1` if you want to install all the components.
5. Enter `1` to create a single database or a Master database for later replication. Enter `2` if you want to create a replica of an existing master database. For additional information, see Using multiple servers (DSA).

```
Select the database replication:
[1]  Single or master database
[2]  Replicated database
Choose one of the options above or press  <Enter>  to accept the
 default value: [1]
```

6. To use a local database, enter `1`:

```
Select the database:
[1]  Use a local database
[2]  Use a remote database
Choose one of the options above or press  <Enter>  to accept the
 default value: [1]
```

   The local database name of BigFix server is `BFENT`. The local database name of Web Reports is `BESREPOR`.

   Note: To use a remote DB2 client node for BigFix, see Installing and configuring DB2 *(on page 190)*.

7. Enter the location where the downloaded files for the Clients are stored:

```
Choose the server's root folder:
Specify the location for the server's root folder or
press <Enter> to accept the default value: /var/opt/BESServer
```

8. Enter the location where Web Reports stores its files:

```
Choose the Web Reports server's root folder:
Specify the location for the Web Reports server's root folder or
```

```
press <Enter> to accept the default
 value: /var/opt/BESWebReportsServer
```

9. Enter the Web Reports HTTPS port number:

```
Choose the Web Reports server's port number:
Specify the port number or press <Enter> to accept the default value:
 8083
```

10. Enter the WebUI HTTPS port number:

```
Specify the port number or press <Enter> to accept the default value:
 443
```

11. Enter the WebUI HTTP redirect port number:

```
Specify the port number or press <Enter> to accept the default value:
 80
```

12. Specify the name of the local DB2 instance used by BigFix, or accept the default name.



Note: The DB2 instance to be used is always the instance local to the system where you are installing the server. If you are performing an installation with a remote database, you must use the DB2 instance specified on the DB2 client and not the one specified on the remote DB2 server.

13. Enter the user name for the DB2 local administrative user. The default is `db2inst1`.

```
DB2 local administrative user
Specify the user name of the local DB2 instance owner that you want to
 use or press <Enter>
to accept the default value: db2inst1
```

14. Enter the DB2 local administrative user password.

```
DB2 local administrative user password:
Specify the password of the local DB2 administrative user:
```

15. Enter `1` to apply an optimized configuration to the DB2 instance or `2` to skip the configuration.
16. Enter the name of the BigFix administrative user.

```
Create the initial administrative user:
Specify the Username for the new user or press <Enter> to accept the
 default value:
```

17. Enter the password of the BigFix administrative user.

```
Create the initial administrative user:
Specify the password for the new user:
```

18. If the local firewall is running, the installation program allows you to configure it automatically.

```
Firewall configuration
The firewall of the operating system is active on the local server.
To enable the communication using the specified ports you can:
[1]  Configure the firewall now
[2]  Configure the firewall later
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```



Note: If you run the installation on a RHEL 7 system, you might be using **firewalld** instead of **iptables** for managing the firewall. In this case you have to configure the firewall rules manually as a post-installation step.

19. To run the installation using a BES license authorization file, enter `1`.

```
Choose the setup type that best suits your needs:
[1]  I want to install with a BES license authorization file
[2]  I want to install with a production license that I already have
[3]  I want to install with an existing masthead
Choose one of the options above or press <Enter> to accept the default
 value: [1]
```

Note: If you already ran a first installation, or part of it, you can specify option `2` or `3`, to install with an existing production license (`license.crt, license.pvk`) or an existing masthead (`masthead.afxm`).

20. Specify if you want to connect to the internet through a proxy.

```
Proxy usage
[1]  Use the proxy to access the internet
[2]  Do not use the proxy
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```

Note: If you chose to use a proxy, before moving to the next step, perform the steps described in Configuring the proxy *(on page 223)*.

21. If you chose to install with a BES license authorization file, specify its location.
22. Specify the DNS name or IP address of the computer where you are installing the server. This name is saved in your license and will be used by clients to identify the BigFix server. It cannot be changed after a license is created.
23. If you chose to install with a BES license authorization file, specify the password to be used to encrypt the Site Admin Private Key file that will be generated.

```
Site admin private key password:
Specify the related site admin private key password:
```

24. Specify the size in bits of the key to be used to encrypt the HTTPS traffic.

```
Key Size Level
Provide the key size that you want to use:
[1]  'Min' Level (2048 bits)
[2]  'Max' Level (4096 bits)
Choose one of the options above or press <Enter> to accept the
 default: [2]
```

25. Choose the folder where the installation will save the generated files: `license.crt`, `license.pvk` and `masthead.afxm`.

```
Choose License Folder:
Specify a folder for your private key (license.pvk), license
 certificate
(license.crt), and site masthead (masthead.afxm) or press  <Enter>  to
 accept
the default: ./license
```

26. Decide how to send your activation request to HCL. If your computer is connected to the Internet, you can submit it now by entering `1`. If you choose `1`, move to the next installation step.

    If you choose `2`, see Submitting the license request (on page 220).

27. If you chose to install with a production license that you already had, specify the following:
    a. The location of the license certificate file.
    b. The location of the Site Admin Private Key file.
    c. The Site Admin Private Key password.
28. Specify the encoding used to store the content:

```
Specify the value of the encoding that will be used for the content
 (FXF Encoding)
[1]   Thai                                              - [874]
[2]   Japanese                                          - [932]
[3]   Chinese (simplified)                              - [936]
[4]   Korean                                            - [949]
[5]   Chinese (traditional)                             - [950]
[6]   Central European languages ( Latin 2 )            - [1250]
[7]   Cyrillic                                         - [1251]
[8]   Western European languages ( Latin 1 )           - [1252]
[9]   Greek                                           - [1253]
[10]  Turkish                                         - [1254]
[11]  Hebrew                                          - [1255]
```

```
[12]  Arabic                                                - [1256]
[13]  Baltic                                                - [1257]
[14]  Vietnamese                                            - [1258]
Choose one of the options above or press <Enter> to accept the default
 value: [8]
```

29. Choose `1` to accept the default masthead values or `2` to customize them.

    If you decide to use custom values, see Customizing the masthead parameters (on page 226).

30. **Case 1**: If you chose to install using a BES license authorization file, the following messages confirm that your license request was successfully processed:

    ```
    Info: The license authorization file was successfully processed.
    Info: The license authorization file can be used only once.
    It was renamed
     to ./license/LicenseAuthorization.BESLicenseAuthorization.used_201808
    01
    to indicate that it has already been used.
    Info: If you want to run the installation again, start from the
     just-generated
    ./license/license.crt and ./license/license.pvk
    ```

    Case 2 If you chose to install with a production license that you already had, specify the folder where the license files will be saved.

    ```
    Choose the license folder:
    Specify a folder for your site masthead (masthead.afxm) or press
     <Enter> to accept the default value:
    ./license
    ```

    Case 3 If you chose to install with an existing masthead file, specify the following:

    a. The location of the Site Admin Private Key file.
    b. The Site Admin Private Key password.
    c. The location of the deployment masthead file.

31. Specify whether the Web Reports service will be run by the root user or not.

```
Use root user for Web Reports
If you specify true, Web Reports service will run with root
 privileges.
[1]  True
[2]  False
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```

32. If you chose to run the Web Reports service with a user different from root, specify the name of an existing user.

```
Web Reports non-root user name
Specify the name of the non-root user for Web Reports (the user must
 already exists).
```

33. Enter the port number for the DB2 connection to create the DB2 instance:

```
DB2 Connection:
Specify the DB2 Port Number or press <Enter>  to accept the default:
 50000
```

The BigFix Server installation is now complete. You can now install the BigFix Console on a Windows™ system and log in with the account you created during the installation of the server. The default BigFix administrative user is `BFAdmin`.

You can find the installation log `BESinstall.log` and the `BESAdmin` command line traces `BESAdminDebugOut.txt` in the `/var/log` folder.

Installing the Server with MS SQL

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.

Starting from Patch 2, BigFix Server on Windows Server (2019 or later) and on Linux Red Hat 9 systems supports Google Cloud SQL for Microsoft SQL Server 2022 database. Only the SQL Authentication is supported. DSA configurations are not supported with Google Cloud SQL for Microsoft SQL Server 2022 database.

Starting from Patch 4, BigFix Server on Windows Server (2019 or later) and on Linux Red Hat 9 systems supports:

- Azure SQL Managed Instance for Microsoft SQL Server 2022 database
- Amazon RDS for Microsoft SQL Server 2022 database.

Only the SQL Authentication is supported. DSA configurations are not supported.

Before running the installation, to ensure you have all the prerequisites, see Server requirements (on page 76).

Note: The installation program installs all prerequisites using Yum. For information about how to configure Yum and Yum repositories, see Configuring Yum and Yum Repositories.

To install the BigFix Server in your production environment, perform the following steps:

1. From the shell where you extract the server package, move to the installation directory, `ServerInstaller_ms_11.0.xxx-rhe7.x86_64` and enter the following command:

   ```
   ./install.sh
   ```

   If you want to generate a response file for future unattended installations, add the -g option followed by the path where to store the response file, as follows:

   ```
   ./install.sh -g response.txt
   ```

2. To install the Production, enter `2`.



Note: If you enter `1` to run the evaluation installation, consider that this type of installation does not support the enhanced security option. For more information about this feature, see Security Configuration Scenarios.

3. After reading the License Agreement, enter `1` to accept it and continue.
4. Select `1` if you want to install all the components.
5. To use a local database, enter `1`:

```
Select the database:
[1]  Use a local database
[2]  Use a remote database
Choose one of the options above or press  <Enter>  to accept the
 default value: [1]
```

   The local database name of BigFix server is `BFEnterprise`. The local database name of Web Reports is `BESReporting`.
6. Enter the location where the downloaded files for the Clients are stored:

```
Choose the server's root folder:
Specify the location for the server's root folder or
press <Enter> to accept the default value: /var/opt/BESServer
```

7. Enter the location where Web Reports stores its files:

```
Choose the Web Reports server's root folder:
Specify the location for the Web Reports server's root folder or
press <Enter> to accept the default
 value: /var/opt/BESWebReportsServer
```

8. Enter the Web Reports HTTPS port number:

```
Choose the Web Reports server's port number:
Specify the port number or press <Enter> to accept the default value:
 8083
```

9. Enter the WebUI HTTPS port number:

```
Specify the port number or press <Enter> to accept the default value:
 443
```

10. Enter the WebUI HTTP redirect port number:

```
Specify the port number or press <Enter> to accept the default value:
 80
```

11. Enter the local SQL Server administrative user:

```
Specify the user name of the local SQL Server instance owner that you
 want to use or
press <Enter> to accept the default value: sa
```

12. Enter the local SQL Server administrative user password:

```
Specify the password of the local SQL Server administrative user:

Enter the password again for verification:
```

13. Enter the name of the BigFix administrative user.

```
Create the initial administrative user:
Specify the Username for the new user or press <Enter> to accept the
 default value: BFAdmin
```

14. Enter the password of the BigFix administrative user.

```
Create the initial administrative user:
Specify the password for the new user:
```

15. If the local firewall is running, the installation program allows you to configure it automatically.

```
Firewall configuration
The firewall of the operating system is active on the local server.
To enable the communication using the specified ports you can:
[1]  Configure the firewall now
[2]  Configure the firewall later
```

```
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```

Note: If you run the installation on a RHEL 7 system, you might be using **firewalld** instead of **iptables** for managing the firewall. In this case you have to configure the firewall rules manually as a post-installation step.

16. To run the installation using a BES license authorization file, enter `1`.

```
Choose the setup type that best suits your needs:
[1]  I want to install with a BES license authorization file
[2]  I want to install with a production license that I already have
[3]  I want to install with an existing masthead
Choose one of the options above or press <Enter> to accept the default
 value: [1]
```

Note: If you already ran a first installation, or part of it, you can specify option `2` or `3`, to install with an existing production license (`license.crt, license.pvk`) or an existing masthead (`masthead.afxm`).

17. Specify if you want to connect to the internet through a proxy.

```
Proxy usage
[1]  Use the proxy to access the internet
[2]  Do not use the proxy
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```

Note: If you chose to use a proxy, before moving to the next step, perform the steps described in Configuring the proxy *(on page 223)*.

18. If you chose to install with a BES license authorization file, specify its location.

19. Specify the DNS name or IP address of the computer where you are installing the server. This name is saved in your license and will be used by clients to identify the BigFix server. It cannot be changed after a license is created.
20. If you chose to install with a BES license authorization file, specify the password to be used to encrypt the Site Admin Private Key file that will be generated.

```
Site admin private key password:
Specify the related site admin private key password:
```

21. Specify the size in bits of the key to be used to encrypt the HTTPS traffic.

```
Key Size Level
Provide the key size that you want to use:
[1]  'Min' Level (2048 bits)
[2]  'Max' Level (4096 bits)
Choose one of the options above or press <Enter> to accept the
 default: [2]
```

22. Choose the folder where the installation will save the generated files: `license.crt`, `license.pvk` and `masthead.afxm`.

```
Choose License Folder:
Specify a folder for your private key (license.pvk), license
 certificate
(license.crt), and site masthead (masthead.afxm) or press  <Enter>  to
 accept
the default: ./license
```

23. Decide how to send your activation request to HCL. If your computer is connected to the Internet, you can submit it now by entering `1`. If you choose `1`, move to the next installation step.

    If you choose `2`, see Submitting the license request (on page 220).
24. If you chose to install with a production license that you already had, specify the following:

a. The location of the license certificate file.
b. The location of the Site Admin Private Key file.
c. The Site Admin Private Key password.

25. Specify the encoding used to store the content:

```
Specify the value of the encoding that will be used for the content
 (FXF Encoding)
[1]   Thai                                          - [874]
[2]   Japanese                                      - [932]
[3]   Chinese (simplified)                          - [936]
[4]   Korean                                        - [949]
[5]   Chinese (traditional)                         - [950]
[6]   Central European languages ( Latin 2 )        - [1250]
[7]   Cyrillic                                      - [1251]
[8]   Western European languages ( Latin 1 )        - [1252]
[9]   Greek                                         - [1253]
[10]  Turkish                                       - [1254]
[11]  Hebrew                                        - [1255]
[12]  Arabic                                        - [1256]
[13]  Baltic                                        - [1257]
[14]  Vietnamese                                    - [1258]
Choose one of the options above or press <Enter> to accept the default
 value: [8]
```

26. Choose `1` to accept the default masthead values or `2` to customize them.

    If you decide to use custom values, see Customizing the masthead parameters (on page 226).

27. **Case 1**: If you chose to install using a BES license authorization file, the following messages confirm that your license request was successfully processed:

```
Info: The license authorization file was successfully processed.
Info: The license authorization file can be used only once.
```

```
It was renamed
 to ./license/LicenseAuthorization.BESLicenseAuthorization.used_201808
01
to indicate that it has already been used.
Info: If you want to run the installation again, start from the
 just-generated
./license/license.crt and ./license/license.pvk
```

Case 2 If you chose to install with a production license that you already had, specify the folder where the license files will be saved.

```
Choose the license folder:
Specify a folder for your site masthead (masthead.afxm) or press
 <Enter> to accept the default value:
./license
```

Case 3 If you chose to install with an existing masthead file, specify the following:

a. The location of the Site Admin Private Key file.
b. The Site Admin Private Key password.
c. The location of the deployment masthead file.

28. Specify whether the Web Reports service will be run by the root user or not.

```
Use root user for Web Reports
If you specify true, Web Reports service will run with root
 privileges.
[1]  True
[2]  False
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```

29. If you chose to run the Web Reports service with a user different from root, specify the name of an existing user.

```
Web Reports non-root user name
Specify the name of the non-root user for Web Reports (the user must
 already exists).
```

30. Specify the local SQL Server port number.

```
Specify the local SQL Server port number or press <Enter> to accept
 the default value: 1433
```

SQL Server named instances are only supported by specifying the related TCP port number, so for a named instance to be used with the BigFix Server it is recommended to set the TCP port as static.

The BigFix Server installation is now complete. You can now install the BigFix Console on a Windows™ system and log in with the account you created during the installation of the server. The default BigFix administrative user is `BFAdmin`.

You can find the installation log `BESinstall.log` and the `BESAdmin` command line traces `BESAdminDebugOut.txt` in the `/var/log` folder.

Submitting the license request

How to submit a license request.

To install a production copy of BigFix, you must first purchase a license from HCL.

During the installation you can choose different types of setup depending on the license input file you have:

```
I want to install with a BES license authorization file
   I want to install with a Production license that I already have
   I want to install with an existing masthead
```

BES license authorization file

After you purchase a license from HCL you receive a BigFix license authorization file. You must use this file the first time you run a production installation.

The sales agent will want to know how many clients you intend to install. Based on this, the agent creates, signs, and emails you a **License Authorization** file, which will have a name like `CompanyName.BESLicenseAuthorization`.

If you run this installation and do not have access to the Internet, a temporary request (`beslicense.request`) is generated to request a production license (`license.crt`) from the BigFix License Server and a `license.pvk` private key file. You can leave the installation in pending status until you receive the production license.

Copy the request named `request.BESLicenseRequest` on to a machine with access to Internet, visit the BigFix website, post your request, and download your certificate. After you downloaded the certificate, copy it to the machine on which you are installing the server and continue the installation. If you exited the installation, to install the server you must run the installation using the option that requires an existing **Production license** file.



Note: The DNS/IP address that you choose becomes a permanent part of your deployment and must never change. For flexibility, it is strongly recommended that you use a DNS name instead of a static IP address.

The installation program collects further information about your deployment and then creates the digital signature key `license.pvk` and a file called the action site masthead. This file combines configuration information (IP addresses, ports, and so on.) and license information (how many Clients are authorized and for how long) together with a public key that is used to verify the digital signatures.

Production license

Use this option if you have already the production license `license.crt` and the private key file on the machine on which you are installing the server, but did not complete the server installation.

An existing masthead

Use this type of installation to reinstall the BigFix server or a DSA server. The input file needed to run this installation is the action site masthead file that was generated during the first installation. The action site masthead has the extension `.afxm` and acts as a configuration file with parameters such as the BigFix server IP address or server name, port number, and locking behavior. It contains information necessary for the digital signature security scheme that BigFix uses (the masthead contains the public key information), and the licensing information that allows BigFix users to run BigFix with a specified number of users for a specified length of time. The BigFix Server installer requires the masthead file be in the server installation folder.

Decide how to send your activation request to HCL. If your computer is not connected to the Internet or cannot reach the HCL license servers, you must select option `2`. If you chose `2`, a request file named `request.BESLicenseRequest` is generated.

To submit the request file to HCL, use the following Web site: http://support.bigfix.com/bes/forms/BESLicenseRequestHandler.html

You can, then, continue the installation by importing the certificate file that you received (`license.crt`) or exit from the installation and rerun it at a later time.

```
Import License Certificate
[1]  Continue with the installation importing the certificate
 (license.crt).
[2]  Exit from the installation, I will import the certificate at a later
 time.
```

If you exit the installation, you can rerun `./install.sh` later and repeat all the steps specifying that you want to use the generated license with option `2`:

```
Choose the setup type that best suits your needs:
[1]  I want to install with a BES license authorization file
[2]  I want to install with a Production license that I already have
[3]  I want to install with an existing masthead
```

From this point on, you can proceed with the normal installation procedure.

Configuring the proxy

This procedure explains how to set up a proxy connection when installing the BigFix server.

If you need to set up a proxy connection after the server installation, see Setting a proxy connection on the server (on page 552).

1. Specify if you want to connect to the internet through a proxy.

```
Proxy usage
[1]  Use the proxy to access the internet
[2]  Do not use the proxy
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```

2. Enter the proxy hostname or IP address.

```
Proxy hostname
Specify the hostname or the IP address of the Server that acts as a
 proxy:
```

3. Specify, if required, the port number of your proxy.

```
Proxy port
Specify the port of the proxy or press <Enter> if this parameter is
 not required:
```

4. You can accept the default proxy settings or, alternatively, you can assign different values.

```
Advanced proxy parameters
The proxy will be configured using the following defaults:
 Proxy user: none
 Proxy password: none
 Proxy tunneling capability: let proxy decide
 Authentication method: all methods allowed by the proxy
```

```
 Proxy exception list: localhost,127.0.0.1
 Use the proxy for downstream notification: false
[1]  Use the default values
[2]  Set advanced proxy parameters
Choose one of the options above or press <Enter> to accept the default
 value: [1]
```

Note:

- If you want to enable FIPS mode, ensure that the proxy configuration is set up to use an authentication method other than `digest`, `negotiate` or `ntlm`.
- If you specify to use the `negotiate` authentication method on a server or relay, different authentication methods might be used.
- The proxy configuration specified at installation time is saved in the server configuration file `BESServer.config` and it is used also at runtime.

5. Specify the user name used to connect to the proxy, if required, or leave it empty. If you specify a user name, you will also be prompted to enter its password.

```
Proxy user
Specify the proxy username or press <Enter> if no user is required:
```

6. Specify the authentication method to be used for your proxy.

```
Proxy authentication method
The proxy authentication method can be one of the values listed below
 or a combination of them separated
by a comma.
Warning: If you want to enable FIPS compliant cryptography consider
 that digest, ntlm, and negotiate
are not compatible with FIPS.
 basic
 digest
```

```
 ntlm
 negotiate
Specify the proxy authentication method or press <Enter> if all
 methods allowed by the proxy can be used :
```

7. Specify, if required, a proxy exception list.

```
Proxy exception list
Note: If the exception list field is kept empty, the communication to
 "localhost" and "127.0.0.1" will not
pass through the proxy by default. If you specify something in the
 exception list field, ensure that you
add also these two values, otherwise the proxy will be used to reach
 them.
Specify the exception list using comma (,) to separate entries or
 press <Enter> to keep it empty:
```

8. Specify if you want to enforce the proxy tunneling capability.

```
Proxy tunneling capability
Choose one of the following options:
[1]  Enforce proxy tunneling
[2]  Let proxy decide
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```

9. Specify if you want to use the proxy also for downstream notifications.

```
Enable the use of the proxy for downstream notification.
If you specify true, the proxy is used by the BES server also for
 communicating with the relay.
[1]  True
[2]  False
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```

10. Optionally, you can test if the connection to the proxy can be successfully established.

```
Test the connection using the proxy
If you want to enable FIPS 140-2 compliant cryptography, select "Test
 the connection using FIPS".
Warning: There are some proxy authentication methods that are not
 compatible with FIPS.
[1]  Test the connection
[2]  Test the connection using FIPS
[3]  Do not test the connection
Choose one of the options above or press <Enter> to accept the default
 value: [1]
```

Customizing the masthead parameters

How to customize the masthead parameters.

At the following installation step, choose `2`:

```
Advanced masthead parameters
The masthead will be created using the following defaults:
 Server port number: 52311
 Use of FIPS 140-2 compliant cryptography: Disabled
 Gather interval: One Day
 Initial action lock: Unlocked
 Action lock controller: Console
 Action lock exemptions: Disabled
 Unicode filenames in archives: Enabled
The above default values are suitable for most of BigFix deployments.
[1]  Use default values
[2]  Use custom values
Choose one of the options above or press <Enter> to accept the default
 value: [1]
```

You can change the following masthead parameters:

Server port number

Specify the number of the server port. The default value is: `52311`.

```
Server port number
Specify the server port or press <Enter> to accept the default:
 52311
```

Note: Do not use port number `52314` for the network communication between the BigFix components because it is reserved for proxy agents.

Enable use of FIPS 140-2 compliant cryptography

Use this setting to specify whether or not to be compliant with the Federal Information Processing Standard in your network. Enter 1 to enable it, `2` to disable it. The default value is `2`.

```
Enable the use of FIPS 140-2 compliant cryptography
[1]  Use of FIPS enabled
[2]  Use of FIPS disabled
Choose one of the options above or press <Enter> to accept the d
efault value: [2]
```

Note: Enabling FIPS mode prevents the use of some authentication methods when connecting to a proxy. If you chose to use a proxy to access the Internet or to communicate with subcomponents, ensure that you selected an authentication method other than `digest`, `negotiate` or `ntlm`.

Gathering interval

This option determines how long the clients wait without hearing from the server before checking whether new content is available. The default waiting time is one day (option `6`).

```
Gathering interval
Specify the time interval that you want to use. The default valu
e is suitable for most of the BigFix deployments.
[1]  Fifteen minutes
[2]  Half an hour
[3]  One hour
[4]  Eight hours
[5]  Half day
[6]  One day
[7]  Two days
[8]  One week
[9]  Two weeks
[10]  One month
[11]  Two months
Choose one of the options above or press <Enter> to accept the d
efault value: [6]
```

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 (option `3`) and to lock specific clients later on, as required.

```
Initial action lock
[1]  Locked
[2]  Lock duration
[3]  Unlocked
Choose one of the options above or press <Enter> to accept the d
efault value: [3]
```

Action lock controller

This parameter determines who can change the action lock state. By default, it is the Console (option `1`).

```
Action lock controller
[1]  Console
[2]  Client
[3]  Nobody
Choose one of the options above or press <Enter> to accept the d
efault value: [1]
```

Enable lock exemptions

In rare cases, you might need to exempt a specific URL from any locking actions. This setting allows you to disable or disable this function. The default choice is to leave it disabled (option `2`).

```
Enable lock exemptions
[1]  Lock exemption enabled (fairly unusual)
[2]  Lock exemption disabled
Choose one of the options above or press <Enter> to accept the d
efault value: [2]
```

Enable the use of Unicode file names in archives

This setting specifies the codepage used to write file names in the BigFix archives. The default choice is to use Unicode (option `1`).

```
Enable the use of Unicode filenames in archives
[1]  The use of Unicode filenames in archives is enabled.
[2]  The use of Unicode filenames in archives is disabled.
Choose one of the options above or press <Enter> to accept the d
efault value: [1]
```

After this step, the `masthead.afxm` file is created with the specified parameters.

Installing Web Reports Standalone with DB2

If you run a fresh installation of Web Reports, the HTTPS configuration is automatically enabled on port 8083.

After the installation completes successfully, you can switch to the HTTP configuration by changing the value of the `_WebReports_HTTPServer_UseSSLFlag` setting to 0. For more information, see Customizing HTTPS on Web Reports.

To install the BigFix Web Reports in your production environment, perform the following steps:

1. From the shell where you extract the server package, move to the installation directory
   - `ServerInstaller_11.0.xxx-rhe7.x86_64`

   and enter the following command:

   ```
   ./install.sh
   ```
2. Install the Production type by entering `2`, because the Evaluation type does not allow to install the components separately.
3. After reading the License Agreement, enter `1` to accept it and continue.

   Note: If you have already installed the BigFix server, the License Agreement is not displayed.

4. Select `3` if you want to install the Web Reports component only:

   ```
   Select the BigFix features that you want to install:
   [1]  All components (server, client, Web Reports and WebUI)
   [2]  Server and client only
   [3]  Web Reports only
   [4]  WebUI and client only
   [5]  Server, WebUI and client only
   [6]  Web Reports, WebUI and client only
   [7]  Server, Web Reports and client only
   Choose one of the options above or press <Enter> to accept the default
    value: [1]
   ```
5. To use a local database, enter `1`:

   ```
   Select the database:
   [1]  Use a local database
   ```

```
[2]  Use a remote database
Choose one of the options above or press  <Enter>  to accept the
 default: [1]
```

The local database name is `BESREPOR`.

Note:

To use a remote database for BigFix Web Reports, you must perform the following steps:

a. Install the Web Reports DB2® server on the remote workstation.
b. Install a DB2® client on the workstation from where you run the BigFix Server installation
c. Connect the DB2® server to the DB2® client installed on the workstation from where you run the installation, that is, the port of the DB2® database (default 50000) must be reachable by the workstation where the installation is running.
d. Provide the following information in the installation procedure:
    i. The remote DB2 node
    ii. The DB2 port number
    iii. The user name of the local DB2 instance owner for the remote DB2 client and the remote DB2 server.

Important: Before entering the user names of the DB2 instance owners ensure that the related DB2 instances are up and running. The DB2 instance names used to install the BES Root Server cannot contain the following special characters: blanks, tabs `\t`, returns `\n` and `; & | " ' < >`

6. Enter the location where the Web Reports server stores its files:

```
Choose the WebReports server's root folder:
Specify the location for the WebReports server's root folder or
press  <Enter>  to accept the default: /var/opt/BESWebReportsServer
```

7. Enter the Web Reports server port number:

```
Web Reports will be automatically configured to use HTTPS.
Choose the Web Reports server HTTPS port number.
Specify the port number or press <Enter> to accept the default value:
 8083
```

8. You can specify a name of the DB2 instance used by BigFix different from the name of the DB2 user.

```
DB2 instance name
The DB2 instance used by the BigFix requires specific configuration
for performance optimization. It is therefore suggested that you use
a dedicated DB2 instance.
Specify the name of the DB2 instance that you want to use or press
 <Enter>
to accept the default value: db2inst1
```

9. Enter the user name for the DB2® Local Administrative user. The default is `db2inst1`.

```
DB2 local administrative user
Specify the user name of the local DB2 instance owner that you want to
 use
or press <Enter> to accept the default value: db2inst1
```

10. Enter the DB2® Local Administrative user password.

```
DB2 local administrative user password:
Specify the password of the local DB2 administrative user:

Enter the password again for verification:
```

11. Enter 1 to configure the specified DB2 instance.

```
DB2 instance configuration
The specified DB2 instance can be configured to optimize the BigFix
 performance.
```

```
Be aware that the configuration settings will be applied to all
 databases that belong
to the selected DB2 instance.
[1]  Configure the specified DB2 instance.
[2]  Skip the DB2 instance configuration.
Choose one of the options above or press <Enter> to accept the default
 value: [1]
```

12. If the local firewall is running, enter 2 to perform the configuration later:.

```
Firewall configuration
The firewall of the operating system is active on the local server.
To enable the communication using the specified ports you can:
[1]  Configure the firewall now
[2]  Configure the firewall later
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```

13. Specify the masthead file (default is `/etc/opt/BESServer/actionsite.afxm`):

```
Deployment masthead
Specify the masthead file (masthead.afxm or actionsite.afxm) for your
 deployment
or press <Enter> to accept the default
 value: /etc/opt/BESServer/actionsite.afxm
```

14. Specify the DNS name or IP address of the machine on which to install Web Reports. This name is saved in your license and will be used by clients to identify the BigFix server. It cannot be changed after a license is created.

```
WebReports server DNS name
Enter the DNS name of your BigFix WebReports server
or press <Enter> to accept the default: 'hostname'
```

15. You can specify the user name you want to use to install the Web Reports component. You can choose either the root user, or another existing user.

```
Use root user for WebReports
If you specify true, WebReports service will run with root privileges.
[1]  True
[2]  False
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```

If you choose option 2, you can specify a non-root user. This user does not need to have any particular rights or to belong to a group with particular rights to be used as a Web Reports installation user:

```
WebReports non-root user name
Specify the name of the non-root user for WebReports (the user must
 already exists).
```

Note: If you are upgrading from an earlier version, the Web Reports service user remains the same as before the upgrade.

16. Enter the port number for the DB2 connection to create the DB2 instance:

```
####################
DB2 Connection:
Specify the DB2 Port Number or press <Enter>  to accept the default:
 50000
```

The BigFix Web Reports installation is now complete.

You can see installation errors in the `BESinstall.log` and the `BESAdmin` command line traces in the `BESAdminDebugOut.txt` files under the `/var/log` directory.

Installing Web Reports Standalone with MS SQL

If you run a fresh installation of Web Reports, the HTTPS configuration is automatically enabled on port 8083.

After the installation completes successfully, you can switch to the HTTP configuration by changing the value of the `_WebReports_HTTPServer_UseSSLFlag` setting to 0. For more information, see Customizing HTTPS on Web Reports.

To install the BigFix Web Reports in your production environment, perform the following steps:

1. From the shell where you extract the server package, move to the installation directory
   - `ServerInstaller_ms_11.0.xxx-rhe7.x86_64`

   and enter the following command:

   ```
   ./install.sh
   ```
2. Install the Production type by entering `2`, because the Evaluation type does not allow to install the components separately.
3. After reading the License Agreement, enter `1` to accept it and continue.

   Note: If you have already installed the BigFix server, the License Agreement is not displayed.

4. Select `3` if you want to install the Web Reports component only:

   ```
   Select the BigFix features that you want to install:
   [1]  All components (server, client, Web Reports and WebUI)
   [2]  Server and client only
   [3]  Web Reports only
   [4]  WebUI and client only
   [5]  Server, WebUI and client only
   [6]  Web Reports, WebUI and client only
   [7]  Server, Web Reports and client only
   Choose one of the options above or press <Enter> to accept the default
    value: [1]
   ```
5. To use a local database, enter `1`:

   ```
   Select the database:
   [1]  Use a local database
   ```

```
[2]  Use a remote database
Choose one of the options above or press  <Enter>  to accept the
 default: [1]
```

The local database name is `BESReporting`.

6. Enter the location where the Web Reports server stores its files:

```
Choose the WebReports server's root folder:
Specify the location for the WebReports server's root folder or
press  <Enter>  to accept the default: /var/opt/BESWebReportsServer
```

7. Enter the Web Reports server port number:

```
Web Reports will be automatically configured to use HTTPS.
Choose the Web Reports server HTTPS port number.
Specify the port number or press <Enter> to accept the default value:
 8083
```

8. If the local firewall is running, enter 2 to perform the configuration later:

```
Firewall configuration
The firewall of the operating system is active on the local server.
To enable the communication using the specified ports you can:
[1]  Configure the firewall now
[2]  Configure the firewall later
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```

9. Specify the masthead file (default is `/etc/opt/BESServer/actionsite.afxm`):

```
Deployment masthead
Specify the masthead file (masthead.afxm or actionsite.afxm) for your
 deployment
or press <Enter> to accept the default
 value: /etc/opt/BESServer/actionsite.afxm
```

10. Specify the DNS name or IP address of the machine on which to install Web Reports. This name is saved in your license and will be used by clients to identify the BigFix server. It cannot be changed after a license is created.

```
WebReports server DNS name
Enter the DNS name of your BigFix WebReports server
or press <Enter> to accept the default: 'hostname'
```

11. You can specify the user name you want to use to install the Web Reports component. You can choose either the root user, or another existing user.

```
Use root user for WebReports
If you specify true, WebReports service will run with root privileges.
[1]  True
[2]  False
Choose one of the options above or press <Enter> to accept the default
 value: [2]
```

If you choose option 2, you can specify a non-root user. This user does not need to have any particular rights or to belong to a group with particular rights to be used as a Web Reports installation user:

```
WebReports non-root user name
Specify the name of the non-root user for WebReports (the user must
 already exists).
```



Note: If you are upgrading from an earlier version, the Web Reports service user remains the same as before the upgrade.

12. Enter the SQL Server remote host for BES Web Reports component:

```
Specify the hostname or IP of the remote SQL Server system:
hostname
```

13. Enter the SQL Server remote port for BES Web Reports component:

```
Specify the remote SQL Server port number or press <Enter> to accept
 the default value:
1433
```

14. Enter the SQL Server remote administrative user for BES Web Reports component:

```
Specify the username for the remote SQL Server administrative user or
 press <Enter> to accept the default value:
sa
```

15. Enter the SQL Server remote administrative user password for BES Web Reports component:

```
Specify the password of the remote SQL Server administrative user:
password
Enter the password again for verification:
password
```

The BigFix Web Reports installation is now complete.

You can see installation errors in the `BESinstall.log` and the `BESAdmin` command line traces in the `BESAdminDebugOut.txt` files under the `/var/log` directory.

Installing the WebUI Standalone with DB2

Install the WebUI on just one computer per deployment.

If you want to add the WebUI to a computer where the BigFix client is already installed, you can only do it by running the Fixlet named **Install BigFix WebUI Service**.

Note: You can install the WebUI component on Red Hat Linux 8 (64-bit) or Red Hat Linux 9 (64-bit).

The BigFix server installer can be used to install the WebUI only if the target machine has no BigFix client. If you want to use it to install the WebUI on a remote computer, you must first generate the WebUI authentication certificates for that computer. Log in to the computer

where you installed the BigFix server and run the `/BESAdmin.sh -createwebuicredentials` command. For more details about the command, see BESAdmin Linux Command Line (on page 422). Copy the generated certificates folder to the remote computer before running the installer on it. The WebUI must connect to the same DB2 instance used by the BigFix server, so ensure that this connection is possible from the remote computer.

To install the BigFix WebUI in your production environment, perform the following steps:

1. From the shell where you extract the server package, move to the installation directory
   - `ServerInstaller_11.0.xxx-rhe7.x86_64`

   and enter the following command:

   ```
   ./install.sh
   ```
2. Install the Production type by entering `2`, because the Evaluation type does not allow to install the components separately.
3. After reading the License Agreement, enter `1` to accept it and continue.
4. Select `4` if you want to install the WebUI component only:

   ```
   Select the BigFix features that you want to install:
   [1]  All components (server, client, Web Reports and WebUI)
   [2]  Server and client only
   [3]  Web Reports only
   [4]  WebUI and client only
   [5]  Server, WebUI and client only
   [6]  Web Reports, WebUI and client only
   [7]  Server, Web Reports and client only
   Choose one of the options above or press <Enter> to accept the default
    value: [1]
   ```
5. Enter the WebUI HTTPS port number:

   ```
   WebUI's port number
   Specify the port number or press <Enter> to accept the default value:
    443
   ```
6. Enter the WebUI HTTP redirect port number::

```
WebUI's redirect port number
Specify the port number or press <Enter> to accept the default value:
 80
```

7. Specify from which folder the WebUI credentials must be taken:

```
Choose the folder to pick the WebUI credentials from
The WebUI needs a set of certificates as credentials to authenticate
 itself to the BigFix server. These certificates must be generated on
 the server
by running the BESAdmin command --createwebuicredentials and the
 folder containing them must be copied to this computer.
Specify the folder containing the WebUI credentials or press <Enter>
 to accept the default value: ./cert_webui
```

8. Specify the masthead file for your deployment:

```
Deployment masthead
Specify the masthead file (masthead.afxm or actionsite.afxm) for your
 deployment or press <Enter> to accept the default value:
/etc/opt/BESServer/actionsite.afxm
```

9. Specify the DNS name of the machine on which to install the WebUI:

```
WebUI DNS name
This name is used by the server to identify the WebUI. Enter the DNS
 name of the WebUI or press <Enter> to accept the default value:
hostname
```

10. Enter the remote DB2 node:

```
DB2 remote hostname
Specify the hostname of the remote DB2 system:
hostname
```

11. Enter the remote DB2 port number:

```
DB2 remote port
Specify the remote DB2 port number or press <Enter> to accept the
 default value: 50000
```

12. Enter the DB2 remote administrative user:

```
DB2 remote administrative user
Specify the username for the remote DB2 administrative user or press
 <Enter> to accept the default value: db2inst1
```

13. Enter the DB2 remote administrative user password:

```
DB2 remote administrative user password
Specify the password of the remote DB2 administrative user:
password
Enter the password again for verification:
password
```

Installing the WebUI Standalone with MS SQL

Install the WebUI on just one computer per deployment.

If you want to add the WebUI to a computer where the BigFix client is already installed, you can only do it by running the Fixlet named **Install BigFix WebUI Service**.

Note: You can install the WebUI component on Red Hat Linux 8 (64-bit).

The BigFix server installer can be used to install the WebUI only if the target machine has no BigFix client. If you want to use it to install the WebUI on a remote computer, you must first generate the WebUI authentication certificates for that computer. Log in to the computer where you installed the BigFix server and run the `/BESAdmin.sh -createwebuicredentials` command. For more details about the command, see BESAdmin Linux Command Line (on page 422). Copy the generated certificates folder to the remote computer before running the installer on it. The WebUI must connect to the same MS SQL instance used by the BigFix server, so ensure that this connection is possible from the remote computer.

To install the BigFix WebUI in your production environment, perform the following steps:

1. From the shell where you extract the server package, move to the installation directory
   - `ServerInstaller_ms_11.0.xxx-rhe7.x86_64`

   and enter the following command:

   ```
   ./install.sh
   ```
2. Install the Production type by entering `2`, because the Evaluation type does not allow to install the components separately.
3. After reading the License Agreement, enter `1` to accept it and continue.
4. Select `4` if you want to install the WebUI component only:

   ```
   Select the BigFix features that you want to install:
   [1]  All components (server, client, Web Reports and WebUI)
   [2]  Server and client only
   [3]  Web Reports only
   [4]  WebUI and client only
   [5]  Server, WebUI and client only
   [6]  Web Reports, WebUI and client only
   [7]  Server, Web Reports and client only
   Choose one of the options above or press <Enter> to accept the default
    value: [1]
   ```
5. Enter the WebUI HTTPS port number:

   ```
   WebUI's port number
   Specify the port number or press <Enter> to accept the default value:
    443
   ```
6. Enter the WebUI HTTP redirect port number::

   ```
   WebUI's redirect port number
   Specify the port number or press <Enter> to accept the default value:
    80
   ```
7. Specify from which folder the WebUI credentials must be taken:

```
Choose the folder to pick the WebUI credentials from
The WebUI needs a set of certificates as credentials to authenticate
 itself to the BigFix server. These certificates must be generated on
 the server
by running the BESAdmin command --createwebuicredentials and the
 folder containing them must be copied to this computer.
Specify the folder containing the WebUI credentials or press <Enter>
 to accept the default value: ./cert_webui
```

8. Specify the masthead file for your deployment:

```
Deployment masthead
Specify the masthead file (masthead.afxm or actionsite.afxm) for your
 deployment or press <Enter> to accept the default value:
/etc/opt/BESServer/actionsite.afxm
```

9. Specify the DNS name of the machine on which to install the WebUI:

```
WebUI DNS name
This name is used by the server to identify the WebUI. Enter the DNS
 name of the WebUI or press <Enter> to accept the default value:
hostname
```

10. Enter the SQL Server remote host:

```
SQL Server remote host
Specify the hostname or IP of the remote SQL Server system:
hostname
```

11. Enter the SQL Server remote port:

```
SQL Server remote port
Specify the remote SQL Server port number or press <Enter> to accept
 the default value: 1433
```

12. Enter the SQL Server remote administrative user:

```
SQL Server remote administrative user
Specify the username for the remote SQL Server administrative user or
 press <Enter> to accept the default value: sa
```

13. Enter the SQL Server remote administrative user password:

```
SQL Server remote administrative user password
Specify the password of the remote SQL Server administrative user:
password
Enter the password again for verification:
password
```

Verifying Server Installation

Verify that an installation has completed successfully.

Perform the following steps:

1. Ensure that the following message is displayed to the standard output or in the installation log file `/var/log/BESInstall.log`.

```
The installation of 'BigFix' completed successfully.
You can now proceed to install the BigFix console on a Windows system
 and you can log on as 'BFAdmin'.
The BigFix console installer is available in the folder
 '/var/opt/BESInstallers'.
```

2. Ensure that the services associated with each installed components are up and running by entering the following commands:

```
/etc/init.d/besserver status
/etc/init.d/besfilldb status
/etc/init.d/besgatherdb status
/etc/init.d/besclient status
/etc/init.d/beswebreports status
```

Or the following commands, starting from BigFix Platform Version 11.0.2 or later, but only for RHEL, SLES and derived platforms. For more details, see Managing the BigFix Services *(on page 273)*.

```
systemctl status besserver
systemctl status besfilldb
systemctl status besgatherdb
systemctl status besclient
systemctl status beswebreports
```

3. Ensure that local or remote databases are created by switching to the local DB2 Administrative user (default: `db2inst1`) and running the list database command:

```
su - db2inst1
db2 list db directory
```

Check that the following databases are created:

- Server component: `BFENT`
- Web Reports component: `BESREPOR`

On BigFix Linux for MS SQL Server version, when the database is installed on Red Hat, ensure it is running with the command:

```
systemctl status mssql-server
```

Connect to the database (example with mssql-tools on Linux or Management Studio on Windows) to verify that the following databases are created:

- Server component: `BFEnterprise`
- Web Reports component: `BESREporting`

4. Launch the BigFix Console and provide the credentials of the first BigFix user created at installation time to ensure that the Console connects to the Server. The user default value for the production installation is `BFAdmin`, for the evaluation installation is

`EvaluationUser`. Ensure that the client installed by default on the server machine is registered.

5. Ensure that you can log on to the Web Reports from the Console by selecting **Tools -> Launch WebReports** and providing the credentials of the first user created at installation time.

Silent installation

How to perform a silent installation.

To run a silent installation enter the following command:

```
./install.sh -f response_file -opt keyword=value
```

where:

response_file

Is the file containing the keywords to install the product.

keyword=value

Is the keyword and the value of the response file you want to override.

Use the silent mode to install the BigFix server or to run problem determination on a failed installation.

Note: In the response file you can specify a subset of keywords, such as the keywords common to different systems. The missing or invalid keywords are requested by the installation program. The silent installation runs in unattended way only if all the required keywords are specified in the response file.

You can create a response file during an installation by redirecting the installation parameters in a response file using the following command:

```
./install.sh -g response_file
```

This is an example of response file for a production server installation with DB2 as deployed database:

```
##BigFix GENERATED RESPONSE FILE
BES_PREREQ_INSTALL="install"
IS_EVALUATION="false"
LA_ACCEPT="true"
COMPONENT_SRV="true"
COMPONENT_WR="true"
COMPONENT_WEBUI="true"
SINGLE_DATABASE="true"
LOCAL_DATABASE="true"
BES_WWW_FOLDER="/var/opt/BESServer"
WR_WWW_FOLDER="/var/opt/BESWebReportsServer"
WR_WWW_PORT="8083"
WEBUI_PORT="443"
WEBUI_REDIRECT_PORT="80"
INSTALL_DB2="yes"
DB2_INSTANCE_NAME="db2inst1"
DB2_DAS_USERNAME="dasusr1"
DB2_FENCED_USERNAME="db2fenc1"
DB2_INSTALL_DIR="/opt/ibm/db2/V11.5"
DB2_PORT="50000"
DB2_USERS_PWD="P@$$w0rd1"
TEM_USER_NAME="MyAdmin"
TEM_USER_PWD="P@$$w0rd1"
CONF_FIREWALL="yes"
BES_SETUP_TYPE="prodlic"
USE_PROXY="true"
PROXY_HOST="PROXYHOST.mydomain.com"
PROXY_PORT="3128"
ADV_PROXY_DEFAULT="false"
PROXY_USER="hans"
PROXY_PWD="P@$$w0rd1"
PROXY_METH="basic"
```

```
PROXY_EXLIST="localhost,127.0.0.1"
PROXY_SECTUNNEL="false"
PROXY_DOWN="false"
TEST_PROXY="nofips"
BES_CERT_FILE="/BigFix/license.crt"
BES_LICENSE_PVK="/BigFix/license.pvk"
BES_LICENSE_PVK_PWD="P@$$w0rd1"
ENCODE_VALUE="1252"
ADV_MASTHEAD_DEFAULT="false"
BES_SERVER_PORT="52311"
ENABLE_FIPS="true"
BES_GATHER_INTERVAL="5"
INITIAL_LOCK="2"
LOCK_CONTROLLER="0"
ENABLE_LOCK_EXEMPT="false"
ENABLE_ARCHIVE_UTF8="true"
BES_LIC_FOLDER="./license"
WR_USEROOT="false"
WR_NONROOT_USER_NAME="MyNoAdmin"
```

This is an example of response file for a production server installation with MS SQL as deployed database:

```
##BigFix GENERATED RESPONSE FILE
IS_EVALUATION="false"
LA_ACCEPT="true"
COMPONENT_SRV="true"
COMPONENT_WR="true"
COMPONENT_WEBUI="true"
LOCAL_DATABASE="false"
BES_WWW_FOLDER="/var/opt/BESServer"
WR_WWW_FOLDER="/var/opt/BESWebReportsServer"
WR_WWW_PORT="8083"
```

```
WEBUI_PORT="443"
WEBUI_REDIRECT_PORT="80"
TEM_USER_NAME="BFAdmin"
TEM_USER_PWD=""
CONF_FIREWALL="no"
BES_SETUP_TYPE="prodlic"
USE_PROXY="false"
BES_CERT_FILE="/BigFix/license/license.crt"
BES_LICENSE_PVK="/BigFix/license/license.pvk"
BES_LICENSE_PVK_PWD=""
ENCODE_VALUE="1252"
ADV_MASTHEAD_DEFAULT="false"
BES_SERVER_PORT="52311"
ENABLE_FIPS="false"
BES_GATHER_INTERVAL="0"
INITIAL_LOCK="2"
LOCK_CONTROLLER="0"
ENABLE_LOCK_EXEMPT="false"
ENABLE_ARCHIVE_UTF8="true"
BES_LIC_FOLDER="/BigFix/license"
WR_USEROOT="true"
MSSQL_HOSTNAME="MSSQLHost.mydomain.com"
MSSQL_PORT="1433"
MSSQL_CONN_USER="sa"
MSSQL_CONN_PWD=" P@$$w0rd1"
```

This is an example of response file for an evaluation server installation with DB2 as deployed database:

```
##BIGFIX GENERATED RESPONSE FILE
LA_ACCEPT="true"
IS_EVALUATION="true"
CREDENTIAL_USER_FIRSTNAME="John"
```

```
CREDENTIAL_USER_LASTNAME="Smith"
CREDENTIAL_EMAIL="john.smith@mydomain.com"
CREDENTIAL_ORG="HCL US"
SRV_DNS_NAME="DNSHOST.mydomain.com"
BES_SERVER_PORT="52311"
WR_WWW_PORT="8083"
CONF_FIREWALL="no"
DB2_INSTANCE_NAME="db2inst1"
DB2_ADMIN_USER="db2inst1"
DB2_ADMIN_PWD="P@$$w0rd1"
DB2_PORT="50000"
BES_LIC_FOLDER="/BigFix"
PVK_KEY_SIZE="max"
ENCODE_VALUE="1252"
USE_PROXY="true"
ADV_PROXY_DEFAULT="false"
PROXY_USER="none"
PROXY_HOST="PROXYHOST.mydomain.com"
PROXY_PORT="3128"
TEST_PROXY="nofips"
WEBUI_PORT="443"
WEBUI_REDIRECT_PORT="80"
```

This is an example of response file for an evaluation server installation with MS SQL as deployed database:

```
##BigFix GENERATED RESPONSE FILE
IS_EVALUATION="true"
LA_ACCEPT="true"
CREDENTIAL_USER_FIRSTNAME="john"
CREDENTIAL_USER_LASTNAME="smith"
CREDENTIAL_EMAIL="john.smith@domain.com"
CREDENTIAL_ORG="HCL"
```

```
SRV_DNS_NAME="DNSHOST.mydomain.com"
BES_SERVER_PORT="52311"
WR_WWW_PORT="8083"
WEBUI_PORT="443"
WEBUI_REDIRECT_PORT="80"
CONF_FIREWALL="no"
PVK_KEY_SIZE="max"
MSSQL_ADMIN_USER="sa"
MSSQL_ADMIN_PWD="password"
BES_LIC_FOLDER="/BigFix"
USE_PROXY="false"
ENCODE_VALUE="1252"
MSSQL_PORT="1433"
```

where:

Table 3. Response file keywords

Keyword	Values
LA_ACCEPT	Accepts the License Agreement: `true` to accept and continue `false` to exit the installation

IS_PREREQ_CHECK	Available values are:
	
	`true`
	`false`

IS_EVALUATION	Specifies the type of installation: `true` to run an evaluation installation `false` to run a production installation

Table 3. Response file keywords

(continued)

Keyword	Values
	Note: The evaluation installation does not support the enhanced security option. For more information about this feature see Security Configuration Scenarios.
CREDENTIAL_USER	Specifies the user name. An example is: `John Smith`.
	Note: Valid in the evaluation installation only
CREDENTIAL_USER_FIRSTNAME	Specifies the user first name. An example is: `John`.
	Note: Valid in the evaluation installation only
CREDENTIAL_USER_LASTNAME	Specifies the user last name. An example is: `Smith`.
	Note: Valid in the evaluation installation only
CREDENTIAL_EMAIL	Specifies the user email address. An example is: `john.smith@mycompany.com`.
	Note: Valid in the evaluation installation only
CREDENTIAL_ORG	Specifies the user's organization. An example is: `HCL US`.

Table 3. Response file keywords

(continued)

Keyword	Values
	Note: Valid in the evaluation installation only
COMPONENT_SRV	Specifies to install the BigFix server component: `true` to install the server and client `false` to not install the server and the client
COMPONENT_WR	Specifies to install the BigFix Web Reports component: `true` to install Web Reports `false` to not install Web Reports
COMPONENT_WEBUI	Specifies to install the BigFix WebUI component: `true` to install the WebUI `false` to not install the WebUI
SINGLE_DATABASE	Creates a master database for later replication or if you only need a single database in your deployment. `true` to create a single database `false` to create a replicated database
LOCAL_DATABASE	Uses a local or remote database: `true` to use a local database `false` to use a remote database through a DB2 client

Table 3. Response file keywords

(continued)

Keyword	Values
DB2_ADMIN_USER	Specifies the user name of the local DB2 Administrative user. Only if DB2 is already installed.
DB2_ADMIN_PWD	Specifies the password of the local DB2 Administrative user. Only if DB2 is already installed.
DB2INST_CONFIGURE	Configures the database during the BigFix installation: `yes` to configure the DB2 `no` to not configure the DB2 Only if DB2 is already installed.
MASTER_DNS	Used only if SINGLE_DATABASE = false. Specifies the hostname of the computer where the master server database is located.
DB2_MASTER_PORT	Used only if SINGLE_DATABASE = false. Specifies the database port number of the system where the master server database is located.
DB2_MASTER_USER	Used only if SINGLE_DATABASE = false. Specifies the username of the DB2 administrative user of the system where the master server database is located.
DB2_MASTER_PWD	Used only if SINGLE_DATABASE = false. Specifies the password of the DB2 administrative user of the system where the master server database is located.

Table 3. Response file keywords

(continued)

Keyword	Values
BES_WWW_FOLDER	Specifies the installation folder of the BigFix server. The default value is `/var/opt/BESServer`.
WR_WWW_FOLDER	Specifies the installation folder of Web Reports. The default value is `/var/opt/BESWebReports-Server`.
WR_WWW_PORT	Specifies the Web Reports port number. The default value is `8083` if you are installing BigFix Version 9.5.2 and the configuration is HTTPS. The default value is `8080` if you are installing BigFix Version 9.5 and the configuration is HTTP.
WR_USEROOT	Specifies if the Web Reports service runs as root: `true` to run the service as root `false` to run the service with an user different from root
WR_NONROOT_USER_NAME	Specifies the user with which the Web Reports service runs.
INSTALL_DB2	Installs DB2 together with the BigFix server: `yes` to install DB2 `no` to not install DB2
DB2_INSTANCE_NAME	Specifies the name of the BigFix database instance. The default value is `db2inst1`. **Note:** Starting from BigFix V9.5, you can install the product on a dedicated DB2 in-

Table 3. Response file keywords

(continued)

Keyword	Values
	stance with a name different from the DB2 user name. Ensure that the DB2 instance name used to install the BES Root Server cannot contain the following special characters: blanks, tabs `\t`, returns `\n` and `; & | " ' < >`
DB2_DAS_USERNAME	Specifies the username of the account under which the DB2 administration server (DAS) runs. The default value is `dasusr1`.
DB2_FENCED_USERNAME	Specifies the user name of the account used to run user defined functions (UDFs) and stored procedures outside of the address space used by the DB2 database. The default user is `db2fenc1`.
DB2_INSTALL_DIR	Specifies the directory where to install DB2. For example: `/opt/hcl/db2/V10.5`.
DB2_PORT	Specifies the DB2 port. The default value is `50000`.
DB2_HOSTNAME	Used only if LOCAL_DATABASE = false. Specifies the hostname of the computer where the DB2 database is installed.
DB2_CONN_USER	Used only if LOCAL_DATABASE = false. Specifies the username for the remote DB2 administrative user.
DB2_CONN_PWD	Used only if LOCAL_DATABASE = false. Specifies the password for the remote DB2 administrative user.

Table 3. Response file keywords

(continued)

Keyword	Values

BES_PREREQ_INSTALL	Available values are:
	`ignore`
	`install`
	`exit`

BES_PREREQ_REPO_ENABLE	Used if any prerequisite packages need some repositories to be enabled. Allowed values:
	`enable: the installation tries to enable the needed repositories`
	`exit: exit the installation`

BES_PREREQ_DB2_INSTALL	Available values are:
	`ignore`
	`install`
	`exit`

DB2_SETUP_FILE	Specifies the setup file to install DB2. For example: `../server_r/db2setup`.
DB2_EXTR_DIR	Specifies the location to extract the DB2 setup; used if the folder where the installation is started is not writable.
DB2_USERS_PWD	Specifies the DB2 user password.
MSSQL_ADMIN_USER	Specifies the username for the local SQL Server administrative user.
MSSQL_ADMIN_PWD	Specifies the password for the local SQL Server administrative user.

Table 3. Response file keywords

(continued)

Keyword	Values
MSSQL_PORT	Specifies the SQL Server port.
MSSQL_HOSTNAME	Used only if LOCAL_DATABASE = false. Specifies the SQL Server remote host.
MSSQL_CONN_USER	Used only if LOCAL_DATABASE = false. Specifies the username for the remote SQL Server administrative user.
MSSQL_CONN_PWD	Used only if LOCAL_DATABASE = false. Specifies the password for the remote SQL Server administrative user.
TEM_USER_NAME	Specifies the BigFix user ID to define the initial administrative user. The default value is `IEMAdmin`. **Note:** Valid in the production installation only. In the evaluation installation the default user is `EvaluationUser` and the password is the password of the DB2 instance user.
TEM_USER_PWD	Specifies the password to define the initial administrative user. **Note:** Valid in the production installation only. In the evaluation installation the default user is `EvaluationUser` and the password is the password of the DB2 instance user.

Table 3. Response file keywords

(continued)

Keyword	Values
CONF_FIREWALL	Configures the firewall to enable the BigFix server or relay to connect to the Internet: `yes` to set the firewall configuration `no` not to set the firewall configuration
BES_SETUP_TYPE	Specifies the type of setup to run: `authfile` to install with a BES license authorization file `prodlic` to install with a Production license that is already available `masthead` to install with an existing masthead
BES_AUTH_FILE	Specifies the path of the authorization file. An example of path is: `/opt/iemlic/LicenseAuthorization.BESLicenseAuthorization`.
SRV_DNS_NAME	Specify the DNS name or IP address of the machine on which to install the server. This name is saved in your license and will be used by clients to identify the BigFix server. It cannot be changed after a license is created.
BES_LICENSE_PVK_PWD	Specifies the password of the `license.pvk` file.
ENCODE_VALUE	Specifies the deployment encoding to use when communicating with the infrastructure. The default value is 1252.
PVK_KEY_SIZE	Specifies the size in bits of the public key (`license.crt`):

Table 3. Response file keywords

(continued)

Keyword	Values

	`min`
	Corresponds to 2048 bits.
	`max`
	Corresponds to 4096 bits. This is the default value.

BES_LIC_FOLDER	Specifies the License folder where the installation generates and saves `license.crt`, `license.pvk` and `masthead.afxm`. An example of License folder is `/tmp/ServerInstaller_9.5-rhel/offlic`.
SUBMIT_LIC_REQUEST	Submits the request to HCL for getting the license certificate: `yes` to submit a request from this machine over the Internet for a license certificate (`license.crt`) and saved in your credential folder. `no` to save the request to a file and manually submit it to HCL (http://support.bigfix.com/bes/forms/BESLicenseRequestHandler.html). This method might be necessary if your deployment is isolated from the public Internet.
CONTINUE_FROM_CRT	Used only if SUBMIT_LIC_REQUEST = no. Allowed values:

Table 3. Response file keywords

(continued)

Keyword	Values
	`yes: continue with the installation by importing the certificate (license.crt)` `no: exit the installation`
USE_PROXY	Specifies a proxy connection to enable the BigFix server to connect to the Internet during the installation: `true` to set the proxy. `false` to not set the proxy.
PROXY_USER	Specifies the user of the proxy. If the proxy does not require authentication, you must set `PROXY_USER` to `NONE`.
PROXY_PWD	Specifies the password of the proxy user.
PROXY_HOST	Specifies the hostname of the computer where the proxy is running.
PROXY_PORT	Specifies the port of the computer where the proxy is running.
ADV_PROXY_DEFAULT	Accepts the default proxy configuration settings: `true` to use the default values `false` to use custom values.
PROXY_METH	Restricts the set of authentication methods that can be used. You can specify more than one method separated by a comma. Available methods are:

Table 3. Response file keywords

(continued)

Keyword	Values

	`basic`
	`digest`
	`negotiate`
	`ntlm`
	
	By default the proxy chooses the authentication method to use.

PROXY_EXLIST	Specifies a comma-separated list of computers, domains, and subnetworks that must be reached without passing through the proxy. For information about the syntax to use, see Setting a proxy connection on the server *(on page 552)*.
PROXY_SECTUNNEL	Specifies whether or not the proxy is enforced to attempt tunneling. Available values are: `true` to enable proxy tunneling. `false` to not enable proxy tunneling.

PROXY_DOWN	Specifies if all HTTP communications in your BigFix environment, including downstream communications, pass through the proxy. Available values are:
	
	`true`
	`false`

TEST_PROXY	Specifies if and how the connection to the proxy must tested. This is an optional step. Available values are:

Table 3. Response file keywords

(continued)

Keyword	Values
	`nofips` to test the connection without using FIPS. `fips` to test the connection using FIPS. `no` to not test the connection.
BES_MASTHEAD_FILE	Specifies the path to the masthead file.
BES_CERT_FILE	Specifies the path to the license certification file.
BES_LICENSE_PVK	Specifies the path to the private key file.
ADV_MASTHEAD_DEFAULT	Specifies whether or not to accepts the default masthead settings. Available values are: `true` to use the default values `false` to use custom values.
BES_SERVER_PORT	Specifies the number of the server port. The default value is: 52311.
ENABLE_FIPS	Specifies whether or not to enable FIPS 140-2 compliant cryptography. Available values are: `true` to enable FIPS. `false` to not enable FIPS.
BES_GATHER_INTERVAL	Specifies how long the clients wait without hearing from the server before they check whether new content is available. Available values are: `0` for fifteen minutes. `1` for half an hour. `2` for one hour. `3` for eight hours.

Table 3. Response file keywords

(continued)

Keyword	Values
	`4` for half a day. `5` for one day. `6` for two days. `7` for one week. `8` for two weeks. `9` for one month. `10` for two months.

INITIAL_LOCK	Specifies the initial lock state of all clients 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. Available values are:
	`0`
	`1`
	`2`

LOCK_CONTROLLER	Specifies who can change the action lock state. Available values are: `0` to allow any Console operator with management rights to change the lock state of any client in the network. This is the default value. `1` to delegate control over locking to the end user. `2`
LOCK_DURATION	Specifies the number of minutes that the clients must be locked.

Table 3. Response file keywords

(continued)

Keyword	Values

ENABLE_LOCK_EXEMPT	Specifies if specific URLs must be exempted from locking actions. Available values are:
	
	`true`
	`false`

EXCEPTION_URL	Specifies the URL to except from locking actions. Use the following format `'http://domain'`.
ENABLE_ARCHIVE_UTF8	Specifies the codepage used to write filenames in the BigFix archives. Available values are: `true` to write filenames UTF-8 codepage. `false` to not write filenames UTF-8 code-page.
IS_SILENT	Forces the installation to end with a message if a required parameter is missing: `true` to force the installation to end if a re-quired parameter is missing. `false` to prompt the user for the missing pa-rameter. If a parameter is missing the installation variable associated with the missing parameter is reported in the error message.
WEBUI_DNS_NAME	Used if the WebUI is not installed on the same com-puter where the server is installed. Specifies the hostname used by the server to identify the WebUI.

Table 3. Response file keywords

(continued)

Keyword	Values
WEBUI_PORT	Specifies the WebUI port number. The default value is 443.

WEBUI_REDIRECT	Specifies to enable the HTTP redirect on the Web-UI. Allowed values:
	`true`
	`false`
	If false, the redirect is disabled and the WebUI does not respond to the HTTP request.

WEBUI_REDIRECT_PORT	Specifies the WebUI redirect port number. The default value is 80.
WEBUI_SRC_CERT_FOLDER	Used if the WebUI is not installed on the same computer where the server is installed. Specifies the folder containing the WebUI credentials; the Web-UI needs a set of certificates as credentials to authenticate itself to the BigFix server. These certificates must be generated on the server by running the BESAdmin command -createwebuicredentials and the folder containing them must be copied to the computer where the WebUI will be installed.
DB2_WEBUI_HOSTNAME	Specifies the DB2 hostname for WebUI component.
DB2_WEBUI_PORT	Specifies the DB2 port for WebUI component.
DB2_WEBUI_ADMIN_USER	Specifies the DB2 administrative user for WebUI component.

Table 3. Response file keywords

(continued)

Keyword	Values
DB2_WEBUI_PWD	Specifies the DB2 administrative password for WebUI component.
MSSQL_WEBUI_HOSTNAME	Specifies the SQL Server hostname for WebUI component.
MSSQL_WEBUI_PORT	Specifies the SQL Server port for WebUI component.
MSSQL_WEBUI_ADMIN_USER	Specifies the SQL Server administrative user for WebUI component.
MSSQL_WEBUI_PWD	Specifies the SQL Server administrative password for WebUI component.

IGNORE_WARNING	Specifies to continue the installation if some prerequisite checks failed. Allowed values:
	`yes: continue the installation`
	`no: exit the installation`

IGNORE_SPACE_WARNING	Specifies to continue the installation if one or more of the filesystems involved in the product installation does not have the required free space. Refer to the installation log file '$1' for additional details. Allowed values:
	`yes: continue the installation`
	`no: exit the installation`

IGNORE_REPL_CRED_WARNING	Used if a master server database is configured. Specifies to continue the installation if the specified credentials for the DB2 administrative user and

Table 3. Response file keywords

(continued)

Keyword	Values

	the DB2 port number do not match those specified on the master server. Proceeding with the installation you can have only two servers of your deployment configured in replication mode. Allowed values:
	`yes: continue the installation`
	`no: exit the installation`

IGNORE_DB_TYPE_MISMATCH_WARNING	Specifies to continue the upgrade BigFix server with a package using a database type different from the installed one. Allowed values:
	`yes: continue the installation`
	`no: exit the installation`

IGNORE_DB2_WARNING	Specifies to continue the installation if the installed version of DB2 does not match the minimum version requirement. Allowed values:
	`yes: continue the installation`
	`no: exit the installation`

IGNORE_PREVIOUS_OS_VERSION_FOR_MSSQL	Specifies to continue the installation of the package using SQL Server on a computer with operating system version older than 9.0. Allowed values:
	`yes: continue the installation`
	`no: exit the installation`

Installing the Client on Linux

For more details about how to install the Clients, see section Installing the clients (on page 283).

Installing the Console

You can install the BigFix console on any Windows computer that can make a network connection via HTTPS port 52311 to the Server.

Except in testing or evaluation environments, it is not recommended to run the Console on the Server computer due to the performance and security implications of having the publisher key credentials on a computer that is running a database or web server. Using the BigFix console you can monitor and fix problems on all managed computers across the network.

To install the console, follow these steps:

1. Go to `/var/opt/BESInstallers` directory.
2. Copy the `Console` folder to a Windows workstation. Use the `Console` folder of the same build level.
3. From the `Console` directory on the Windows workstation run: `setup.exe`



Note: By default the local operating system firewall is enabled. To allow the Console to connect to the BigFix Server, ensure that the firewall is configured to allow tcp and udp communications through the Server port (default 52311) and tcp communications through Web Reports Ports (default 80).

If you need to manually configure the local firewall you can run the following commands:

```
iptables -I INPUT -p tcp --dport < Server_Port > -j ACCEPT
iptables -I INPUT -p udp --dport < Server_Port > -j ACCEPT
iptables -I INPUT -p tcp --dport < WebReports_Port > -j ACCEPT
service iptables save
```

For more details about using the Console program see the BigFix Console Users Guide .

Installation Folder Structure

After the BigFix installation, you can see the following folder structure.

Server Folder Structure:

```
/var/opt/BESInstallers
/var/opt/BESInstallers/Client (Client installer)
/var/opt/BESInstallers/Console (Console installer)

/var/opt/BESServer
  besserver.config (Configuration file)
  besserver.config.default (Default configuration file)

/var/opt/BESServer/FillDBData/FillDB.log (FillDB service log)

/var/opt/BESServer/GatherDBData/GatherDB.log (GatherDB service log)

/opt/BESServer
/opt/BESServer/bin (Server binaries)
/opt/BESServer/reference (Rest API xsd templates)

/etc/opt/BESServer
  actionsite.afxm (Masthead file)

/etc/init.d
or /usr/lib/systemd/system (from Version 11.0.2 or later, but only for
 RHEL, SLES and derived platforms.
For more details, see Managing the BigFix Services (on page 273))
  besserver (Server service)
  besfilldb (FillDB service)
  besgatherdb (GatherDB service)
```

If you want to move the content of the directories:

```
/var/opt/component_folder
/opt/component_folder
```

For example the `/var/opt/BESServer` directory, to a new location, you can use the UNIX symbolic link feature to point to the new directory.

Web Reports Folder Structure:

```
/var/opt/BESWebReportsServer
  beswebreports.config (Configuration file)
  beswebreports.config.default (Default configuration file)

/opt/BESWebReportsServer
/opt/BESWebReportsServer/bin (WebReports binaries)

/etc/opt/BESWebReportsServer
 actionsite.afxm (Masthead file)

/etc/init.d
or /usr/lib/systemd/system (from Version 11.0.2 or later, but only for
 RHEL, SLES and derived platforms.
For more details, see Managing the BigFix Services (on page 273))
  beswebreports (WebReports service)
```

If you want to move the content of the directories:

```
/var/opt/component_folder
/opt/component_folder
```

For example the `/var/opt/BESWebReportsServer` directory, to a new location, you can use the UNIX symbolic link feature to point to the new directory.

Client Folder Structure:

```
/var/opt/BESClient
   besclient.config (Configuration file)
```

```
   besclient.config.default (Default configuration file)

/opt/BESClient
/opt/BESClient/bin (Client binaries)

/etc/opt/BESClient
  actionsite.afxm (Masthead file)

/etc/init.d
or /usr/lib/systemd/system (from Version 11.0.2 or later, but only for
 RHEL, SLES and derived platforms.
For more details, see Managing the BigFix Services (on page 273))
   besclient  (besclient service)
```

If you want to move the content of the directories:

```
/var/opt/component_folder
/opt/component_folder
```

For example the `/var/opt/BESClient` directory, to a new location, you can use the UNIX symbolic link feature to point to the new directory.

Install Log Files:

```
/var/log/
   BESInstall.log       (Installer log file)
   BESAdminDebugOut.txt (Administrator Tool debug information)
   BESRelay.log         (Relay log file)
```

Be aware that if one of the following folders does not exist, the installation procedure fails:

```
/opt
 /etc
 /var
```

Common files:

```
/var/opt/BESCommon
```

This folder contains files which are relevant to the identification of BigFix components. You must not modify or remove them.

Managing the BigFix Services

Procedure to manage the services.

You can start, stop, restart, or query the status of Linux BigFix services using the `service` command, when available, as shown below:

```
service service stop
service service start
service service restart
service service status
```

Alternatively, for the RHEL, SUSE and derived distributions listed below:

1. Red Hat Enterprise Linux (RHEL)
2. CentOS
3. Oracle Linux
4. Amazon Linux
5. Rocky Linux
6. SUSE
7. OpenSUSE Leap

It is possible to use the following commands:

```
systemctl stop service
systemctl start service
systemctl restart service
systemctl status service
```

For the others Linux distributions supported by BigFix Platform, it is possible to use the following commands:

```
/etc/init.d/service stop
/etc/init.d/service start
/etc/init.d/service restart
/etc/init.d/service status
```

Note: Up to BigFix Version 11.0.1, for backward compatibility reasons, it is possible to use the commands above also to manage the services of RHEL, SUSE and derived distributions listed above.

where *service* can be one of the following services:

```
besclient
besexplorer
besfilldb
besgatherdb
bespluginportal
besrelay
besserver
beswebreports
beswebui
```

Note: Ensure you do not use the systemctl command to manage a service, except on RHEL, SUSE and derived distributions.

If you installed the BES Server Plugin Service (MFS), use these commands to start and stop it, up to BigFix Version 11.0.2:

```
service mfs start
service mfs stop
```

or

```
/etc/init.d/mfs start
/etc/init.d/mfs stop
```

while you can use these commands to start and stop it, from BigFix Version 11.0.3 or later.

```
systemctl start mfs
systemctl stop mfs
```

Changing the DB2 port

After you install the DB2 database of the BigFix server, you can change the DB2 instance connection port and set it in the BigFix configuration files.

Perform the following steps:

1. Stop all the BigFix services and all applications connected to the DB2 instance.
2. Change the DB2 connection port:

   ```
   #su - db2inst1
   $db2 update dbm cfg using SVCENAME <new_port_number>
   $db2stop; db2start
   ```

3. Open the configuration file: `/var/opt/BESServer/besserver.config`
4. Go to `[Software\BigFix\EnterpriseClient\Settings\Client \_BESServer_Database_Port]` and set the new port number as follows:

   ```
   value  = "<new_port_number>"
   ```

5. Open the configuration file: `/var/opt/BESWebReportsServer/beswebreports.config`
6. Go to `[Software\BigFix\Enterprise Server\FillAggregateDB]` and set the new port number as follows:

   ```
   Port  = "<new_port_number>"
   ```

7. Start all the BigFix services.

Removing the BigFix components from Linux

If you have one or more BigFix components installed on a computer, you can remove all of them at once, or one at a time.

To uninstall one or more BigFix components installed on a local Linux system, run the following steps:

1. Look for the installed BigFix RPM packages by entering the following command:

   ```
   rpm -qa | grep BES
   ```

2. Remove the Server, the WebUI, the Client, and Web Reports RPM files:

   ```
   rpm -e BESWebUI
   rpm -e BESWebReportsServer
   rpm -e BESRootServer
   rpm -e BESRelay
   rpm -e BESClientDeployTool
   rpm -e BESAgent
   ```

   Note: You cannot remove BESAgent until you remove all components depending on it (BESRootServer, BESWebUI, and so on).

3. Remove the following files and folders:

   ```
   /etc/opt/BES*
   /opt/BES*
   /tmp/BES
   /var/log/BES*
   /var/opt/BES*
   ```

   Where `BES*` is a prefix followed by the name of a BigFix component, for example "BESClient".

Note: If you do not plan to remove all BigFix components, keep the folder `/var/opt/BESCommon`.

4. If your database engine is DB2, perform this step. Otherwise, skip to step 5.

   If your `BFENT` and `BESREPOR` databases are local, you can run these commands:

   ```
   su - db2inst1
   db2 drop db BFENT
   db2 drop db BESREPOR
   ```

   If your `BFENT` and `BESREPOR` databases are remote, you can run these commands:

   ```
   su - db2inst1
   db2 attach to TEM_REM user <UserName> using <Password>
   db2 drop db BFENT
   db2 drop db BESREPOR
   db2 detach
   db2 uncatalog node TEM_REM
   ```

5. If your database engine is SQL Server, perform this step.

   Note: This step only applies if you are using an on-premise SQL Server (local or remote). In case your databases are on Cloud, follow the specific Cloud provider guidelines to remove your databases.

   Remove the `BFEnterprise` and `BESReporting` databases. For example, you can use the sqlcmd and run these commands:

   ```
   sqlcmd -U sa -S localhost,1433
   USE master;
   ALTER DATABASE BFEnterprise SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
   DROP DATABASE BFEnterprise;
   GO
   ALTER DATABASE BESReporting SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
   ```

```
DROP DATABASE BESReporting;
GO
```

To remove a local SQL Server and all its databases, you can run this command:

```
sudo yum remove mssql-server
```

DSA on Linux

Note: DSA is available for installations with DB2 only,

Installing Additional Linux Servers (DSA)

For each additional server that you want to add to your deployment, ensure that they are communicating with each other, and then perform the following steps.

1. Download the BigFix Server installer having the same version as the one installed on the master server. Ensure that each server uses the same DB2 version.
2. Copy the `license.pvk` and `masthead.afxm` files from the master server to each computer where you intend to install an additional DSA Server.
3. Each DSA Server must have **its own** DB2 database engine, either local or remote. Do not use the same database engine to store the databases of two different DSA servers. Each DSA Server must be able to access its own database engine and also the database engines of the other DSA Servers. Use the same user name and password to access all your database engines.
4. Run the `install.sh` script on each computer that you want to configure as an additional Server.
5. On the `Select Install Type` prompt, choose:

```
[2] Production: Install using a production license or an authorization
 from
a production license
```

6. On the `Select the HCL BigFix Features you want to install` prompt, choose a combination of components that includes the BigFix Server. Do not install the WebUI component on the secondary DSA servers.
7. On the `Select Database Replication` prompt, choose:

   ```
   [2] Replicated Database.
   ```

8. On the `Select Database` prompt, choose `[1] Use Local Database` (typical for most applications).

   Note: You can also select a remote database hosted on a different computer. In this case ensure that the computer you are installing BigFix on can resolve the hostname of the remote server where the database resides.

9. On the `DB2 Local Administrative User` prompt, assuming you chose `Use Local Database` earlier, enter the user name and password of the DB2 administrative user for the database on the computer where the installation script is running.
10. Enter the Web Server Root folder path.
11. If you chose to install the Web Reports component, enter the requested information.
12. Specify the location of `license.pvk` and its password.
13. Specify the location of the existing `masthead.afxm` file that was generated when installing the master server.
14. On the `Secondary Server DNS Name` prompt, enter the DNS name of the new server. This name must be resolvable by other servers and by clients.
15. On the `DB2 Connection` prompt, enter the port number of the local DB2 instance.
16. Enter the information to allow the new server to connect to the DB2 instance of the master server:

    On the `Master Server Database Hostname` prompt, specify the hostname of the master server database host.

    On the `Master Server Database Port` prompt, specify the database port number of the master server database host.

    On the `Master Server Database Administrative User` prompt, specify the user name of the DB2 administrative user of the master server database host.

On the `Master Server Database Administrative User Password` prompt, specify the password of the DB2 administrative user of the master server database host.

17. When installing a new DSA server in an environment where the custom CA is already present, you must also install the custom CA (using the `setcustomca` BESAdmin command) there for the new server to work.
18. On the master server, run the resign security data command by using the BigFix Administration tool.

    ```
    ./BESAdmin.sh -resignsecuritydata
    ```

    For additional information, see BESAdmin Linux Command Line (on page 422).
19. Verify that the other servers have been replicated.

Authenticating Additional Servers (DSA)

Multiple servers can provide a higher level of service for your BigFix installation.

If you choose to add Disaster Server Architecture (DSA) to your installation, you will be able to recover from network and systems failures automatically while continuing to provide local service. To take advantage of this function, you must have one or more additional servers with a capability at least equal to your primary server. Because of the extra expense and installation involved, you should carefully think through your needs before committing to using DSA.

Your servers can communicate with each other using the DB2 inter-server authentication option.

Before installing the additional Linux Servers, install the DB2 server on each machine that you want to add to your deployment. The version of the DB2 server must be the same as the DB2 server installed on the Master Server.

Using DB2 Authentication

With this technique, each Server is given a login name and password, and is configured to accept the login names and passwords of all other Servers in the deployment.

The password for this account typed in clear text is obfuscated in the configuration file on each server, after the restart of the FillDB service. To authenticate your servers using DB2 Authentication, follow these steps:

1. Choose a single login name (for example, `db2inst1`), and a single password to be used by all servers in your deployment for inter-server authentication.
2. On the Master Server, open the `/var/opt/BESServer/besserver.config` file.
3. Add or modify the following keywords in the `[Software\BigFix\Enterprise Server \FillDB]` section:

```
ReplicationUser = <login name>
ReplicationPassword = <password>
ReplicationPort = <DB2_port>
ReplicationDatabase = BFENT
```

4. Restart the `FillDB` service.

Note:

This choice must be made on a deployment-wide basis; you cannot mix domain-authenticated servers with DB2-authenticated servers. `ReplicationUser`, `ReplicationPassword`, and `ReplicationPort` must be uniquely defined in all the server configuration files of your DSA environment. All BigFix servers in your deployment must be running the same version of DB2 server.

Uninstalling a Linux replication server

To uninstall a replication server, call the database-stored procedure `delete_replication_server`, which removes the specified ID from the replication set.

Ensure you specify the identifier of the server to delete.

If you are using the DB2 database, log in and run the following procedure:

```
call dbo.delete_replication_server(n)
```

where *n* is the identifier of the server to delete.

If you are using the Microsoft SQL Server database, log in and run the following procedure:

```
exec BFEnterprise.dbo.delete_replication_server n
```

where *n* is the identifier of the server to delete.

Chapter 10. Installing the clients

Install the BigFix client on every computer in your network that you want to administer, including the computer that is running the console.

This allows that computer to receive important Fixlet messages such as security patches, configuration files, or upgrades.

Using the Client Deploy Tool

You can use the Client Deploy Tool (CDT) to install Windows, UNIX and Mac target computers.

The Client Deploy Tool helps you roll out targets in an easy way, but there are some requirements and conditions:

- Depending on whether you are using the Client Deploy Tool on a Windows or on a Linux system, you can use the tool to install different platforms:

  CDT on Windows

  Installs Windows, UNIX and Mac target computers.

  CDT on Linux

  Installs UNIX and Mac target computers.

Target prerequisites

To successfully deploy target computers from the Client Deploy Tool, ensure that you satisfy the following prerequisites, depending on the target operating system.

Prerequisites needed for the **UNIX/MAC** target computers:

- The bash shell must be installed.

  For AIX and Solaris target computers, which do not have a bash shell installed by default, the Korn shell can be used.
- The SCP and SSH protocols must be enabled. The SSH protocol must be enabled on port 22.

- The root user must exist, or any other user with SUDO privileges enabled.
- The user with SUDO privileges must be configured as not requiring TTY.
- The user, configured to access the target computer by using the SSH key authentication, must be one of the following:
    - root
    - a user configured to run SUDO without a password.
- The mount permissions for “/var” must not be configured with the "noexec" flag enabled.



Note: You can deploy on Windows target computers only if you are using the Client Deploy Tool on a Windows system.

Prerequisites needed for the **Windows** target computers:

- From the Control Panel, go to **Network and Internet > Network and Sharing Center > Change advanced sharing settings** and in the "current profile" section, select the "Turn on file and printer sharing" option.
- Launch services.msc and ensure that the "Remote Registry" service is not disabled. It is sufficient to have it in Manual mode, the operating system will start it when needed.
- Restart the workstation for the changes to take effect, if required.
- Ensure that the administrative shares are not explicitly disabled. Locate the registry values:

  ```
  [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
  LanmanServer\Parameters]
  ```

  "AutoShareServer" should be 1 (explicitly enabled) or absent.

  "AutoShareWks" should be 1 (explicitly enabled) or absent.
- Ensure that the firewall is not blocking the "File and Printer Sharing" service on port 445.

  From the Control Panel, go to **System and Security > Windows Firewall > Advanced settings**.

In the "Windows Firewall with Advanced Security" panel, perform the following checks:

- Inbound Rules: The service "File and Printer Sharing (SMB-In)" is allowed to access the local port 445.
- Outbound Rules: The service "File and Printer Sharing (SMB-Out)" is allowed to access the remote port 445.

- The remote computers you want to deploy to must be reachable using the Windows Remote Procedure Call (RPC) protocols.



Note: The Client Deploy Tool will not work if there is a firewall blocking traffic between you and the remote computer or if the remote computer has a personal firewall blocking traffic. By default, RPC uses port 135 as well as a random port above 1024. If you are using a firewall, you might want to configure the RPC port to a specific port number so that you can lock it down and allow traffic across that port without opening the firewall completely (see:http://support.microsoft.com/kb/154596). RPC can use TCP or UDP ports so you should allow for both. The Client Deploy Tool itself does not make use of any other ports beyond what RPC utilizes. After the client is installed, it will use whichever port you have specified for your license (TCP/UDP 52311 by default).

You cannot have any network or security policies in place that might prevent the application from connecting to the remote computer and running a service that uses the domain administrator credentials to copy files from a shared location and run them locally on the computer.

Client Deploy Tool wizard

The Client Deploy Tool wizard was introduced with BigFix Version 9.5 Patch 7.

Recommended scenario

You can use the Client Deploy Tool (CDT) to install Windows, UNIX and Mac target computers.

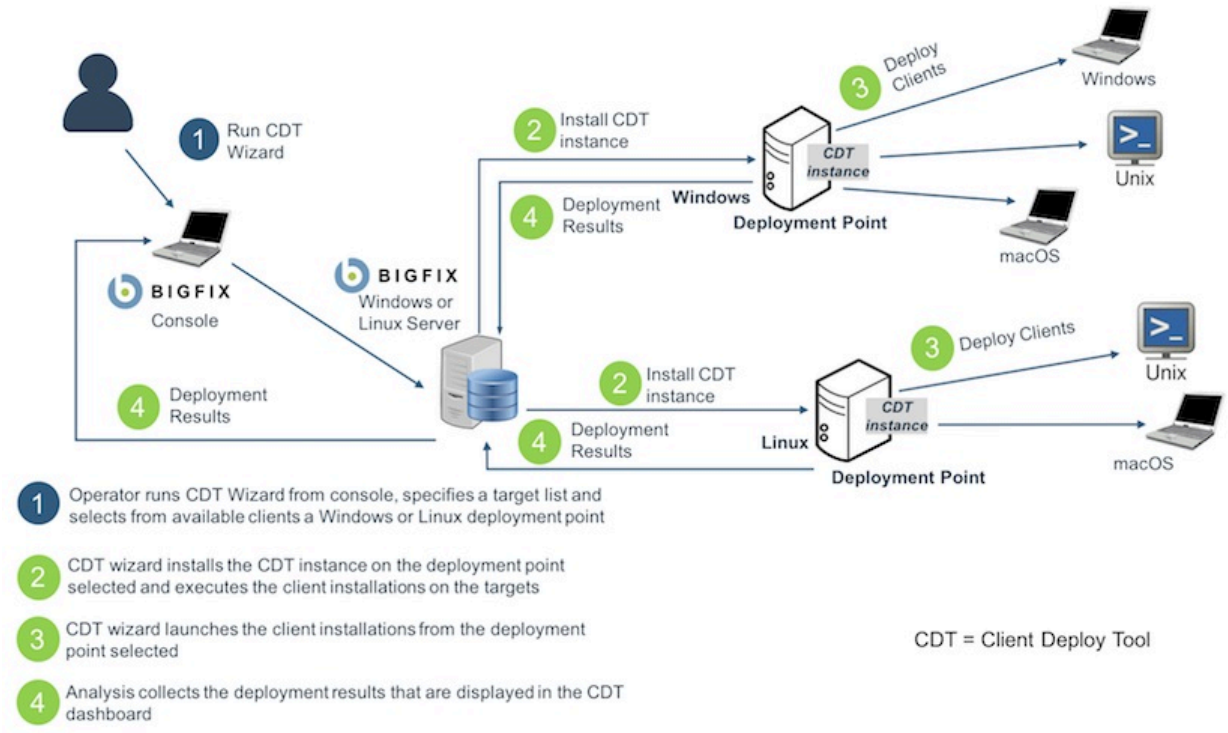
Client Deploy Tool: Recommended scenario

This scenario describes the recommended steps to install and use the Client Deploy Tool in your environment. The scenario is supported both if you have a BigFix Windows or Linux server.

From the BigFix console, perform these steps:

1. Install the Client Deploy Tool on the target computers of your environment and deploy the client computers by running the Client Deploy Tool wizard. For details about this operation, see Deploying clients from the console *(on page 287)*.
2. After using the Client Deploy Tool wizard, you can view the deployment results in the Client Deploy Tool dashboard. For details about this operation, see Viewing the deployment results in the dashboard *(on page 291)*.

The following graphic shows the details of this process. Only the first step (in blue) is manually run by an operator, while the other three steps (in green) are performed automatically when running the Client Deploy Tool wizard.



Deploying clients from the console

How to deploy clients using the **Client Deploy Tool Wizard**.

Prerequisites:

Before deploying the BigFix clients from the **Client Deploy Tool Wizard**, ensure that you locate and activate globally the following analyses:

Table 4. Analyses to activate globally

ID	Name
204	BES Component Versions
2814	Collect BES Client Deploy Tool Reports

From the BigFix console, click **Wizards > All Wizards > Client Deploy Tool Wizard**.

The **Client Deploy Tool Wizard** will guide you through the deployment of the BigFix clients.

You can also launch the same wizard from the **Unmanaged Assets** view of the console by performing these steps:

1. Select one or more assets in the view.
2. Right-click them. A menu opens.
3. Click **Install BigFix Client**.

Note:

The displayed Client Deploy Tool Wizard already contains the following prefilled information:

- The names of the assets selected in step 1.
- The selection of the operating systems installed on the assets selected in step 1, found by the Nmap scan run on the assets.

In the **Set Target Credentials** page of the wizard:

1. Click **Add Targets**.
2. Enter the host name or the IP address of the target computers. You can enter the computer credentials now or specify them later.
3. If you want to specify the credentials of target computers you already added, or change them, select those computers from the list. Select the target computers displayed on the list.
4. Click **Set Credentials**.
5. Enter the user name and password needed to access the target computers.
6. As an alternative, you can select the **Use Key File** check box to use a private key file with the SSH authentication method, instead of using a password. You can also specify a passphrase if it was specified when generating the private key. Only ASCII characters are supported. Use a private key file in PEM or OpenSSH format.



Note: The SSH authentication method is not supported to install Windows target clients.



Note:

If you use the SSH key authentication, specify a user that is one of the following:

- root
- a user configured to run SUDO without a password.



Note: If you enabled FIPS mode and you use SSH key authentication with passphrase to install the clients, ensure that your private key is encrypted using a FIPS compliant algorithm.



Note: The SSH key types supported to access the target computers through SSH key authentication are RSA, ECDSA and ED25519.

7. **Note:** By clicking the small magnifying glass icons in the table header, you display a search box, useful to easily locate specific devices if you have a very long list of targets. By clicking a column header, you sort the rows by the values of that column.

8. Click Next to proceed to the **Select Deployment Point** page.

In the **Select Deployment Point** page of the wizard:

1. Select the target computers displayed on the list.
2. Click **Select Deployment Point**.
3. Select a deployment point from the list.

   Note:

   The deployment points displayed by this list are a set of:

   - All computers on which the Client Deploy Tool can be installed, and which are relevant for the Fixlet "Install/Update BigFix Client Deploy Tool Version 11". Remove 'yes' from the 'Is Preferred' filter to have the list of all computer selections available.
   - All computers on which a standalone Client Deploy Tool (any Version) is installed. If an older version is found, the Client Deploy Tool will be upgraded to Version 11.
   - All computers with a BigFix Console Version 11.

4. Click Next to proceed to the **Set Deployment Point Credentials** page. This page is skipped if we are installing the client on UNIX targets **only**.

In the **Set Deployment Point Credentials** page of the wizard:

1. Select a deployment point displayed on the list.
2. Click **Set Credentials**.
3. Enter the credentials needed to access the deployment point.
4. Click Next to proceed to the **Set Advanced Settings** page.

In the **Set Advanced Settings** page of the wizard:

1. Select the operating systems to be deployed, if not already prefilled by the results of the Nmap scan run on the assets.
2. Select the client version to be installed on your target computers.
3. Select the **Show advanced settings** check box.

   (Optional) In the "Custom Settings" section, add a list of custom client settings to apply to each client that will be deployed by the Client Deploy Tool.

   You can input these settings in either a table or a text box. If you use the table, add a row for each setting, then enter its name and value in the respective columns. If you use the text box, add a new line for each setting, then enter its name and value in the format "name=value".

   (Optional) In the "Proxy Settings" section, if the clients to be deployed must communicate through a proxy, enter the following information:

   Address

   The host name that is used to reach the proxy.

   Port

   The port that is used to communicate with the proxy.

   Username

   The user name that is used to authenticate with the proxy if the proxy requires authentication.

   Password

   The password that is used to authenticate with the proxy if the proxy requires authentication.

   (Optional) There is also a section where you can enter a custom installation path for the Windows target computers.
4. Click Next to proceed to the **Deploy Clients** page.

In the **Deploy Clients** page of the wizard:

1. Review the following summary information displayed by the wizard:
    - The list of all target computers on which the BigFix client will be installed. The user names needed to access the target computers. The deployment point that will be used for each target computer.
    - If new deployment points were specified, a list of the deployment points on which the Client Deploy Tool will be installed.
    - On each deployment point, the BigFix client packages that will be downloaded, if not already present.
2. If the summary information is correct, click **Deploy**.

After clicking Deploy, the **Client Deploy Tool Dashboard** displays the details of your BigFix client deployments. For more information about the dashboard, see Viewing the deployment results in the dashboard (on page 291).

Viewing the deployment results in the dashboard

How to view the deployment results.

Prerequisites:

Before viewing the deployment results in the dashboard, ensure that you locate and activate globally the following analyses:

Table 5. Analyses to activate globally

ID	Name
204	BES Component Versions
2814	Collect BES Client Deploy Tool Reports

Moreover, ensure that the clients and server/console clocks are synchronized to make the dashboard work properly.

From the BigFix console, click **Dashboards > All Dashboards > Client Deploy Tool Dashboard**.

The **Client Deploy Tool Dashboard** shows you the details of your BigFix client deployments.

The Dashboard displays the following information about the client deployments:

- The host name or IP address of the target computer on which you deployed the client.
- The operating system installed on the target computer.
- The deployment point used.
- The deployment status.
- In case of a failed deployment, the installation error message.
- The date and time on which the deployment occurred.

Note: By clicking the small magnifying glass icons in the table header, you display a search box, useful to easily locate specific devices if you have a very long list of targets. By clicking a column header, you sort the rows by the values of that column.

By clicking **Deploy BigFix Clients**, you open the **Client Deploy Tool Wizard**.

For more information about the Wizard, see Deploying clients from the console (on page 287).

By clicking **Upload Deploy Logs**, you upload the Client Deploy Tool target log files to the BigFix server by running a Fixlet.

For more information about the Fixlet, see Uploading the target logs to the server (on page 307).

Client Deploy Tool Fixlet

The Client Deploy Tool Fixlets were introduced with BigFix Version 9.5 Patch 5.

Installing the Client Deploy Tool from the console

As a prerequisite to install the Client Deploy Tool, ensure that you have a BigFix client installed on the target computers. The Fixlet is relevant if the BigFix client is installed, while the console is not. The Client Deploy Tool and the console are mutually exclusive. If the console is installed, the Client Deploy Tool is already installed under the same console directory.

To install the Client Deploy Tool from the BigFix console by running a Fixlet, perform the following steps:

1. Log in to the BigFix console.
2. Open the **Fixlets and Tasks** icon in the Domain Panel.
3. In the search bar, enter Client Deploy Tool.
4. Select the Fixlet named **Install/Update BigFix Client Deploy Tool (Version 11)**.
5. Click **Take Action**.
6. Select the target computers on which you want to perform the installation. Typically, a BigFix server or one or more relays.
7. Click **OK**. Verify the status of the Fixlet.

After performing the Client Deploy Tool installation, you can use the tool to deploy the target computers as described in Deploying clients by using a Fixlet (on page 293).

By running the Fixlet named **Uninstall Client Deploy Tool**, you can remove the Client Deploy Tool installations.

Deploying clients by using a Fixlet

Deploy client computers from the BigFix console by running the **Install BigFix Clients with Client Deploy Tool** Fixlet. This Fixlet cannot be imported into a custom site. You must use it from the BES Support site.

To deploy the clients from the console, perform the following steps:

1. Log in to the BigFix console.
2. Open the **Fixlets and Tasks** icon in the Domain Panel.
3. In the search bar, enter Client Deploy Tool.
4. Select the Fixlet named **Install BigFix Clients with Client Deploy Tool**.
5. In the **Description** tab, you must:
    - Select all the operating systems associated to the client computers that you want to deploy. Select the client version that you want to deploy.
    - Enter the credentials needed to access the computer on which you installed the Client Deploy Tool. These credentials are required only if you installed the

Client Deploy Tool on a Windows system. Specify either a domain administrator account with all necessary permissions or any administrator account with full local administrative permissions on the Client Deploy Tool computer.

- **Note:** In the "Targets" section, specify the computers to which you want to deploy the BigFix Client and the credentials to access them and run the installation. If a set of computers can be accessed with the same credentials, you can enter it as a single group of targets.

  Computers

  In this text box, specify one or more computers with the same credentials, using one of the following formats:
  - A list of hostnames, each on a new line.
  - A list of IP addresses, each on a new line.
  - An IP address range.

  If you specify a list, place each hostname or IP on its own line in the text box. To create a new line in the text box, press Enter. If you specify a range of IP addresses, enter it in the following format:

  ```
  192.0.2.1-20
  ```

  Username

  The user ID needed to access the target computers. You need to specify a user with sufficient permissions to install the BigFix Client. For example, Administrator on Windows or root on Linux.

  Password

  The password associated to the user ID of the target computers.
- Select the **Show advanced settings** check box.

(Optional) In the "Custom Settings" section, add a list of custom client settings to apply to each client that will be deployed by the Client Deploy Tool.

You can input these settings in either a table or a text box. If you use the table, add a row for each setting, then enter its name and value in the respective

columns. If you use the text box, add a new line for each setting, then enter its name and value in the format "name=value".

(Optional) In the "Proxy Settings" section, if the clients to be deployed must communicate through a proxy, enter the following information:

Address

The host name that is used to reach the proxy.

Port

The port that is used to communicate with the proxy.

Username

The user name that is used to authenticate with the proxy if the proxy requires authentication.

Password

The password that is used to authenticate with the proxy if the proxy requires authentication.

6. Click **Take Action**.
7. On the **Target** tab, select one or more devices, which are target computers on which you installed the Client Deploy Tool instances.
8. Click **OK**.
9. Verify the status and the exit code of the Fixlet. The exit code 0 represents the Success status.

Note: When running the Fixlet, depending on the operating systems that you selected in the **Description** tab of the Fixlet, the platform-specific packages are cached on the BigFix server and downloaded only on the client computers where the Client Deploy Tool is installed. Packages are downloaded only if they are not found; they are not overwritten each time you run the Fixlet.

Note:

If you deploy old BigFix client versions, such as 9.1 and 9.2, or any other version older than the Client Deploy Tool version installed using the **Install BigFix Clients with Client Deploy Tool** Fixlet, the locally installed Client Deploy Tool user interface might no longer work.

This known issue is caused by the `BESClientsCatalog.xml` file level that is replaced with the version deployed by the Fixlet. To solve this known issue, manually run again the Fixlet to deploy the latest client version and the latest valid `BESClientsCatalog.xml` catalog file will be replaced again.

It is highly recommended to use the Fixlet instead of the locally installed Client Deploy Tool user interface to deploy the clients.

Client Deploy Tool standalone

Installing the Client Deploy Tool with MSI

As a prerequisite to run this procedure, ensure that you did not install a BigFix Console. If you already installed the Console, you already have the Client Deploy Tool installed in the *`console_dir`*`\BESClientDeploy` folder.

You can use the Microsoft™ Installer (MSI) version of the Client Deploy Tool to interpret the package and perform the installation automatically. This MSI version of the tool (`BigFixClientDeploy.msi`) is stored in the `BESInstallers\ClientDeployTool` folder of the Windows server.

To install the Client Deploy Tool, perform the following steps:

1. Copy the `BigFixClientDeploy.msi` program and all the related transformation files in the `BESInstallers\ClientDeployTool` directory of a Windows system.
2. Run the `BigFixClientDeploy.msi` program in one of the following ways:
   - `msiexec.exe /i c:\BESInstallers\ClientDeployTool \BigFixClientDeploy.msi TRANSFORMS=`*`TransformList`* `/qn`

     The `/qn` command performs a silent installation.

- `msiexec.exe /i c:\BESInstallers\ClientDeployTool \BigFixClientDeploy.msi INSTALLDIR="c: \myclientdeploytool_install_dir" TRANSFORMS=TransformList`

  This command installs the tool in the specified directory (`INSTALLDIR="c: \myclientdeploytool_install_dir"`).

Note:

`TRANSFORMS=TransformList` specifies what transform files (`.mst`) must be applied to the package. TransformList is a list of paths separated by semicolons. The following table describes the supplied transform files, the resulting language, and the numerical value to use in the **msiexec** command line.

Table 6. Transform file list

Language	Transform File name	Value
U.S. English	1033.mst	1033
German	1031.mst	1031
French	1036.mst	1036
Spanish	1034.mst	1034
Italian	1040.mst	1040
Brazilian Portuguese	1046.mst	1046
Japanese	1041.mst	1041
Korean	1042.mst	1042
Simplified Chinese	2052.mst	2052
Traditional Chinese	1028.mst	1028



You can find the full list of installation options at the Microsoft™ site Command-Line Options. To create a Group Policy Object (GPO) for the BigFix agent deployments, see the Microsoft™ knowledge base article: https://support.hcl-software.com/csm?id=kb_article&sysparm_article=KB0022695.

3. Start the BES client service.

Deploying clients from the tool

Deploying the client computers by using the Client Deploy Tool user interface is obsolete. This option is available only on a Windows system. To install client computers, use either the Client Deploy Tool wizard or the Fixlet from the console.

Deploy the target computers by performing the following steps:

1. The BigFix Client Deploy Tool is installed with the BigFix console or can be installed on a separate system using the MSI Installer Package located in the `BES Installers\ClientDeployTool` directory. Launch the tool directly from **Start > Programs >BigFix > BigFix Client Deploy**.
2. The resulting dialog offers two ways to deploy the targets:

   Find computers using Active Directory

   This option is valid only for the client deployment on Windows targets.

   The BigFix Client Deploy tool contacts the Active Directory server to get a list of all the computers in the domain. It checks each of the computers to see if the client is already installed and displays this information in a list.

   The Client Deploy Tool starts by getting a list of computers from the Active Directory server or from a provided list and remotely checks if the Client service is already installed on each computer. If it is, it reports **Installed** along with the status of the Client service such as **Running**, **Stopped**, and so on. If it cannot determine the status due to a permissions problem or any other issue, it reports **Status Unknown**.

Otherwise it reports **Not Installed** , unless it cannot communicate with the computer at all, in which case it reports **Not Responding**.

Find computers specified in a list

This option is valid both for the client deployment on Windows and UNIX targets.

Based on how your network resolves computer addresses, you must provide a list of computer names, IP address ranges, or host names. The list must have one name / IP address range / host name per line. Using this option, the Client Deploy Tool does not attempt to discover any computers, but instead attempts to install directly to all the listed computers.

3. Type in a **user name** and **password** that has administrative access to the computers. In most cases, this is a domain administrator account with all necessary permissions. If you are using the computer list option, you can specify a local account on the remote computers, such as the local administrator account that has administrative privileges. The rest of the client deployment process uses this user name/password, so if the account does not have the appropriate access on the remote computers, you receive access denied errors.
4. When the list of computers is displayed, shift- and control-click to select the computers that you want to administer with BigFix. Click **Next.**
5. You see a list of the computers that you selected. The default options are usually sufficient, but you might want to select **Advanced Options** to configure the following installation parameters:

File Transfer

This option is valid only on Windows targets.

You can choose to **push** the files out to the remote server for installation or to have the files **pulled** from the local computer. Unless there are security policies in place to prevent it, for most cases choose to push the files.

Note: The pull option is valid only if the target computer belongs to an Active Directory domain and if you use the domain administrator credentials.

Connection Method

This option is valid only on Windows targets.

You can connect to the remote computers either using the **Service Control Manager** (SCM), which is recommended, or the **task scheduler** if the SCM does not work.

Installation Path

This option is valid only on Windows targets.

Specify a path for the client, or accept the default (recommended).

Verification

Select this check box to verify that the client service is running after waiting for the installation to finish, to know if the installation completed successfully.

Custom Settings

Add one or more custom settings to each client deployed, either by entering them in the form of a Name / Value pair or by using the option **Load from file**.

6. If the clients to install need to communicate through a proxy, configure the proxy connection by clicking **Proxy Settings**.
   In the **Proxy Settings** panel, specify:
   - The host name or IP Address and, optionally, the port number to communicate with the proxy machine.
   - The credentials of the user defined on the proxy machine that will be used when establishing the connection.

Select the **Use Internet Explorer proxy settings** check box if you want that the proxy settings are retrieved from the Internet Explorer configuration of the Windows system where the client was installed. This check box works only for Windows targets.

For more information about configuring a proxy connection, see Setting up a proxy connection (on page 545).

Click **OK** to save the proxy configuration.

7. To begin the installation, click **Start**.
8. When completed, a log of successes and failures is displayed. For more details about the log files, see Log files (on page 305).

If you want to deploy from the tool non-Windows target computers, ensure that on the Windows system where your Client Deploy Tool instance was installed:

1. The `BESClientsCatalog.xml` file is stored in the `C:\Program Files(x86)\BigFix Enterprise\BES Client Deploy\` directory.
2. The packages containing the client images are stored in the `C:\Program Files (x86)\BigFix Enterprise\BES Client Deploy\BigFixInstallSource\ClientInstaller` directory.

These files will be present after you deploy at least once the target computers by running the Fixlet named **Install BigFix Clients with Client Deploy Tool**.

Troubleshooting the client deployment

After running a Fixlet to deploy target computers using the Client Deploy Tool, if some target computers were not deployed correctly, you can verify some Client Deploy Tool directories and their contents.

If your Client Deploy Tool instance was on a UNIX target computer, you can perform the following checks:

1. Verify that the `BESClientsCatalog.xml` file is present in the `/var/opt/BESClientDeployTool` directory.
2. Verify that the packages containing the client images are stored in the `/var/opt/BESClientDeployTool/BigFixInstallSource/ClientInstaller` directory.

If your Client Deploy Tool instance was on a Windows target computer, you can perform the following checks:

1. Verify that the `BESClientsCatalog.xml` file is present in the `C:\Program Files(x86)\BigFix Enterprise\BES Client Deploy\` directory.
2. Verify that the packages containing the client images are stored in the `C:\Program Files (x86)\BigFix Enterprise\BES Client Deploy\BigFixInstallSource\ClientInstaller` directory.

Troubleshooting other common errors and problems on UNIX and MAC target computers:

- To successfully deploy UNIX and MAC target computers using the Client Deploy Tool and a user with SUDO privileges, the user with SUDO privileges must be configured as not requiring TTY.
- The deployment of the BigFix Client to Linux target fails and the following message is recorded in the BESClientDeployTool.log: "SSHException in CopyFile: Channel open failure (connect failed)".
  Meaning: Verify the Linux target configuration by running the command

  ```
  sudo grep MaxSessions /etc/ssh/sshd_config
  ```

  and check if the 'no more sessions' error in recorded in the /var/log/secure file. In this case, increase the MaxSessions in sshd_config and run the command

  ```
  sudo systemctl restart sshd
  ```

  The Client Deploy Tool requires multiple simultaneous channels to complete the deployment. The standard OpenSSH default for MaxSessions is 10.

Troubleshooting other common errors and problems on Windows target computers:

The following net use command:

```
net use * \\targetcomputer\admin$ /user:domain\user password
```

can be used to discover which type of error the Client Deploy Tool is running into with the Windows target computers.

- If you receive in the Client Deploy Tool the message "**Offline**", typically with net use you get the following error:

  Error: System error 53 has occurred. The network path was not found.

  Meaning: The computer cannot be contacted.
- If you receive in the Client Deploy Tool the message "**Connection Failed**", with net use you get one of the following errors:

  Error: System error 53 has occurred. The network path was not found.

  Meaning: ADMIN$ share is not available.

  Error: System error 1219 has occurred. Multiple connections to a server or shared resource by the same user, using more than one user name, is not allowed. Disconnect all previous connections to the server or shared resource and try again.

  Meaning: If the computer used to run the Client Deployment Tool already has a connection to a remote machine ADMIN$ share, using a different credential, this error occurs.

  Error: System error 1311 has occurred. There are currently no logon servers available to service the logon request.

  Meaning: The domain server is not available for authentication.

  Error: System error 1326 has occurred. Logon failure: unknown user name or bad password.

  Meaning: Incorrect administrative user name or password.
- If you receive in the Client Deploy Tool the message "**Access is Denied**" or "**Windows Error: Logon failure: unknown user name or bad password**", with net use you get the following error:

Error: System error 5 has occurred. Access is denied.

Meaning: User name/password are correct, but the account does not have permission to ADMIN$ share.

Error: No network provider accepted the given network path.

Meaning: The client or the server could not be resolved during the client deploy tool process.

The "Access is Denied" or "Windows Error: Logon failure: unknown user name or bad password" status indicates that the Client Deploy Tool is unable to connect to the computer to determine if the client is installed. In addition, it is likely that you are also able to deploy the client through the Client Deploy Tool if this error is encountered. The following conditions might be causing this error message:

- An incorrect user name/password was supplied.
- The user account might be locked.
- Insufficient permissions/privileges on the target computer.
- File and Print sharing is disabled on the target computer.
- The Windows Firewall might be blocking the Client Deploy Tool.
- The Windows policy might be blocking the Client Deploy Tool. Try to add or modify the following registry value:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\
Policies\System]
"LocalAccountTokenFilterPolicy"=dword:00000001
```

- If you receive in the Client Deploy Tool an **RPC failure** message, this error occurs when:
  - The remote computer is turned off or unreachable.
  - The remote computer has the RPC service disabled or not functioning.
  - The remote computer has the "file and printer sharing" option disabled.
  - The remote computer is running a personal firewall that blocks the connection attempts. Or the connection attempt is blocked by a firewall in between the computer with the Client Deploy Tool and the remote computer.

Ensure that these issues are not the cause of the problem and then run the Client Deploy Tool again.

Note: After restarting the computer reporting an RPC error, in some cases the RPC error no longer occurs.

Note: As a way to test and see if RPC is listening, Microsoft has a tool named "RPC Ping" which can be downloaded from the following Microsoft article: http://support.microsoft.com/kb/831051.

- If you receive in the Client Deploy Tool the message "**Windows Error 0000046a: Not enough server storage is available to process this command**", this error indicates that the IRPStackSize value is set too low on the target computer resulting in not enough resources being allocated to use a local device. Increase the IRPStackSize value on the target. For more information, see the following Microsoft article: http://support.microsoft.com/kb/106167.

  Try to deploy the client after the value was increased. If the client deployment fails with the same error message, increment the IRPStackSize value and try to deploy the client again.

Log files

Each time the Client Deploy Tool starts, a log file is created.

The log file is named `BESClientDeployTool.log` and it is located in the following directories:

Linux systems

In the `/var/opt/BESClientDeployTool` directory.

Windows systems

In the Client Deploy Tool installation directory.

You cannot disable the log file nor customize its logging level.

The maximum size that the log file can reach is 50 MB. After reaching that level, the log file restarts.

Client log files (UNIX platforms only)

For each UNIX target computer, you have two files which log the stdout and the stderr activity on the remote machine, respectively.

The log files are named `<client_ip_address>.out` for the stdout output and `<client_ip_address>.err` for the stderr output.

You can find these files in the `ClientLogs` folder located in the following directories:

Linux systems

In the `/var/opt/BESClientDeployTool` directory.

Windows systems

In the Client Deploy Tool installation directory.

You cannot disable the log files nor customize their logging level.

The maximum size that the `ClientLogs` folder can reach is 100 MB. After reaching that level, the files stop logging.

Client log files (Windows platforms only)

For each Windows target computer, you have two log files. The log files are named `<client_ip_address>_InstallerService.log` and `<client_ip_address>_ClientInstaller.log`

The first file contains information provided by the installer service, while the second file contains the verbose log of the client installer.

You can find these files in the `ClientLogs` folder located in the following directories:

Linux systems

In the `/var/opt/BESClientDeployTool` directory.

Windows systems

In the Client Deploy Tool installation directory.

Uploading the target logs to the server

Procedure to upload the Client Deploy Tool target log files to the BigFix server.

Perform the following steps:

1. Log in to the BigFix console.
2. Click the **Fixlets and Tasks** icon in the Domain Panel.
3. In the search bar, enter Upload Deploy.
4. Select the Task named **Upload BES Client Deploy Tool Logs**.
5. Click **Take Action**.
6. In the Applicable Computers tab, select the target computers from which you want to retrieve the Client Deploy Tool log files.
7. Click **OK**. Verify the status of the Task.

After running this Task, the log files are uploaded to the BigFix server directory `*Installation_dir*/UploadManagerData/BufferDir/sha1/xx/xxxxxxx`

where:

xx

Represents the last two digits of the Client ID.

xxxxxxx

Represents the full Client ID.

The names of the uploaded log files use the following prefixes:

`cdtMainLog_` for the main Client Deploy Tool log file, which is `BESClientDeployTool.log`

`cdtClientLogs_` for each log file taken from the `ClientLogs` folder of the Client Deploy Tool.

The files are uploaded to the BigFix server only if their size is smaller than 50 MB, unless you specified a different limit by using the appropriate client setting.

The Task is relevant if the following conditions are satisfied:

- The `BESClientDeployTool.log` file exists in the installation directory of the Client Deploy Tool.
- The `ClientLogs` folder exists in the installation directory of the Client Deploy Tool.

These conditions are met if you started the Client Deploy Tool at least once.

Important: To clean up the Client Deploy Tool reports older than a week, you can use the task named **BES Client Deploy Tool cleanup of obsolete reports**.

Limitations in Client Deploy Tool

The Client Deploy Tool allows you to deploy BigFix clients on all the platforms supported by BigFix Platform.

Known limitations

For performance reasons, the Client Deploy Tool was tested to deploy one hundred target computers (clients) at a time.

Installing the Client on AIX

How to install the client on AIX.

To install and run BigFix Agent and Relay on AIX, starting from Patch 1, the system needs to match all of the following conditions:

- At least version 16.1.0.10 of both LLVM runtime libraries (libc++.rte fileset) and XL C++ runtime libraries (xlC.rte fileset) must be installed
- if XL C/C++ compiler (xlCcmp fileset) is installed, then it must be at version 16.1.0.10 or higher
- if libc++.rte fileset version 17.1.0.0 or higher is installed, then at least version 17.1.1.4 of IBM Open XL C/C++ runtime libraries (libc++.rte, libc++abi.rte and libunwind.rte filesets) must be installed

The 16.x filesets can be found in the latest fix version available of IBM XL C/C++ Runtime for AIX 16.1, while the 17.x filesets (along with the latest xlC runtime available) can be found in the latest fix version available of IBM Open XL C/C++ Runtime for AIX 17.1.1.

Perform the following steps:

1. Download the corresponding BigFix client package file to the IBM AIX computer.
2. Copy the BESAgent to the IBM AIX computer.
3. Run the following command:

```
installp -agqYXd ./BESAgent-11.0.xxx.x.ppc_aixxx.pkg BESClient
```



Important: Starting with release IBM AIX 7.2 with Technology Level 4, the software packages of the AIX operating system are digitally signed. The installation process verifies the digital signatures of the software package and takes action based on the digital signature policy. In order to successfully install/upgrade the BigFix Client, the Digital Signature Policy must be set to 'none'.

4. Copy your action site masthead to the client computer (the masthead contains configuration, license, and security information). The action site masthead (`actionsite.afxm`) can be found in your BES Installation folders (by default they are placed under `C:\Program Files (x86)\BigFix Enterprise\BES Installers\Client` on Windows and `/var/opt/BESInstallers/Client/` on Linux). If the masthead is not named `actionsite.afxm`, rename it to `actionsite.afxm` and place it on the computer at the following location: `/etc/opt/BESClient/actionsite.afxm`.



Note: The directory `/etc/opt/BESClient/` is not automatically created by the installer. If it does not exist, create it manually.

The masthead file for each BigFix server is downloadable at `http://servername:port/masthead/masthead.afxm` (example: `http://bes.BigFix.com:52311/masthead/masthead.afxm`).

5. Start the BigFix client by running the following command:

```
/etc/rc.d/rc2.d/SBESClientd start
```

Note:

You can install the client on Virtual I/O Server (VIOS). As a prerequisite, run the oem_setup_env command. The command places the user into the OEM software installation and setup environment.

AIX Fixlet Content

To get the Fixlet content for the AIX BigFix agent, subscribe your BigFix server to the appropriate Fixlet site.

To subscribe to a new Fixlet site, perform the following steps:

1. Go to a computer with the BigFix console installed.
2. Download the masthead.
3. When prompted to open or save the file, click **Open** to open the BigFix console.
4. Log into the BigFix console with your username and password.
5. After logged in, the BigFix console asks if you wish to subscribe to the Patches for AIX Fixlet site, click **OK**.
6. Type in your private key password and click **OK**.
7. After the BigFix console subscribes to the site, it starts gathering new Fixlet messages from the site.

Installing the Client on Linux

The BigFix client can always be installed by manually running the client installer on each computer.

This is a quick and effective mechanism for installing the client on a small number of computers.

Note: If the actionsite masthead with which you installed the client computer contains the fallback relay, and no other relay option is provided to the client

computer using a configuration file, then ensure that the newly installed client computer can connect to the fallback relay of your environment.

Alma Linux Installation Instructions

How to install the client on Alma Linux 9.5.

To install the client perform the following steps:

1. Download the corresponding BigFix client RPM file to the Alma Linux computer.
2. Install the RPM by running the command

   ```
   rpm -ivh client_RPM_path
   ```

3. Copy your actionsite masthead to the client computer (the masthead contains configuration, license, and security information). The action site masthead (`actionsite.afxm`) can be found in your BES Installation folders (by default they are placed under `C:\Program Files (x86)\BigFix Enterprise\BES Installers\Client` on Windows and `/var/opt/BESInstallers/Client/` on Linux). If the masthead is not named `actionsite.afxm`, rename it to `actionsite.afxm` and place it on the computer at the following location: `/etc/opt/BESClient/actionsite.afxm`.

   Note: The directory `/etc/opt/BESClient/` is not automatically created by the installer. If it does not exist, create it manually.

   The masthead file for each BigFix server is downloadable at `http://servername:port/masthead/masthead.afxm` (example: `http://bes.BigFix.com:52311/masthead/masthead.afxm`).
4. Start the BigFix client by running the command:

   ```
   systemctl start besclient
   ```

Note:

The BigFix client installation on an Alma Linux 9.5 operating system can be performed also on computers having SELinux enabled. With SELinux enabled, the following SELinux settings are supported:

selinux = enforcing, policy = targeted.

Amazon Linux Installation Instructions

How to install the client on Amazon Linux.

To install the client perform the following steps:

1. Download the corresponding BigFix client RPM file to the Amazon Linux computer.
2. Install the RPM by running the command

   ```
   rpm -ivh client_RPM_path
   ```

3. Copy your actionsite masthead to the client computer (the masthead contains configuration, license, and security information). The action site masthead (`actionsite.afxm`) can be found in your BES Installation folders (by default they are placed under `C:\Program Files (x86)\BigFix Enterprise\BES Installers\Client` on Windows and `/var/opt/BESInstallers/Client/` on Linux). If the masthead is not named `actionsite.afxm`, rename it to `actionsite.afxm` and place it on the computer at the following location: `/etc/opt/BESClient/actionsite.afxm`.

   Note: The directory `/etc/opt/BESClient/` is not automatically created by the installer. If it does not exist, create it manually.

   The masthead file for each BigFix server is downloadable at `http://servername:port/masthead/masthead.afxm` (example: `http://bes.BigFix.com:52311/masthead/masthead.afxm`).
4. Start the BigFix client by running the command:

   ```
   systemctl start besclient
   ```

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

```
/etc/init.d/besclient start
```

Note: When upgrading to BigFix Version 11.0.2 or later, the `/etc/init.d/besclient` file, if present, is removed by the upgrade process and the `/etc/init.d/besclient start|stop|status|restart` commands are no longer available.

Note:

The BigFix client installation on Amazon Linux operating system can be performed also on computers having SELinux enabled. With SELinux enabled, the following SELinux settings are supported:

selinux = enforcing, policy = targeted.

Signed RPM packages

The RPM packages are signed with a PGP key. For more details, see Red Hat Installation Instructions (on page 317).

CentOS Installation Instructions

How to install the client on CentOS.

To install the client perform the following steps:

1. Download the corresponding BigFix client RPM file to the CentOS computer.
2. Install the RPM by running the command

   ```
   rpm -ivh client_RPM_path
   ```

3. Copy your actionsite masthead to the client computer (the masthead contains configuration, license, and security information). The action site masthead

(`actionsite.afxm`) can be found in your BES Installation folders (by default they are placed under `C:\Program Files (x86)\BigFix Enterprise\BES Installers\Client` on Windows and `/var/opt/BESInstallers/Client/` on Linux). If the masthead is not named `actionsite.afxm`, rename it to `actionsite.afxm` and place it on the computer at the following location: `/etc/opt/BESClient/actionsite.afxm`.



Note: The directory `/etc/opt/BESClient/` is not automatically created by the installer. If it does not exist, create it manually.

The masthead file for each BigFix server is downloadable at `http://servername:port/masthead/masthead.afxm` (example: `http://bes.BigFix.com:52311/masthead/masthead.afxm`).

4. Start the BigFix client by running the command:

```
systemctl start besclient
```

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

```
/etc/init.d/besclient start
```



Note: When upgrading to BigFix Version 11.0.2 or later, the `/etc/init.d/besclient` file, if present, is removed by the upgrade process and the `/etc/init.d/besclient start|stop|status|restart` commands are no longer available.



Note:

The BigFix client and relay installation on a CentOS Version 6 and 7 operating system can be performed also on computers having SELinux enabled. With SELinux enabled, the following SELinux settings are supported:

selinux = enforcing, policy = targeted.

Signed RPM packages

The RPM packages are signed with a PGP key. For more details, see Red Hat Installation Instructions (on page 317).

Oracle Linux Installation Instructions

How to install the client on Oracle Linux.

Before installing the client on Oracle Enterprise Linux 9, ensure that you have installed the initscripts package.

To install the client perform the following steps:

1. Download the corresponding BigFix client RPM file to the Oracle Linux computer.
2. Install the RPM by running the command

   ```
   rpm -ivh client_RPM_path
   ```

3. Copy your actionsite masthead to the client computer (the masthead contains configuration, license, and security information). The action site masthead (`actionsite.afxm`) can be found in your BES Installation folders (by default they are placed under `C:\Program Files (x86)\BigFix Enterprise\BES Installers\Client` on Windows and `/var/opt/BESInstallers/Client/` on Linux). If the masthead is not named `actionsite.afxm`, rename it to `actionsite.afxm` and place it on the computer at the following location: `/etc/opt/BESClient/actionsite.afxm`.

   Note: The directory `/etc/opt/BESClient/` is not automatically created by the installer. If it does not exist, create it manually.

   The masthead file for each BigFix server is downloadable at `http://servername:port/masthead/masthead.afxm` (example: `http://bes.BigFix.com:52311/masthead/masthead.afxm`).
4. Start the BigFix client by running the command:

   ```
   systemctl start besclient
   ```

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

```
/etc/init.d/besclient start
```

Note: When upgrading to BigFix Version 11.0.2 or later, the `/etc/init.d/besclient` file, if present, is removed by the upgrade process and the `/etc/init.d/besclient start|stop|status|restart` commands are no longer available.

Note:

The BigFix client installation on Oracle Linux operating system can be performed also on computers having SELinux enabled. With SELinux enabled, the following SELinux settings are supported:

selinux = enforcing, policy = targeted.

Note: On Oracle Enterprise Linux 9, the user interface component of the client is displayed only if you log in choosing GNOME on Xorg.

Signed RPM packages

The RPM packages are signed with a PGP key. For more details, see Red Hat Installation Instructions (on page 317).

Raspbian Installation Instructions

How to install the client on Raspbian.

Perform the following steps:

1. Download the corresponding BigFix client Raspbian package file to the Raspberry PI client.
2. Install the package by running the command:

```
dpkg -i client_package_path
```

3. Copy your actionsite masthead to the client (the masthead contains configuration, license, and security information). The action site masthead (`actionsite.afxm`) can be found in your BES Installation folders (by default they are placed under `C:\Program Files (x86)\BigFix Enterprise\BES Installers\Client` on Windows and `/var/opt/BESInstallers/Client/` on Linux). If the masthead is not named `actionsite.afxm`, rename it to `actionsite.afxm` and place it on the client at the following location: `/etc/opt/BESClient/actionsite.afxm`.

   Note: The directory `/etc/opt/BESClient/` is not automatically created by the installer. If it does not exist, create it manually.

   The masthead file for each BigFix server can be downloaded from `http://servername:port/masthead/masthead.afxm` (example: `http://bes.BigFix.com:52311/masthead/masthead.afxm`).
4. Start the BigFix client by running the command:

   ```
   systemctl start besclient
   ```

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

   ```
   /etc/init.d/besclient start
   ```

Red Hat Installation Instructions

How to install the client on Red Hat.

Before installing the client on Red Hat Enterprise Linux™ 7, 8 or 9, ensure that you have disabled the SELinux process or, if you want to keep SELinux enabled, that the following settings are configured:

selinux = enforcing, policy = targeted.

Then, ensure that you have:

- Installed the Athena library (libXaw package) that is used by the user interface component of the client.
- Installed the initscripts package before installing the client on Red Hat Enterprise Linux™ 9.

Note: For customers planning to implement fapolicyd in their Red Hat Enterprise Linux™ environments, the BigFix Client is supported on Red Hat Enterprise Linux™ 8 and 9 with fapolicyd enabled and operating in enforcing mode. Certification has been performed using fapolicyd-1.3.2 with a "deny all permit by exception" policy in compliance with Red Hat Enterprise Security Technical Implementation Guide. Although the BigFix client is installed via an RPM package and fapolicyd trusts files in the RPM database, additional configuration of fapolicyd is required to ensure the full functionality of the BigFix Client. See Configuring fapolicyd to allow BigFix Client operations *(on page 322)*.

To install the client perform the following steps:

1. Download the corresponding BigFix client RPM file to the Red Hat computer.
2. Install the RPM by running the command

   ```
   rpm -ivh client_RPM_path
   ```

   Note:

   Starting from BigFix Version 9.5.9, if you are installing the signed packages and you have not imported the public key for that signature, you receive the following warning:

   ```
   BESAgent-9.5.9.xx-rhe6.x86_64.rpm: Header V4 RSA/SHA256
   Signature, key ID 3e83b424: NOKEY
   ```

3. Copy your actionsite masthead to the client computer (the masthead contains configuration, license, and security information). The action site masthead (`actionsite.afxm`) can be found in your BES Installation folders (by default they

are placed under `C:\Program Files (x86)\BigFix Enterprise\BES Installers\Client` on Windows™ and `/var/opt/BESInstallers/Client/` on Linux™). If the masthead is not named `actionsite.afxm`, rename it to `actionsite.afxm` and place it on the computer at the following location: `/etc/opt/BESClient/actionsite.afxm`.

Note: The directory `/etc/opt/BESClient/` is not automatically created by the installer. If it does not exist, create it manually.

The masthead file for each BigFix Server can be downloaded at `http://servername:port/masthead/masthead.afxm` (example: `http://bes.BigFix.com:52311/masthead/masthead.afxm`).

4. Start the BigFix client by running the command:

```
systemctl start besclient
```

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

```
/etc/init.d/besclient start
```

Note: When upgrading to BigFix Version 11.0.2 or later, the `/etc/init.d/besclient` file, if present, is removed by the upgrade process and the `/etc/init.d/besclient start|stop|status|restart` commands are no longer available.

Note: On Red Hat Enterprise Linux™ 9, the user interface component of the client is displayed only if you log in choosing GNOME on Xorg.

Note: On Red Hat Enterprise Linux™ 10, the user interface component of the client is not supported.

Signed Client Red Hat RPM packages

Starting from BigFix Version 9.5.9, the Red Hat RPM packages are signed with a PGP key.

Starting from BigFix Version 10.0.8, the signature of the Red Hat RPM package for BigFix Agent contains the SHA256 digest and header, thus you can install the BigFix Agent on Red Hat systems with FIPS mode enabled.

The RPM packages available for download are stored, divided by product version and platform, in the following repository: http://support.bigfix.com/bes/release/.

Run the following command to check if the package file is signed:

```
rpm -qpi <package>.rpm
```

In the command output the content of the Signature field shows if the package is signed or not:

- The package is signed if the Signature field is not empty.
- The package is not signed if the Signature field does not contain any value.

This is a sample output that you can get if the package is signed:

```
 Name : BESAgent
Version     : 10.0.0.133
Release     : rhe6
Architecture: x86_64
Install Date: (not installed)
Group       : Applications/Security
Size        : 54525522
License     : (c) Copyright HCL Technologies Limited 2001-2020 ALL RIGHTS
 RESERV                                          ED
Signature   : RSA/SHA256, Sun 29 Mar 2020 11:01:24 PM CEST, Key ID
 f103a7e216055                                          553
Source RPM  : BESAgent-10.0.0.133-rhe6.src.rpm
Build Date  : Sun 29 Mar 2020 08:52:46 PM CEST
Build Host  : platbuild-rhel-6-x86-64-2.platform.bes.prod.hclpnp.com
```

```
Relocations : (not relocatable)
Packager    : HCL Technologies Limited
Vendor      : HCL Technologies Limited
URL         : http://www.bigfix.com/
Summary     : BigFix Agent
Description :
BigFix Agent for Linux.
```

If the package is signed, you can download and import the public key for that signature by running the BES Support Fixlet named **Import BigFix public GPG key for RedHat RPMs**.

If you download the key, then import it into your local machine keystore by using the command:

```
rpm --import <keyfile>
```

where the key file can be a URL or a local file.

The BigFix public key available for download is stored in the following repository: http://support.bigfix.com/bes/release/

At this point, you can proceed to install the RPM package on the client system by running the command:

```
'rpm -i <package name>'
```

or an equivalent command.

If you did not import the public key, during the client installation you might see a warning message saying that the signature cannot be verified. This message does not prevent your RPM package from being installed successfully on the client system.

Red Hat Fixlet Content

To get the Patch content for the Red Hat BigFix agent, you need to subscribe your BigFix server to the appropriate Patch site.

To subscribe to a new Patch site, perform the following steps:

1. From the BigFix Management domain, click License Overview dashboard.
2. Scroll down to view the available content sites.
3. Click **Enable** to select the version of Patches for RHEL site to which you want to subscribe.
4. Open the Manage Sites node and select your newly subscribed site. From the site dialog, click the Computer Subscriptions tab to assign the site to the appropriate computers.
5. From the Operator Permissions tab, select the operators that you want to associate with this site and their level of permission.
6. Click **Save Changes**.

You are now subscribed to a Patches for RHEL site. For more information on patching your endpoints, see Patch for Red Hat Enterprise Linux.

Configuring fapolicyd to allow BigFix Client operations

When fapolicyd (File Access Policy Daemon) is enabled on Red Hat Enterprise Linux™ and operating in enforcing mode, fapolicyd blocks certain BigFix client operations. Below are the affected scenarios and the methods to whitelist the BES Client services.

Whitelisting the Besclient and QnA

The RPM inspector uses the BESClient - RPMHelper process to retrieve information from the RPM database.

To whitelist the besclient RPM-Helper, modify the **90-deny-execute.rules** or, if present, modify any **deny custom rule** (i.e 99-deny-all.rules) file located in the `/etc/fapolicyd/rules.d/` folder. The following lines must precede the original rules:

```
allow perm=open auid=-1 exe=/opt/BESClient/bin/BESClient : all
allow perm=open auid=-1 exe=/opt/BESClient/bin/qna : all
```

Whitelisting the Nmap scanner

Asset Discovery uses the Nmap security scanner to scan networks for the Identification of network assets.

To whitelist the Nmap scanner installed with the Fixlet “Designate Nmap Scan Point - Red Hat Enterprise Linux | CentOS”, modify the **90-deny-execute.rules** or if present, modify the **deny custom rule** (i.e 99-deny-all.rules) file located in the `/etc/fapolicyd/rules.d/` folder. The following line must precede the original rule:

```
allow perm=open auid=-1 exe=/var/opt/BESClient/BESScanner-NMAP/nmap : all
```

Whitelisting the Inventory scanner

BES Inventory and License site

- **Fixlet “Install or Upgrade Scanner (9.2.33)” fails with fapolicy enabled and enforcing "deny all permit by exception" policy**

  The installation of the scanner included in the “BES Inventory and License” requires some shared libraries to install the scanner.

  To whitelist shared libraries used during the installation of the software scanner, modify the **41-shared-obj.rules** file located in the `/etc/fapolicyd/rules.d/` folder. The following lines must precede the original rules:

  ```
  allow perm=open exe=/opt/tivoli/cit/bin/wscancfg trust=0 : all
  ```

- **Fixlet “Run Full Hardware Scan” fails with fapolicy enabled and enforcing "deny all permit by exception" policy**

  The scanner included in the “BES Inventory and License” uses some processes to perform the hardware scan.

  To whitelist the processes used during the run of the Hardware scanner, modify the **41-shared-obj.rules** file located in the `/etc/fapolicyd/rules.d/` folder. The following lines must precede the original rules:

  ```
  allow perm=open exe=/opt/tivoli/cit/bin/wscanhw trust=0 : all
  allow perm=open exe=/opt/tivoli/cit/bin/cpuid trust=0 : all
  allow perm=open exe=/opt/tivoli/cit/bin/diskscan trust=0 : all
  ```

  Additionally, the following rule must be placed in **90-deny-execute.rules** or if present, modify any **deny custom rule** (i.e 99-deny-all.rules) file, preceding the original rules:

```
allow perm=execute exe=/opt/tivoli/cit/bin/wscanhw trust=0 :
 path=/opt/tivoli/cit/bin/cpuid ftype=application/x-executable trust=0
allow perm=execute exe=/opt/tivoli/cit/bin/wscanhw trust=0 :
 path=/opt/tivoli/cit/bin/diskscan ftype=application/x-executable
 trust=0
```

After saving the rules files execute the `fagenrules --load` command to update the active rules and restart the fapolicyd service.

Rocky Linux Installation Instructions

How to install the client on Rocky Linux.

Before installing the client on Rocky Linux, ensure that you have:

- Installed the Athena library (libXaw package) that is used by the user interface component of the client.
- Installed the initscripts package before installing the client on Rocky Linux™ 9.
- Added UDP port 52311 before installing the client on Rocky Linux™ 9, if you want your client to receive UDP messages. Use the following commands to list and add UDP port 52311:

  ```
  firewall-cmd --get-default-zone
  firewall-cmd --get-active-zones
  firewall-cmd --list-ports
  firewall-cmd --zone=public --add-port=52311/udp
  firewall-cmd --list-ports
  ```

To install the client perform the following steps:

1. Download the corresponding BigFix client RPM file to the Rocky Linux computer.
2. Install the RPM by running the command

   ```
   rpm -ivh client_RPM_path
   ```

3. Copy your actionsite masthead to the client computer (the masthead contains configuration, license, and security information). The action site masthead (`actionsite.afxm`) can be found in your BES Installation folders (by default they are placed under `C:\Program Files (x86)\BigFix Enterprise\BES Installers\Client` on Windows and `/var/opt/BESInstallers/Client/` on Linux). If the masthead is not named `actionsite.afxm`, rename it to `actionsite.afxm` and place it on the computer at the following location: `/etc/opt/BESClient/actionsite.afxm`.

   Note: The directory `/etc/opt/BESClient/` is not automatically created by the installer. If it does not exist, create it manually.

   The masthead file for each BigFix server is downloadable at `http://`*`servername`*`:`*`port`*`/masthead/masthead.afxm` (example: `http://bes.BigFix.com:52311/masthead/masthead.afxm`).
4. Start the BigFix client by running the command:

   ```
   systemctl start besclient
   ```

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

   ```
   /etc/init.d/besclient start
   ```

   Note: When upgrading to BigFix Version 11.0.2 or later, the `/etc/init.d/besclient` file, if present, is removed by the upgrade process and the `/etc/init.d/besclient start|stop|status|restart` commands are no longer available.

Note:

On Rocky Linux 8, the user interface component of the client is displayed only if you log in choosing Standard on Xorg.

Note:

On Rocky Linux 9, the user interface component of the client is displayed only if you log in choosing GNOME on Xorg.

Signed RPM packages

The RPM packages are signed with a PGP key. For more details, see Red Hat Installation Instructions (on page 317).

SUSE Linux Enterprise (64-bit) Installation Instructions

How to install the client on SUSE Linux Enterprise (64-bit).

Prerequisites

- Before running the BigFix Client user interface in a GNOME desktop environment on SLES/SLED 15, download and install `libXaw8-1.0.13-1.26.x86_64.rpm`.
- All RPM libraries installed on the system must be at the same level.

To install the client, do the following steps:

1. Disable the SELinux process.
2. Download the corresponding BigFix client RPM file to the SUSE computer.
3. Install the RPM by running the command

   ```
   rpm -ivh client_RPM_path
   ```

4. Copy your actionsite masthead to the client computer (the masthead contains configuration, license, and security information). The action site masthead (`actionsite.afxm`) can be found in your BES Installation folders (by default they are placed under `C:\Program Files (x86)\BigFix Enterprise\BES Installers\Client` on Windows and `/var/opt/BESInstallers/Client/` on Linux). If the masthead is not named `actionsite.afxm`, rename it to `actionsite.afxm` and place it on the computer at the following location: `/etc/opt/BESClient/actionsite.afxm`.

Note: The directory `/etc/opt/BESClient/` is not automatically created by the installer. If it does not exist, create it manually.

The masthead file for each BigFix server is downloadable at `http://servername:port/masthead/masthead.afxm` (example: `http://bes.BigFix.com:52311/masthead/masthead.afxm`).

5. Start the BigFix client by running the command:

```
systemctl start besclient
```

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

```
/etc/init.d/besclient start
```

Note: When upgrading to BigFix Version 11.0.2 or later, the `/etc/init.d/besclient` file, if present, is removed by the upgrade process and the `/etc/init.d/besclient start|stop|status|restart` commands are no longer available.

Note: On SLES 16, the user interface component of the client is not supported.

OpenSUSE Leap (64-bit) Installation Instructions

How to install the client on OpenSUSE Leap (64-bit).

Prerequisites

- Up to BigFix Agent version 11.0.1, on OpenSUSE Leap 15.4/15.5 x86-64, install the `sysvinit-tools` rpm package before installing the client.

To install the client, do the following steps:

1. Disable the SELinux process.
2. Download the corresponding BigFix client RPM file to the OpenSUSE Leap computer.
3. Install the RPM by running the command

   ```
   rpm -ivh client_RPM_path
   ```

4. Copy your actionsite masthead to the client computer (the masthead contains configuration, license, and security information). The action site masthead (`actionsite.afxm`) can be found in your BES Installation folders (by default they are placed under `C:\Program Files (x86)\BigFix Enterprise\BES Installers\Client` on Windows and `/var/opt/BESInstallers/Client/` on Linux). If the masthead is not named `actionsite.afxm`, rename it to `actionsite.afxm` and place it on the computer at the following location: `/etc/opt/BESClient/actionsite.afxm`.

   Note: The directory `/etc/opt/BESClient/` is not automatically created by the installer. If it does not exist, create it manually.

   The masthead file for each BigFix server is downloadable at `http://servername:port/masthead/masthead.afxm` (example: `http://bes.BigFix.com:52311/masthead/masthead.afxm`).
5. Start the BigFix client by running the command:

   ```
   systemctl start besclient
   ```

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

   ```
   /etc/init.d/besclient start
   ```

   Note: When upgrading to BigFix Version 11.0.2 or later, the `/etc/init.d/besclient` file, if present, is removed by the upgrade process and the `/etc/init.d/besclient start|stop|status|restart` commands are no longer available.

Note: On OpenSUSE Leap 15.4, 15.5 and 15.6 the user interface component of the client is displayed only if you log in choosing GNOME on Xorg.

Note: On OpenSUSE Leap 16, the user interface component of the client is not supported.

Ubuntu/Debian (64-bit) Installation Instructions

How to install the client on Ubuntu/Debian (64-bit).

To install the client perform the following steps:

1. Download the corresponding BigFix client DEB package file to the Ubuntu/Debian computer.
2. Copy your actionsite masthead to the client computer (the masthead contains configuration, license, and security information). The action site masthead (`actionsite.afxm`) can be found in your BES Installation folders (by default they are placed under `C:\Program Files (x86)\BigFix Enterprise\BES Installers\Client` on Windows and `/var/opt/BESInstallers/Client/` on Linux). If the masthead is not named `actionsite.afxm`, rename it to `actionsite.afxm` and place it on the computer at the following location: `/etc/opt/BESClient/actionsite.afxm`.

   Note: The directory `/etc/opt/BESClient/` is not automatically created by the installer. If it does not exist, create it manually.

   The masthead file for each BigFix server is downloadable at `http://servername:port/masthead/masthead.afxm` (example: `http://bes.BigFix.com:52311/masthead/masthead.afxm`).
3. Install the DEB by running the command

   ```
   dpkg -i client_ package_path
   ```
4. Start the BigFix client by running the command:

```
systemctl start besclient
```

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

```
/etc/init.d/besclient start
```

Note: On Debian 11, the user interface component of the client is displayed only if you log in choosing GNOME on Xorg.

Note: On Ubuntu 22.04 LTS and 24.04 LTS, the user interface component of the client is displayed only if you log in choosing Ubuntu on Xorg.

Installing the Client on Mac

How to install the Mac client.

Perform the following steps:

1. Download the corresponding BigFix client package file to the Mac computer.
2. Copy the PKG file to any directory and copy the masthead file for your deployment into the same directory. Ensure that the masthead file is named `actionsite.afxm`.
3. You might optionally include a pre-defined settings file (`clientsettings.cfg`) in the same directory as the PKG file and the `actionsite.afxm` file, to create custom settings for the Mac client at installation time, for example to assign the new Client to a specific parent relay. For more information, see Mac Clients *(on page 484)*.

   Note: If you previously uninstalled a BigFix Client on the same Mac system, and you plan to reinstall it using the `clientsettings.cfg` file, ensure to reboot the system before starting the Client installation.

4. Launch the PKG installer by double-clicking the PKG file (such as `BESAgent-10.0.xxx.x-BigFix_MacOS11.0.pkg`) and run through the installer. The agent starts up after the installation completes as long as the masthead file is included in the installation directory.

Note:

- The agent uninstaller is available in the .pkg install. It is located in: `/Library/BESAgent/BESAgent.app/Contents/MacOS/BESAgentUninstaller.sh`
- The agent .dmg package is no longer available.

Note: The BESAgent service can access all of the user's private files and folders in the Mac system only if you add full disk access permission to it. It can be done manually by the user from the **Privacy** tab of the **Security & Privacy Preferences** panel or by using MDM services. For a BigFix MCM Device this can be done automatically at installation time, see Deploy BigFix Agent for more information.

Mac Fixlet Content

To get the Fixlet content for the Mac BigFix agent, subscribe your BigFix server to the appropriate Fixlet site.

To subscribe to a new Fixlet site, perform the following steps:

1. Go to a computer with the BigFix console installed.
2. Download the masthead.
3. When prompted to open or save the file, click **Open** to open the BigFix console.
4. Log into the BigFix console with your username and password.
5. After logged in, the BigFix console asks if you wish to subscribe to the Patches for Mac OS X Fixlet site, click **OK**.

6. Type in your private key password and click **OK**.
7. After the BigFix console subscribes to the site, it starts gathering new Fixlet messages from the site.

Installing the Client on Solaris 11

As a prerequisite, all Solaris agents must have the SUNWlibC package installed.

Starting from BigFix Version 9.5.13, you can install the **Solaris 11** client using a **.p5p** client package format by performing the following steps:

1. Download the corresponding BigFix client package file to the Solaris computer.
2. Copy your actionsite masthead to the Solaris BigFix client computer (the masthead contains configuration, license, and security information). The action site masthead (`actionsite.afxm`) can be found in your BigFix Installation folders (by default they are placed under `C:\Program Files (x86)\BigFix Enterprise\BES Installers\Client` on Windows and `/var/opt/BESInstallers/Client/` on Linux). If the masthead is not named `actionsite.afxm`, rename it to `actionsite.afxm` and place it on the computer at the following location: `/etc/opt/BESClient/actionsite.afxm`. The masthead file for any BigFix server can also be downloaded at `http://`*`servername`*`:`*`port`*`/masthead/masthead.afxm` (example: `http://bes.BigFix.com:52311/masthead/masthead.afxm`).

   Note: You might need to create the directory `/etc/opt/BESClient/` if it does not already exist.

3. Before installing the **.p5p** client package, as upgrading from the old SVR4 packages is not supported, ensure that no previous agent is already installed by running the command:

   ```
   pkginfo BESagent
   ```

   If necessary, remove the old BigFix agent before installing the new one with the command:

   ```
   pkgrm BESagent
   ```

4. If the computer is **not** a **global zone** or is a **global zone without children**, you can directly install the new agent (**.p5p** client package) by running the command:

```
pkg install -g <path to package file>/BESAgent-<...>.p5p BESagent
```

5. Otherwise, if the computer is a **global zone with children**, you must create a permanent repository and set the publisher, for example by running the command:

```
pkgrepo create /var/opt/BESClient_solaris_repo
pkgrecv -s <path to package file>/BESAgent-<...>.p5p
 -d /var/opt/BESClient_solaris_repo BESagent
pkg set-publisher --search-first -p /var/opt/BESClient_solaris_repo
```

   Then, you can install with the command:

```
pkg install BESagent
```

   The **--search-first** option is needed to avoid that other, not available, publishers might cause issues.

   The created repository is also available for the children. Therefore, there is no need to create it on the non-global zones.

   Note: You can install on the children using the **-r** option. For details about the **-r** option and other command options, see https://docs.oracle.com/cd/E36784_01/html/E36870/pkg-1.html

Uninstalling the client

To uninstall the client, installed using a **.p5p** client package format, run the command:

```
pkg uninstall BESagent
```

This command can be used for uninstalling both on a global zone or on the local zones.

Note: You can uninstall on the children using the **-r** option. For details about the **-r** option and other command options, see https://docs.oracle.com/cd/E36784_01/html/E36870/pkg-1.html

Upgrading the client

Upgrading manually from a client installed using the old SVR4 package to a new client with the **.p5p** client package is not supported.

You can upgrade the Solaris 11 client by running the Fixlet named **Updated Solaris Client - BigFix version 9.5.X Now Available!**

Troubleshooting the upgrade

When upgrading the BigFix Client on an Oracle Solaris 11 local zone using the IPS package, the action might fail, with the `/tmp/BESClientUpgradeFixlet.log` file on the local zone showing the following error message:

```
Cannot enable or disable a system publisher
```

The error might be due to a corruption of the BigFix publisher in the Oracle Solaris zone environment.

The pkg publisher command executed on the global zone does not show any BigFix publisher, while executed on the local zones shows the status disabled,syspub.

As a workaround to solve this issue, try one of the following options:

- If not already done, submit the **Updated Solaris Client - BigFix version 9.5.14 Now Available!** Fixlet to the global zone. Wait for the action to complete.

  Only when the action completes, submit the same Fixlet to the local zone(s).
- On the global zone, run the command:

  ```
  pkg set-publisher --enable BigFix
  ```

  Check the publisher on the local zone(s) to verify if the "disabled" status is no longer shown.

  Try upgrading the local zone(s) again.
- Uninstall and reinstall the client on the local zone(s).

Troubleshooting the uninstallation

After running the pkg uninstall BESagent command, you might receive the following error message:

```
DESC: A service failed - a method is failing in a retryable manner but too
 often.
AUTO-RESPONSE: The service has been placed into the maintenance state.
IMPACT: svc:/BESClient:default is unavailable.
```

Despite the error message, the client is correctly uninstalled. You can ignore it.

Known limitations

Limitation **1**

The Image Packaging System (IPS) does not support product directories symbolically linked. When using symbolic links, the package installation fails with the following error message:

```
pkg: Requested operation failed for package
 pkg://software.bigfix.com/BESagent@.........
Cannot install '.....'; parent directory ...... is a link to ........
To continue, move the directory to its original location and try again.
```

Limitation **2**

Uninstalling the IPS package moves the files added after the installation under `$IMAGE_META/lost+found`

Where the default value for IMAGE_META is `/var/pkg`

As a side effect, a subsequent installation of the agent will result as a new one (different ID, lost cache), unless you manually restore the files from `$IMAGE_META/lost+found` to `/var/opt/BESClient` before you perform the new installation.

Installing the Client on Windows

You can install the BigFix client manually by running the Client installer on each computer.

Use this method to install the client on a small number of computers and if you installed the BigFix server on a Windows system. Run this sequence of steps to run the client installation:

1. You can install the client using one of the following methods:
   - Log on to the computer with administrator privileges and copy the **BES Installers\Client** folder from the installation computer to the local hard drive.
   - Run the Installation Guide (available at **Start > Programs > BigFix > BigFix Installation Guide**) and click the button marked **Browse Install Folders** to open the **BigFix Installers** folder and display the **Client** folder.
2. You might optionally include a pre-defined settings file (`clientsettings.cfg`) in the same directory as the `setup.exe` file, to create custom settings for the Windows client at installation time, for example to assign the new Client to a specific parent relay. For more information, see Windows Clients *(on page 484)*.
3. After you have copied the Client folder to the target computer, double-click **setup.exe** from that folder to launch the installer.
4. After the welcome panel, you are prompted for a location to install the software. You can accept the default or click **Browse** to select a different location.
5. After the files have been moved, click **Done** to exit the installer. The BigFix Client application is now installed and will automatically begin working in the background. Repeat this process on every computer in your network that you want to place under BigFix administration.

Installing the Client using the .exe setup

You can download the BigFix Client setup in EXE format directly from https://support.bigfix.com/bes/release/

The BigFix Client installer .exe is also shipped with:

- The BigFix Installation Generator for Windows, which places it in the folder `C:\Program Files (x86)\BigFix Enterprise\BES Installers\Client`
- The BigFix Server installer for Red Hat Enterprise Linux, inside the .tgz archive folder `ServerInstaller_10.0.0.133-rhe6.x86_64/repos`

To install the BigFix Client, you also need the masthead file. You can find it on the computer where you installed the BigFix Server:

- On Windows, it is named masthead.afxm and stored in the folder `C:\Program Files (x86)\BigFix Enterprise\BES Installers\Client`
- On Linux, it is named actionsite.afxm and stored in the folder `/var/opt/BESInstallers/Client`

If you already installed the BigFix Installation Generator (Windows-only), you can run the Installation Guide (available at Start > Programs > BigFix > BigFix Installation Guide), click **Browse Install Folders** and copy the Client installer folder, which contains both the Client setup and the masthead file.

Now, you are ready to perform the following steps:

1. You might optionally include a pre-defined settings file (`clientsettings.cfg`) in the same directory as the `setup.exe` file, to create custom settings for the Windows client at installation time, for example to assign the new Client to a specific parent relay. For more information, see Windows Clients *(on page 484)*.
2. After you have copied the Client folder to the target computer, double-click **setup.exe** from that folder to launch the installer.
3. After the welcome panel, you are prompted for a location to install the software. You can accept the default or click **Browse** to select a different location.
4. After the files have been moved, click **Done** to exit the installer. The BigFix Client application is now installed and will automatically begin working in the background. Repeat this process on every computer in your network that you want to place under BigFix administration.

Via the command line

The BigFix client installer setup.exe is an MSI-based setup created with InstallShield. The .exe file can be run silently as described in Running Installations in Silent Mode. The /v"..." part of the command line contains the options and properties that are passed to the underlying MSI engine. A list of standard command line options for the Windows Installer is available at https://docs.microsoft.com/en-us/windows/win32/msi/command-line-options.

To perform a silent Client installation, follow these steps:

```
setup.exe /s /v"/L*vx! \"C:\ClientInstallLog.txt\" SETUPEXE=1
REBOOT=ReallySuppress MSIRESTARTMANAGERCONTROL=Disable /qn"
```

To log some more information specific to the InstallShield .exe wrapper:

```
setup.exe /s /debuglog"C:\ClientInstallLogIS.txt" /v"/L*vx!
\"C:\ClientInstallLog.txt\" SETUPEXE=1 REBOOT=ReallySuppress
MSIRESTARTMANAGERCONTROL=Disable /qn"
```

To change the default installation location, the appropriate form of the command is:

```
setup.exe /s /v"/L*vx! \"C:\ClientInstallLog.txt\" INSTALLDIR=
\"PathToInstallationFolder\" SETUPEXE=1 REBOOT=ReallySuppress
MSIRESTARTMANAGERCONTROL=Disable /qn"
```

Where "PathToInstallationFolder" is the full windows path to the folder where the Client is to be installed.

If you do not want the setup to start the BES Client service at the end of the installation, add STARTAGENTSERVICE=0 in the /v"..." part of the command.

Note: The Windows user running setup.exe must have Administrative privileges on the computer, must be able to write to the folder that contains the "setup.exe" and to the chosen log location, otherwise the installation might fail and create no log.

Installing the Client using the .msi setup

You can install the BigFix Client silently using either the .exe setup or the .msi setup.

To get the BigFix Client installer in MSI format, proceed as follows.

On Windows:

1. Download the BigFix Installation Generator from https://support.bigfix.com/bes/release/
2. Run the Installation Generator on a Windows machine
3. Search in the folder `C:\Program Files (x86)\BigFix Enterprise\BES Installers\ClientMSI`

On Linux:

1. Download the BigFix Server installer for RHEL from https://support.bigfix.com/bes/release/
2. Extract the .tgz archive
3. Search in the folder `ServerInstaller_10.0.0.133-rhe6.x86_64/repos/ClientMSI-10.0.0.133`

In that folder, you will find:

- BESClientSetupMSI.exe, a tool to customize the .msi (optional)
- BigFixAgent.msi, the main setup file (required)
- 1028.mst, ..., 2052.mst, the translation files for the setup (required)

To install the BigFix Client, you also need the masthead file. You can find it on the computer where you installed the BigFix Server:

- On Windows, it is named masthead.afxm and stored in the folder `C:\Program Files (x86)\BigFix Enterprise\BES Installers\Client`
- On Linux, it is named actionsite.afxm and stored in the folder `/var/opt/BESInstallers/Client`

Via the command line

To perform a silent Client installation, follow these steps:

1. If you need to customize the BigFixAgent.msi setup, use the BESClientSetupMSI.exe tool on a Windows machine as documented here *(on page 342)*. The tool needs all the files of the ClientMSI installer folder to run.
2. Copy BigFixAgent.msi setup, its .mst files and the actionsite.afxm masthead file to a folder of your target computer, for instance `C:\ClientMSI`.
3. Run the BigFixAgent.msi setup.

   You can run the .msi from the command line like this:

   ```
   msiexec.exe /i BigFixAgent.msi /L*vx! "PathToLogFile"
   INSTALLDIR="PathToInstallationFolder" TRANSFORMS=1033.mst
   REBOOT=ReallySuppress MSIRESTARTMANAGERCONTROL=Disable /qn
   ```

   L*vx! "PathToLogFile" are the logging options followed by a custom file path for the log file.

   INSTALLDIR is a property to specify the installation folder. You can omit it to install the client in the default folder.

   TRANSFORMS=transform.mst is a property to specify the transform file(s) to apply to the setup. Each .mst file contains a translation for the setup. For more information on this property, see https://docs.microsoft.com/en-us/windows/win32/msi/transforms.

   The following table describes the supplied transform files, listed by language.

Language	Transform File name
U.S. English	1033.mst
German	1031.mst
French	1036.mst
Spanish	1034.mst
Italian	1040.mst
Brazilian Portuguese	1046.mst

Language	Transform File name
Japanese	1041.mst
Korean	1042.mst
Simplified Chinese	2052.mst
Traditional Chinese	1028.mst

REBOOT=ReallySuppress and MSIRESTARTMANAGERCONTROL=Disable are properties set to avoid rebooting the system after the installation.

/qn is an option performs a silent installation. You can omit it to run the installation interactively.

This example shows how to silently install the BigFix client in the default folder in the English language:

```
msiexec.exe /i "BigFixAgent.msi" TRANSFORMS="1033.mst"
REBOOT=ReallySuppress MSIRESTARTMANAGERCONTROL=Disable /qn
```

For more information on the command line options for the MSI engine, see https://docs.microsoft.com/en-us/windows/win32/msi/command-line-options.

If you do not want the setup to start the BES Client service at the end of the installation, add STARTAGENTSERVICE=0 to your command.

Using Group Policies

You can, using Active Directory Group Policy Objects (GPO), define a policy requiring that the Client is installed on every machine in a particular group (Organizational Unit, Domain, and so on).

This policy is applied every time a user logs in to the specified domain, making it a very effective way to deploy the client if GPO is enabled. For more details consult your Active Directory administrator.

For more details about Group Policy Objects (GPO), see Using GPO to deploy the BigFix Client.

BES Client MSI Editing Tool

You can use the latest version of the BESClientSetupMSI.exe tool to operate on the BigFix Client MSI package (BigFixAgent.msi) as follows:

- Store an updated masthead in the MSI package, enabling the BigFix Client to start up using the latest configuration parameters of the BigFix deployment.
- Store the BigFix Relay information in the MSI package, leading the BigFix Client to connect to that Relay at first start. If the Relay is an authenticating Relay, it is also possible to specify the password that the BigFix Client will have to use for the Manual key exchange.
- Verify whether or not the masthead and the manual key exchange password stored in the MSI package match the specified values.

BES Client MSI Editing Tool usage for storing values:

```
BESClientSetupMSI.exe [ /relayserver1 <relay URL> [ /secureregistration
 <password> ] ] <masthead file path> <client installer path> [ /silent ]
```

BES Client MSI Editing Tool usage for verifying values:

```
BESClientSetupMSI.exe /verify [ /secureregistration <password> ] <masthead
 file path> <client installer path> [ /silent ]
```

where:

- */relayserver1 <relay URL>* is an optional argument that allows storing the information about the BigFix Relay to which the BigFix Client will have to connect to at first start. If this argument is not specified, the BigFix Client will attempt connection to the BigFix Server as specified in the masthead.
- */secureregistration <password>* is an optional argument that allows storing or verifying the password for the manual key exchange in case the specified BigFix Relay is an authenticating Relay.
- <*masthead file path*> is a required argument that specifies the absolute or relative path of the masthead file to be stored or verified in the MSI package.

- *<client installer path>* is a required argument that specifies the absolute or relative path of the MSI package to be updated or verified.
- */silent* is an optional argument that prevents the command result window from being displayed. When this argument is used, it is still possible to know whether the command completed successfully or not by looking at its exit code.
- */verify* is an alternate required argument that allows verifying whether or not the masthead and the manual key exchange password stored in the MSI package match the specified values.

Note: If you want to store the latest masthead file, you can point to the `ActionSite.afxm` file located under the `C:\Program Files (x86)\BigFix Enterprise\BES Client` folder on the BigFix Server system.

The BES Client MSI Editing Tool returns:

- Zero if the MSI package was updated or verified successfully.
- A non-zero exit code in case of failure.

Examples:

Displays usage information:

```
BESClientSetupMSI.exe
```

Stores a masthead in the MSI package in case both are located in the same directory as BESClientSetupMSI.exe:

```
BESClientSetupMSI.exe masthead.afxm BigFixAgent.msi
```

Same as above, but without displaying a success or failure message:

```
BESClientSetupMSI.exe masthead.afxm BigFixAgent.msi /silent
```

Stores the specified masthead and BigFix Relay in the specified MSI package:

```
BESClientSetupMSI.exe /relayserver1
 http://relay_host_or_IP:52311/bfmirror/downloads/ <path>\masthead.afxm
 <path>\BigFixAgent.msi
```

Stores the specified masthead, BigFix Relay and manual key exchange password in the specified MSI package:

```
BESClientSetupMSI.exe /relayserver1
 http://relay_host_or_IP:52311/bfmirror/downloads/ /secureregistration
 password <path>\masthead.afxm <path>\BigFixAgent.msi
```

Verifies whether or not the masthead stored in the MSI package matches the specified one:

```
BESClientSetupMSI.exe /verify <path>\masthead.afxm <path>\BigFixAgent.msi
```

Verifies whether or not the manual key exchange password stored in the specified MSI package is equal to "mYp@ssw0rd":

```
BESClientSetupMSI.exe /verify /secureregistration mYp@ssw0rd
 <path>\masthead.afxm <path>\BigFixAgent.msi
```

Embedding in a Common Build

If your organization employs a specific build image or common operating environment (COE) on a CD or image that is used to prepare new computers, you can include the Client in this build.

To create the image, follow these steps:

For Windows operating systems

1. Install the client on the computer to be imaged. The BigFix client immediately attempts to connect to the server. If it successfully connects to the server, it is assigned a **ComputerID**. This ComputerID is unique to that particular computer, so it should *not* be part of a common build image. The next steps delete this ID.

2. Stop the client by opening the Windows Services dialog and stopping the **BES Client service**.
3. Delete the computer-specific identifier (computer ID) by opening the registry to `HKLM\Software\Wow6432Node\BigFix\EnterpriseClient\GlobalOptions` and deleting the values `ComputerID`, `RegCount`, and `ReportSequenceNumber`.
4. Delete the __BESData folder from the installation directory of the BigFix client.
5. Delete the "KeyStorage" folder from the installation directory of the BigFix client.

The BigFix Client is now ready to be imaged.

Note: If the Client is started again for any reason (*including a system restart*), it re-registers with the server and **you will need to perform steps 2 to 3 again**. The Server has built-in conflict detection and resolution so if for any reason you fail to delete the ID, the Server can detect that there are multiple Clients with the same ComputerID and forces the Client to re-register to ensure that everything works normally. However, it is advisable to perform the steps above to avoid having a grayed-out Client (the first imaged computer) in the computer list in the Console.

For more information, see Avoiding duplicates when a Client is restored (on page 346).

For Linux operating systems

1. Install the client on the computer to be imaged.
2. Stop the client by running `/etc/init/besclient stop`.
3. Delete the computer-specific identifier from the `.config` file to prevent all copies of the machine from registering with the same client ID to the server.
4. Delete the __BESData folder. The default location is `/var/opt/BESClient`.
5. Delete the "KeyStorage" folder. The default location is `/var/opt/BESClient`.

The BigFix Client is now ready to be imaged.

For more information, see Avoiding duplicates when a Client is restored (on page 346).

For Macintosh operating systems

1. Install the client on the computer to be imaged.
2. Stop the client by using `sudo "/Library/BESAgent/BESAgent.app/Contents/MacOS/BESAgentControlPanel.sh" -stop`.
3. Delete the computer-specific identifier to prevent all copies of the machine from registering with the same client ID to the server.
    - If they exist, remove **RegCount**, **ReportSequenceNumber**, and **ComputerID** from the client preferences folder: `/Library/Preferences/com.bigfix.besagent.plist`.
    - Delete the __BESData folder. The default location is `/Library/Application Support/BigFix/BES Agent`.
    - Delete the "KeyStorage" folder. The default location is `/Library/Application Support/BigFix/BES Agent`.

The BigFix Client is now ready to be imaged.

For more information, see Avoiding duplicates when a Client is restored (on page 346).

Avoiding duplicates when a Client is restored

When the BigFix client is rolled back or restored from a snapshot, the next time it registers itself with the BigFix server, it receives a new Computer ID. The BigFix client with the old Computer ID becomes inactive because it does not report with the same Computer ID. As a result, the BigFix console shows duplicated entries for the same computer. Additionally, the affected computer loses all information: action history, discovery, and so on.

To avoid this situation, you must first back up some client data of the computer whose identity you want to preserve and then follow the procedure described below.

Client data backup

1. Stop the BigFix client service.
2. Store the following data.

a. The registry key value for the `ComputerID` from the following location.
   - Windows: `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\EnterpriseClient\GlobalOptions`
   - Linux: `/var/opt/BESClient/besclient.config` under the section `Software\BigFix\EnterpriseClient\GlobalOptions`
   - MacOS: `/Library/Preferences/com.bigfix.BESAgent.plist`

b. The `BES Client\KeyStorage` folder from the default installation directory of the BigFix client.

Procedure to keep the client identity

1. The BigFix server can match the data that is stored aside to the BigFix client that is reinstalled or reverted from a snapshot when the **ClientIdentityMatch** parameter of the BigFix server is set to 100. By default, the parameter is set to 0. To change the value of the parameter, go to the computer on which the BigFix server is installed and perform the following steps:
   a. Windows: Go to **Start > BigFix Administrative Tool > Advanced Options** and set the value of the ClientIdentityMatch parameter to 100.
   b. Linux: Run the following command.

```
./BESAdmin.sh -setadvancedoptions
 -sitePvkLocation=/root/backup/license.pvk
-sitePvkPassword=pippo000 -update clientIdentityMatch=100
```

   Note: For Solaris SPARC systems, the client identity match feature requires the logical domains to be enabled. If they are not enabled, an identity match cannot be performed because the “uuid of hardware” and “serial of hardware” properties do not return the information on which the feature relies.

2. Ensure that you complete this action before you install the BigFix client on the computer whose identity you want to preserve.
3. Install the BigFix client on the computer whose identity you want to preserve. After the installation of the BigFix client completes, the computer automatically registers with the BigFix server and receives a unique Computer ID.

4. Stop the BigFix client.
5. Move to the following location:
    - Windows: `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\EnterpriseClient\GlobalOptions`
    - Linux: `/var/opt/BESClient/besclient.config` under the section `[Software\BigFix\EnterpriseClient\GlobalOptions]`
    - MacOS: `/Library/Preferences/com.bigfix.BESAgent.plist`

    and remove the values for the following parameters:
    - `RegCount`
    - `ComputerID`
    - `ReportSequenceNumber`
6. Delete the `__BESData` and `KeyStorage` folders from the installation directory of the BigFix client.
7. Restore the previously stored `ComputerID` and `KeyStorage`.
8. Start the BigFix client.

Enabling encryption on Clients

When installed, you can set up your Clients to encrypt all outgoing reports to protect data such as credit card numbers, passwords, and other sensitive information.

Note: You must have encryption enabled for your deployment before enabling it for your Clients. In particular, for the required option, your clients will become silent if you enable them without first setting up your deployment.

To enable encryption, follow these steps:

1. Open the BigFix Console.
2. From the **BigFix Management** Domain, open the **Computer Management** folder and click the **Computers** node.
3. Select the computer or set of computers that you want to employ encryption for.
4. From the right-click context menu, select **Edit Computer Settings**.
5. From the **Edit Settings** dialog, click **Add**.

6. In the **Add Custom Setting** dialog, enter the setting name as

   _BESClient_Report_Encryption (note the underline starting the name).

   There are three possible values for this setting:

   required

   Causes the Client to always encrypt. If there is no encryption certificate available in the masthead or if the target computer (Relay or Server) cannot accept encryption, the Client will not send reports.

   optional

   The Client encrypts if it can, otherwise it sends its reports in clear-text. This is the default value for this setting.

   none

   No encryption is done, even if an encryption certificate is present. This allows you to turn off encryption after you enable it.
7. Click **OK** to accept the value and **OK** again to complete the setting. You must enter your private key password to deploy the setting action.

For additional information about encryption, see Encryption (on page 354).

Chapter 11. BigFix Administration Tool

The BigFix Administration Tool, also called BESAdmin, is the tool we use to perform configuration changes and maintenance operations.

On Windows, it has both a GUI (graphical user interface) and a CLI (command line interface). On Linux, it only has a CLI.

BESAdmin Windows GUI

The Installer automatically creates the BigFix Administration Tool when it installs the other components of the Console program.

To prevent conflicts, avoid running BESAdmin commands concurrently. Only one BESAdmin instance should run at a time within a BigFix environment.

Starting with version 11.0.7, opening the BESAdmin GUI on Windows will display an error message and exit if another BESAdmin GUI is already open or a BESAdmin command is running on the same computer. This error message will not be logged.

This program operates independently of the Console and is intended for Administrative Operators only. You can find it from the Start menu: **Start > All Programs > BigFix > BigFix Administration Tool.** To run the program, you must first browse to the private key (license.pvk).

You can also change your administrative password through this interface. After you have selected the private key file, click **OK** to continue. You must supply your private key password to proceed.

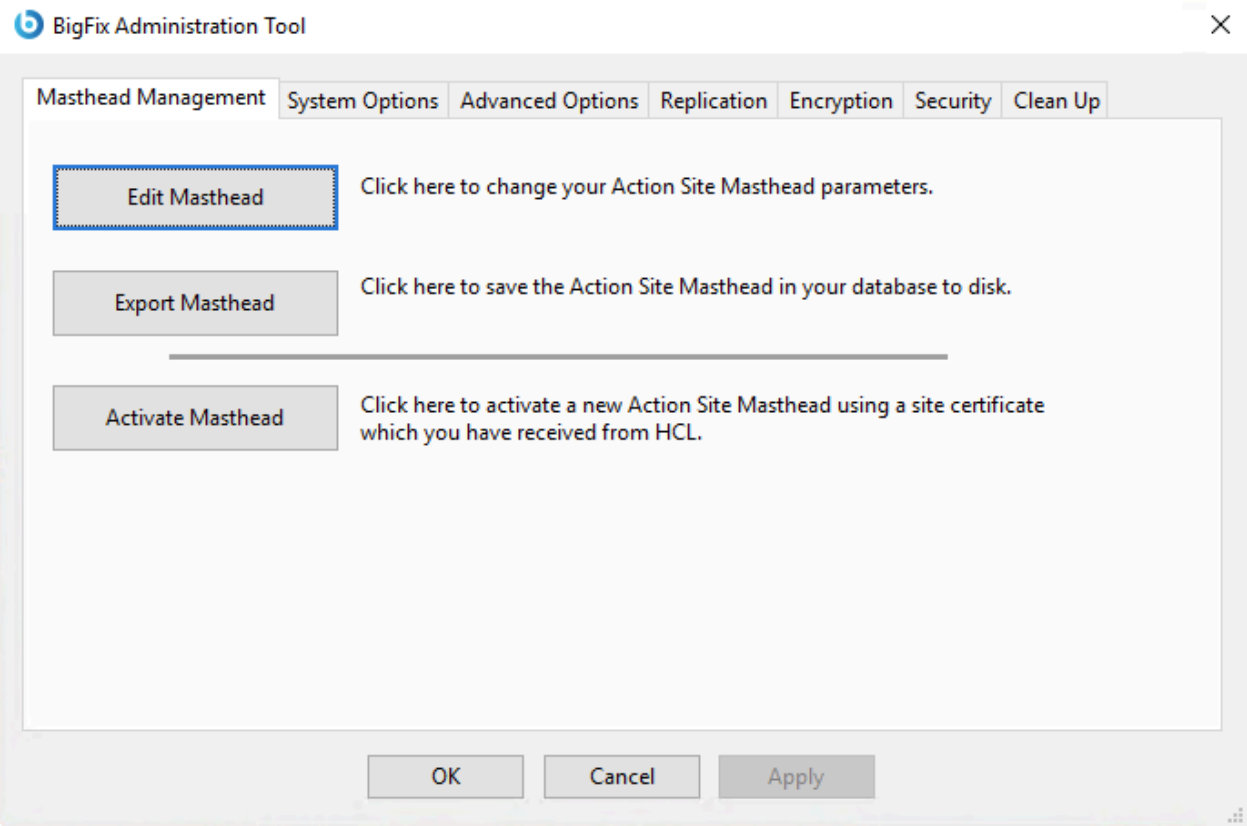


Note: When you change the private key password you change only the password of the local file; the other DSA BigFix servers are not updated. They continue to use their own license file and password unless it is replaced from the changed license file.

Use the BigFix console to perform the user management tasks.

Masthead Management

Click the first tab to view the **Masthead Management** dialog.



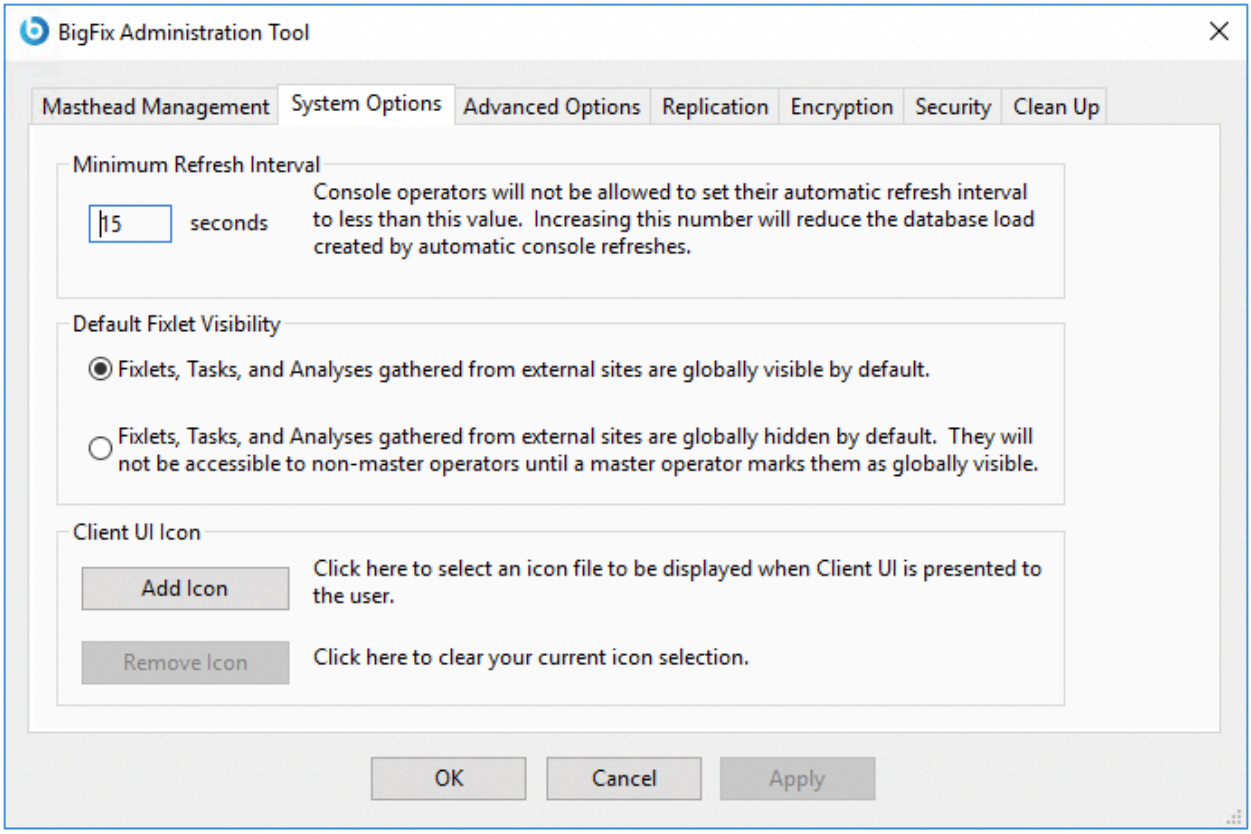
If you do not yet have a masthead, which is required to run the Console, this dialog provides an interface to **Request** and subsequently **Activate** a new masthead. If you have an existing masthead, you can edit it to change gathering intervals and locking. You can also export your masthead, which can be useful if you want to extend your BigFix network to other servers.

System Options

The second tab opens the **System Options** dialog.

The first option sets a baseline minimum for refresh intervals. This refers to the Fixlet list refresh period specified in the Preferences dialog of the Console. The default period is 15 seconds, but if your network can handle the bandwidth, you can lower this number to make

the Console more responsive. Conversely, if your network is strained, you might want to increase this minimum.



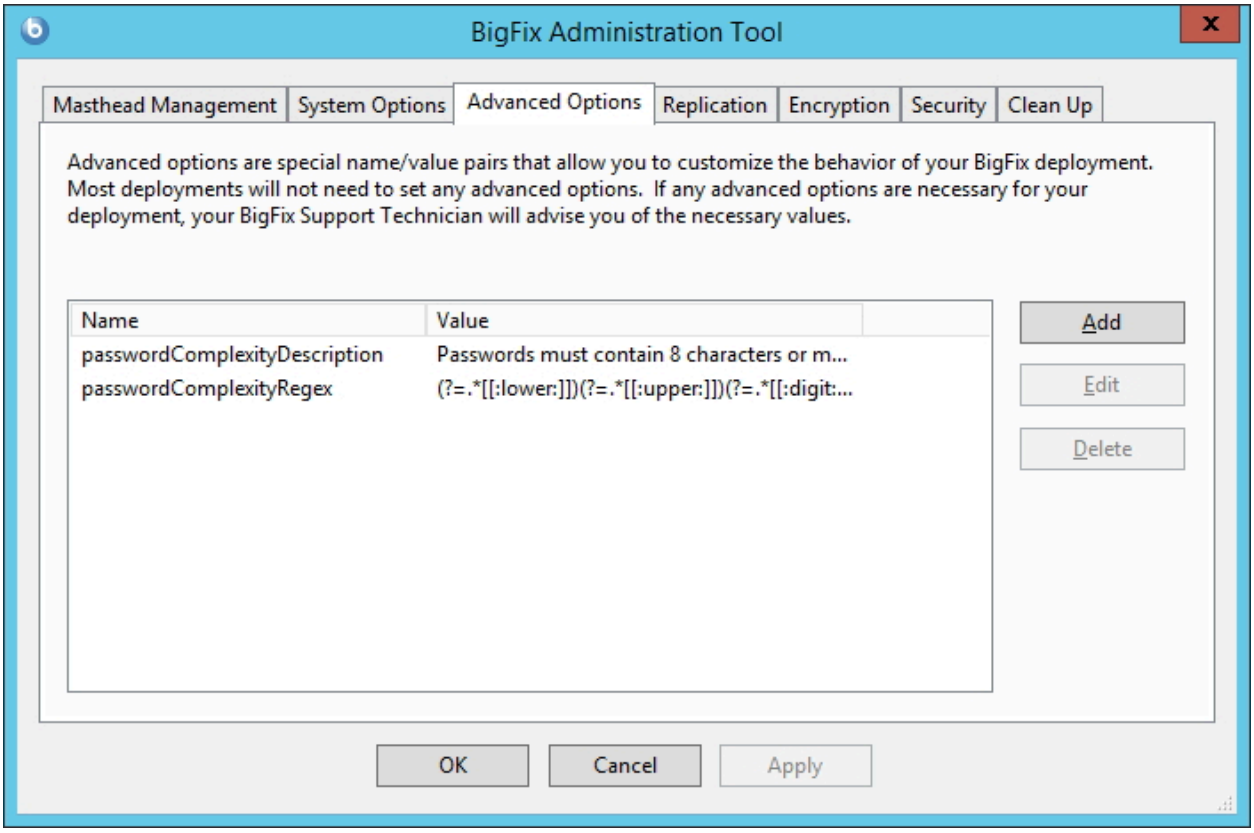
Use this dialog to set the default visibility of external sites. These sites are, by default, globally visible to all Console operators. To give you extra control, you can set the visibility to hidden, and then adjust them individually through the Console. You must be an administrator or a master operator to make these hidden sites become visible.

Use this dialog to add your own logo to any content that is presented to the user on the Client system. Branding can be important to reassure your users that the information has corporate approval.

Advanced Options

The third tab opens the **Advanced Options** dialog.

This dialog lists any global settings that apply to your particular installation.



These options are name/value pairs, and are typically supplied by your HCL Support. As an example, if you are subscribed to the Power Management site, one of these options allows you to enable the WakeOnLAN function.

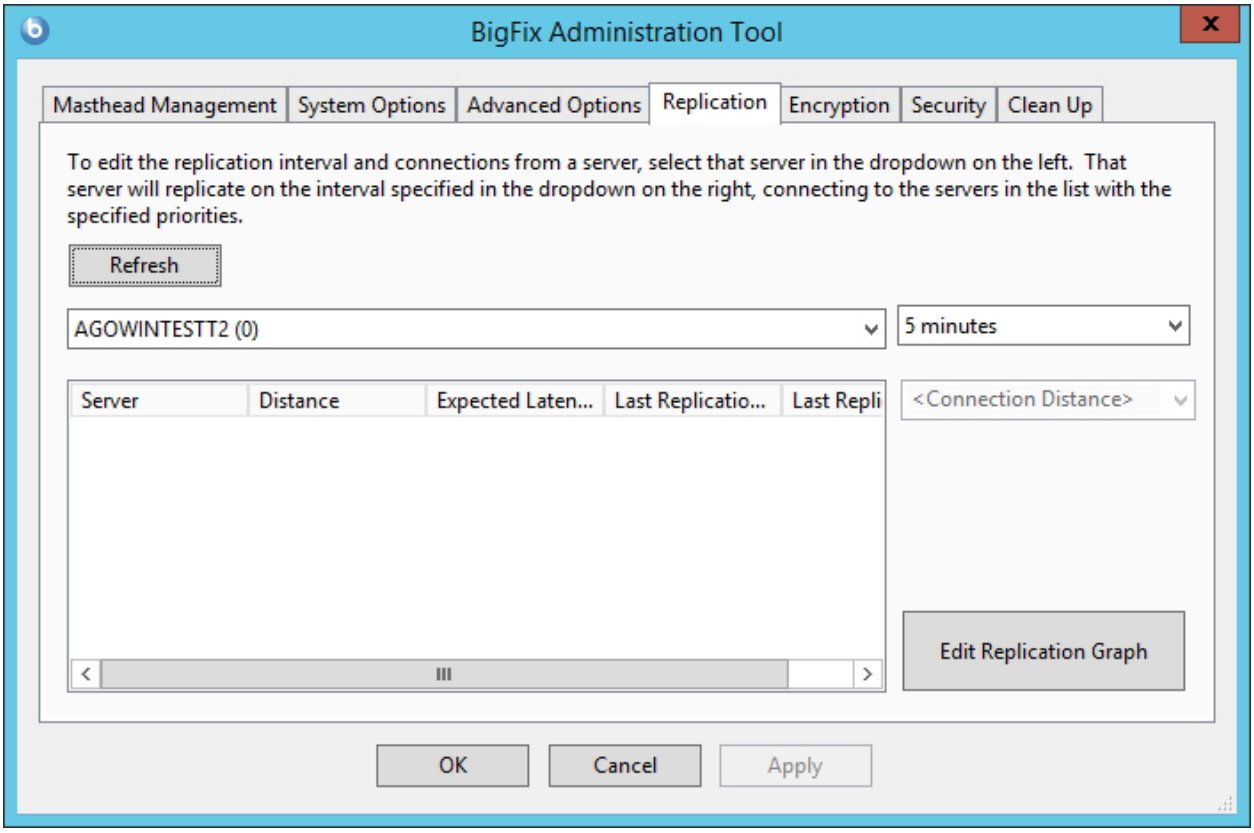
Some of these settings overlap with special registry keys that can be set to influence the behavior of individual consoles. As a rule of thumb, if the setting represents a boolean option, the consoles will have the default behavior unless either the registry or the Advanced Deployment Options specify the non-default behavior.

For a list of available options that you can set, see List of advanced options.

Replication

The fourth tab opens the **Replication** dialog.

Use this dialog to visualize your replication servers.

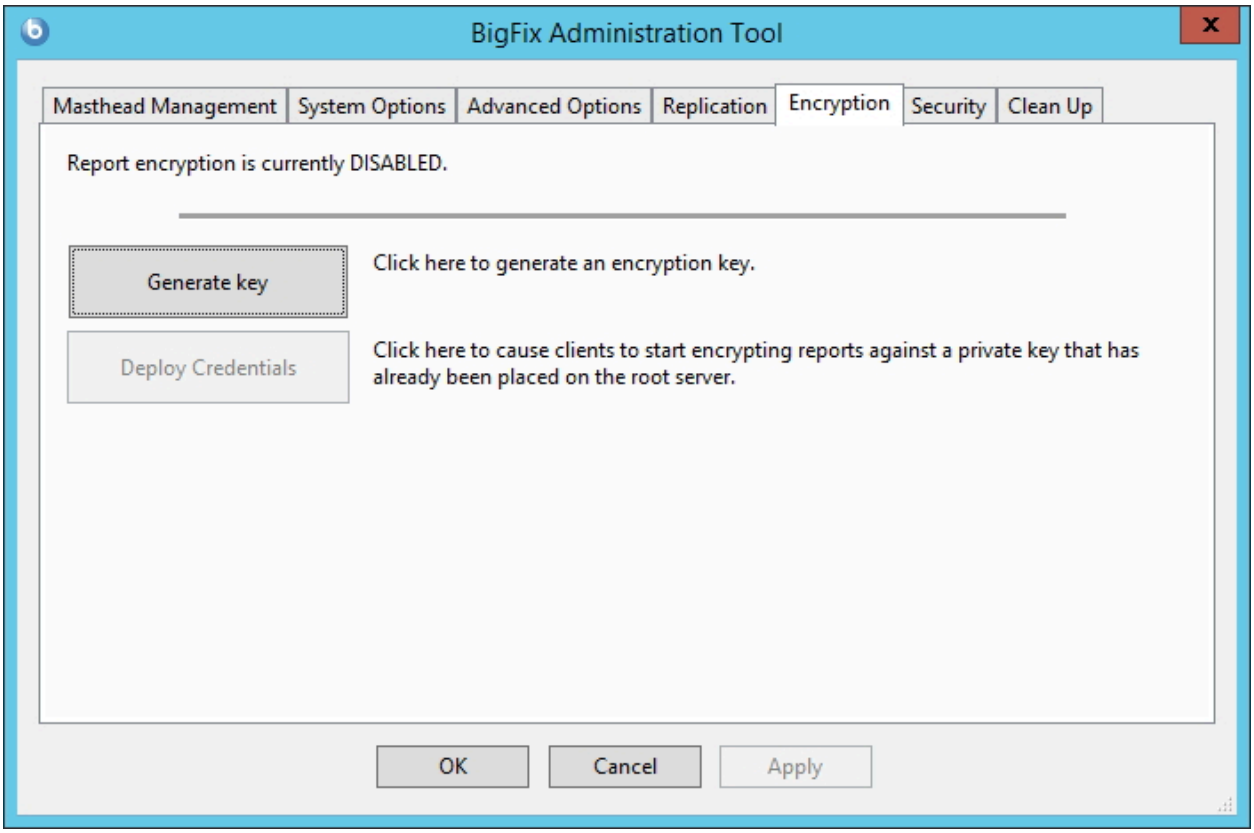


For more information, see Managing Replication (DSA) on Windows systemsthe Configuration Guide..

Encryption

The fifth tab opens the **Encryption** dialog. Use this dialog if you want that the Client encrypts reports to be sent to the server.

This is useful if the reports contain confidential information. You can use this tab to generate a new encryption key or to disable encryption altogether.



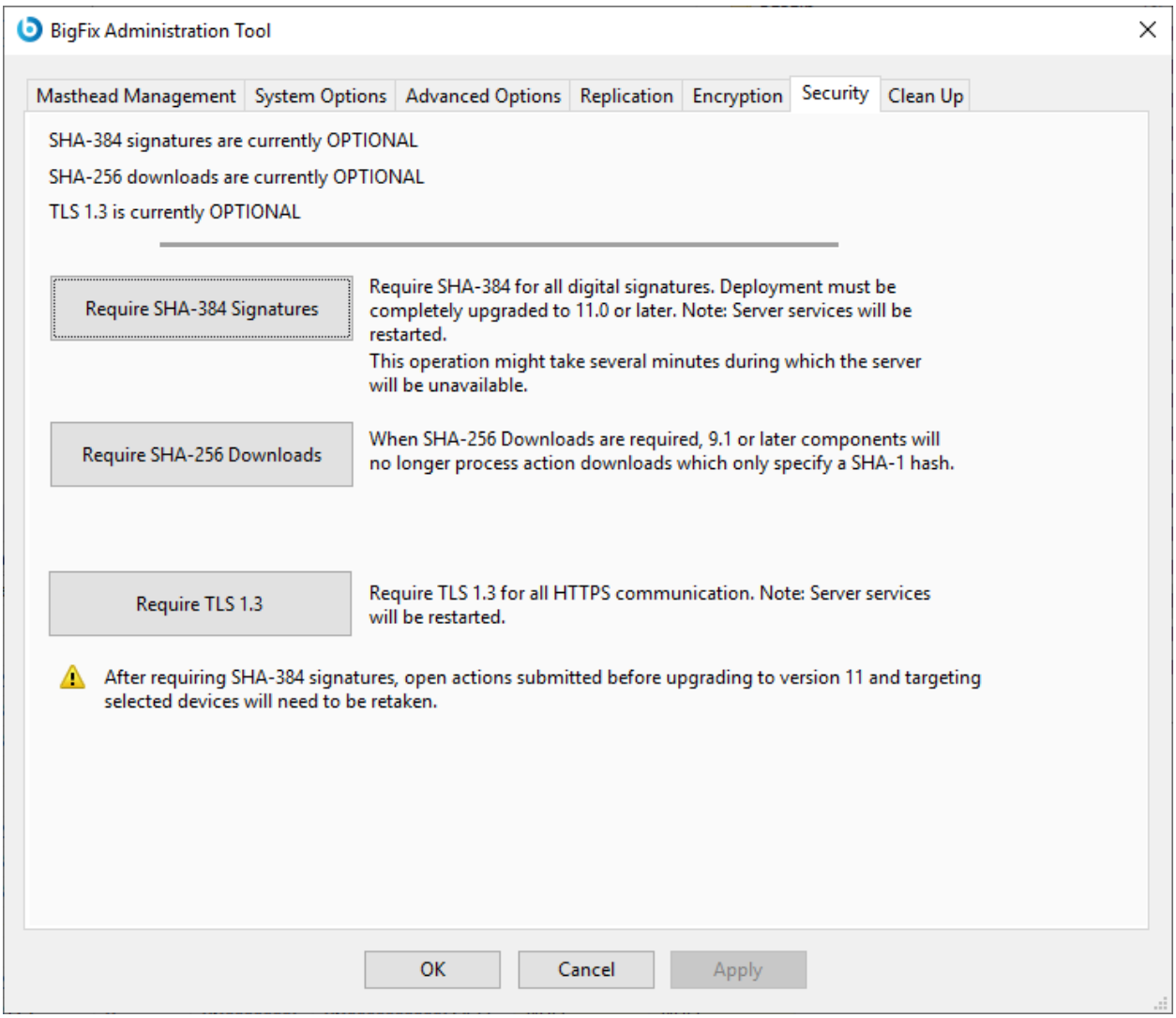
If you click **Generate Key**, the server creates a public key and a private key. The private key is stored in the database on the server. The public key is stored in the master actionsite. As soon as the clients receive the master actionsite, they start to encrypt the reports with the public key. On the server, the reports are decrypted using the private key.

If you configured your environment so that the top level relays are in a secure location with the server, you can delegate the responsibility to decrypt reports to the relays to reduce the workload on the server. This is the list of steps to run if you want to set this configuration:

1. In the **Encryption** tab, generate the key pair, private and public, on the server.
2. Manually copy the private key on the relays to delegate for decryption.
3. In the **Security** tab, click **Enable Enhanced Encryption**. After you click that button, the master actionsite is sent across the BigFix network and the clients start to encrypt reports with the public key.
4. When a relay that has the private key, receives the encrypted reports, it decrypt them and forward the reports in clear text to the server.

Security

The sixth tab opens the **Security** dialog.



Click the **Require SHA-384 Signatures** button to adopt the SHA-384 cryptographic digest algorithm for all digital signatures for content verification.

To require SHA-384 ensure that the following conditions are satisfied:

- Ensure that the updated license was gathered. After upgrading to Version 11, the BigFix Server will gather a new license with the SHA-384 certificate.
- Unsubscribe from all external sites that do not support SHA-384. BigFix Administration Tool provides the list of all subscribed sites for which the SHA-384 signature is not present.
- After upgrading to Version 11, existing WebUI certificates must be recreated to include the SHA-384 signature.

Note: If you use this setting you break the compatibility with BigFix version 10.0 or earlier version components.

Note: When you change the required type of signature, it is recommended that you take a new server backup.

Click the **Require SHA-256 Downloads** button to adopt the SHA-256 integrity check for file download integrity check.

Click the **Require TLS 1.3** button to use the TLS 1.3 protocol only and to disable the TLS 1.2 protocol for communications among the BigFix components.

Note: If you use this setting you break backward compatibility with BigFix version 10.0 or earlier version components.

Note: If you use this setting the Console SAML authentication works only on a Windows version that supports the TLS 1.3. For more details, see Assumptions and requirements.

To enforce SHA-384 on a Disaster Server Architecture (DSA) server in Linux environments:

You must enable it only on the primary server by running the command:

./BESAdmin.sh -securitysettings -sitePvkLocation=<path+license.pvk> -requireSHA384Signatures

You do not have to enable it on the replica servers. You might be requested to run the command:

./BESAdmin.sh syncmastheadandlicense -sitePvkLocation=<path+license.pvk> [-sitePvkPassword=<password>]

on the replica servers to ensure that the updated action site is propagated to the replica servers as well.

To enforce SHA-384 on a Disaster Server Architecture (DSA) server in Windows environments:

You must enable it only on the primary server by following the procedure described in On Windows Systems.

You must run the .\BESAdmin.exe command on the replica servers to ensure that the updated action site is propagated to the replica servers as well.

Click the **Require SHA-256 Downloads** button to change all download verification to use only the SHA-256 algorithm. Existing custom actions might need to be edited to conform to the prefetch action script syntax.

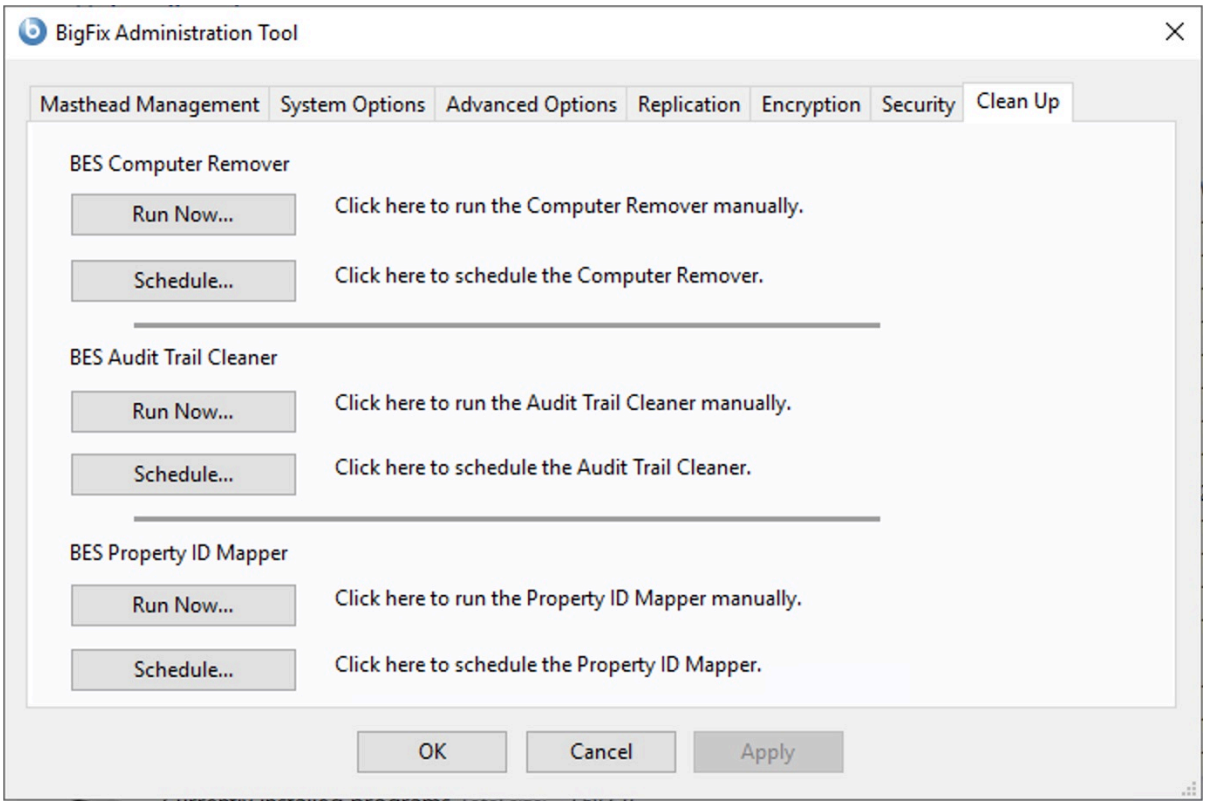


Note: If you do not select this option, the file download integrity check is run using the SHA-1 algorithm.

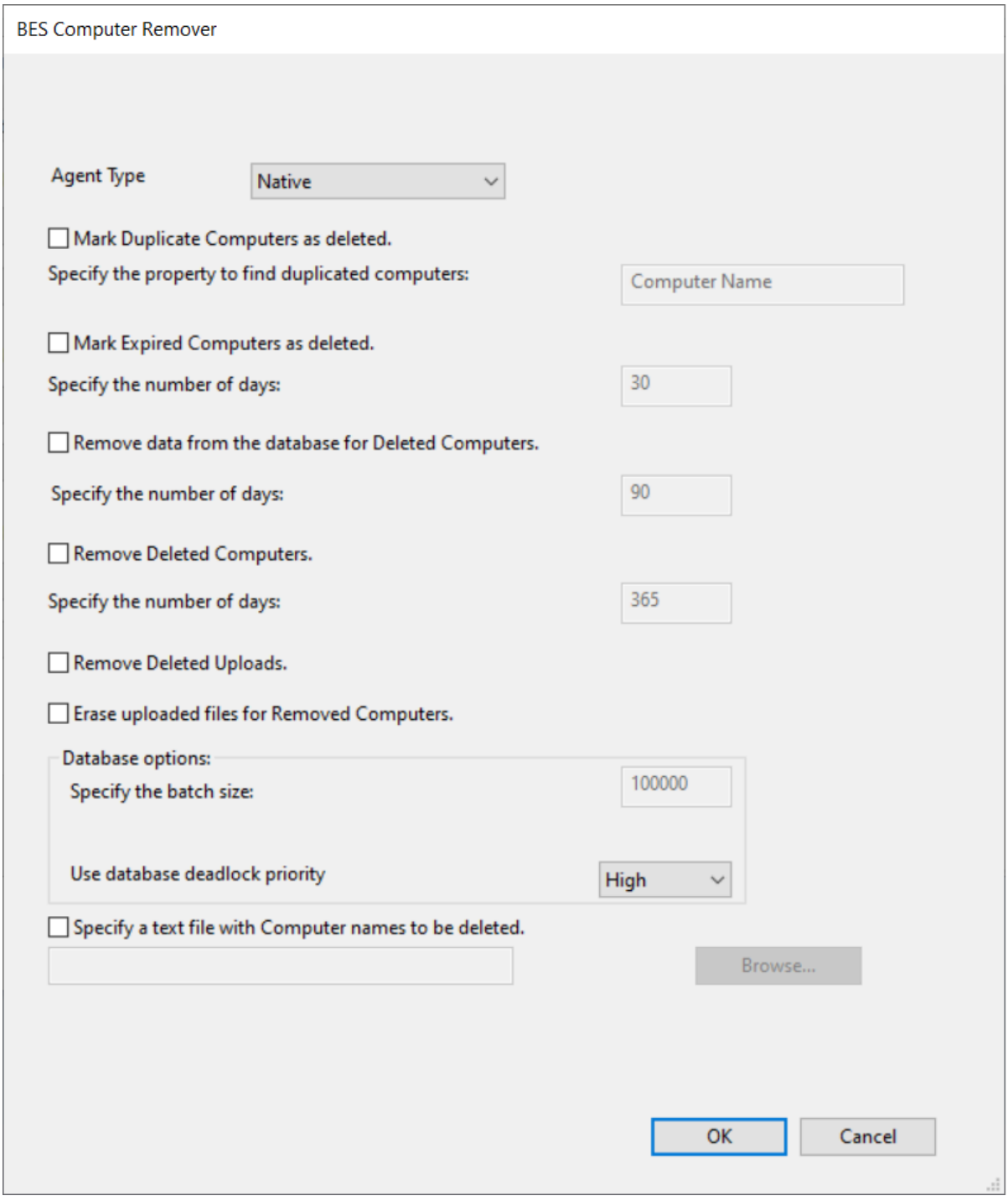
For more information about the BigFix Enhanced Security feature, the supported security configuration and enhanced security requirements evaluation, see Security Configuration Scenarios.

Computer Remover

Click **Run Now** in the `BES Computer Remover` section to remove computers.



The following window is displayed:



The tool separates the deletion operations to mark a computer as being deleted in the database and to be removed from BigFix Console or Web Reports. Marking a computer as being deleted does not remove any data from the database and if the computer reports

back in, it is restored. The tool also supports the removal of data about computers from the database to free disk space and allow the database to run faster.



Note: Up to Bigfix version 10.0.7, the clean up process affects all computers. From version 10.0.8 and later, the process requires the **Agent Type** field to be set: only the devices that match the specified **Agent Type** are targeted by the clean up procedure.

You can specify to remove the following data:

- Duplicated computers by selecting **Mark Duplicate Computers as deleted** and specifying the name of the duplicated computer. The computer is marked as deleted if a computer exists with the specified property.

  Note: To successfully delete the duplicated computer, specify in the text field of the `BES Computer Remover` panel the property name in English, for example the computer name. The `BES Computer Remover` tool, like all other Clean Up tools, interacts directly with the database, and the property names are stored inside the database in English. For this reason, the property names must be specified in English.

- Expired computers by selecting **Mark Expired Computers as deleted** and specifying a number of days for the native computers and a number of hours for the proxied computers. The computers are marked as deleted if they have not sent any reports after the specified amount of time.
- Deleted Computers by selecting **Remove data from the database for Deleted Computers** and specifying a number of days for the native computers and a number of hours for the proxied computers, and the batch size. The data from computers that are already marked as deleted and have not sent any reports after the specified number of days is removed.
- Removed computers by selecting **Remove Deleted Computers** and specifying a time frame. The computers are those ones marked as deleted since the specified amount of time. Up to BigFix Version 10.0.7, the minimum value is 30 days. With

BigFix Version 10.0.8 and later, the minimum number of days is set to 7 for the native computers, and the time interval is expressed in hours for the proxied devices. The minimum allowed value in hours is 24.

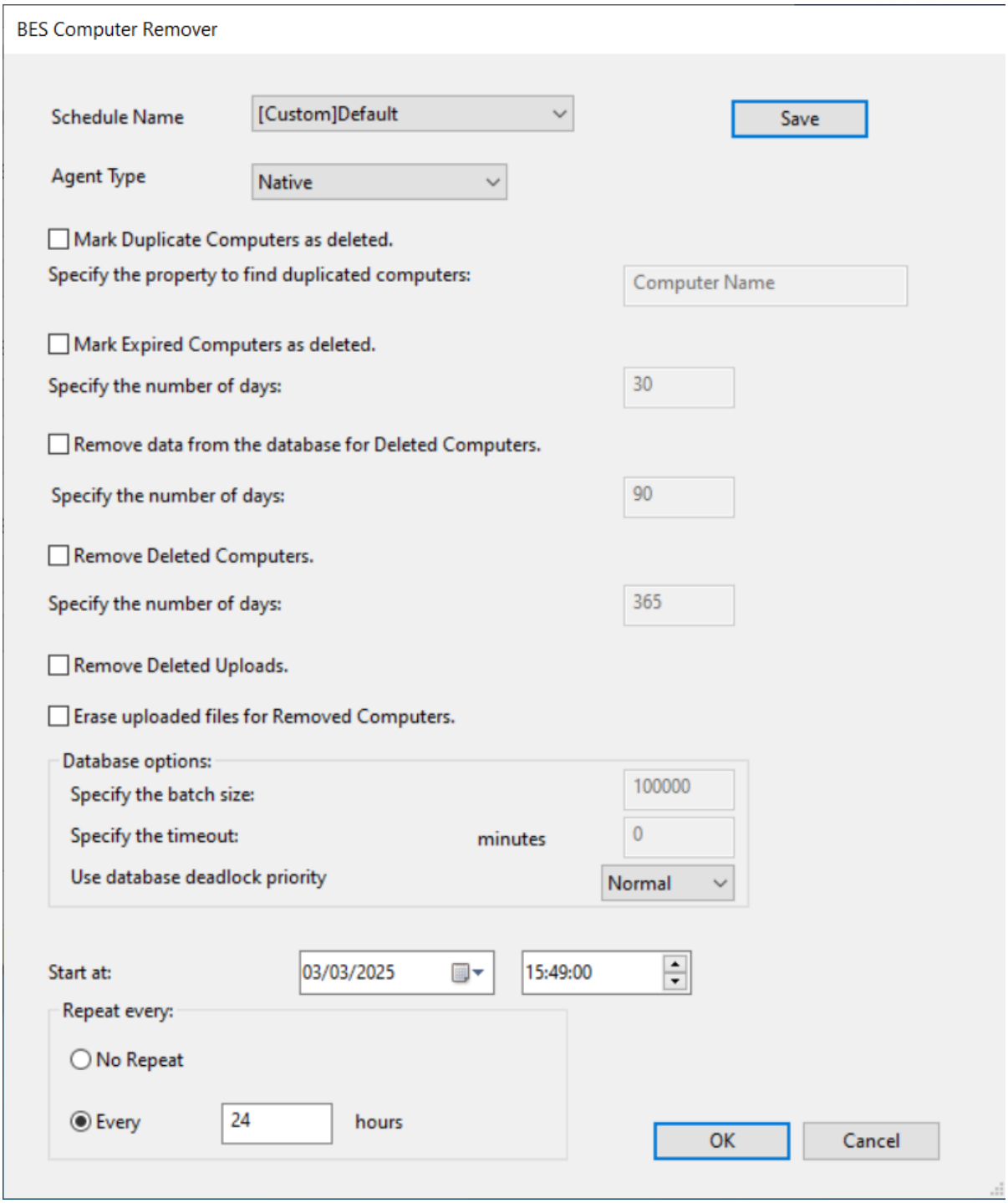
- Uploaded files by selecting **Remove Deleted Uploads**. The uploaded files are those ones marked as deleted. This option does not apply for non-native agents.
- Uploaded files related to deleted computers by selecting **Erase uploaded files for Removed Computers**. The uploaded files are those ones of clients whose definition has been removed from the database. This option does not apply for non-native agents.
- **Use database deadlock priority** - You can set the priority in case of transaction deadlocks in the Microsoft SQL Server database. Available values are Low, Normal and High. The default value is Normal. Setting the value to High and Low influences how Microsoft SQL Server resolves the potential deadlocks. Setting the priority to **High** means that the BigFix Administration Tool transaction is more likely to survive in case of deadlocks. Setting the priority to **Normal** means that Microsoft SQL Server will decide which transactions will survive based on rollback costs. Setting the priority to **Low** means that the BigFix Administration Tool transaction is more likely to be terminated in case of deadlocks.
- Specific computers by selecting **Specify a text file with Computer names to be deleted** and entering the name of a text file containing a list of computer names separated by new lines. This option applies only to native agents, and is not available if you schedule the computer deletion operation.



Note: When removing computer names by using a text file, you can use the "%" character as a wildcard in the text file, and the tool will remove all computer names starting with the text string that you specified before the "%" character.

To automate the process of removing computers from the BigFix Console and deleting the data from the database, you can schedule the process by clicking **Schedule** in the `BES Computer Remover` section of the BigFix Administration Tool.

The following window is displayed:



The time frames are expressed in days for the native agent types, and in hours for the proxied agent types.

You can specify the date and time to start the computer deletion and also a period of time to make this deletion operation recursive.

In **Specify the timeout**, you can also specify a timeout value in minutes by which the scheduled process must complete.

In **Use database deadlock priority**, you can set the priority in case of transaction deadlocks in the Microsoft SQL Server database. Available values are Low, Normal and High. The default value is Normal. Setting the value to High and Low influences how Microsoft SQL Server resolves the potential deadlocks. Setting the priority to **High** means that the BigFix Administration Tool transaction is more likely to survive in case of deadlocks. Setting the priority to **Normal** means that Microsoft SQL Server will decide which transactions will survive based on rollback costs. Setting the priority to **Low** means that the BigFix Administration Tool transaction is more likely to be terminated in case of deadlocks.

Using this window, you can:

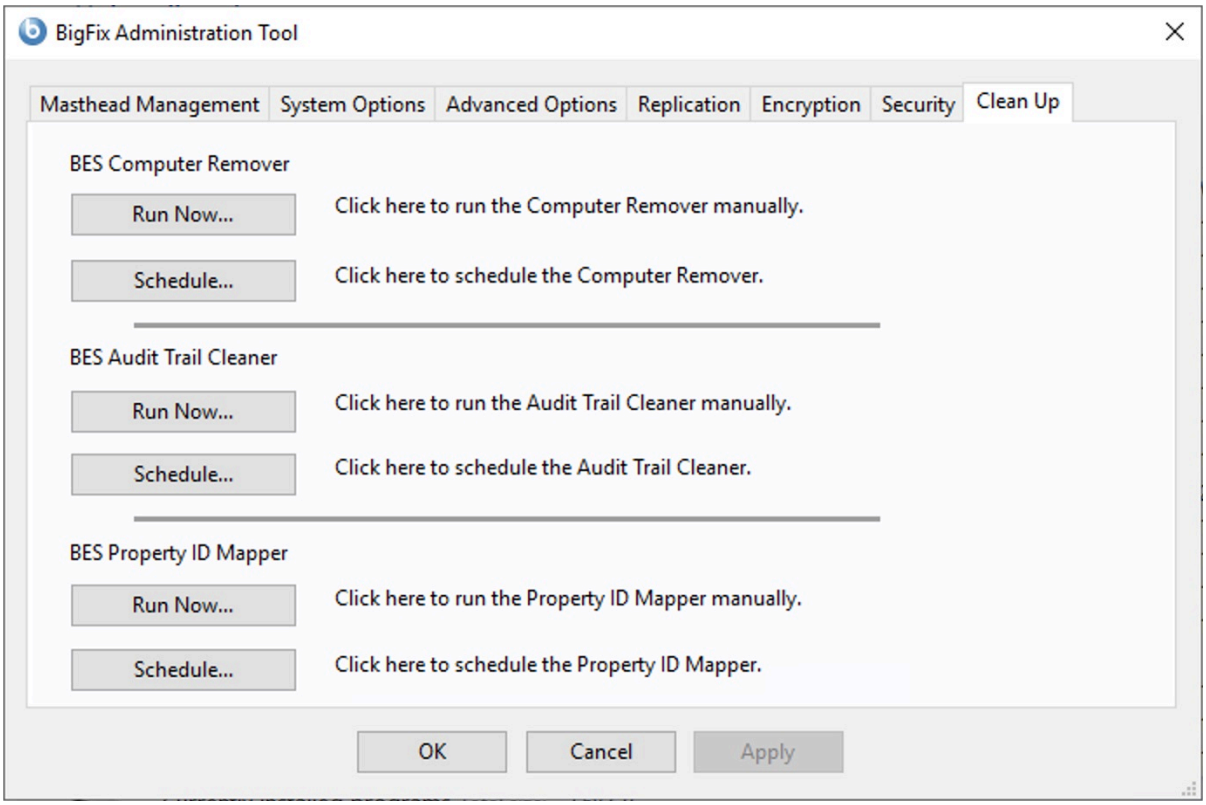
- Specify the agent type as needed (only BigFix Version 10.0.8 and later).
- Save a new schedule task, which is saved into the BigFix Database. With BigFix version 10.0.8 and later, multiple tasks can be saved, provided that a unique Schedule Name is specified. BigFix versions from 10.0.1 to 10.0.7 allow only one saved task.
- Modify the saved schedule task as needed. With BigFix version 10.0.8 and later, a Delete button is provided to remove the saved schedules from the database.

Audit Trail Cleaner

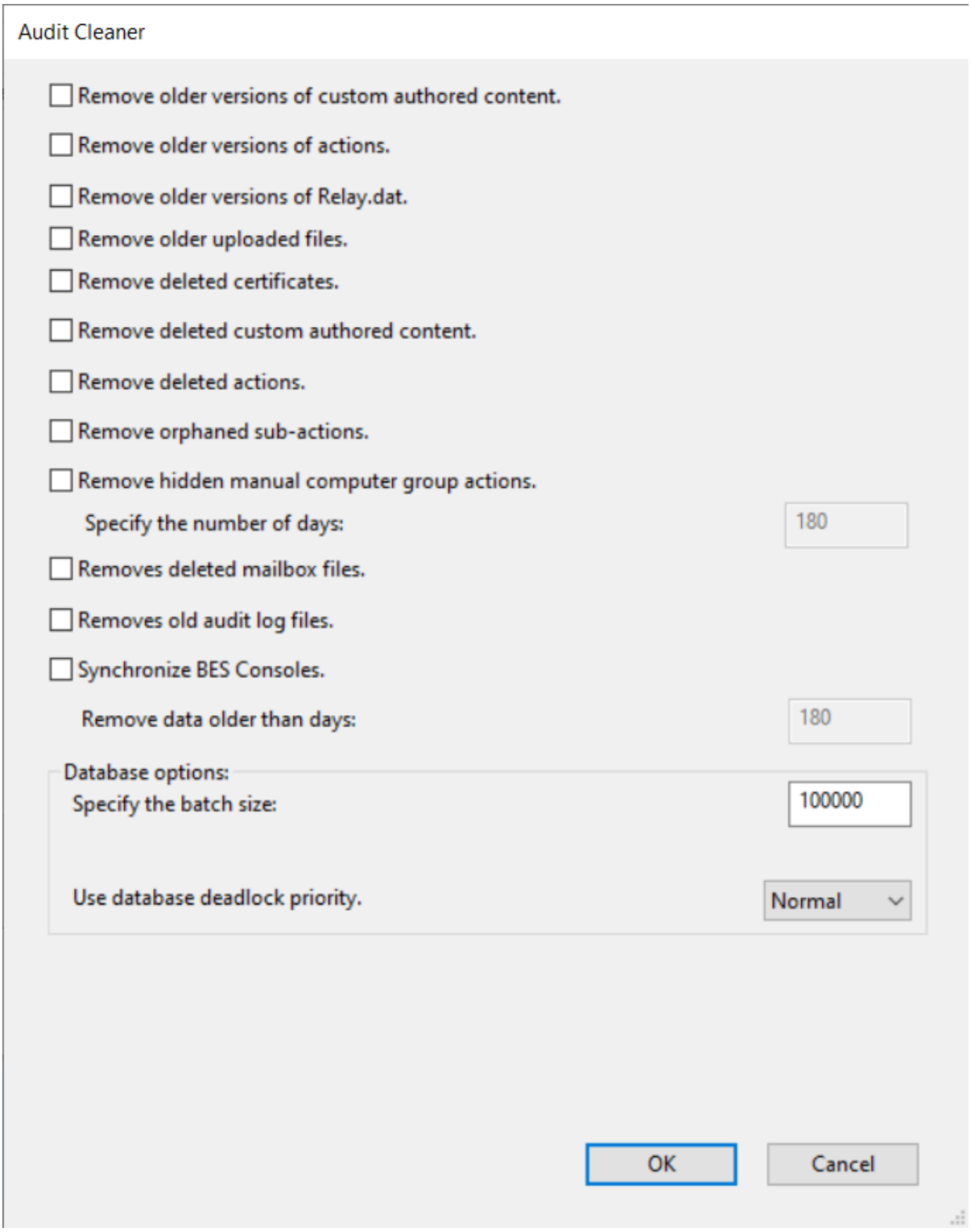
Use this tool to clear historical data, used as audit trail, from the bfenterprise database. The audit trail contains deleted and previous versions of Fixlets, tasks, baselines, properties, mail boxes, actions, and analyses.

The audit trail is not used by BigFix and can be deleted to reduce the database size. Before running this tool and removing the audit trail from the product database, create a historic archive of the current database and save it to a secure location to preserve the audit trail history.

Click **Run Now** in the `BES Audit trail Cleaner` section to clear data:



The following window is displayed:



The tool can count and delete the following sets of data:

- **Remove older versions of custom authored content** - Every edit to Fixlets, tasks, baselines, and analyses creates a new version; the earlier versions can be deleted.
- **Remove older versions of actions** - Any time you stop or start an action a new version is created; the earlier versions can be deleted.
- **Remove older versions of Relay.dat** - Any time you install or uninstall a new relay a new version is created; the earlier versions can be deleted.
- **Remove older uploaded files** - Removes the old files uploaded by the Archive Manager on the BigFix server. This option deletes the old files after an expiration period (default 180 days) from when they were uploaded.
- **Remove deleted certificates** - The old certificates that were deleted.
- **Remove deleted custom authored content** - When you delete a Fixlet, task, baseline, or analysis using the console, the data is marked as deleted in the database and preserved. The deleted content, including all the earlier versions, and the corresponding client reports can be deleted.
- **Remove deleted actions** - When you delete an action using the console, the data is marked as deleted in the database and preserved. The deleted actions, including all the earlier versions, and the corresponding client reports can be deleted.
- **Remove orphaned sub-actions** - The orphaned sub-actions from multiple action groups that were deleted.
- **Remove useless action results** - Earlier versions of BigFix V7.2.4.6 might cause clients to report ActionResults which were not used in any way but would use up space in the database. These useless ActionResults can be deleted.
- **Remove hidden manual computer group actions** - Manual Computer Groups create hidden actions that add and remove computers to and from groups and the actions can build up over time. This option deletes actions after an expiration period (default 180 days) from when they were created.
- **Remove deleted mailbox files** - Deleted Mailbox Files are stored in a table in the database and can be removed.
- **Remove old audit log files** - Remove old server_audit.log files to prevent the server running out of disk space. If the **_BESAdminAudit_Logging_LogDirectoryPath** client setting is not used, on Windows the audit log files may not be deleted from the Audit

Cleaner **Removes old audit log files** function. For more details, refer to Configuring logs.

- **Synchronize BES Consoles** - The BigFix Console maintains a local cache of the database that will become unsynchronized when data is removed with this tool. To prevent this from happening, the tool sets a flag in the database to force all BigFix Consoles to re-load the cache when they next start up.

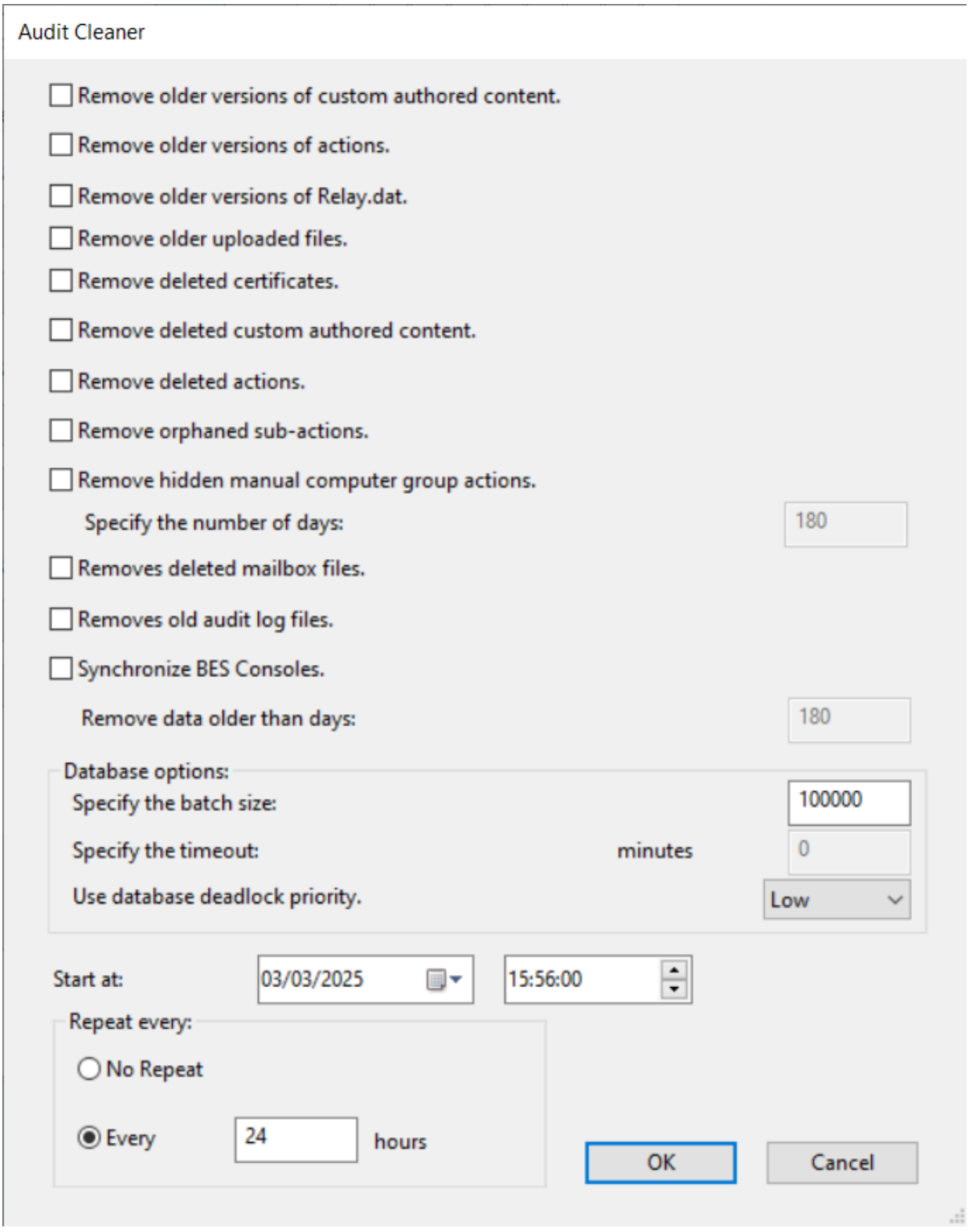
Remove data older than days - You can specify to remove data earlier than a specified date. The default value is 99 days.

Specify the batch size - Deleting large sets of data causes the SQL transaction log to quickly increase in size, the log will temporarily be larger than the data being removed until the database is shrunk. You can also specify batched deletions to remove results in sets.

Use database deadlock priority - You can set the priority in case of transaction deadlocks in the Microsoft SQL Server database. Available values are Low, Normal and High. The default value is Normal. Setting the value to High and Low influences how Microsoft SQL Server resolves the potential deadlocks. Setting the priority to **High** means that the BigFix Administration Tool transaction is more likely to survive in case of deadlocks. Setting the priority to **Normal** means that Microsoft SQL Server will decide which transactions will survive based on rollback costs. Setting the priority to **Low** means that the BigFix Administration Tool transaction is more likely to be terminated in case of deadlocks.

To automate the process of removing all this data from the the database you can schedule the process by clicking **Schedule** in the `BES Audit trail Cleaner` of the BigFix Administration Tool.

The following window is displayed:



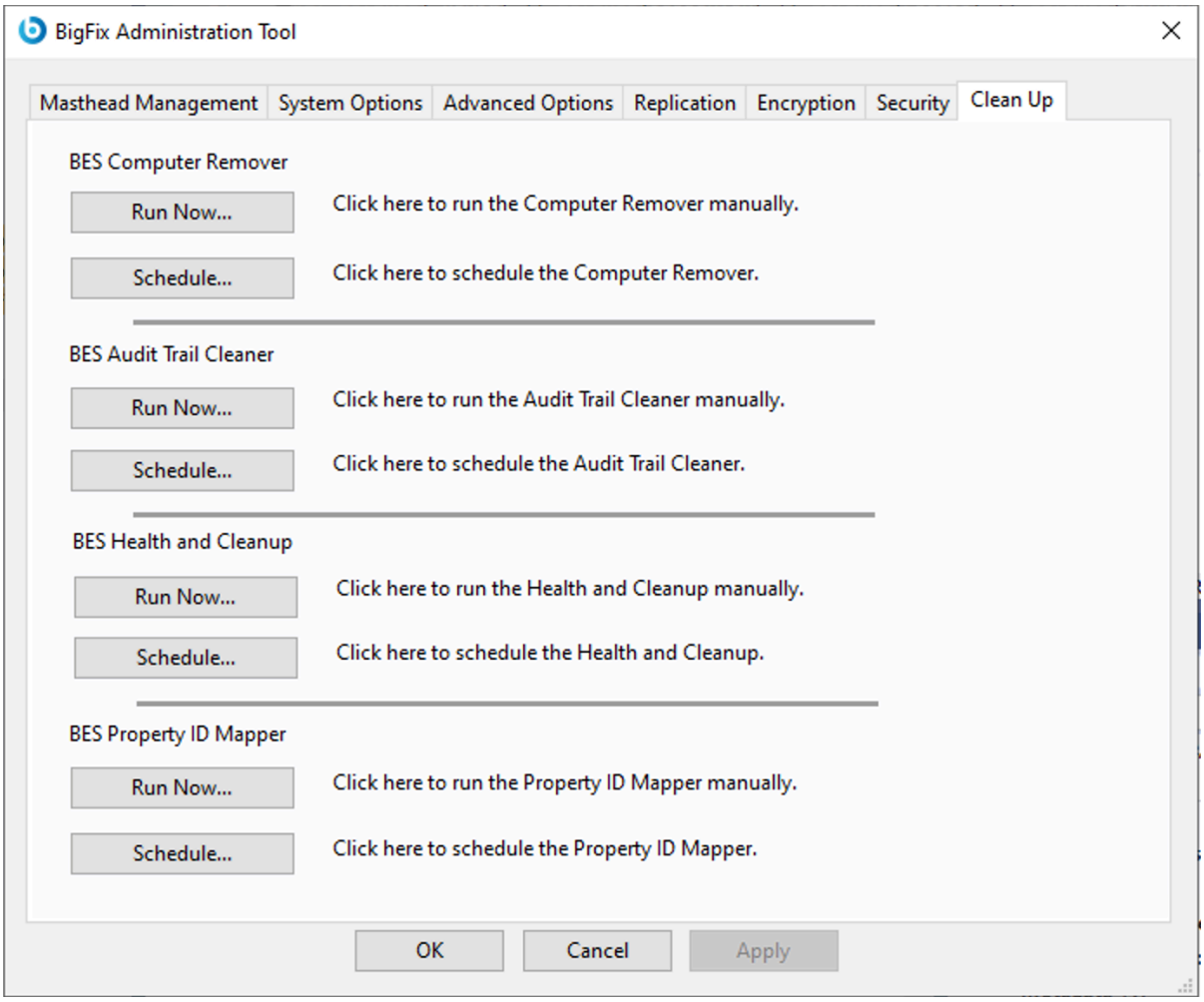
You can specify the date to start the data deletion and also a period of time to make this deletion operation recursive.

In **Specify the timeout**, you can also specify a timeout value in minutes by which the scheduled process must complete.

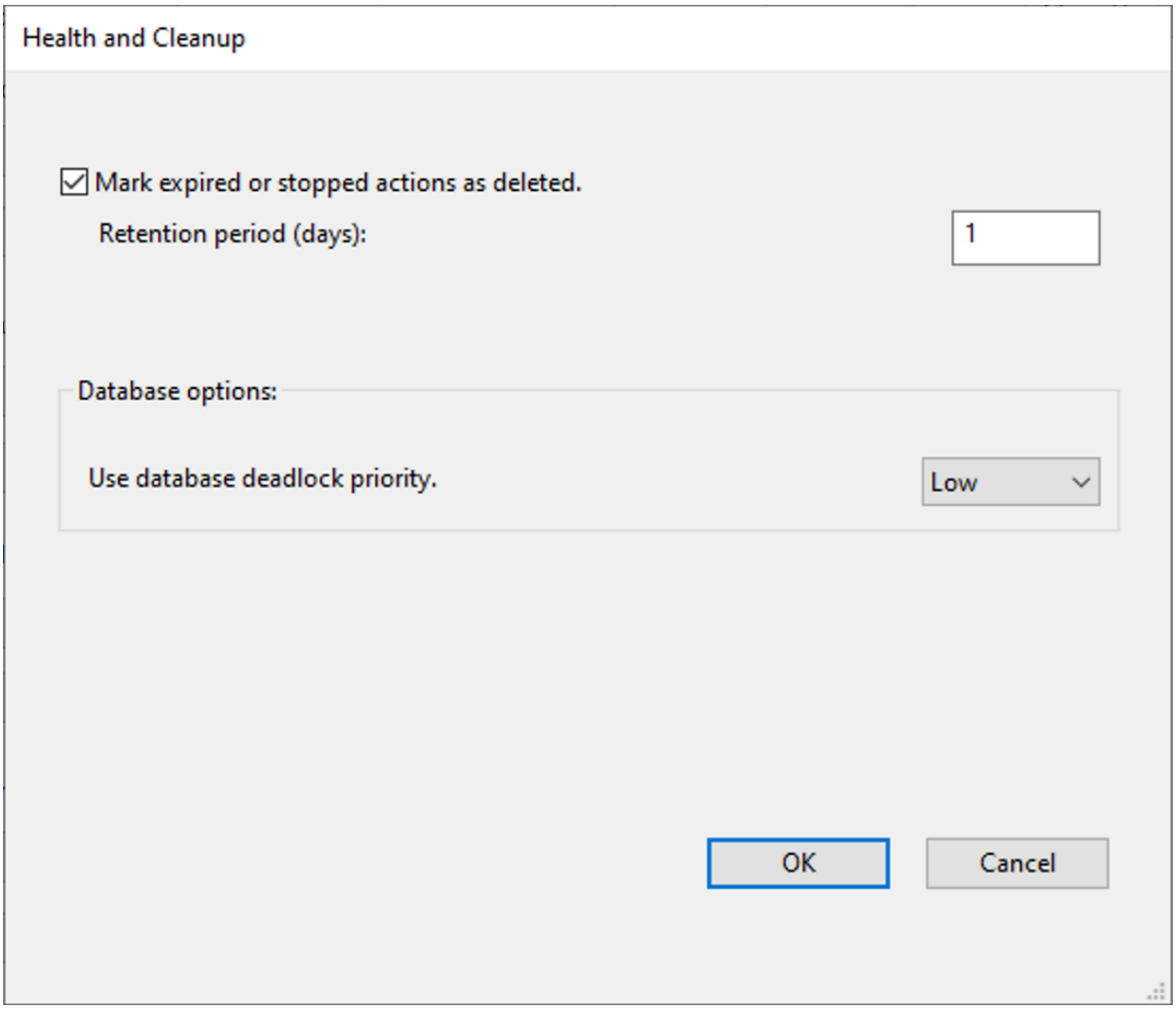
In **Use database deadlock priority**, you can set the priority in case of transaction deadlocks in the Microsoft SQL Server database. Available values are Low, Normal and High. The default value is Normal. Setting the value to High and Low influences how Microsoft SQL Server resolves the potential deadlocks. Setting the priority to **High** means that the BigFix Administration Tool transaction is more likely to survive in case of deadlocks. Setting the priority to **Normal** means that Microsoft SQL Server will decide which transactions will survive based on rollback costs. Setting the priority to **Low** means that the BigFix Administration Tool transaction is more likely to be terminated in case of deadlocks.

Health and Cleanup

Click **Run Now...** in the `BES Health and Cleanup` section to mark as deleted the expired or stopped actions if a specific number of days, which is configurable, has passed since the actions were stopped or expired. This feature is available starting from BigFix Version 11.0.6 and later.



The following window is displayed:



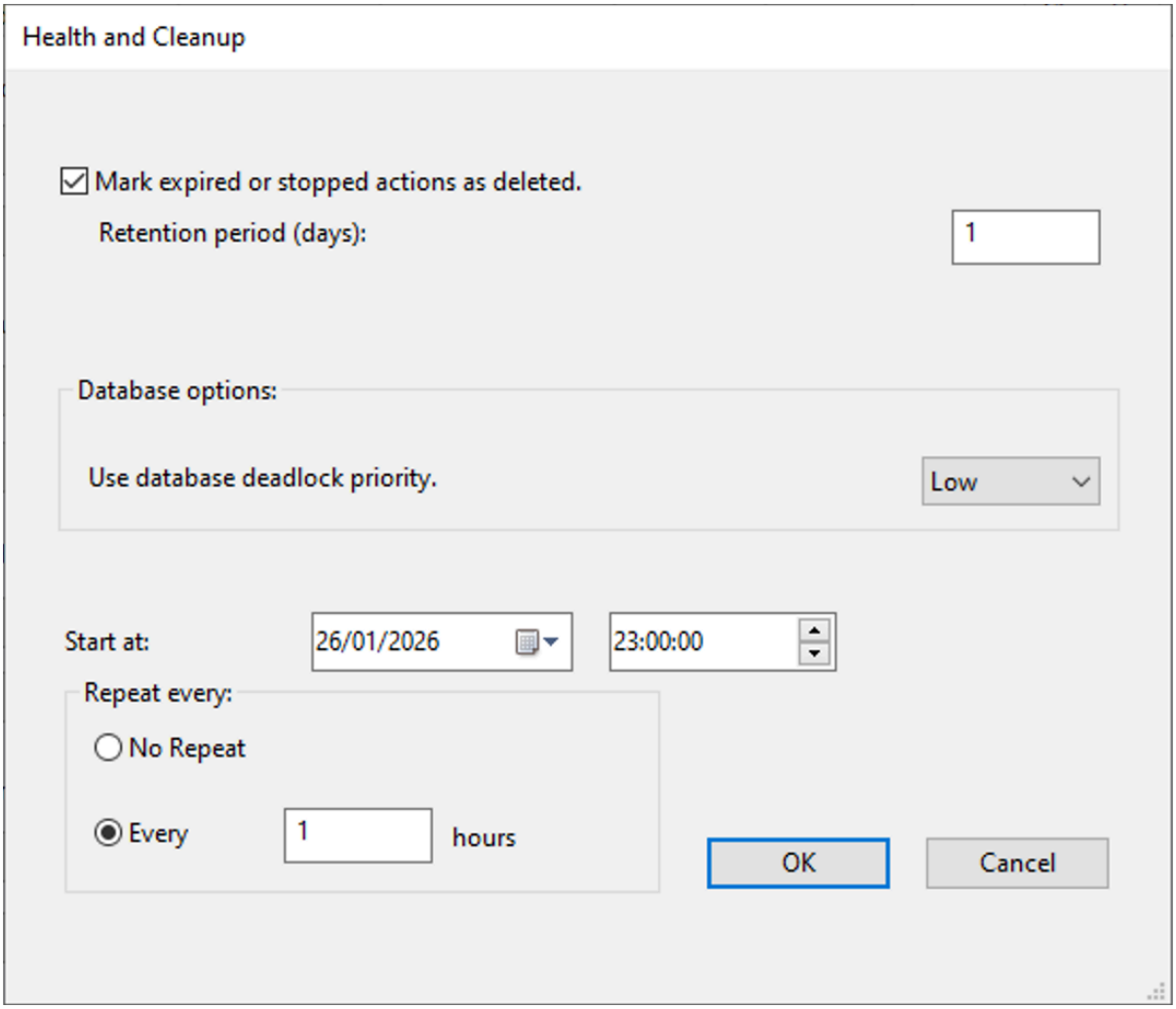
You can specify to mark the following data:

- Expired or stopped actions by selecting **Mark expired or stopped actions as deleted**. The expired or stopped actions are marked as deleted after the specified amount of time.
- After selecting the **Mark expired or stopped actions as deleted** check box, you can specify, in the **Retention period (days)** text edit, the amount of days after which the expired or stopped actions will no longer be visible in the BigFix Console.
- In **Use database deadlock priority**, in the **Database options** box, you can set the priority in case of transaction deadlocks in the Microsoft SQL Server database. Available values are Low, Normal and High. The default value is Normal. Setting

the value to High and Low influences how Microsoft SQL Server resolves the potential deadlocks. Setting the priority to **High** means that the BigFix Administration Tool transaction is more likely to survive in case of deadlocks. Setting the priority to **Normal** means that Microsoft SQL Server will decide which transactions will survive based on rollback costs. Setting the priority to **Low** means that the BigFix Administration Tool transaction is more likely to be terminated in case of deadlocks.

To automate the process of marking as deleted the expired or stopped actions, which are displayed in the BigFix Console, and updating the data in the database, you can schedule the process by clicking **Schedule...** in the `BES Health and Cleanup` section of the BigFix Administration Tool GUI.

The following window is displayed:



In addition to the fields present in the **Run Now...** dialog and already described above, the **Schedule...** dialog contains the following fields:

1. In **Start at**, you can define the date and time when the **Health and Cleanup** schedule task will be executed for the first time. The task will not run before the specified date and time. The selected time is based on the server local time.
2. The **Repeat every** section controls whether the **Health and Cleanup** schedule task should be executed only once or repeated periodically.
    - **No Repeat** When selected, the **Health and Cleanup** schedule task runs **only once** at the date and time specified in **Start at**.
    - **Every** When selected, the **Health and Cleanup** schedule task is executed repeatedly at a fixed interval. The time interval is defined in hours. The first execution occurs at the **Start at** date and time. Subsequent executions occur every N hours after the initial run. This option is suitable for regular cleanup activities that need to run continuously.

Using this window, you can:

- Create a new schedule task, which is saved into the BigFix Database. BigFix allows to save one **Health and Cleanup** schedule task only.
- Modify the saved **Health and Cleanup** schedule task as needed.

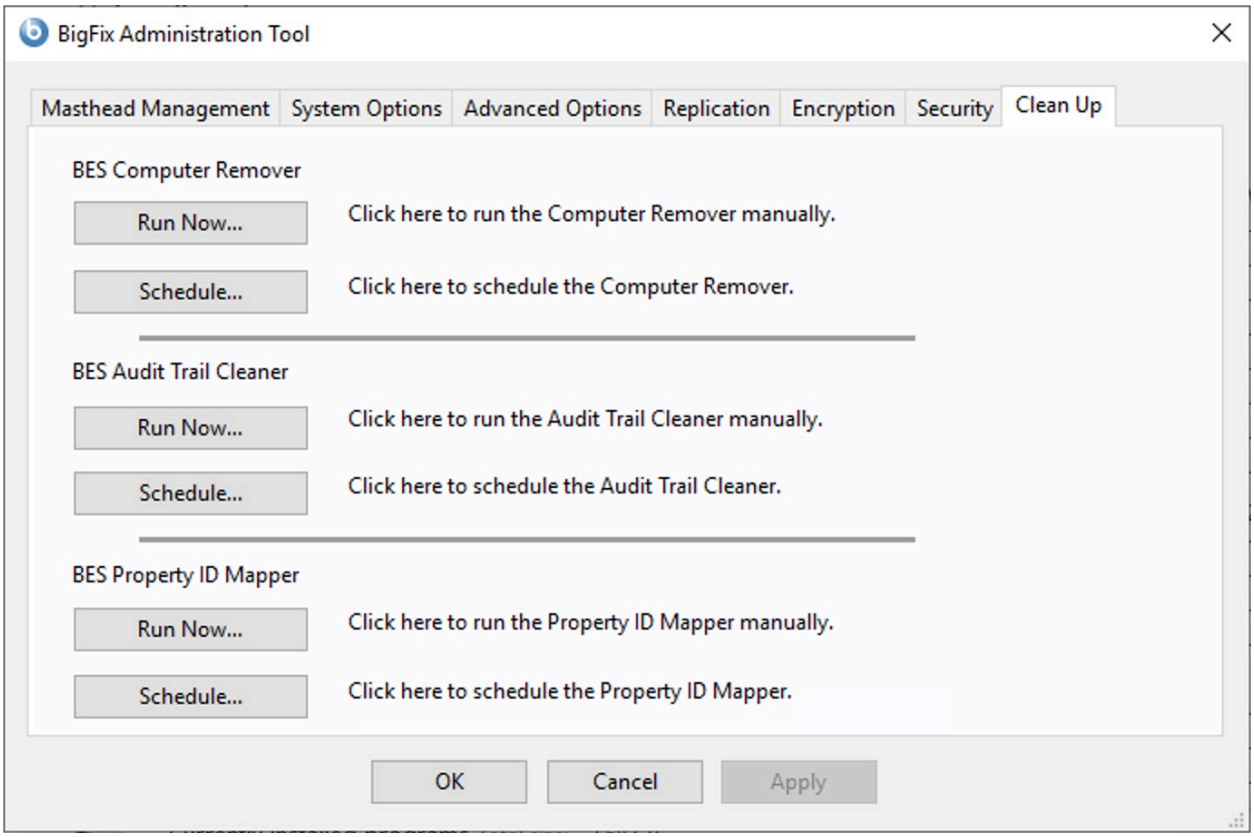
Property ID Mapper

After updating a property you must run the Property ID Mapper tool to update the `PropertyIDMap` table of the BFEnterprise database with the corresponding changes.

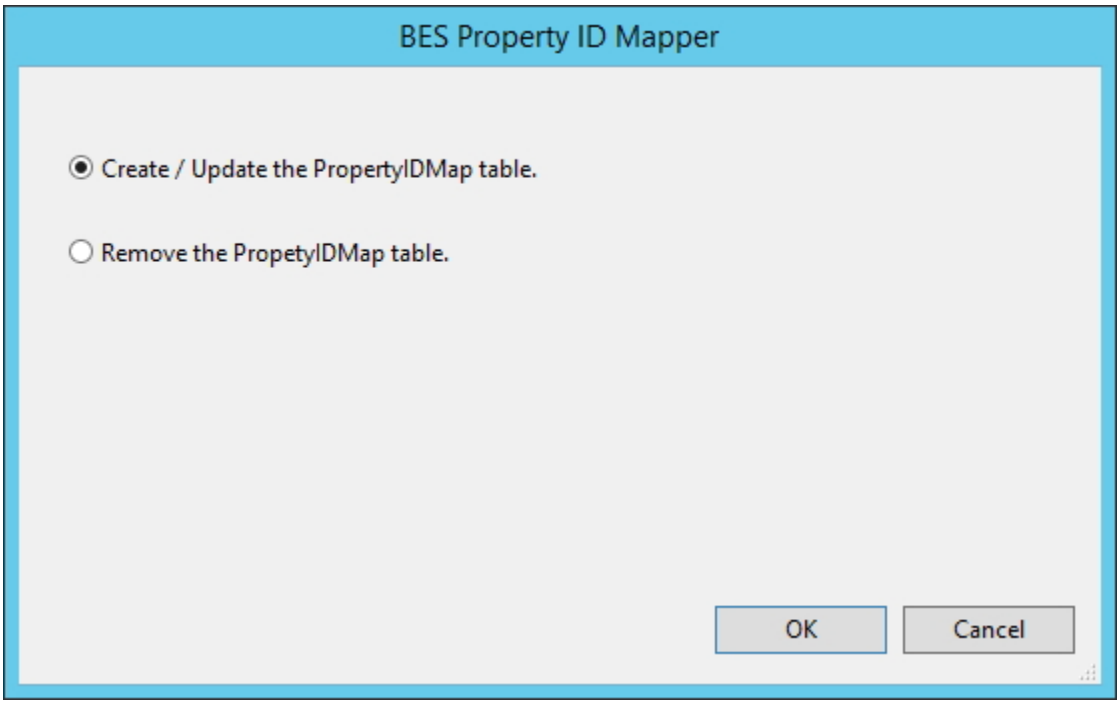
This table maps retrieved property names to the SiteID, AnalysisID, PropertyID used to reference properties in the QUESTIONRESULTS and LONGQUESTIONRESULTS tables.

The tool creates the PropertyIDMap table if it does not exist.

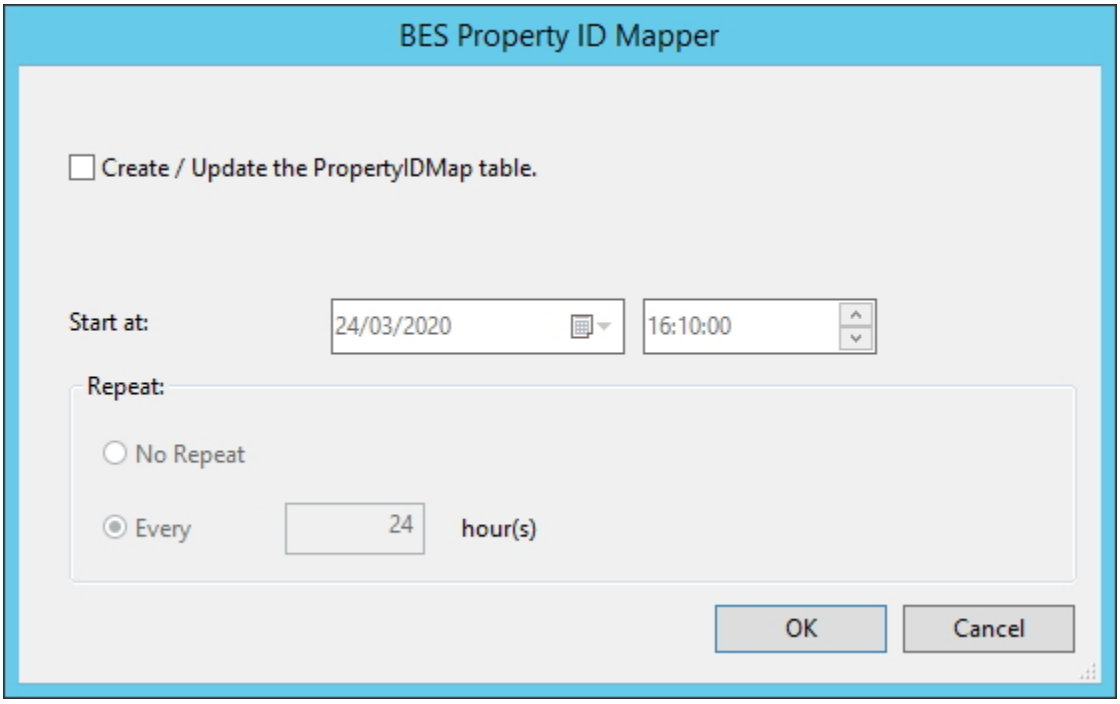
After creating or deleting a property, click **Run Now** in the `BES Property ID Mapper` section:



The following window is displayed to create, update, or remove the PropertyIDMap table:



To automate the process of creating or updating the PropertyIDMap table, you can schedule the process by clicking **Schedule** in the `BES Property ID Mapper` section of the BigFix Administration Tool.



BESAdmin Windows Command Line

The installation automatically downloads the BigFix Administration Tool program `BESAdmin.exe`, in the `%PROGRAM FILES%\BigFix Enterprise\BES Server` directory.

To prevent conflicts, avoid running BESAdmin commands concurrently. Only one BESAdmin instance should run at a time within a BigFix environment.

Starting with version 11.0.7, a BESAdmin command will exit without taking action if the BESAdmin GUI is already open or another BESAdmin command is running on the same computer. No error message will be logged. If the command is run silently, it will not display any error message.

You can run the script `BESAdmin.exe` to perform additional operations. To run this script from the command prompt, use the following command:

```
.\BESAdmin.exe /service { arguments}
```

where *service* can be one of the following:

```
audittrailcleaner
checkcontrolflowguard
checkdbindexfrag
checkdbinfo
checksqlserverparallelism
converttoidpoperators
converttoldapoperators
createexplorercredentials
createrepositorysiteregistrar
createtokenkey
createwebuicredentials
findinvalidactions
findinvalidsignatures
getcertificatebundle
healthcleanup
minimumSupportedClient
minimumSupportedRelay
propagateAllOperatorSites
propertyidmapper
removecomputers
reportencryption
resetDatabaseEpoch
resignsecuritydata
revokeexplorercredentials
revoketokenkey
revokewebuicredentials
rotateexplorercredentials
rotaterepositorysiteregistrar
rotateserversigningkey
```

```
rotatetokenkey
rotatewebuicredentials
setcontrolflowguard
securitysettings
setClientCertificatePolicy
setcustomca
setproxy
setshareableserverdatapath
setsqlserverparallelism
syncmastheadandlicense
updatepassword
```

Note: The notation `<path+license.pvk>` used in the command syntax that is displayed across this topic stands for *`path_to_license_file/`*`license.pvk`.

Each service has the following *`arguments`* :

audittrailcleaner

You can run this service to remove historical data from the bfenterprise database that is stored to serve as an audit trail. This audit trail slowly increases in size over the lifetime of a BigFix deployment. The audit trail contains deleted and earlier versions of Fixlets, tasks, baselines, properties, mailbox files, actions, and analyses. The audit trail is not used by BigFix in any way and can be deleted to reduce the database size. BigFix recommends that you create a historic archive of the current database and save it to a secure location before you run this tool to preserve the audit trail, thus removing it from the product database, but not completely deleting the history.

The service can count and delete the following sets of data:

- **Older Versions of Custom Authored Content** (`/oldcontent`): Every edit to Fixlets, Tasks, Baselines, and Analyses creates a new version, the earlier versions can be deleted.
- **Older Versions of Actions** (`/oldactions`): Any time that you stop or start an Action, a new version is created; the earlier versions can be deleted.
- **Older Versions of relay.dat** (`/oldrelaydatfile`): Any time that you install or uninstall a new relay, a new version is created; the earlier versions can be deleted.
- **Older uploaded files** (`/deleteolduploadedfiles`): Removes the old files uploaded by the Archive Manager on the BigFix server. This option deletes the old files after an expiration period (default 180 days) from when they were uploaded.
- **Deleted Custom Authored Content (all versions)** (`/deletedcontent`): When you delete a Fixlet, Task, Baseline, and Analysis using the console, the data is marked as deleted in the database and preserved. The deleted content, including all of the earlier versions, and the corresponding client reports can be deleted.
- **Deleted Actions(all versions)** (`/deletedactions`): When you delete an action using the console, the data is marked as deleted in the database and preserved. The deleted actions, including all of the earlier versions, and the corresponding client reports can be deleted.
- **Useless Action Results** (`/uselessactionresults`): Earlier versions of BigFix might cause clients to report ActionResults that were not used in any way but would use up space in the database. These useless ActionResults can be deleted.
- **Orphaned sub-actions** (`/orphanedsubactions`): From multiple action groups that were deleted.
- **Hidden Manual Computer Group Actions** (`/hiddenactions`): Manual Computer Groups create hidden actions that add and remove computers to and from groups and the actions can build up over time. This option deletes actions after an expiration period (default 180 days) from when they were created.

- **Older Version of Mailbox Files** (`/deletedmailbox`): Deleted Mailbox Files are stored in a table in the database and can be removed.
- **Old audit log files** (`/deleteauditlogs`): Removes old audit log files to prevent the server running out of disk space.
- **Synchronizing BES Consoles** (`/syncconsoles`): The BigFix Console maintains a local cache of the database that becomes not synchronized when data is removed with this tool. To avoid such inconsistencies, the tool sets a flag in the database to force all BigFix Consoles to reload the cache when the Console is started up.
- **Removing data older than** (`/olderthan`): Removes data earlier than a specified date.
- **Batched deletion** (`/batchsize`): Deleting large sets of data causes the SQL transaction log to quickly increase in size, the log temporarily becomes larger than the data being removed until the database is shrunk. Batched deletion removes results in sets.
- **Deadlock Priority** (`/databaseDeadlockPriority`): Sets the priority for the service in case of transaction deadlocks in the database. Available values are Low, Normal and High. The default value is Normal.

The syntax of this service changes depending on the action you specify:

```
.\BESAdmin.exe /audittrailcleaner { /displaysettings | /run [del
ete_data_options] |
          /schedule [delete_data_options] [scheduling options] |
 /preview [delete_data_options]
           [preview options] }
```

```
.\BESAdmin.exe /audittrailcleaner /displaysettings
```

```
.\BESAdmin.exe /audittrailcleaner /run [ /oldcontent ] [ /oldact
ions ]
          [ /oldrelaydatfile ] [ /deleteolduploadedfiles ] [ /de
letedcontent ] [ /deletedactions ] [ /hideUI ]
```

```
          [ /uselessactionresults ] [ /orphanedsubactions ] [ /h
iddenactions=<days> ]
          [ /deletedmailbox ] [ /deleteauditlogs ] [ /syncconsol
es ] [ /olderthan=<days> ] [ /batchsize=<size> ] [ /databaseDead
lockPriority=<Low|Normal|High> ]
```

```
.\BESAdmin.exe /audittrailcleaner /sitePvkLocation=<path+license
.pvk>
      [ /sitePvkPassword=<password> ] /schedule
 [ [ /oldcontent ] [ /oldactions ]
      [ /oldrelaydatfile ] [ /deleteolduploadedfiles ] [ /delete
dcontent ] [ /deletedactions ] [ /uselessactionresults ]
      [ /orphanedsubactions ] [ /hiddenactions=<days> ] [ /delet
edmailbox ] [ /deleteauditlogs ] [ /syncconsoles ]
      [ /olderthan=<days> ] [ /batchsize=<size> ] [ /databaseDea
dlockPriority=<Low|Normal|High> ] [ /cleanstarttime=<yyyymmdd:hh
mm>
      [ /cleanperiodicinterval=<hours> ] ] | /disable ]
```

```
.\BESAdmin.exe /audittrailcleaner /preview [ [ /oldcontent ] [ /
oldactions ] [
      /oldrelaydatfile ] [ /deleteolduploadedfiles ] [ /deletedc
ontent ] [ /deletedactions ] [ /uselessactionresults ] [
      /orphanedsubactions ] [ /hiddenactions=<days> ] [ /deleted
mailbox ] [ /deleteauditlogs ] [ /olderthan=<days> ]
      [ /databaseDeadlockPriority=<Low|Normal|High> ] | [ /sched
uled ] ]
```

where:

- **`displaysettings`** shows the settings that are previously set with the `schedule` action.
- **`run`** runs the tool with the specified settings. Before you use this option, check the settings that affect the database by using the **`preview`** action. Use option `/hideUI` to avoid pop-up windows notifying action results.
- **`schedule`** schedules the tool to run at the specified time at each specified interval. To disable the schedule action, use the `disable` option.
- **`preview`** shows the number of database rows affected by the specified settings. If no setting is passed to the preview option, the preview performs the count by setting all options to true and using the default values for dates. Use the `scheduled` option to preview the scheduled settings.

For more information about the cleanup tasks log files, see Logging Cleanup Tasks Activities (on page 467).

checkcontrolflowguard

You can use this service to retrieve the Microsoft Control Flow Guard (CFG) enablement of the program specified by the `/binaryName` parameter.
The syntax to run this service is:

```
.\BESAdmin.exe /checkcontrolflowguard /binaryName=<programName.exe> [/hideUI]
```

Use the `/hideUI` option to avoid pop-up windows notifying action results.

For more details about the Microsoft Control Flow Guard (CFG), see Enabling Microsoft Control Flow Guard on BigFix Server.

checkdbindexfrag

You can use this service to to list and filter the indexes fragmentation of the BigFix database on your SQL Server. In the BigFix Administration Tool panel the top five indexes matching the filter are displayed, the full list can be found in the BESAdminDebugOut.txt log file.

The syntax to run this service is:

```
.\BESAdmin.exe /checkdbindexfrag [/fragPercent=<integer>] [/page
Count=<integer>]
```

You can specify the following optional parameters to filter the result:

- `/fragPercent=<integer>` specifies the the minimum percentage fragmentation value to search for. Default is 10.
- `/pageCount=<integer>` specifies the minimum page count value to search for. Default is 1000.

checkdbinfo

You can use this service to list the main configuration settings of the BigFix database on your SQL Server, such as DB version, server memory, compatibility level, and so on.
The syntax to run this service is:

```
.\BESAdmin.exe /checkdbinfo
```

checksqlserverparallelism

You can use this service to check for common SQL Server configuration issues in your database instance with regard to the effective use of multiple CPU cores. Some of these issues can be fixed by simply changing configuration parameters, others may require more advanced configuration changes to be fully solved. When executed without additional options, this service only checks if the MaxDoP and CTFP settings are set to the recommended values on the given environment.
The syntax to run this service is:

```
.\BESAdmin.exe /checksqlserverparallelism [/extraChecks] [/extra
Info]
[/ctfpTolerance=<0.0 .. 1.0>]
```

You can specify the following optional parameters:

- `/extraChecks` performs additional checks, to detect additional issues, such as "under-utilization of licensed cores" and "uneven distribution of used cores".
- `/extraInfo` is an optional flag to show more information, such as the number of used cores per NUMA node and the number of hardware NUMA nodes.
- `/ctfpTolerance=<0.0 .. 1.0>` specifies a tolerance margin for the CTFP setting; if not specified, it defaults to 0.1, i.e. a CTFP within 10% of the generally recommended value is considered acceptable.

To run this command, you must have these permissions on the database:

- The `view server state` permission is required.
- In addition, when using the `/extraInfo` option, it is also required that 'master' is mapped among the databases that can be managed (User Mapping) and has the `execute permission` to run the `sys.xp_readerrorlog` procedure.

converttoidpoperators

You can move BigFix operators from one Identity Provider to another already configured. You must provide a mapping file, where each line has the name of the user to convert from, followed by a tab, followed by the name of the new user to convert to. Specify the name using the same format that the user will use to log into the BigFix Console, *domain\user*, *user@domain*, or *user*.
The syntax to run this service is:

```
.\BESAdmin.exe /convertToIdpOperators /mappingFile:<file>
```



Note: If two Identity Providers serve the same domain, the tool will move the users from the current Identity Provider to the next one able to authenticate them: since there is no Identity Provider priority at the moment, no order can be guaranteed if there are more than two Identity Providers authenticating the same users.

converttoldapoperators

You can convert local operators to LDAP operators, so that they can log in with their LDAP credentials. Optionally you can use the `mappingFile` argument to specify a file, the mapping file, where each line has the name of the user to convert, followed by a tab, followed by the name of the user in LDAP/AD. Specify the name using the same format that the user will use to log into the BigFix Console, *domain\user*, *user@domain*, or *user*. If you do not specify a mapping file, all users are converted assuming their name in LDAP/AD is the same as their local user name.
The syntax to run this service is:

```
.\BESAdmin.exe /convertToLDAPOperators [/mappingFile:<file>]
```

createexplorercredentials

Use this service to create a BigFix Explorer certificate for the input hostname, in a similar way as the one for the WebUI certificate. The following files will be created in the output folder:

- The certificate file of the ExplorerCertificate "authentication": **auth_cert.crt**
- The key file of the ExplorerCertificate "authentication": **auth_key.key**
- The certificate file of the ExplorerCACertificate "authentication": **ca.crt**
- The certificate file of the WebUICACertificate "authentication": **apica.crt**.

Use the following syntax to run the command:

```
.\BESAdmin.exe /createexplorercredentials
/sitePvkLocation:<path+license.pvk>
/sitePvkPassword:<password>
/explorerCertDir:<path>
/explorerHostname:<BigFixExplorerHostnameOrIP>
[ /f ]
```

This service generates a folder named cert_*explorerHostname* in the path specified by the **explorerCertDir** option.

explorerCertDir

Specifies the path to the parent folder of the new folder containing the certificates. This folder must exist.

explorerHostname

Specifies the hostname or IP address of the computer that will host your BigFix Explorer.

f

Forces new certificate creation, without checking if a certificate for the same hostname exists (if it exists, it will overwrite it).

createrepositorysiteregistrar

Use this service to create and store the permanent External Repository Site Registrar. It uses the deployment credentials supplied to BESAdmin and must be run before importing an External Repository Site when a registrar does not already exist. The installer creates it automatically for a new Root Server installation. Use the `rotaterepositorysiteregistrar` command only when a replacement is required. This command relates to a feature introduced in 11.0.7 named External Repository Sites.

```
.\BESAdmin.exe /createrepositorysiteregistrar /sitePvkLocation=<
path+license.pvk>
/sitePvkPassword=<password>
```

createtokenkey

Use this service to create a secret key to issue authentication tokens, which is necessary to use the bearer token authentication feature introduced in 11.0.6. For more information, see Configuring bearer token authentication.

```
.\BESAdmin.exe /createtokenkey /sitePvkLocation=<path+license.pv
k>
/sitePvkPassword=<password>
```

createwebuicredentials

Use this service to generate the certificates used as WebUI credentials. Use the following syntax to run the command:

```
.\BESAdmin.exe /createwebuicredentials
/sitePvkLocation:<path+license.pvk>
/sitePvkPassword:<pwd> /webUICertDir:<path>
/webUIHostname:<WebUIHostnameOrIP>
[ /f ]
```

This service generates a folder named cert_*WebUIHostnameOrIP* in the path specified by the **webUICertDir** option.

webUICertDir

Specifies the path to the parent folder of the new folder containing the certificates. This folder must exist.

webUIHostname

Specifies the hostname or IP address of the computer that will host your WebUI.

f

Does not check if a WebUI certificate already exists before creating a new one. If it exists for the same hostname, it will overwrite it.

Note: If you need to generate WebUI credentials certificates, but you have no WebUI in your deployment, then set:

webUICertDir



To the BigFix server folder. For example, `BigFix Enterprise\BES Server`.

webUIHostname

To the BigFix server IP address or hostname.

findinvalidactions

You can check for invalid actions in the database by specifying the following parameter:

- (Optional) -deleteInvalidActions: Deletes invalid actions.

The syntax to run this service is:

```
.\BESAdmin.exe /findinvalidactions [ /deleteInvalidActions ]
/sitePvkLocation=<path+license.pvk> [ /sitePvkPassword=<pwd> ]
```

findinvalidsignatures

You can check the signatures of the objects in the database by specifying the following parameters:

-resignInvalidSignatures (optional)

Attempts to resign any invalid signatures that `BESAdmin` finds.

-deleteInvalidlySignedContent (optional)

Deletes contents with invalid signatures.

For more information about invalid signatures, see Resolving invalidly signed content problems in the console.

The syntax to run this service is:

```
.\BESAdmin.exe /findinvalidsignatures
[ /resignInvalidSignatures | /deleteInvalidlySignedContent ]
```

getcertificatebundle

You can export the certificate bundle (PEM) used by the current version of BES Admin. In the bundle, there are all the certificates of all the authorized chains in the masthead. So, for example, with SHA384 forced, in the bundle there is only the SHA384 chain. The generated file, named bundle.pem, is located in the folder specified by the bundleCertDir=<path> option. If you specify the **getCustomCA** option, the command exports any custom CA certificates previously set by the user.
The syntax to run this service is:

```
.\BESAdmin.exe /getcertificatebundle /sitePvkLocation=<path+lice
nse.pvk>
[ /sitePvkPassword=<pwd> ] /bundleCertDir=<path>
 [ /getCustomCA ]
```

healthcleanup

You can mark as deleted the expired or stopped actions if a specific number of days, which is configurable, has passed since the actions were stopped or expired. This feature is available starting from BigFix Version 11.0.6 and later.
The general syntax to run this service is:

```
.\BESAdmin.exe /healthcleanup { /displaySettings  | /run [run_he
althcleanup_options]
       | /schedule [schedule_healthcleanup_options]
       | /preview [preview_healthcleanup_options] }
```

Depending on the action that is specified, the syntax changes as follows:

```
.\BESAdmin.exe /healthcleanup /displaySettings
```

```
.\BESAdmin.exe /healthcleanup /run  [ /deleteExpiredStoppedActio
ns=<days> ]
       [ /databaseDeadlockPriority=<Low|Normal|High> ]
```

```
.\BESAdmin.exe /healthcleanup /schedule  [ [ /deleteExpiredStopp
edActions=<days> ]
```

```
[ /databaseDeadlockPriority=<Low|Normal|High> ]
[ /startTime=<YYYYMMDD:HHMM> [ /periodicInterval=<Hours> ] ] | /
disable | /delete ]
```

```
.\BESAdmin.exe /healthcleanup /preview [ [ /deleteExpiredStopped
Actions=<days> ]
[ /databaseDeadlockPriority=<Low|Normal|High> ] | /scheduled ]
```

where:

- **displaySettings** shows the settings that are previously set with the schedule action.
- **run** runs the tool with the specified settings.
  - **deleteExpiredStoppedActions=<days>** Defines the retention period, in days, after which expired or stopped actions will be deleted. The number represents how many days have passed since the action was stopped or expired.
  - **databaseDeadlockPriority=<Low|Normal|High>** Sets the priority during the cleanup operation in case of transaction deadlocks in the database. Available values are Low, Normal and High. The default value is Normal. This setting is applicable only for Microsoft SQL Server databases while it must not be set on DB2.
- **schedule** Creates or updates a scheduled Health and Cleanup task using the specified options. The task will run automatically starting from the given start time and with the selected cleanup settings.
  - **deleteExpiredStoppedActions=<days>** Defines the retention period, in days, after which expired or stopped actions will be deleted. The number represents how many days have passed since the action was stopped or expired.
  - **databaseDeadlockPriority=<Low|Normal|High>** Sets the priority during the cleanup operation in case of transaction deadlocks in the database. Available values are Low, Normal and High.

The default value is Normal. This setting is applicable only for Microsoft SQL Server databases while it must not be set on DB2.

- **`startTime=<YYYYMMDD:HHMM>`** Specifies the date and time when the scheduled task will start. For example, `startTime=20260203:1430`
- **`periodicInterval=<Hours>`** Specifies the interval, in hours, at which the scheduled task will run after the start time. If set to 0, the task will run only once. If set to a value greater than 0, the task will run repeatedly at the specified hourly interval. For example `periodicInterval=6`
- **`disable`** Disables the existing scheduled Health and Cleanup task, if present.
- **`delete`** Deletes the existing scheduled Health and Cleanup task, if present.

- **`preview`** Displays a preview of the Health and Cleanup operation without performing any changes to the database. The preview shows how many actions would be affected using the specified options or the currently configured values.
  - **preview (without additional options)** Generates a preview using the default Health and Cleanup settings.
  - **preview (with options)** When combined with options such as `deleteExpiredStoppedActions=<days>` or `databaseDeadlockPriority=<Low|Normal|High>`, the preview is generated using the provided values.
  - **preview -scheduled** Generates a preview using the values saved in the currently scheduled Health and Cleanup task. If no scheduled Health and Cleanup task exists or if the task is not active, an error message is displayed.

minimumSupportedClient

This service defines the minimum version of the BigFix Agents that are used in your BigFix environment.

Note: Based on this setting, the BigFix components can decide when it is safe to assume the existence of newer functions across all the component in the deployment. Individual agent interactions might be rejected if the interaction does not comply with the limitations that are imposed by this setting.

The currently allowed values are:

- **0.0** that means that no activity issued by BigFix Agents with versions earlier than V9.0, such as archive files and reports uploads, are prevented from running or limited. This behavior applies also if the `minimumSupportedClient` service is not set.
- **9.0** that means that:
  - Unsigned reports, such as the reports sent by BigFix Clients earlier than V9.0, are discarded by FillDB.
  - The upload of an unsigned archive file that is generated on a BigFix Client earlier than V9.0, by an **archive now** command for example, fails.

If you ran a fresh installation of BigFix V9.5.6 or later using a BES Authorization file, by default all the BigFix Clients earlier than V9.0 are prevented from joining your environment because the `minimumSupportedClient` service is automatically set to **9.0**.

The value assigned to this service, if set, remains unchanged:

- If you upgraded to V9.5.6 or later.
- If you installed BigFix V9.5.6 or later using an existing masthead.

In both cases, if the service did not exist before, it will not exist afterward as well.

The current value `<VALUE>` assigned in your environment to the `minimumSupportedClient` service is displayed in the line `x-bes-minimum-supported-client-level: <VALUE>` of the masthead file. You can see the

current value by running the following query on the BigFix Server, using the Fixlet Debugger or the BigFix Query Application available on the BigFix WebUI:

```
Q: following text of last ": " of line whose (it starts with
 "x-bes-minimum-supported-client-level:" ) of masthead of site "
actionsite"
```

The syntax to run this service is:

```
.\BESAdmin.exe [/sitePvkFile=<path+license.pvk>] [/sitePassword=
<password>]
/minimumSupportedClient=<version>.<release>
```

If you omit `[/sitePvkFile=<path+license.pvk>] [/sitePassword=<password>]`, you will be requested to enter the site key and password in a pop-up window.

For example, if you want to state that Agents earlier than V9.0 are not supported in your BigFix environment, you can run the following command:

```
.\BESAdmin.exe /minimumSupportedClient=9.0
```

minimumSupportedRelay

You can use this service, added with BigFix V9.5.6, to enforce specific criteria that affects the BigFix Agent registration requests. If this service is enabled, V9.5.6 Agents can continue to register to the V9.5.6 BigFix environment if their registration requests are signed and sent across the Relays hierarchy using the HTTPS protocol.



Note: Based on this service, the BigFix components can decide when it is safe to enable newer functions across all the component in the deployment. Individual agent interactions might be rejected if they do not comply with the limitations that are imposed by this setting.

The currently allowed values are:

- **0.0.0** that means that the BigFix Server accepts and manages:
  - Signed and unsigned registration requests coming from BigFix Agents.
  - Registration requests delivered from BigFix Agents that use the HTTP or the HTTPS protocols.

  This behavior applies by default when you upgrade from previous versions to BigFix V9.5.6 or later. In this case, the `minimumSupportedRelay` service is not added automatically to your configuration during the upgrade. Consider that this value is not displayed when you run the query to see the current value that is assigned in your environment to the `minimumSupportedRelay` service.
- **9.5.6** or later, which means that:
  - The BigFix Server enforces that registration requests coming from BigFix Agents V9.5.6 or later must be properly signed.
  - The BigFix Server and the Relays V9.5.6 or later enforce the use of the HTTPS protocol when exchanging BigFix Agent registration data.

  These are side effects of enforcing this behavior:
  - BigFix Agents earlier than V9.0 cannot send registration requests to the BigFix Server because they cannot communicate using the HTTPS protocol.
  - Because BigFix Relays with versions earlier than V9.5.6 cannot handle correctly signed registration requests, any BigFix Client that uses those Relays might be prevented from continuing to register, or might fall back to a different parent Relay or directly to the Server.

If you ran a fresh installation of BigFix V9.5.6 or later using a License Authorization file, be aware that the side effects that are listed apply to your BigFix deployment because, in this particular installation scenario, the `minimumSupportedRelay` service is automatically set to **9.5.6** by default.

The current value `<VALUE>` assigned in your environment to the `minimumSupportedRelay` service is displayed in the line `x-bes-minimum-supported-relay-level: <VALUE>` of the masthead file. You can see the current value by running the following query on the BigFix Server, using the Fixlet Debugger or the BigFix Query Application available on the BigFix WebUI:

```
Q: following text of last ": " of line whose (it starts with
"x-bes-minimum-supported-relay-level:" ) of masthead of site "ac
tionsite"
```

This query displays a value only when `<VALUE>` is set to **9.5.6**; if it is set to **0.0.0**, it does not display a value.
The syntax to run this service is:

```
.\BESAdmin.exe [/sitePvkFile=<path+license.pvk>] [/sitePvkPasswo
rd=<password>]
/minimumSupportedRelay=<version>.<release>.<modification>
```

If you omit `[/sitePvkFile=<path+license.pvk>] [/sitePwkPassword=<password>]`, you must to enter the site key and password in a pop-up window.

For example, if you want that only the registration requests that are signed and carried through HTTPS are managed by your BigFix Server, you can run the following command:

```
.\BESAdmin.exe /minimumSupportedRelay=9.5.6
```

propagateAllOperatorSites

This service forces the server to propagate a new version of every operator site. This command is useful after a server migration because you can be sure that data are available for clients to gather and it prevents from failures. This is the command syntax:

```
.\BESAdmin.exe /propagateAllOperatorSites
```

propertyidmapper

This service creates, updates, and deletes a table (PropertyIDMap) in the BFEnterprise database that maps retrieved property names for the SiteID, AnalysisID, PropertyID used to reference properties in the QUESTIONRESULTS and LONGQUESTIONRESULTS tables. It creates the PropertyIDMap table if it does not exist (requires table creation permissions). This service must be run after creating or deleting a property to update the PropertyIDMap table with changes.

The general syntax of this service is the following:

```
.\BESAdmin.exe /propertyidmapper  { /displaysettings | /run [pro
perty_idmapper_options]
       |  /schedule [property_idmapper_options] [scheduling opti
ons] }
```

The syntax of this service changes depending on the action you specify:

```
.\BESAdmin.exe /propertyidmapper /displaysettings
```

```
.\BESAdmin.exe /propertyidmapper /run [ /createtable ] [ /remove
table ]
      [ /lookupproperty=<propertyname> ] [ /hideUI ]
```

```
.\BESAdmin.exe /propertyidmapper /schedule [ /createtable /start
time=<yyyymmdd:hhmm>
     [ /interval=<hours> ] | /disable ]
```

where:

- **`displaysettings`** shows the settings that are previously set with the `schedule` action.
- **`run`** runs the tool with the specified settings. Use option `/hideUI` to avoid pop-up windows notifying action results.
- **`schedule`** schedules the tool to run at the specified time at each specified interval. To disable the schedule action, use the `disable` option.

For more information about the cleanup tasks log files, see Logging Cleanup Tasks Activities (on page 467).

removecomputers

The service runs database operations for the following sets of data:

- **Expired Computers** (`/deleteExpiredComputers`) Marks computers as deleted if they have not reported in recently.
- **Deleted Computers** (`/purgeDeletedComputers`): Physically deletes the computer related data from the database for computers that are already marked as deleted and have not reported in for a long time. It deletes the data related to an agent (such as the action results or the properties, and so on), not the agent itself that remains logically deleted (IsDeleted = 1) on the database. Therefore, as a consequence, if the same agent becomes active again, it is recognized and will reuse its previous computer ID.
- **Duplicate Computers** (`/deleteDuplicatedComputers`): Marks older computers as deleted if a computer exists with the same computer name.
- **Removal of deleted Computers** (`/removeDeletedComputers`): Physically deletes the computer information from the database for computers that are marked as deleted (IsDeleted = 1) since at least the indicated number of days (minimum 7) or the indicated number of hours (minimum 24). It deletes the information of the agent itself (such as the computer ID, and so on). Therefore, as a consequence, if the same agent becomes active again, a totally new computer ID will be assigned to the agent.
- **Removal of uploaded Files** (`/removeDeletedUploads`): Physically removes from the database the definition of uploaded files that are marked as deleted. It does not apply to non-native agents.
- **Removal of uploaded files of removed computers** (`/eraseUploadFilesForRemovedComputers`): Physically removes from the BigFix server file system all files uploaded by clients whose definition

has been removed from the database. It does not apply to non-native agents.

- **Removal of Computers by name** (`/removeComputersFile`): Accepts a text file with a list of computer names that are separated by new lines and removes them from the deployment.
- **Batched deletion** (`/batchsize`): Deleting large sets of data causes the SQL transaction log to quickly increase in size, the log temporarily becomes larger than the data being removed until the database is shrunk. Batched deletion removes results in sets.
- **Deadlock Priority** (`/databaseDeadlockPriority`): Sets the priority for the service in case of transaction deadlocks in the database. Available values are Low, Normal and High. The default value is Normal.

The general syntax of this service is:

```
.\BESAdmin.exe /removecomputers  { /displaySettings [display_set
tings_options] | /run [remove_computers_options]
       | /schedule [remove_computers_options] [scheduling option
s]
       | /preview [remove_computers_options] [preview options] }
```

Depending on the action, you specify, the syntax changes as follows:

```
.\BESAdmin.exe /removecomputers /displaySettings [ /name=<TaskNa
me> ]
```

```
.\BESAdmin.exe /removecomputers /run [ /deleteExpiredComputers=<
days> ]
    [ /removeDeletedComputers=<days> ] [ /removeDeletedUploads ]
    [ /eraseUploadFilesForRemovedComputers ]
    [ /purgeDeletedComputers=<days> ]
    [ /deleteDuplicatedComputers [ /duplicatedPropertyName=<Prop
ertyName> ] ]
```

```
    [ /removeComputersFile=<path> ] [ /batchSize=<batch size> ]
 [ /databaseDeadlockPriority=<Low|Normal|High> ] [ /hideUI ]
```

```
.\BESAdmin.exe /removecomputers /schedule [ [ /name=<TaskName> ]
 [ /agentType=<AgentType> ] [ /deleteExpiredComputers=<days> ] [
 /purgeDeletedComputers=<days> ]
[ /removeDeletedComputers=<days> ] [ /removeDeletedUploads ] [ /
eraseUploadFilesForRemovedComputers ]
[ /deleteDuplicatedComputers [ /duplicatedPropertyName=<Property
Name> ] ] [ /batchSize=<batch size> ] [ /databaseDeadlockPriorit
y=<Low|Normal|High> ]
[ /removeStartTime=<YYYYMMDD:HHMM> [ /removePeriodicInterval=<Ho
urs> ] ] | [ /disable -name=<TaskName> ] | [ /delete -name=<Task
Name> ] | [ /list ] |
[ /update [ /name=<TaskName> ]
 [ /deleteExpiredComputers=<days> ] [ /purgeDeletedComputers=<da
ys> ]
[ /removeDeletedComputers=<days> ] [ /removeDeletedUploads ] [ /
eraseUploadFilesForRemovedComputers ]
[ /deleteDuplicatedComputers [ /duplicatedPropertyName=<Property
Name> ] ] [ /batchSize=<batch size> ]
[ /databaseDeadlockPriority=<Low|Normal|High> ] [ /removeStartTi
me=<YYYYMMDD:HHMM> [ /removePeriodicInterval=<Hours> ] ] ] ]
```

```
.\BESAdmin.exe /removecomputers /preview [ [ /deleteExpiredCompu
ters=<days> ]
    [ /removeDeletedComputers=<days> ] [ /removeDeletedUploads ]
    [ /eraseUploadFilesForRemovedComputers ]
    [ /purgeDeletedComputers=<days> ][ /deleteDuplicatedComputer
s
    [ /duplicatedPropertyName=<PropertyName> ] ] [ /databaseDead
lockPriority=<Low|Normal|High> ] | [ /scheduled ] [ /name=<TaskN
ame> ] ]
```

```
```

where:

- **`displaySettings`** shows the settings that are previously set with the `schedule` action.
- **`run`** runs the tool with the specified settings. Before you use this option, check the settings that affect the database by using the `preview` action. Use option `/hideUI` to avoid pop-up windows that notify the action results.
- **`schedule`** schedules the tool to run at the specified time at each specified interval. To disable the schedule action, use the `disable` option.
- **`preview`** shows the number of database rows that are affected by the specified settings. If no setting is passed to the preview option, the preview performs the count by setting all options to true and using the default values for dates. Use the `scheduled` option to preview the scheduled settings.

Note: When using option `/removeDeletedComputers`, the number of days must be not less than 7 or the number of hours must be not less than 24.

For information about the cleanup tasks log files, see Logging Cleanup Tasks Activities (on page 467).

reportencryption

You can generate, rotate, enable, and disable encryption for report messaging by running:

```
.\BESAdmin.exe /reportencryption { /status |
  /generatekey [/privateKeySize=<min|max>]
               [/deploynow=yes | /deploynow=no /outkeypath=<path
>]
```

```
               /sitePvkLocation=<path+license.pvk> [/sitePvkPass
word=<password>] |
  /rotatekey [/privateKeySize=<min|max> ]
             [/deploynow=yes
 | /deploynow=no /outkeypath=<path> ]
             /sitePvkLocation=<path+license.pvk> [/sitePvkPasswo
rd=<password>] |
  /enablekey /sitePvkLocation=<path+license.pvk> [/sitePvkPasswo
rd=<password>] |
  /disable /sitePvkLocation=<path+license.pvk> [/sitePvkPassword
=<password>] }
```

where:

status

Shows the status of the encryption and which arguments you can use for that status.

generatekey

Allows you to generate a new encryption key.

rotatekey

Allows you to change the encryption key.

enablekey

Allows you to enable the encryption key.

disable

Allows you to put the encryption key in PENDING state. If you run again the `reportencryption` command with the `disable` argument, the encryption changes from PENDING state to DISABLED.

deploynow=yes

Deploys the report encryption key to the server for decryption.

deploynow=no -outkeypath=<path>

The encryption key is not deployed to the server but it is saved in the `outkeypath` path.

For more information about this command and its behavior, see Managing Client Encryption.

resetDatabaseEpoch

To clear all console cache information in BigFix Enterprise Service. After you run this command:

```
.\BESAdmin.exe /resetDatabaseEpoch
```

subsequent console logins reload their cache files.

resignsecuritydata

You must resign all of the users content in the database by entering the following command:

```
.\BESAdmin.exe /resignSecurityData
```

if you get one of the following errors:

```
class SignedDataVerificationFailure
HTTP Error 18: An unknown error occurred while transferring data
 from the server
```

when trying to login to the BigFix console. This command resigns security data by using the existing key file. You can also specify the following parameter:

```
/mastheadLocation=<path+/actionsite.afxm>
```

The complete syntax to run this service is:

```
.\BESAdmin.exe /resignsecuritydata /sitePvkLocation=<path+licens
e.pvk>
```

```
[ /sitePvkPassword=<password> ] /mastheadLocation=<path+/actions
ite.afxm>
```

revokeexplorercredentials

Use this service to revoke the BigFix Explorer certificate associated to the given hostname. Use the following syntax to run the command:

```
.\BESAdmin.exe /revokeexplorercredentials
/sitePvkLocation:<path+license.pvk>
/sitePvkPassword:<password>
/explorerHostname:<BigFixExplorerHostnameOrIP>
```

If an authentication certificate is issued for the specified `hostname`, this certificate is revoked and the BigFix Explorer instance running on that `explorerHostname` can no longer connect to the root server.

After revoking the credentials for a BigFix Explorer host, it will no longer connect to the root server. You can either remove the BES Explorer installation, or generate new credentials for that host, and replace the old certificate files on that host.

If your environment has multiple BigFix Servers, see this section (on page 418).

revoketokenkey

Use this service to revoke the secret key used to issue authentication tokens. After revoking it, all issued tokens will be invalid and operators will not be able to create new one.

```
.\BESAdmin.exe /revoketokenkey /sitePvkLocation=<path+license.pv
k>
/sitePvkPassword=<password>
```

revokewebuicredentials

You can revoke the authentication certificate of a specified WebUI instance. The syntax to run this service is:

```
.\BESAdmin.exe /revokewebuicredentials
/hostname=<host>
/sitePvkLocation=<path+license.pvk>
/sitePvkPassword=<pvk_password>
```

If an authentication certificate is issued for the specified `hostname`, this certificate is revoked and the WebUI instance running on that `hostname` can no longer connect to the root server.

After revoking the credentials for a WebUI host, it will no longer connect to the root server. You can either remove the WebUI installation, or generate new credentials for that host, and replace the old certificate files on that host.

If your environment has multiple BigFix Servers, see this section (on page 418).

rotateexplorercredentials

Use this service to rotate one BigFix Explorer certificate associated to the given hostname, or the whole BigFix Explorer CA. Use the following syntax to run the command:

```
.\BESAdmin.exe /rotateexplorercredentials
/sitePvkLocation:<path+license.pvk>
/sitePvkPassword:<password>
/explorerCertDir:<path>
/explorerHostname:<BigFixExplorerHostnameOrIP>
| /rotateCA
```

You can either rotate one BigFix Explorer certificate, by hostname, if the certificate associated to that given hostname was compromised. The files of the new certificate will be copied in the <explorerCertDir> path.
Or you can rotate the BigFix Explorer Certificate Authority and all BigFix Explorer certificates. The files of the new certificates will be copied in a new dedicated folder, in the <explorerCertDir> path, for each BigFix Explorer

instance that had a not yet revoked certificate before the command was run. The command

```
.\BESAdmin.exe /rotateexplorercredentials
/sitePvkLocation:<path+license.pvk>
/sitePvkPassword:<password>
/explorerCertDir:<path>
/rotateCA
```

will:

- Stop the Root Server
- Revoke each BigFix Explorer certificate created until that moment.
- Delete the ExplorerCACertificate and ExplorerCAKey files on the Root Server and in the Admin Fields database table.
- Start the Root Server that will recreate the files and entries of the previous point.
- Recreate all the certificates for the previous BigFix Explorer instances.

After the command was completed successfully, you must perform the following additional steps to allow Explorer to use the new credentials generated:

1. Stop the Explorer service.
2. Copy and replace the files apica.crt, auth_cert.crt, auth_key.key and ca.crt from the cert_hostname directory, created by the command in the explorerCertDir path, into the BES Explorer\cert directory. If the file root_ca.crt was also created, copy that file too.
3. Start the Explorer service.

If your environment has multiple BigFix Servers, see this section (on page 418).

rotaterepositorysiteregistrar

This command replaces the External Repository Site Registrar and records the corresponding revocation information. Run it with the standard BESAdmin deployment credentials when the registrar must be rotated in order to invalidate it. This command relates to a feature introduced in 11.0.7 named External Repository Sites.

```
.\BESAdmin.exe /rotaterepositorysiteregistrar /sitePvkLocation=<
path+license.pvk>
[ /sitePvkPassword=<password> ]
```

rotateserversigningkey

You can rotate the server private key to have the key in the file system match the key in the database. The command creates a new server signing key, resigns all existing content using the new key, and revokes the old key.
The syntax to run this service is:

```
.\BESAdmin.exe /rotateserversigningkey
/sitePvkLocation=<path+license.pvk>
[ /sitePvkPassword=<password> ]
```

If your environment has multiple BigFix Servers, see this section (on page 418).

rotatetokenkey

Use this service to rotate the secret key used to issue authentication tokens. The old key will be invalidated and replaced by a new one. The tokens created with the old key will no longer work.

```
.\BESAdmin.exe /rotatetokenkey /sitePvkLocation=<path+license.pv
k>
/sitePvkPassword=<password>
```

rotatewebuicredentials

Use this service to rotate one BigFix WebUI certificate associated to the given hostname, or the whole BigFix WebUI CA. Use the following syntax to run the command:

```
.\BESAdmin.exe /rotatewebuicredentials /sitePvkLocation=<path+license.pvk>
[ /sitePvkPassword=<password> ]  /webUICertDir=<path>
{ /webUIHostname=<WebUIHostnameOrIP> | /rotateCA }
```

You can either rotate one BigFix WebUI certificate, by hostname, if the certificate associated to that given hostname was compromised. The files of the new certificate will be copied in the `webUICertDir` path.
Or you can rotate the BigFix WebUI Certificate Authority and all BigFix WebUI certificates. The files of the new certificates will be copied in a new dedicated folder, in the `webUICertDir` path, for each BigFix WebUI instance that had a not yet revoked certificate before the command was run. The command

```
.\BESAdmin.exe /rotatewebuicredentials /sitePvkLocation=<path+license.pvk>
[ /sitePvkPassword=<password> ] /webUICertDir=<path> /rotateCA
```

will:

- Stop the Root Server
- Revoke each BigFix WebUI certificate created until that moment.
- Delete the WebUICACertificate, WebUICAKey, PlatCertificate, PlatKey, APIServerCertificate and APIServerKey files on the Root Server and in the Admin Fields database table.
- Start the Root Server that will recreate the files and entries of the previous point.
- Recreate all the certificates for the previous BigFix WebUI instances.

After the command was completed successfully, you must perform the following additional steps to allow the WebUI server to use the new credentials generated:

1. Stop the WebUI service.
2. Copy and replace the files auth_cert, auth_key and ca.crt from the cert_*hostname* directory, created by the command in the webUICertDir path, into the BES WebUI\cert directory. If the file root_ca.crt was also created, copy that file too.
3. Start the WebUI service.
4. If you have any active BigFix Explorer instance, stop it, copy the file ca.crt as apica.crt in the BES Explorer\cert directory. If the file root_ca.crt was also created, copy that file too. Restart the instance.

If your environment has multiple BigFix Servers, see this section (on page 418).

securitysettings { /status }

You can view the status of the security settings set in your BigFix environment.

The syntax to run this service is:

```
.\BESAdmin.exe /securitysettings /sitePvkLocation=<path+license.
pvk>
/sitePvkPassword=<password>
{ /status }
```

securitysettings { /requireSHA384Signatures [/requireSHA256Downloads] }

You can enable the security option that adopts the SHA-384 cryptographic digest algorithm for all digital signatures. You can ensure that data has not changed after you download it using the SHA-256 algorithm.

The syntax to run this service is:

```
.\BESAdmin.exe /securitysettings /sitePvkLocation=<path+license.
pvk>
/sitePvkPassword=<password>
{ /requireSHA384Signatures [/requireSHA256Downloads] }
```

securitysettings { /allowSHA256Signatures | /requireSHA256Downloads | /allowSHA1Downloads}

You can ensure that the file download integrity check is run using the SHA-256 algorithm. You can ensure that data has not changed after you download it using the SHA-256 algorithm. You can ensure that the file download integrity check is run using the SHA-1 algorithm.

The syntax to run this service is:

```
.\BESAdmin.exe /securitysettings /sitePvkLocation=<path+license.pvk>
/sitePvkPassword=<password>
{ /allowSHA256Signatures | /requireSHA256Downloads | /allowSHA1Downloads}
```

securitysettings { /hideFromFieldFromMasthead | /showFromFieldFromMasthead }

You can specify if you want to show or hide the value displayed by the From field in the masthead which contains the email address of the license assignee. During a fresh installation the value is hidden and the option "hideFromFieldFromMasthead" is set to 1. During an upgrade the value remains unchanged.

The syntax to run this service is:

```
.\BESAdmin.exe /securitysettings
{ /hideFromFieldFromMasthead | /showFromFieldFromMasthead }
[/sitePvkLocation=<path+license.pvk>] [/sitePvkPassword=<pvk_password>]
```



Note: You can modify the "hideFromFieldFromMasthead" option from the BESAdmin command line only. Doing it from the BESAdmin interface is not supported because the masthead will not be regenerated when modifying the settings from the advanced settings panel of the interface.

securitysettings { /testTLSCipherList | /setTLSCipherList | /listTLSCiphers | /removeTLSCipherList }

To test if a TLS cipher list is compatible with the BigFix components, run the following command:

```
.\BESAdmin.exe /securitysettings /sitePvkLocation=<path+license.pvk> /sitePvkPassword=<password> /testTLSCipherList=<cipher_1>:<cipher_2>:..:<cipher_n>
```

After identifying a suitable TLS cipher list, you can set it by running the following command:

```
.\BESAdmin.exe /securitysettings /sitePvkLocation=<path+license.pvk> /sitePvkPassword=<password> /setTLSCipherList=<cipher_1>:<cipher_2>:..:<cipher_n>
```

To list all the TLS ciphers that are currently enabled, run the following command:

```
.\BESAdmin.exe /securitysettings /sitePvkLocation=<path+license.pvk> /sitePvkPassword=<password> /listTLSCiphers
```

To remove a TLS cipher list from the deployment masthead and return to the default cipher list, run the following command:

```
.\BESAdmin.exe /securitysettings /sitePvkLocation=<path+license.pvk> /sitePvkPassword=<password> /removeTLSCipherList
```

securitysettings { /enableLocalOperators / disableLocalOperators }

You can specify if you want to enable or disable the login to the BigFix environment (BigFix Console, Web Reports, Rest API and Web UI) of the local operators. The enabled/disabled choice will be stored in the BFEnterprise

database. After disabling the login of the local operators, access will be granted only to LDAP users.
The syntax to run this service is:

```
.\BESAdmin.exe /securitysettings
{ /enableLocalOperators | /disableLocalOperators }
[/sitePvkLocation=<path+license.pvk>] [/sitePvkPassword=<pvk_pas
sword>]
```

Note: The local operators are enabled by default.

Note: When trying to disable the local operators, if the "REST API credentials for BES Server Plugin Service" are set and if the configured user is a local operator, an error message is displayed and the option is not set.

Note: When trying to disable the local operators, if the "SOAP API credentials for BES Server Plugin Service" are set, a non-blocking warning message is displayed and the option is set.

setcontrolflowguard

You can use this service to overwrite the Microsoft Control Flow Guard (CFG) system enablement for the program specified by the `/binaryName` parameter, where *<programName.exe>* is one of the following BigFix Server executables:

- BESAdmin.exe
- BESRootServer.exe
- FillDB.exe
- GatherDB.exe

These are the only BigFix binaries generated with the needed Microsoft Control Flow Guard (CFG) build options; enabling a binary not built with the correct flag is ignored by Windows OS.

The syntax to run this service is:

```
.\BESAdmin.exe /setcontrolflowguard { /enable | /disable | /remo
ve } { /all | /binaryName=<programName.exe> } [/hideUI]
```

- The `/enable` or `/disable` parameters overwrite the OS Windows system settings; the `/remove` parameter removes an overwrite previously set.
- When passing the `/all` parameter, the command changes the Microsoft Control Flow Guard (CFG) enablement of all the BigFix Server executables described above.
- Use the `/hideUI` option to avoid pop-up windows notifying action results.

For more details about the Microsoft Control Flow Guard (CFG), see Enabling Microsoft Control Flow Guard on BigFix Server.

setClientCertificatePolicy

Starting from Patch 6, you can use this service to set the maximum validity period for newly issued client certificates and define how long before expiration a BigFix client should attempt the renewal.

The syntax to run this service is:

```
.\BESAdmin.exe /setClientCertificatePolicy [/maxValidity <days>]
 [/startRefreshBefore <days>]
```

You need to specify the following parameters:

- `/maxValidity <days>` sets the maximum validity period for new BigFix client certificates.

  Value: An integer representing the number of days (e.g., 395 for approximately 13 months, 730 for 2 years). Alternatively, use the word

default to revert to the system default (e.g., 13 months for clients 10.0.7 and later).

Effect: This setting applies to certificates issued after the policy is applied and propagated. Existing certificates will be recreated in accordance with the new policy settings upon renewal.

- `/startRefreshBefore <days>` specifies how long before the certificate expiration date the BigFix client should attempt to initiate a renewal.

  Value: An integer representing the number of days (e.g., 60 for 60 days). Alternatively, use the word **default** to revert to the system default (e.g., 45 days).

  Note: The value for startRefreshBefore must be less than the maxValidity value.

  Recommendation: Use a value that provides ample opportunity for intermittently offline BigFix clients to renew successfully.

Policy Propagation

The `maxValidity` and `startRefreshBefore` settings are embedded in the **ActionSite.afxm** file using the following headers:

- **x-bes-client-certificate-max-validity**
- **x-bes-client-certificate-refresh-before**

Changes to these settings trigger the generation of a new version of the masthead. Existing BigFix clients receive these updated settings during regular masthead checks. The BigFix Server uses these settings to correctly handle client certificate requests.

setcustomca

Starting from Patch 4, you can use this service to install a custom CA intermediate certificate in the BigFix Platform.

If you use a CA intermediate certificate along with the CA root certificate, the syntax to run this service is:

```
.\BESAdmin.exe /setcustomca /customCARootCrtLocation=<path>
/customCACrtLocation=<path> /customCAPvkLocation=<path> [/custom
CAPvkPassword=<password>]
```

You need to specify the following parameters:

- `/customCARootCrtLocation` specifies the path where the CA root certificate is located.
- `/customCACrtLocation` specifies the path where the custom CA intermediate certificate is located.
- `/customCAPvkLocation` specifies the path where the private key of the custom CA intermediate certificate is located.
- `/customCAPvkPassword` (Optional) specifies the password associated with the private key of the custom CA intermediate certificate.

Otherwise, if you do not use a CA intermediate certificate but the CA root certificate only, the syntax to run the service is:

```
.\BESAdmin.exe /setcustomca /customCACrtLocation=<path>
/customCAPvkLocation=<path> [/customCAPvkPassword=<password>]
```

where you need to specify the following parameters:

- `/customCACrtLocation` specifies the path where the CA root certificate is located.
- `/customCAPvkLocation` specifies the path where the private key of the CA root certificate is located.
- `/customCAPvkPassword` (Optional) specifies the password associated with the private key of the CA root certificate.

For this service, it is also available a UI that you can launch by running:

```
.\BESAdmin.exe /setcustomca
```

without additional options.

Note: The CA root and intermediate certificates and the related keys are intended as server certificates/keys, they are not end-user certificates and keys.

Later on, you can uninstall the CA intermediate certificate and restore the standard functionality of the BigFix Platform by running this service:

```
.\BESAdmin.exe /removecustomca
```

If your environment has multiple BigFix Servers, see this section (on page 418).

setproxy

If your enterprise uses a proxy to access the Internet, you must set a proxy connection to enable the BigFix server to gather content from sites and to do component-to-component communication or to download files.

For information about how to run the command and about the values to use for each argument, see Setting a proxy connection on the server (on page 552).

setshareableserverdatapath

Starting from Patch 4, you can use this service to configure a folder for the BigFix Server data files that can be shared with another BigFix Server that is part of the same failover cluster. If you are configuring the BigFix Server to create an High Availability (HA) environment with a shared storage, you can use this command to place those files on a shared storage instead of replicating them. It adjusts your BigFix settings in the Windows registry and optionally copies the existing data files to the specified folder, without deleting the current copy. It does not change the locations of your BigFix logs.

The syntax to run this service is:

```
.\BESAdmin.exe /setshareableserverdatapath /topath="<folder path
>" [/copydata]
[/sitePvkFile=<path+license.pvk>] [/sitePassword=<password>] [/h
ideUI]
```

You need to specify the following parameters:

- `/topath` a parameter to specify the path of the folder that will be used for the BigFix Server shareable data files from now on.
- `/copydata` is an optional flag to copy the shareable data files from their current location to the new folder.
- `/hideUI` is an optional flag to avoid pop-up windows.

If you pass the `/copydata` flag together with the `/hideUI` flag, it is recommended looking at the BESAdmin log file to ensure the file copy operations are complete before restarting the BigFix Server.

After you verify the file copy is successful, you can delete the old copy of the data, without deleting the folders still used by the log files. For example, if you were using the default folder paths, you could remove the following folders.

```
[BigFix Server folder]\Encryption Keys
[BigFix Server folder]\FillDBData\fastbufferdir"
[BigFix Server folder]\FillDBData\BufferDir"
[BigFix Server folder]\GatherDBData\gather"
[BigFix Server folder]\Mirror Server"
[BigFix Server folder]\TrustedDownloadCerts"
[BigFix Server folder]\wwwrootbes"
[BigFix Server folder]\UploadManagerData\BufferDir
[BigFix Server folder]\UploadManagerData\TransientDir
```

setsqlserverparallelism

You can use this service to change a few SQL Server configuration parameters on your database instance for a more effective use of multiple CPU cores. You

can pass `"auto"` as the parameter value to let BESAdmin calculate and set an appropriate value for the parameter.
The syntax to run this service is:

```
.\BESAdmin.exe /setsqlserverparallelism { [ /maxdop={<integer>|"
auto"} ] [ /ctfp={<integer>|"auto"} ] }
```

You need to specify one or more of the following parameters:

- `/maxdop={<integer>|"auto"}` specifies the MaxDop value.
- `/ctfp={<integer>|"auto"}` specifies the CTFP value.

The value set for MaxDoP and CTFP must be a natural number (an integer >= 0).

To run this command, you must have these permissions on the database: either `sysadmin` or the `serveradmin` server role permissions are required.

syncmastheadandlicense

When you upgrade this product, you must use this option to synchronize the updated license with the masthead and resign all content in the database with SHA-384. Use the /hideUI option to avoid pop-up windows notifying action results.

The syntax to run this service is:

```
.\BESAdmin.exe /syncmastheadandlicense /sitePvkLocation=<path+li
cense.pvk>
[/sitePvkPassword=<pvk_password>] [/hideUI]
```

updatepassword

You can modify the password that is used for authentication by product components in specific configurations.

The syntax to run this service is:

```
.\BESAdmin.exe /updatepassword /type=<server_db|dsa_db>
[/password=<password>] /sitePvkLocation=<path+license.pvk>
[/sitePvkPassword=<pvk_password>]
```

where:

type=server_db

Specify this value to update the password that is used by the server to authenticate with the database.

If you modify this value, the command restarts all the BigFix server services.

type=dsa_db

Specify this value to update the password that is used in a DSA configuration by a server to authenticate with the database.

The settings `/password` and `/sitePvkPassword` are optional. If they are not specified in the command syntax, their value is requested interactively at run time. The password set by this command is obfuscated.

Special procedures for environments with multiple BigFix Server

Environments with multiple BigFix Server (DSA, HA, and DR) require special procedures to successfully achieve certain configuration changes, maintaining consistency and coherence.

In such environments, the following BESAdmin commands may require additional steps or multiple applications:

- `rotateexplorercredentials` and `revokeexplorercredentials`
- `rotateserversigningkey`
- `rotatewebuicredentials` and `revokewebuicredentials`
- `setcustomca` and `removecustomca`

Special procedures for DSA environments

If your BigFix environment is configured as a **DSA**, you may need to run the aforementioned BESAdmin commands on each BigFix Server to obtain the desired effect, not just on the primary.

For rotating the Server signing keys in a DSA environment, see this wiki article.

Similarly, when setting or removing a custom CA in a DSA environment, you will need to run the corresponding BESAdmin command on each BigFix Server computer. This operation is best performed with all BigFix Servers stopped and Consoles closed.

Special procedure for HA and DR environments

If your BigFix environment is configured for **HA or DR** and you intend to run one of the aforementioned BESAdmin commands, consider the following.

Note: Those BESAdmin commands will restart the BigFix Server if it is running, which may cause the Failover Cluster Manager to change the owner node of the BigFix Server cluster role and try to start it on a different computer. To avoid that, the first step will be ensuring no BigFix Server is running and only keep the SQL Server Availability Group running.

Perform the following steps:

- using the Failover Cluster Manager, stop the BigFix Server cluster role
- on the owner node of the BigFix Server role, run your `BESAdmin` command (e.g. `setcustomca`)
- on the same computer, run the `ServerKeyTool` command to **decrypt** the BigFix key files, instructions here
- **copy** the folder with the decrypted keys **to any other** computer hosting a BigFix Server
- on those computers, delete the encrypted keys in the `[BigFix Server folder]` and run the `ServerKeyTool` command to **encrypt** the BigFix key files you copied, instructions here
- using the Failover Cluster Manager, start the BigFix Server cluster role back up on the original owner node.

Working with TLS cipher lists

All network communications between the BigFix components and the internet are encrypted by using the TLS protocol standard. Starting from Version 9.5.11, master operators can

control which TLS ciphers should be used for encryption. A master operator can set a deployment-wide TLS cipher list in the masthead by using BESAdmin.

The TLS cipher list is a colon-delimited list of cipher suites or cipher families. To disable a cipher suite or cipher family, precede the name with "!".

The default TLS 1.2 cipher list which is `HIGH:!ADH:!AECDH:!kDH:!kECDH:!kRSA:!PSK:!SRP:!SHA1` is used when no TLS cipher list at all or no TLS 1.2 cipher suite is present in the masthead.

The default TLS 1.3 cipher list which is `TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256` is used when no TLS cipher list at all or no TLS 1.3 cipher suite is present in the masthead.

This defines the master set of TLS cipher suites from which you can select. Cipher suites that are not in this master set are either insecure or incompatible with the BigFix components. In addition, the TLS cipher list must include at least one cipher suite using RSA key exchange for the BigFix HTTPS servers. The following BESAdmin commands help you create the TLS cipher list:

testTLSCipherList

To test if a particular TLS cipher list is compatible with the BigFix components, run the following command:

```
.\BESAdmin.exe /securitysettings /sitePvkLocation=<path+license.pvk> /sitePvkPassword=<password> /testTLSCipherList=<cipher_1>:<cipher_2>:..:<cipher_n>
```

For example:

```
.\BESAdmin.exe /securitysettings /sitePvkLocation=C:\licenses\license.pvk /sitePvkPassword=bigfix /testTLSCipherList="TLSv1.2:!ADH:!AECDH:!kDH:!kECDH:!PSK:!SRP:!NULL"
```

If the command runs successfully, BESAdmin provides a detailed list of all TLS cipher suites that are enabled. If unsuccessful, BESAdmin provides a detailed list of which cipher suites are insecure or incompatible.

setTLSCipherList

After identifying a suitable TLS cipher list, you can set it with the following command:

```
.\BESAdmin.exe /securitysettings /sitePvkLocation=<path+license.pvk> /sitePvkPassword=<password> /setTLSCipherList=<cipher_1>:<cipher_2>:..:<cipher_n>
```

For example:

```
.\BESAdmin.exe /securitysettings /sitePvkLocation=C:\licenses\license.pvk /sitePvkPassword=bigfix /setTLSCipherList="TLSv1.2:!ADH:!AECDH:!kDH:!kECDH:!PSK:!SRP:!NULL"
```

If the command is unsuccessful, BESAdmin provides a detailed list of which cipher suites are insecure or incompatible. The ciphers on the list are arranged in an order of preference. To modify the order by key length, add @STRENGTH.

Note: BESAdmin does not verify if the name of a particular cipher suite or cipher family is available; it only checks the final set of TLS cipher suites that is implied by the colon delimited list.

listTLSCiphers

For a detailed list of all the TLS ciphers that are currently enabled, run the following command:

```
.\BESAdmin.exe /securitysettings /sitePvkLocation=<path+license.pvk> /sitePvkPassword=<password> /listTLSCiphers
```

For example:

```
.\BESAdmin.exe /securitysettings /sitePvkLocation=C:\licenses\license.pvk /sitePvkPassword=bigfix /listTLSCiphers
```

removeTLSCipherList

To remove a TLS cipher list from the deployment masthead and return to the default cipher list, run the following command:

```
.\BESAdmin.exe /securitysettings /sitePvkLocation=<path+license.pvk> /sitePvkPassword=<password> /removeTLSCipherList
```

For example:

```
.\BESAdmin.exe /securitysettings /sitePvkLocation=C:\licenses\license.pvk /sitePvkPassword=bigfix /removeTLSCipherList
```

The detailed ciphers that are available for a given cipher family depends on the version of OpenSSL that is in use. At its core, the TLS cipher list is the OpenSSL cipher string. For more details, see OpenSSL Cryptography and SSL/TLS Toolkit. Do not use this feature if you are not familiar with the basics of TLS cryptography.

Note: These commands return two different lists:

- The TLS ciphers list accepted by the BigFix components (such as the Root Server, Relay, Web Reports..) during the handshake when they receive incoming requests. This restricted list is used when the BigFix components act like the servers and it is mentioned in the BigFix Administration Tool as **ciphers allowed in server connections**. The communication between the BigFix components cannot use ciphers that are not included in this list.
- The additional TLS ciphers list used by the BigFix components, when starting an external connection (e.g. when a file must be downloaded from an external URL). This list follows the **Additional TLS 1.X ciphers allowed in client connections** message.

BESAdmin Linux Command Line

The BigFix Server installer places the script to run the BigFix Administration Tool, `BESAdmin.sh`, in the `/opt/BESServer/bin` directory.

To prevent conflicts, avoid running BESAdmin commands concurrently. Only one BESAdmin instance should run at a time within a BigFix environment.

Starting with version 11.0.7, a BESAdmin command on Linux will print an error and exit without taking action if another BESAdmin command is already running on the same computer. This error message will not be logged.

With this tool you can edit the masthead file, check the signatures of the objects in the database, enable and disable enhanced security, resign all of the users content in the database, rotate the server private key, configure the Console and Web Reports login, resign the database content, and synchronize the masthead with the updated license.

Run this script, as super user from the command prompt, using the following syntax:

```
./BESAdmin.sh -service {arguments}
```

where *service* can be one of the following:

```
audittrailcleaner
changeprivatekeypassword
createexplorercredentials
createrepositorysiteregistrar
createtokenkey
createwebuicredentials
editmasthead
findinvalidactions
findinvalidsignatures
getcertificatebundle
healthcleanup
importlicense
minimumsupportedclient
minimumsupportedrelay
propagateoperatorsites
propertyidmapper
removecomputers
repair
```

```
reportencryption
resetdatabaseepoch
resignsecuritydata
revokeexplorercredentials
revoketokenkey
revokewebuicredentials
rotateexplorercredentials
rotaterepositorysiteregistrar
rotateserversigningkey
rotatetokenkey
rotatewebuicredentials
securitysettings
setadvancedoptions
setClientCertificatePolicy
setcustomca
setproxy
syncmastheadandlicense
testproxyconnection
updatepassword
```

Note: The notation `<path+license.pvk>` used in the command syntax stands for *`path_to_license_file`*`/license.pvk`.

Each service has the following *`arguments`*:

audittrailcleaner

You can run this service to remove historical data from the bfenterprise database that is stored to serve as an audit trail. This audit trail slowly increases in size over the lifetime of a BigFix deployment. The audit trail contains deleted and earlier versions of Fixlets, tasks, baselines, properties, mailbox files, actions, and analyses. The audit trail is not used by BigFix in any way and can be deleted to reduce the database size. BigFix recommends that

you create a historic archive of the current database and save it to a secure location before running this tool to preserve the audit trail, thus removing it from the product database, but not completely deleting the history.

The service can count and delete the following sets of data:

- **Older Versions of Custom Authored Content** (`-oldcontent`): Every edit to Fixlets, Tasks, Baselines, and Analyses, creates a new version, the earlier versions can be deleted.
- **Older Versions of Actions** (`-oldactions`): Any time you stop or start an Action, a new version is created; the earlier versions can be deleted.
- **Older Versions of relay.dat** (`-oldrelaydatfile`): Any time you install or uninstall a new relay, a new version is created; the earlier versions can be deleted.
- **Older uploaded files** (`-deleteolduploadedfiles`): Removes the old files uploaded by the Archive Manager on the BigFix server. This option deletes the old files after an expiration period (default 180 days) from when they were uploaded.
- **Deleted Custom Authored Content (all versions)** (`-deletedcontent`): When you delete a Fixlet, Task, Baseline, and Analysis using the console, the data is marked as deleted in the database and preserved. The deleted content, including all of the earlier versions, and the corresponding client reports can be deleted.
- **Deleted Actions(all versions)** (`-deletedactions`): When you delete an action using the console, the data is marked as deleted in the database and preserved. The deleted actions, including all of the earlier versions, and the corresponding client reports can be deleted.
- **Useless Action Results** (`-uselessactionresults`): Earlier versions of BigFix might cause clients to report ActionResults that were not used in any way but would use up space in the database. These useless ActionResults can be deleted.
- **Orphaned sub-actions** (`-orphanedsubactions`): From multiple action groups that were deleted.

- **Hidden Manual Computer Group Actions** (`-hiddenactions`): Manual Computer Groups create hidden actions that add and remove computers to and from groups and the actions can build up over time. This option deletes actions after an expiration period (default 180 days) from when they were created.
- **Older Version of Mailbox Files** (`-deletedmailbox`): Deleted Mailbox Files are stored in a table in the database and can be removed.
- **Old audit log files** (`-deleteauditlogs`): Removes old audit log files to prevent the server running out of disk space.
- **Synchronizing BES Consoles** (`-syncconsoles`): The BigFix Console maintains a local cache of the database that becomes not synchronized when data is removed with this tool. To prevent this situation from happening, the tool sets a flag in the database to force all BigFix Consoles to reload the cache when the Console is started up.
- **Removing data older than** (`-olderthan`): Removes data earlier than a specified date. The default value is 99 days.
- **Batched deletion** (`-batchsize`): Deleting large sets of data causes the SQL transaction log to quickly increase in size, the log becomes temporarily larger than the data being removed until the database is shrunk. Batched deletion removes results in sets.
- **Deadlock Priority** (`-databaseDeadlockPriority`): Sets the priority for the service in case of transaction deadlocks in the database. Available values are Low, Normal and High. The default value is Normal.



Note: The `-databaseDeadlockPriority` command option is available only if you are using Microsoft SQL Server database, not if you are using DB2 database.

The syntax of this service changes depending on the action you specify:

```
./BESAdmin.sh -audittrailcleaner { -displaysettings | -run [dele
te_data_options] |
```

```
          -schedule [delete_data_options] [scheduling options] |
 -preview [delete_data_options]
           [preview options] }
```

```
./BESAdmin.sh -audittrailcleaner -displaysettings
```

```
./BESAdmin.sh -audittrailcleaner -run [ -oldcontent ] [ -oldacti
ons ]
          [ -oldrelaydatfile ] [ -deleteolduploadedfiles ] [ -de
letedcontent ] [ -deletedactions ]
          [ -uselessactionresults ] [ -orphanedsubactions ] [ -h
iddenactions=<days> ]
          [ -deletedmailbox ] [ -deleteauditlogs ] [ -syncconsol
es ] [ -olderthan=<days> ] [ -batchsize=<size> ] [ -databaseDead
lockPriority=<Low|Normal|High> ]
```

```
./BESAdmin.sh -audittrailcleaner -sitePvkLocation=<path+license.
pvk>
      [ -sitePvkPassword=<password> ] -schedule
 [ [ -oldcontent ] [ -oldactions ]
      [ -oldrelaydatfile ] [ -deleteolduploadedfiles ] [ -delete
dcontent ] [ -deletedactions ] [ -uselessactionresults ]
      [ -orphanedsubactions ] [ -hiddenactions=<days> ] [ -delet
edmailbox ] [ -deleteauditlogs ] [ -syncconsoles ]
      [ -olderthan=<days> ] [ -batchsize=<size> ] [ -databaseDea
dlockPriority=<Low|Normal|High> ] [ -cleanstarttime=<yyyymmdd:hh
mm> ]
      [ -cleanperiodicinterval=<hours> ] ] | [ -disable ]
```

```
./BESAdmin.sh -audittrailcleaner -preview [ [ -oldcontent ] [ -o
ldactions ] [
      -oldrelaydatfile ] [ -deleteolduploadedfiles ] [ -deletedc
ontent ] [ -deletedactions ] [ -uselessactionresults ] [
```

```
      -orphanedsubactions ] [ -hiddenactions=<days> ] [ -deleted
mailbox ] [ -deleteauditlogs ] [ -olderthan=<days> ]
      [ -databaseDeadlockPriority=<Low|Normal|High> ] | [ -sched
uled ] ]
```

where:

- **`displaysettings`** shows the settings that are previously defined with the `schedule` action.
- **`run`** runs the tool with the specified settings. Before you use this option, check the settings that affect the database by using the **`preview`** action.
- **`schedule`** schedules the tool to run at the specified time at each specified interval. To disable the schedule action, use the `-disable` option.
- **`preview`** shows the number of database rows that are affected by the specified settings. If no setting is passed to the preview option, the preview performs the count by setting all options to true and using the default values for dates. Use the `-scheduled` option to preview the scheduled settings.

For information about the cleanup tasks log files, see Logging Cleanup Tasks Activities (on page 467).

changeprivatekeypassword

You can use this service to be prompted for a new password to associate to the `license.pvk` file. Use the following syntax to run the command:

```
./BESAdmin.sh -changeprivatekeypassword -sitePvkLocation=<path+l
icense.pvk>
[ -sitePvkPassword=<password> ]
```

createexplorercredentials

Use this service to to create a BigFix Explorer certificate for the input hostname, in a similar way as the one for the WebUI certificate. The following files will be created in the output folder:

- The certificate file of the ExplorerCertificate "authentication": **auth_cert.crt**
- The key file of the ExplorerCertificate "authentication": **auth_key.key**
- The certificate file of the ExplorerCACertificate "authentication": **ca.crt**
- The certificate file of the WebUICACertificate "authentication": **apica.crt**.

Use the following syntax to run the command:

```
./BESAdmin.sh -createexplorercredentials
-sitePvkLocation=<path+license.pvk>
-sitePvkPassword=<password>
-explorerCertDir=<path>
-explorerHostname=<BigFixExplorerHostnameOrIP>
[ -f ]
```

This service generates a folder named cert_*explorerHostname* in the path specified by the **explorerCertDir** option.

explorerCertDir

Specifies the path to the parent folder of the new folder containing the certificates. This folder must exist.

explorerHostname

Specifies the hostname or IP address of the computer that will host your BigFix Explorer.

f

Forces new certificate creation, without checking if a certificate for the same hostname exists (if it exists, it will overwrite it).

createrepositorysiteregistrar

Use this service to create and store the permanent External Repository Site Registrar. It uses the deployment credentials supplied to BESAdmin and must be run before importing an External Repository Site when a registrar does not already exist. The installer creates it automatically for a new Root Server installation. Use the `rotaterepositorysiteregistrar` command only when a replacement is required. This command relates to a feature introduced in 11.0.7 named External Repository Sites.

```
./BESAdmin.sh -createrepositorysiteregistrar -sitePvkLocation=<p
ath+license.pvk>
-sitePvkPassword=<password>
```

createtokenkey

Use this service to create a secret key to issue authentication tokens, which is necessary to use the bearer token authentication feature introduced in 11.0.6. For more information, see Configuring bearer token authentication.

```
./BESAdmin.sh -createtokenkey -sitePvkLocation=<path+license.pvk
>
-sitePvkPassword=<password>
```

createwebuicredentials

Use this service to generate the certificates used as WebUI credentials. Use the following syntax to run the command:

```
./BESAdmin.sh -createwebuicredentials
-sitePvkLocation=<path+license.pvk>
-sitePvkPassword=<password> -webUICertDir=<path>
-webUIHostname=<WebUIHostnameOrIP>
[ -f ]
```

This service generates a folder named cert_*WebUIHostnameOrIP* in the path specified by the **webUICertDir** option.

webUICertDir

Specifies the path to the parent folder of the new folder containing the certificates. This folder must exist.

webUIHostname

Specifies the hostname or IP address of the computer that will host your WebUI.

f

Does not check if a WebUI certificate already exists before creating a new one. If it exists for the same hostname, it will overwrite it.

Note: If you need to generate WebUI credentials certificates, but you have no WebUI in your deployment, then set:

webUICertDir

To the BigFix server folder (`/var/opt/BESServer`).

webUIHostname

To the BigFix server IP address or hostname.

editmasthead

You can edit the masthead file by specifying the following parameters:

```
advGatherSchedule (optional, integer)
 values:
    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)
 values:
    0=console,
    1=client,
    2=nobody
advInitialLockState (optional, integer)
 values:
    0=Locked,
    1=timed (specify duration),
    2=Unlocked
advInitialLockDuration (optional, integer)
 values:
   ( duration in seconds )
advActionLockExemptionURL (optional, string)
advRequireFIPScompliantCrypto (optional, boolean)
advEnableFallbackRelay (optional,boolean)
advFallbackRelay (optional, string)
```

The syntax to run this service is:

```
./BESAdmin.sh -editmasthead -sitePvkLocation=<path+license.pvk>
[ -sitePvkPassword=<password> ][ -display ]
[ -advGatherSchedule=<0-10> ] [ -advController=<0-2> ]
[ -advInitialLockState=<0|2> | -advInitialLockState=1
-advInitialLockDuration=<num> ] [ -advActionLockExemptionURL=<ur
l> ]
[ -advRequireFIPScompliantCrypto=<true|false> ] [ -advEnableFall
backRelay=0 |
-advEnableFallbackRelay=1 -advFallbackRelay=<host> ]
```

For additional information, see Editing the Masthead on Linux systems in the *BigFix Configuration Guide*.For additional information, see Editing the Masthead on Linux systems in the *BigFix Configuration Guide*.

findinvalidactions

You can check for invalid actions in the database by specifying the following parameter:

- (Optional) -deleteInvalidActions: Deletes invalid actions.

The syntax to run this service is:

```
./BESAdmin.sh -findinvalidactions [ -deleteInvalidActions ]
-sitePvkLocation=<path+license.pvk> [ -sitePvkPassword=<password
> ]
```

findinvalidsignatures

You can check the signatures of the objects in the database by specifying the following parameters:

-list (optional)

Lists all invalid signatures that `BESAdmin` finds.

-resignInvalidSignatures (optional)

Attempts to resign any invalid signatures that `BESAdmin` finds.

-deleteInvalidlySignedContent (optional)

Deletes contents with invalid signatures.

For additional information about invalid signatures, see Resolving invalidly signed content problems in the console. The syntax to run this service is:

```
./BESAdmin.sh -findinvalidsignatures
[ -list | -resignInvalidSignatures | -deleteInvalidlySignedConte
nt ]
```

getcertificatebundle

You can export the certificate bundle (PEM) used by the current version of BES Admin. In the bundle, there are all the certificates of all the authorized chains in the masthead. So, for example, with SHA384 forced, in the bundle there is only the SHA384 chain. The generated file, named bundle.pem, is located in the folder specified by the bundleCertDir=<path> option. If you specify the **getCustomCA** option, the command exports any custom CA certificates previously set by the user.
The syntax to run this service is:

```
./BESAdmin.sh -getcertificatebundle -sitePvkLocation=<path+licen
se.pvk>
[ -sitePvkPassword=<password> ] -bundleCertDir=<path> [ -getCust
omCA ]
```

healthcleanup

You can mark as deleted the expired or stopped actions if a specific number of days, which is configurable, has passed since the actions were stopped or expired. This feature is available starting from BigFix Version 11.0.6 and later.
The general syntax to run this service is:

```
./BESAdmin.sh -healthcleanup  { -displaySettings  | -run [run_he
althcleanup_options]
       | -schedule [schedule_healthcleanup_options]
       | -preview [preview_healthcleanup_options] }
```

Depending on the action that is specified, the syntax changes as follows:

```
./BESAdmin.sh -healthcleanup -displaySettings
```

```
./BESAdmin.sh -healthcleanup -run  [ -deleteExpiredStoppedAction
s=<days> ]
       [ -databaseDeadlockPriority=<Low|Normal|High> ]
```

```
./BESAdmin.sh -healthcleanup -schedule  [ [ -deleteExpiredStoppe
dActions=<days> ]
```

```
[ -databaseDeadlockPriority=<Low|Normal|High> ]
[ -startTime=<YYYYMMDD:HHMM> [ -periodicInterval=<Hours> ] ] | -
disable | -delete ]
```

```
./BESAdmin.sh -healthcleanup -preview [ [ -deleteExpiredStoppedA
ctions=<days> ]
[ -databaseDeadlockPriority=<Low|Normal|High> ] | -scheduled ]
```

where:

- **displaySettings** shows the settings that are previously set with the schedule action.
- **run** runs the tool with the specified settings.
  - **deleteExpiredStoppedActions=<days>** Defines the retention period, in days, after which expired or stopped actions will be deleted. The number represents how many days have passed since the action was stopped or expired.
  - **databaseDeadlockPriority=<Low|Normal|High>** Sets the priority during the cleanup operation in case of transaction deadlocks in the database. Available values are Low, Normal and High. The default value is Normal. This setting is applicable only for Microsoft SQL Server databases while it must not be set on DB2.
- **schedule** Creates or updates a scheduled Health and Cleanup task using the specified options. The task will run automatically starting from the given start time and with the selected cleanup settings.
  - **deleteExpiredStoppedActions=<days>** Defines the retention period, in days, after which expired or stopped actions will be deleted. The number represents how many days have passed since the action was stopped or expired.
  - **databaseDeadlockPriority=<Low|Normal|High>** Sets the priority during the cleanup operation in case of transaction deadlocks in the database. Available values are Low, Normal and High.

The default value is Normal. This setting is applicable only for Microsoft SQL Server databases while it must not be set on DB2.

- **`startTime=<YYYYMMDD:HHMM>`** Specifies the date and time when the scheduled task will start. For example, `startTime=20260203:1430`
- **`periodicInterval=<Hours>`** Specifies the interval, in hours, at which the scheduled task will run after the start time. If set to 0, the task will run only once. If set to a value greater than 0, the task will run repeatedly at the specified hourly interval. For example `periodicInterval=6`
- **`disable`** Disables the existing scheduled Health and Cleanup task, if present.
- **`delete`** Deletes the existing scheduled Health and Cleanup task, if present.

- **`preview`** Displays a preview of the Health and Cleanup operation without performing any changes to the database. The preview shows how many actions would be affected using the specified options or the currently configured values.
  - **preview (without additional options)** Generates a preview using the default Health and Cleanup settings.
  - **preview (with options)** When combined with options such as `deleteExpiredStoppedActions=<days>` or `databaseDeadlockPriority=<Low|Normal|High>`, the preview is generated using the provided values.
  - **preview -scheduled** Generates a preview using the values saved in the currently scheduled Health and Cleanup task. If no scheduled Health and Cleanup task exists or if the task is not active, an error message is displayed.

importlicense

You can use this service to import an updated license. This service allows you to update the license manually in isolated BigFix environments.

```
./BESAdmin.sh -importlicense
 -sitePvkLocation=<path+license.pvk>
[ -sitePvkPassword=<password> ] -licenselocation=<path+license.c
rt>
```

The `license.crt` file contains the updated license to import.

minimumsupportedclient

This service defines the minimum version of the BigFix Agents that are used in your BigFix environment.

Note: Based on this setting, the BigFix components can decide when it is safe to assume the existence of newer functions across all the component in the deployment. Individual agent interactions might be rejected if the interaction does not comply with the limitations that are imposed by this setting.

The currently allowed values are:

- **0.0**, which means that no activity that is issued by BigFix Agents earlier than V9.0, such as archive files and reports uploads, is prevented from running or limited. This behavior applies also if the `minimumsupportedclient` service is not set.
- **9.0**, which means that:
  - Unsigned reports, such as the reports sent by BigFix Clients earlier than V9.0, are discarded by FillDB.
  - The upload of an unsigned archive file that is generated on a BigFix Client earlier than V9.0, by an **archive now** command, for example, fails.

If you ran a fresh installation of BigFix V9.5.6 or later using a BES Authorization file, by default all the BigFix Clients earlier than

V9.0 are prevented from joining your environment because the `minimumsupportedclient` service is automatically set to **9.0**.

The value that is assigned to this service, if set, remains unchanged:

- If you upgraded to V9.5.6 or later.
- If you installed BigFix V9.5.6 or later using an existing masthead.

In both cases, if the service did not exist before, it will not exist afterward as well.
The current value `<VALUE>` assigned in your environment to the `minimumsupportedclient` service is displayed in the line `x-bes-minimum-supported-client-level: <VALUE>` of the masthead file. You can see the current value by running the following query on the BigFix Server from the BigFix Query Application available on the BigFix WebUI:

```
Q: following text of last ": " of line whose (it starts with
 "x-bes-minimum-supported-client-level:" ) of masthead of site "
actionsite"
```

The syntax to run this service is:

```
 ./BESAdmin.sh -sitePvkLocation=<path+license.pvk> [-sitePvkPass
word=<password>]
    -minimumsupportedclient=<version>.<release>
```

If you omit to specify `[sitePvkPassword=<password>]`, you are prompted to enter the password interactively when the **BESAdmin.sh** runs.

For example, if you want to state that Agents earlier than V9.0 are not supported in your BigFix environment, you can run the following command:

```
 ./BESAdmin.sh -sitePvkLocation=/license/license.pvk -minimumsup
portedclient=9.0
```

minimumsupportedrelay

You can use this service, added with BigFix V9.5.6, to enforce specific criteria that affect the BigFix Agent registration requests. If this service is enabled, V9.5.6 Agents can continue to register to the V9.5.6 BigFix environment if their registration requests are signed and sent across the Relays hierarchy using the HTTPS protocol.

Note: Based on this service, the BigFix components can decide when it is safe to enable newer functions across all the component in the deployment. Individual agent interactions might be rejected if they do not comply with the limitations that are imposed by this setting.

The currently allowed values are:

- **0.0.0**, which means that the BigFix Server accepts and manages:
  - Signed and unsigned registration requests coming from BigFix Agents.
  - Registration requests delivered from BigFix Agents using the HTTP or the HTTPS protocols.

  This behavior applies by default when you upgrade from previous versions to BigFix V9.5.6 or later. In this case, the `minimumsupportedrelay` service is not added automatically to your configuration during the upgrade.
- **9.5.6** or later, which means that:
  - The BigFix Server enforces that registration requests coming from BigFix Agents V9.5.6 or later must be properly signed.
  - The BigFix Server and the Relays V9.5.6 or later enforce the use of the HTTPS protocol when BigFix Agent registration data is exchanged.

  Enforcing this behavior has the following side effects:

- BigFix Agents earlier than V9.0 cannot send registration requests to the BigFix Server because they cannot communicate using the HTTPS protocol.
- Because BigFix Relays with versions earlier than V9.5.6 cannot handle correctly signed registration requests, any BigFix Client that uses those Relays might be prevented from continuing to register, or might fall back to a different parent Relay or directly to the Server.

If you ran a fresh installation of BigFix V9.5.6 or later using a License Authorization file, be aware that the side effects that were just listed apply to your BigFix deployment because, in this particular installation scenario, the `minimumsupportedrelay` service is automatically set to **9.5.6** by default.

The current value `<VALUE>` assigned in your environment to the `minimumsupportedrelay` service is displayed in the line `x-bes-minimum-supported-relay-level: <VALUE>` of the masthead file. You can see the current value by running the following query on the BigFix Server from the BigFix Query Application available on the BigFix WebUI:

```
Q: following text of last ": " of line whose (it starts with
"x-bes-minimum-supported-relay-level:" ) of masthead of site "ac
tionsite"
```

This query displays a value only when `<VALUE>` is set to **9.5.6**; if it is set to **0.0.0**, it does not display a value.
The syntax to run this service is:

```
 ./BESAdmin.sh -sitePvkLocation=<path+license.pvk> [-sitePvkPass
word=<password>]
    -minimumsupportedrelay=<version>.<release>.<modification>
```

If you omit to specify `[sitePvkPassword=<password>]`, you are prompted to enter the password interactively when the **BESAdmin.sh** runs.

For example, if you want that only the registration requests that are signed and carried through HTTPS are managed by your BigFix Server, you can run the following command:

```
 ./BESAdmin.sh -sitePvkLocation=/license/license.pvk -minimumsup
portedrelay=9.5.6
```

propagateoperatorsites

This service forces the server to propagate a new version of the operator sites. This command is useful after a server migration because you can be sure that data is available for clients to gather and it prevents from failures. This is the command syntax:

```
./BESAdmin.sh -propagateoperatorsites { -propagateAllOperatorSit
es |
-propagateOperatorSite=<MastheadUsername> }
```

propertyidmapper

This service creates, updates, and deletes a table (PropertyIDMap) in the BFEnterprise database that maps retrieved property names for the SiteID, AnalysisID, PropertyID used to reference properties in the QUESTIONRESULTS and LONGQUESTIONRESULTS tables. It creates the PropertyIDMap table if it does not exist (requires table creation permissions). This service must be run to update the PropertyIDMap table after creating or deleting a property.

The general syntax of this service is the following:

```
./BESAdmin.sh -propertyidmapper  { -displaysettings | -run [prop
erty_idmapper_options]
       |  -schedule [property_idmapper_options] [scheduling opti
ons] }
```

The syntax of this service changes depending on the action you specify:

```
./BESAdmin.sh -propertyidmapper -displaysettings
```

```
./BESAdmin.sh -propertyidmapper -run [ -createtable ] [ -removet
able ]
      [ -lookupproperty=<propertyname> ]
```

```
./BESAdmin.sh -propertyidmapper -schedule [ -createtable -startt
ime=<yyyymmdd:hhmm>
     [ -interval=<hours> ] | -disable ]
```

where:

- **`displaysettings`** shows the settings that are previously set with the `schedule` action.
- **`run`** runs the tool with the specified settings. Before you use this option, check the settings that affect the database by using the `preview` action.
- **`schedule`** schedules the tool to run at the specified time at each specified interval. To disable the schedule action, use the `-disable` option.

For more information about the cleanup tasks log files, see Logging Cleanup Tasks Activities (on page 467).

removecomputers

The service runs database operations for the following sets of data:

- **Expired Computers** (`-deleteExpiredComputers`) Marks computers as *deleted* if they did not report in recently.
- **Deleted Computers** (`-purgeDeletedComputers`): Physically deletes the computer related data from the database for computers that are already marked as deleted and have not reported in for a long time. It deletes the data related to an agent (such as the action results or the properties, and so on), not the agent itself that remains logically deleted (IsDeleted = 1) on the database. Therefore, as a consequence, if the same agent becomes active again, it is recognized and will reuse its previous computer ID.

- **Duplicate Computers** (`-deleteDuplicatedComputers`): Marks older computers as deleted if a computer exists with the same computer name.
- **Removal of deleted Computers** (`-removeDeletedComputers`): Physically deletes the computer information from the database for computers that are marked as deleted (IsDeleted = 1) since at least the indicated number of days (minimum 7) or the indicated number of hours (minimum 24). It deletes the information of the agent itself (such as the computer ID, and so on). Therefore, as a consequence, if the same agent becomes active again, a totally new computer ID will be assigned to the agent.
- **Removal of uploaded Files** (`-removeDeletedUploads`): Physically removes from the database the definition of uploaded files that are marked as deleted. It does not apply to non-native agents.
- **Removal of uploaded files of removed computers** (`-eraseUploadFilesForRemovedComputers`): Physically removes from the BigFix server filesystem all files uploaded by clients whose definition has been removed from the database. It does not apply to non-native agents.
- **Removal of Computers by name** (`-removeComputersFile`): Accepts a text file with a list of computer names that are separated by new lines and removes them from the deployment.
- **Batched deletion** (`-batchsize`): Deleting large sets of data causes the SQL transaction log to quickly increase in size, the log becomes temporarily larger than the data being removed until the database is shrunk. Batched deletion removes results in sets.

- **Deadlock Priority** (`-databaseDeadlockPriority`): Sets the priority for the service in case of transaction deadlocks in the database. Available values are Low, Normal and High. The default value is Normal.

  Note: The `-databaseDeadlockPriority` command option is available only if you are using Microsoft SQL Server database, not if you are using DB2 database.

The general syntax of this service is:

```
./BESAdmin.sh -removecomputers  { -displaySettings [display_sett
ings options]  | -run [remove_computers_options]
       | -schedule [remove_computers_options] [scheduling option
s]
       | -preview [remove_computers_options] [preview options] }
```

Depending on the action that is specified, the syntax changes as follows:

```
./BESAdmin.sh -removecomputers -displaySettings [ -name=<TaskNam
e> ]
```

```
./BESAdmin.sh -removecomputers -run [ -agentType=<AgentType> ] [
 -deleteExpiredComputers=<days> ]
    [ -removeDeletedComputers=<days> ] [ -removeDeletedUploads ]
    [ -eraseUploadFilesForRemovedComputers ]
    [ -purgeDeletedComputers=<days> ]
    [ -deleteDuplicatedComputers [ -duplicatedPropertyName=<Prop
ertyName> ] ]
    [ -removeComputersFile=<path> ] [ -batchSize=<batch size> ]
 [ -databaseDeadlockPriority=<Low|Normal|High> ]
```

```
./BESAdmin.sh -removecomputers -schedule [ [ -name=<TaskName> ]
 [ -agentType=<AgentType> ] [ -deleteExpiredComputers=<days> ] [
 -purgeDeletedComputers=<days> ]
```

```
[ -removeDeletedComputers=<days> ] [ -removeDeletedUploads ] [ -
eraseUploadFilesForRemovedComputers ]
[ -deleteDuplicatedComputers [ -duplicatedPropertyName=<Property
Name> ] ] [ -batchSize=<batch size> ] [ -databaseDeadlockPriorit
y=<Low|Normal|High> ]
[ -removeStartTime=<YYYYMMDD:HHMM> [ -removePeriodicInterval=<Ho
urs> ] ] | [ -disable -name=<TaskName> ] | [ -delete -name=<Task
Name> ] | [ -list ] |
[ -update [ -name=<TaskName> ]
 [ -deleteExpiredComputers=<days> ] [ -purgeDeletedComputers=<da
ys> ]
[ -removeDeletedComputers=<days> ] [ -removeDeletedUploads ] [ -
eraseUploadFilesForRemovedComputers ]
[ -deleteDuplicatedComputers [ -duplicatedPropertyName=<Property
Name> ] ] [ -batchSize=<batch size> ]
[ -databaseDeadlockPriority=<Low|Normal|High> ] [ -removeStartTi
me=<YYYYMMDD:HHMM> [ -removePeriodicInterval=<Hours> ] ] ] ]
```

```
./BESAdmin.sh -removecomputers -preview [ [ -agentType=<AgentTyp
e> ] [ -deleteExpiredComputers=<days> ]
    [ -removeDeletedComputers=<days> ] [ -removeDeletedUploads ]
    [ -eraseUploadFilesForRemovedComputers ]
    [ -purgeDeletedComputers=<days> ][ -deleteDuplicatedComputer
s
    [ -duplicatedPropertyName=<PropertyName> ] ] [ -databaseDead
lockPriority=<Low|Normal|High> ] | [ -scheduled ] [ -name=<TaskN
ame> ] ]
```

where:

- **`displaySettings`** shows the settings that are previously set with the `schedule` action.
- **`run`** runs the tool with the specified settings. Before you use this option, check the settings that affect the database by using the `preview` action.
- **`schedule`** schedules the tool to run at the specified time at each specified interval. To disable the schedule action, use the `-disable` option.
- **`preview`** shows the number of database rows that are affected by the specified settings. If no setting is passed to the preview option, the preview performs the count by setting all options to true and using the default values for dates. Use the `-scheduled` option to preview the scheduled settings.



Note: When using option `-removeDeletedComputers`, the number of days must be not less than 7 or the number of hours must be not less than 24.

For more information about the cleanup tasks log files, see Logging Cleanup Tasks Activities (on page 467).

repair

You can use this command to handle an inconsistency between the keys that are stored in the database and the keys stored on the filesystem.

```
./BESAdmin.sh -repair -sitePvkLocation=<path+license.pvk>
[ -sitePvkPassword=<password> ]
```

If the keywords `ServerSigningKey` and `ClientCAKey` do not exist, they are created under `/var/opt/BESServer`: This command also updates the licenses of sites.

reportencryption

You can generate, rotate, enable, and disable encryption for report messaging by running:

```
./BESAdmin.sh -reportencryption { -status |
  -generatekey [-privateKeySize=<min|max>]
               [-deploynow=yes | -deploynow=no -outkeypath=<path
>]
               -sitePvkLocation=<path+license.pvk> [-sitePvkPass
word=<password>] |
  -rotatekey [-privateKeySize=<min|max> ]
             [-deploynow=yes | -deploynow=no
 -outkeypath=<path> ]
             -sitePvkLocation=<path+license.pvk> [-sitePvkPasswo
rd=<password>] |
  -enablekey -sitePvkLocation=<path+license.pvk> [-sitePvkPasswo
rd=<password>] |
  -disable -sitePvkLocation=<path+license.pvk> [-sitePvkPassword
=<password>] }
```

where:

status

Shows the status of the encryption and which arguments you can use for that status.

generatekey

Allows you to generate a new encryption key.

rotatekey

Allows you to change the encryption key.

enablekey

Allows you to enable the encryption key.

disable

Allows you to put the encryption key in PENDING state. If you run again the `reportencryption` command with the `disable`

argument, the encryption changes from PENDING state to DISABLED.

deploynow=yes

Deploys the report encryption key to the server for decryption.

deploynow=no -outkeypath=<path>

The encryption key is not deployed to the server but it is saved in the `outkeypath` path.

For more information about this command and its behavior, see Managing Client Encryption.

resetdatabaseepoch

To clear all console cache information in BigFix Enterprise Service V7.0 or later versions. After running this command:

```
./BESAdmin.sh -resetdatabaseepoch
```

subsequent console logins reload their cache files.

resignsecuritydata

If you get one of the following errors:

```
class SignedDataVerificationFailure
HTTP Error 18: An unknown error occurred while transferring data
 from the server
```

when you try to log in to the BigFix console, you must resign all the user content in the database by entering the following command:

```
./BESAdmin.sh -resignSecurityData
```

This command resigns security data that uses the existing key file. You can also specify the following parameter:

```
-mastheadLocation=<path+actionsite.afxm>
```

The complete syntax to run this service is:

```
./BESAdmin.sh -resignsecuritydata -sitePvkLocation=<path+license
.pvk>
[ -sitePvkPassword=<password> ] [ -mastheadLocation=<path+action
site.afxm> ]
```

revokeexplorercredentials

Use this service to revoke the BigFix Explorer certificate associated to the given hostname. Use the following syntax to run the command:

```
./BESAdmin.sh -revokeexplorercredentials
-sitePvkLocation=<path+license.pvk>
-sitePvkPassword=<password>
-explorerHostname=<BigFixExplorerHostnameOrIP>
```

If an authentication certificate is issued for the specified `hostname`, this certificate is revoked and the BigFix Explorer instance running on that `explorerHostname` can no longer connect to the root server.

After revoking the credentials for a BigFix Explorer host, it will no longer connect to the root server. You can either remove the BES Explorer installation, or generate new credentials for that host, and replace the old certificate files on that host.

If your environment has multiple BigFix Servers, see this section (on page 464).

revoketokenkey

Use this service to revoke the secret key used to issue authentication tokens. After revoking it, all issued tokens will be invalid and operators will not be able to create new one.

```
./BESAdmin.sh -revoketokenkey -sitePvkLocation=<path+license.pvk
>
-sitePvkPassword=<password>
```

revokewebuicredentials

You can revoke the authentication certificate of a specified WebUI instance. The syntax to run this service is:

```
./BESAdmin.sh -revokewebuicredentials
-hostname=<host>
-sitePvkLocation=<path+license.pvk>
-sitePvkPassword=<pvk_password>
```

If an authentication certificate is issued for the specified `hostname`, this certificate is revoked and the WebUI instance running on that `hostname` can no longer connect to the root server.

After revoking the credentials for a WebUI host, it will no longer connect to the root server. You can either remove the WebUI installation, or generate new credentials for that host, and replace the old certificate files on that host.

If your environment has multiple BigFix Servers, see this section (on page 464).

rotateexplorercredentials

Use this service to rotate one BigFix Explorer certificate associated to the given hostname, or the whole BigFix Explorer CA. Use the following syntax to run the command:

```
./BESAdmin.sh -rotateexplorercredentials
-sitePvkLocation=<path+license.pvk>
-sitePvkPassword=<password>
-explorerCertDir=<path>
-explorerHostname=<BigFixExplorerHostnameOrIP>
| -rotateCA
```

You can either rotate one BigFix Explorer certificate, by hostname, if the certificate associated to that given hostname was compromised. The files of the new certificate will be copied in the <explorerCertDir> path.
Or you can rotate the BigFix Explorer Certificate Authority and all BigFix Explorer certificates. The files of the new certificates will be copied in a

new dedicated folder, in the <explorerCertDir> path, for each BigFix Explorer instance that had a not yet revoked certificate before the command was run. The command

```
./BESAdmin.sh -rotateexplorercredentials
-sitePvkLocation=<path+license.pvk>
-sitePvkPassword=<password>
-explorerCertDir=<path>
-rotateCA
```

will:

- Stop the Root Server
- Revoke each BigFix Explorer certificate created until that moment.
- Delete the ExplorerCACertificate and ExplorerCAKey files on the Root Server and in the Admin Fields database table.
- Start the Root Server that will recreate the files and entries of the previous point.
- Recreate all the certificates for the previous BigFix Explorer instances.

After the command was completed successfully, you must perform the following additional steps to allow Explorer to use the new credentials generated:

1. Stop the Explorer service.
2. Copy and replace the files apica.crt, auth_cert.crt, auth_key.key and ca.crt from the cert_hostname directory, created by the command in the explorerCertDir path, into the /var/opt/BESExplorer/cert directory. If the file root_ca.crt was also created, copy that file too.
3. Start the Explorer service.

If your environment has multiple BigFix Servers, see this section (on page 464).

rotaterepositorysiteregistrar

This command replaces the External Repository Site Registrar and records the corresponding revocation information. Run it with the standard BESAdmin deployment credentials when the registrar must be rotated in order to invalidate it. This command relates to a feature introduced in 11.0.7 named External Repository Sites.

```
./BESAdmin.sh -rotaterepositorysiteregistrar -sitePvkLocation=<p
ath+license.pvk>
[ -sitePvkPassword=<password> ]
```

rotateserversigningkey

You can rotate the server private key to have the key in the file system match the key in the database. The command creates a new server signing key, resigns all existing content that uses the new key, and revokes the old key.

The syntax to run this service is:

```
./BESAdmin.sh -rotateserversigningkey
-sitePvkLocation=<path+license.pvk>
[ -sitePvkPassword=<password> ]
```

If your environment has multiple BigFix Servers, see this section (on page 464).

rotatetokenkey

Use this service to rotate the secret key used to issue authentication tokens. The old key will be invalidated and replaced by a new one. The tokens created with the old key will no longer work.

```
./BESAdmin.sh -rotatetokenkey -sitePvkLocation=<path+license.pvk
>
-sitePvkPassword=<password>
```

rotatewebuicredentials

Use this service to rotate one BigFix WebUI certificate associated to the given hostname, or the whole BigFix WebUI CA. Use the following syntax to run the command:

```
./BESAdmin.sh -rotatewebuicredentials -sitePvkLocation=<path+lic
ense.pvk>
[ -sitePvkPassword=<password> ] -webUICertDir=<path>
{ -webUIHostname=<WebUIHostnameOrIP> | -rotateCA }
```

You can either rotate one BigFix WebUI certificate, by hostname, if the certificate associated to that given hostname was compromised. The files of the new certificate will be copied in the `webUICertDir` path.
Or you can rotate the BigFix WebUI Certificate Authority and all BigFix WebUI certificates. The files of the new certificates will be copied in a new dedicated folder, in the `webUICertDir` path, for each BigFix WebUI instance that had a not yet revoked certificate before the command was run. The command

```
./BESAdmin.sh -rotatewebuicredentials -sitePvkLocation=<path+lic
ense.pvk>
[ -sitePvkPassword=<password> ] -webUICertDir=<path> -rotateCA
```

will:

- Stop the Root Server
- Revoke each BigFix WebUI certificate created until that moment.
- Delete the WebUICACertificate, WebUICAKey, PlatCertificate, PlatKey, APIServerCertificate and APIServerKey files on the Root Server and in the Admin Fields database table.
- Start the Root Server that will recreate the files and entries of the previous point.
- Recreate all the certificates for the previous BigFix WebUI instances.

After the command was completed successfully, you must perform the following additional steps to allow the WebUI server to use the new credentials generated:

1. Stop the WebUI service.
2. Copy and replace the files auth_cert, auth_key and ca.crt from the cert_*hostname* directory, created by the command in the webUICertDir path, into the /var/opt/BESWebUI/cert directory. If the file root_ca.crt was also created, copy that file too.
3. Start the WebUI service.
4. If you have any active BigFix Explorer instance, stop it, copy the file ca.crt as apica.crt in the /var/opt/BESExplorer/cert directory. If the file root_ca.crt was also created, copy that file too. Restart the instance.

If your environment has multiple BigFix Servers, see this section (on page 464).

securitysettings

You can configure security options to follow the NIST security standards by running the command:

```
./BESAdmin.sh -securitysettings -sitePvkLocation=<path+license.p
vk>
[ -sitePvkPassword=<password> ]
{ -status | -requireSHA384Signatures [-requireSHA256Downloads]
| -allowSHA256Signatures | -requireSHA256Downloads | -allowSHA1D
ownloads}
[ -testTLSCipherList | -setTLSCipherList | -listTLSCiphers | -re
moveTLSCipherList ]
[ -hideFromFieldFromMasthead | -showFromFieldFromMasthead ]
[ -enableLocalOperators | -disableLocalOperators]
[ -requireTLS13 | -allowTLS12 ]
```

where:

status

Shows the status of the security settings set in your BigFix environment.

Example:

```
./BESAdmin.sh -securitysettings
 -sitePvkLocation=/root/backup/license.pvk
-sitePvkPassword=mypassw0rd -status

Enhanced security is currently ENABLED
SHA-256 downloads are currently OPTIONAL
```

requireSHA384Signatures | allowSHA256Signatures

Enables or disables the security option that adopts the SHA-384 cryptographic digest algorithm for all digital signatures.



Warning: If you use the **requireSHA384Signatures** setting you break the compatibility with BigFix version 10.0 or earlier version components.

For more information about the BigFix SHA-384 enforcement and the supported security configuration, see Security Configuration Scenarios.

requireSHA256Downloads

Ensures that the file download integrity check is run using the SHA-256 algorithm.

allowSHA1Downloads

Ensures that the file download integrity check is run using the SHA-1 algorithm.

testTLSCipherList | setTLSCipherList | listTLSCiphers | removeTLSCipherList

To test if a TLS cipher list is compatible with the BigFix components, run the following command:

```
/BESAdmin.sh -securitysettings
 -sitePvkLocation=<path+license.pvk>
 -sitePvkPassword=<password>
-
testTLSCipherList=<cipher_1>:<cipher_2>:..:<cipher_n
>
```

After identifying a suitable TLS cipher list, you can set it by running the following command:

```
/BESAdmin.sh -securitysettings
 -sitePvkLocation=<path+license.pvk>
 -sitePvkPassword=<password>
-
setTLSCipherList=<cipher_1>:<cipher_2>:..:<cipher_n>
```

To list all the TLS ciphers that are currently enabled, run the following command:

```
/BESAdmin.sh -securitysettings
 -sitePvkLocation=<path+license.pvk>
 -sitePvkPassword=<password>
-listTLSCiphers
```

To remove a TLS cipher list from the deployment masthead and return to the default cipher list, run the following command:

```
/BESAdmin.sh -securitysettings
 -sitePvkLocation=<path+license.pvk>
 -sitePvkPassword=<password>
-removeTLSCipherList
```

hideFromFieldFromMasthead | showFromFieldFromMasthead

You can specify if you want to show or hide the value displayed by the From field in the masthead which contains the email

address of the license assignee. During a fresh installation the value is hidden and the option "hideFromFieldFromMasthead" is set to 1. During an upgrade the value remains unchanged.
For example, if you want to hide the value, run the command as follows:

```
./BESAdmin.sh -securitysettings
 -sitePvkLocation=<path+license.pvk>
-sitePvkPassword=<password>
 -hideFromFieldFromMasthead
```

enableLocalOperators | disableLocalOperators

You can specify if you want to enable or disable the login to the BigFix environment (BigFix Console, Web Reports, Rest API and Web UI) of the local operators. The enabled/disabled choice will be stored in the BFEnterprise database. After disabling the login of the local operators, access will be granted only to LDAP users.
For example, if you want to disable the login of the local operators, run the command as follows:

```
./BESAdmin.sh -securitysettings
 -sitePvkLocation=<path+license.pvk>
-sitePvkPassword=<password> -disableLocalOperators
```

Note: The local operators are enabled by default.

Note: When trying to disable the local operators, if the "REST API credentials for BES Server Plugin Service" are set and if the configured user is a local operator, an error message is displayed and the option is not set.

Note: When trying to disable the local operators, if the "SOAP API credentials for BES Server Plugin Service" are set, a non-blocking warning message is displayed and the option is set.

requireTLS13

Requires the use of TLS 1.3 for HTTPS communications among BigFix components.

Warning: If you use the requireTLS13 setting you break the compatibility with BigFix version 10.0 or earlier version components.

Note: If you use this setting the Console SAML authentication works only on a Windows version that supports the TLS 1.3. For more details, see Assumptions and requirements.

allowTLS12

Allows the use of TLS 1.2 for HTTPS communications among BigFix components.

setadvancedoptions

You can list or configure any global settings that apply to your particular installation. The complete syntax to run this service is:

```
./BESAdmin.sh -setadvancedoptions -sitePvkLocation=<path+license
.pvk>
[-sitePvkPassword=<password>]
{ -list | -display
| [ -f ] -delete option_name
```

```
| [ -f ] -update option_name=option_value }
```

For example:

- To customize the Console or Web Report login banner, enter following command:

```
./BESAdmin.sh -setadvancedoptions -sitePvkLocation=/root/ba
ckup/license.pvk
-sitePvkPassword=pippo000 -update loginWarningBanner='new m
essage'
```

- If your BigFix Server is V9.5.7 or later, to avoid having duplicate computer entries when the endpoints are detected as possible clones by the Server, run the following command:

```
./BESAdmin.sh -setadvancedoptions -sitePvkLocation=/root/ba
ckup/license.pvk
-sitePvkPassword=pippo000 -update clientIdentityMatch=100
```

For a list of available options that you can set, see List of advanced options.

setClientCertificatePolicy

Starting from Patch 6, you can use this service to set the maximum validity period for newly issued client certificates and define how long before expiration a BigFix client should attempt the renewal.

The syntax to run this service is:

```
./BESAdmin.sh -setClientCertificatePolicy [-maxValidity <days>]
 [-startRefreshBefore <days>]
```

You need to specify the following parameters:

- `-maxValidity <days>` sets the maximum validity period for new BigFix client certificates.

  Value: An integer representing the number of days (e.g., 395 for approximately 13 months, 730 for 2 years). Alternatively, use the word **default** to revert to the system default (e.g., 13 months for clients 10.0.7 and later).

  Effect: This setting applies to certificates issued after the policy is applied and propagated. Existing certificates will be recreated in accordance with the new policy settings upon renewal.
- `-startRefreshBefore <days>` specifies how long before the certificate expiration date the BigFix client should attempt to initiate a renewal.

  Value: An integer representing the number of days (e.g., 60 for 60 days). Alternatively, use the word **default** to revert to the system default (e.g., 45 days).

  Note: The value for startRefreshBefore must be less than the maxValidity value.

  Recommendation: Use a value that provides ample opportunity for intermittently offline BigFix clients to renew successfully.

Policy Propagation

The `maxValidity` and `startRefreshBefore` settings are embedded in the **ActionSite.afxm** file using the following headers:

- **x-bes-client-certificate-max-validity**
- **x-bes-client-certificate-refresh-before**

Changes to these settings trigger the generation of a new version of the masthead. Existing BigFix clients receive these updated settings during regular masthead checks. The BigFix Server uses these settings to correctly handle client certificate requests.

setcustomca

Starting from Patch 4, you can use this service to install a custom CA intermediate certificate in the BigFix Platform.

If you use a CA intermediate certificate along with the CA root certificate, the syntax to run this service is:

```
./BESAdmin.sh -setcustomca -customCARootCrtLocation=<path>
-customCACrtLocation=<path> -customCAPvkLocation=<path> [-custom
CAPvkPassword=<password>]
```

You need to specify the following parameters:

- `-customCARootCrtLocation` specifies the path where the CA root certificate is located.
- `-customCACrtLocation` specifies the path where the custom CA intermediate certificate is located.
- `-customCAPvkLocation` specifies the path where the private key of the custom CA intermediate certificate is located.
- `-customCAPvkPassword` (Optional) specifies the password associated with the private key of the custom CA intermediate certificate.

Otherwise, if you do not use a CA intermediate certificate but the CA root certificate only, the syntax to run the service is:

```
./BESAdmin.sh -setcustomca -customCACrtLocation=<path>
 -customCAPvkLocation=<path> [-customCAPvkPassword=<password>]
```

where you need to specify the following parameters:

- `-customCACrtLocation` specifies the path where the CA root certificate is located.
- `-customCAPvkLocation` specifies the path where the private key of the CA root certificate is located.
- `-customCAPvkPassword` (Optional) specifies the password associated with the private key of the CA root certificate.

For this service, it is also available a UI that you can launch by running:

```
./BESAdmin.sh -setcustomca
```

without additional options.

Note: The CA root and intermediate certificates and the related keys are intended as server certificates/keys, they are not end-user certificates and keys.

Later on, you can uninstall the CA intermediate certificate and restore the standard functionality of the BigFix Platform by running this service:

```
./BESAdmin.sh -removecustomca
```

If your environment has multiple BigFix Servers, see this section (on page 464).

setproxy

If your enterprise uses a proxy to access the Internet, you must set a proxy connection to enable the BigFix server to gather content from sites and to do component-to-component communication or to download files.

For more information about how to run the command and about the values to use for each argument, see Setting a proxy connection on the server (on page 552).

syncmastheadandlicense

When you upgrade the product, you must use this option to synchronize the update license with the masthead and resign all content in the database with SHA-256. The syntax to run this service is:

```
./BESAdmin.sh -syncmastheadandlicense -sitePvkLocation=<path+lic
ense.pvk>
[-sitePvkPassword=<password>]
```

testproxyconnection

You can test the proxy connection. The syntax to run this service is:

```
BESAdmin.sh -testproxyconnection -proxyHost=<host> [ -proxyPort=
<port> ]
[ -proxyUser=<user> -proxyPassword=<pass> ] [ -proxyExcList=<lis
t> ] [ -proxyAuthMeth=<method> ]
[ -proxySecTunnel=<true|false> ] [ -fips ]
```

updatepassword

You can modify the password that is used for authentication by product components in specific configurations.

The syntax to run this service is:

```
./BESAdmin.sh -updatepassword -type=<server_db|dsa_db>
[-password=<password>] -sitePvkLocation=<path+license.pvk>
[-sitePvkPassword=<pvk_password>]
```

where:

-type=server_db

Specify this value to update the password that is used by the server to authenticate with the database.

If you modify this value, the command restarts all the BigFix server services.

-type=dsa_db

Specify this value to update the password that is used in a DSA configuration by a server to authenticate with the database.

The settings `-password` and `-sitePvkPassword` are optional, if they are not specified in the command syntax their value is requested interactively at run time. The password set by this command is obfuscated.

Special procedures for environments with multiple BigFix Server

Environments with multiple BigFix Server (DSA) require special procedures to successfully achieve certain configuration changes, maintaining consistency and coherence.

In such environments, the following BESAdmin commands may require additional steps or multiple applications:

- `rotateexplorercredentials` and `revokeexplorercredentials`
- `rotateserversigningkey`
- `rotatewebuicredentials` and `revokewebuicredentials`
- `setcustomca` and `removecustomca`

Special procedures for DSA environments

If your BigFix environment is configured as a **DSA**, you may need to run the aforementioned BESAdmin commands on each BigFix Server to obtain the desired effect, not just on the primary.

For rotating the Server signing keys in a DSA environment, see this wiki article.

Similarly, when setting or removing a custom CA in a DSA environment, you will need to run the corresponding BESAdmin command on each BigFix Server computer. This operation is best performed with all BigFix Servers stopped and Consoles closed.

Working with TLS cipher lists

All network communications between the BigFix components and the internet are encrypted by using the TLS protocol standard. Starting from Version 9.5.11, master operators can control which TLS ciphers should be used for encryption. A master operator can set a deployment-wide TLS cipher list in the masthead by using BESAdmin.

The TLS cipher list is a colon-delimited list of cipher suites or cipher families. To disable a cipher suite or cipher family, precede the name with "!".

The default TLS cipher list which is `HIGH:!ADH:!AECDH:!kDH:!kECDH:!PSK:!SRP` is used when no TLS cipher list is present in the masthead.

Starting from Version 10 Patch 3 and later, the default TLS cipher list, used when no TLS cipher list is present in the masthead, is `HIGH:!ADH:!AECDH:!kDH:!kECDH:!kRSA:!PSK:!SRP`.

This defines the master set of TLS cipher suites from which you can select. Cipher suites that are not in this master set are either insecure or incompatible with the BigFix components. In addition, the TLS cipher list must include at least one cipher suite using RSA key exchange for the BigFix HTTPS servers. The following BESAdmin commands help you create the TLS cipher list:

testTLSCipherList

To test if a particular TLS cipher list is compatible with the BigFix components, run the following command:

```
/BESAdmin.sh -securitysettings -sitePvkLocation=<path+license.pvk> -sitePvkPassword=<password>
-testTLSCipherList=<cipher_1>:<cipher_2>:..:<cipher_n>
```

For example:

```
/BESAdmin.sh -securitysettings -sitePvkLocation=/opt/BESInstallFiles/license.pvk -sitePvkPassword=bigfix -testTLSCipherList='TLSv1.2:!ADH:!AECDH:!kDH:!kECDH:!PSK:!SRP:!NULL'
```

If the command runs successfully, BESAdmin provides a detailed list of all TLS cipher suites that are enabled. If unsuccessful, BESAdmin provides a detailed list of which cipher suites are insecure or incompatible.

Note: In bash, *"!"* is treated as a special character. You must either escape it with "\" or enclose it within single quotes *"'"*. Double quotes *""* are insufficient.

setTLSCipherList

After identifying a suitable TLS cipher list, you can set it with the following command:

```
/BESAdmin.sh -securitysettings -sitePvkLocation=<path+license.pvk> -sitePvkPassword=<password>
-setTLSCipherList=<cipher_1>:<cipher_2>:..:<cipher_n>
```

For example:

```
/BESAdmin.sh -securitysettings -sitePvkLocation=/opt/BESInstallFiles/license.pvk -sitePvkPassword=bigfix -setTLSCipherList='TLSv1.2:!ADH:!AECDH:!kDH:!kECDH:!PSK:!SRP:!NULL'
```

If the command is unsuccessful, BESAdmin provides a detailed list of which cipher suites are insecure or incompatible. The ciphers on the list are arranged in an order of preference. To modify the order by key length, add @STRENGTH.

Note: BESAdmin does not verify if the name of a particular cipher suite or cipher family is available; it only checks the final set of TLS cipher suites that is implied by the colon delimited list.

listTLSCiphers

For a detailed list of all the TLS ciphers that are currently enabled, run the following command:

```
/BESAdmin.sh -securitysettings -sitePvkLocation=<path+license.pvk> -sitePvkPassword=<password> -listTLSCiphers
```

For example:

```
/BESAdmin.sh -securitysettings -sitePvkLocation=/opt/BESInstallFiles/license.pvk -sitePvkPassword=bigfix -listTLSCiphers
```

removeTLSCipherList

To remove a TLS cipher list from the deployment masthead and return to the default cipher list, run the following command:

```
/BESAdmin.sh -securitysettings -sitePvkLocation=<path+license.pvk> -sitePvkPassword=<password> -removeTLSCipherList
```

For example:

```
/BESAdmin.sh -securitysettings -sitePvkLocation=/opt/BESInstallFiles/license.pvk -sitePvkPassword=bigfix -removeTLSCipherList
```

The detailed ciphers that are available for a given cipher family depends on the version of OpenSSL that is in use. At its core, the TLS cipher list is the OpenSSL cipher string. For more details, see OpenSSL Cryptography and SSL/TLS Toolkit. Do not use this feature if you are not familiar with the basics of TLS cryptography.

Logging Cleanup Tasks Activities

You can run the cleanup tasks, on the BigFix Server.

On Windows systems:

From the Clean Up tab of the BigFix Administration Tool, or from the command line using the `BESAdmin.exe` program as described in BESAdmin Windows Command Line *(on page 376)*.

On Linux systems:

From the command line using the `BESAdmin.sh` bash shell script as described in BESAdmin Linux Command Line *(on page 422)*.

By default the information about the processing of the cleanup tasks is logged in the following files:

- `/var/log/BESTools.log` on Linux systems.
- `C:\Program Files (x86)\BigFix Enterprise\BES Server\BESTools.log` on Windows systems.

and the maximum size of the log file is 5 MB. When the size exceeds that value the log file rotates.

Note: The `BESTools.log` file was introduced with BigFix Version 9.5 Patch 5.

You can use the configuration settings `_BESTools_Logging_LogPath` and `_BESTools_Logging_LogMaxSize` to specify a different path and a different maximum size for logging the cleanup tasks activities.

The same log file is used both when you run the tools using BESAdmin and when you schedule them in the RootServer.

Chapter 12. Post-installation configuration steps

After having run the installation, make sure that you read the following topics and run the requested activities if needed.

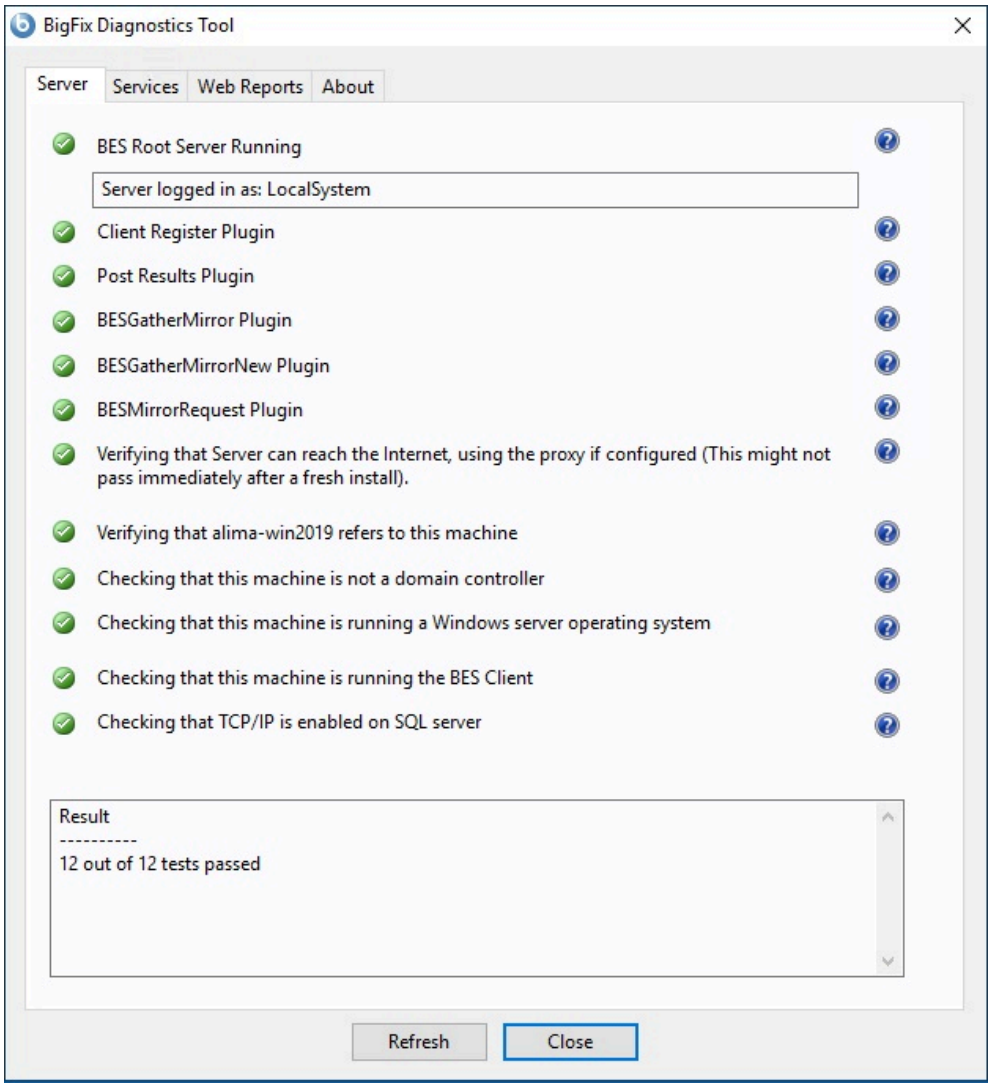
Post-installation steps

After you install the product, perform these steps to verify that the installation runs successfully and to complete the basic configuration steps.

1. Run the following step to verify that the installation runs successfully:

   On Windows:

   From **Start > All Programs > BigFix** run the BigFix Server Diagnostics tool to verify that all the installation and configuration steps completed successfully.



If all the buttons are green, click **Close** to exit the Diagnostic tool, otherwise address the problem to be sure that the server is working correctly.

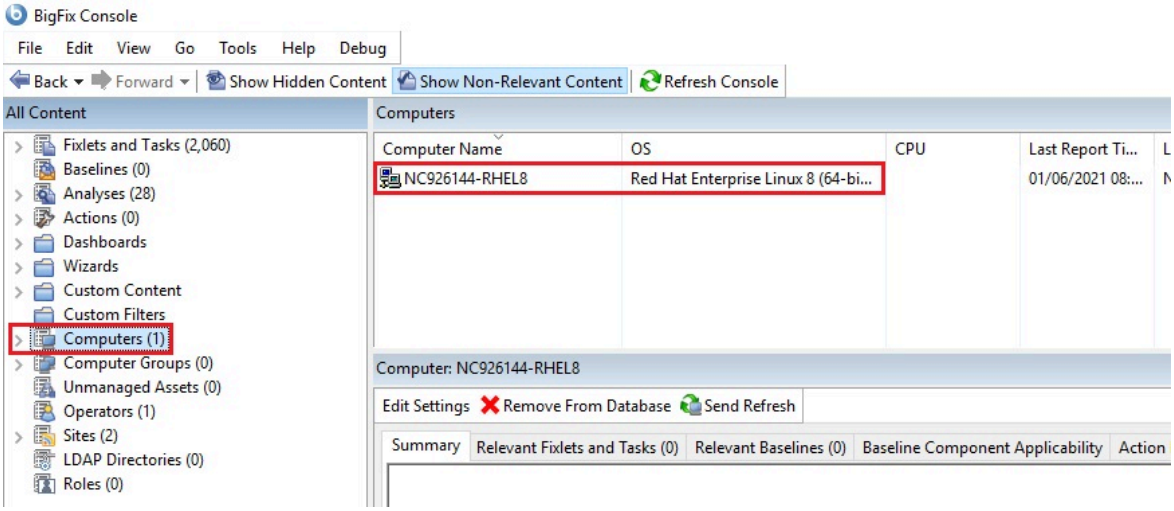
On Linux:

Ensure that the following services are up and running:

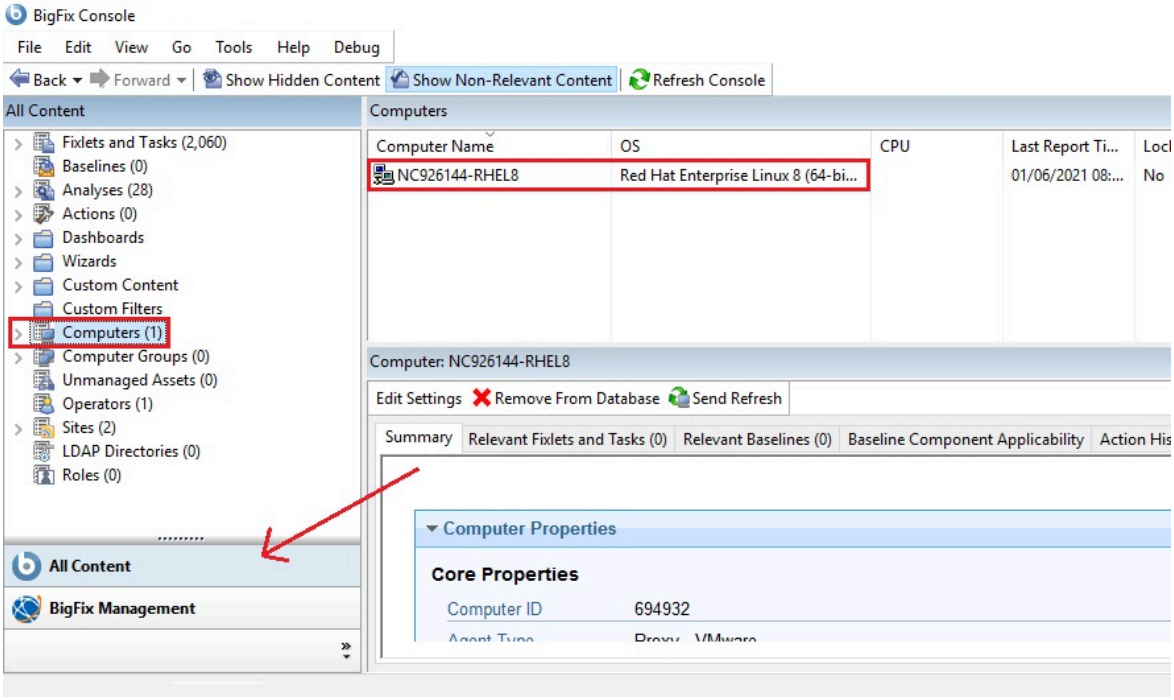
```
besfilldb
besgatherdb
besserver
beswebreports
```

Use the command `service service status` to check the status of the services.

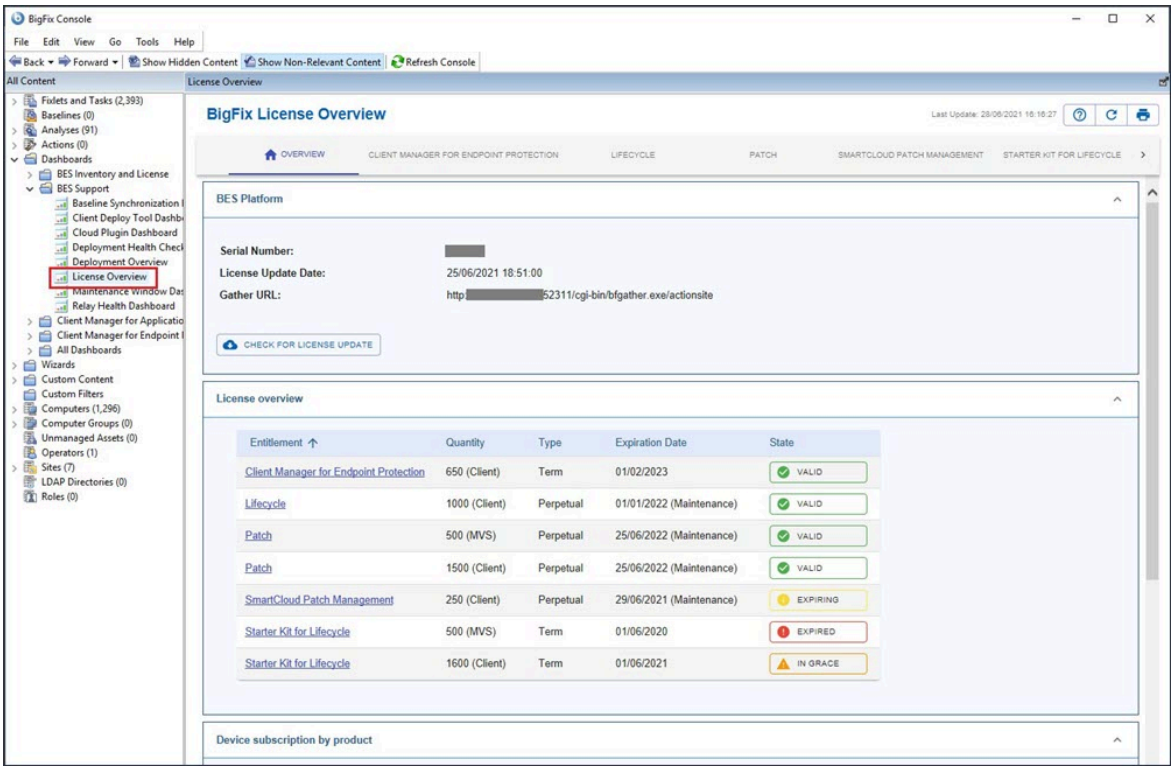
2. Open the BigFix console and verify that the client is registered.



3. From the console, verify that the **All Content** and **BigFix Management** domains have been created.



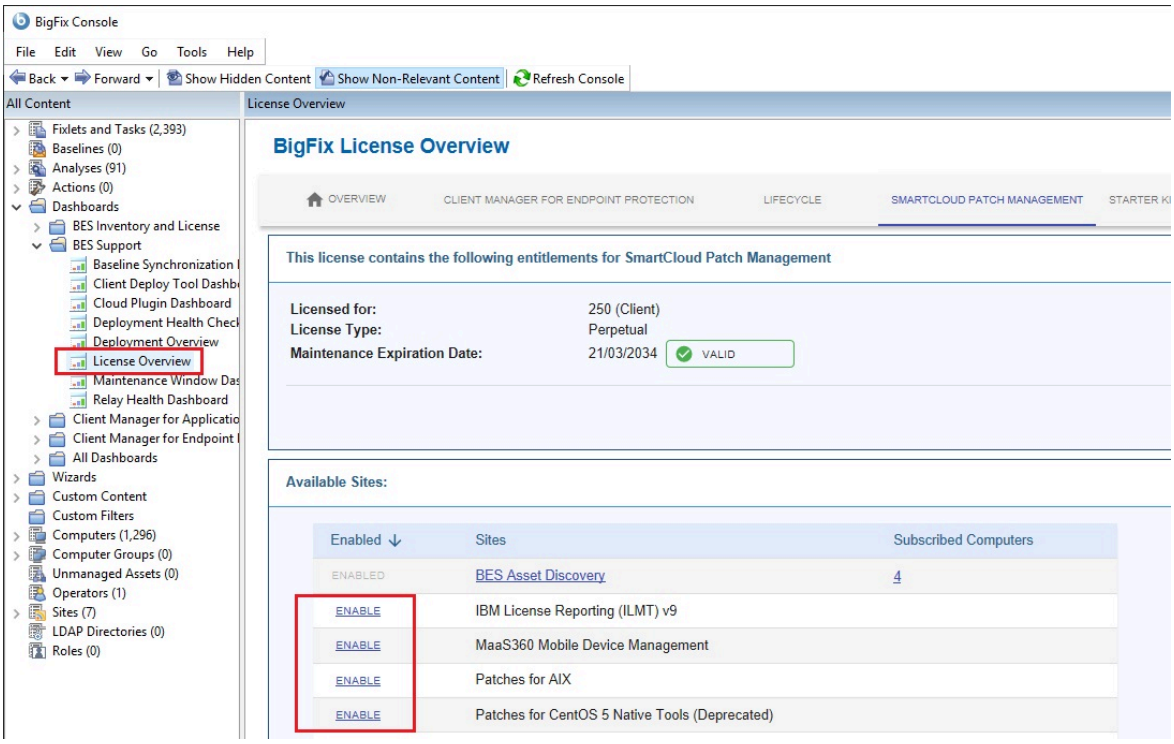
4. After installation, the program is automatically set up to subscribe to certain management and maintenance sites. Depending on the terms of your license, you might have subscriptions to other sites as well. In this way content from those Sites automatically flows into your enterprise and is evaluated for relevance on all computers running the BigFix client. Subscribe to these sites from the **BigFix Management** domain, by selecting the **License Overview** dashboard.



   The License Overview dialog appears, listing available sites.

   Select the desired product by clicking the corresponding tab or the name in the **License overview** table.

5. Enable the entitled sites by clicking the **Enable** button associated with the site to which you want to subscribe.



6. Enter your password to subscribe to the site. The new site is now listed in the **Manage Sites** node of the domain panel. You can also subscribe to a site by using a masthead file.
7. Open the **Manage Sites** node and select your newly subscribed site.
8. From the site dialog, click the **Computer Subscriptions** tab to assign the site to the appropriate computers
9. From the **Operator Permissions** tab, select the operators you want to associate with this site and their level of permission.
10. Click Save Changes when you are done.

You can now use the product.

Starting and stopping the BigFix server

Complete the following steps to start and stop the BigFix server installed on a Windows system:

Steps to start BigFix:

Start the following Windows services in the specified order:

```
BES Root Service
BES FillDB
BES GatherDB
BES Client
BES Web Reports Service
```

Steps to stop BigFix:

Stop the following Windows services in the specified order:

```
BES Web Reports Service
BES Client
BES GatherDB
BES FillDB
BES Root Service
```

Complete the following steps to start and stop the BigFix server installed on a Linux system:

Steps to start BigFix:

Run the following services in the specified order:

```
service besserver start
service besfilldb start
service besgatherdb start
service beswebreports start
service besclient start
```

Steps to stop BigFix:

Run the following services in the specified order:

```
service besclient stop
service beswebreports stop
service besgatherdb stop
service besfilldb stop
```

```
service besserver stop
```

Note: For more details on BigFix Service Management on Linux platforms, see Managing the BigFix Services *(on page 273)*.

Subscribing to content sites

Sites are collections of Fixlets, tasks, analysis, that are created internally by you, by HCL, or by vendors. You subscribe to a site and agree on a schedule for downloading the latest batch.

You can add a new site subscription by acquiring a masthead file from a vendor or from HCL. You can subscribe to a site also by using the Licensing Dashboard.

Sites are generally devoted to a single topic, such as security or the maintenance of a particular piece of software or hardware. However, several sites can share characteristics and are then grouped into domains, which might include a set of typical job tasks of your various Console managers. For example, the person responsible for patching and maintaining a common operating environment can find Support sites and Patching sites for various operating systems all bundled into the Patch Management Domain.

You can also set up your own custom site and populate it with Fixlets that you have developed specifically for your own network. You and other operators can then send and receive the latest in-house patches and quickly deploy them to the appropriate locations and departments.

Upon installation, the program is automatically set up to subscribe to certain management and maintenance sites. Depending on the terms of your license, you might have subscriptions to other sites as well. This means that content from those sites automatically flows into your enterprise and is evaluated for relevance on all computers running the BigFix client. These sites, in turn automatically register with an appropriate domain, providing a simple way to divide the content into functional sections.

Subscribing with a masthead

To subscribe to a site using a masthead file, follow these steps:

1. Find an appropriate site. Finding a site is equivalent to finding a site masthead file, which has an extension of `.efxm`. There are several ways to do this:

   Fixlet sites:

   HCL might post a links list to new sites as they become available.

   Fixlet subscriptions:

   Sometimes a Fixlet message might offer a subscription. Click the Fixlet action to start the subscription.

   Download mastheads:

   You can also subscribe to a site by downloading a masthead file from a vendor's website. After the masthead is saved to your computer, you can activate it in one of the following ways:
   - Double-click the masthead, or
   - Select **Add External Site Masthead** from the **Tools** menu, browse the folder containing the masthead, and click **Open**.
2. You are prompted for your private key password. Type it in and click **OK**.

The masthead is propagated to all Clients, which immediately begin to evaluate the Fixlet from the new site.

Subscribing with the Licensing Dashboard

You can subscribe to a Fixlet site also by using the Licensing Dashboard in BigFix Management, found in the Domain Panel:

1. Open the **BigFix Management** domain and scroll to the top to view the associated dashboards.
2. From the **Licensing Dashboard**, select the sites you want to subscribe to.

Changing the database password

How to change the database password.

After you install the database of the BigFix server, you can change its password by running the following command:

- On Windows operating systems:

```
.\BESAdmin.exe /updatepassword /type=<server_db|dsa_db>
[/password=<password>] /sitePvkLocation=<path+license.pvk>
[/sitePvkPassword=<pvk_password>]
```

- On UNIX operating systems:

```
./BESAdmin.sh -updatepassword -type=<server_db|dsa_db>
[-password=<password>] -sitePvkLocation=<path+license.pvk>
[-sitePvkPassword=<pvk_password>]
```



Note: This procedure on UNIX also updates the database password for Web Reports, if the Web Reports component is installed on the same system where the BigFix server is installed.

where:

type=server_db

If you changed the database instance password, specify this value to update the password used by the server to authenticate with the database.

If you modify this value, the command restarts all the BigFix server services.

type=dsa_db

If you changed the database instance password on a server of a DSA configuration, specify this value to update the password used in a DSA configuration by remote servers to authenticate with the database.

For example:

```
./BESAdmin.sh -updatepassword -sitePvkLocation=/mylicenses/license.pvk
-sitePvkPassword=******* -type=server_db
```

The settings `-password` and `-sitePvkPassword` are optional; if they are not specified in the command syntax their value is requested interactively at runtime. The password set by this command is obfuscated.

Changing the database password on UNIX on Web Reports

On UNIX operating systems, to change the database password on the local and remote Web Reports server, complete the following steps:

1. Stop the `beswebreports` service:

   ```
   service beswebreports stop
   ```

2. Open the configuration file: `/var/opt/BESWebReportsServer/beswebreports.config`
3. Go to `[Software\BigFix\Enterprise Server\FillAggregateDB]` and set:

   ```
   Password = "newpassword"
   ```

4. Start the `beswebreports` service:

   ```
   #service beswebreports start
   ```

   After restart, passwords are obfuscated and substituted again with `""` in the configuration files.

Chapter 13. Managing relays

Relays can significantly improve the performance of your installation.

Relays lighten both upstream and downstream burdens on the server. Rather than communicating directly with a server, clients can instead be instructed to communicate with designated relays, considerably reducing both server load and client and server network traffic. Relays improve performance by:

- **Relieving downstream traffic**. Using relays, the BigFix server does not need to distribute files, such as patches or software packages, and Fixlets to every Client. Instead, the file is sent once to the relay, which in turn distributes it to the clients.
- **Reducing upstream traffic**. In the upstream direction, relays can compress and package data (including Fixlet relevance, action status, and retrieved properties) from the clients for even greater efficiency.
- **Reducing congestion on low-bandwidth connections**. If you have a server communicating with computers in a remote office over a slow connection, designate one of those computers as a relay. Then, the server sends only a single copy to the relay (if it needs it). That relay, in turn, distributes the file to the other computers in the remote office over its own fast LAN.

Establishing the appropriate relay structure is one of the most important aspects of deploying BigFix to a large network. When relays are fully deployed, an action with a large download can be quickly and easily sent out to tens of thousands of computers with minimal WAN usage.

A recommended configuration is the connection of 500 - 1000 clients to each relay and the use of a parent child relay configuration.



Note: If the connection between a relay and server is unusually slow, it might be beneficial to connect the relay directly to the Internet for downloads.

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). To prevent any security vulnerabilities, configure the internet-facing relays in your deployment as authenticating. For details, see Setting up internet relays (on page 494).

Relay requirements and recommendations

Generally, a relay uses minimal resources and does not have a noticeable impact on the performance of the computer running it.

However, if several clients simultaneously request files from a relay, a significant amount of the computer's resources might be used to serve those files.

The requirements for a relay computer vary widely depending on three main factors:

- The number of connected clients that are downloading files.
- The size of each download.
- The period of time allotted for the downloads.

For details about the relay system requirements, see BigFix 11 - Detailed System Requirements.

Note: On Linux computers in which the deployment port is not the default (52311), an installation of Perl is required for the relay functions to work as expected.

Here are some further recommendations:

- Computers running the relays must have BigFix agent installed.
- Configure the internet-facing relays in your BigFix deployment as authenticating relays.
- Workgroup file servers and other server-quality computers that are always turned on are good candidates for installing a relay.
- The BigFix relay must have a two-way TCP connection to its parent (which can be a server or another relay).

- For BigFix Version 10 GA, the Fixlet to install the relay requires at least Internet Explorer 5.0 or later versions to work correctly. For BigFix Version 10 Patch 1 and later, this requirement is no longer needed.
- The BigFix relay cache size can be configured, but is set to 1GB by default. It is recommended that you have at least 2 GB available for the relay cache to prevent hard drive bottlenecks.
- It is recommended to have at least one Relay per geographic location for bandwidth reasons.
- Consider throttling the bandwidth usage for Relays downloading files on very slow pipes. It is recommended to throttle the bandwidth usage for Clients that are connecting on dial-up or slow VPN connections.

Setting up a relay

To set up a relay, you must designate a Windows, Red Hat Enterprise Linux, or Solaris computer that is running a client to act as the relay.

The BigFix clients on your network detect the new relays and automatically connect to them. To configure a client computer as a relay, run the following steps:

1. Log in to the BigFix console.
2. Open the **Fixlets and Tasks** icon in the Domain Panel and click **Tasks Only**.
3. Double-click the task labeled **Install BigFix relay** (it might include a version number after it). This task is relevant when there is at least one client that meets the requirements for the relay.
4. Choose your deployment option by selecting one of the actions in the task. You can target single or multiple computers with this action.

After the relays have been created, Clients can be made to automatically discover and connect to them, always seeking the Relay that is the fewest hops away.

Installing and upgrading a relay from the command line

You can install or upgrade a relay from the command line using the `setup.exe` installer command.

The same installer command is issued by the Install or Upgrade Relay Fixlet on the target relevant clients.

For information about the `setup.exe` syntax and the available switches, see Command-Line Options.

This is the list of additional options that you can use when using the `setup.exe` installer:

RESTARTBESCLIENT

Set it to 0 to prevent the BES Client service from restarting while installing or upgrading the relay. For example, if you want to install the relay in an unattended mode but you want to prevent the client from starting while processing you can run the following command:

```
setup.exe /s /v"RESTARTBESCLIENT=0 /qn"
```

STARTRELAYSERVICE

Set this option to 0 to prevent the BES Relay service from starting while installing or upgrading the relay. For example, if you want to install the relay in an unattended mode but you want to prevent it from starting you can run the following command:

```
setup.exe /s /v"STARTRELAYSERVICE=0 /qb"
```

Note: This option is available starting from BigFix version 9.5 Patch 3.

REBOOT

Set this option to `ReallySuppress` if you want to prevent the relay machine from rebooting. For example, if you want to install the relay in an unattended

mode but you want to prevent the system from rebooting you can run the following command:

```
setup.exe /s /v"REBOOT=ReallySuppress /qn"
```

Assigning relays to clients

When you have set up a relay you must direct BigFix clients on your network to gather from that relay, instead of from your server.

You can:

- Assign relays manually as it is described in the following topics:
  - Assigning relay at client installation time *(on page 483)*
  - Manually assigning relays to existing clients *(on page 488)*
- Assign relays automatically, that means to allow clients to identify the closest relay to connect to, as it is described in the following topics:
  - Automatically assigning relays at client installation time *(on page 489)*
  - Automatically assigning relays to existing clients *(on page 489)*

  If you select this method, you can also choose to exploit the relay affiliation functionality. Using this functionality you create groups of affiliated clients and you assign relays to the affiliation group. For more information about this functionality and how to use it, see Using relay affiliation *(on page 490)*.

  For more information and considerations about automatic relay assignment, see Notes about automatic relay assignment (on page 492).

Assigning relay at client installation time

By default, the BigFix clients are configured to connect to the main BigFix server at installation time.

If you want you can configure the BigFix client to assign a specific BigFix relay at the time of client installation. Depending on the client operating system, you must perform different steps as described in the following topics:

- Windows Clients *(on page 484)*
- UNIX Clients *(on page 486)*
- Mac Clients *(on page 484)*

Windows Clients

To set a relay, create a file named `clientsettings.cfg` in the BigFix Client installation folder (setup.exe) with one of the following contents:

- The file can contain one line like the following:

  ```
  IP=http://relay.domain.com:52311/bfmirror/downloads/
  ```

  which is automatically expanded out to define concurrently the following two settings on the Client:

  ```
  __RelaySelect_Automatic=0
  __RelayServer1=http://relay.domain.com:52311/bfmirror/downloads/
  ```

- You can directly define the two settings, optionally with other additional settings, such as a secondary relay, for example:

  ```
  __RelaySelect_Automatic=0
  __RelayServer1=http://relay.domain.com:52311/bfmirror/downloads/
  __RelayServer2=http://relay2.domain.com:52311/bfmirror/downloads/
  ```

Mac Clients

You can optionally use the `clientsettings.cfg` file to create custom settings on a Mac Client, for example to assign the new Client to a specific parent relay.

The `clientsettings.cfg` file must be available in the same directory as the PKG file and the `actionsite.afxm` file.

To set a relay, create a file named `clientsettings.cfg` in the BigFix Client installation folder with one of the following contents:

- The file can contain one line like the following:

  ```
  IP=http://relay.domain.com:52311/bfmirror/downloads/
  ```

  which is automatically expanded out to define concurrently the following two settings on the Client:

  ```
  __RelaySelect_Automatic=0
  __RelayServer1=http://relay.domain.com:52311/bfmirror/downloads/
  ```

- You can directly define the two settings, optionally with other additional settings, such as a secondary relay, for example:

  ```
  __RelaySelect_Automatic=0
  __RelayServer1=http://relay.domain.com:52311/bfmirror/downloads/
  __RelayServer2=http://relay2.domain.com:52311/bfmirror/downloads/
  ```

The agent installer is launched via the Terminal program as a privileged user by running the following command:

```
{sudo} /
Library/BESAgent/BESAgent.app/Contents/MacOS/BESAgentControlPanel.sh
```

The `sudo` command is not strictly needed but, authenticating as a Super User, you can perform the installation with no problems. This script has a few options that are listed if you run it without options.

Note:

- The QnA executable is also included in the client installation package. On Macintosh clients, to use it you must launch the Terminal program and run:

  ```
  {sudo} /Library/BESAgent/BESAgent.app/Contents/MacOS/QnA
  ```

  The `sudo` command is optional but some inspectors run only if you are Super User (root).

- The agent uninstaller is available in the .pkg install. It is located in: `/Library/BESAgent/BESAgent.app/Contents/MacOS/BESAgentUninstaller.sh`
- The agent .dmg package is no longer available.
- If you want to use the Client Compliance API on the Mac OSX system, you must request the client compliance library to the HCL Support team.

UNIX Clients

To assign a relay to your UNIX client at installation time, perform the following steps:

1. Create the `besclient.config` file under `/var/opt/BESClient/` with the following lines:

```
[Software\BigFix\EnterpriseClient]
EnterpriseClientFolder = /opt/BESClient

[Software\BigFix\EnterpriseClient\GlobalOptions]
StoragePath = /var/opt/BESClient
LibPath = /opt/BESClient/BESLib

[Software\BigFix\EnterpriseClient\Settings\Client\__RelayServer1]
effective date = [Enter current date and time in standard format]
value = http://relay.domain.com:52311/bfmirror/downloads/

[Software\BigFix\EnterpriseClient\Settings\Client\__RelayServer2]
effective date = [Enter current date time in standard format]
value = http://relay2.domain.com:52311/bfmirror/downloads/

[Software\BigFix\EnterpriseClient\Settings\Client\__RelaySelect_Automa
tic]
effective date = [Enter current date time in standard format]
value = 0
```

2. Ensure that the directory and file are owned by root and are not writable by anyone else. In this way, when you run the UNIX client installer to install the client, the installer does not re-create or overwrite `/var/opt/BESClient/besclient.config` with the following settings:

```
[Software\BigFix\EnterpriseClient]
EnterpriseClientFolder = /opt/BESClient

[Software\BigFix\EnterpriseClient\GlobalOptions]
StoragePath = /var/opt/BESClient
LibPath = /opt/BESClient/BESLib
```

3. In `effective date = [Enter current date and time in standard format]` set the date and time. An example of the standard format of the date and time is the following:

```
Wed, 06 Jun 2012 11:00:00 -0700
```

   You cannot specify `effective date = {now}` because the `{}` brackets imply the use of inline relevance, and **now** is a keyword.

4. In `value = http://relay.domain.com:52311/bfmirror/downloads/` modify `relay.domain.com` to be your desired relay.



Tip: You can obtain and verify the current content of the `besclient.config` by assigning a relay manually for a particular Linux client, and then copying the particular lines from its `besclient.config` file to use on other systems.

Adding More Settings

To add other client settings during the installation of the new client, include a line for each client setting to be set during client installation, for example, the file might look similar to:

```
__RelayServer1=http://relay.domain.com:52311/bfmirror/downloads/
_BESClient_Inspector_ActiveDirectory_Refresh_Seconds=43200
```

```
_BESClient_Log_Days=10
...
```

Note: On UNIX clients, the besclient.config file must contain the following sections before any other client settings:

```
[Software\BigFix\EnterpriseClient]
 EnterpriseClientFolder = /opt/BESClient
```

```
[Software\BigFix\EnterpriseClient\GlobalOptions]
 StoragePath = /var/opt/BESClient
 LibPath = /opt/BESClient/BESLib
```

For more information about the client settings you can set, see Deploy the agent so that it starts with specific settings.

Manually assigning relays to existing clients

You might want to manually specify exactly which clients must connect to which relay.

You can do this by performing the following steps:

1. Start the Console and select the **BigFix Management** Domain. From the Computer Management folder, click **Computers** to see a list of clients in the list panel.
2. Select the set of computers you want to attach to a particular Relay.
3. Right-click this highlighted set and choose **Edit Computer Settings** from the pop-up menu. As with creating the relays (above), the dialog boxes are slightly different if you selected one or multiple computers.
4. Check the box labeled **Primary Relay** and then select a computer name from the drop-down list of available Relay servers.
5. Similarly, you can assign a **Secondary Relay**, which will be the backup whenever the Primary Relay Server is unavailable for any reason.
6. Click **OK** .

Automatically assigning relays at client installation time

As you install clients, you might want them to automatically discover the closest relay by default.

Set this up by completing the following steps:

1. Open the **Edit Computer Settings** dialog by right-clicking any computer in the computers list on the BigFix console.
2. Click the button labeled **More Options**.
3. In the **Settings** tab check the following settings:
    - Relay Selection Method
    - Automatically Locate Best Relay
4. Select the **Target** tab.
5. Click the button labeled **All computers with the property**.
6. In the window below, select **All Computers**.
7. Select the **Constraints** tab.
8. Clear the **Expires On** box.
9. Click **OK**.

As new clients are installed, they now automatically find and connect to the closest relay without any further action.

Automatically assigning relays to existing clients

You can configure clients to automatically find the closest relay and point to that computer instead of the server.

This is the recommended technique, because it dynamically balances your system with minimal administrative overhead. Clients can determine which relays are the fewest number of hops away, so your topology is optimized.

This behavior is key when your network configuration is constantly shifting as laptops dock and undock, as computers start up or shut down, or as new hardware is added or removed. Clients can dynamically assess the configuration to maintain the most efficient connections as your network changes.

To make sure that your clients are set up to automatically discover relays run the following steps:

1. Start up the Console and select the **BigFix Management** Domain. From the Computer Management folder, click the **Computers** node to see a list of Clients in the list panel.
2. Shift- and ctrl-click to select the set of computers you want to automatically detect relays. Press **Ctrl-A** to select the entire set of clients.
3. Right-click this highlighted set and choose **Edit Computer Settings** from the pop-up menu. Depending on whether you selected one or more computers, the dialog boxes are slightly different. Typically, you select all the Clients in your network, so you will see the multiple-select dialog.
4. Check **Relay Selection Method**.
5. Click **Automatically Locate Best Relay**.
6. Click **OK**.

Using relay affiliation

Relay affiliation provides a more sophisticated control system for automatic relay selection.

The feature is very flexible and can be used in many different ways, but the primary use case is to allow the BigFix infrastructure to be segmented into separate logical groups.

A set of clients and relays can be put into the same affiliation group such that the clients only attempt to select the relays in their affiliation group. This feature is built on top of automatic relay selection and you should understand that process (see the previous section) before implementing relay affiliation.

Relay affiliation applies only to the automatic relay selection process. The manual relay selection process (see next section) is unaffected even if computers are put into relay affiliation groups.

Choosing relay affiliation group names

There are no predefined relay affiliation group names; you can choose any group names that are logical to your deployment of BigFix.

Observe the following naming rules:

- Do not use special characters (including ".") when choosing names
- Group names are not case-sensitive
- Leading and trailing white spaces are ignored in comparisons

The ordering of relay affiliation groups is important for the client. The asterisk (*) has a special meaning in a relay affiliation list; it represents the set of unaffiliated computers. Unaffiliated computers are clients or relays that do not have any relay affiliation group assignments or have the asterisk group listing.

Note: The labels, defined in the client setting `_BESRelay_Register_Affiliation_AdvertisementList` and delimited by semi-colon (;), must not be bigger than 64 characters.

For more information, see Relay affiliation.

Assigning clients to relay affiliation groups

Clients are assigned to one or more relay affiliation groups through the client setting:

```
_BESClient_Register_Affiliation_SeekList
```

Set the client setting to a semi-colon (;) delimited list of relay affiliation groups, for example:

```
AsiaPacific;Americas;DMZ
```

Associating relays and server to affiliation groups

Relays and servers can be assigned to one or more affiliation groups through the client setting:

```
_BESRelay_Register_Affiliation_AdvertisementList
```

Set also client setting to a semi-colon (;) delimited list of relay affiliation groups, for example:

```
AsiaPacific;DMZ;*
```

Note: Relays and servers are not required to have a SeekList setting. The SeekList is used only by the client.

Notes about automatic relay assignment

The BigFix clients use a sophisticated algorithm to calculate which relay is the closest on the network.

The algorithm uses small ICMP packets with varying TTLs to discover and assign the most optimal relay. If multiple optimal relays are found, the algorithm automatically balances the load. If a relay goes down, the clients perform an auto-failover. This represents a major improvement over manually specifying and optimizing relays. However, there are a few important notes about automatic relay selection:

- ICMP must be open between the client and the relay. If the client cannot send ICMP messages to the relays, it is unable to find the optimal relay (in this case it uses the failover relay if specified or picks a random relay).
- Sometimes fewer network hops are not a good indication of higher bandwidth. In these cases, relay auto-selection might not work correctly. For example, a datacenter might have a relay on the same high-speed LAN as the clients, but a relay in a remote office with a slow WAN link is fewer hops away. In a case like this, manually assign the clients to the appropriate optimal relays.
- Relays use the DNS name that the operating system reports. This name must be resolvable by all clients otherwise they will not find the relay. This DNS name can be overridden with an IP address or different name using a task in the Support site.
- Clients can report the distance to their corresponding relays. This information is valuable and should be monitored for changes. Computers that abruptly go from one hop to five, for example, might indicate a problem with their relays.

Adjusting the BigFix Server and Relays

To get the best performance from BigFix, you might need to adjust the server and the relays.

There are two important ways of adjusting the flow of data throughout your network, throttling and caching:

Throttling Outgoing Download Traffic

Throttling allows you to set the maximum data rate for the BigFix Server. Here is how to change the data rate:

1. Open **Fixlets and Tasks** icon in the Domain Panel navigation tree and then click **Tasks Only**.
2. In the find window above the Tasks List, type "throttle" to search for the appropriate Task.
3. From the resulting list, click the task labeled **Server Setting: Throttle Outgoing Download Traffic**. A task window opens below. Make sure the **Description** tab is selected. There are three choices:
    - **Set the limit on total outgoing download traffic.** This choice allows you to directly set the maximum number of kilobytes per second you want to grant to the server.
    - **Disable the setting.** This option lets you open the download traffic on the BigFix Server to full throttle.
    - **Get more information.** This option opens a browser window with more detailed information about bandwidth throttling.
4. If you select a throttle limit, then from the subsequent **Take Action** dialog you can select a set of computers to throttle. Click **OK** to propagate the task.

Download Cache Size

BigFix Servers and Relays maintain a cache of the downloads most recently requested by Clients, helping to minimize bandwidth requirements.

1. Open **Fixlets and Tasks** icon in the Domain Panel navigation tree and then click **Tasks Only**.
2. In the find window above the Tasks List, type "cache" to search for the appropriate Task.

3. From the resulting list, click the task labeled **Relay / Server Setting: Download Cache Size**. A task window opens below. Make sure the **Description** tab is selected. Select the link to change the download cache size on the listed computers. This list might include Relays as well as the BigFix Server.
4. Enter the number of megabytes to cache. The default is 1024 MB, or one gigabyte.
5. From the subsequent **Take Action** dialog, select a set of computers and click **OK**.

Assigning a relay when the server is unreachable

After you install the client, it connects to and registers with the main BigFix server.

After the client registers with the main server, a master operator can assign the client to a primary relay as well as configure it to fail over to a secondary relay if the primary relay becomes unavailable.

In some cases, when the client is installed, it might be unable to reach the main server directly across the local area network or Internet. For example, if the client workstation is in a remote office and cannot make a connection through the enterprise firewall to reach the main server. In this case you must set up a DMZ relay that has been given access through a hole in the firewall. For more information, see Setting Internet Relays (on page 494).

You must also deploy the remote office client installer with a configuration file to set the client primary relay during installation. Specify the primary relay in the configuration file to register the client with a relay that it can connect to (such as the DMZ relay). For more information see Assigning Relay at Client Installation Time (on page 483).

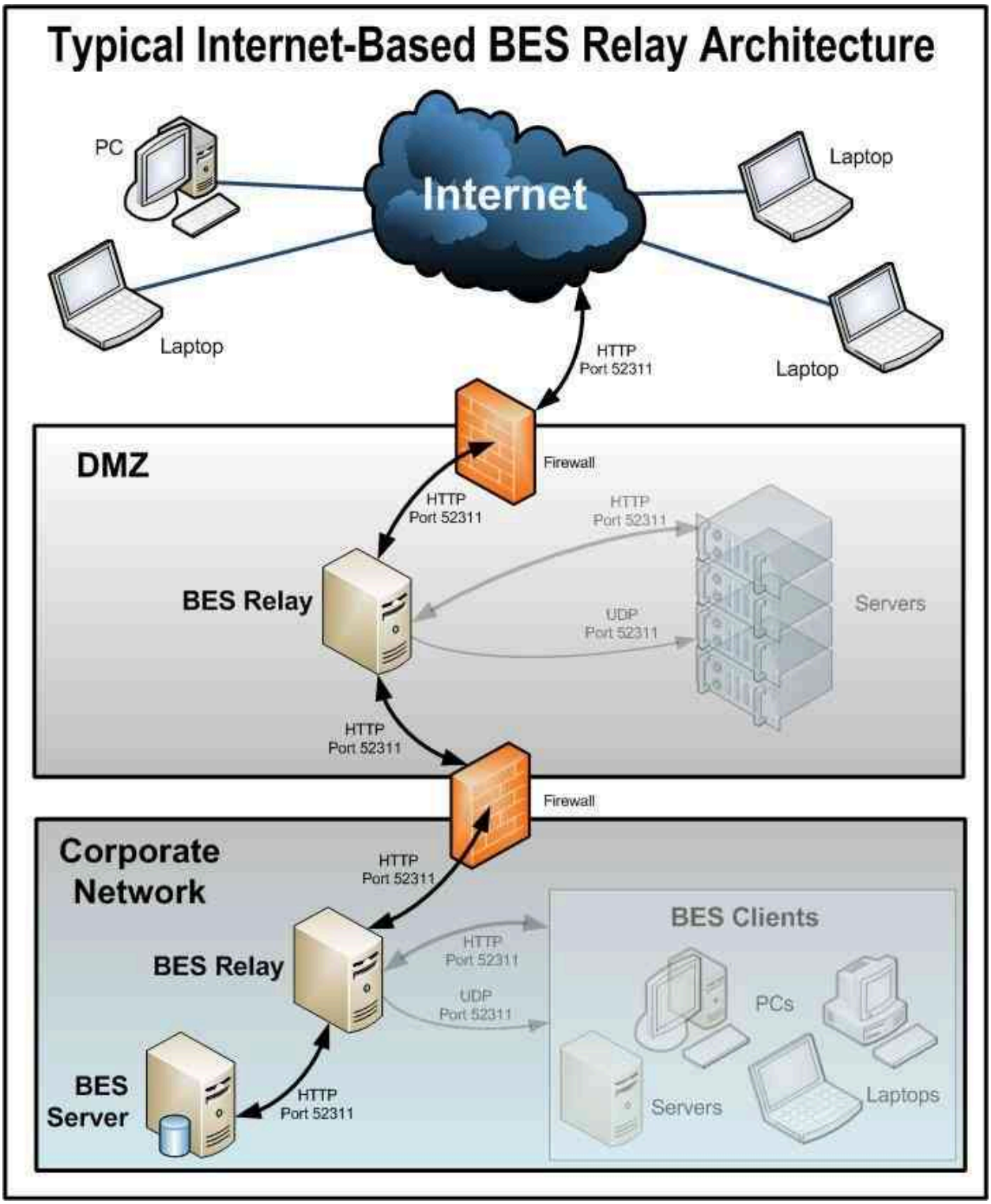
Setting up internet relays

You can configure your relays to manage clients that are only connected to the Internet without using VPN as if they were within the corporate network.

Using this approach, you can manage computers that are outside the corporate network (at home, in airports, at coffee shops, and so on.) using BigFix to:

- Report their updated properties and Fixlet status.
- Enforce new security policies defined by a Console operator.
- Accept new patch or application deployments.

This configuration is especially useful for managing mobile devices that might often be disconnected from the corporate network. The following picture shows a typical Internet-based relay, as it might exist in a DMZ network:



Setting up an Internet-facing relay enables external clients to find and connect to a relay. In our picture the clients can select the following types of relay:

- **Manual Relay Selection**: Clients can be configured using the console to manually select the Internet-facing relay DNS-alias (or IP address) as their primary, secondary, or failover relay. For more details about the failover relay setting, see Registration.
- **Automatic Relay Selection**: If ICMP traffic has been allowed from the Internet to a DMZ-based Internet relay, then automatic relay selection can be leveraged to allow clients to find the closest relay as they move from location to location (either within a corporate network or on the Internet). For external clients on the Internet, the only relay they are able to find and connect to is the Internet-facing relay (because ICMP traffic from the Internet would be blocked to the relays within the corporate network).



Note: You can use the feature Relay Affiliation to configure clients to find the most appropriate relay. For more details, see Relay affiliation.

This is how the relays, clients, and firewalls are configured in a typical internet-based BigFix relay architecture:

1. A relay is deployed in a DMZ and the internal DMZ firewall allows only BigFix traffic (HTTP Port 52311) between the DMZ relay and a designated relay within the corporate network. The design above suggests bidirectional traffic as opposed to only allowing the Internet-facing relay to initiate network connections to the relay within the internal corporate network. This enables quicker client response times because immediate notifications of new content are made to the Internet-facing relay thus maintaining a real-time synchronization of content. If the bidirectional communication between the Internet-facing BigFix relay and the relay in the corporate network is not allowed, the Internet-facing relay must be configured to periodically poll its parent (the relay within the corporate network) for new content. For more details about configuring command polling, see Command polling .
2. BigFix deployments that include 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). 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.

For instructions on how to set the internet-facing relays as authenticating, see Authenticating relays.

3. After relay communication is established between the DMZ and the internal corporate network, the external firewall also has to be opened to allow Internet-based client traffic (HTTP port 52311) to reach the DMZ relay. In addition, allowing ICMP traffic through the external firewall to the Internet-facing relay can aid in the external client auto-relay selection process.
4. A DNS-alias (or IP address) is assigned to the relay that enables external clients to find the DMZ-based Internet relay. The DNS-alias must be resolvable to a specific IP address.
5. To make the relay aware of the DNS-alias (or IP address) deploy the BES Relay Setting: Name Override Fixlet to the DMZ-based Internet relay.
6. Disable the relay diagnostics for Internet relays (by configuring `_BESRelay_Diagnostics_Enable`) or password-protect the page (by configuring `_BESRelay_Diagnostics_Password`). For details about the configuration settings, see Relay diagnostics.

   For more information about relay diagnostics, see Relay and Server diagnostics.
7. With the entire BigFix communication path established from the Internet through the DMZ-based Internet relay and ultimately to the main server, the next step depends on the various relay selection methods available in a given BigFix infrastructure.
8. Dynamic Policy Settings can be applied to Internet-based clients to allow for configurations better suited to external agents. For example, because the normal notification method (a UDP ping on port 52311) for new content might not reach external clients, dynamic settings can be used to have clients check for new content more frequently than the default period of 24 hours. For more information on setting up command-polling, see Changing the gather interval for a BigFix Client via the command polling client settings.

Related reference

List of settings and detailed descriptions

Related information

Relay diagnostics

Authentication

BigFix Configuration Settings

Viewing which relay is assigned to a client

How to see which clients are selecting which relays.

Run the following steps:

1. Start up the console and select the **BigFix Management** Domain.
2. From the **Computer Management** folder, click **Computers** to see a list of clients.
3. Look in the **Relay** column in the List Panel (this column might be hidden; in which case you might need to right-click the column headings and make sure **Relay** is checked). The BigFix Relay columns show information including the Relay method, service, and computer.

By default, the clients attempt to find the closest relay (based on the fewest number of network hops) every six hours.

Viewing the relay chain on the client

Starting from Version 9.5 Patch 13, the capability to view the relay chain on a specific BigFix client was added to the product.

To use this capability, both the client and the relay/server must be at Version 9.5 Patch 13 level or later. The purpose of this capability is to allow the client to trace his relay chain for each registration.

The relay chain information is stored within a new client folder named `RelayChain` located under the `BESClient/_BESData/_Global` client directory.

The Relay chain information is saved each day into a text file and each file follows the naming convention "yyyymmdd.txt"

Where:

yyyymmdd

Represents the year, month and day on which the relay chain information was saved.

A Sample TXT file is displayed below:

```
At 05:05:13 +0100 - S - s:11668927(server_hostname) -
 r:6843826(relay1_hostname)
- r:1083414982(relay2_hostname) - c:12183892(client_hostname)
At 11:05:10 +0100 - S - s:11668927(server_hostname) -
 r:6843826(relay1_hostname)
- r:1083414982(relay2_hostname) - c:12183892(client_hostname)
At 12:13:34 +0100 - S - s:11668927(server_hostname) -
 r:1083414982(relay2_hostname)
- c:12183892(client_hostname)
```

Note:

- If the computer ID is related to the server, it will be preceded by *s:*
- If the computer ID is related to the relay, it will be preceded by *r:*
- If the computer ID is related to the client, it will be preceded by *c:*

Each Relay chain information follows the same pattern:

```
computerID(computer_hostname)
```

Each Relay chain TXT file follows the same pattern:

```
At [hh:mm:ss] [local_zone] - [registration_response] -
 [computerID_1(hostname)]
-[computerID_2(hostname)] - ... - [computerID_3(hostname)]
```

Where:

hh:mm:ss

The time on which the registration occurred.

local_zone

The related time zone.

registration_response

The status of the registration. It can have two values: **S** for a successful registration. **F** for a failed registration.

computerID_1(hostname)

(Successful registration) It is the BigFix server.

computerID_2(hostname)

(Successful registration) It is the top-level relay that connects directly to the BigFix server.

...

(Successful registration) The child relays follow, if present in your environment.

computerID_3(hostname)

(Successful registration) It is the client that connects to the relay, or child relay if present in your environment.

The maximum amount of TXT files saved within the new client folder named `RelayChain` can be specified by using a new configuration setting named `_BESClient_Relay_Chain_Days`.

For more details about the setting, see Relay Management.

Failed Registration Scenario 1

In the following example, if the relay chain is not available, the string `N/A` is displayed instead of the computer ID. This error occurs, for example, when the client is at Version 9.5 Patch 13 level and the relay/server is at an older level.

A Sample TXT file is displayed below:

```
At 23:36:01 +0100 - F - N/A
```

Failed Registration Scenario 2

In the following example, the error `GetURL Failed` refers to a failed registration caused by a communication issue between two components inside the relay chain.

A Sample TXT file is displayed below:

```
At 08:04:55 +0100 - F - GetURL Failed
```

Failed Registration Scenario 3

In the following example, the registration failed at Relay with ID 6843826.

A Sample TXT file showing the relay chain is displayed below:

```
At 16:25:44 +0100 - F - r:6843826(relay1_hostname)
- r:1083414982(relay2_hostname) - c:12183892(client_hostname)
```

Viewing the relay chain on the BigFix server

After running the Fixlets named "TROUBLESHOOTING: Run BES Client Diagnostics" on Windows and "TROUBLESHOOTING: Run BES Client Diagnostics (Linux/UNIX/Mac)" on Linux and macOS, the BESClient data is uploaded to the BigFix server in the Upload Manager directory. Among other data, also the relay chain information of the client gets uploaded.

You can view the client relay chain information in the following BigFix server directory:

Windows

BigFix_Server_installation_path\UploadManagerData

Where the server installation path is by default `C:\Program Files (x86)\BigFix Enterprise\BES Server`

Linux

BigFix_Server_installation_path/UploadManagerData

Where the server installation path is by default `/var/opt/BESServer`

Chapter 14. Relay Health

Relay Best Practices

BigFix Relays play an important function in your deployment and maintaining proper health will prevent unnecessary issues from occurring. BigFix Relays aggregate gathering and downloading of content as well as report posting. Carefully review this document to ensure you are properly configuring your deployment's Relays.

More information on BigFix Relays can be found in Managing relays (on page 479).

Following are the best practices for maintaining the BigFix Relay health. These should be verified periodically in every deployment.

1. Ensure all internet-facing Relays are not allowing unsecured access (HTTP) from anyone
2. Ensure all BigFix Clients are reporting to a Relay
3. Ensure BigFix Clients are reporting to an intended nearby relay
4. Ensure there are <5000 Clients reporting to any individual relay
5. Ensure a minimal number of tiers in the relay hierarchy
6. Use redundant relays or Peer-to-peer mode in parts of the network with slow WAN connections

Internet-facing BigFix Relays are properly secured

Description

Putting a BigFix Relay in the DMZ is the proper way to manage laptops even when they are outside the corporate network, but additional security considerations should be made when doing so. Ensure Internet-facing BigFix Relays are configured to use Authenticating relays and have the Relay Diagnostics page password-protected or disabled. This configuration prevents unauthorized access to the internet-facing Relay and any of the download packages and content/actions on the Relay that may contain sensitive company information (e.g. software, vulnerability info, passwords, etc). When a Relay is authenticating, only your BigFix Clients can connect to it and they will use TLS (HTTPS)

communication for all BigFix traffic. It also prevents unauthorized access to the Relay Diagnostics page which may contain sensitive information.

How to Verify

In order to verify Relay Authentication, view Task: BES Client Setting: Enable Relay Authentication (ID #1297) in the BigFix Console. Any relays that are still relevant to this task are **not** authenticating. Deploy this task to any internet-facing Relays that are still applicable.

In order to verify Relay Diagnostics configuration, click the **Computers** node in the BigFix Console and view the **Client Settings** of any internet-facing relays to see if the appropriate Relay Diagnostics settings are set. Use the **Edit Settings** option to change values, as needed.

How to Troubleshoot Issues

Use Edit Settings and/or the Enable Relay Authentication task for each internet-facing Relay to adjust the configuration, as needed.

All BigFix Clients are using a BigFix Relay

Description

BigFix Clients must connect to either the BigFix Server or a BigFix Relay to gather the latest information about Fixlets and actions, download files, and post their information. In most deployments of BigFix, especially mid to large deployments, it is recommended that all the BigFix Clients use a BigFix Relay instead of using the BigFix Server. This tends to lead to better performance because the BigFix Clients can get the latest actions and download files faster and as a result, you see the BigFix Client action status update quickly. If some BigFix Clients are using the BigFix Server instead of a BigFix Relay, it is not necessarily a problem, but it is recommended that as few BigFix Clients report directly to the BigFix Server as possible to free up the BigFix Server for other tasks.

How to Verify

The easiest way to verify which BigFix Clients are using BigFix Relays is to use the **Relay** column in the BigFix Console. Look in the BigFix Console under the "Computers" tab. On the left, expand the "By Retrieved Properties" section and expand the "By Relay" filter (if you don't see the "By Relay" filter, right-click on the column headings and make sure "Relay"

is checked). This will show you the breakdown of where the BigFix Clients are currently reporting. A healthy deployment will have very few computers reporting to the DNS name of the BigFix Server (except the BigFix Relays).

Note: The Primary BigFix Relay and Secondary BigFix Relay show which BigFix Relays the BigFix Clients are supposed to choose if they are set to manual relay selection and the "Relay" column shows which BigFix Relay the BigFix Client currently has selected.

How to Troubleshoot Issues

There can be a number of reasons why the BigFix Clients are not currently reporting to a BigFix Relay:

- The BigFix Clients are set to manual BigFix Relay selection and no BigFix Relay is currently set.
- The BigFix Clients cannot resolve the BigFix Relay's DNS name.
- The BigFix Clients cannot contact the BigFix Relay because of NATs or firewalls.
- The BigFix Relay is not working properly.

Information on how to troubleshoot these issues and more are available at: BigFix Clients not choosing their specified BigFix Relay.

BigFix Clients are using a nearby BigFix Relay

Description

One of the primary benefits of BigFix Relays is that they can server as "distribution points" for large files, such as patches or applications. This ability allows for greatly reduced network usage especially across slow WAN pipes (the files are distributed to the BigFix Relay across the WAN and distributed from the BigFix Relay to the BigFix Clients locally). However, BigFix Clients must be properly set up to use the local BigFix Relay, otherwise, you will use more WAN bandwidth than necessary. You can set BigFix Clients to either automatically find their closest BigFix Relay based on network hops or manually select a BigFix Relay for BigFix Clients. In general, automatic BigFix Relay selection is suggested because it simplifies administration.

How to Verify

There are two basic ways to verify that the BigFix Clients are using a nearby BigFix Relay:

- The BigFix Clients will return the number of hops to the BigFix Relay that it is using (this will work only if the BigFix Client is using automatic relay selection). You can view these values in the BigFix Console or in a report to help determine if the BigFix Clients are choosing appropriate BigFix Relays. Look at the "Distance to BigFix Relay" retrieved property in your BigFix Console to view this information.
- Use the Task: BES Client Setting Relay Selection Controls (ID #154) to manage ICMP settings used for automatic relay selection.
- Using the BigFix Console or a report, you can view which BigFix Relays the BigFix Clients are using in each subnet or in each location. See Viewing which relay is assigned to a client *(on page 499)*. This will give you a good idea if any BigFix Clients are using the wrong BigFix Relays because the BigFix Clients in each location should usually all be using the same BigFix Relay(s). To view this information in the BigFix Console, filter "By Location" or "By Subnet" and then look at "By Relay" for each subnet/location to see the BigFix Relay distribution.

How to Troubleshoot Issues

- If the BigFix Clients are not using a particular BigFix Relay, try the suggestions listed at BigFix Clients not choosing their specified BigFix Relay.
- If the BigFix Clients are using automatic selection and you believe they are incorrectly choosing the wrong BigFix Relay, you might was to do a "tracert" from the BigFix Client to the BigFix Relay because there might be additional network hops that you were not aware of.
- If it appears that only a few BigFix Clients that are using automatic selection are choosing a non-optimal BigFix Relay, you can prompt them to immediately choose a new BigFix Relay (by default they will attempt to find a better BigFix Relay every 6 hours) by sending them a custom action with the action command **relay select**.

There are fewer than 5000 BigFix Clients using any BigFix Relay

Description

One of the main benefits of BigFix Relays is that they act as distribution points for files so that the main BigFix Server does not have to provide the file to each BigFix Client, but if there are too many BigFix Clients pointing at any single BigFix Relay, the BigFix Relay will become swamped when an action is sent out (especially if the file is large). This will cause actions to take longer to deploy while the BigFix Clients are waiting to get the files from the BigFix Relays. In most deployments, an optimal number of BigFix Clients reporting to each BigFix Relay is around 1000. Using recommendations from the Capacity Planning Guide, Relays can support up to 5000 Clients. Relays will still function beyond these limits, but responsiveness of Clients reporting to such Relays will be impacted.

How to Verify

In the BigFix Console, expand the **Computers** node and expand the **By Retrieved Properties > By Relay** filter. This will list each BigFix Relay that is being used along with how many BigFix Clients are reporting to each BigFix Relay. Alternatively, look at the **Relay Health Dashboard** under **Dashboards > By Site > BES Support**.

How to Troubleshoot Issues

If you are using manual relay selection and there are too many BigFix Clients using a BigFix Relay, then you should assign some of your BigFix Clients to a different BigFix Relay to reduce the load. If too many BigFix Clients are using a BigFix Relay and they are set to automatic relay selection, then you can add a BigFix Relay to the same subnet as the other BigFix Relay and the BigFix Clients will automatically distribute themselves among all BigFix Relays the same distance apart. Alternately, you can set BigFix Clients to manually point to a specific BigFix Relay, if necessary.

BigFix Relays all point to the BigFix Server or a top level relay

Description

In most deployments, especially smaller deployments, all BigFix Relays should be manually assigned to point directly back to the BigFix Server. Alternately, if there are many BigFix Relays, it is a good idea to have one or more BigFix Relay designated as a "top level" BigFix Relay and all the other BigFix Relays can point directly to the top level BigFix Relay. Note that unless there is a compelling network bandwidth limitation, it is generally better to have as few levels to the BigFix Relay as possible because each level introduces a little bit of

latency for the BigFix Client reporting. BigFix Relays should not use automatic BigFix Relay selection.

How to Verify

In the BigFix Console, click on the **Computers** tab and expand the **By Retrieved Properties > By Relay Installed > Yes > By Relay** filter. This will list the BigFix Relays the each of the BigFix Relays are using. Alternatively, look at the **Relay Health Dashboard** under **Dashboards > By Site > BES Support** to view the relay hierarchy.

How to Troubleshoot Issues

If the BigFix Relay hierarchy is not set properly, set the BigFix Relays to all manually point to a top level BigFix Relay or the main BigFix Server.

Redundant BigFix Relays are set up for slow WAN pipes

Description

Putting a BigFix Relay in each location with a slow WAN link is vital to save bandwidth; however, if the BigFix Relay computer is turned off, crashes, loses network connectivity, or for any reason is inaccessible, then the BigFix Clients will attempt to find their next closest BigFix Relay and if this occurs during an action push, you will potentially overwhelm the WAN pipe. One way to reduce the risk of this is to set up redundant BigFix Relays in each location that is connected over a slow WAN pipe. In this case, if one BigFix Relay goes down, the other local BigFix Relay will be used by the BigFix Clients. Alternatively, starting in BigFix 9.5.11, peer to peer mode can be used to allow BigFix clients to share downloads with each other without pulling directly from a Relay. In this configuration, if a Relay is not available or is turned off, one Client in the subnet will request the download from an upstream Relay and share it with other Clients on the subnet.

How to Verify

In order to verify this, you will need a subnet property or location property setup that will allow you to get an idea of the location of the BigFix Relays. You will also need to know which subnets/location are connected through a slow pipe. With this information, you can open the BigFix Console, click on the **Computers** tab and expand the **By Retrieved Properties > By Relay Installed > Yes > By Location/BySubnet** filter. This will show how

many BigFix Relays are in each location. In order to use peer to peer mode, see Working with PeerNest and configure the endpoints accordingly.

How to Troubleshoot Issues

You will need to add redundant BigFix Relays as necessary for each location. In the case of a very slow WAN connection, it is recommended to have at least one relay that is not shared by a user, and that would never be turned off. Use tasks "PeerNest: Enable Client" (ID #3664) and "PeerNest: Disable Client" (ID #3661) to enable/disable peer to peer mode on Clients, as needed.

Chapter 15. Introduction to Tiny Core Linux - BigFix Virtual Relay

Follow the step-by-step sequence of operations needed to build the virtual machine, from the downloading of the ISO image to the complete setup and configuration of the BigFix Virtual Relay.

You can use Tiny Core Linux - BigFix Virtual Relay to deploy new relays throughout your virtual enterprise in a cost-effective way.

Deploy a Virtual Relay, based on the Tiny Core Linux platform, to achieve the following benefits:

- Low number of security exposures, because a minimal set of services is installed and required.
- Low number of resources, because a reduced amount of RAM/CPU/HD is required.
- Low maintenance and deployment efforts are needed.
- Possibility to deploy either a single virtual relay instance by using the manual deployment or to deploy a big number of instances by using the auto-deployment feature.

Architectural overview

Tiny Core Linux (TCL) runs from a RAM copy created at boot time.

Besides being fast, Tiny Core Linux protects system files from changes and ensures a pristine system on every reboot. Tiny Core Linux is easy, fast, simple to be renewed, and stable.

The TCL solution is a mini Linux distribution, which provides the following advantages:

- A minimal set of services installed by default.
- The capability to easily and quickly customize the base core system with new extensions.

To have a BigFix Virtual Relay up and running, the following additional libraries are required:

- acl.tcz
- attr.tcz
- syslinux.tcz
- nspr.tcz
- nss.tcz
- popt.tcz
- tzdata.tcz
- sqlite3.tcz

Note:

You can download the libraries listed in the table from the following website:

http://distro.ibiblio.org/tinycorelinux/

In the Downloads section.

The BigFix Virtual Relay uses a RAM disk for the core operating system files. The Linux Kernel `core.gz` loads the virtual machine at boot time, which gets extracted directly into the RAM disk. The `core.gz` is distributed as part of the Tiny Core Linux product and contains the root file system and system support applications.

The BigFix client and relay are supplied as Tiny Core extensions.

You can deploy the BigFix Virtual Relay by:

1. Creating a virtual machine and installing Tiny Core Linux on it, as described in Phase 1 - Configuring the Tiny Core Linux virtual machine *(on page 513)*.
2. Creating a virtual machine template that is needed to deploy the BigFix Virtual Relay, as described in Phase 2 - Preparing the BigFix Virtual Relay template *(on page 521)*.
3. Using the template, deploying the BigFix Virtual Relay instance, as described in Phase 3 - Configuring the BigFix Virtual Relay instance *(on page 531)*.

The deployment can be separated into phases.

Phase 1 - Configuring the Tiny Core Linux virtual machine

In Phase 1, you set up a virtual machine with the Tiny Core Linux operating system that can instantiate the BigFix relay.

The setup process comprises three steps:

1. Download the ISO image.
2. Create a virtual machine.
3. Install Tiny Core Linux on the virtual machine.

Downloading the ISO image

The ISO image is available at http://tinycorelinux.net.

In the Downloads section, select a version supported by BigFix, and download the related CorePlus .iso file.

For an updated list of Tiny Core versions supported by BigFix, refer to the BigFix 11 - Detailed System Requirements page.

Creating a virtual machine

Starting from your ESXi hypervisor, create a new virtual machine.

Complete the following steps:

1. Create a new **Custom** virtual machine.
2. Provide the virtual infrastructure specific information depending on your environment, such as:
    - Name and location (for example: Specify "Template BigFix VR")
    - Host and cluster
    - Resource pool
    - Data store
    - Virtual machine version. Choose the latest version.
3. Provide the virtual machine settings. Set the guest operating system by selecting **Linux** and specify the **Other 2.6.x Linux (32-bit)** as minimum version. This choice

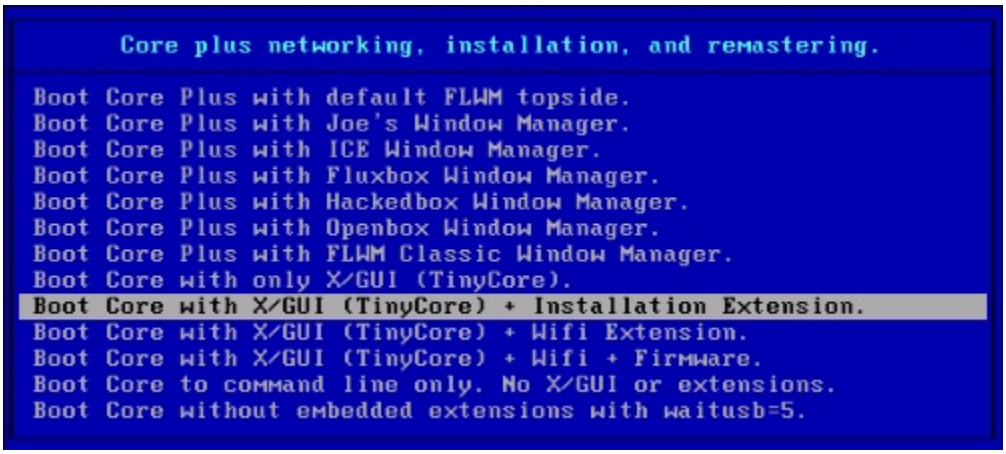
allows you to use the VMware Tools. The last tested configuration is Tiny Core 12.0 with "Other 4.x Linux (32-bit)".

4. Specify the virtual machine resources:
    - For details about the BigFix Relay resources configuration, refer to the Capacity Planning Guide at BigFix Performance & Capacity Planning Resources.
    - According to your available networks, select the name of your network and select **VMXNET3** as its adapter. This choice allows you to use the network driver provided by VMware Tools.
    - Set the disk size according to your needs and choose **IDE 0** as "Virtual Device Node". IDE is the only supported Virtual Device Node.
5. From the summary panel, review the configuration displayed, enable the editing of the virtual machine settings, and click **Continue**.
6. Edit the virtual machine settings to mount the CorePlus ISO image previously downloaded:
    a. Browse the data store for the ISO file that you downloaded from http://tinycorelinux.net
    b. Enable the DVD-ROM to be connected to the virtual machine at startup time.
    c. Click **Finish**.

Installing Tiny Core Linux on the virtual machine

Start the virtual machine that you created and follow these steps.

Choose the installation option shown in the following screen capture:



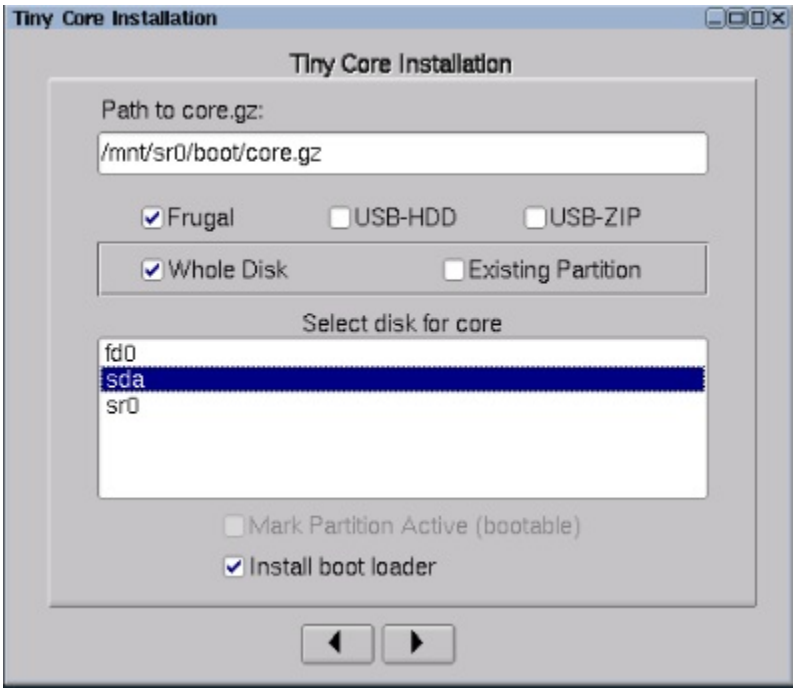
Run the Tiny Core installer by clicking TC_Install:



Start the Tiny Core installation and follow these panels to install it on an empty hard drive:

Leave the pre-filled path as core.gz (default path).

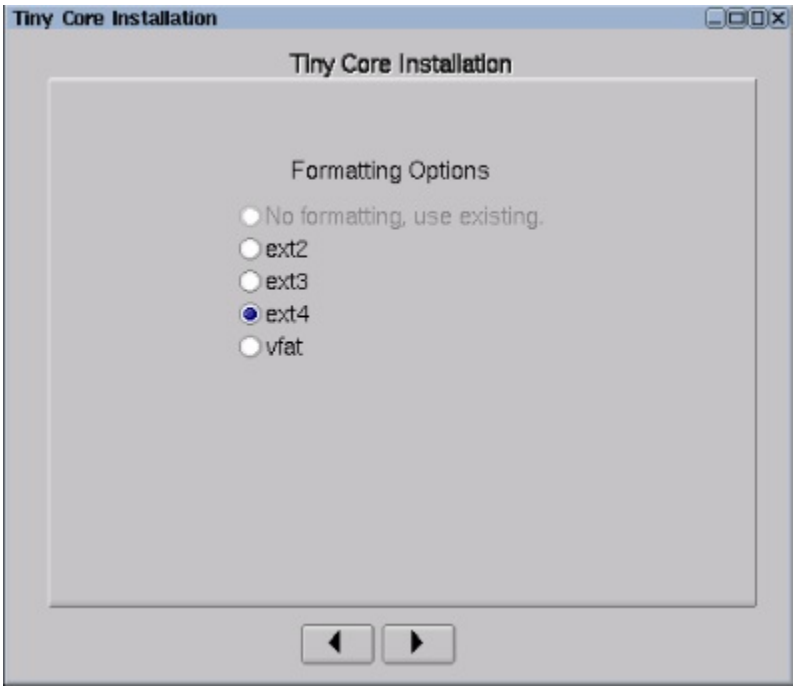
Select the check box **Whole Disk** and select **sda** as the core disk.



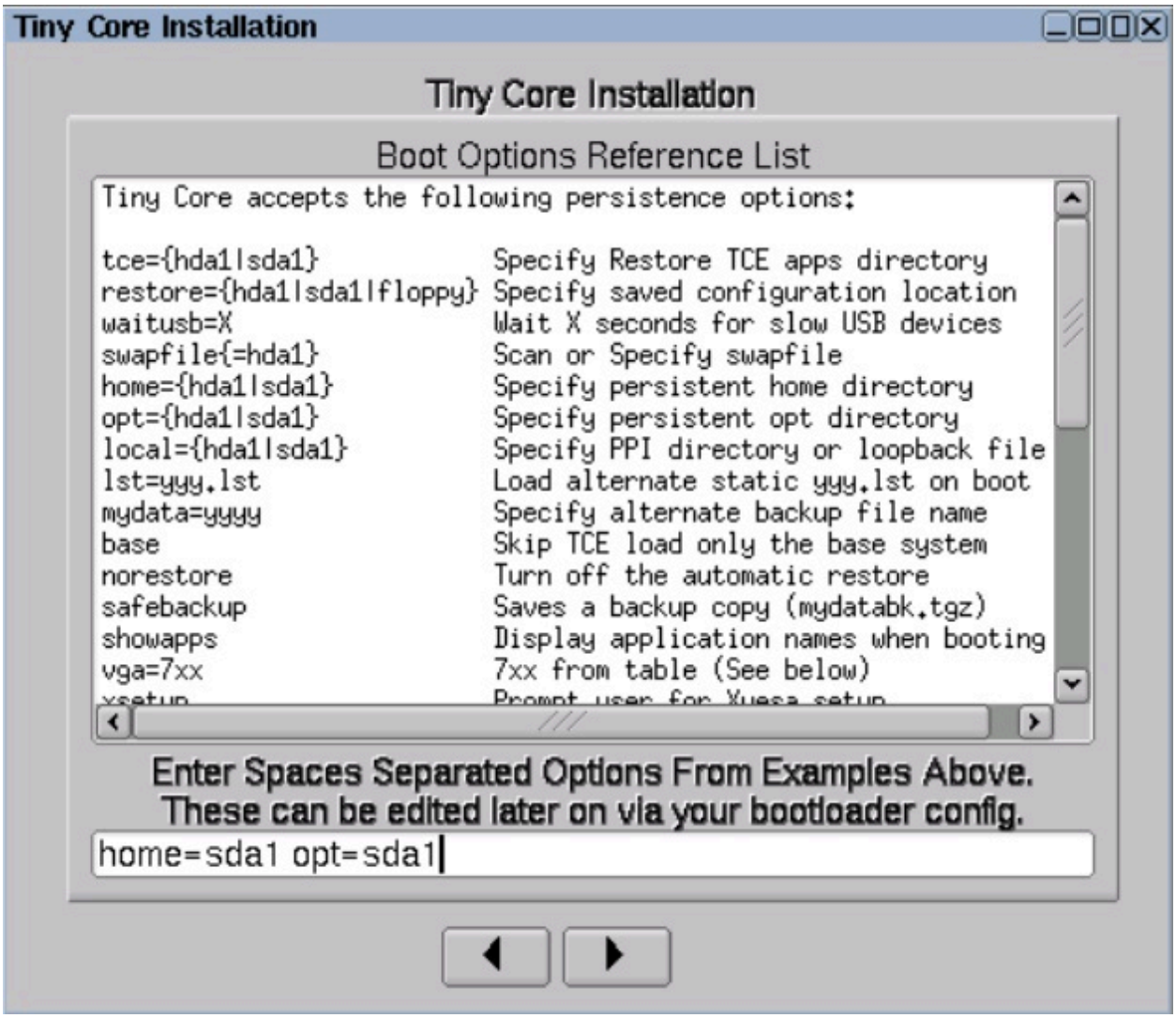
Note: Frugal is the typical installation method for Tiny Core Linux. You basically have the system in two files, vmlinuz and core.gz, whose location is specified by the boot loader.

Any user files and extensions are stored outside the base operating system.

Format the new partition. It is recommended that you select the **ext4** option to support the Linux permissions.



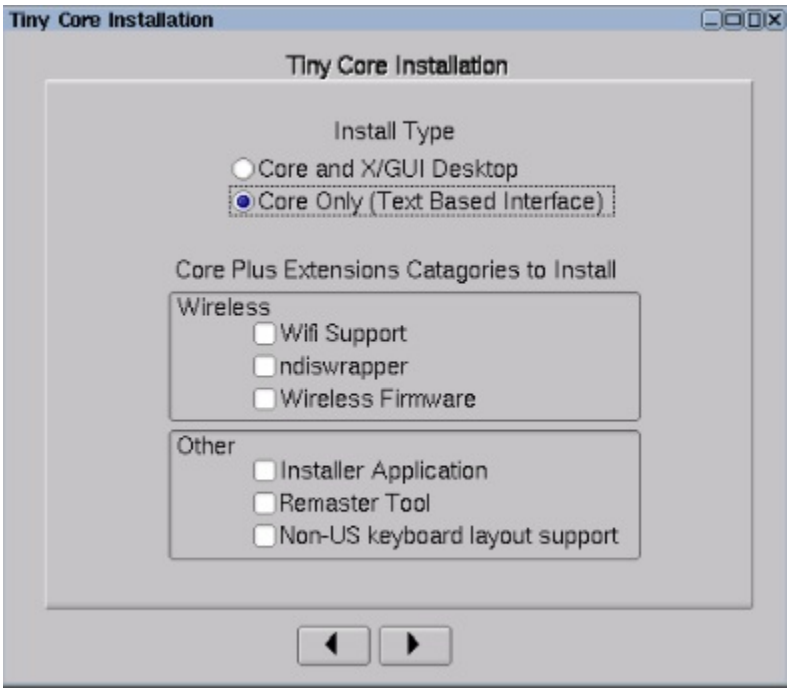
If you want to use additional boot codes, such as screen resolution or keyboard mapping, enter them now.



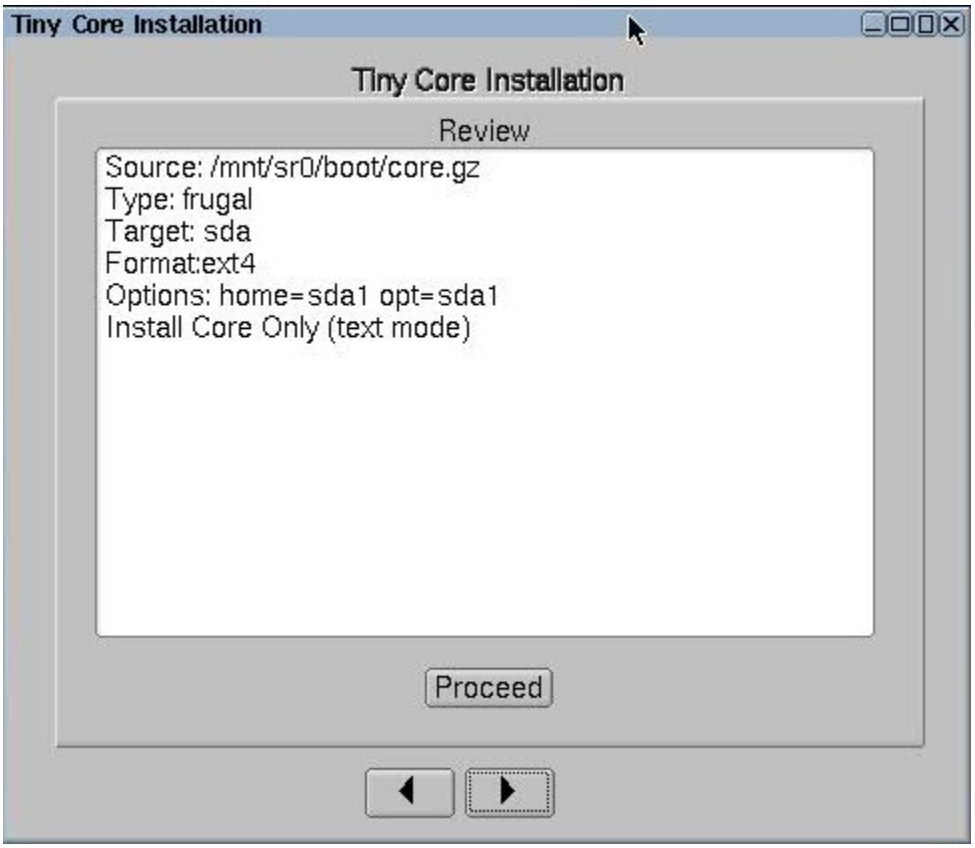
Set the location of the persistent **home** directory **home=sda1**.

Set the location of the persistent **opt** directory **opt=sda1**.

Note: These two directories become persistent when managing these two file systems.

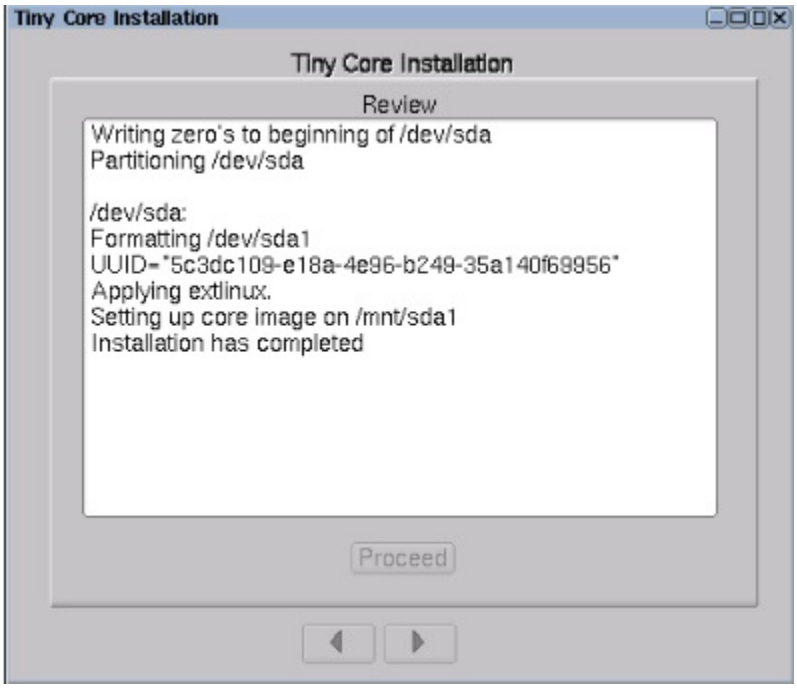


Select the **Core Only (Text Based Interface)** option to have a virtual machine with only the CLI.



If everything is OK, click **Proceed**.

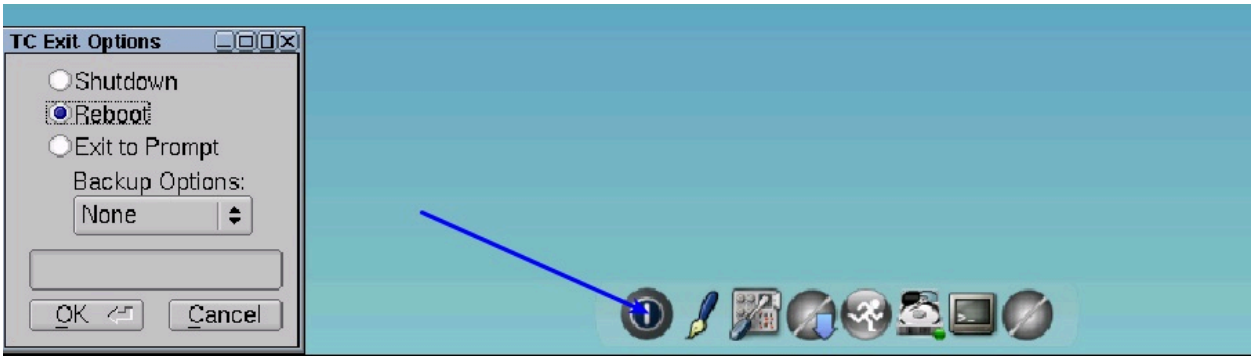
The time required varies depending on the size of your hard drive.



Tiny Core Linux is now installed.

Restart the machine by clicking Exit, selecting **Reboot**, and clicking **OK**.

By choosing to reboot, you ensure the data persistence. Do not use Shutdown at this stage.



Phase 2 - Preparing the BigFix Virtual Relay template

After installing Tiny Core Linux on the virtual machine, shut down the virtual machine.

Edit the properties to connect the `BESRelay-x.x.x.xx-tcl.i686.iso` file:

1. Browse the path where you downloaded and saved the ISO image.
2. Select the `BESRelay-x.x.x.xx-tcl.i686.iso` file.
3. Click **OK**.

Start the virtual machine.

After starting the virtual machine, launch the setup to configure the template from the command line:

1. Mount the BigFix ISO image by running: `mount /mnt/sr0/`
2. Untar the tar file provided in the ISO image by running: `tar -xvf /mnt/sr0/bessetup.tar`
3. Launch the setup by running the `./setup` command.



The Virtual Machine Template Configuration Tool displays.

Specify if you are using the network or a local folder for the complete template setup:



Note: The entire setup is performed either using the network or a local folder. Both are used to customize the virtual relay template.

If you specified "n" (no), see Template setup and customization from a local folder (on page 523).

If you specified "y" (yes), see Template setup and customization from the network (on page 526).

Template setup and customization from a local folder

How to create and customize the template using a local folder.

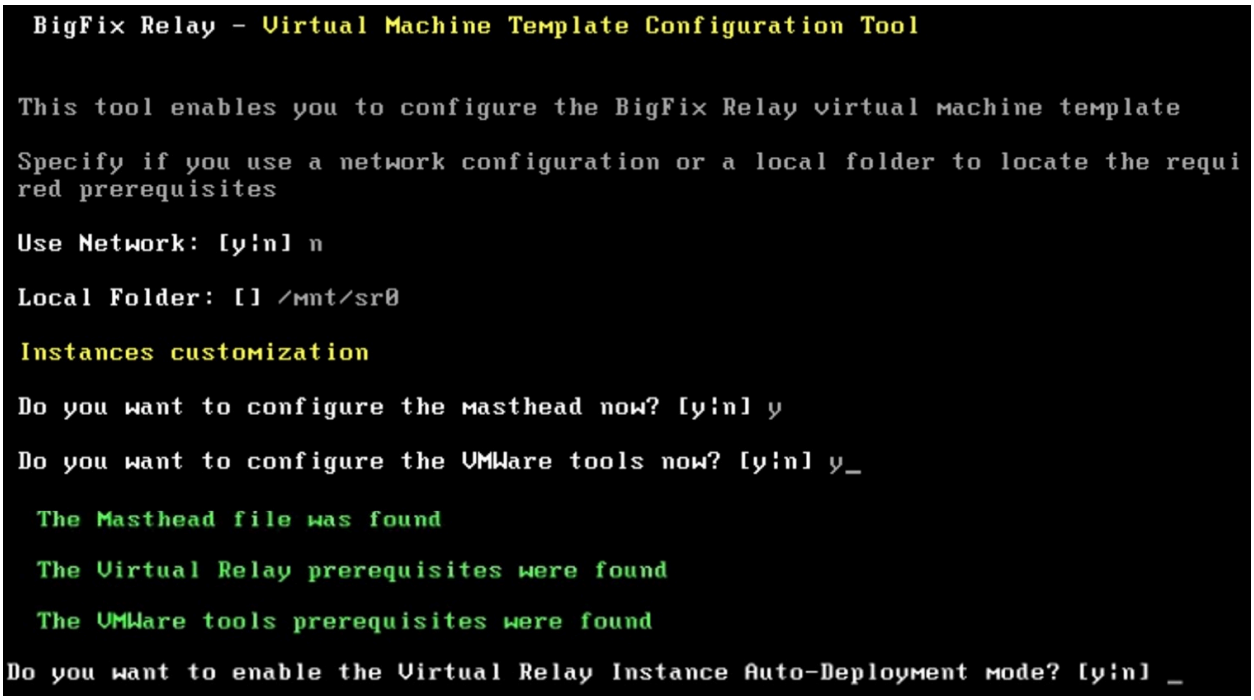
To use this option, you must have previously downloaded all the required files needed for the:

- Operating system prerequisites (mandatory to install and run the Virtual Relay).
- Masthead configuration (optional).
- VMware Tools configuration (optional).

For details about the operating system prerequisites, see Architectural overview (on page 511).

For the masthead configuration, ensure that you retrieve, from the BigFix server (or relay), the `actionsite.afxm` file.

For details about the VMware Tools prerequisites, see VMware and Open VM tools (on page 539).



Note: Depending on which selections you made in the configuration tool, if some required prerequisites cannot be located in the local folder specified, the configuration tool displays warnings or error messages stating which prerequisites are missing and which configurations cannot be performed.

In the previous panel, you must:

- Specify if you want to install the masthead file.
- Specify if you want to install the VMware Tools.

If you specified "y" (yes) to both the previous selections, the following choice is displayed:



Choose if you want to enable the automated deployment.

Note: Only if you choose to not enable the automated deployment (Auto-Deployment mode), you have the possibility to decide if you want to configure the manual deployment of the virtual relay instances in DHCP mode or using the static IP parameters. If you decide to enable the automated deployment, this choice is not displayed.

The virtual relay instances will be deployed according to the choice you make.

For more details about the two different types of deployment, automated or manual, see Phase 3.

If you specified "y" (yes) to the Auto-Deployment mode, enter the user password required by the tc user:



If you specified "n" (no) to the Auto-Deployment mode, you must choose if you are using the DHCP mode or the static IP address for the deployment of the instances:



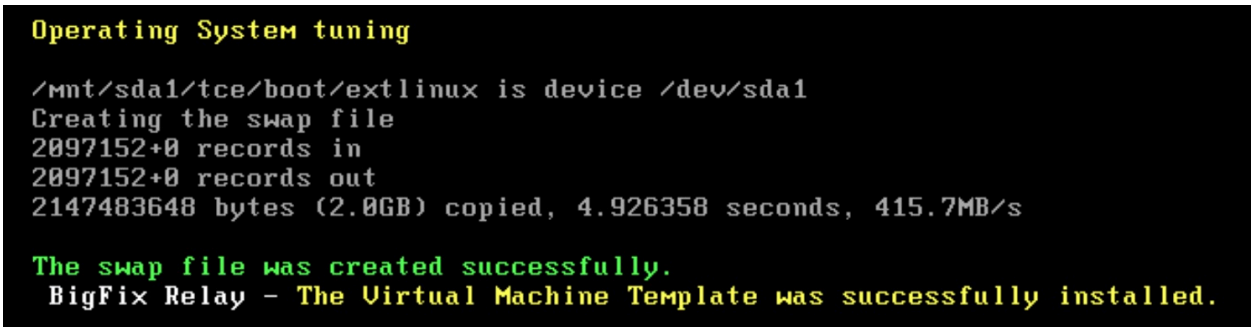
If you specify "y" (yes) to the DHCP mode, the instances will be deployed requiring only the host name; the network settings will be automatically in DHCP mode.

If you specify "n" (no) to the DHCP mode, the instances will be deployed requiring the host name and the network parameters.

Enter the country code related to your time zone and, if your country has more than one time zone, enter the appropriate number for the time zone that you want to set.



The operating system automatically creates a swap area to optimize the template creation.

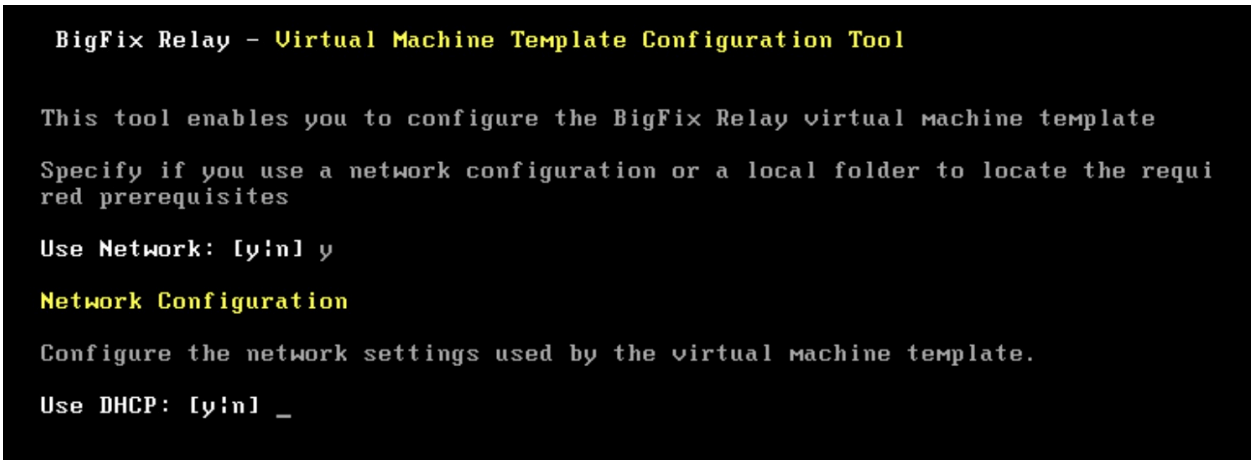


When the operation is completed, the template is successfully installed and you can power off the virtual machine as follows:

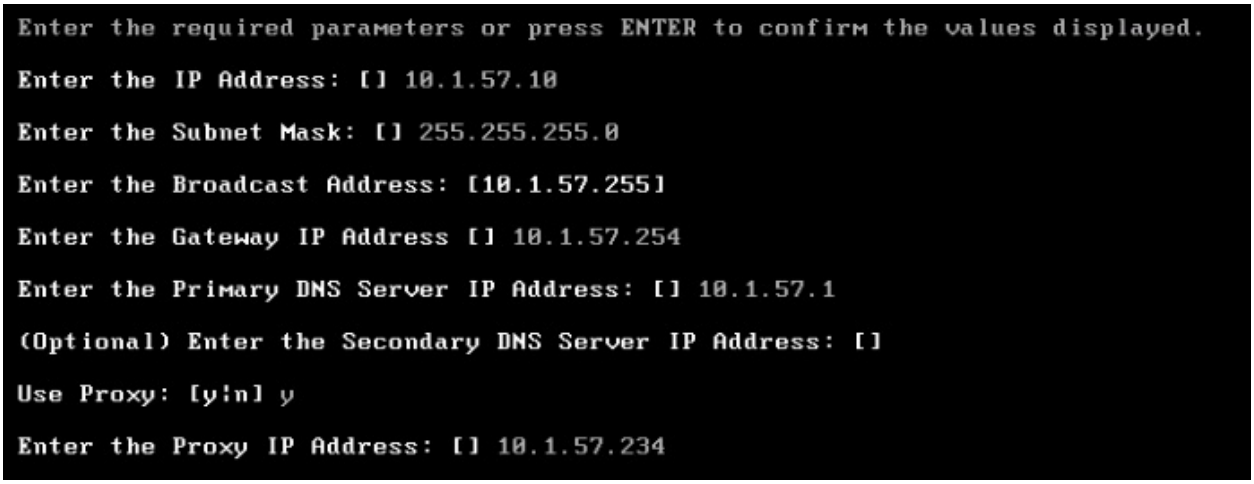


Template setup and customization from the network

To use this option, you must have Internet access from the virtual machine on which you are configuring this template.



If you specified "n" (no) to the Use DHCP option, enter the static IP parameters of your network.



```
Enter the Proxy Port: [] 1234

Enter the Proxy User: []

Review the entered parameters and verify if they are correct.

     Use DHCP: n
   IP Address: 10.1.57.10
      Netmask: 255.255.255.0
    Broadcast: 10.1.57.255
      Gateway: 10.1.57.254
  Primary DNS: 10.1.57.1
Secondary DNS:
    Use Proxy: y
Proxy IP Address: 10.1.57.234
      Proxy Port: 1234

Is the configuration correct? [y;n]
```

Confirm that the network configuration is correct by entering "y" (yes).

Otherwise, enter "n" (no) to restart the parameter input.

If you specified "y" (yes) to the Use DHCP option, ensure that the DHCP server of your network is up and running.

```
Review the entered parameters and verify if they are correct.

     Use DHCP: y

Is the configuration correct? [y;n] _
```

To customize the instance template, complete the following selections:

- Specify if you want to install the masthead file.
- Specify if you want to install the VMware Tools.

```
Instances customization

Do you want to configure the masthead now? [y;n] y

Do you want to configure the VMWare tools now? [y;n] y
```

If you specified "y" (yes) to both selections, the following choice is displayed:



Choose if you want to enable automated deployment.

The virtual relay instances will be deployed according to the choice you make.

For more details about the two different types of deployment, automated or manual, see Phase 3.

If you specified "y" (yes) to the Auto-Deployment mode, enter the user password required by the **tc** user:



If you specified "n" (no) to the Auto-Deployment mode, you must choose if you are using the DHCP mode for the deployment of all future instances:



If you specified "y" (yes) to the DHCP mode, the instances will be deployed requiring only the host name; the network settings will be automatically in DHCP mode.

If you specified "n" (no) to the DHCP mode, the instances will be deployed requiring the host name and the network parameters.

If during the template creation you specified "y" (yes) for the masthead setup, the following panel is displayed:

```
Masthead Installation

BigFix Server (or Relay) IP Address: [] 10.1.57.20

BigFix Server (or Relay) Deployment Port Number: [52311] _
```

Enter the BigFix server parameters, IP address and port number.

Note: Even if you specify the relay IP address, the Virtual Relay will connect directly to the server and not to the indicated relay.

Enter the country code related to your time zone and, if your country has more than one time zone, enter the appropriate number for the time zone that you want to set.

```
Time zone setup

Enter the country code to set the time zone: [] it

Countries list:

 1 - Europe/Rome

Enter the number specific for your time zone: [] 1

The time zone was changed to Europe/Rome

Is the time zone correct? [y|n] _
```

The operating system automatically creates a swap area to optimize the template creation.

```
Operating System tuning

/mnt/sda1/tce/boot/extlinux is device /dev/sda1
Creating the swap file
2097152+0 records in
2097152+0 records out
2147483648 bytes (2.0GB) copied, 4.926358 seconds, 415.7MB/s

The swap file was created successfully.
 BigFix Relay - The Virtual Machine Template was successfully installed.
```

When the operation is completed, the template is successfully installed and you can power off the virtual machine as follows:

```
tc@virtualrelay:~$ sudo poweroff_
```

Creating the BigFix Virtual Relay template

After the virtual machine was powered off, you can generate the virtual machine template.

Complete these steps from the Virtual Center:

1. Right-click the virtual machine that you created.
2. Select **Template > Convert to Template**.

Phase 3 - Configuring the BigFix Virtual Relay instance

Depending on the choice you made when creating the template (manual deployment or auto-deployment), the method of deploying future virtual relay instances changes.

From the template, deploy the virtual machine according to your specific virtual center architecture, in terms of data store and network availability.

In case of auto-deployment, see Deploying a new BigFix Virtual Relay instance by using Auto-Deployment (on page 531).

In case of manual deployment, see Manually deploying a new BigFix Virtual Relay instance (on page 533).

Deploying a new BigFix Virtual Relay instance by using Auto-Deployment

You can use this function to automatically deploy a consistent number of virtual relays based on the template created in Phase 2 by answering "y" (yes) to the option "Do you want to enable the Virtual Relay Instance Auto-Deployment mode?".

To deploy with the Auto-Deployment mode, you must create a script that automatically instantiates the virtual machines and provides the correct configuration files for each virtual machine.

In case of auto-deployment, some configuration files are needed to deploy the instance. Before the virtual machine receives the required configuration files, it remains in a waiting state.

To use the auto-deployment mode, the following three configuration files are required:

- `besclient.config` which must be copied into the `/var/opt/BESClient` local directory.
- `besrelay.config` which must be copied into the `/var/opt/BESRelay` local directory.
- `network.conf` which must be copied into the `/opt/bigFix/config` local directory.

The first two files contain some additional properties for the client and relay configurations. For the content of these files, see BigFix Platform on HCL Knowledge Center.

Note: The `besclient.config` and `besrelay.config` files are optional files. The client uses the default files if they are not provided. You must provide them only if you want to customize the virtual relay at startup time. For example, if you want to connect the virtual relay to another relay and not to the BigFix server, which is the default behavior.

The network.conf file contains the details about the network parameters of the deployed instances. It defines if the machine should start in DHCP mode or by using static IP settings. It must be the last configuration file to be copied. For details about the two different `network.conf` templates, see Auto-Deployment (on page 541).

After receiving the configuration file, the instance automatically starts the BESRelay and the BESClient services and registers with the BigFix server or with the relay specified in the previously defined configuration file.

The auto-deployment is particularly useful because:

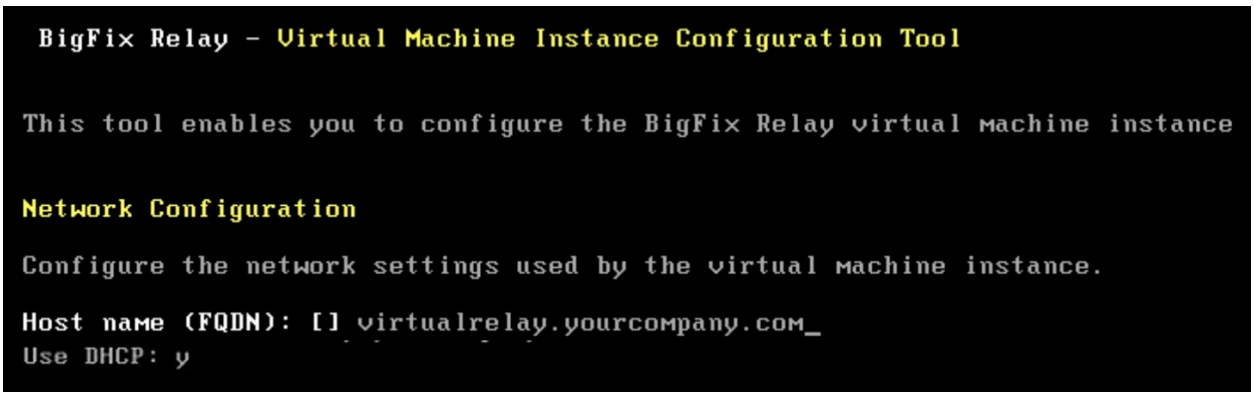
- No further configurations are required.
- It allows an easy and rapid multiple deployment.

Note: Before enabling the auto-deployment mode, ensure that you have installed vSphere PowerCLI, needed to copy the `network.conf` file and the other configuration files.

Manually deploying a new BigFix Virtual Relay instance

In case of manual deployment, the network parameters request is automatically displayed.

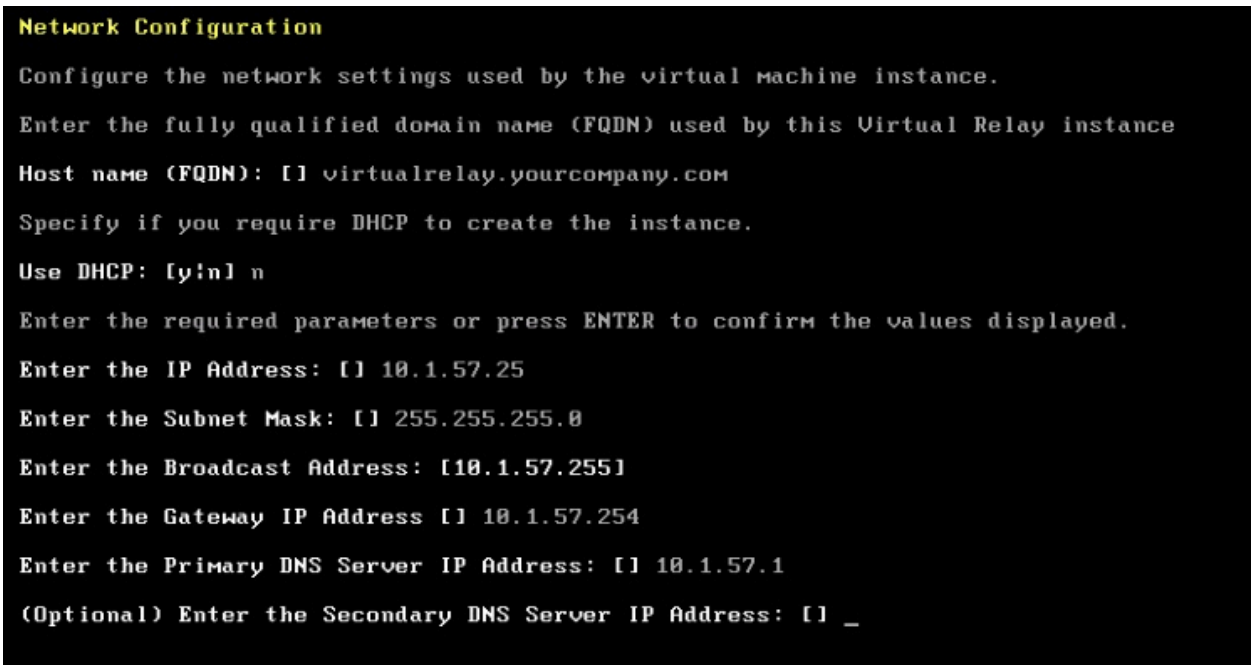
If you specified "y" (yes) to the DHCP mode when creating the template, you are required to enter the host name used by the virtual machine instance and review the displayed DHCP information.



If you specified "n" (no) to the DHCP mode when creating the template, you are required to enter the host name used by the virtual machine instance and all the related network parameters.



Enter the static IP parameters of your network.



Review or confirm that the displayed parameters are correct by pressing ENTER.



Confirm that the network configuration is correct by entering "y" (yes).

If during the template creation you specified "n" (no) for the masthead setup, the following panel is displayed:

```
Masthead Installation
BigFix Server (or Relay) IP Address: [] 10.1.57.20
BigFix Server (or Relay) Deployment Port Number: [52311] _
```

Enter the BigFix server parameters, IP address and port number.

Note: You cannot provide as an input into the configuration procedure of Tiny Core an authenticating relay (_BESRelay_Comm_Authenticating=1), because the current configuration procedure does not allow you to enter the password required to perform a registration to an authenticating relay. Therefore, the first parent relay of a Tiny Core relay cannot be an authenticating one. But once registered, the Tiny Core relay can be moved as a child of an authenticating relay and it can be configured to be itself an authenticating relay.

Note: Even if you specify the relay IP address, the Virtual Relay will connect directly to the server and not to the indicated relay.

The instance was configured successfully and the BESClient and the BESRelay services begin to start.

```
 BigFix Relay - The Virtual Machine Instance was successfully installed.

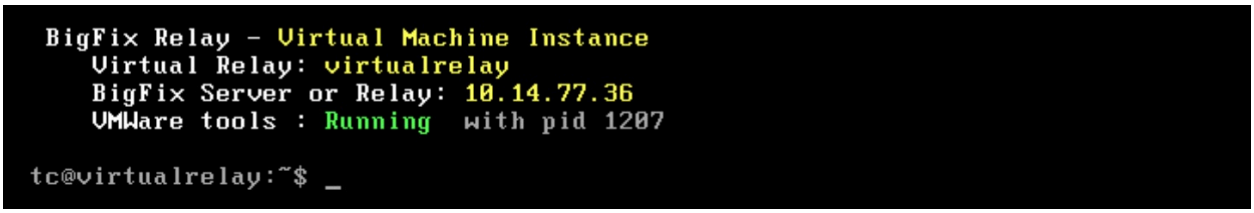
Successfully started the BigFix BESRelay

Successfully started the BigFix BESClient
```

Log in as **tc** user.

After logging in, the following information is available:

- The Virtual Relay name.
- The BigFix server IP address.
- The VMware Tools status (Running, Not running, Not installed).



You can check the status of the BESClient and the BESRelay services as follows:



The Virtual Relay instance is ready to use.

Maintenance

After completing Phase 3, the Virtual Relay is fully operational and communicates with the BigFix core server.

Additional configuration, such as setting the maximum relay cache size or changing the relay advertised name, can be done directly from the BigFix Console, exactly like any other BigFix relay.

No in-band mechanism for remotely accessing the shell is provided by default.

Using a Fixlet, you can manage the following two types of upgrade:

Client and relay components

The client and relay components upgrade will be performed by using a specific Fixlet which allows the components upgrade within the BigFix infrastructure.

Tiny Core Linux platform

The Tiny Core Linux platform upgrade will be performed by using a specific Fixlet which allows the component upgrade within the BigFix infrastructure.

Changing the shell password

To change the shell password, log in using the existing password, then issue the following command:

```
tc@virtualrelay:~$ sudo passwd tc_
```

You are prompted to enter the new password again.

To ensure the password persistence, insert the following three lines into the `/opt/.filetool.lst` file:

etc/passwd

etc/shadow

etc/group

and issue the following command:

```
tc@virtualrelay:~$ filetool.sh -b_
```

Note: These steps are performed automatically if you deploy using the Auto-Deployment mode.

Rebooting the Virtual Relay

If you are within the shell and need to reboot the system, issue the following command:

```
tc@virtualrelay:~$ sudo reboot_
```

Updating the Virtual Relay instance

After creating and deploying the virtual relay instance, you can still modify the deployed instance in terms of masthead installation or network configuration.

From the operating system command line, launch the script named `bigFixLaunchMastHeadConfiguration` to modify the BigFix server or relay IP address for the masthead installation.

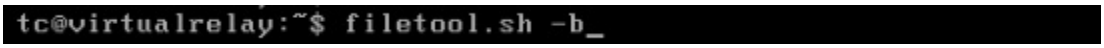
From the operating system command line, launch the script named `bigFixLaunchNetworkConfiguration` to modify the current network parameters of the instance. For example, you can change an existing DHCP configuration into a network configuration that uses static IP parameters.

Time zone

After creating the template, you can still modify the current time zone set during the configuration by launching the script named `bigFixTimeZone` from the template shell command line as follows:

```
tc@virtualrelay:~$ bigFixTimeZone
Time zone setup
Enter the country code to set the time zone: [] it
Countries list:
 1 - Europe/Rome
Enter the number specific for your time zone: [] 1
The time zone was changed to Europe/Rome
Is the time zone correct? [y|n] _
```

After setting the new time zone, run the following operating system command:



Troubleshooting

Some known issues.

- All the steps performed to configure the BigFix Virtual Relay template and instance are logged in the `besSetup.log` file available in the `/opt/bigFix/log` directory. Only the steps that belong to the normal flow are logged, not the additional steps described in Modifying the Virtual Relay template *(on page 542)*.
- Ensure that you start or restart the BigFix relay service using sudo permissions or logged in as root. Otherwise, an error message is returned and the relay service is not started.

Limitations

Some known limitations.

Double network interface

When creating the virtual machine, you can specify only one network adapter. Currently, the second network adapter is not supported.

VMware and Open VM tools

If during the template creation you specified "y" (yes) to the VMware Tools configuration, only a minimal set of Open VMware Tools was installed, because they are running on a Tiny Core Linux machine with a Text-based user interface only.

The minimal set provides the following capabilities:

- Automatic mouse grab and ungrab
- VMXnet3 network module instead of the PCNet32 module
- Host shared folder access using HGFS (mounted at /mnt/hgfs)

- Time synchronization with host
- Copy/paste to/from host/guest

To have the minimal set of Open VMware Tools up and running, the following additional libraries are automatically installed during the template creation (phase 2):

- fuse.tcz
- libdnet.tcz
- openssl-1.1.1.tcz
- open-vm-tools.tcz
- glib2.tcz
- libffi.tcz
- gamin.tcz
- pcre.tcz

Note:

You can download the libraries listed in the table from the following website:

http://distro.ibiblio.org/tinycorelinux/

In the Downloads section.

If during the template creation you specified "n" (no) to the VMware Tools configuration, you cannot use this feature.

You can set up the VMware Tools and install the full Open VMware Tools by using the following operating system command line:

```
tce-load -wi open-vm-tools.tcz
```

```
sudo touch /opt/bigFix/status/vmwareToolsReady
```

If you install the Open VMware Tools, you have the following capabilities:

- Automatic resizing of the desktop with the guest window including full screen support (reposition bar to upper left of desktop).
- Automatic mouse grab and ungrab.
- VMXnet3 network module instead of the PCNet32 module.
- Host shared folder access using HGFS (mounted at `/mnt/hgfs`).
- Time synchronization with the host.
- Copy/paste to/from host/guest.

Auto-Deployment

To instantiate the virtual relay instance by using the Auto-Deployment mode, you must copy the `network.conf` file into the `/opt/bigFix/config` directory of the instance that you are deploying.

Depending on the `network.conf` content, you enable the machine to start either in DHCP mode or with static IP settings.

If you want to use the machine with DHCP settings, insert in the `network.conf` file the following lines:

```
FQDN=virtualrelay.yourcompany.com
DHCP=y
```

If you want to use the machine with static IP settings, insert in the `network.conf` file the following lines:

```
FQDN=virtualrelay.yourcompany.com
DHCP=n
IPADDRESS=10.1.57.25
NETMASK=255.255.255.0
BROADCAST=10.1.57.255
GATEWAY=10.1.57.254
DNS1=10.1.57.100
```

Note: When creating the `network.conf` file, ensure that you follow the format of the provided samples. Do not leave any blank spaces in the lines that you insert.

Note: After creating the `network.conf` file, ensure that you convert it into a UNIX format, by using a tool such as the dos2unix utility.

Modifying the Virtual Relay template

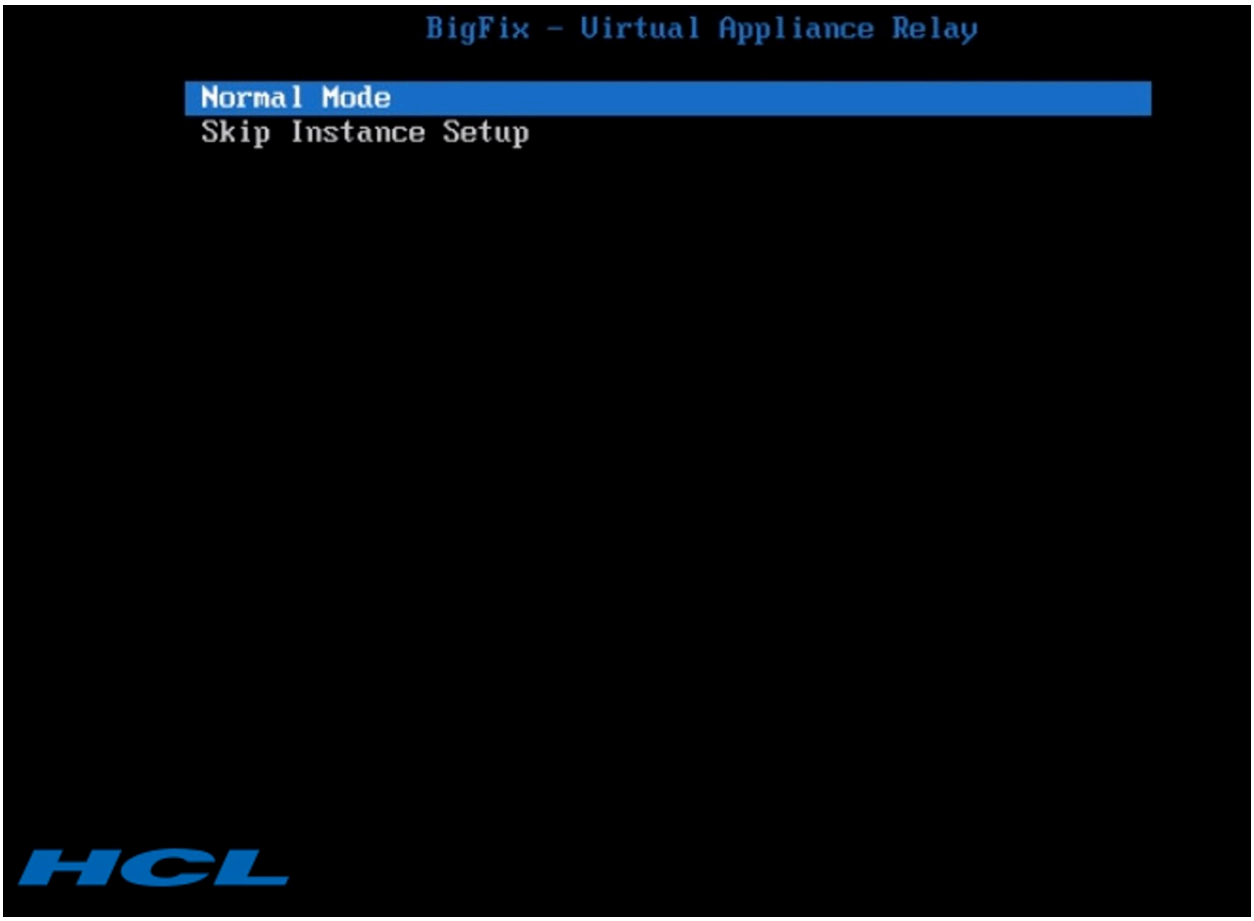
After you create the Virtual Relay template, you can deploy all the required Virtual Relay instances.

After the deployment, when pressing the space bar during the boot phase displayed by the following screen capture:



A menu opens and you can choose between two different options.

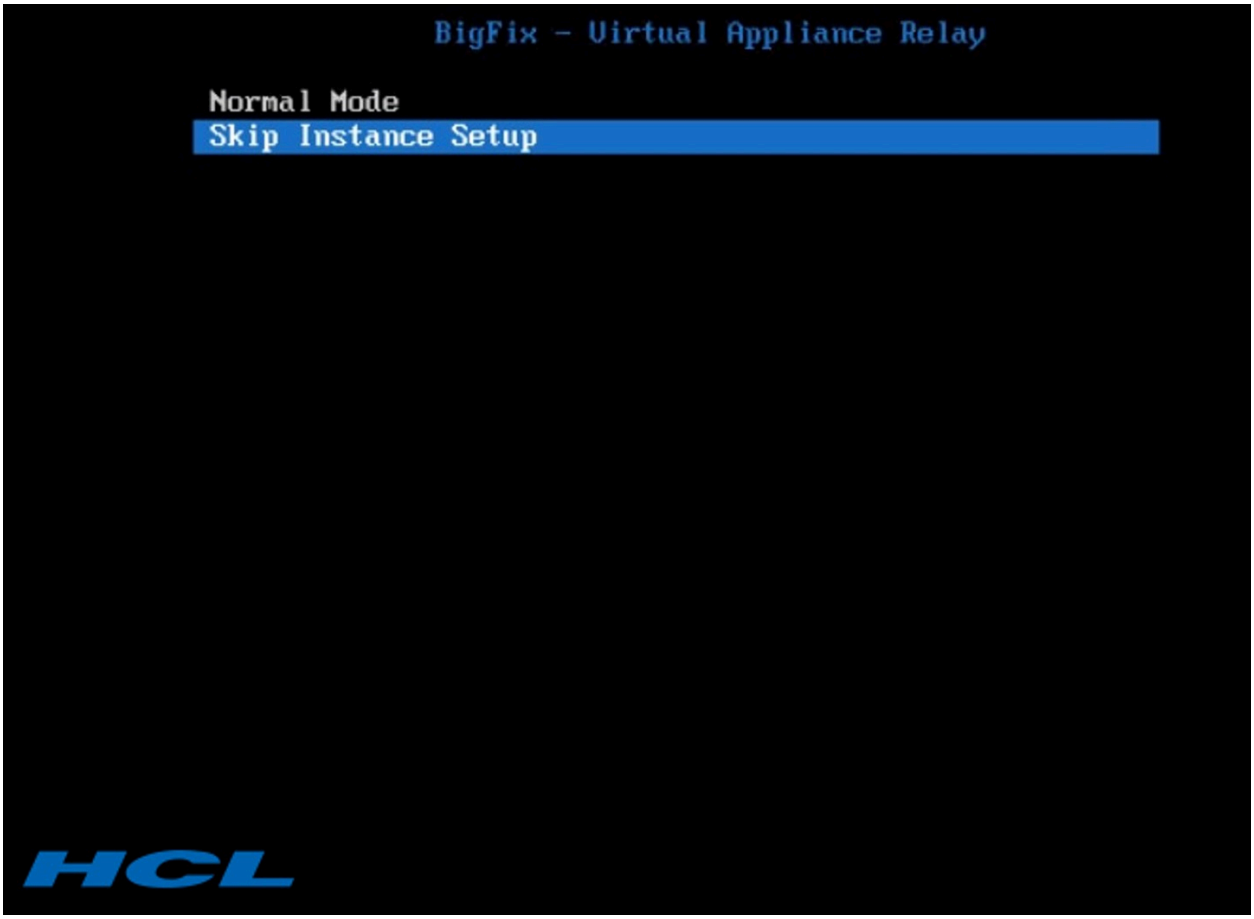
Normal Mode: allows you to continue with the instance setup.



Skip Instance Setup: allows you to skip the instance setup and add additional extensions to the existing instance that can, later on, be converted into a new template (as described in Creating the BigFix Virtual Relay template (on page 531)) according to your business needs and used for future instance deployments, or simply used for the single deployment of the current instance that you modified.

If you configured the template using the network configuration option, you can install additional extensions by using the initial network settings.

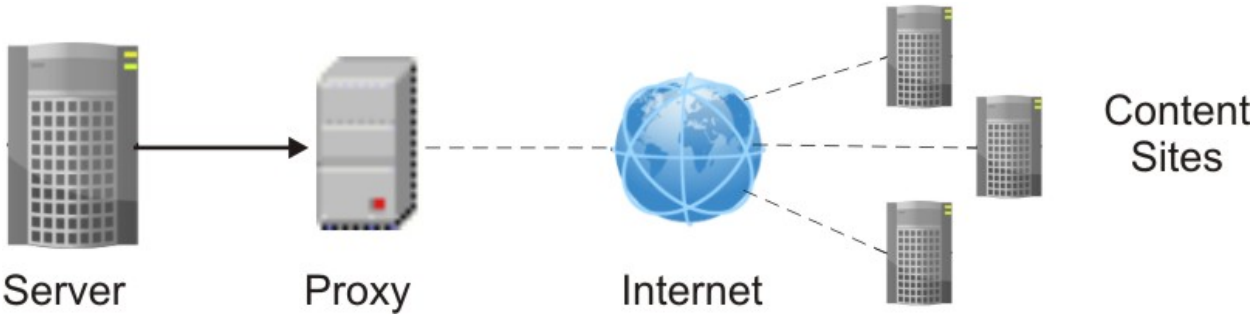
Otherwise, if you configured the template using the local folder option, you can install additional extensions by downloading these additional packages locally.



Chapter 16. Setting up a proxy connection

If your enterprise uses a proxy to access the Internet, your BigFix environment can use that communication path to gather content from sites.

In this case, you must configure the connection to the proxy on the BigFix server.



During a fresh installation, you are asked if you want to configure the communication through a proxy. The configuration settings that you enter are saved and used at run time to gather content from sites. For information about configuring a proxy connection at installation time, see Installing the Windows primary server (on page 143) for Windows systems, or Installing the Server (on page 204) for Linux systems.

To specify or modify the configuration for communicating with a proxy after installation, follow the instructions provided in Setting a proxy connection on the server *(on page 552)*.

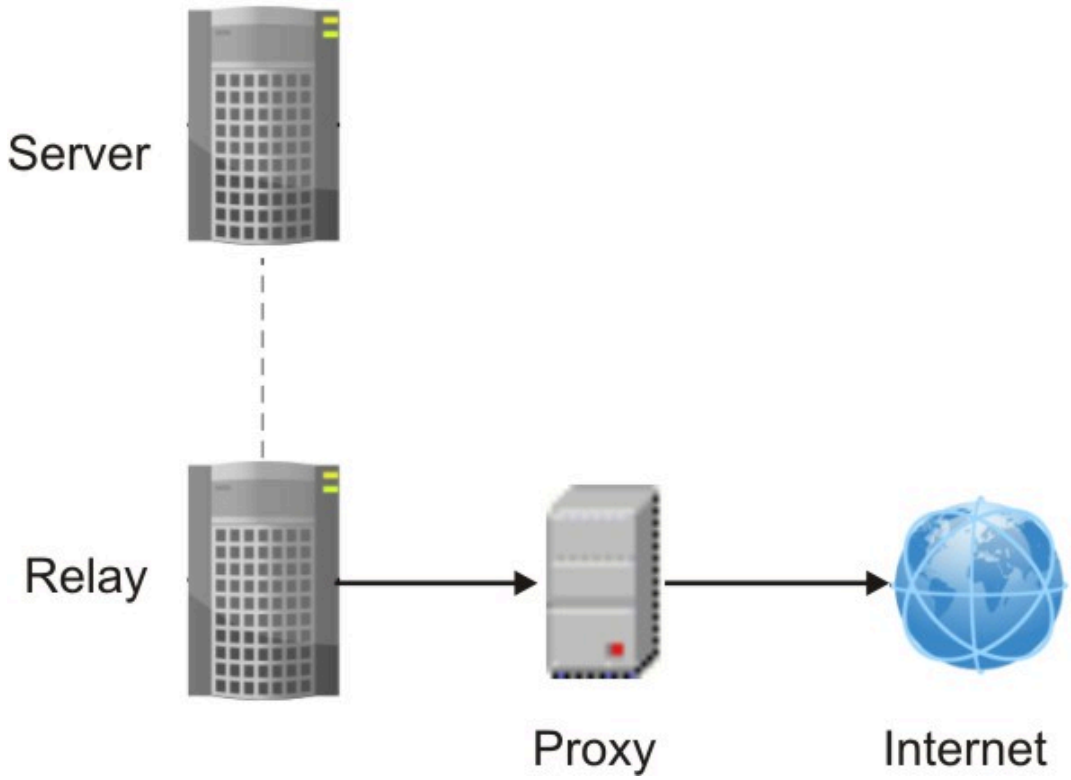
Important: If this configuration step is needed and you skip it, your environment will not work properly. A symptom of this misbehavior is that the site contents are not displayed on the console.

Note: You can also keep your system physical disconnected from the Internet by using an air-gapped implementation. For more information about this implementation, see Downloading files in air-gapped environments.

In addition to the gather process, the BigFix server or a relay can use the proxy connection to do component-to-component communication or to download files from the Internet.

The following list shows the most common proxy configurations that apply to a BigFix environment:

A relay connected to the Internet through a proxy to download files



To set this configuration on the relay:

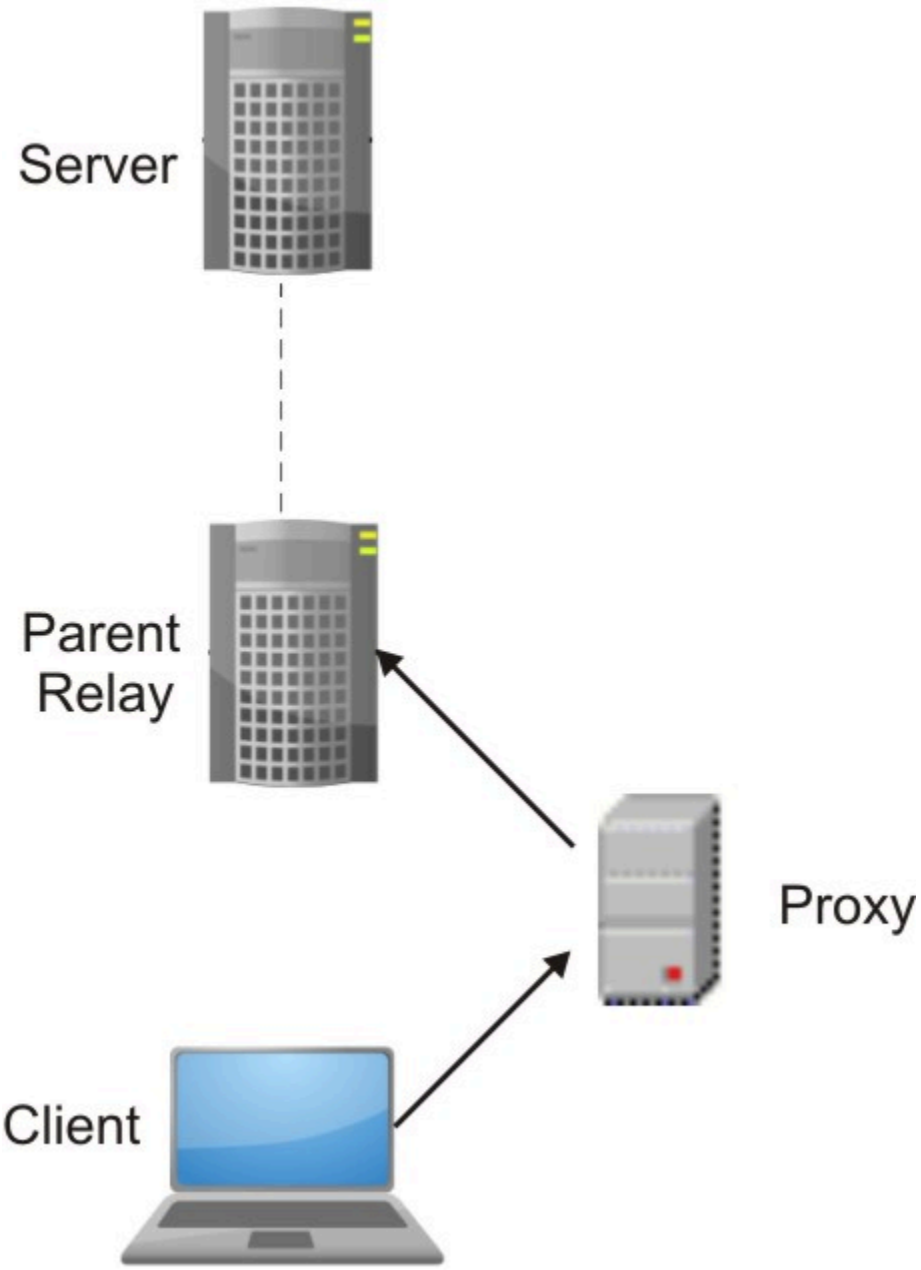
1. Run the steps that are described in Setting up a proxy connection on a relay *(on page 557)* to configure on the relay the communication to the proxy.
2. From the BigFix console set on the relay the following additional values to ensure that data is downloaded exclusively from the internet rather than from the parent relay:

```
_BESGather_Download_CheckParentFlag = 0
_BESGather_Download_CheckInternetFlag = 1
```

For more information about these configuration settings, see Gathering content.

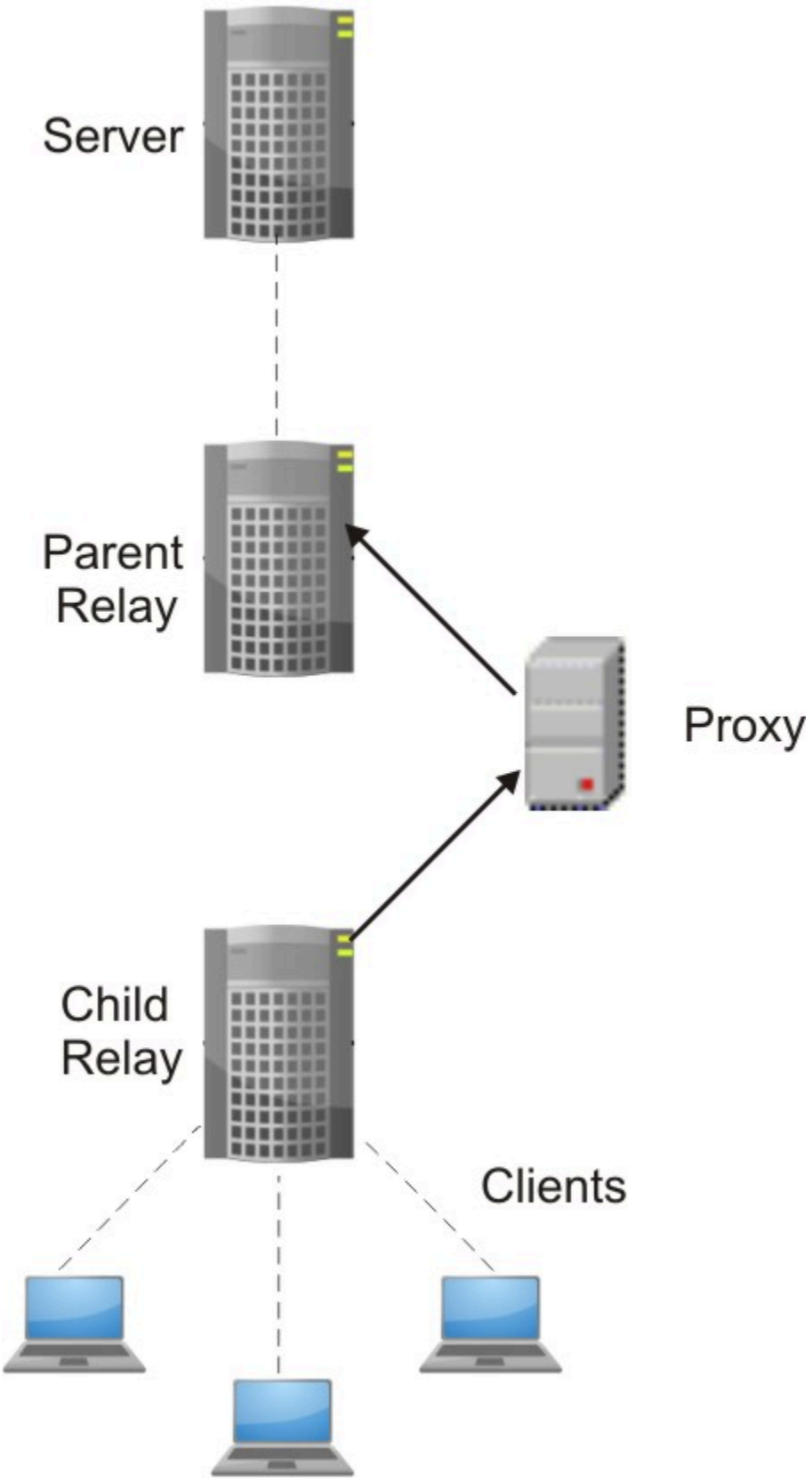
Note: To prevent communication from the relay to the server from going through the proxy, ensure that the proxy exception list is set on the relay as follows: `"127.0.0.1, localhost, <serverIP_addess>, yourdomain.com"`.

A client connected through a proxy to communicate with its parent relay



To set this configuration, on the client run the steps that are described in Setting up a proxy connection on a client *(on page 561)* and in Enabling client polling *(on page 550)*.

A relay connected through a proxy to communicate with a parent relay


Server
Parent Relay
Proxy
Child Relay
Clients

Complete these steps to implement this configuration:

- On the child relay run the steps that are described in Setting up a proxy connection on a relay *(on page 557)* and in Enabling client polling *(on page 550)*.
- Disable the relay notifier on the parent relay.

Enabling client polling

If a HTTP proxy exists between a parent relay and a client or child relay, or between the server and a child node, apply this workaround to bypass a limitation, that is caused by the proxy and that affects downstream communications.

An HTTP proxy does not forward the UDP protocol, which is the protocol used for sending notifications to clients in an BigFix environment. In such a configuration, the client on the child node must be able to ping and to query the relay on the parent node for new instructions.

To allow this behavior, complete the following configuration steps from the console on the client on the child node:

1. Open the console and go to the **Computer** section under the **All Content** domain.
2. Select the computer where the client is installed.
3. Right-click the computer and select **Edit Settings**.
4. Select **Add** to create a custom setting.
5. Enter the **Setting Name** and **Setting Value** as specified in the following table:

Table 7. Prerequisites to configure a proxy communication on a client

Setting	Description
`_BESClient_Comm_CommandPollEnable = 1`	Enables the client to poll its parent relay for new actions.

Setting	Description
`_BESClient_Comm_CommandPollIntervalSeconds = nnn`	Determines how often the client checks with its parent relay to gather or refresh content if `_BESClient_Comm_CommandPollEnable` is enabled. To prevent performance degradation, avoid specifying settings that are less than 900 seconds. Value range is 0-4294967295.
`__RelaySelect_Automatic = 0`	Specifies that the client is not configured for automatic parent relay selection. Clients that are configured for automatic parent relay selection cannot communicate through a proxy with their parent relay because they must be able to ping the relay.

6. Click **OK** to activate the settings.

Important: If you skip this step, the relays cannot communicate with child nodes through the proxy.

Connecting the console to the server through a proxy

If the BigFix console is installed on the same system as the BigFix server and if a proxy configuration is set in Internet Explorer, then ensure that, in that configuration, the **Bypass proxy server for local addresses** and the **Exceptions** settings are correctly specified.

If the BigFix console is installed on a system different from the system where the BigFix server is installed, then ensure that the proxy is correctly configured in Internet Explorer on the console system.

Setting a proxy connection on the server

When you install BigFix, you are asked if you want to set up communication with a proxy.

If you did not configure the connection to the proxy at installation time or if you want to modify the existing configuration, you can edit the proxy configuration settings after installation by running the following command:

On Windows systems:

```
%PROGRAM FILES%\BigFix Enterprise\BES Server\BESAdmin.exe /setpr
oxy
/sitePvkLocation=<path+license.pvk> /sitePvkPassword=<password>
[/proxy:<proxy_host>[:<proxy_port>]
[/user:<proxy_username> /pass:<proxy_password>]
[/delete]
[/exceptionlist:<proxy_exceptionlist>]
[/proxysecuretunnel:{false|true}]
[/proxyauthmethods:{basic|digest|negotiate|ntlm}]
[/proxydownstream:{false|true}]] |
[/delete]
```

On Linux systems:

```
/opt/BESServer/bin/BESAdmin.sh -setproxy
-sitePvkLocation=<path+license.pvk> -sitePvkPassword=<password>
[-proxy=<proxy_host>[:<proxy_port>]
[-user=<proxy_username> -pass=<proxy_password>]
[-exceptionlist=<proxy_exceptionlist>]
[-proxysecuretunnel={false|true}]
[-proxyauthmethods={basic|digest|negotiate|ntlm}]
[-proxydownstream={false|true}]] |
[-delete] |
[-display]
```

Note: The notation `<path+license.pvk>` used in the command syntax stands for *`path_to_license_file`*`/license.pvk`.

where you can specify the following keys:

proxy

It sets the host name or IP address and, optionally, the port number of the proxy machine. By default the value of *proxy_port* is 80.

user

It sets the user name that is used to authenticate with the proxy, if the proxy requires authentication.
If you installed your BigFix server on a Windows system and your proxy requires Kerberos Authentication, use the format *user@mydomain.com*.

Note: The Kerberos Authentication is supported only on Windows systems. This authentication method is not supported if you installed your BigFix server on a Linux system.

If your proxy requires NTLM Authentication, specify the NTLM user.

If your proxy requires the realm name notation, specify the *proxy_user* as *user@mydomain.com* or *mydomain\user*.

Note: The Linux shell manages the back slash "\" as an escape character. Specify either *mydomain\\user* or *"mydomain\user"* to use the notation *mydomain\user* if you run the command in a Linux shell.

pass

It sets the password that is used to authenticate with the proxy, if the proxy requires authentication. The value that is assigned to the password is stored encrypted.

delete

If specified, it deletes all the settings defined in BigFix for communicating with the specified proxy.

display

If specified, it displays the proxy communication settings defined in BigFix. This argument applies only to Linux systems.

exceptionlist

If set, it is a comma-separated list of computers and domains that must be reached without passing through the proxy. In this syntax blank spaces have no influence. Each name in this list is matched as either a domain, which contains the hostname, or the hostname itself. For example, `yourdomain.com` would match `yourdomain.com`, `yourdomain.com:80`, and `www.yourdomain.com`, but not `www.notyourdomain.com`. You can assign the following sample values to `<proxy_exceptionlist>`:

```
localhost,127.0.0.1, yourdomain.com
localhost,127.0.0.1,yourdomain.com,8.168.117.65
"localhost,127.0.0.1, yourdomain.com, 8.168.117.65"
```

By default, if you do not specify the **exceptionlist** setting, BigFix prevents diverting internal communications from being diverted towards the proxy. This is equivalent to setting **exceptionlist:localhost,127.0.0.1**. To maintain this behavior, ensure that you add `localhost, 127.0.0.1` to the list of exceptions when specifying the **exceptionlist** setting.

proxysecuretunnel

If set, it defines whether or not the proxy is enforced to attempt tunneling. By default the proxy does not attempt tunneling.

proxyauthmethods

If set, it restricts the set of authentication methods that can be used. You can specify more than one value separated by a comma, for example:

```
proxyauthmethods:basic,ntlm
```

By default there is no restriction for the authentication method. The proxy chooses which authentication method must be used.



Note: If you specify to use the `negotiate` authentication method on a Linux server or relay, a different authentication method might be used.



Note:

If you want to enable FIPS mode, ensure that the proxy configuration uses:

- An authentication method other than `digest` on Windows systems.
- An authentication method other than `digest`, `negotiate` or `ntlm` on Linux systems.

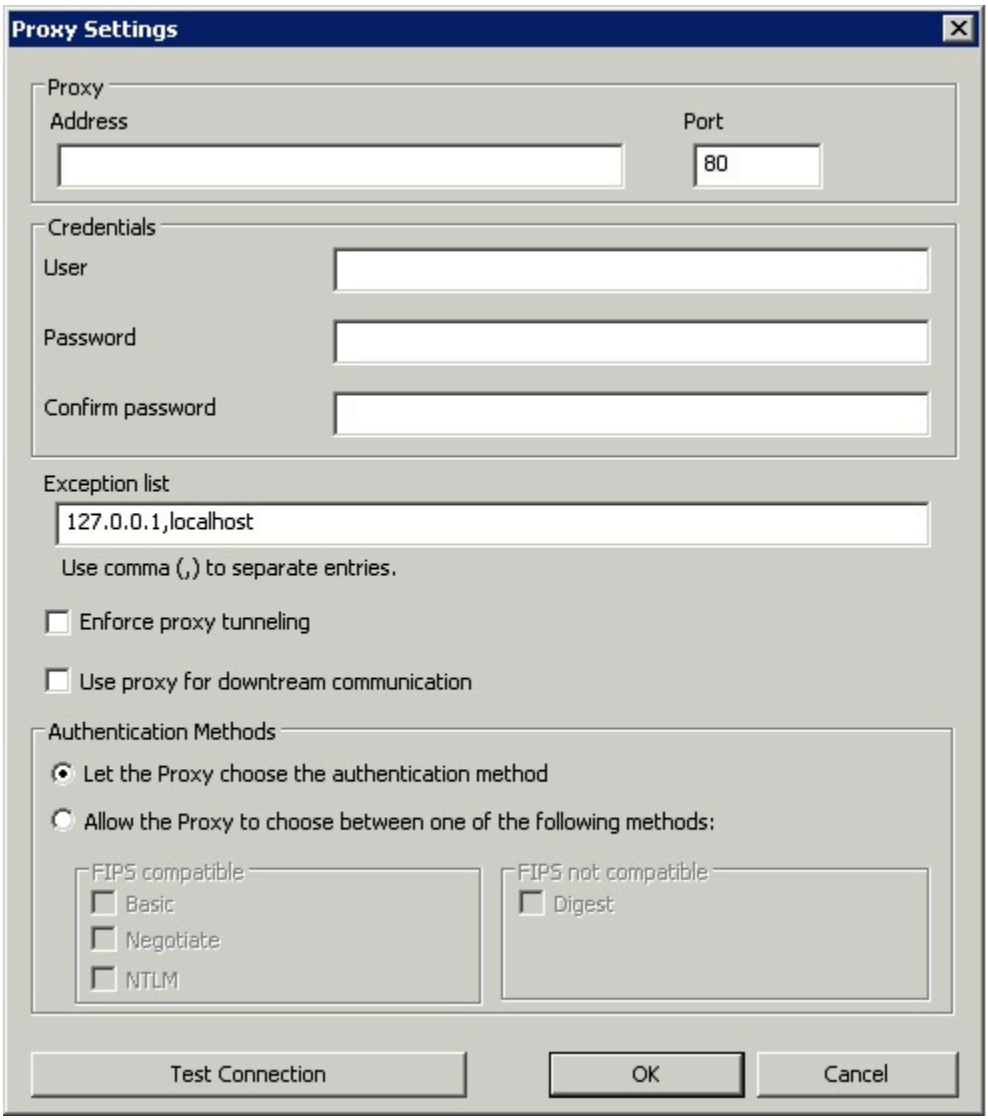
proxydownstream

If set to **true**, this setting indicates that all HTTP communications in your BigFix environment also pass through the proxy. If you do not specify this setting, by default the value **false** is assumed.



Note: If you migrate an existing BigFix proxy configuration and the **_Enterprise Server _ClientRegister _Proxy*** keys are specified, by default `proxydownstream` is set to **true**.

On Windows servers the command `BESAdmin.exe /setproxy` opens the Proxy settings panel filled in the current proxy settings.



The same panel is displayed whenever you run the `BESAdmin.exe` command to set one or more specific proxy settings. Check that the values displayed are correct, modify them if necessary and then click **OK** to confirm the changes.
The proxy configuration settings are stored:

On Windows systems:

In the registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\EnterpriseClient\Settings\Client\`.

On Linux systems:

In the following sections of the `besserver.config` file:

- `[SOFTWARE\BigFix\EnterpriseClient\Settings\Client\_Enterprise Server_ClientRegister_ProxyServer]`
- `[SOFTWARE\BigFix\EnterpriseClient\Settings\Client\_Enterprise Server_ClientRegister_ProxyPort]`
- `[SOFTWARE\BigFix\EnterpriseClient\Settings\Client\_Enterprise Server_ClientRegister_ProxyUser]`
- `[SOFTWARE\BigFix\EnterpriseClient\Settings\Client\_Enterprise Server_ClientRegister_ProxyPass]`

Important: Whenever you run the `BESAdmin` command to define a proxy setting, ensure that you specify all not default setting previously defined otherwise they will be set to blank. This behavior applies to both Windows and Linux systems.

Note: Ensure that you use the **Edit Settings** dialog box on the BigFix Console to update any proxy values that you set through the **Edit Settings** dialog box.

Note: If a HTTP proxy exists between the server and a child node, ensure that you follow the instructions provided in Enabling client polling *(on page 550)* to enable downstream communications.

Note: The BES components that access the internet run, by default, as SYSTEM account on the Windows systems and as root on the Linux systems.

For additional configuration settings that you can use to configure your BigFix environment, see List of settings and detailed descriptions.

Setting up a proxy connection on a relay

Complete the following steps to allow the relay to communicate with its parent components.

1. Open the console and go to the **Computer** section under the **All Content** domain.
2. Select the computer where the relay is installed.
3. Right-click the computer and select **Edit Settings**.
4. Select **Add** to create custom settings.
5. Enter the **Setting Name** and **Setting Value** listed in the following table:

Table 8. Proxy configuration settings for server and relays

Setting Name	Setting Value	Details
`_Enterprise Server _ClientRegister _ProxyServer`	Sets the hostname that is used to reach the proxy.	Default Value: None Type: String Value Range: N/A Mandatory: No
`_Enterprise Server _ClientRegister _ProxyPort`	Sets the port that is used by the proxy server.	Default Value: None Type: Numeric Value Range: N/A Mandatory: No
`_Enterprise Server _ClientRegister _ProxyUser`	Sets the user name that is used to authenticate with the proxy if the proxy requires authentication.	Default Value: None Type: String Value Range: N/A Mandatory: No (depending on the authentication method)
`_Enterprise Server _ClientRegister _ProxyPass`	Sets the password that is used to authenticate with the proxy if the proxy requires authentication.	Default Value: None Type: String Value Range: N/A Mandatory: No (depending

Setting Name	Setting Value	Details
		on the authentication method)
`_Enterprise Server _ClientRegister _ProxySecureTunnel`	If set, it defines whether or not the proxy is enforced to attempt tunneling. By default the proxy does not attempt tunneling.	Default Value: false Type: Boolean Value Range: 0 | 1 Mandatory: No
`_Enterprise Server _ClientRegister _ProxyAuthMethods Allowed`	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 connection on the server *(on page 552)*.	Default Value: None (Any) Type: String Value Range: basic|digest|negotiate|ntlm Mandatory: No
`_Enterprise Server _ClientRegister _ProxyUseForDownstreamComm`	If set to **1**, this setting indicates that all downstream communications in your environment pass through the proxy.	Default Value: 0 Type: Numeric Value Range: 0 | 1 Mandatory: No
`_Enterprise Server _ClientRegister`	Specifies the computers, for example the parent relay, domains and subnetworks that must be reached by the relay without pass-	Default Value: localhost, 127.0.0.1 (internal communications

Setting Name	Setting Value	Details
`_ProxyExceptionList`	ing through the proxy. Use the following format: `"localhost, 127.0.0.1, hostname1, hostname2, IP_Addr_A, IP_Addr_B, domain_Z, domain_Y, ..."` By default internal communications 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 *(on page 545)* to learn more about using the proxy exception list on a relay thru the samples.	are not diverted towards the proxy) Type: String Value Range: N/A Mandatory: No

For more information about how to specify the proxy settings, see Setting a proxy connection on the server (on page 552).

6. Click **OK** to send activate the configuration settings.

Note: If a HTTP proxy exists between the relay and a client or a child relay, ensure that you follow the instructions provided in Enabling client polling *(on page 550)* to enable downstream communications.

Complete these additional steps if you want to allow your relay to download files from the internet through the proxy:

1. Open the console and go to the **Computer** section under the **All Content** domain.
2. Select the computer where the relay is installed.
3. Right-click the computer and select **Edit Settings**.
4. Select **Add** to create the following custom settings:

```
_BESGather_Download_CheckInternetFlag = 1
_BESGather_Download_CheckParentFlag = 0
```

5. Click **OK** to activate the configuration settings.

For additional configuration settings that you can use to configure your BigFix environment, see List of settings and detailed descriptions.

Setting up a proxy connection on a client

Set the proxy connection on the client.

Complete the following steps:

1. Open the console and go to the **Computer** section under the **All Content** domain.
2. Select the computer where the client is installed.
3. Right-click the computer and select **Edit Settings**.
4. Select **Add** to create a custom setting.
5. Enter the **Setting Name** and **Setting Value** as described in the table contained in the next section.
6. Click **OK** to activate the settings.

Table 9. Proxy client configuration settings

Setting Name	Setting Value	Details
`_BESClient_Comm_ProxyServer`	Sets the hostname that is used to reach the proxy.	Default Value: None Type: String

Table 9. Proxy client configuration settings (continued)

Setting Name	Setting Value	Details
		Value Range: N/A Mandatory: Yes
`_BESClient_Comm_ProxyPort`	Sets the port that is used to communicate with the proxy.	Default Value: None Type: Numeric Value Range: N/A Mandatory: No
`_BESClient_Comm_ProxyUser`	Sets the user name that is used to authenticate with the proxy if the proxy requires authentication.	Default Value: None Type: String Value Range: N/A Mandatory: No (depending on the authentication method)
`_BESClient_Comm_ProxyPass`	Sets the password that is used to authenticate with the proxy if the proxy requires authentication.	Default Value: None Type: String Value Range: N/A Mandatory: No (depending on the authentication method)
`_BESClient_Comm_ProxyManualTryDirect`	Specifies whether direct connections can be used. 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:	Default Value: 2 Type: Numeric Value Range: 0-2 Mandatory: No

Table 9. Proxy client configuration settings (continued)

Setting Name	Setting Value	Details
	0 Do not try direct connection.	
	1 Try direct connection if a proxy connection cannot be established.	
	2 Try direct connection first.	

Note: On Linux systems, at run time, the BigFix searches and, if specified, uses the configuration stored in the client configuration file. If the requested configuration is not specified in the client configuration file, the product searches for it in the server configuration file, or in the relay configuration file. Consider this behavior when defining the proxy configuration on the BigFix server or relay.

Note:

The BigFix client allows using the proxy basic authentication on any platform. While the following authentications can be used on Windows platforms only:

- Digest
- Negotiate
- NTLM

Note:

When the connection to the relay succeeds, the resulting proxy is locked in for subsequent communications and the values for the proxy server and proxy port are saved as AutoProxyServer and AutoProxyPort in the Global section of the client settings.

If the client is installed on a Windows system where Internet Settings are configured to use a proxy, then, by default, BigFix uses the Internet Settings configuration to communicate with the proxy. The following table shows the additional settings and behaviors that you can optionally specify on Windows platform:

Table 10. Proxy client additional configuration settings on Windows systems

Setting Name	Setting Value	Details
`_BESClient_Comm _ProxyAutoDetect`	Specifies whether the system uses the proxy configuration settings that are specified for Internet Settings. The following values are available: **0** Use the values that are specified in `_BESClient_Comm _ProxyServer` and `_BESClient_Comm_ ProxyPort`. **1** Use the Internet Settings configuration. **Important:** Ensure that at least one user is logged in to the	Default Value: 0 Type: Boolean Value Range: 0-1 Mandatory: No

Table 10. Proxy client additional configuration settings on Windows systems (continued)

Setting Name	Setting Value	Details
	client to be able to get the Internet Settings configuration.	
`_BESClient_Comm_ProxyAutoDetectTryDirect`	Specifies whether direct connections can be used when the system uses the proxy configuration settings that are specified for Internet Settings. This setting is valid only if `_BESClient_Comm_ProxyAutoDetect = 1`. The following values are available: **0** Do not try direct connection. **1** Try direct connection if a proxy connection cannot be established. **2** Try direct connection first.	Default Value: 1 Type: Numeric Value Range: 0-2 Mandatory: No

For additional configuration settings that you can use to configure your BigFix environment, see List of settings and detailed descriptions.

Best practices to consider when defining a proxy connection

Consider the following tips and tricks to avoid common problems.

- After you set the communication through the proxy on a Windows server, use the BigFix Diagnostic tools to verify that the server can successfully reach the Internet.
- Check the `GatherDB.log` file that is in the `BES Server\GatherDBData` folder to verify that the server can gather data from the Internet.
- Check in the firewall rules if any file types are blocked. In this case, if the content to gather from a site contains at least one file with this file type, then the entire content of that site is not gathered.
- Ensure that the password specified in `ProxyPass` on the server, or in `_Enterprise Server_ClientRegister_ProxyPass` on the client or relay did not expire.
- Make sure that the proxy allows the downloading of arbitrary files from the Internet (for example, it does not block .exe downloads or does not block files with unknown extensions).
- Most of the files in BigFix are downloaded from `bigfix.com` or `microsoft.com` using HTTP port 80. However, it is recommended that you allow the proxy service to download from any location using HTTP, HTTPS, or FTP because some downloads might use these protocols.
- Make sure that the proxy is bypassed for internal network and component-to-component communications because it might cause problems with how the BigFix server works and is inefficient for the proxy. Use the `ProxyExceptionList` setting, if needed, to exclude local systems from the communication through the proxy.
- The setting `ProxyExceptionList` was introduced in BigFix version 9.0.835.0 for Windows and Linux systems. If you are using BigFix version 9.0 and you have problems using content that downloads files from the local server, upgrade to BigFix version 9.0.835.0 or later.
- On the BigFix server installed on a Linux system, at runtime the client configuration file is read before the server configuration file. Ensure that you update common settings on both components to avoid conflicts.
- By default the HTTP and HTTPS connections time out after 10 seconds, DNS resolution time included. When this happens the HTTP 28 error is logged. In your environment, if the proxy server or the DNS server takes a longer time to establish the TCP connection, you can increase the number of seconds before the connection times out by editing the setting `_HTTPRequestSender_Connect_TimeoutSeconds`.

The `_HTTPRequestSender_Connect_TimeoutSeconds` setting affects all the BigFix components, including the Console and the Client, running on the machine for which this setting is set. No other BigFix component running on other machines in the deployment is affected by the setting. As a best practice, be careful when increasing the value of this setting and try to keep it as low as possible to avoid opening too many sockets concurrently risking socket exhaustion and eventual loss of service.

Troubleshooting proxy connection

Common problems you might find with a proxy connection.

Setting up a proxy connection to have your environment work properly is not an easy task, and you might have problems with your configuration. This is a list of the most common issues along with their solutions.

Error `Unable to get site content (failed to pass sha1 hash value checks` in `BESRelay.log`

Error in getting site content.

This error is a symptom that something is interfering with the process of downloading external Fixlet sites, for example patches for Windows. You might notice that you do not gather the latest version of external content sites you are subscribed to, or that you do not gather at all from sites whose name is listed in the License overview dashboard. In this instance, the error `Unable to get site content (failed to pass sha1 hash value checks)` is listed in the `BesRelay.log` file; this file is located in:

Windows operating systems

- 32bit systems: `C:\Program Files\BigFix Enterprise\BES Server`
- 64bit systems: `C:\Program Files (x86)\BigFix Enterprise\BES Server`

Linux operating systems

`/var/log/BESRelay.log`

The reason is often a firewall or a web proxy that is filtering the content exchanged between the BigFix server and the sync.bigfix.com server.

To fix the problem, ask the system administrator to allow downloading traffic for the BES Gather service on port 80. If the problem persists, complete a traffic packet analysis to find the problem. An useful tool to perform a full network protocol analysis is Wireshark.

Error `Unexpected HTTP response: 503 Service Unavailable` in GatherDB.log

Error for service unavailability.

The BigFix server communication with the Gather database is performed through the IP address 127.0.0.1 and the database does not need a proxy configuration. In the event the error message `Unexpected HTTP response: 503 Service Unavailable` is repeatedly listed in the `GatherDB.log` file, you must check that the environment variable HTTP_PROXY has not been set if it is not used in your environment, or you must set the environment variable NO_PROXY to the value 127.0.0.1. After completing these changes, restart the `BESGatherDB` service.

The log file `GatherDB.log` is located in:

Windows operating systems

- 32bit systems: `C:\Program Files\BigFix Enterprise\BES Server\GatherDBData`
- 64bit systems: `C:\Program Files (x86)\BigFix Enterprise\BES Server\GatherDBData`

Linux operating systems

`/var/opt/BESServer/GatherDBData`

How to check if the proxy configuration is correct

Checks for proxy configuration.

In a command prompt of the BigFix server, run the following command:

```
curl -x <proxy hostname>:<proxy port> --proxy-user <username>:<password>
http://sync.bigfix.com/cgi-bin/bfgather/bessupport
```

where:

- `<proxy hostname>:<proxy port>` are the host name or IP address and the port number of the proxy server.
- `<username>:<password>` are the username and password used to authenticate to the proxy server.

If you get a result similar to the following:

```
MIME-Version: 1.0
Content-Type: multipart/signed; protocol="application/x-pkcs7-signature";
micalg="sha-256,sha1"; boundary="----B64F18ABD54D2355B260FA851C81B467"

This is an S/MIME signed message

------B64F18ABD54D2355B260FA851C81B467
MIME-Version: 1.0
Content-Type: multipart/x-directory2;
 boundary="=DkNtJtPxyvgg7l(ktApiDsZBnMvYtSm"
FullSiteURL: http://sync.bigfix.com/bfsites/bessupport_1369/__fullsite
FullSiteURLSize: 1507388
SiteDiffBaseURL: http://sync.bigfix.com/bfsites/bessupport_1369/__diffsite
SiteDiffList: 1111111111
Version: 1369



--=DkNtJtPxyvgg7l(ktApiDsZBnMvYtSm

URL: http://sync.bigfix.com/bfsites/bessupport_1369/1Common%20Tasks.fxf
NAME: 1Common%20Tasks.fxf
MODIFIED: Fri, 05 May 2017 22:12:57 +0000
```

```
SIZE: 258246
TYPE: FILE
HASH: 6882d98104c25cd3c27fc343d8cfb73a09c30945
HASHINFO:
 sha256,055b37a65160db26396e72488bd021ffbe6d06ddc19e0c494f5f0cbf73ea8cd1
```

your proxy configuration is correct. If not, ask your system administrator to check why the BigFix synchronization site cannot be accessed.

New site version cannot be gathered

Errors in gathering site version.

If you get the following errors when running the BigFix diagnostic tool:

```
Test failed: The Client Plug-in registration
Reason: Unexpected HTTP response: 504 Proxy Error: Unable to connect to
remote host "127.0.0.1:52311" or host not responding - URL
"http://127.0.0.1:52311/cgi-bin/bfenterprise/clientregister.exe?RequestType
 =
version"(http://127.0.0.1:52311/cgi-bin/bfenterprise/clientregister.exe
?RequestTy pe=version%27),
errno: 111 HTTP request failed with an error.

Test failed: Plug-post results
Reason: Unexpected HTTP response: 504 Proxy Error: Unable to connect to
 remote
host "127.0.0.1:52311" or host not responding - URL
"http://127.0.0.1:52311/cgi-bin/bfenterprise/PostResults.exe"
(http://127.0.0.1:52311/cgi-bin/bfenterprise/PostResults.exe%27),
errno: 111 HTTP request failed with an error.

Test failed: BESGatherMirror plug-in
Reason: Unexpected HTTP response: 504 Proxy Error: Unable to connect to
remote host "127.0.0.1:52311" or host not responding - URL
```

```
"http://127.0.0.1:52311/cgi-bin/bfenterprise/BESGatherMirror.exe/-version]
"([http://127.0.0.1:52311/cgi-bin/bfenterprise/BESGatherMirror.exe/-versio
n%27),
errno: 111 HTTP request failed with an error.
```

you do not gather the new site version. To solve the problem, complete the following steps:

1. Perform the steps needed to set up a proxy connection on the server, as described in Setting a proxy connection on the server *(on page 552)*, and be sure to add the following values in the exception list: `localhost, 127.0.0.1, <server_IP>, <server_hostname>`.
2. Check that the environment variable `http_proxy` is not configured. If it is, remove it and restart your operating system.

Chapter 17. Running backup and restore

You can schedule periodic backups (typically nightly) of the BigFix server and database files, to reduce the risk of losing productivity or data when a problem occurs by restoring the latest backup.

Consider, however, that when you run a disaster recovery, you restore a backup of an earlier working state of BigFix on the server computer or another computer. Depending on how old the backup is you can lose the latest changes or data.

Important:

After restoring the data from the last backup, the BigFix server might restart at an earlier state with a disalignment between its mailbox and that of each relay. In this case the BigFix server needs to resynchronize with the relays that have continued to process requests, otherwise the relays might ignore the requests of the server. To realign the mailboxes, see How to realign mailboxes after restoring a BigFix database backup.

Moreover, in a Windows environment, any configuration involving registry keys is neither saved nor restored. To recover these values, you must restore them after the recovery procedure successfully completes by running the appropriate configuration processes. For example, email server settings must be set up again on recovered Web Reports.

You can also restore a single BigFix DSA server when an unrecoverable failure occurs.

Note: Do not restore the failed DSA server entirely from backup. Due to the complexity of DSA replication we recommend that you install a new server with the same FQDN and follow these procedures: DSA Recovey on Windows *(on page 580)* and DSA Recovey on Linux *(on page 594)*.

If all DSA servers are lost, follow the BigFix server restore procedure, Server Recovery Procedure on Windows (on page 576) and Server Recovery Procedure on Linux.

On Windows systems

If you back up the database and the BigFix Server files, when needed you can restore the BigFix environment on a Windows computer.

Server Backup

How to back up the BigFix Server on Windows.

Consider backing up to a remote system to allow for higher fault tolerance. Perform the following steps:

1. Stop all the BigFix services, including running plug-ins if any. The main BigFix services should be stopped in the following order.

   ```
   BES WebUI
   BES Web Reports Server
   BES Client
   BES GatherDB
   BES FillDB
   BES Root Server
   ```

2. If you have remote processes that may access the BigFix databases, like a remote BigFix WebUI, stop them as well.
3. Back up the `BFEnterprise` and `BESReporting` databases.

   For example, you can use SQL Server Management Studio to connect to the instance hosting your BigFix databases and run the following commands:

   ```
   BACKUP DATABASE BFEnterprise
   TO DISK = 'C:\Path\To\ExistingFolder\BFEnterprise.bak';
   GO
   BACKUP DATABASE BESReporting
   TO DISK = 'C:\Path\To\ExistingFolder\BESReporting.bak';
   GO
   ```

4. Back up the following files and folders used by the BigFix Server and Web Reports:

- `[BigFix Server folder]\BESReportsData`. Web Reports data folder.
- `[BigFix Server folder]\BESReportsServer\wwwroot\ReportFiles`. Support files for Web Reports custom reports.
- `[BigFix Server folder]\Encryption Keys`. Private encryption keys (if using Message Level Encryption).
- `[BigFix Server folder]\Mirror Server\Inbox`. Information for BigFix Clients to get actions and Fixlets.
- `[BigFix Server folder]\Mirror Server\Config\DownloadWhitelist.txt`. White list for BigFix dynamic download.
- `[BigFix Server folder]\UploadManagerData`. Files uploaded by the BigFix Clients.
- `[BigFix Server folder]\wwwrootbes`. Various information about actions, Fixlets, uploads and downloads.

`[BigFix Server folder]` is the BigFix Server installation path, by default `C:\Program Files (x86)\BigFix Enterprise\BES Server`. The locations of your BigFix Server `wwwroot` folder and of your Web Reports `wwwroot` folder might be different if you customized them at installation time.

5. Back up your license key, license certificate, and masthead file.

   The `license.pvk` and `license.crt` files are critical to the security and operation of BigFix. If the private key (`pvk`) files are lost, they cannot be recovered.

   The masthead (`afxm`) file is an important file that contains information about the BigFix Server configuration. To back it up, you can copy the `[BigFix Server folder]\actionsite.afxm` file and rename it to `masthead.afxm`.

6. Decrypt and save the encrypted keys.

   The encrypted keys are files named like `Encrypted...Key` and, by default, they are located in the `[BigFix Server folder]`. Depending on the version of the BigFix Server and on what BigFix components are installed on you environment, you will have a slightly different set of encrypted keys. The possible key files are:

   - `EncryptedAPIServerKey`
   - `EncryptedClientCAKey`
   - `EncryptedCustomClientCAKey`

- `EncryptedCustomServerTLSKey`
- `EncryptedExplorerCAKey`
- `EncryptedPlatKey`
- `EncryptedServerSigningKey`
- `EncryptedWebUICAKey`

Use the ServerKeyTool.exe tool and run the steps documented in this page to decrypt the keys. For example:

```
ServerKeyTool.exe /decrypt
    /dirIn:"C:\Path\To\BigFix Server folder"
    /dirOut:"C:\Path\To\Backup folder"
    /sitePvkLocation:"C:\Path\To\license.pvk"
    /sitePvkPassword:"*****"
```

All the existing encrypted keys stored in the input folder are backed up at once, and the files containing the decrypted keys are stored in the specified destination folder with the filename prefix `Decrypted*`.

7. Analyze the BigFix Server database tables named `DBINFO` and `REPLICATION_SERVERS`.

```
SELECT * FROM [BFEnterprise].[dbo].[DBINFO];
SELECT * FROM [BFEnterprise].[dbo].[REPLICATION_SERVERS];
```

Record all the rows in those tables for verification purposes.

If DNS aliases are being used instead of the actual hostnames of your BigFix Server computers, the field values of those tables should remain the same after a restore procedure. If they contain hostnames, and the hostnames of your BigFix Server computers change, those fields may need manual modification after the restore. If you want to update the CN on the BigFix internal certificates, see How to change the Common Name (CN) on BigFix internal certificates.

8. To avoid the duplication of the BigFix Server computer entry on the BigFix Console, run the steps described in the **Client data backup** section in Avoiding duplicates when a Client is restored *(on page 346)*.
9. Any configuration involving registry keys is neither saved nor restored. After completing a restore procedure, you will have to perform those configurations again.

You should only edit a BigFix registry key if a documented configuration step explicitly calls for it. However, for verification purposes, you should export the registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix`.

Server Recovery

How to restore a BigFix Server on Windows.

1. Ensure that the new BigFix Server computer can be reached on the network using the same URL that is in the masthead file. For example, if you look at the field `X-Fixlet-Site-Gather-URL`, it contains something like `http://192.168.10.32:52311/cgi-bin/bfgather.exe/actionsite` or `http://bigfixserver.company.com:52311/cgi-bin/bfgather.exe/actionsite`.

   Note: To avoid issues when the BigFix clients connect to the BigFix Server before it is fully restored, ensure that it is not available on the network until the recovery is complete.

2. When running the recovery procedure on the same computer that the BigFix Server was installed on, you must first run the `BESRemove` utility and completely remove all BigFix files and settings from that computer. For a detailed guide about removing the BigFix components on Windows, see Removing the BigFix components from Windows *(on page 177)*. If you are performing a full restore, remember to remove the BigFix databases too.
3. If you need to reinstall SQL Server as well, install the same version that you created your database backups with. Remember to enable Mixed Mode Authentication if your previous BigFix Server was using it.
4. Restore the `BFEnterprise` and `BESReporting` databases from your backups.

   You can use SQL Server Management Studio to connect to the instance where you want to restore your BigFix databases and run the following commands:

   ```
   RESTORE DATABASE BFEnterprise
   FROM DISK = 'C:\Path\To\BFEnterprise.bak';
   GO
   ```

```
RESTORE DATABASE BESReporting
FROM DISK = 'C:\Path\To\BESReporting.bak';
GO
```

5. Restore the backed up files and folders, creating the directory structure.
6. Encrypt they key files you decrypted during the backup procedure.

   Use the ServerKeyTool.exe tool and run the steps documented in this page to encrypt the keys. For example:

   ```
   ServerKeyTool.exe /encrypt
       /dirIn:"C:\Path\To\Backup folder"
       /dirOut:"C:\Path\To\BigFix Server folder"
       /sitePvkLocation:"C:\Path\To\license.pvk"
       /sitePvkPassword:"*****"
   ```

   All the decrypted keys are encrypted at once, and the files containing the encrypted keys are stored in the specified destination folder with the filename prefix `Encrypted*`.
7. Go to the BigFix Download Center Platform Release page, find the same patch level of the original installation and go to the corresponding download page. Then download the "Installer Generator" setup file.
8. Run the BigFix Installation Generator on the computer where you want to restore the BigFix Server.
9. When you reach the "Setup Type" dialog, select the option "I want to install with an existing masthead" and choose the Server masthead you previously backed up.

   Do **not** re-use the folders of a previous Installation Generator installation, they likely contain an outdated masthead file.
10. Run the BigFix Installation Guide and, from there, start the BigFix Server installer.
11. In the "License credentials" dialog, make sure that the selected "Masthead file" is the one in the Installation Generator folder or select the masthead you backed up.
12. In the "Select Database Replication" dialog

- select the "Single or Master Database" option if you are restoring your only BigFix Server or the Primary/Master server of a DSA environment
- select the "Replicated Database" option if you are restoring the Secondary/ Replica server of a DSA environment

13. In the "BigFix Master Operator Credentials" dialog, in the "Username" field, specify the same BigFix master operator that you used for the installation you backed up.

    Note: If, later on during the installation, you get a pop-up message saying that "The user name *MyUserName* already exists in the database", you can ignore it.

14. In the "Choose Destination Location" dialog, make sure that "Destination Folder" is the same folder where you restored the files you backed up.
15. In the "Server Properties" dialog, make sure the "WWW Path" leads to the folder where you restored the BigFix Server `wwwroot` folder. Make sure the "URL" matches the one in your masthead.
16. In the "Web Reports Properties" dialog, make sure the "WWW Path" leads to the folder where you restored the Web Reports `wwwroot` folder. Make sure the "Port" is the same as before.
17. Continue the installation procedure until completion.
18. If you customized your previous BigFix installation by performing additional configuration procedures, you will have to perform them again for this installation. For example, remember to re-apply your custom HTTPS configuration and to re-configure your Web Reports email server settings, if you had them in your previous installation. Only edit the BigFix registry keys directly if a documented configuration step calls for it. Moreover, direct modifications of those keys are better performed when the BigFix services are stopped. Do **not** import the BigFix registry settings from your previous installation. You can use the registry keys you backed up for verification purposes.

19. To avoid the duplication of the server computer entry on the BigFix Console, run the steps described in the **Client data backup** section in Avoiding duplicates when a Client is restored *(on page 346)*.
20. Reinstall the BigFix components and plugins that were installed on the original BigFix Server by re-deploying the appropriate Fixlets. For example, reinstall the BigFix WebUI, the BES Server Plugin Service, and any plugins you had.

Verifying restore results

Ensure that your BigFix Server has been restored.

Perform the following steps:

1. Verify that all services are started. On Windows platforms, you can use the BigFix Diagnostics tool.
2. Log in to the BigFix console and verify that the login works and that the database information was restored. In case of a login failure, see Error "Bad sequence parameter" when opening the BigFix Console.
3. Use SQL Server Management Studio to connect to the BFEnterprise database and examine the DBINFO and REPLICATION_SERVERS tables. Compare the current values to the values noted running the Server Backup procedure.
4. Verify that the new BigFix Server is able to connect to the database. Check all the Server logs for error messages on connecting to the database.

   Depending on your database authentication method (Windows versus SQL), it may be necessary to modify the domain/service accounts leveraged by the BigFix Server services (Root Server, GatherDB, FillDB, and Web Reports) to match the account previously leveraged with the old BigFix Server.
5. Reconfigure any appropriate BigFix Server settings.
6. 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 may take some time depending on your DNS services/ infrastructure).
7. Ensure that the BigFix clients and BigFix relays connect to the server when it is available and report data to it. Full recovery with all agents reporting might take from

a few minutes to many hours (depending on the size of the deployment and how long the server was unavailable). At least some agents should be reporting updated information within an hour.

8. After verifying that some agents are reporting to the server, send a blank action: **Tools > Take Custom Action** to all computers. The blank action does not make any changes to the agent computers, but the agents report that they received the blank action.
9. After restoring the data from the last backup, the BigFix server might restart at an earlier state with a disalignment between its mailbox and that of each relay. In this case, the BigFix server needs to resynchronize with the relays that have continued to process requests, otherwise the relays might ignore the requests of the server. To realign the mailboxes, see How to realign mailboxes after restoring a BigFix database backup.
10. Log in to the web reports and ensure that the data was restored.



Note: If a remote datasource is defined in the Web Reports configuration, Web Reports connects to the datasource only after you re-enter the datasource credentials in the Web Reports **Administration > Datasource Settings > Edit** page.

DSA Recovery

When recovering a lost DSA server, all top-level BigFix relays (and therefore the entire deployment) should already be pointing to the remaining DSA server.

It is recommended to leave all relays and clients reporting up to the working DSA server during this recovery procedure. If your existing relay settings do not allow this, isolate the server being restored on the network such that only the working DSA server can connect to it.

1. If the master DSA server fails, run the following procedure on the `BFEnterprise` SQL database to promote the secondary DSA server to master during restoration and replication of the failed server.

```
declare @ServerID varchar(32)
select @ServerID = convert(varchar(32),ServerID) from DBINFO
execute [BFEnterprise].[dbo].[update_adminfields]
 'Z:masterDatabaseServerID',@ServerID
```

   In this way you can install a new DSA server and you can run the Administration Tool on the secondary DSA server during the restoration of the failed server.
2. On the existing DSA server delete the failed DSA server id from the database.
   a. First see what the existing DSA server id is by executing the following SQL statement.

```
select ServerID from DBINFO
```

   b. List the IDs of the DSA servers:

```
select * from REPLICATION_SERVERS
```

   c. After identifying the failed server ID, run the following procedure:

```
execute BFEnterprise.dbo.delete_replication_server <ID>
```

   d. Run the following SQL statements to reset the OriginServerID field referencing the failed server ID:

```
update userinfo
set ManyVersion = dbo.fn_IncrementManyVersion( ManyVersion ),
OriginServerID = null,
OriginSequence = null
where OriginServerID = <ID>
```

```
update custom_sites
set ManyVersion = dbo.fn_IncrementManyVersion( ManyVersion ),
OriginServerID = null,
```

```
OriginSequence = null
where OriginServerID = <ID>
```

```
update SITENAMEMAP
set ManyVersion = dbo.fn_IncrementManyVersion( ManyVersion ),
OriginServerID = NULL,
OriginSequence = NULL
where OriginServerID = <ID>
```

```
update MAILBOX_COMPUTER_FILES
set ManyVersion = dbo.fn_IncrementManyVersion( ManyVersion ),
OriginServerID = NULL,
OriginSequence = NULL
where OriginServerID = <ID>
```

```
update MAILBOX_COMPUTERS
set Version = dbo.fn_IncrementManyVersion( Version ),
OriginServerID = NULL,
OriginSequence = NULL
where OriginServerID = <ID>
```

```
update MAILBOX_FILES
set ManyVersion = dbo.fn_IncrementManyVersion( ManyVersion ),
OriginServerID = NULL,
OriginSequence = NULL
where OriginServerID = <ID>
```

3. Restore the server operating system and database software in a pristine state without any BigFix server or the BigFix database remnants.
4. Restore the following items from backup:
    - `[BigFix Server folder]\BESReportsServer\wwwroot\ReportFiles`
    - `[BigFix Server folder]\Encryption Keys` (can be optionally restored by copying from the secondary server, or a new key generated by the Administration Tool)

- `[BigFix Server folder]\UploadManagerData` (optional, for faster recovery of SUA data if lost server was the SUA Source)
- `[BigFix Server folder]\wwwrootbes\bfmirror\downloads\ActionURLs`
- `[BigFix Server folder]\wwwrootbes\bfmirror\downloads\sha1` (optional, for faster recovery of cached files)
- `cert.pem` file for Web Reports, if using HTTPS
- `BESReporting` database in SQL Server

5. Install BigFix server using the installer and the existing masthead. For additional information see Installing Additional Windows Servers (DSA) *(on page 179)*.
6. Set the following registry values:
   For 32-bit Windows systems, go to `[HKLM\Software\BigFix\Enterprise Server\FillDB]` or for 64-bit Windows systems, go to `[HKLM\Software\Wow6432Node\BigFix\Enterprise Server\FillDB]` and then set the following values:

```
"PerformanceDataPath"[REG_SZ] = "[BigFix Server
 folder]\FillDB\FillDBperf.log"
"UnInterruptibleReplicationSeconds"[DWORD] = 14400 (decimal)
ReplicationDatabase=<DBName>
ReplicationUser=<DBUser>
ReplicationPassword=<DBPassword>
```

7. Restart the BES FillDB service.
8. Install BigFix client and console.
9. After replication completes, run the following procedure on in the SQL database to promote this newly restored BigFix server to be the master server.

```
declare @ServerID varchar(32)
select @ServerID = convert(varchar(32),ServerID) from DBINFO
execute [BFEnterprise].[dbo].[update_adminfields]
 'Z:masterDatabaseServerID',@ServerID
```

10. Reinstall and reconfigure the Plugins. Configuration information can be gathered from the currently operating DSA server or from installation notes and configuration details kept by the Administrator.
11. Set the following registry values and restart the BES FillDB service:

Go to `[HKLM\Software\Wow6432Node\BigFix\Enterprise Server\FillDB]` and then set the following values:

```
"PerformanceDataPath"[REG_SZ] = ""
"UnInterruptibleReplicationSeconds"[DWORD] = 120 (decimal)
```

12. Launch the Administration Tool and update the replication interval on this restored server to the desired level. Typically, this value should match the interval set on the other DSA Server.

    Note: Depending on the size of the deployment, the replication process might take multiple days to complete. To validate its completion, look for a `Replication Completed` message in the `FillDBperf.log` file. Connecting a separate BigFix console to each DSA server and comparing contents is another way to check that the data is synchronized in both deployments.

13. Revalidate the datasource on the Web Reports, editing the existing one.

Note: When you switch the servers, you have to wait for the endpoints to register with the new master server before you can send mailbox actions to them. Endpoints register automatically and periodically to a server by default every 6 hours. In the meantime, if you need to run any actions, this can be accomplished by running them as Dynamically target by property.

On Linux systems

If you back up the database and the BigFix Server files, when needed you can restore the BigFix environment on a Linux computer.

Server Backup

How to back up the BigFix Server on Linux.

Consider backing up to a remote system to allow for higher fault tolerance. Perform the following steps:

1. Stop all the BigFix processes, including running plug-ins if any. To do so, you can use the following commands:

```
systemctl stop beswebui
systemctl stop beswebreports
systemctl stop besclient
systemctl stop besgatherdb
systemctl stop besfilldb
systemctl stop besserver
```

   Up to BigFix Version 11.0.1, for backwards compatibility, you may instead run the commands:

```
/etc/init.d/beswebui stop
/etc/init.d/beswebreports stop
/etc/init.d/besclient stop
/etc/init.d/besgatherdb stop
/etc/init.d/besfilldb stop
/etc/init.d/besserver stop
```

2. If you have remote processes that may access the BigFix databases, like a remote BigFix WebUI, stop them as well.
3. Connect to the database instance hosting the BigFix databases and back them up.
   The BigFix Server database is named
   - `BFENT` if you are using DB2
   - `BFEnterprise` if you are using SQL Server

   The Web Reports database is named
   - `BESREPOR` if you are using DB2
   - `BESReporting` if you are using SQL Server

   Your databases might have different names if you customized them at installation time (not recommended).

   If you are using DB2, you can use the DB2 Client to back up your databases. For example:

```
su - db2inst1
db2 force application all
db2 deactivate database BFENT
db2 deactivate database BESREPOR
db2 backup db BFENT to /Absolute/Path/Of/ExistingFolder
db2 backup db BESREPOR to /Absolute/Path/Of/ExistingFolder
```

If you are using SQL Server, you can use the sqlcmd. For example:

```
sqlcmd -U sa -S localhost,1433
BACKUP DATABASE BFEnterprise
TO DISK = '/Absolute/Path/Of/ExistingFolder/BFEnterprise.bak';
GO
BACKUP DATABASE BESReporting
TO DISK = '/Absolute/Path/Of/ExistingFolder/BESReporting.bak';
GO
```

4. Manually back up the following folders:

```
/var/opt/BESClient
/var/opt/BESCommon
/var/opt/BESServer
/var/opt/BESWebReportsServer
/var/opt/BESWebUI
```

5. Back up your license key, license certificate, and masthead file.

   The `license.pvk` and `license.crt` files are critical to the security and operation of BigFix. If the private key (`pvk`) files are lost, they cannot be recovered.

   The masthead (`afxm`) file is an important file that contains information about the BigFix Server configuration. To back it up, you can copy the `/etc/opt/BESServer/actionsite.afxm` file and rename it to `masthead.afxm`.

6. Analyze the BigFix Server database tables named `DBINFO` and `REPLICATION_SERVERS`.

   If you are using DB2, you can use the DB2 Client. For example:

```
su - db2inst1
db2 connect to BFENT user db2inst1 using *****
db2 "SELECT * FROM dbo.DBINFO"
db2 "SELECT * FROM dbo.REPLICATION_SERVERS"
```

If you are using SQL Server, you can use the sqlcmd. For example:

```
sqlcmd -U sa -S localhost,1433
SELECT * FROM [BFEnterprise].[dbo].[DBINFO];
GO
SELECT * FROM [BFEnterprise].[dbo].[REPLICATION_SERVERS];
GO
```

Record all the rows in those tables for verification purposes.

If DNS aliases are being leveraged for the servers, the field values of those tables should remain the same after a restore procedure. If they contain hostnames, and the hostnames are changing, those fields may need manual modification after the restore. If you want to update the CN on the BigFix internal certificates, see How to change the Common Name (CN) on BigFix internal certificates.

7. To avoid the duplication of the BigFix Server computer entry on the BigFix Console, run the steps described in the **Client data backup** section in Avoiding duplicates when a Client is restored *(on page 346)*.

Server Recovery

How to restore a BigFix Server on Linux.

1. Ensure that the new BigFix Server computer can be reached on the network using the same URL that is in the masthead file. For example, if you look at the field `X-Fixlet-Site-Gather-URL`, it contains something like `http://192.168.10.32:52311/cgi-bin/bfgather.exe/actionsite` or `http://bigfixserver.company.com:52311/cgi-bin/bfgather.exe/actionsite`.

Note: To avoid issues when the BigFix clients connect to the BigFix Server before it is fully restored, ensure that it is not available on the network until the recovery is complete.

2. When running the recovery procedure on the same computer that the BigFix Server was installed on, remove all the installed BigFix components, including any plug-in. Remove all BigFix files and folders. For a detailed guide about removing the BigFix components on Linux, see Removing the BigFix components from Linux *(on page 276)*. If you are performing a full restore, remember to remove the BigFix databases too.
3. Restore the BigFix databases from your backups.

   If you are using DB2, restore the `BFENT` and `BESREPOR` databases. For example:

   ```
   su - db2inst1
   db2 restore db BFENT from /Absolute/Path/Of/Backup/Folder
   db2 restore db BESREPOR from /Absolute/Path/Of/Backup/Folder
   db2 activate database BFENT
   db2 activate database BESREPOR
   ```

   If you are using SQL Server, restore the `BFEnterprise` and `BESReporting` databases. For example:

   ```
   sqlcmd -U sa -S localhost,1433
   RESTORE DATABASE BFEnterprise
   FROM DISK = '/Absolute/Path/Of/BFEnterprise.bak';
   GO
   RESTORE DATABASE BESReporting
   FROM DISK = '/Absolute/Path/Of/BESReporting.bak';
   GO
   ```
4. Restore the folders and files you previously backed up:

   ```
   /var/opt/BESClient
   /var/opt/BESCommon
   ```

```
/var/opt/BESServer
/var/opt/BESWebReportsServer
/var/opt/BESWebUI
```

5. Remove the old password files:

```
/var/opt/BESClient/besclient.obf
/var/opt/BESServer/besserver.obf
/var/opt/BESWebReportsServer/beswebreports.obf
```

6. If you had the WebUI, remove the folder that contains the WebUI certificates:

```
/var/opt/BESWebUI/cert
```

7. Copy the old configuration files to a temporary directory, as they might contain custom settings that you use, then delete them:

```
/var/opt/BESClient/besclient.config
/var/opt/BESServer/besserver.config
/var/opt/BESWebReportsServer/beswebreports.config
/var/opt/BESWebUI/beswebuiservice.config
```

8. Go to the BigFix Download Center Platform Release page, find the same patch level of the original installation and go to the corresponding download page. Then download the "Server" setup file. Make sure to download the version for your database type (DB2 or MS-SQL).
9. Run the BigFix Server installer passing it the option to reuse the existing databases.

```
./install.sh -reuseDb
```

10. When the installer asks you to "Select the BigFix features", choose the same BigFix components as your previous installation.
11. When the installer asks you to "Select the database replication"
    - select the "Single or Master Database" option if you are restoring your only BigFix Server or the Primary/Master server of a DSA environment
    - select the "Replicated Database" option if you are restoring the Secondary/ Replica server of a DSA environment

12. When the installer asks you to "Choose the server's root folder", specify the path where you restored your BigFix Server folder.
13. When the installer asks you to "Choose the Web Reports server's root folder", specify the path where you restored your Web Reports folder.
14. When the installer asks you what port a BigFix component should use, specify the same port that component used in your previous installation.
15. When the installer asks you to "Create the initial administrative user", specify the same BigFix master operator that you used for the installation you backed up.
16. When the installer asks you to "Choose the type of setup", choose the option "I want to install with an existing masthead". Then, when it asks you the "Deployment masthead", specify the path to the masthead you backed up.
17. Continue the installation procedure until completion.
18. If you customized your previous BigFix installation by performing additional configuration procedures, you will have to perform them again for this installation. For example, remember to re-apply your custom HTTPS configuration and to re-configure your Web Reports email server settings, if you had them in your previous installation. Only edit the BigFix configuration (`conf`) files directly if a documented configuration step calls for it. Moreover, direct modifications of those files are better performed when the BigFix services are stopped. You can use the configuration files you backed up for verification purposes.

    ```
    /var/opt/BESClient/besclient.config
    /var/opt/BESServer/besserver.config
    /var/opt/BESWebReportsServer/beswebreports.config
    /var/opt/BESWebUI/beswebuiservice.config
    ```

19. To avoid the duplication of the server computer entry on the BigFix Console, run the steps described in the **Client data backup** section in Avoiding duplicates when a Client is restored *(on page 346)*.
20. Reinstall the BigFix components and plugins that were installed on the original BigFix Server by re-deploying the appropriate Fixlets. For example, reinstall the BigFix WebUI, the BES Server Plugin Service, and any plugins you had.

Verifying restore results

Ensure that your BigFix Server has been restored.

Perform the following steps:

1. Verify that all services are started.
2. Log in to the BigFix console and verify that the login works and that the database information was restored. In case of a login failure, see Error "Bad sequence parameter" when opening the BigFix Console.
3. Use the DB client to connect to BFENT and examine the DBINFO and REPLICATION_SERVERS tables. Compare the current values to the values noted running the Server Backup procedure.
4. Verify that the new BigFix Server is able to connect to the database. Check all the Server logs for error messages on connecting to the database.
5. Reconfigure any appropriate BigFix Server settings.
6. 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 may take some time depending on your DNS services/ infrastructure).
7. Ensure that the BigFix clients and BigFix relays connect to the server when it is available and report data to it. Full recovery with all agents reporting might take from a few minutes to many hours (depending on the size of the deployment and how long the server was unavailable). At least some agents should be reporting updated information within an hour.
8. After verifying that some agents are reporting to the server, send a blank action: **Tools > Take Custom Action** to all computers. The blank action does not make any changes to the agent computers, but the agents report that they received the blank action.
9. After restoring the data from the last backup, the BigFix server might restart at an earlier state with a disalignment between its mailbox and that of each relay. In this case, the BigFix server needs to resynchronize with the relays that have continued to process requests, otherwise the relays might ignore the requests of the server. To realign the mailboxes, see How to realign mailboxes after restoring a BigFix database backup.

10. Log in to the web reports and ensure that the data was restored.



Note: If a remote datasource is defined in the Web Reports configuration, Web Reports connects to the datasource only after you re-enter the datasource credentials in the Web Reports **Administration > Datasource Settings > Edit** page.

Enabling the DB2 database online backup

Starting from BigFix Version 9.5 Patch 3, you can customize your Linux environment to run the online backup of the `BFENT` and `BESREPOR` DB2 databases.

The advantage of running a DB2 online backup is that the backup can run while the applications and the services are accessing the database so there is no outage time window. You can either schedule the online backup or run it from the DB2 command line on the DB2 server.

As a prerequisite for running the DB2 online backup, you must ensure that the DB2 archive logging is active on the DB2 server.

From the BigFix point of view, these are the possible scenarios that you might run into:

You ran a fresh install of BigFix V9.5 Patch 3

In this case the BigFix databases are already enabled to use the DB2 database online backup. You only need to enable the DB2 archive logging function for the BigFix databases on the DB2 server.

You upgraded BigFix from an earlier version to V9.5 Patch 3

If so, you must enable the BigFix databases to use the DB2 database online backup function as indicated in https://bigfix-mark.github.io/.

Automatic DB2 databases backup upon upgrade

You can configure your BigFix Server to automatically run the backup of the `BFENT` and `BESREPOR` databases before and after running the upgrade process.

Enabling the feature

To enable this behavior you must assign an existing path, accessible both by `root` and by the database instance owner, by default `db2inst1`, to the advanced option `automaticBackupLocation`, for example:

```
/opt/BESServer/bin/BESAdmin.sh -setadvancedoptions
    -sitePvkLocation=<pvkLocation>  -sitePvkPassword=<pvkPassword>
    -update automaticBackupLocation="/my/path"
```

This behavior applies both when you run the upgrade from the command line and when you run the upgrade by deploying the upgrade Fixlet.

Two backups are generated for each of the `BFENT` and `BESREPOR` databases during this process, one before the upgrade and one after the upgrade, for a total of four backups. The files containing the backups are stored in the directory that you specified in the `automaticBackupLocation` advanced option.



Note: Ensure that there is enough disk space available on the file system to store the four backup files.

These are sample backup files generated during the upgrade:

```
BFENT.0.db2inst1.DBPART000.20160711142219.001
BESREPOR.0.db2inst1.DBPART000.20160711142240.001

BFENT.0.db2inst1.DBPART000.20160711142306.001
BESREPOR.0.db2inst1.DBPART000.20160711142327.001
```

The fifth digit in the file name is a time stamp. For example, `20160711142219` in the first file means `2016-07-11 at 14:22:19`.

In the installation log `BESInstall.log` you can see the time stamp of each database backup along with the information about whether the backup was generated before or after the upgrade.

Troubleshooting the feature

If one of the initial backup fails, the upgrade process fails. Investigate what prevented the backup from running and rerun the upgrade.

If one of the two final backups fails, the BigFix server is upgraded successfully and you get a warning message informing you that one of the two final backups failed. Investigate what prevented the backup from running and rerun the backup of the database manually.

You find the information about the backup failure in the `/var/log/BESAdminDebugOut.txt` file.

Disabling the feature

If you want to disable the automatic backup feature, run the command:

```
/opt/BESServer/bin/BESAdmin.sh -setadvancedoptions
    -sitePvkLocation=<pvkLocation> -sitePvkPassword=<pvkPassword>
    -delete automaticBackupLocation
```

DSA Recovery

When recovering a lost DSA server, all top-level BigFix relays (and therefore the entire deployment) should already be pointing to the remaining DSA server.

It is recommended to leave all relays and clients reporting up to the working DSA server during this recovery procedure. If your existing relay settings do not allow this, isolate the server being restored on the network such that only the working DSA server can connect to it.

1. If the master DSA server fails, run the following procedure on the `BFEnterprise` database to promote the secondary DSA server to master during restoration and replication of the failed server.

   ```
   db2
     set schema dbo
     select serverid from DBINFO (take count of SERVERID)
     set current function path dbo
   ```

```
 call update_adminFields('Z:masterDatabaseServerID','<serverid>') -
Replace
SERVERID with the value from the previous query
```

In this way, you can install a new DSA server and you can run the Administration Tool on the secondary DSA server during the restoration of the failed server.

2. On the existing DSA server, delete the failed DSA server id from the database:
   a. List the IDs of the DSA servers:

   ```
   select * from REPLICATION_SERVERS
   ```

   b. After identifying the failed server ID, run the following procedure:

   ```
   call dbo.delete_replication_server(ID)
   ```

3. Restore the server operating system and database software in a pristine state without any BigFix server or the BigFix database remnants.
4. If you followed the server backup procedure described at:

   Server Backup (on page 584)

   Follow the server recovery procedure, starting from Step 3, described at:

   Server Recovery (on page 587)

5. Stop the BigFix FillDB process, set the following keywords in the `besserver.config` file and restart the FillDB process.

   ```
   PerformanceDataPath = <Performance_Data_Path_filename>
   UnInterruptibleReplicationSeconds = 14400
   ReplicationDatabase=<DBName>
   ReplicationUser=<DBUser>
   ReplicationPassword=<DBPassword>
   ```

   where `<Performance_Data_Path_filename>` might be `/var/opt/BESServer/FillDBData/FillDBPerf.log`.

6. After replication completes, run the following procedure in the database to promote this newly restored BigFix server to be the master server.

```
db2
  set schema dbo
  select serverid from DBINFO (take count of SERVERID)
  set current function path dbo
  call update_adminFields('Z:masterDatabaseServerID','<serverid>') -
 Replace
SERVERID with the value from the previous query
```

7. Reinstall and reconfigure the plug-ins. Configuration information can be gathered from the currently operating DSA server or from installation notes and configuration details kept by the Administrator.
8. Set the following keywords in the `besserver.config` file and restart the BES FillDB service:

```
PerformanceDataPath = ""
UnInterruptibleReplicationSeconds = 120
```

9. Launch the Administration Tool and update the replication interval on this restored server to the desired level. Typically, this value should match the interval set on the other DSA server.



Note: Depending on the size of the deployment, the replication process might take multiple days to complete. To validate its completion, look for a `Replication Completed` message in the `FillDBperf.log` file. Connecting a separate BigFix console to each DSA server and comparing contents is another way to check that the data is synchronized in both deployments.

10. Revalidate the datasource on the Web Reports, editing the existing one.



Note: When you switch the servers, you have to wait for the endpoints to register with the new master server before you can send mailbox actions to them. Endpoints register automatically and periodically to a server by default every 6 hours. In the meantime, if you need to run any actions, this can be accomplished by running them as Dynamically target by property.

Chapter 18. HA and DR configurations

Starting with BigFix Platform Version 11.0.4, it is easier to use third-party technologies to implement High Availability (HA) and Disaster Recovery (DR) configurations for the BigFix Server component on Windows.

Introduction to HA and DR

The objective of this section is to provide the basic concepts needed to structure the top level of your BigFix environment in a way that can help you achieve "business continuity".

We will use those concepts to lay the foundations of a design that can enable your BigFix environment to remain operational in the event of a simple failure and to make it possible to recover its operational capabilities after a disaster.

Objectives and assumptions

A business continuity plan regarding a software infrastructure, like the one formed by the BigFix Platform components, usually involves making the core system capable of "high availability" (HA) and making it possible to perform an effective "disaster recovery" (DR) for it.

Both concerns involve hardware arrangement decisions and software configuration choices and both require a certain degree of infrastructural redundancy. However, each concern aims to solve a slightly different set of issues and has different constraints.

Depending on your needs, you can implement:

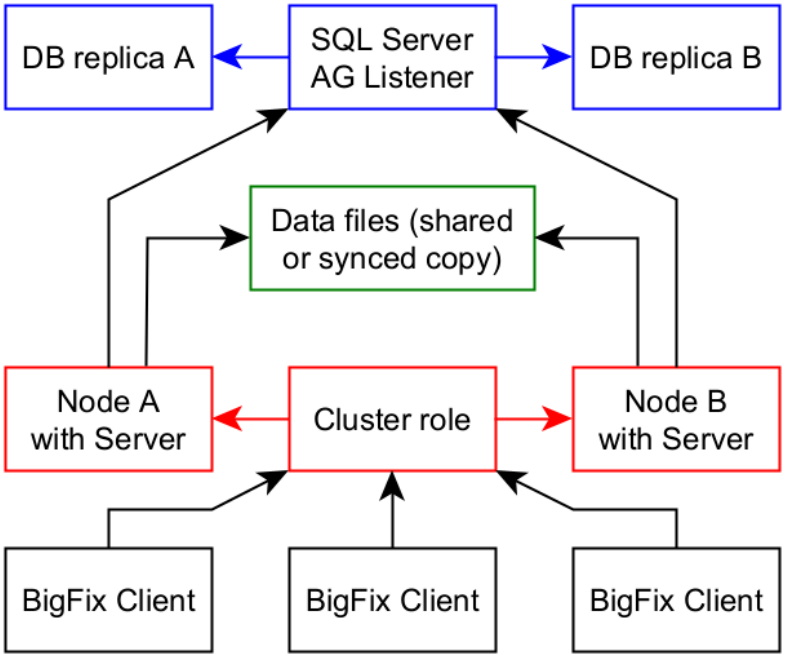
- HA, to maximize uptime by being resilient to the more common issues
- DR, to be able to recover from a catastrophic event, which can cause service interruptions
- both HA and DR

Conceptually, having HA is like being able to replace part of a system without turning it off, while DR is like having a backup in a safe location.

Most of the premises of this introduction are valid in general. However, we do make these two assumptions:

- HA works with low latency and is intended to have no data loss
- DR works with high latency and can admit a degree of data loss

This diagram represents a simplified HA or DR configuration for a BigFix environment.



In the image above:

- blue boxes implement database redundancy
- green boxes implement file storage redundancy (optional)
- red boxes implement computer redundancy

An HA environment can be configured to have just database redundancy and computer redundancy, if the data files are shared on a common storage.

A DR environment always needs all three types of redundancy.

Database redundancy

For database redundancy, SQL Server Enterprise edition provides a feature called "Always On availability groups" (AG).

This feature lets you set up an "availability group", which provides redundancy by coordinating two or more "availability replicas". Each availability replica requires its own SQL Server instance and can contain one or more databases.

At any time, one replica acts as the "primary" of the group and the others act as "secondary". To operate on a database, rather than connecting directly to a specific replica, you connect to the availability group, via an "availability listener". An "availability listener" provides its own hostname and a virtual IP to contact the availability group it is associated with. This arrangement makes it possible to work on a database without needing to know what replica is the current primary.

It is possible to configure an availability replica for HA and a different replica for DR.

The "availability mode" of a database replica should be set to:

- "synchronous commit" for HA
- "asynchronous commit" for DR

A database availability replica configured for HA can also be a "readable secondary", to increase performance for read-heavy database operations.

Computer redundancy

For computer redundancy, Windows Server provides a feature called "Windows Server Failover Cluster" (WSFC).

This feature lets you set up a "failover cluster", which provides redundancy by coordinating two or more "cluster nodes". Each cluster node is a Windows Server computer that was added to the same failover cluster. All computers that are part of the same failover cluster must be part of the same Active Directory domain.

In a failover cluster, cluster nodes take turns in providing a specific service, or running a certain operation. At any time, a single cluster node will be the "owner node" of a specific "cluster role". In the context of a WSFC, a "cluster role" is a set of resources needed by

a computer to provide a specific service. A different cluster node can become the new owner of a cluster role if that role is "moved" to it, or if a "failover" happens. Usually, a role is manually moved off its owner node to perform maintenance on that computer, while a failover is performed automatically when a potential problem is detected.

A cluster role can be assigned its own hostname and virtual IP. Other resources can be applications installed on the cluster nodes. For example, a cluster role can consist of a Windows service running a program that acts as an HTTPS server, a new own hostname and a virtual IP. This arrangement makes it possible to connect to the HTTPS server via the cluster role hostname or IP, without needing to know on what node it is currently running on.

File storage redundancy

When discussing file storage redundancy, it is useful to distinguish between "program files" and "data files". By "program files" we mean the executable and other files created by a fresh installation. By "data files" we mean the files that the program creates or operates on.

A simple solution to achieve redundancy of the program files could be installing the same version of that program on several computers. To keep consistency, this solution would require upgrading all installations together or in rapid succession.

For data files, depending on your needs, full redundancy might not be necessary. You can configure an environment that achieves HA without having to duplicate your data files by placing them on a "shared storage" where your program can find always them, independently of the computer it is running on. In order to be able to perform DR, you can keep an additional copy of your data files on a separate storage at a different location.

Shared file storage

In a failover cluster, you can wrap your program in a cluster role. Only one cluster node can own that role at a time and run it. The other failover cluster nodes can only take over that role when a failover happens and a different computer is chosen as the owner node. If you place your data files on a shared storage, computers that are part of the same failover cluster will only access them one at a time. This arrangement avoids common concurrency issues like deadlocks and race conditions.

For shared file storage, Windows Server provides a feature called "Cluster Shared Volume". This feature enables multiple nodes in a Windows Server failover cluster to simultaneously have read-write access to the same storage volume. This volume can be a reserved space on a disk mounted on a separate computer, which does not have to be a cluster node. If your data files are on a Cluster Shared Volume, a program configured as a cluster role can fail over to a different cluster node and have access to the same data straight away. For more information, see Use Cluster Shared Volumes in a failover cluster.

Synchronized file copy

Having an additional copy of your data files is an important part of being ready to perform DR. It is also a possible way of achieving HA in your environment. Depending on your needs, such a copy can be made by a scheduled backup, or maintained by a continuous process.

You can store additional copies of your data files at different locations and use a synchronization mechanism to update them. For HA, this is optional, but if you choose to implement it this way, your copies must be fully consistent with the original, so you need a fully synchronous file update mechanism. For DR, you do need to have at least an additional copy of your data files, but it can be updated with a slight delay and so the file update mechanism can be asynchronous.

For actual file storage redundancy, Windows Server provides a feature called "Storage Replica". This feature supports several configurations, named "server-to-server", "cluster-to-cluster" and "Stretch Cluster". Each Storage Replica configuration can be set up to act either synchronously or asynchronously. In any case, Storage Replica only support one-to-one replication of a specific volume pair. Storage Replica only supports automatic failover when deployed in a Stretch Cluster configuration. For more information, see Storage Replica overview.

Achieving both HA and DR

You can achieve both HA and DR in several ways. The following is an example.

For HA, you can set up two computers at the same location and make them part of the same failover cluster. Install your program on both of them. Make them use the same database and place your data files on a shared storage within close distance. Create a cluster role and use it to wrap your program. If you program is a server, you can use the role

to provide the network resources to contact it independently of the computer it is running on. Use a second SQL Server instance at the same location to implement an availability group for the program database, ensuring both instances act as synchronous replicas. Configure your two computers to access your program database via an availability group listener.

For DR, you can set up a third computer at a different location. Provide it with its own storage and set up a mechanism to asynchronously replicate your data files from the other location to this one. At this different location, use a third SQL Server instance to add an asynchronous replica to the availability group you created for your program database.

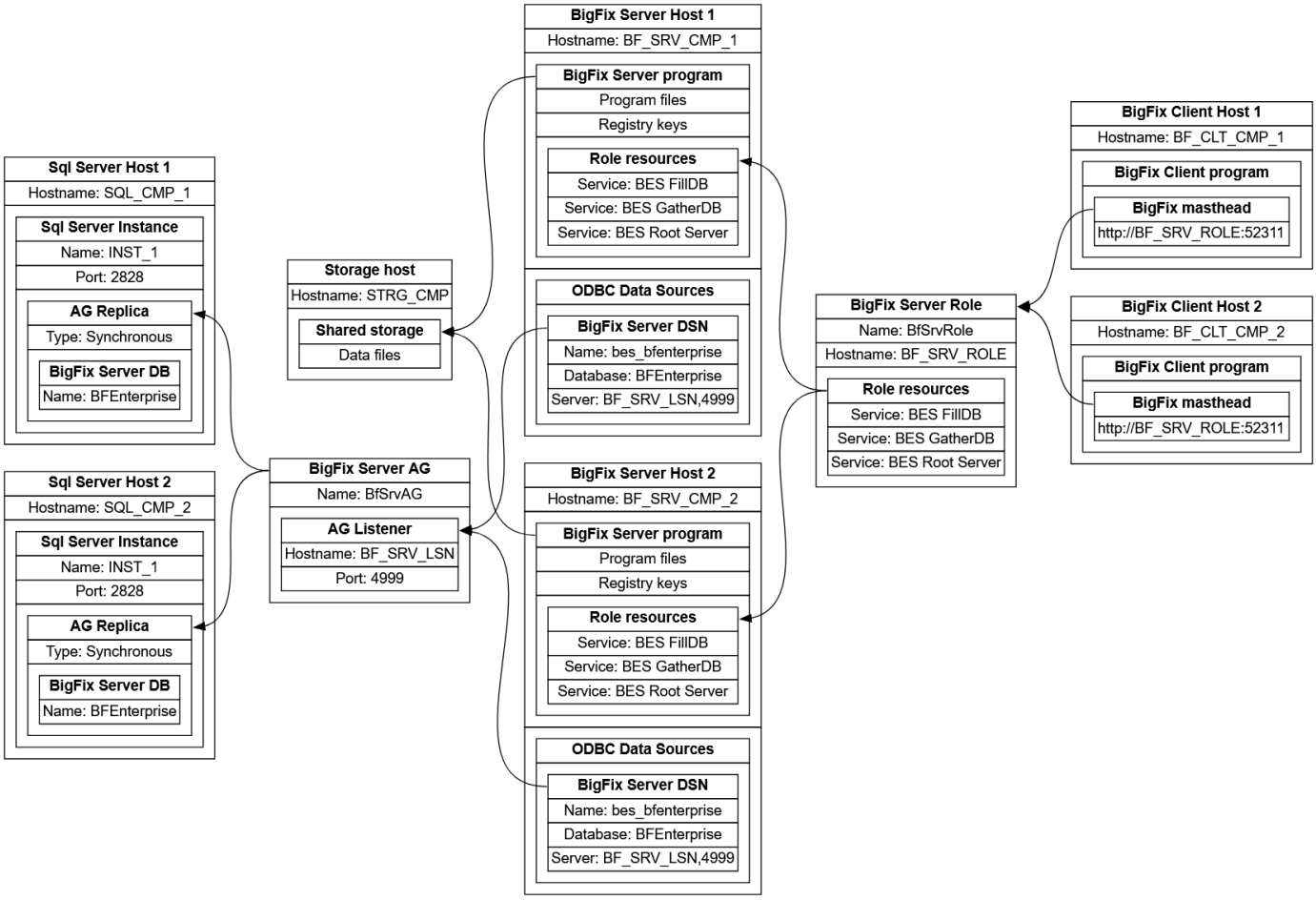
Configuring HA and DR on Windows

How to set up a BigFix Server for High Availability (HA) and Disaster Recovery (DR) on Windows.

The BigFix Server can be installed on more than one Windows Server computer and configured to leverage a set of Microsoft technologies to provide HA and DR.

This is an example HA configuration that will serve as a reference. It can be adapted and expanded to provide DR.

The following picture shows BigFix HA configuration on Windows using a shared storage for data files.



Basic requirements

If you are starting from scratch, we recommend using BigFix Platform version 11.0.4 or greater.

If you already have a BigFix Server installation that you want to use as part of this configuration, we recommend upgrading it to version 11.0.4 or greater.

We will use:

- two computers, one for each BigFix Server installation we will configure
- two SQL Server Enterprise edition instances, installed and ready to use
- a Windows Server Failover Cluster (WSFC), that we will create and configure
- a shared storage, preferably a Cluster Shared Volume, on another computer

The computers hosting the two SQL Server instances can be the same computers we will use to host our two BigFix Server installations, or separate ones.

In this guide, we assume that all computers hosting a BigFix Server and/or a SQL Server instance are part of the same Active Directory domain and can become part of a single failover cluster.

The computer hosting the shared storage should be part of the same Active Directory domain, but it will not need to become part of our cluster.

We will mention possible variations compared to the environment above, but just for comparison.

The hostname or IP in the BigFix masthead

The hostname (or IP) that appears in the BigFix masthead is a key resource to consider.

Open the `ActionSite.afxm` file in your BigFix Server folder and look at the value of the `X-Fixlet-Site-Gather-URL` field. If it is, for example, `http://bfserver.domain.test:52311/cgi-bin/bfgather.exe/actionsite`, then `bfserver.domain.test` is the hostname we refer to.

That hostname (or IP) cannot be changed. If it currently belongs to a computer, you must be able to free it in order to properly set up this configuration.

After completing this configuration, the hostname (or IP) in your masthead file cannot belong to any specific computer as it will be used to redirect the BigFix network traffic to the currently active BigFix Server.

Instead of corresponding to a specific computer, that hostname (or IP) will be associated to a "cluster role" we are going to create in a failover cluster.

For the correct functioning of the BigFix environment we are configuring, it is fundamental that the hostname (or IP) in the masthead becomes exclusively associated to a "cluster role" we are going to create.

At any time, the "cluster node" (a computer) that is running the BigFix Server will be the "owner node" of that cluster role, but it will still have its own hostname and IP.

The mentioned cluster role must not be confused with the entire failover cluster or with the single cluster nodes.

The hostname (or IP) in the BigFix masthead must remain distinct from those used by: the cluster as a whole, the computer hosting the SQL Server instances, and the Availability Group listener we will discuss later.

Freeing the BigFix masthead hostname

If you already consumed your BigFix license authorization (.BESLicenseAuthorization) file and the hostname (or IP) in your BigFix masthead file is in use, you must free it before continuing.

For example, if your masthead contains the hostname `bfserver.domain.test` and you have a computer whose fully qualified domain name (FQDN) is also `bfserver.domain.test`, you should rename that computer from `bfserver` to something else, like `bfhost1`.

Make sure there are no DNS entries mapping the masthead hostname to a specific computer. On the Windows Server computer acting as your Active Domain controller, open the "DSN Manager" application, expand the tree with your NetBIOS domain name (e.g. `DOMAIN`), then the "Forward Lookup Zones", then your DNS domain name (e.g. `domain.test`) and look at the entries. If there are other devices that may act as DNS on your network, verify them as well.

If you renamed a computer to free the masthead hostname, make sure there are no leftover entries in your Active Directory referencing the old computer name. On the Windows Server computer acting as your Active Domain controller, open the "Active Directory Users and Computers" application, expand the tree with your DNS domain name (e.g. `domain.test`), select the "Computers" item and look at the entries.

Creating a Windows Server Failover Cluster

A Windows Server Failover Cluster has its own hostname and IP, which the BigFix Platform components will not make direct use of. Do not confuse them with the hostname and IP we will assign to the cluster role we will create later.

Before continuing, make sure the firewalls of your cluster nodes are configured to allow the kind of network traffic necessary for them to communicate with each other and for the cluster management tools to coordinate them.

Install the "Failover Clustering" Windows Server feature on the computers where you already installed or plan to install the BigFix Server. You can do that by running the following PowerShell command on them.

```
Install-WindowsFeature -Name Failover-Clustering -IncludeManagementTools
```

The "-IncludeManagementTools" part is optional, but you will need to include it on at least one computer in order to be able to use the cluster management tools.

If possible, also install that feature on the computers hosting the SQL Server instances containing your BigFix databases or that you want to use as a replica.

We suggest restarting the computers you install the "Failover Clustering" feature on.

We will now create a failover cluster, making the computers we mentioned part of it and turning them cluster nodes.

In the following steps, we will assume you are using the Failover Cluster Manager, but the same operations can be performed via PowerShell commands.

Open the Failover Cluster Manager, right click on the "Failover Cluster Manager" tree item, then click "Create Cluster...". This opens the "Create Cluster Wizard".

In the "Select Servers" page, add the computers you installed the "Failover Clustering" feature on.

If the next page is titled "Validation Warning", perform the suggested configuration validations tests and remediate any issues they find.

In the "Access Point for Administering the Cluster" page, specify the "Cluster Name" and an IP address, remembering not to use the ones we kept free for the cluster role.

Follow the wizard until completion.

Creating a cluster role with network resources

In this section, we assume that you added the computers hosting the BigFix Server to the same Windows Server Failover Cluster. If that is not the case, make them part of the same cluster before proceeding.

In the following steps, we will assume you are using the Failover Cluster Manager, but the same operations can be performed via PowerShell commands.

Create an empty cluster role. To do so, expand the tree with the cluster name, right click on the "Roles" tree item, then click "Create Empty Role".

To rename the role, right click on it, then click Properties. In the "General" tab, edit the "Name" field and then confirm.

In the same tab, you can assign one or more "Preferred Owner Nodes". Specifying this preference is optional. For example, you can set the first computer you installed the BigFix Server on as your only preferred owner node.

Make sure the "Status" of the new cluster role is "Stopped". To do so, right click on the cluster role you created, then click "Stop Role".

Assign a hostname and an IP to the cluster role. Make sure to assign the same hostname (or IP) that appears in the BigFix masthead. To do so, right click on the cluster role, select "Add Resource", then click "Client Access Point". Specify the hostname in the "Name" field and the IP in the "Address" field. Then proceed and complete the procedure.

Redirecting traffic via a cluster role

As a temporary measure, you might need to redirect the traffic meant for the hostname (or IP) in the BigFix masthead to a computer where you installed the BigFix Server or to one you want to install it on.

To do that, you can use the BigFix Server cluster role, assuming you associated the masthead hostname and IP to it, as we suggested.

In the following steps, we will assume you are using the Failover Cluster Manager, but the same operations can be performed via PowerShell commands.

In the "Roles" view of the Failover Cluster Manager, right click on the cluster role you created, select "Move", then click "Select Node...".

Select the cluster node where you are going to install the BigFix Server, then confirm.

The redirection will only be active if the cluster resources corresponding to the hostname and IP are online. To ensure that, you can start the whole role, or just bring online its network resources.

Using a new license authorization file

If you have a new BigFix license authorization (.BESLicenseAuthorization) file, follow this procedure.

We recommend that the computer on which you are going to install your BigFix Server is part of a Windows Server Failover Cluster. Make sure you already created an empty cluster role and associated a free hostname and a free IP to it.

Make the cluster role redirect traffic from its hostname (or IP) to the computer where you want to install the BigFix Server. For more details, see the "Redirecting traffic via a cluster role" section.

Go to the BigFix Download Center Platform Release page, find the version you want to install (11.0.4 or later), and visit to the corresponding download page. Then download the "Installer Generator" setup file.

Run the BigFix Installation Generator on the computer where you want to install the first BigFix Server.

When you reach the "Setup Type" dialog, choose the option "I want to install with a BigFix license authorization file".

In the "Request License" series of dialogs, when you reach the dialog asking "Please enter the DNS name of your BigFix Server", write the hostname (or IP) that you assigned to your cluster role.

Continue the BigFix Installation Generator setup procedure until completion.

Considerations about the first BigFix Server

Consider how you want your BigFix Server to access its SQL Server database.

If you want to use Windows authentication for database access, you should run the BigFix Server installer as an Active Domain administrator that is also an SQL Server user with sufficient privileges. Do not use a local Windows user for database access, even if the database is local.

If you want to use SQL Server authentication for database access, later on you may have to re-create the login user to ensure it can access all the SQL Server instances you want to use.

After installing the BigFix Installation Generator, run the BigFix Server installer on the same computer.

At the "Select Database Replication" dialog, select the "Single or Master Database" option.

At the "Select Database" dialog, select the "Use Remote Database" option, even if your SQL Server instance is local.

At the "Choose Destination Location" dialog, specify a folder local to the computer that you are installing on.

At the "Installation Wizard Complete" dialog, choose not to start your BigFix Server service.

Preparing to create an SQL Server Availability Group

After the first BigFix Server is installed, you can set up an SQL Server Always On Availability Group (AG) for its database.

To create an Availability Group, you need two computers with the same version of SQL Server Enterprise edition. We recommend that the computers hosting those database instances are part of the same Windows Server Failover Cluster, because it makes achieving HA easier.

One of those SQL Server instances must be the one hosting the database of the BigFix Server. If you installed Web Reports on the same computer as the BigFix Server, you can put their databases in the same AG. This is not mandatory, but it can avoid potential issues.

Before creating the AG, make sure the firewalls of those computers are configured to allow the kind of network traffic necessary for them to communicate with each other and to synchronize their databases.

On each computer hosting an SQL Server instance you want to use for your AG, perform the following steps.

Open the SQL Server Configuration Manager and view the "SQL Server Services" page. Right click the "SQL Server" entry corresponding to main service of your instance, then click "Properties".

In the "Always On Availability Groups" tab, make sure that "Enable Always On Availability Groups" is checked.

In the "Log On" tab, if the computer is part of a cluster, make sure that an Active Directory user is running the service. You should configure the same user for all the SQL Server instances you want to use for your AG.

Click "OK" to confirm your changes and close the dialog. If you get a message stating that you need to restart the SQL Server instance for your changes to take effect, restart it.

Now, connect to the SQL Server instance containing your BigFix databases.

Make sure that the BigFix databases you want to put in your AG are in full recovery mode. To verify this, you can connect to the instance hosting it and run the following query.

```
SELECT name, recovery_model_desc
FROM sys.databases
WHERE name = 'BFEnterprise'
   OR name = 'BESReporting';
GO
```

If your BigFix databases are not in full recovery mode, you can set that mode by running the following commands.

```
ALTER DATABASE BFEnterprise
SET RECOVERY FULL;
GO
ALTER DATABASE BESReporting
SET RECOVERY FULL;
GO
```

You are required to perform a full backup of the databases that you are going to put into your AG. To ensure this, you can connect to the instance hosting them, make sure they are not being written to and then run the following commands.

```
BACKUP DATABASE BFEnterprise
TO DISK = 'C:\Path\To\ExistingFolder\BFEnterprise.bak'
WITH FORMAT, INIT;
GO
BACKUP DATABASE BESReporting
TO DISK = 'C:\Path\To\ExistingFolder\BESReporting.bak'
WITH FORMAT, INIT;
GO
```

Make sure that the database user you are choosing exists on both SQL Server instances and that it has sysadmin privileges.

If you want to access via Windows authentication, this user should be the same Active Domain administrator that installed the BigFix Server and that runs its services.

If you want to access via SQL Server authentication, the login user should have the same name, password and security identifier (SID) on both instances. To verify this, you can run this query on each instance.

```
SELECT name, sid
FROM syslogins
WHERE name = 'YourUser';
GO
```

If your SQL Server user has the same name but a different SID, you can delete that user on an instance and create it again with the same SID it has on the other instance. To specify the SID when creating a user, do as follows.

```
USE master;
GO
CREATE LOGIN 'SameUserName'
WITH PASSWORD = N'SamePassword',
```

```
sid=<SameSID>;
GO
```

Creating an SQL Server Availability Group

To create the AG, connect to the SQL Server instance where the database of the BigFix Server is on. In the following steps, we will assume you are using SQL Server Management Studio, but the same operations can be performed via PowerShell commands.

Right click on the "Always On High Availability" icon and run the "New Availability Group Wizard".

In the "Specify Options" page, specify the "Availability group name", which can be anything you want. This name will not be used to connect to your AG. Then, choose the "Cluster type". If your SQL Server instance is installed on a computer that is part of a failover cluster, that field should default to "Windows Server Failover Cluster". Otherwise, you can set it to "NONE".

In the "Select Databases" page, select the BigFix Server database. If you also see a Web Reports database, you have the option to select it as well.

Note: If you installed both the BigFix Server and Web Reports on the same computer, putting their databases in the same AG will prevent potential issues when upgrading BigFix. On the computer where they are installed, there is pair of OBDC data sources named "enterprise_setup", one 32-bit and the other 64-bit, that is used by the BigFix Server installer during the upgrade. That data source assumes both databases can be accessed the same way.

In the "Specify Replicas" page, there are several things to set. In the "Replicas" tab, there should be one entry in the "Availability Replicas" table, the one that hosts the BigFix Server database. Set its "Availability Mode" to "Synchronous commit".

Click "Add Replica", which opens a "Connect to Server" popup. Use it to connect to the other instance. Now there should be two entries in the table. If you want to use the instance you just added for HA, set its "Availability Mode" to "Synchronous commit". If you want to use it for DR, set it to "Asynchronous commit".

Note that you can configure the "Readable Secondary" property of each replica. You can set that field to "Yes" to increase your database reading performances. You can also configure this later, along with the "Read-Only Routing" options. For more information, see Types of client connections to replicas.

We recommend creating an Availability Group listener. You can create it now or later, as long as you configure it following our recommendations. For more details, see the "Creating an Availability Group listener" section.

In the "Select Data Synchronization" page, choose how the initial data synchronization will occur. The "Automatic seeding" option is probably the most convenient option, provided your SQL Server instances satisfy its requirements. In the "Validation" page, verify that all entries have "Success" as their "Result". Follow the wizard until completion.

To verify that the Availability Group is working as intended, you can right click on it and then click "Show Dashboard". The "Availability group state" should be "Healthy". The "Synchronization State" of the primary replica should be "Synchronized". The "Synchronization State" of the secondary replica should be "Synchronizing" at first and then become "Synchronized" once the database replication is complete.

Creating an Availability Group listener

After creating an Availability Group (AG) for your BigFix Server database, we recommend creating a listener for it. If you used a separate AG for the Web Reports database, we recommend creating a listener for it too.

If you set your AG to act as part of a Window Server Failover Cluster (WSFC), its listener will be able to receive database connections, which it will then forward to one of the database replicas. By default, the Availability Group listener will redirect its connections to the current primary replica. In addition to that, it can be configured to redirect read-only queries to a secondary replica, distributing the load and increasing performances.

We recommend assigning the listener its own hostname and IP. The listener also needs a port, which can be different from the one used by the SQL Server instances hosting its database.

When creating the listener for an AG whose cluster type is WSFC, SQL Server will try to associate it with the hostname and IP you specify. In order to do that, the user running the

SQL Server instance you are connected to needs the permissions to create objects in the Active Directory.

In the following steps, we will assume you are using SQL Server Management Studio, but the same operations can be performed via PowerShell commands.

You can create an AG listener when creating the AG via the "New Availability Group Wizard" procedure, or after that. If you want to create it after, right click on the AG you created, then click on "Add Listener...". In the "DNS Listener Name", enter a free hostname. Fill the "Port" field. Click "Add...", which opens a dialog named "New IP Address". Specify a free IP address and its subnet, which should be the same as that of your SQL Server instances. Confirm and complete the procedure.

If you are setting up a clusterless Availability Group, the listener cannot automatically redirect your traffic to a replica, nor can it receive traffic on its own IP. These limitations make it difficult to achieve HA with such a configuration.

On a clusterless AG, a possible workaround is setting the IP and port of the listener to match the IP and port of the primary replica and finding a way to map the listener hostname to that IP. On failover, such a workaround requires destroying the listener, recreating it with the IP and port of the new primary replica and mapping the listener hostname to the new IP. We recommend finding a more robust solution instead.

Using an Availability Group listener

After creating an Availability Group (AG) and a listener for it, we recommend using that listener to connect to the BigFix databases contained in that group.

However, the first BigFix Server you installed is likely configured to connect directly to an SQL Server instance. To make it use the Availability Group listener, you need to modify its ODBC data sources.

On the computer where you installed the BigFix Server, there are four or six ODBC data sources to update, depending on whether you installed Web Reports alongside it or not. For more information, see Configuring ODBC data sources.

If you configured your BigFix Server to use a local database when you installed it, do the following. Open the Registry Editor. Navigate to the registry key `HKEY_LOCAL_MACHINE`

`\SOFTWARE\Wow6432Node\BigFix\Enterprise Server`. Set the registry key value named `UseRemoteDB` of type `REG_SZ` to `1`.

You should update your ODBC data sources on all computers that use them to connect to the BigFix databases.

If you are not accessing the BigFix databases by specifying a data source name (DSN), update your connection strings to specify the hostname (or IP) and the port of the AG listener.

Installing an additional BigFix Server

This section explains how to install an additional BigFix Server, configuring it to use an SQL Server Always On Availability Group (AG) that you already set up.

We recommend that the computer on which you are going to install your BigFix Server is part of the same Windows Server Failover Cluster as the computer hosting your other BigFix Server(s).

We assume you created an SQL Server Availability Group (AG) for the BigFix Server database and set up a listener for it.

Make sure that any program that may access the BigFix databases or change the BigFix data files is stopped.

On the first computer you installed the BigFix Server on, follow the backup procedure (Server Backup (on page 573)) but skip the database backup steps and ignore the step about avoiding client duplication.

Make the cluster role redirect traffic from its hostname (or IP) to the computer where you want to install the BigFix Server. For more details, see the "Redirecting traffic via a cluster role" section.

On the computer you want to install the BigFix Server on, follow the restore procedure (Server Recovery (on page 576)) but skip the database recovery steps and, during the installation, take the following steps instead of their equivalent in the other guide.

At the "Select Database Replication" dialog, select the "Single or Master Database" option.

At the "Select Database" dialog, select the "Use Remote Database" option, even if you have a local SQL Server instance.

At the "Database Server" dialog, in the "Specify a database engine" field, specify the hostname (or IP) and port of the Availability Group listener.

At the "Choose Destination Location" dialog, specify a folder local to the computer that you are installing on.

At the last dialog, choose not to start the BigFix Server services nor, if present, the Web Reports service.

Ignore the step about avoiding client duplication.

Completing the BigFix Server cluster role

Make sure all BigFix services on the computers where you installed the BigFix Server or the WebUI are stopped.

Make sure that any program that may access the BigFix databases or change the BigFix data files is stopped.

In the following steps, we will assume you are using the Failover Cluster Manager, but the same operations can be performed via PowerShell commands.

One by one, add the BigFix Server services (BES Root Server, BES FillDB, BES GatherDB) to the cluster role.

In the "Roles" view of the Failover Cluster Manager, right click on the cluster role you created, select "Add Resource", then click "Generic Service". Select the "BES Root Server" service, then proceed and complete the procedure.

Repeat the same procedure to add the "BES FillDB" service and the "BES GatherDB" service to the same cluster role.

If you installed Web Reports on the same computers as the BigFix Server, you may also add the "BES Web Reports Server" service to the same cluster role. Otherwise, you could create another cluster role, provide it with a free hostname and a free IP, then add its service to it.

Since BigFix Platform version 11.0.4, the BigFix Client does not need its own role.

The WebUI service, even if installed on the same computer as the BigFix Server, should not be mapped to the same cluster role.

Managing clustered BigFix services

After putting a service in a cluster role, you should no longer start or stop it directly. You should only start and stop those services as part of their role.

You can stop a clustered service by stopping its entire role, or take it offline via the cluster manager. For example, using the Failover Cluster Manager, you can right click on the role and then click "Stop Role", or view the "Resources" tab of the role, right click on the service and then click "Take Offline".

Stopping a service that is part of a cluster role without involving the cluster manager can cause a failover.

Warning: at no point should a BigFix service that is part of a failover cluster role be running on two computers. In a worst-case scenario, this can lead to data corruption.

Configuring the BigFix Client on BigFix Server computers

On each computer where you installed the BigFix Server, add the `_BESClient_RelaySelect_UseMastheadRootServer` client setting and set it to `1`. Then start (or restart) the BigFix Client. This will make it use the hostname (or IP) written in the masthead to try and connect to the right BigFix Server, instead of assuming the local one is always running. For more information, see List of settings and detailed descriptions.

Moving data files to shared storage

A possible solution to implement HA in your BigFix infrastructure is configuring the BigFix Server to use shared storage for its data files.

The BigFix Server program files, like the executables (.exe) and other files that do not change after the installation, should remain local.

The data files that are created by the program after the installation can be moved to a storage shared by all computers the BigFix Server is installed on.

Before continuing, make sure that any program that may access the BigFix databases or change the BigFix data files is stopped.

The path of most BigFix data folders can be configured by editing the corresponding settings. In the following, we will show how to use a BESAdmin command to change the location of the BigFix Server data folders to a common parent folder on a shared storage.

We suggest choosing a destination folder on a Cluster Shared Volume, which looks and acts like a local folder, but is actually located on a different storage volume.

On the first computer you installed the BigFix Server on, run the BESAdmin `/setshareableserverdatapath` command with the `/copydata` option to make it copy the local data files to the shared folder. For example.

```
.\BESAdmin.exe /setshareableserverdatapath /topath=<shared storage
 dir> /copydata
/sitePvkFile=<path+license.pvk> /sitePassword=<password>
```

BESAdmin writes the location of the copied files and folders to its log, named `BESAdminDebugOut.txt`. After you verify the file copy is successful, you can delete the local copy of the data, but keep the folders used by the log files. For more information, see BESAdmin Windows Command Line (on page 376).

The command will not change the location of the log files. You can configure your log locations to use the shared storage, but it is entirely optional.

On the other computers you installed the BigFix Server on, run the same command without the `/copydata` option.

Upgrading a clustered BigFix environment

Before upgrading a BigFix Server whose services are part of a cluster role, make sure that any program, local or remote, that may access the BigFix databases or change the BigFix data files is stopped.

Before starting the BigFix Server, Web Reports, or the WebUI in a cluster environment, you need to upgrade all those components on all computers they are installed on. Since BigFix Platform version 11.0.4, you can choose not to start those components at the end of their installation and upgrade.

HA and DR limitations

As of BigFix Platform Version 11.0.4, HA and DR configurations using Microsoft technologies are subject to certain limitations, which should be taken into account.

At no point should two computers in the same cluster run the BigFix Server (or the WebUI) at the same time. To ensure this does not happen, only start and stop services that are part of a cluster role via the cluster manager.

Operations that stop or restart a service within a cluster role without using the cluster manager can trigger a failover of that role. For example, operations that change the BigFix masthead can automatically restart the BigFix Server services, causing the role that contains them to failover. We recommend switching to a manual failover before performing such operations.

There is currently no way to change the masthead location of the BigFix Server and move it to a shared storage folder. You should find a way to synchronize it across your BigFix Server installations, so it is ready in case of a failover.

The WebUI service may frequently restart itself, especially if you have automatic updates enabled, which is the default. We do not recommend putting it in the same role as the BigFix Server services.

There is currently no recommended procedure to move the Web Reports and WebUI data files to shared storage.

Chapter 19. Upgrading

The steps to upgrade the BigFix Platform.

For more details about how to prepare for the upgrade, see Before upgrading (on page 621).

For more details about planning the upgrade, see Upgrade steps (on page 624).

Upgrade paths to BigFix 11

The following tables describe the upgrade paths to BigFix 11:

- **Server upgrade**

  Table 11. Server Upgrade

Upgrade from	Windows Upgrade	Linux Upgrade
9.5	No	No
10.0	Yes	Yes
	Note: The BigFix server must be at Version 10.0.7 or later before you can upgrade it to Version 11.0.	**Note:** The BigFix server must be at Version 10.0.7 or later before you can upgrade it to Version 11.0.

- **Client upgrade**

  Table 12. Client Upgrade

Upgrade from	Windows Upgrade	UNIX Upgrade	Mac Upgrade
9.5	Yes	Yes	Yes
10.0	Yes	Yes	Yes

Before upgrading

Things to do before upgrading.

If you are upgrading from BigFix Version 10:

- Carefully review the minimum OS requirements and Database requirements *(on page 81)* for the BigFix Server.
- Ensure that the current version you are upgrading from is at Version 10.0.7 or later.
- Enable the "Enhanced Security", as described in V11 Overview *(on page 2)*.
- BigFix clients and relays versions earlier than Version 9.5 are not supported.

Perform these steps before upgrading the BigFix components:

1. Ensure that the BigFix WebUI service is stopped before starting to upgrade the BigFix components, and do not start it again until the overall upgrade procedure is complete. Perform this step regardless of whether the WebUI is installed on the same system as the BigFix Server or on a different system.
2. Ensure that all remote BigFix Web Reports servers are stopped and do not start them again until the overall upgrade procedure is complete.
3. If you are using the Microsoft SQL Server Replication feature, you must temporarily disable it.
4. Close all BigFix consoles.
5. Back up your BigFix server and database as described in Server Backup *(on page 573)*.
6. Back up your `license.pvk`, `license.crt`, and `masthead.afxm` to a separate location on the BigFix server or to a USB key.
7. Ensure that you have enough free space for the BigFix database. The free space projection involves a number of factors: the row counts of specific tables, the density of the data pages storing the tables and indexes, and specific database options for log management and table compression.
8. Upgrade your SQL database engine, if needed.

In a Distributed Server Architecture (DSA) environment, increase the replication interval to prevent the replication from failing repeatedly during the upgrade. For additional information, see Changing the replication interval on Windows systems and Changing the replication interval on Linux systems.

If all needed requirements for automatic upgrade are satisfied, you can perform an automatic upgrade via Fixlet. Otherwise, you must perform a manual upgrade.

Upgrade prerequisite checks

The following prerequisite checks are automatically run by the upgrade procedure on the BigFix Server.

If any of these checks fails, the upgrade does not start and the procedure exits with an error message. You can rerun the upgrade after you ensure that the condition that failed is satisfied. The upgrade procedure checks that:

- The size of the Upload Manager Buffer Directory and the number of files that it contains do not exceed the 90% of the values that are specified in the following settings:

  ```
  _BESRelay_UploadManager_BufferDirectoryMaxSize
  _BESRelay_UploadManager_BufferDirectoryMaxCount
  ```

  If the check fails: Fix the issue by running the following Fixlets available in the BES Support site, and then rerun the upgrade:

  ```
  2695: WARNING: Upload Manager Directory exceeds file max number - BES
   Server
  2696: WARNING: Upload Manager Directory Full - BES Server
  ```

- The site names listed in the following columns of the database contain only ASCII characters:

  ```
  LOCAL_OBJECT_DEFS.Name
  LOCAL_OBJECT_DEFS.Sitename
  ```

  If the check fails: Rename the site names and fix the content referencing them.

- The size of the data related to custom content and the actions that are contained in the `ACTION_DEFS.Fields` and `LOCAL_OBJECT_DEFS.Fields` fields does not exceed the maximum size allowed in the new database schema.

  If the check fails: Shorten the exceeding data length.
- If you are using DB2, the table spaces hosting the following tables have enough disk space available to store the WebUI indexes:

  ```
  COMPUTERS
  VERSIONS
  DASHBOARDDATA
  EXTERNAL_ANALYSIS_PROPERTY_TRANSLATIONS
  EXTERNAL_ANALYSIS_TRANSLATIONS
  EXTERNAL_FIXLET_ACTION_SETTINGS_USER_GROUPS
  EXTERNAL_FIXLET_TRANSLATIONS
  ```

  If the check fails: Move the tables to USERSPACE1. For information about how to accomplish this task, see https://www.ibm.com/support/knowledgecenter/en/SSEPGG_10.5.0/com.ibm.db2.luw.sql.rtn.doc/doc/r0055069.html.
- The number of rows contained in the `DBINFO` database table does not exceed one.

  If the check fails: Remove the unnecessary rows from the table.
- The data contained in each of the following fields does not exceed the maximum index length:

  ```
  DASHBOARDDATA.NAME
  UNMANAGEDASSET_FIELDS_TYPES.FIELDNAME
  WEBUI_DATA.NAME
  ```

  If the check fails: Shorten the data contained in the fields as required.

If you are in a DSA environment and any of the prerequisite checks fails on the primary server, which is the first one to be upgraded, run these steps:

1. Run the needed corrective actions on the primary server.
2. Wait for a full replica on the secondary servers to ensure that the same corrective actions are applied to their databases as well.
3. After the full replica completes, rerun the upgrade on the primary server and then on the secondary servers.

Upgrade steps

Things to take into account when planning the upgrade.

You must upgrade the BigFix Platform components in a specific order.

The following components must be upgraded in rapid succession, in this order:

1. Primary BigFix Server
2. Secondary BigFix Servers, if any
3. BigFix Consoles
4. Remote BigFix Web Reports servers
5. BigFix Explorer
6. BigFix WebUI

The following components can be upgraded at a later time, in this order:

1. BigFix Relays. If they are configured in a hierarchy, upgrade the top-level ones first.
2. BigFix Plugin Portal, if any
3. BigFix Clients

A BigFix Console can only connect and operate if it has the same version as the BigFix Server. You can use a Console to start an automatic upgrade, but you should then close it and you will not be able to use it while the Server upgrades.

Servers, relays, and clients do not need to match versions and the upgrade of these components can occur at different times. Clients with earlier versions can continue to report to later versions of relays or servers, but might not have all the functionality of the later release.

During the upgrade, the versions of the different components must respect this rule: `server version >= relay version >= client version`

As a best practice, follow the instructions provided in Running backup and restore (on page 572) to make a recovery copy of your BigFix Server environment and to ensure that, if needed, you can run the rollback (on page 637).

Existing BigFix proxy configurations are automatically migrated to the V10.0 proxy configuration settings and behavior. For more information about BigFix V11.0 proxy configuration settings, see Setting up a proxy connection (on page 545).

For large deployments, the server upgrade might take several minutes.

After upgrading:

- You might experience a slower deployment. The upgrade downtime might create a backlog of client reports and it might take several hours for the BigFix server to process this backlog after the upgrade has been completed.
- If users cannot log in anymore, passwords containing non-ASCII characters might have been corrupted. If so, try resetting the password to the same value or to a new one.

Automatic upgrade

How to automatically upgrade the BigFix Platform components.

Before attempting an automatic upgrade, see Before upgrading (on page 621), as you might need to perform a few preparatory steps to ensure it succeds.

You can perform an automatic upgrade using the "BigFix - Updated Platform Server Components version x.x.x Now Available!" Fixlet, provided that it becomes relevant and that certain conditions are met.

The automatic upgrade Fixlet does not become relevant:

- if you are not running the BigFix services as LocalSystem account.
- if you are using a remote database with Windows authentication.

In this case, the "Updated Windows Server/Console Components - Manual Upgrade Required - BigFix version *x.x.x*" Fixlet should become relevant.

In a Distributed Server Architecture (DSA) environment, we recommend performing a manual upgrade.

See Manual upgrade on Windows (on page 626) and Manual upgrade on Linux (on page 633).

Manual upgrade on Windows

How to manually upgrade the BigFix Platform components on Windows.

You should run a manual upgrade if you have:

- A Distributed Server Architecture (DSA) environment
- BigFix services not running as LocalSystem account
- A BigFix Server which uses a remote database with Windows authentication
- A BigFix environment having remote WebUI or remote Web Reports server

If you have a DSA environment, ensure that:

- before starting the upgrade, all DSA servers have received the new masthead from the primary BigFix Server
- during the upgrade of a DSA server, no BigFix service is running on any of the other DSA servers, as well as any process involved in the replication

Note: The Administrator privileges are required to perform the upgrade of the server components.

Before attempting the upgrade, see Before upgrading (on page 621), as you might need to perform a few preparatory steps to ensure it succeds.

In order to perform a manual upgrade, see the following sections in the order in which they are listed below.

Upgrading the installation generator and the primary server

From the computer where you installed the BigFix Installation Generator, perform these steps.

1. Download and run the new BigFix Server Installation Generator installer from BigFix Enterprise Suite Download Center.
2. Click **Yes** when you are prompted to upgrade and follow the installer instructions.

To manually upgrade the BigFix primary server and all its components, use the Bigfix Installation Guide and perform these steps:

1. If it is not already running, launch the Installation Guide (Start > Programs > BigFix > BigFix Installation Guide).
2. A dialog box opens, prompting you to select a component to install. Click the links on the left, in order from top to bottom, to install the BigFix components. You can also Browse Install Folders. The component installers include:
    - Install Server
    - Install Console
    - Install Clients
    - Install WebUI
3. The BigFix server, console, clients and WebUI all have their own installers. Follow the instructions for each, as described in the following sections.

Upgrading a BigFix Server with a remote database

Recommended steps for upgrading a BigFix Server with a remote database.

The automatic upgrade Fixlet cannot upgrade certain configurations using a remote database. In those cases, you need to upgrade the BigFix Server manually.

Furthermore, remote database setups might encounter problems during upgrade and require resetting the database connection settings after manually running the BigFix Server installer. These steps can help you identify erroneous configurations after the upgrade.

Pre-upgrade checklist:

1. Back up your database.
2. On the computer hosting the BigFix Server, back up the ODBC registry keys by running the following commands from the Windows Command Prompt (cmd.exe):

```
reg export "HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\ODBC\ODBC.INI"
 "%UserProfile%\Desktop\ODBC.INI 32-bit.reg" /Y
reg export "HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI"
 "%UserProfile%\Desktop\ODBC.INI 64-bit.reg" /Y
```

   The 32-bit and 64-bit registry keys represent the ODBC data sources that memorize the database connection parameters. The data source names (DSN) used by the BigFix Platform are:

   enterprise_setup

   always present, used by the server installer for the upgrade

   bes_bfenterprise

   present if the BigFix Server is installed

   LocalBESReportingServer

   present if the Web Reports server is installed
3. Identify the authentication method and the account used to connect to the database. If the `bes_bfenterprise` data source contains a `Trusted_Connection` parameter set to Yes, then **Windows Authentication** is being used and the account is the same that is running the BigFix Server services. Otherwise, **SQL Server Authentication** is being used and the account is the one saved in the `User` value of the registry key `HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\BigFix\Enterprise Server\Database`
4. If you are using SQL Server Authentication, on the computer hosting the BigFix Server, backup the registry key `HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\BigFix\Enterprise Server\Database`. If the BigFix Web Reports server is installed on the same computer, also backup the registry key `HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\BigFix\Enterprise Server\FillAggregateDB`.

5. Ensure that the account used to connect to the database has the sysadmin server role. After the upgrade, you can lower its permissions, as long as it remains a db_owner of the BigFix databases.
6. If Windows Authentication is being used, it is critical that the account used to run the BigFix Server installer is the same that is used to access the database and to run the BigFix Server services.

Troubleshooting steps:

If the installation does not complete successfully, run the following steps:

1. Check the bes_bfenterprise ODBC data source. It should point to the BigFix Server database which, by default, is called BFEnterprise. Verify that it uses the same authentication mode as before the upgrade. The BESRootServer, FillDB and GatherDB services fail to connect to the database if this DSN is not configured correctly.
2. If BigFix Web Reports is installed, check the LocalBESReportingServer ODBC data source. It should point to the Web Reports database which, by default, is called BESReporting. Verify that it uses the same authentication mode as before the upgrade. You will receive an error message prior to the Web Reports login request if this is not configured correctly.
3. If you are using Windows Authentication, check that the BESRootServer, FillDB, GatherDB and BESWebReportsServer services are configured to run with the same account that they should use to connect to the database.
4. If you are using SQL Server Authentication, check that the username and password registry keys are set correctly. The passwords are encrypted. If you suspect they might be wrong, you can reset them as follows. Stop all BigFix services, change the registry keys, then restart all services. The passwords will be encrypted again when the BigFix services are back up and running.
5. If your BigFix Console gives an error message that the database has the wrong version, ensure that the BigFix Console was updated to the same version as the BigFix Server. If it was, and it still shows the same error, contact HCL support.

Upgrading the secondary server

How to manually upgrade the secondary BigFix server or any additional BigFix server.

Upgrading the secondary/additional server

You can upgrade the BigFix secondary server by copying the BigFix Server installer folder to the remote computer that is running the BigFix Server, replacing the file masthead.afxm in the installer folder with the updated masthead, and then running setup.exe.

1. Copy the BigFix Server installer folder to the remote BigFix Server computer. The default location of the BigFix Server installer folder is `%PROGRAMFILES(x86)%\BigFix Enterprise\BES Installers\Server`.

   Note: If you have a remote database, prior to upgrading see Upgrading a BigFix Server with a remote database *(on page 627)*.

2. Replace the file `masthead.afxm` in the installer folder with the current deployment masthead. The replacement file can be downloaded from *https://servername:port/masthead/masthead.afxm* or copied from the BigFix Server computer, where it is named `ActionSite.afxm` and saved in the folder `%PROGRAMFILES(x86)%\BigFix Enterprise\BES Server`. After placing the new file in the installer folder, ensure it is named `masthead.afxm`, renaming it if necessary.
3. Run the BigFix Server installer (`setup.exe`) on the BigFix Server computer.

   Note: Before starting to upgrade, the upgrade procedure runs a set of prerequisite checks. If any of them fails, you can do the required corrective actions and then rerun the upgrade procedure. For more information, see Upgrade prerequisite checks *(on page 622)*.

4. Follow the installer instructions to upgrade.
5. Run the Administration Tool `BESAdmin.exe` to distribute the updated license.

Note:

1. Before upgrading, it is advisable to delete or rename the `server_audit.log` file to avoid inaccurate interpretation of file encoding.
2. The License Key Password cannot contain double quotes and cannot be longer than 35 characters.

Upgrading the Console

How to manually upgrade the Console.

1. Copy the BigFix Console installer folder from the BigFix primary server to all computers that are running the BigFix Console. The default location for the BigFix Console installer folder is `%PROGRAMFILES(x86)%\BigFix Enterprise\BES Installers\Console`.
2. Replace the file `masthead.afxm` in the installer folder with the current deployment masthead. The replacement file can be downloaded from https://servername:port/masthead/masthead.afxm or copied from the BigFix Server computer, where it is named `ActionSite.afxm` and saved in the folder `%PROGRAMFILES(x86)%\BigFix Enterprise\BES Server`. After placing the new file in the installer folder, ensure it is named `masthead.afxm`, renaming it if necessary.
3. Run the BigFix Console installer (`setup.exe`) on all the computers currently running the BigFix Console.

Upgrading the remote WebUI

How to manually upgrade the remote WebUI.

1. Copy the BigFix WebUI installer (default location is `%PROGRAMFILES(x86)%\BigFix Enterprise\BES Installers\WebUI`) from the BigFix primary server to the client computer that is running the BigFix WebUI.
2. Stop the WebUI Service.
3. Run the installer on the computer hosting the WebUI. The installer detects the WebUI and offers to upgrade it for you.

4. Follow the installer instructions to upgrade the BigFix WebUI.
5. Start the WebUI Service.

Upgrading the Relays

How to manually upgrade the Relays.

To upgrade the BigFix Relays from the BigFix console, apply the **Updated Windows Relay** Fixlet to all relevant relays.

Note: The Fixlets might take some time to be reported as relevant when the server has been upgraded to a new "major version". For example, when the server has been upgraded from Version 10 to Version 11.

Upgrading the Clients

How to manually upgrade the Clients.

You can upgrade the BigFix Clients by copying the BigFix Client installer folder from the BigFix primary server to each computer that is running the BigFix Client.

Replace the file `masthead.afxm` in the installer folder with the current deployment masthead. The replacement file can be downloaded from https://servername:port/masthead/masthead.afxm or copied from the BigFix Server computer, where it is named `ActionSite.afxm` and saved in the folder `%PROGRAMFILES(x86)%\BigFix Enterprise\BES Server`. After placing the new file in the installer folder, ensure it is named `masthead.afxm`, renaming it if necessary.

The default location for the BigFix Client installer folder is `C:\Program Files\BigFix Enterprise\BES Installers\Client`.

Upgrading the remote Web Reports server

How to manually upgrade the remote Web Reports server.

To upgrade a stand-alone Web Reports server on Windows, copy the BigFix Server installer folder from the BigFix primary server to the remote BigFix Web Reports computer. The

default location of the BigFix Server installation folder is `%PROGRAMFILES(x86)%\BigFix Enterprise\BES Installers\Server`.

Replace the file `masthead.afxm` in the installer folder with the current deployment masthead. The replacement file can be downloaded from https://servername:port/masthead/masthead.afxm or copied from the BigFix Server computer, where it is named `ActionSite.afxm` and saved in the folder `%PROGRAMFILES(x86)%\BigFix Enterprise\BES Server`. After placing the new file in the installer folder, ensure it is named `masthead.afxm`, renaming it if necessary.

Run the `BigFix-BES-Server` setup.exe, which detects the Web Reports installation and offers to upgrade it for you.

If Web Reports is installed on the same computer as the BigFix server, the installer upgrades them together.

Note: The user running the upgrade must have DBO permissions on the BigFix databases.

Manual upgrade on Linux

How to manually upgrade the BigFix Platform components on Linux.

You should run a manual upgrade if you have:

- A Distributed Server Architecture (DSA) environment
- A BigFix Server which uses a remote database
- A BigFix environment having remote WebUI or remote Web Reports server

If you have a DSA environment, ensure that:

- before starting the upgrade, all DSA servers have received the new masthead from the primary BigFix Server
- during the upgrade of a DSA server, no BigFix service is running on any of the other DSA servers, as well as any process involved in the replication

Note: The root privileges are required to perform the upgrade of the server components. The 'sudo' utility cannot be used.

Before attempting the upgrade, see Before upgrading (on page 621), as you might need to perform a few preparatory steps to ensure it succeds.

In order to perform a manual upgrade, see the following sections in the order in which they are listed below.

Upgrading the server

Do the following steps to upgrade the server:

1. Copy the BigFix installable image to the BigFix server computer and extract it to a folder.
2. On the BigFix server computer, run the BigFix server upgrade script:

```
./install.sh -upgrade [-opt BES_LICENSE_PVK=<path+license.pvk>]
             [-opt BES_LICENSE_PVK_PWD=<password>]
```

where:

`-opt BES_LICENSE_PVK=<path+license.pvk>`

Specifies the private key file (*`filename`*`.pvk`). This private key file and its password are required to update the product license and perform the required SHA-256 signature updates in the BigFix database.

Note: The notation `<path+license.pvk>` used in the command syntax stands for *`path_to_license_file`*`/license.pvk`.

`-opt BES_LICENSE_PVK_PWD=<password>`

Specifies the password associated to the private key file (*`filename`*`.pvk`).

The use of the optional parameters `BES_LICENSE_PVK` and `BES_LICENSE_PVK_PWD` depends on the current release or patch level that the installer is upgrading and, in the event the upgrade procedure requires to sign again the database, they are explicitly asked for during the upgrade process. As an alternative, you can specify them anyway and, if they are not required, they are ignored by the upgrade process.

The `install.sh` server script upgrades all the components it detects on the local server.

If a Web UI instance previously installed with a Fixlet is detected, also the Web UI component is upgraded.

Note: Before starting to upgrade, the upgrade procedure runs a set of prerequisite checks. If any of them fails, you can do the required corrective actions and then rerun the upgrade procedure. For more information, see Upgrade prerequisite checks *(on page 622)*.

3. Run the Administration Tool (`./BESAdmin.sh` on Linux) to distribute the updated license:

```
/opt/BESServer/bin/BESAdmin.sh -syncmastheadandlicense
 -sitePvkLocation=<path+license.pvk>
                    -sitePvkPassword=<password>
```

Note:

1. For troubleshooting information see `/var/log/BESInstall.log` and `/var/log/BESAdminDebugOut.txt` files.
2. After upgrading the server to Version 9.5.5, due to data movement in the database, the Web UI must select and get new data. This might temporarily prevent the FillDB process from processing the client reports.

Upgrading the Console

Do the following steps to upgrade the console:

1. Copy the BigFix console installation folder (default is: `/var/opt/BESInstallers/Console`) to all Windows computers that are running the BigFix console.
2. Run the BigFix console installer (`setup.exe`) on all the Windows computers currently running the BigFix console.



Note: The BigFix console does not run on Linux computers.

Upgrading the relays

To upgrade the BigFix relays from the BigFix console, apply one of the following Fixlets, depending on the operating system you are using:

- Updated AIX Relay
- Updated Amazon Linux 2 Relay
- Updated CentOS Linux Relay
- Updated Raspbian Linux Relay
- Updated Red Hat Enterprise Linux Relay
- Updated SuSE Linux Enterprise Relay
- Updated Tinycore Linux Relay
- Updated Ubuntu Linux Relay



Note: The Fixlets might take some time to be reported as relevant when the server has been upgraded to a new "major version". For example, when the server has been upgraded from Version 10 to Version 11.

Upgrading the clients

You can upgrade the BigFix clients in several ways:

- Upgrade BigFix clients individually by copying the BigFix client installable image to each computer that is running the BigFix client, and then running the setup program as follows:

  ```
  rpm -U xxx.rpm
  ```

  where *xxx* is the name of the client installable image.
- Upgrade the BigFix clients by using the BigFix Client Deployment Tool, with a log in script, or with another deployment technology. Simply run the new BigFix Client installer on the computer with the old BigFix client.

Upgrading the Web Reports and WebUI standalone servers

On Linux, to upgrade a stand-alone Web Reports or WebUI server (or both), download the `ServerInstaller_10.x.x-rhe6.x86_64.tgz` installer archive, decompress it and run the install.sh server upgrade script:

```
./install.sh -upgrade
```

To upgrade a standalone BigFix root server, WebUI, or Web Reports server, you can also use the Fixlet BigFix - Updated Platform Server Components version 10.x.x.

Rollback

How to roll back your BigFix Server if you made a backup.

If you made a recovery copy of your BigFix Server data and configuration as described in Server Backup (on page 573) (Windows) or Server Backup (on page 584) (Linux), you can roll back your BigFix Server to its original status, before running the upgrade, following the instructions provided in Server Recovery (on page 576) (Windows) or Server Recovery (on page 587) (Linux).

Chapter 20. SQL Server parallelism optimization

The performance of an SQL Server database instance can often be improved by small tweaks. Performance might also be hindered by simple oversights. In fact, some SQL Server parallelism settings have suboptimal default values. Moreover, they have to be re-tuned after an hardware upgrade. Other issues might arise from inadvertent hardware configurations, especially when SQL Server is hosted on a virtual machine (VM).

In particular, it is beneficial to customize these instance settings:

- maximum degree of parallelism (MaxDoP)
- cost threshold for parallelism (CTFP)

Starting from BigFix 10.0.2, you can use the `/checksqlserverparallelism` BESAdmin command to check if the MaxDoP and CTFP settings of your database instance are configured appropriately, and to detect other issues described later.

SQL Server MaxDoP values for best performance

Microsoft recommends MaxDoP settings in **Configure the max degree of parallelism Server Configuration Option**.

Server configuration	Number of processors	SQL Server 2008-2014 (10.x-12.x)	SQL Server 2016 (13.x) and newer
Server with single NUMA node	Fewer than or equal to 8 logical processors	Keep MAXDOP at or below the number of logical processor	Keep MAXDOP at or below the number of logical processors
Server with single NUMA node	More than 8 logical processors	Keep MAXDOP at 8	Keep MAXDOP at 8

Server with multiple* NUMA nodes	Fewer than or equal to 8 logical processors per NUMA node	Keep MAXDOP at or below the number of logical processors per NUMA node	Keep MAXDOP at or below the number of logical processors per NUMA node
Server with multiple* NUMA nodes	More than 8 logical processors per NUMA node	Keep MAXDOP at 8	Keep MAXDOP at half the number of logical processors per NUMA node with a MAX value of 16

* The number of NUMA nodes refers to the quantity of software NUMA (soft-NUMA) nodes, if the soft-NUMA feature is enabled, and to the total hardware NUMA nodes otherwise.

On SQL Server 2016 (13.x) and later versions, the soft-NUMA feature is enabled by default and set to automatically split hardware NUMA nodes with more than 8 logical processors into smaller soft-NUMA nodes.The soft-NUMA feature can be configured so that soft-NUMA nodes are created manually, or it can be completely disabled. For more information, refer to **Soft-NUMA (SQL Server)**.

MaxDoP changes do not require that you restart the SQL Server 2012 or 2019.

SQL Server CTFP values for best performance

Microsoft does not provide recommendations for setting the CTFP.

A common suggestion is setting it to a value between 15 and 50, with the understanding that the best value for it depends on the workload.

For the database workload that the BigFix Server generates, test results show that setting CTFP to 50 yields better performance than leaving it at 5.

CTFP changes do not require that you restart SQL Server 2012 and later versions such as 2019.

When you are installing the BigFix Server using Amazon RDS for Microsoft SQL Server 2022, the setup can recommend optimized values for the MaxDoP and CTFP parameters, but cannot apply them automatically. You can manually configure the SQL Server parallelism

settings we recommend via the Amazon Dashboard by creating a custom “parameter group” for your RDS server and specifying the parameter values in it.

Troubleshooting scenario 1: Under-utilization of licensed cores in a VM

Because of a licensing restriction, SQL Server might not be able to use all available CPU cores.

In particular, the license of some SQL Server editions (Express, Web and Standard) is "limited to the lesser of n sockets or m cores."

For example, a database instance of SQL Server 2019 Express Edition is "limited to the lesser of 1 socket or 4 cores" for its maximum compute capacity. See the Scale Limits section of Editions and supported features of SQL Server 2019 (15.x). Because of that, this edition of SQL Server can use only 4 cores on the same socket.

This limitation might lead to unexpected issues when SQL Server is installed on a virtual machine. In fact, using a common VM configuration (that uses many virtual sockets with few cores per socket) can severely limit the number of cores that SQL Server can use, because of the SQL Server license limitations.

In the example, SQL Server 2019 Express can use up to 4 cores, but if it is installed on a VM with 4 cores and 4 sockets (1 core per socket), it can use a single core.

In another example, the SQL Server 2019 Web license lets you use "the lesser of 4 sockets or 16 cores".

If your VM has 16 (virtual) sockets and 1 core per socket, you can only use 4 cores out of 16. That is, you will use 4 sockets with 1 core each.

However, SQL Server can use all cores if you change your VM CPU configuration to use, for example, 4 sockets and 4 cores per socket.

When you create a new VM, carefully choose the number of CPU sockets and the number of total CPU cores.

If you are using the VMware vSphere Client, when creating a VM or editing its settings, you can expand the "CPU" menu of the "Virtual Hardware" tab to configure the number of "cores per socket", which determines the number of sockets.

To detect whether SQL Server is running in a VM, you can run this query and check for the following result:

`virtual_machine_type >= 1`.

```
SELECT virtual_machine_type
FROM sys.dm_os_sys_info
```

To detect how many cores (logical processors) SQL Server can use, run this query:

```
select COUNT(*) AS sqlUsedLogicProcs
from sys.dm_os_schedulers
where status = 'VISIBLE ONLINE'
```

The total number of logical processors that SQL Server can detect (but not necessarily use) is returned by this query:

```
SELECT cpu_count AS LogicalCpuCount
FROM sys.dm_os_sys_info
```

To find the active SQL Server license (edition), run this query:

```
SELECT SERVERPROPERTY( 'edition' )
```

Microsoft provides complete list of returned values in SERVERPROPERTY (Transact-SQL). Ignore the Azure values.

Troubleshooting scenario 2: Uneven distribution of used cores

Even when SQL Server can use all licensed cores, performances might not be optimal.

Ideally, SQL Server is licensed to use all cores on the (virtual or physical) hardware of the computer.

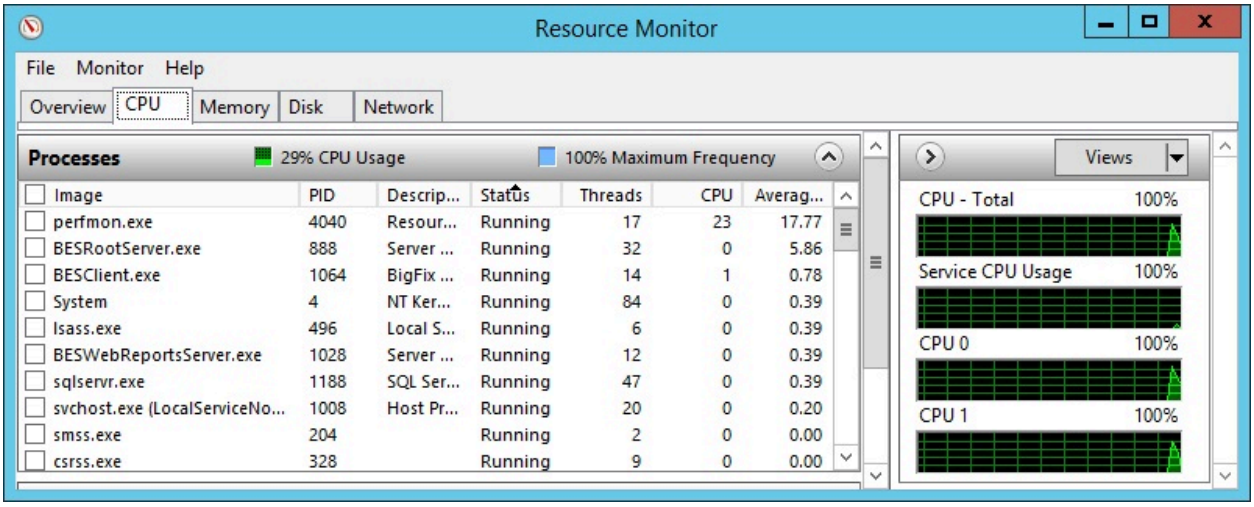
If SQL Server cannot use all cores on the computer, the impact on performance is smaller when the cores it can use are evenly distributed among the hardware NUMA nodes of that computer. If SQL Server can use only a small portion of the available cores, which is not ideal, it can use only the cores on the NUMA nodes that host the n sockets that it is licensed to use.

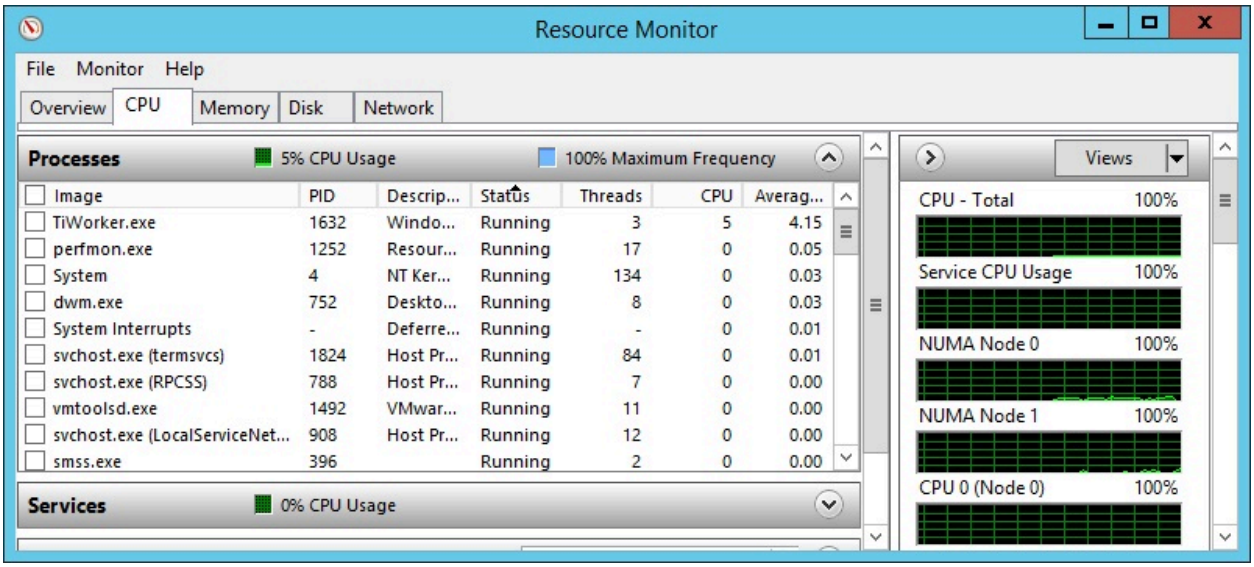
On physical hardware, the number of hardware NUMA nodes usually matches the number of sockets, or, less commonly, is a multiple of it. For example, a physical socket can contain one or more NUMA nodes.

On virtual hardware, the opposite can happen. The number of hardware NUMA nodes can be smaller than the number sockets. That is, multiple sockets can be part of the same hardware NUMA node.

On Windows, you can use Resource monitor (resmon.exe) to check the number of hardware NUMA nodes on your computer.

In the "CPU" tab, the panel on the right shows a graph for each NUMA node and CPU processor.





If the panel shows only CPU graphs, that means there is only one NUMA node that hosts all CPUs.

Alternatively, the following query returns the total number of hardware NUMA nodes on the computer that hosts SQL Server:

```
select COUNT( DISTINCT memory_node_id ) as hwNumaNodes
from sys.dm_os_memory_nodes
where memory_node_id <> 64
```

In SQL Server 2016 and later, the automatic soft-NUMA feature splits virtual or physical hardware NUMA nodes with more than 8 cores into multiple soft-NUMA nodes. This split does not necessarily solve the performance degradation that unevenly assigned cores to different hardware NUMA nodes causes; it might in fact only mask it.

You can use this query to detect the number of logical processors that are used on the software or hardware NUMA nodes in use:

```
select COUNT(*) as usedNumaNodes,
MIN(online_scheduler_count) as minUsedLogicProcsPerNumaNode,
MAX(online_scheduler_count) as maxUsedLogicProcsPerNumaNode
from sys.dm_os_nodes
where online_scheduler_count > 0 and node_state_desc not like '%DAC%'
```

An example scenario is SQL Server Web on a computer with 2 sockets and 20 cores (10 per socket).

 **Remember:** SQL Server Web can use the lesser of 4 sockets or 16 cores.

In this setup, SQL Server Web can use all the 16 licensed cores out of the total 20 that the system provides.

However, how the 16 used cores are chosen among the total of 20 can make a difference in performance.

Assuming there is a NUMA node for each socket, the used cores might be unevenly distributed, like this example:

- 10 cores in NUMA node 0
- 6 cores in NUMA node 1

This example shows a better distribution of the used cores:

- 8 cores in NUMA node 0
- 8 cores in NUMA node 1

The distribution of used cores between NUMA nodes depends on how the CPU affinity mask is set. It can be changed using this command:

```
ALTER SERVER
CONFIGURATION SET PROCESS AFFINITY CPU
```

For more information, refer to the "Setting process affinity" section of this Microsoft article: ALTER SERVER CONFIGURATION (Transact-SQL)

Gathering additional information

To gather additional information, you use BESAdmin and pass the `/extrainfo` flag to the `/checksqlserverparallelism` command.

Also, you can inspect the SQL Server logs to extract useful details.

If no output is returned, the log lines of interest might be deleted by the log rotation.

This query inspects the logs and looks for the total number of sockets and cores and the quantity of cores that is used in accordance with the SQL Server license:

```
SET NOCOUNT ON;
DECLARE @logData TABLE( LogDate DATETIME, ProcInfo NVARCHAR(64), LogText
 NVARCHAR(1024) );
INSERT INTO @logData
EXEC sys.xp_readerrorlog 0, 1, N'SQL Server detected ', N' socket', null,
 null, N'DESC';
SELECT TOP 1 [LogText]
FROM @logData;
```

Example output:

```
SQL Server detected 1 sockets with 2 cores per socket and 2 logical processors per socket, 2 total logical processors; using 2 logical processors based on SQL Server licensing. This is an informational message; no user action is required.
```

This query inspects the logs and looks for the CPU mask that is used to choose the CPU cores to use on each NUMA node:

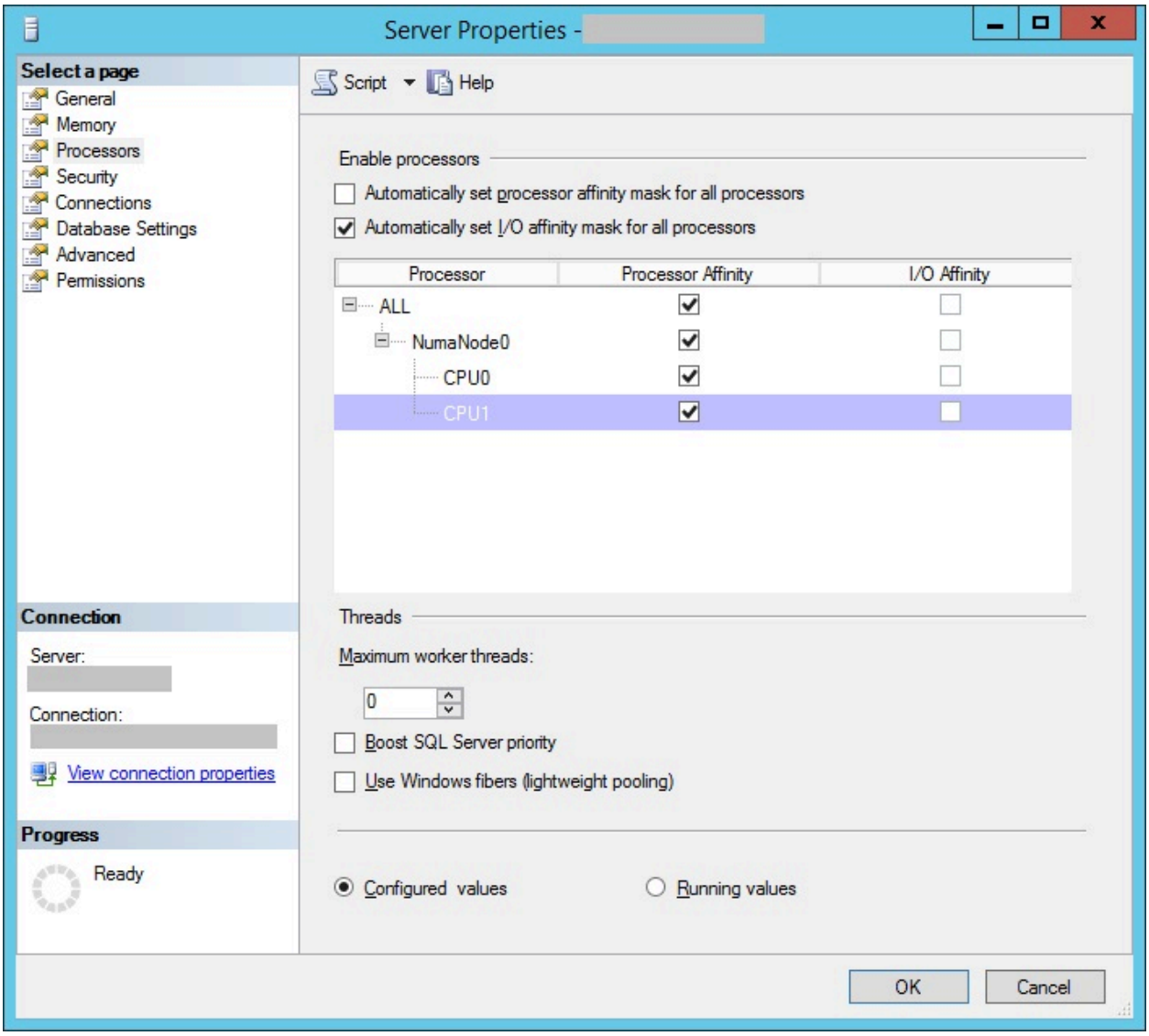
```
SET NOCOUNT ON;
DECLARE @logData TABLE( LogDate DATETIME, ProcInfo NVARCHAR(64), LogText
 NVARCHAR(MAX) );
INSERT INTO @logData
EXEC sys.xp_readerrorlog 0, 1, N'Node configuration: ', N' CPU mask: ',
 null, null, N'DESC';
SELECT [LogText]
FROM @logData;
LogText
```

Example output:

```
Node configuration: node 0: CPU mask: 0x0000000000000003:0 Active CPU mask: 0x0000000000000003:0. This message provides a description of the NUMA configuration for this computer. This is an informational message only. No user action is required.
```

If the CPU mask was set manually, you can view it by using SQL Server Management Studio.

Right-click your DB instance, click **Properties**, and then click **Processors**.


Server Properties -
Select a page
General
Memory
Processors
Security
Connections
Database Settings
Advanced
Permissions
Script
Help
Enable processors
Automatically set processor affinity mask for all processors
Automatically set I/O affinity mask for all processors
Processor
Processor Affinity
I/O Affinity
ALL
NumaNode0
CPU0
CPU1
Threads
Maximum worker threads:
0
Boost SQL Server priority
Use Windows fibers (lightweight pooling)
Configured values
Running values
Connection
Server:
Connection:
View connection properties
Progress
Ready
OK
Cancel

Chapter 21. Known limitations and workarounds

This section describes the known limitations and possible workarounds.

Copying VMware machines into another vCenter

Known limitation: When you copy into a vCenter VMware machines, that were created on another vCenter, this operation may create InstanceUUID duplications and cause wrong correlated representations.

Workaround: Create a new unique InstanceUUID for the VMware machine you migrated from one vCenter to another.

Client and Session relevance expressions with a large number of elements

Known limitation: The evaluation of client or session relevance expressions containing a large number of elements could be expensive, and result in the crash of the process running them (Client, Plugin Portal, FixletDebugger, QnA, WebReports, and so on) depending on the hardware or software resources of the machine.

Workaround: Define the client or session relevance expressions with a limited number of elements inside. For example, avoid the use of relevance expressions containing an high number of logical conditions or hundreds of elements in a set.

Computer Name for Windows is limited to 15 characters

Known limitation: The Computer Name property for the Windows agents retrieves the Netbios name of the computer, which is limited to 15 characters. See http://support.microsoft.com/kb/909264.

Workaround: If you want something different from the Netbios name on Windows, you must use a different Inspector (retrieved property). You can use such properties as host name, dns name to provide additional values as needed.

License Key Password is limited to 35 characters

Known limitation: The License Key Password is limited to 35 characters and cannot contain double quotes.

Workaround: Avoid using the double quotes and ensure that the License Key Password is not longer than 35 characters.

Visual C++ 2012 Redistribution Packages installation requirement

Known limitation: Visual C++ 2012 Redistribution Packages must be installed on the BigFix Client machine before deploying the task ID 978 "Enable Encryption for Clients" of the "BES Support" site.

Workaround: None.

Changing the domain of AD/LDAP operators can cause issues on Web Reports

Known limitation: Modifying the domain of AD/LDAP non-master operators causes Web Reports to manage the AD/LDAP user as a different operator. The new operator will not have access to the private reports and to other private objects associated to the original operator.

Workaround: None.

SQL Server Database collation

Known limitation: Some SQL Server collation may not work with BigFix due to incompatible chars used in table names. Known collation not working:

- SQL_Latin1_General_CP1254_CI_AS

Workaround: Use an alternative collation, such as SQL_Latin1_General_CP1_CI_AS.

Running REST API query involving plus symbols '+'

Known limitation: Running through a REST API query, a relevance expression with plus symbols '+' is not correctly evaluated. The limitation impacts both

Web Reports and BigFix Explorer when they are used to resolve the REST API query.

Workaround: Change the plus symbol '+' with the corresponding percent encoded string '%2B'. For example, the following relevance expression containing the string 'Notepad++' is not able to match Fixlets having this string in the name:

```
names of fixlets whose ( name of it contains "Notepad++" ) of be
s sites
```

To make the relevance expression work, it needs to be modified as follows, replacing all '+' characters with '%2B':

```
names of fixlets whose ( name of it contains "Notepad%2B%2B" ) o
f bes sites
```

Cannot login to Web Reports if the SID length overcomes 255 characters

Known limitation: When performing the login on Web Reports, with a remote AD/LDAP operator, the SID length of a user group cannot overcome the limit of 255 ASCII characters.

Workaround: Modify the user group SID length, at AD/LDAP level, to ensure that the request does not overcome the limit of 255 ASCII characters.

Related information

Limitations in Client Deploy Tool *(on page 308)*

Known limitations *(on page 335)*

Appendix A. Logging paths

This page lists the default paths of the BigFix log files relative to: installation, upgrade, running components, plugins, and audit.

For information detailing how to enable, disable, and configure these logs, see Configuring logs.

Running components logs

Root Server log

- Windows: `C:\Program Files (x86)\BigFix Enterprise\BES Server\BESRelay.log`
- Linux: `/var/log/BESRelay.log`

Server audit log

- Windows: `C:\Program Files (x86)\BigFix Enterprise\BES Server \server_audit.log`
- Linux: `/var/opt/BESServer/server_audit.log`

FillDB log

- Windows: `C:\Program Files (x86)\BigFix Enterprise\BES Server\FillDBData \FillDB.log`
- Linux: `/var/opt/BESServer/FillDBData/FillDB.log`

GatherDB log

- Windows: `C:\Program Files (x86)\BigFix Enterprise\BES Server\GatherDBData \GatherDB.log`
- Linux: `/var/opt/BESServer/GatherDBData/GatherDB.log`

Gather Status Report

- Windows: `http://127.0.0.1:52311/cgi-bin/bfenterprise/BESGatherMirrorNew.exe`
- Linux: Status Report does not exist.

Relay log

- Windows: `C:\Program Files (x86)\BigFix Enterprise\BES Relay\logfile.txt`
- Linux: `/var/log/BESRelay.log`

BESAdmin log

- On **Windows**, there are two or more BESAdmin log files:
  - For each user that runs BESAdmin, there is a specific log file

    `C:\Users\<USERNAME>\AppData\Local\BigFix\BESAdminDebugOut.txt`

    For example: `C:\Users\Administrator\AppData\Local\BigFix\BESAdminDebugOut.txt`
  - When BESAdmin is called by a Fixlet (or run as LocalSystem), the log file is

    `C:\Windows\System32\config\systemprofile\AppData\Local\BigFix\BESAdminDebugOut.txt`
- On **Linux** systems, there is a single log file, located in the following folder:
  - `/var/log/BESAdminDebugOut.txt`

BESAdmin audit log

- On **Windows**, there are two or more BESAdmin log files:
  - For each user that runs BESAdmin, there is a specific log file

    `C:\Users\<USERNAME>\AppData\Local\BigFix\besadmin_audit.log`

    For example: `C:\Users\Administrator\AppData\Local\BigFix\besadmin_audit.log`
  - When BESAdmin is called by a Fixlet (or run as LocalSystem), the log file is

    `C:\Windows\System32\config\systemprofile\AppData\Local\BigFix\besadmin_audit.log`

- On **Linux** systems, there is a single log file, located in the following folder:
  - `/var/log/besadmin_audit.log`

BESTools log

- Windows: `C:\Program Files (x86)\BigFix Enterprise\BES Server\BESTools.log`
- Linux: `/var/log/BESTools.log`

BES Server Plugin Service log

The directory of the log file is `C:\Program Files\BigFix Enterprise\BES Server \Applications\Logs`.



Note: If you change the name or the path of log files, to avoid character display problems, ensure to use names that have only ASCII characters and not non-ASCII characters.

Web Reports database initialization log

- Windows: `C:\Users\<USERNAME>\AppData\Local\BigFix \BESInitializeBESReportsDB.txt`

  For example: `C:\Users\Administrator\AppData\Local\BigFix \BESInitializeBESReportsDB.txt`
- Linux: `/var/log/BESInitializeBESReportsDB.txt`

Web Reports server log

- Windows: `C:\Program Files (x86)\BigFix Enterprise\BES Server \BESWebReports.log`
- Linux: `/var/log/BESWebReports/BESWebReports.log`

The Web Reports log is only enabled by default since Version 11.0.6.

BigFix Explorer log

- Windows: `C:\Program Files (x86)\BigFix Enterprise\BES Explorer \BESExplorer.log`
- Linux: `/var/log/BESExplorer.log`

BigFix MCP Server application log

- Windows: `C:\Program Files (x86)\BigFix Enterprise\BES MCP Server \BESMCPServer.log`
- Linux: `/var/log/BESMCPServer.log`

BigFix MCP Server audit log

- Windows: `C:\Program Files (x86)\BigFix Enterprise\BES MCP Server\data \mcp_audit.log`
- Linux: `/var/opt/BESMCPServer/data/mcp_audit.log`

Plugin Portal log

The BigFix Plugin Portal log file is located in the following default paths.

- Windows: `C:\Program Files (x86)\BigFix Enterprise\BES Plugin Portal \BESPluginPortal.log`
- Linux: `/var/log/BESPluginPortal.log`

Each cloud plugin stores its log files in its own folder.

- 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`

WebUI logs

The BigFix WebUI has several logs. There is a single log for the service and several for the NodeJS processes.

BES WebUI Service application log

- Windows: `C:\Program Files (x86)\BigFix Enterprise\BES WebUI\service-wrapper.log`
- Linux: `/var/opt/BESWebUI/service-wrapper.log`

WebUI NodeJS processes log folder

- Windows: `C:\Program Files (x86)\BigFix Enterprise\BES WebUI\WebUI\logs`
- Linux: `/var/opt/BESWebUI/WebUI/logs`

Client log

The BigFix Client records its current activity into a log file with the current date as the file name in the format `[year][month][day].log`. If an active log reaches 512K in size it will be moved to a backup (.bkg) file and a new log will be started for the current day. If the log reaches 512K again the backup will overwrite the existing backup. Both the active and backup logs will be deleted after ten days. These are the default locations of the BigFix Client logs for each operating system:

- Windows: `C:\Program Files\BigFix Enterprise\BES Client\__BESData\__Global\Logs`
- UNIX, Linux: `/var/opt/BESClient/__BESData/__Global/Logs`
- Mac: `/Library/Application Support/Bigfix/BES Agent/__BESData/__Global/Logs`

Fixlet installation and upgrade logs

If you install or upgrade a BigFix component using a Fixlet, the path of the installation log is determined by the Fixlet.

The paths of the setup logs of the installation Fixlets are:

- `{BigFix Client folder}\BesClientDeployToolInstall.log`, for the Client Deploy Tool (CDT)
- `{BigFix Client folder}\BesConsoleInstall.log`, for the Console
- `{BigFix Client folder}\BesPluginPortalInstall.log`, for the Plugin Portal
- `{BigFix Client folder}\BesExplorerInstall.log`, for the Explorer

- `{BigFix Client folder}\BesRelayInstall.log`, for the Relay
- `{BigFix Client folder}\BesWebUiInstall.log`, for the WebUI Service

The paths of the setup logs of the upgrade Fixlets are:

- `C:\BesInstallationGeneratorUpgrade.log`
- `C:\BesServerUpgrade.log`, for the Server and local Web Reports
- `C:\BesWebReportsUpgrade.log`, for standalone (remote) Web Reports
- `{BigFix Client folder}\BesClientDeployToolInstall.log`, for the Client Deploy Tool (CDT)
- `{BigFix Client folder}\BesClientUpgrade.log`, for the Windows Client
- `/Library/Logs/BESAgent.log`, for the Mac Client
- `{BigFix Client folder}/BesClientUpgrade.log`, for all the others Clients
- `{BigFix Explorer folder}\BesExplorerUpgrade.log`, for the Explorer
- `{BigFix Relay folder}\BesRelayUpgrade.log`, for the Relay
- `{BigFix Server API folder}\BesServerApiUpgrade.log`, for the Server API
- `{BigFix WebUI folder}\BesWebUiUpgrade.log`, for the WebUI Service

Manual installation and upgrade logs on Windows

Before BigFix 10.0.8, no log is created by default when manually running a BigFix installer on Windows.

Such logs can still be created by passing specific options to the installers or by changing the default Windows settings as described here to globally enable the installation logging.

Starting from BigFix 10.0.8, if you run an .exe setup manually to install or upgrade a BigFix component, the installation log will be saved in the `%LocalAppData%` folder.

When .exe setups are run manually, their logs are named:

- `BesClientInstall.log`, for the Client
- `BesConsoleInstall.log`, for the Console
- `BesInstallationGeneratorInstall.log`, for the Installation Generator
- `BesPluginPortalInstall.log`, for the Plugin Portal
- `BesRelayInstall.log`, for the Relay

- `BesServerApiInstall.log`, for the Server API
- `BesServerInstall.log`, for the Server and/or Web Reports
- `BesWebUiInstall.log`, for the WebUI Service

If you run an .msi setup manually to install or upgrade a BigFix component, the installation log will be saved in the `%temp%` folder. Their names are randomly generated but follow the pattern: `Msi*.log`.

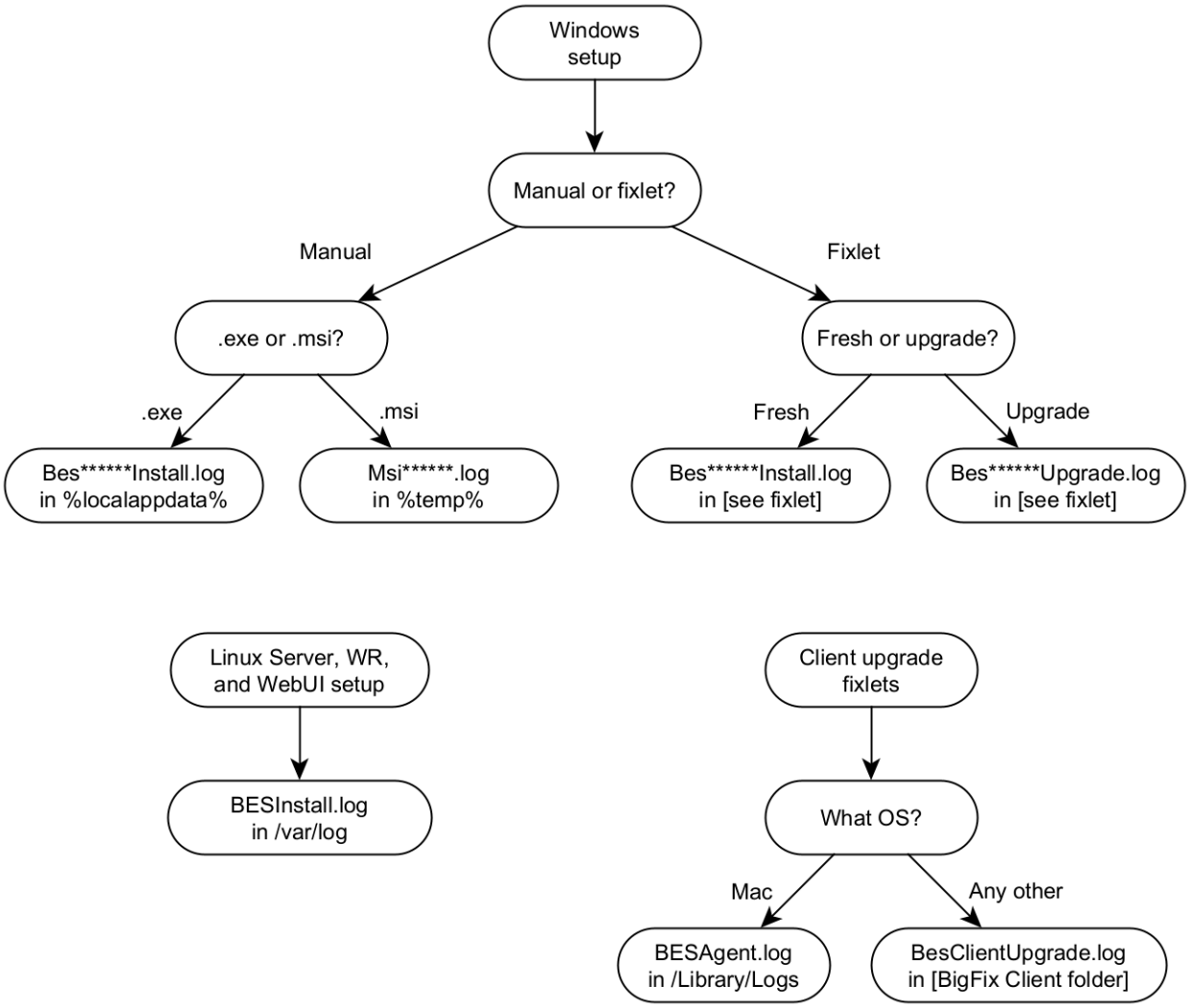
Manual installation and upgrade logs on Linux

On Linux, the BigFix Server installer always produces a log by default and saves it in `/var/log/BESInstall.log`.

Remember that the BigFix Server installer can install several components at once: the Server, Web Reports, the WebUI and the Client.

The Client upgrade Fixlets create a log since BigFix Version 10 Patch 8.

Chart of installation log locations


Windows setup
Manual or fixlet?
Manual
Fixlet
.exe or .msi?
Fresh or upgrade?
.exe
.msi
Fresh
Upgrade
Bes******Install.log in %localappdata%
Msi******.log in %temp%
Bes******Install.log in [see fixlet]
Bes******Upgrade.log in [see fixlet]
Linux Server, WR, and WebUI setup
BESInstall.log in /var/log
Client upgrade fixlets
What OS?
Mac
Any other
BESAgent.log in /Library/Logs
BesClientUpgrade.log in [BigFix Client folder]

Appendix B. Uninstalling the BigFix client

To uninstall the BigFix client installed on the various operating systems, see the following sections.

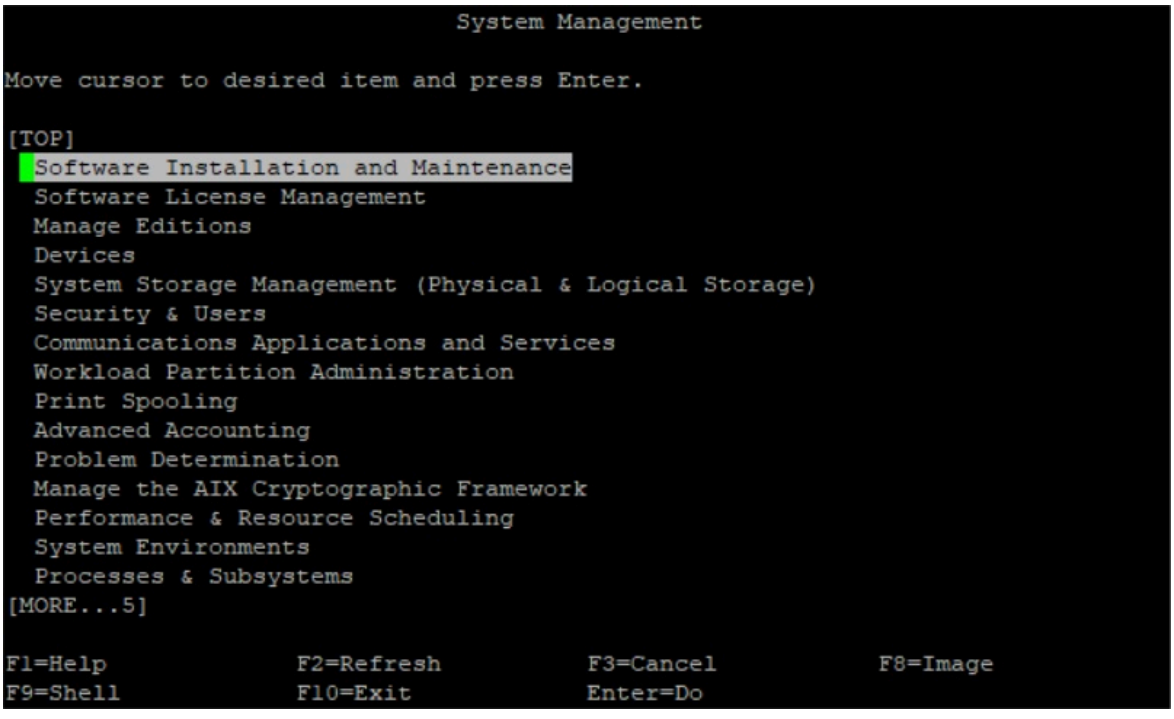
Uninstalling the BigFix Client on AIX

To uninstall the BigFix client installed on an AIX system, run the following steps:

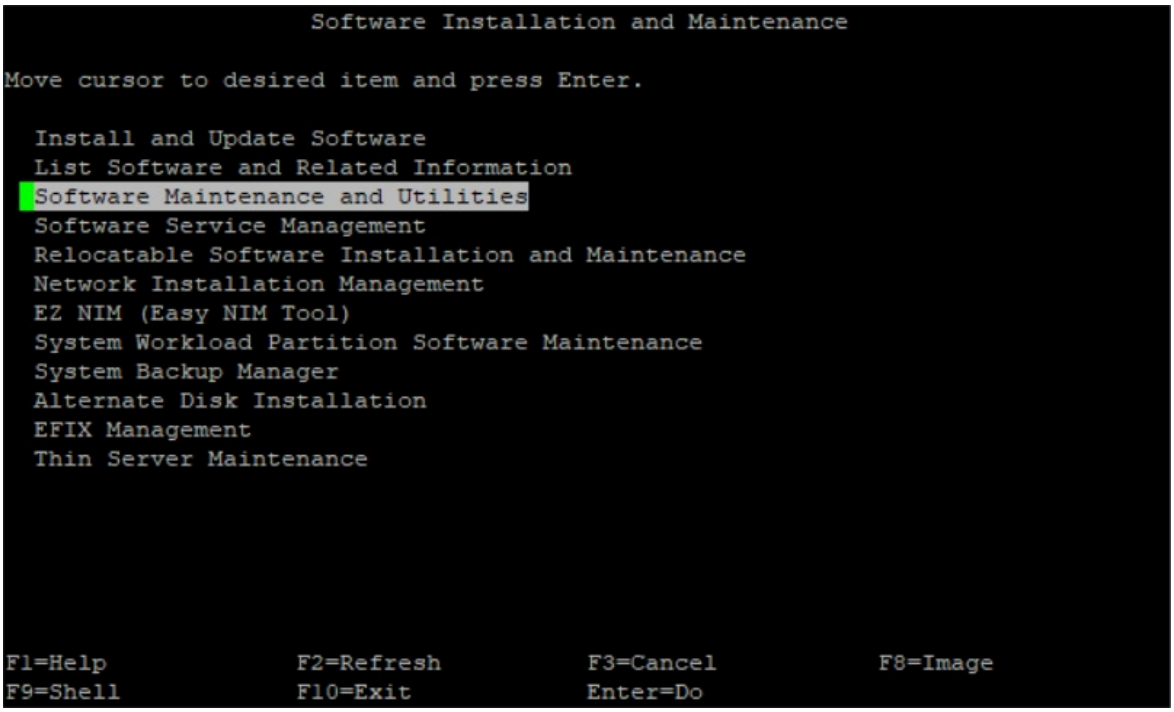
1. From your AIX terminal, run "smitty".

```
# smitty
```

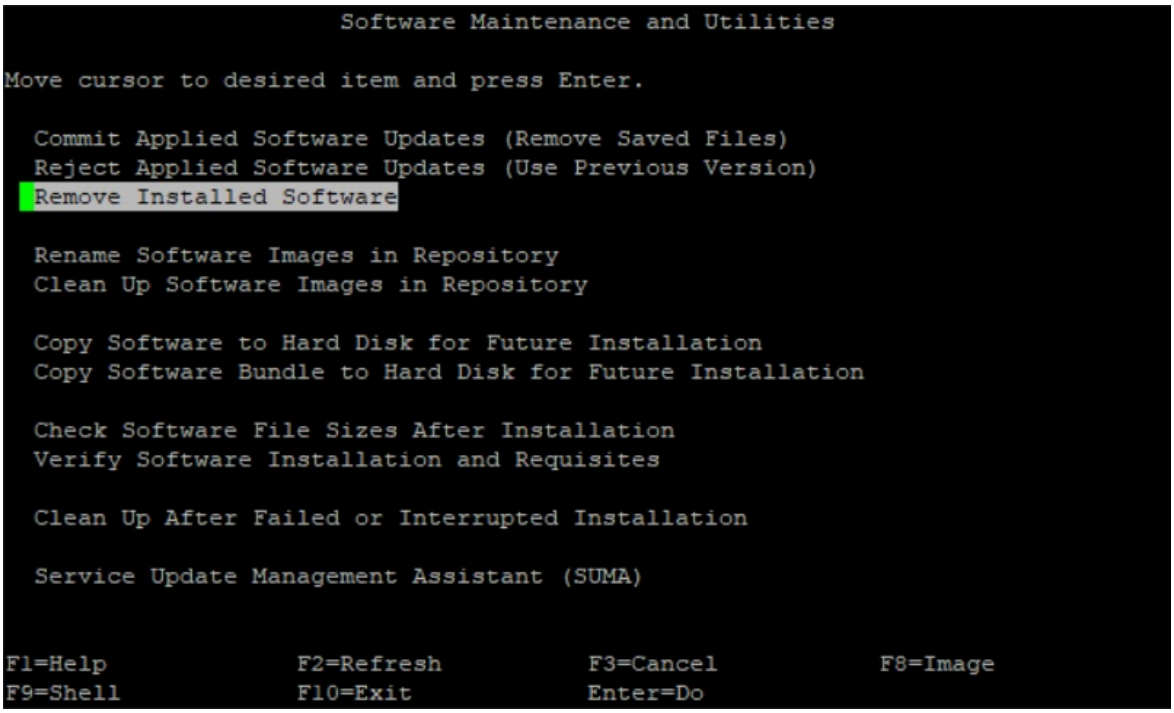
2. Select "Software Installation and Maintenance".



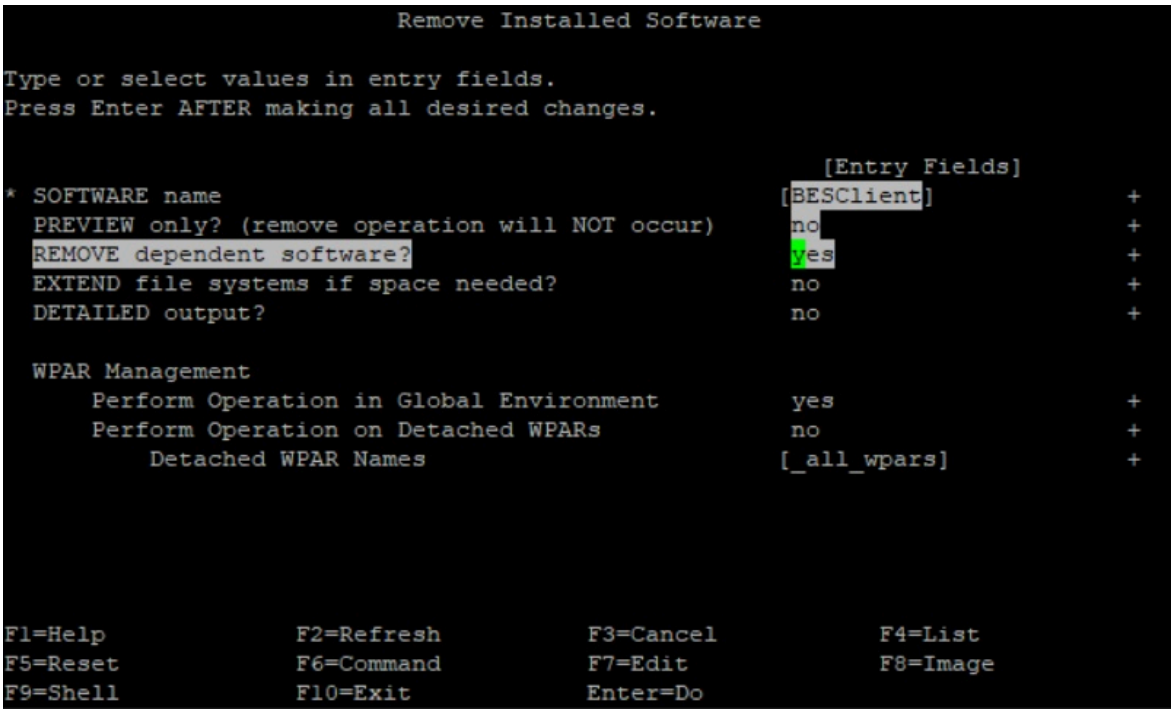
3. Select "Software Maintenance and Utilities".



4. Select "Remove Installed Software".



5. Set the "PREVIEW only" entry field to "no" and the "REMOVE dependent software" entry field to "yes".



6. Press Enter to remove the software.

Uninstalling the BigFix Client on Linux

To uninstall the BigFix client installed on a Linux operating system, follow the appropriate steps.

To manually uninstall the client on **Red Hat Enterprise Linux (RHEL)** and **SUSE Linux Enterprise Server (SLES)**:

1. Stop the BigFix client process

   ```
   systemctl stop besclient
   ```

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

   ```
   /etc/init.d/besclient stop
   ```

2. Run the following RPM command to find the installed package name

```
rpm -qa | grep -i BESAgent
```

3. Uninstall the installed RPM package returned in step 2

```
rpm -e BESAgent-XXX
```

4. Manually remove the following directories

```
rm -rf /etc/opt/BESClient
rm -rf /opt/BESClient
rm -rf /tmp/BES
rm -rf /var/opt/BESClient
rm -rf /var/opt/BESCommon
```

To manually uninstall the client on **Ubuntu Linux**, **Debian Linux** and **Raspbian**:

1. Stop the BigFix client process

```
/etc/init.d/besclient stop
```

2. Run the following command to find the installed package name

```
dpkg -l | grep -i BESAgent
```

3. Uninstall the installed package returned in step 2

```
dpkg --purge BESAgent-XXX
```

4. Manually remove the following directories

```
rm -rf /etc/opt/BESClient
rm -rf /opt/BESClient
rm -rf /tmp/BES
rm -rf /var/opt/BESClient
rm -rf /var/opt/BESCommon
```

Uninstalling the BigFix Client on Solaris

If you have a BigFix Relay installed on the same computer, first remove the BigFix Relay and then proceed with the BigFix client uninstallation.

If you installed the client using the SVR4 (.pkg file) format

To uninstall the BigFix client installed on a Solaris operating system, if you installed it using the legacy SVR4 (.pkg file) format, perform the following steps:

1. Stop the agent process before removing it

   ```
   /etc/init.d/besclient stop
   ```

2. Uninstall the BigFix Client by running the following command

   ```
   pkgrm BESagent
   ```

3. Manually remove the following directories

   ```
   rm -rf /etc/opt/BESClient
   rm -rf /var/opt/BESClient
   rm -rf /opt/BESClient
   rm -rf /var/opt/BESCommon
   ```

If you installed the client using the IPS (.p5p file) format

To uninstall the BigFix client installed on a Solaris 11 operating system, if you installed it using the IPS (.p5p file) format, perform the following steps:

1. Stop the agent process before removing it

   ```
   /etc/init.d/besclient stop
   ```

2. Uninstall the BigFix Client by running the following command

   ```
   pkg uninstall BESagent
   ```

Uninstalling the IPS package moves the files added at runtime under **$IMAGE_META/lost+found**. The default value for IMAGE_META is **/var/pkg**. The uninstall command shows a message reporting the actual paths, for example:

```
The following unexpected or editable files and directories were
salvaged while executing the requested package operation; they
have been moved to the displayed location in the image:
/var/opt/BESClient
 -> /var/pkg/lost+found/var/opt/BESClient-20190320T135633Z
```

3. Manually remove all the directories listed in the uninstall message, for example:

```
rm -rf /var/pkg/lost+found/var/opt/BESClient-20190320T135633Z
```

4. In addition, manually remove the `/etc/opt/BESClient` directory by running the following command

```
rm -rf /etc/opt/BESClient
rm -rf /var/opt/BESCommon
```

Uninstalling the BigFix Client on Mac

How to uninstall the BigFix client installed on a Mac OS operating system.

The Mac OS package provides an uninstallation script, so to remove the BigFix client use *sudo* to run the following script:

```
sudo /Library/BESAgent/BESAgent.app/Contents/MacOS/BESAgentUninstaller.sh
```

Appendix C. Glossary

This glossary provides terms and definitions for the BigFix software and products.

The following cross-references are used in this glossary:

- *See* refers you from a nonpreferred term to the preferred term or from an abbreviation to the spelled-out form.
- *See also* refers you to a related or contrasting term.

A *(on page 664)* B *(on page 665)* C *(on page 666)* D *(on page 668)* E *(on page 670)* F *(on page 670)* G *(on page 670)* L *(on page 671)* M *(on page 671)* N *(on page 672)* O *(on page 672)* P *(on page 673)* R *(on page 673)* S *(on page 674)* T *(on page 676)* U *(on page 676)* V *(on page 677)* W *(on page 677)*

A

action

1. See Fixlet *(on page 670)*.
2. A set of Action Script commands that perform an operation or administrative task, such as installing a patch or rebooting a device.

Action Script

Language used to perform an action on an endpoint.

agent

See BigFix agent *(on page 665)*.

ambiguous software

Software that has an executable file that looks like another executable file, or that exists in more than one place in a catalog (Microsoft Word as a standalone product or bundled with Microsoft Office).

audit patch

A patch used to detect conditions that cannot be remediated and require the attention of an administrator. Audit patches contain no actions and cannot be deployed.

automatic computer group

A computer group for which membership is determined at run time by comparing the properties of a given device against the criteria set for group membership. The set of devices in an automatic group is dynamic, meaning that the group can and does change. See also computer group *(on page 666)*.

B

baseline

A collection of actions that are deployed together. A baseline is typically used to simplify a deployment or to control the order in which a set of actions are applied. See also deployment group *(on page 668)*.

BigFix agent

The BigFix code on an endpoint that enables management and monitoring by BigFix.

BigFix client

See BigFix agent *(on page 665)*.

BigFix console

The primary BigFix administrative interface. The console provides a full set of capabilities to BigFix administrators.

BYOD

Bring Your Own Device (BYOD) refers to employees using personal devices to connect to their organizational networks and access work-related systems and potentially sensitive or confidential data.

C

client

A software program or computer that requests services from a server. See also server *(on page 675)*.

client time

The local time on a BigFix client device.

Cloud

A set of compute and storage instances or services that are running in containers or on virtual machines.

Common Vulnerabilities and Exposures Identification Number (CVE ID)

A number that identifies a specific entry in the National Vulnerability Database. A vendor's patch document often includes the CVE ID, when it is available. See also National Vulnerability Database *(on page 672)*.

Common Vulnerabilities and Exposures system (CVE)

A reference of officially known network vulnerabilities, which is part of the National Vulnerabilities Database (NVD), maintained by the US National Institute of Standards and Technology (NIST).

component

An individual action within a deployment that has more than one action. See also deployment group *(on page 668)*.

computer group

A group of related computers. An administrator can create computer groups to organize systems into meaningful categories, and to facilitate deployment of content to multiple computers. See also automatic computer group *(on page 665)* and manual computer group *(on page 671)*.

console

See BigFix console *(on page 665)*.

content

Digitally-signed files that contain data, rules, queries, criteria, and other instructions, packaged for deployment across a network. BigFix agents use the detection criteria (Relevance statements) and action instructions (Action Script statements) in content to detect vulnerabilities and enforce network policies.

content relevance

A determination of whether a patch or piece of software is eligible for deployment to one or more devices. See also device relevance *(on page 669)*.

Coordinated Universal Time (UTC)

The international standard of time that is kept by atomic clocks around the world.

corrupt patch

A patch that flags an operator when corrections made by an earlier patch have been changed or compromised. This situation can occur when an earlier service pack or application overwrites later files, which results in patched files that are not current. The corrupt patch flags the situation and can be used to re-apply the later patch.

custom content

BigFix code that is created by a customer for use on their own network, for example, a custom patch or baseline.

CVE

See Common Vulnerabilities and Exposures system *(on page 666)*.

CVE ID

See Common Vulnerabilities and Exposures Identification Number *(on page 666)*.

D

data stream

A string of information that serves as a source of package data.

default action

The action designated to run when a Fixlet is deployed. When no default action is defined, the operator is prompted to choose between several actions or to make an informed decision about a single action.

definitive package

A string of data that serves as the primary method for identifying the presence of software on a computer.

deploy

To dispatch content to one or more endpoints for execution to accomplish an operation or task, for example, to install software or update a patch.

deployment

Information about content that is dispatched to one or more endpoints, a specific instance of dispatched content.

deployment group

The collection of actions created when an operator selects more than one action for a deployment, or a baseline is deployed. See also baseline *(on page 665)*, component *(on page 666)*, deployment window *(on page 669)*, and multiple action group *(on page 672)*.

deployment state

The eligibility of a deployment to run on endpoints. The state includes parameters that the operator sets, such as 'Start at 1AM, end at 3AM.'

deployment status

Cumulative results of all targeted devices, expressed as a percentage of deployment success.

deployment type

An indication of whether a deployment involved one action or multiple actions.

deployment window

The period during which a deployment's actions are eligible to run. For example, if a Fixlet has a deployment window of 3 days and an eligible device that has been offline reports in to BigFix within the 3-day window, it gets the Fixlet. If the device comes back online after the 3-day window expires, it does not get the Fixlet. See also deployment group *(on page 668)*.

device

An endpoint, for example, a laptop, desktop, server, or virtual machine that BigFix manages; an endpoint running the BigFix Agent.

device holder

The person using a BigFix-managed computer.

device property

Information about a device collected by BigFix, including details about its hardware, operating system, network status, settings, and BigFix client. Custom properties can also be assigned to a device.

device relevance

A determination of whether a piece of BigFix content applies to applies to a device, for example, where a patch should be applied, software installed, or a baseline run. See also content relevance *(on page 667)*.

device result

The state of a deployment, including the result, on a particular endpoint.

Disaster Server Architecture (DSA)

An architecture that links multiple servers to provide full redundancy in case of failure.

DSA

See Disaster Server Architecture *(on page 669)*.

dynamically targeted

Pertaining to using a computer group to target a deployment.

E

endpoint

A networked device running the BigFix agent.

F

filter

To reduce a list of items to those that share specific attributes.

Fixlet

A piece of BigFix content that contains Relevance and Action Script statements bundled together to perform an operation or task. Fixlets are the basic building blocks of BigFix content. A Fixlet provides instructions to the BigFix agent to perform a network management or reporting action.

Full Disk Encryption

To reduce a list of items to those that share specific attributes.

G

group deployment

A type of deployment in which multiple actions were deployed to one or more devices.

H

Hybrid cloud

The utilization of distinct sets of cloud services (typically public and private) with integration and/or orchestration across them.

L

locked

An endpoint state that prevents most of the BigFix actions from running until the device is unlocked.

M

MAG

See multiple action group *(on page 672)*.

management rights

The limitation of console operators to a specified group of computers. Only a site administrator or a master operator can assign management rights.

manual computer group

A computer group for which membership is determined through selection by an operator. The set of devices in a manual group is static, meaning they do not change. See also computer group *(on page 666)*.

master operator

A console operator with administrative rights. A master operator can do everything that a site administrator can do, except creating operators.

masthead

A collection of files that contain the parameters of the BigFix process, including URLs to Fixlet content. The BigFix agent brings content into the enterprise based on subscribed mastheads.

MCM and BigFix Mobile

Refers to the offering by Bigfix that is common for both Modern Client Management to manage laptops (Windows and macOS) and BigFix Mobile to manage mobile devices (Android, iOS, and iPadOS).

mirror server

A BigFix server required if the enterprise does not allow direct web access but instead uses a proxy server that requires password-level authentication.

Multicloud

The utilization of distinct sets of cloud services, typically from multiple vendors, where specific applications are confined to a single cloud instance.

multiple action group (MAG)

A BigFix object that is created when multiple actions are deployed together, as in a baseline. A MAG contains multiple Fixlets or tasks. See also deployment group *(on page 668)*.

N

National Vulnerability Database (NVD)

A catalog of officially known information security vulnerabilities and exposures, which is maintained by the National Institute of Standards and Technology (NIST). See also Common Vulnerabilities and Exposures Identification Number *(on page 666)*.

NVD

See National Vulnerability Database *(on page 672)*.

O

offer

A deployment option that allows a device holder to accept or decline a BigFix action and to exercise some control over when it runs. For example, a device

holder can decide whether to install a software application, and whether to run the installation at night or during the day.

open-ended deployment

A deployment with no end or expiration date; one that runs continuously, checking whether the computers on a network comply.

operator

A person who uses the BigFix WebUI, or portions of the BigFix console.

P

patch

A piece of code added to vendor software to fix a problem, as an immediate solution that is provided to users between two releases.

patch category

A description of a patch's type and general area of operation, for example, a bug fix or a service pack.

patch severity

The level of risk imposed by a network threat or vulnerability and, by extension, the importance of applying its patch.

R

relay

A client that is running special server software. Relays spare the server and the network by minimizing direct server-client downloads and by compressing upstream data.

Relevance

BigFix query language that is used to determine the applicability of a piece of content to a specified endpoint. Relevance asks yes or no questions and evaluates the results. The result of a Relevance query determines whether

an action can or should be applied. Relevance is paired with Action Script in Fixlets.

S

SCAP

See Security Content Automation Protocol *(on page 674)*.

SCAP check

A specific configuration check within a Security Content Automation Protocol (SCAP) checklist. Checks are written in XCCDF and are required to include SCAP enumerations and mappings per the SCAP template.

SCAP checklist

A configuration checklist that is written in a machine-readable language (XCCDF). Security Content Automation Protocol (SCAP) checklists have been submitted to and accepted by the NIST National Checklist Program. They also conform to a SCAP template to ensure compatibility with SCAP products and services.

SCAP content

A repository that consists of security checklist data represented in automated XML formats, vulnerability and product name related enumerations, and mappings between the enumerations.

SCAP enumeration

A list of all known security related software flaws (CVEs), known software configuration issues (CCEs), and standard vendor and product names (CPEs).

SCAP mapping

The interrelationship of enumerations that provides standards-based impact measurements for software flaws and configuration issues.

Security Content Automation Protocol (SCAP)

A set of standards that is used to automate, measure, and manage vulnerability and compliance by the National Institute of Standards and Technology (NIST).

server

A software program or a computer that provides services to other software programs or other computers. See also client *(on page 666)*.

signing password

A password that is used by a console operator to sign an action for deployment.

single deployment

A type of deployment where a single action was deployed to one or more devices.

site

A collection of BigFix content. A site organizes similar content together.

site administrator

The person who is in charge of installing BigFix and authorizing and creating new console operators.

software package

A collection of Fixlets that install a software product on a device. Software packages are uploaded to BigFix by an operator for distribution. A BigFix software package includes the installation files, Fixlets to install the files, and information about the package (metadata).

SQL Server

A full-scale database engine from Microsoft that can be acquired and installed into the BigFix system to satisfy more than the basic reporting and data storage needs.

standard deployment

A deployment of BigFix that applies to workgroups and to enterprises with a single administrative domain. It is intended for a setting in which all Client computers have direct access to a single internal server.

statistically targeted

Pertaining to the method used to target a deployment to a device or piece of content. Statically targeted devices are selected manually by an operator.

superseded patch

A type of patch that notifies an operator when an earlier version of a patch has been replaced by a later version. This occurs when a later patch updates the same files as an earlier one. Superseded patches flag vulnerabilities that can be remediated by a later patch. A superseded patch cannot be deployed.

system power state

A definition of the overall power consumption of a system. BigFix Power Management tracks four main power states Active, Idle, Standby or Hibernation, and Power Off.

T

target

To match content with devices in a deployment, either by selecting the content for deployment, or selecting the devices to receive content.

targeting

The method used to specify the endpoints in a deployment.

task

A type of Fixlet designed for re-use, for example, to perform an ongoing maintenance task.

U

UTC

See Coordinated Universal Time *(on page 667)*.

V

virtual private network (VPN)

An extension of a company intranet over the existing framework of either a public or private network. A VPN ensures that the data that is sent between the two endpoints of its connection remains secure.

VPN

See virtual private network *(on page 677)*.

vulnerability

A security exposure in an operating system, system software, or application software component.

W

Wake-from-Standby

A mode that allows an application to turn a computer on from standby mode during predefined times, without the need for Wake on LAN.

Wake on LAN

A technology that enables a user to remotely turn on systems for off-hours maintenance. A result of the Intel-IBM Advanced Manageability Alliance and part of the Wired for Management Baseline Specification, users of this technology can remotely turn on a server and control it across the network, thus saving time on automated software installations, upgrades, disk backups, and virus scans.

WAN

See wide area network *(on page 677)*.

wide area network (WAN)

A network that provides communication services among devices in a geographic area larger than that served by a local area network (LAN) or a metropolitan area network (MAN).

Appendix D. 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

Appendix E. Accessibility features for BigFix

Accessibility features assist users who have a disability, such as restricted mobility or limited vision, to use information technology content successfully.

Accessibility features

BigFix includes the following major accessibility features:

- Keyboard-only operation
- Operations that use a screen reader

BigFix uses the latest W3C Standard, WAI-ARIA 1.0 (http://www.w3.org/TR/wai-aria/), to ensure compliance to US Section 508 (http://www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-section-508-standards/section-508-standards), and Web Content Accessibility Guidelines (WCAG) 2.0 (http://www.w3.org/TR/WCAG20/). To take advantage of accessibility features, use the latest release of your screen reader in combination with the latest web browser that is supported by this product.

The BigFix online product documentation is enabled for accessibility.

Keyboard navigation

This product uses standard navigation keys.

BigFix uses the following keyboard shortcuts.

Table 13. Keyboard shortcuts in BigFix

Action	Shortcut for Internet Explorer	Shortcut for Firefox
Move to the Contents View frame	Alt+C, then press Enter and Shift+F6	Shift+Alt+C and Shift +F6

Interface information

The BigFix user interfaces do not have content that flashes 2 - 55 times per second.

BigFix Web UI

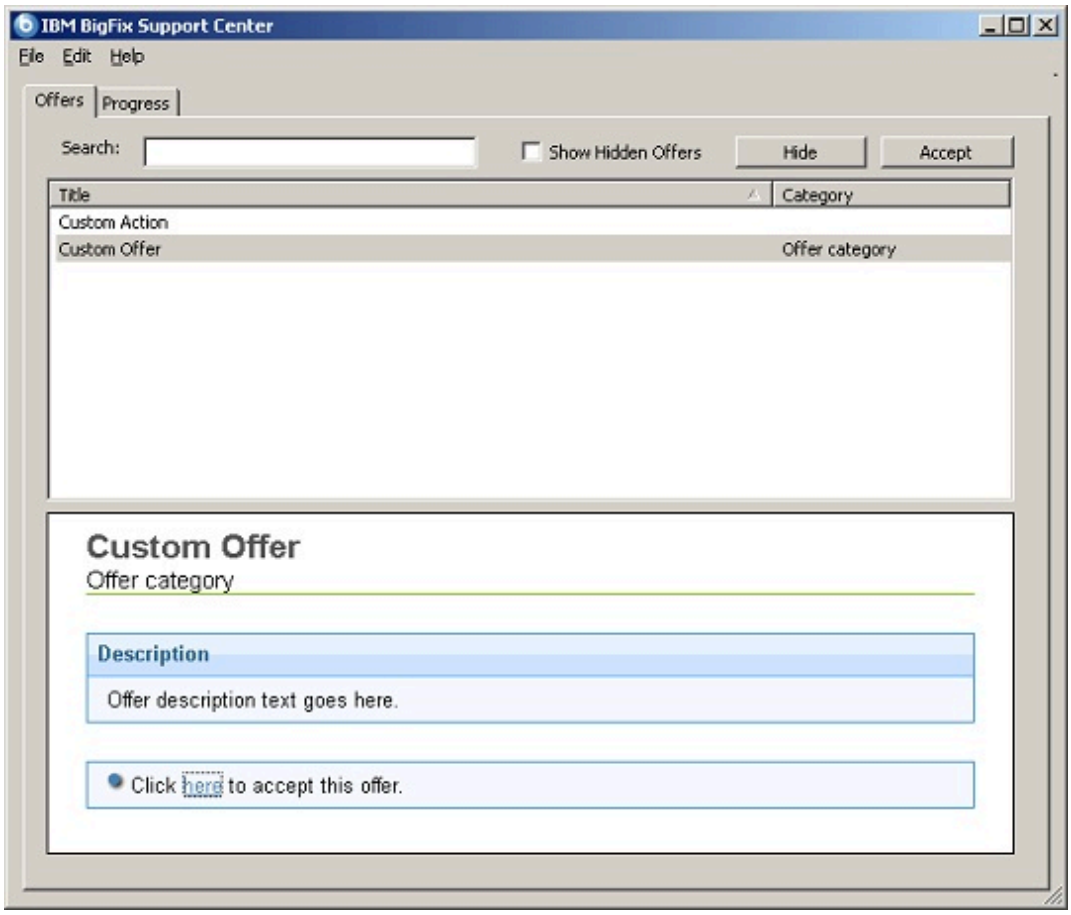
The BigFix web user interfaces rely on cascading style sheets to render content properly and to provide a usable experience. The application provides an equivalent way for low-vision users to use a user's system display settings, including high-contrast mode. You can control font size by using the device or web browser settings.

The BigFix web user interface includes WAI-ARIA navigational landmarks that you can use to quickly navigate to functional areas in the application.

BigFix Client Dashboard

The BigFix Client Dashboard shows message boxes to the end-users logged in to the client computer. It pops up on the managed end user systems when an action or an offer is triggered. The messages displayed include pre-action messages, action running messages, and shutdown and restart messages. In this dashboard use:

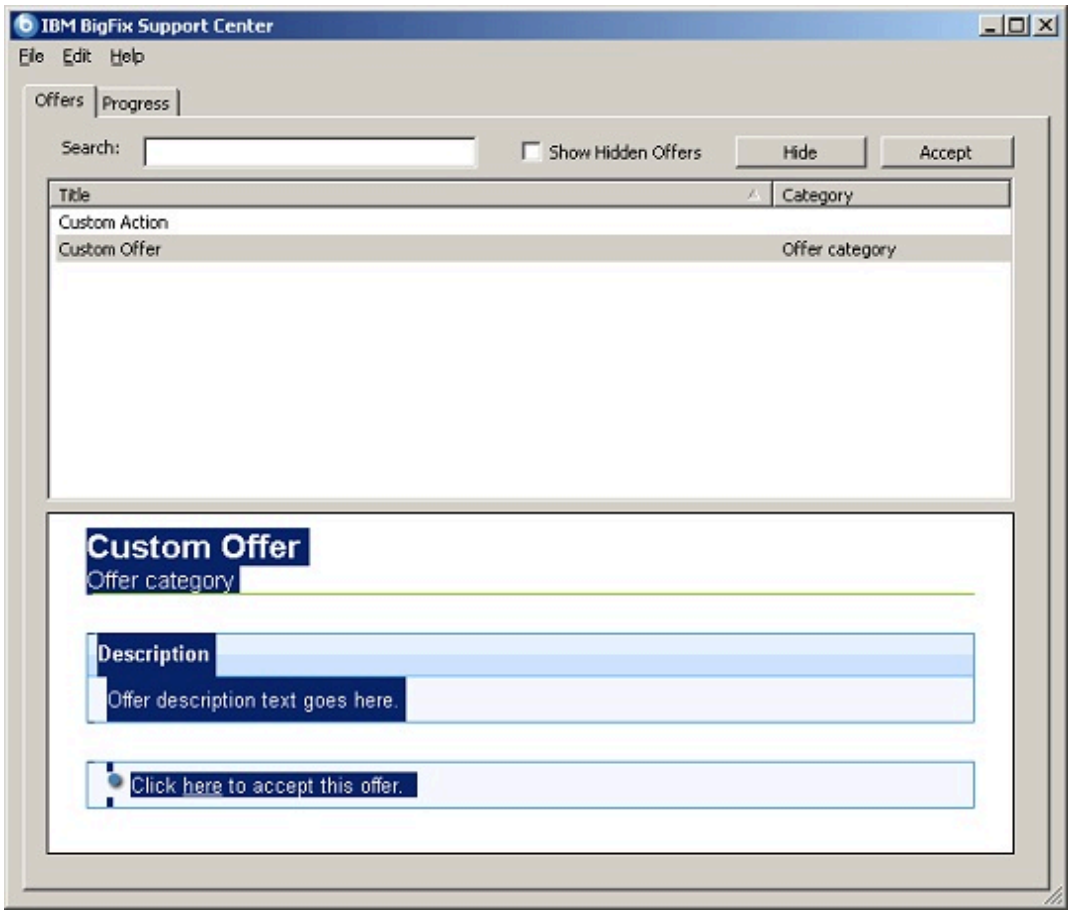
- The Tab key to navigate to the offer list.
- Up and Down arrows to move selection.
- Space to select an offer.
- Ctrl+A to select all text in the HTML description pane.
- Ctrl+C to copy the selected text.



Within the selected offer use:

- Tab key to navigate to the HTML description pane.
- Once position on the "Click here to accept this offer" button, Space to accept the offer.

Use Ctrl+A to select all text in the HTML description pane and then Ctrl+C to copy it.



Vendor software

BigFix includes certain vendor software that is not covered under the HCL license agreement. HCL makes no representation about the accessibility features of these products. Contact the vendor for the accessibility information about its products.

Related accessibility information

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.