BigFix Platform
Configuration Guide
Special notice

Before using this information and the product it supports, read the information in Notices (on page 162).
Edition notice

This edition applies to version 9.5 of BigFix and to all subsequent releases and modifications until otherwise indicated in new editions.
Contents

Chapter 1. Introduction......................................................................................................................... 1
  What is new in V9.5.......................................................................................................................... 1
  Service Management Connect.......................................................................................................... 20
  Terms used in this guide................................................................................................................... 21

Chapter 2. BigFix Site Administrator and Console Operators..................................................... 22
  The Site Administrator...................................................................................................................... 22
  The Console Operators..................................................................................................................... 24
  Different ways to define a Console Operator................................................................................... 24
  Adding Local Operators.................................................................................................................. 25
  Mapping authorized activities with permissions............................................................................. 29
  Operators and analysis.................................................................................................................... 31
  Monitoring Operators..................................................................................................................... 31

Chapter 3. Integrating with LDAP..................................................................................................... 34
  Integrating with a Generic LDAP.................................................................................................... 34
  Integrating with Active Directory.................................................................................................. 35
  Integrating the Windows server with Active Directory................................................................. 36
  Integrating the Linux server with Active Directory........................................................................ 40
  Adding LDAP Operators............................................................................................................... 48
  Associating an LDAP group............................................................................................................ 51

Chapter 4. Enabling SAML V2.0 authentication for LDAP operators............................................ 53
  What Is SAML 2.0.......................................................................................................................... 53
  How SAML works.......................................................................................................................... 54
  Which BigFix user interfaces integrate with SAML V2.0.............................................................. 55
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>How BigFix integrates with SAML V2.0</td>
<td>55</td>
</tr>
<tr>
<td>Assumptions and requirements</td>
<td>56</td>
</tr>
<tr>
<td>What changes from the BigFix user's perspective</td>
<td>57</td>
</tr>
<tr>
<td>How to configure BigFix to integrate with SAML 2.0</td>
<td>59</td>
</tr>
<tr>
<td><strong>Chapter 5. Using multiple servers (DSA)</strong></td>
<td>63</td>
</tr>
<tr>
<td>Disaster Server Architecture (DSA)</td>
<td>63</td>
</tr>
<tr>
<td>Configuring relay failover</td>
<td>65</td>
</tr>
<tr>
<td>Message Level Encryption and DSA</td>
<td>66</td>
</tr>
<tr>
<td>Managing Replication (DSA) on Windows systems</td>
<td>67</td>
</tr>
<tr>
<td>Changing the replication interval on Windows systems</td>
<td>67</td>
</tr>
<tr>
<td>Switching the master server on Windows systems</td>
<td>68</td>
</tr>
<tr>
<td>Managing Replication (DSA) on Linux systems</td>
<td>68</td>
</tr>
<tr>
<td>Changing the replication interval on Linux systems</td>
<td>68</td>
</tr>
<tr>
<td>Switching the master server on Linux systems</td>
<td>70</td>
</tr>
<tr>
<td>Schema next tables regenerated during upgrade</td>
<td>71</td>
</tr>
<tr>
<td><strong>Chapter 6. Server object IDs</strong></td>
<td>73</td>
</tr>
<tr>
<td><strong>Chapter 7. Customizing HTTPS for Gathering</strong></td>
<td>74</td>
</tr>
<tr>
<td><strong>Chapter 8. Configuring secure communication</strong></td>
<td>77</td>
</tr>
<tr>
<td>Customizing HTTPS on Web Reports</td>
<td>78</td>
</tr>
<tr>
<td>Customizing HTTPS manually on Windows systems</td>
<td>81</td>
</tr>
<tr>
<td>Customizing HTTPS manually on Linux systems</td>
<td>82</td>
</tr>
<tr>
<td>Customizing HTTPS on REST API</td>
<td>83</td>
</tr>
<tr>
<td>Customizing HTTPS manually on Windows systems</td>
<td>84</td>
</tr>
<tr>
<td>Customizing HTTPS manually on Linux systems</td>
<td>85</td>
</tr>
<tr>
<td>Private key and certificate format</td>
<td>86</td>
</tr>
</tbody>
</table>
Chapter 9. Downloading files in air-gapped environments…………………………………….. 94
  Overview......................................................................................................................... 94
    Non-extraction usage overview.................................................................................. 94
  Extraction usage overview.......................................................................................... 98
  Requirements.................................................................................................................. 100
  Using the Airgap tool.................................................................................................... 102
    Non-extraction usage................................................................................................. 102
  Extraction usage........................................................................................................... 118
  Log files......................................................................................................................... 126

Chapter 10. Getting client information by using BigFix Query.............................................. 128
  BigFix Query requirements........................................................................................... 128
  BigFix Query restrictions.............................................................................................. 129
  Who can use BigFix Query........................................................................................... 129
  How to run BigFix Query from the WebUI..................................................................... 131
  How BigFix manages BigFix Query requests............................................................. 131

Chapter 11. Persistent connections...................................................................................... 137

Chapter 12. Relays in DMZ................................................................................................. 140

Chapter 13. Peer to peer mode........................................................................................... 144

Chapter 14. Archiving Client files on the BigFix Server....................................................... 147
  Archive manager settings.............................................................................................. 148
  Creating a Custom Action............................................................................................. 148
  Archive Manager........................................................................................................... 148
    Archive Manager internal variables.......................................................................... 148
  Archive Manager Index File Format............................................................................. 149
  Upload Manager............................................................................................................ 150
Virtualized environments and virtual machines

Chapter 15. Maintenance and Troubleshooting

Monitoring relays health

Relay and Server diagnostics

Virtualized environments and virtual machines

Chapter 16. Support

Notices

Index
Chapter 1. Introduction

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

What is new in V9.5

BigFix Platform Version 9.5 provides new features and enhancements.

**Patch 14:**

**Security vulnerabilities and library upgrades**

- The libssh2 external library level was upgraded to Version 1.9.0.
- The OpenLDAP external library level was upgraded to Version 2.4.48.

**Added support for BigFix Agent**

Added support for BigFix Agent running on:

- SUSE Linux Enterprise 15 PPC 64-bit.
- Red Hat Enterprise Linux 8 x86 64-bit.
- MacOS 10.15.

**Patch 13:**

**Relays in DMZ**

You can configure parent relays outside a demilitarized zone (DMZ) to initiate connections to child relays that are within the DMZ network. This means that relay-to-relay communication is always initiated from the parent relay. You can use this feature to avoid opening firewall ports from the DMZ to the internal
secure network which in turns helps toughen the security of your environment.

For details, see Relays in DMZ (on page 140).

Troubleshoot issues more efficiently by persisting the relay chain on the BigFix Client

The Relay chain is identified for each client and it consists of a set of Relays involved in the registration between the client and the server to which the client is registered. With this feature, you can allow the client to trace the relay chain for each registration and ensure that the relay information is available on the client side. This helps you troubleshoot issues related to client-to-server communications more efficiently, and improve the data reported by the BES Client Diagnostics task.

For details, see Viewing the relay chain on the client (on page ).

Install BigFix agent with IPS format (.p5p package) on Solaris 11

On Solaris 11, the BigFix agent installation package is now available as IPS (Image Packaging System), which is the latest Solaris packaging technology. The old version of the installation package is also still available. You can therefore choose an installation option that best suits your requirements.

For details, see Solaris 11 installation Instructions (on page ).

Delete registry keys by using actionscript

You can now delete not just the values of the registry keys set on the clients, but the keys themselves as a whole by using actionscripts. This operation also has a 64-bit equivalent. This feature helps you maintain the Windows registry keys, for example by removing the keys that are no longer used.
For details, see `regkeydelete` and `regkeydelete64`.

**Removal of Adobe Flash Player dependency in Web Reports component**

As a preparatory step to deal with end of support (EOS) of Adobe Flash Player in the year 2020, the Adobe Flash Player dependency was removed from the Web Reports functionality. However, your experience of viewing the graphs remains the same.

**Run queries in client context**

BigFix extends the ability of the Agent to run queries when submitted through the Fixlet Debugger or REST API. This allows you to run any relevance for tasks such as troubleshooting or investigations directly from these interfaces.

For details, see [BigFix Query](#).

**Added support for BigFix Agent on Raspberry Pi**

Added support for running Agent on Raspbian 9 Raspberry Pi 3 models B and B+.

For details, see [Raspbian (64-bit) Installation Instructions](#).

**Added support for BigFix Agent SLES 15 on Intel**

Added support for BigFix Agent running on SUSE Linux Enterprise 15 x86_64 on Intel.

**Security vulnerabilities and library upgrades**

- The OpenSSL toolkit level was upgraded to Version 1.0.2r.
- The libcURL file transfer library level was upgraded to Version 7.64.0.

**Patch 12:**

**Security vulnerabilities and library upgrades**
In this version, security vulnerabilities were addressed and some libraries were upgraded.

- The OpenSSL toolkit level was upgraded to Version 1.0.2q.
- The jQuery library level was upgraded to Version 3.0.0.
- The jQuery UI library level was upgraded to Version 1.12.1.
- The jqPlot (jQuery plugin) level was upgraded to Version 1.0.9.

**Patch 11:**

**Reduce network traffic and relay infrastructure costs by exchanging cached files with peers (PeerNest)**

This version introduces peer-to-peer configuration which will help you reduce the relay infrastructural costs. In a peer-to-peer setup, endpoints in a subnet coordinate their download activities in order to download binaries only once from the relay, thus reducing the network traffic outside of the subnet. With this setup, you can facilitate a faster and direct exchange of binaries between endpoints and remove the need for every client to download the same binary from a relay, allowing the removal of dedicated relays from branch offices.

For details, see [Peer to peer mode (on page 144)](#).

**Improve real-time visibility by delivering notifications to clients across firewalls through client-established, persistent connections**

The BigFix Query function relies on a UDP based notification where the relay notifies the clients of a new query. Firewalls or NAT may block this notification mechanism. Through the new persistent connection feature, a persistent connection initiated by the client is used by the relay to manage the UDP based notification. This allows the delivery of any type of notification, thus offering a faster alternative to command
polling. A persistent connected client also acts as a UDP notification forwarder (proxy) for the other clients in the same subnet which can reduce the number of connections and optimize relay performance. The relay can deliver notifications to clients through client-established, persistent connections.

For details, see Persistent connections (on page 137).

**Prevent BES server overload and network congestion by defining a fallback relay**

You can now define a fallback relay for your clients when they fail to connect to any relay specified in their settings.

For details, see Step 2 - Requesting a license certificate and creating the masthead (on page ) and Editing the Masthead on Linux systems (on page ).

**Simplify the installation and upgrade of the WebUI component including it as part of the BigFix Platform installation**

The installation of the BigFix Platform (both evaluation and production versions) on both Windows and Linux now includes the option to install the WebUI component as well, offering a convenient alternative to the fixlet-based installation. The upgrade of the WebUI component will be executed as part of the platform components update process, and as noted in 9.5.10, the WebUI can now scale to manage 120,000 endpoints from either a Linux or Windows BES Server installation.

For details, see Installing the WebUI (on page ) (Windows) and (Optional) - Installing the WebUI Standalone (on page ) (Linux).
Enhance corporate security by specifying the TLS ciphers that can be used in network communications between the BigFix components and the internet

Starting in this version, 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.

For details, see Working with TLS cipher lists (on page 6).

Enhance security and reduce load on the BES root server by automatically shutting down the BigFix Console after a period of inactivity

Starting in this version, you can control the maximum amount of time to keep an inactive session of BigFix console alive. After the timeout, the BigFix console is closed.

For details, see List of advanced options (on page 6).

Enhance the security of your BigFix Server by optionally disabling access to the Internet

Starting in this version, you can control whether your server accesses the Internet for updating the license and gathering the sites or not by using a configuration setting.

For details, see Airgap Mode (on page 6).

Gather WebUI content more securely through HTTPS and in an optimized manner

- WebUI: Gather BES sites with HTTPS by default

You can gather license updates and external sites by using the HTTPS protocol on a BigFix server or in an airgapped
environment. For details, see Customizing HTTPS for Gathering (on page 74).

- Optimize Gathering from Synch Servers

  The Gathering process has been optimized with more effective handling of Gather errors.

**Establish an increased level of security when creating new users by assigning them minimal permissions**

  When you create users, they are assigned minimum permissions (read-only) by default, which offers an additional level of security.

  For details, see List of advanced options (on page 74) (look up defaultOperatorRolePermissions) and Adding Local Operators (on page 25).

**Enhanced security and visibility with more detailed server audit logs**

  The server audit logs now include the following items:

  - Messages for deletion of computers from the console or through API
  - Messages for deletion of actions
  - Audit entries are presented in a single line and contain the same number of field delimiters. Field delimiters are present even if no value exists for a specific field. Since the format of the audit fields is subject to change over time, each line has a version number as the first entry. The current format includes texts from existing audit log messages (which are in old format) and presents them in the last field.

  The server generates audit logs for two new events: the deletion of an action and the removal of a computer.

  For details, see Server audit logs (on page 74).
Reduce the costs of managing relay infrastructure through a new Dashboard that summarizes relay health across the entire network

You can now monitor the status of your relays across the entire network by using the Relay Health dashboard. The Relay Health Dashboard shows you specific details about the relays in your BigFix environment.

For details, see Relay Health Dashboard (on page  ).

Configure the default behavior of Timeout Override on clients

Starting in this version, you can define the default behavior for timeout and disposition on a specific client for all the programs or processes triggered by any wait or waithidden commands, unless it is specified differently in an override section of that specific wait or waithidden command definition.

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

Optimize and accelerate Platform REST API interactions

You can now control and reduce the number of fields returned by a REST request by using the ?fields= parameter to limit the fields returned for a given resource when using the API resources /api/actions and /api/action/{action id}/status.

For details, see Action and Computer.

Accelerate fixlet creation and testing by using the FastQuery interface in Fixlet Debugger

Fixlet Debugger is extended to use FastQuery interface in addition to Local Fixlet Debugger Evaluator and Local Client Evaluator. You can choose a remote endpoint to evaluate relevance.
Save time when working in tight maintenance windows by enabling group actions to start before sub action downloads are available

Group actions with pre-cached downloads now start without requiring all sub-action downloads to be available on the client, provided the downloads for the first relevant sub-action are available. Additionally, the server and relay caches are primed by continuing with as many download requests as possible even under a 'disk limited' constraint.

For details, see Enabling data pre-cache (on page ).

Other Enhancements

- Improved documentation on configuration settings. For details, see BigFix Configuration Settings (on page ).
- Added changes to the client component for enabling a new version of the self-service application (SSA).

Patch 10:

CDT Key file option and custom installation path

When installing the BigFix clients from the Client Deploy Tool (CDT) Wizard, you can access the target computers through the SSH key authentication. You can also specify for the Windows target computers a custom installation path, if you do not want to use the default installation path.

For more information, see Deploying clients from the console (on page ).

TLS-encrypted SMTP connection for Web Reports
When setting up an email address from Web Reports, you can upgrade the SMTP connection to TLS.

For more information, see Setting Up Email (on page ).

**Windows authentication leveraged in command line utilities**

You can use your Windows credentials to authenticate to BigFix utilities such as the PropagateFiles.exe tool and the IEM CLI.

For more information, see Creating special custom sites whose name begins with FileOnlyCustomSite (on page ).

**Windows performance, efficiency, and maintenance improvements**

- The FillDB configuration was modified to permit more efficient database bulk insert and update operations. Given that FillDB is responsible for pushing client reports into the database, this results in a more responsive and more efficient BigFix.

- The Microsoft SQL Server configuration was updated to provide improved concurrency and scalability options for BigFix.

- The BigFix provided Microsoft SQL Server index management scripts were rewritten to ensure indexes are better managed, with improved fault tolerance while consuming fewer system resources and reducing application impact. This has a positive impact on the long term performance, scalability, and stability of BigFix.

**Added support for BigFix Agent SLES 11 and 12 on Power 9**

Added support for the following BigFix Agents:

- SUSE Linux Enterprise 11 PPC on Power 9 (P8 compatibility mode)
- SUSE Linux Enterprise 12 PPC on Power 9 (P9 mode)
Added support for BigFix Agent on Mac OS 10.14

Added support for BigFix Agent on MacOS 10.14.

Note: 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 custom content. For more information, see Client requirements (on page ).

64-bit enablement for the Mac OS agent

The Mac OS agent binaries are now 64-bit applications.

Changes in the disaster recovery, hardware migration and roll back procedures

The changes introduced by some of the security enhancements have an impact on the disaster recovery, hardware migration and roll back procedures. For more details about these procedures, see:

Server Backup (on page )

Server Recovery (on page )

Removing the Product Components on Linux systems (on page )

BigFix Server Migration on Linux

Changed signing key for the Red Hat installation packages

Starting from BigFix Version 9.5.10, the Red Hat RPM packages for Server, Agent and Relay are signed with a new PGP key, different than the one used in Version 9.5.9. Also the CentOS BigFix Agent and Relay use the same Red Hat binaries. The same applies to Oracle Linux BigFix Agent.

For more information, see Red Hat Installation Instructions (on page ).

Patch 9:
Added signature to the Red Hat installation packages

Starting from BigFix Version 9.5.9, the Red Hat RPM packages for Server, Agent and Relay are signed with a PGP key. Also the CentOS BigFix Agent and Relay use the same Red Hat binaries. The same applies to the Oracle Linux BigFix Agent.

For more information, see Red Hat Installation Instructions (on page ).

Ability for endpoints to constrain the download action if the Agent is not connected to the designated (preferred) Relay

BigFix 9.5.9 introduces the capability to prevent starting actions requiring downloads when the BigFix Agent is not connected to a preferred Relay. In such scenario, you can avoid that actions are executed if the total size of the downloads associated to the action exceeds a configurable value.

For more information, see Download (on page ).

Ability for Web Reports to restrict access to some properties

BigFix 9.5.9 introduces a new client setting that allows to configure a list of properties that will be blacklisted for Web Reports. In such scenario, you can prevent reporting on large or privacy sensitive data and you can limit the memory usage.

For more information, see the _WebReports_Properties_Blacklist setting in Web Reports (on page ).

Improved Relay scalability by supporting 5000 endpoints per Relay

BigFix leaf relays for the Windows and Linux platforms can be configured now to manage up to 5000 endpoints.

For the implementation guidelines, see the BigFix capacity planning guide: Capacity Planning, Performance, and Management Guide.
Added support for AIX 7.2 on Power 9

Added support for BigFix Agent and Relay on AIX 7.2 on Power 9.

Patch 7:

New database offered during the installation

When performing a fresh installation of BigFix Server Version 9.5 Patch 7, if no database engine is detected, you can choose whether to install Microsoft SQL Server 2016 SP1 Evaluation or to manually install another SQL Server version. The provided evaluation version is valid for 180 days.

Slimmed down Windows installation files

When performing a fresh installation or an upgrade to Patch 7, the SQL Server installer is provided as a separate file and is no longer contained in the BigFix server installer which is now smaller.

Client Deploy Tool enhancements

- Added a new wizard to distribute the agents on all supported platforms
- Added a new dashboard to view the results of the deployments
- Added the possibility to upload the target log files to the BigFix server.

Names of files and folders using local encoding on UNIX and Linux clients

You can specify the names of files and folders of UNIX and Linux clients in their local encoding, even if it is different from the encoding on the BigFix server. 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://www.bigfix.com).

**Read from and write to files, having different encoding**

You can read from and write to files, having different encodings using the encoding inspector. For additional information see Reading and writing files in the specific encodings (on page ) and [BigFix Developer site](https://www.bigfix.com).

**Enhanced Client identity matching when Clients are detected**

You can use the new setting (`clientIdentityMatch`) to allow the BigFix Server to use the existing computer information to try to match the identity of a Client and reassign the same `ComputerID` to computers that might have been rolled back or restored and avoid having duplicate computer entries.

**New options when running commands as a user local to the target**

The `override` action script command has been improved with new options to run commands on the target client as user different from the logged on user. For more information, see the `override command` on the BigFix Developer web site.

**Improved SSL configuration documentation**

The documentation of SSL configuration has been updated to ensure a major consistency across the different BigFix applications. See the overview of the SSL configuration containing certificate requirements and links to the SSL configuration procedures for all BigFix applications: HTTPS across BigFix applications (on page ).

**Patch 6:**

**Security enforcement enhancements**
• Two new masthead parameters, minimumSupportedClient and minimumSupportedRelay are added to enforce a higher level of security in the deployment. For more information, see Additional administration commands (on page ) for Windows servers, or Running the BigFix Administration Tool (on page ) for Linux servers.

• You can use a new advanced option (requireSignedRegistration ) to ensure that a client registration request is not accepted if there is at least one relay in the registration chain that is not upgraded to the same version of BigFix that is installed on the Server.

**New security check on Fixlet/task content**

A new security check was added to parse the content of the imported or generated Fixlet and tasks, and identify the existence of possible script content. If such content is detected, a Warning Panel is displayed to the Console Operator.

**OpenSSL Initialization changes**

Starting from 9.5.6, each BigFix component initializes OpenSSL in FIPS Mode based on the existence of the client setting _BESClient_Cryptography_FipsMode, and the client masthead.

**Default status of Relay Diagnostic page changed**

On both the Server and the Relay components, the Relay Diagnostic page is now disabled by default. The Relay Diagnostic page can be enabled again by setting _BESRelay_Diagnostics_Enable = 1 on those components.

**Additional changes**

• Resigning of Mac Clients with new certificates
• Console Qualification for Windows 10 Creators Update
Patch 5:

Enablement for the BigFix Detect application

Client Deploy Tool enhancements

• Enabled the agents distribution on all supported platforms by using a new Fixlet
• Enabled the distribution of the old agent versions, including agent versions that are no longer supported in BigFix Version 9.5

Added capability to run Fixlet actions as a specific user and to specify the context for the actions

Specified under which specific user context a specific action must be run on the endpoint

Airgap tool enhancements

• Added capability to gather information on external sites without accessing a BigFix server in a secure deployment
• Added file download capability

Enhanced the FillDB component to process agent reports by using a multi-thread approach

Improved BigFix Platform performance by leveraging multi-core server resources

Added capability for a Non-Master Operator to stop other Non-Master Operator actions

Enhanced the BigFix evaluation installation to avoid ripping and replacing the BigFix deployment if transition to production license is needed
Improved the user experience for "Try and Buy" scenarios and promoted the evaluation environment to production environment without installing again

**Enhanced the REST API for Baseline support**

Enabled REST API to perform major baseline functionality available on the console

**Enhanced the BigFix agent application usage summary inspector**

Collected the process executable path

**Enhanced the Mac OS version of BigFix agent and inspectors**

- Detected applications installed into the /Library path
- Improved Wi-Fi inspectors
- Leveraged spotlight search when using inspectors for searching Mac installed applications
- Enabled the process inspectors to report the process path name

**Improved the BigFix database layer to enable direct access from Web UI**

- Enabled the Web UI not to depend on ETL and ensured backward compatibility with current Web UI versions still leveraging ETL
- Improved the Web UI scalability and performance

**Enhanced the Client UI end-user experience**

- Made running message dialog optionally not dismissible
- Made running message dialog optionally topmost

**Enhanced the Self Service application enablement**
• Allowed REST API blocking "action-ui-metadata" mime field included in the baseline and MAG definition
• Added timestamp information of when the offer was issued in the Offer Available message

Security enhancements

• Changed non-FIPS OpenSSL Windows library to use ASLR
• Created native Red Hat Enterprise Linux (RHEL) Version 6 based agent and relay to allow the client installation when the operating system is in FIPS mode

Patch 3:

Enablement for Remote Web UI deployment

You can deploy the Web UI on a remote endpoint rather than on the BigFix Server.

Enablement for BigFix Query enhancements

You can target BigFix Query requests to dynamic groups.

Enablement for BigFix Software Distribution enhancements

You can use the Self-Service catalog from the Client UI when using the SWD application.

Enablement for DB2 HADR

You can run the database backup without requiring the shutdown of the BigFix Server.

Enablement for BigFix Patch enhancements

A new inspector is added to the set of Client inspectors to allow the Patch application to discover broken filesets on AIX agents.

Added support for new platforms and database levels

• Microsoft SQL 2016 support
• Tiny core Linux support for relay.
• BigFix agent now supported on:
  ◦ SUSE Linux Enterprise 12 on Power 8 Little Endian
  ◦ Ubuntu 16.04 on Power 8 Little Endian
  ◦ Windows Server 2016 and System Center 2016
  ◦ Windows 10 Anniversary Update
  ◦ Mac OS 10.12 (Sierra)

**Migrated BigFix Platform manuals to the new BigFix Developer site**

The content of the following manuals was reworked, improved, and migrated to the BigFix Developer website, the new repository for the BigFix Platform development and customization documentation:

• Relevance Guide
• Action Guide
• API Reference Guide

Earlier versions of these manuals in PDF format are still available for download at https://help.hcltechsw.com/bigfix/9.5/platform/welcome/BigFix_Platform_welcome.html.

**Additional enhancements**

• SHA-2 signing certificate for Windows binaries
• Capability to install and run the Web Reports as a non-administrative user.

**Patch 2:**

**BigFix Query**

You can use this function to retrieve information and run relevance queries on client workstations from the WebUI BigFix Query Application or by using REST APIs. This function is
available only for BigFix Lifecycle or BigFix Compliance Version 9.5 Patch 2 or later licenses. For more information, see Getting client information by using BigFix Query (on page 128).

Version 9.5

Unicode support

BigFix Platform V9.5 gathers data from BigFix clients deployed with different code pages and languages, encodes the data into UTF-8 format, and reports it back to the BigFix server.

HTTPS gathering

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

SAML V2.0 integration

Single-sign-on and CAC/PIV authentication support for BigFix LDAP operators connecting to the console.

Database cleanup tools

You can use the BESAdmin interface or the BESAdmin command line to remove data about computers, custom Fixlets, properties, analyses, and actions and to update the PropertyIDMap table with changes.

FillDB log rotation

It is active by default with LogFileSizeLimit set to 100 MB.

For more information about the changes and the enhancements introduced with V9.5, see the Release Notes.

Service Management Connect

Connect, learn, and share with Service Management professionals: product support technical experts who provide their perspectives and expertise.
Access Service Management Connect at Endpoint Management. Use Service Management Connect to:

- Become involved with transparent development, an ongoing, open engagement between other users and HCL developers. You can access early designs, sprint demonstrations, product roadmaps, and prerelease code.
- Connect one-on-one with the experts to collaborate.
- Read blogs to benefit from the expertise and experience of others.
- Use wikis and forums to collaborate with the broader user community.

Terms used in this guide

BigFix terms are not always labelled with BigFix.

The following terms are all BigFix terms, but are used throughout the guide without being labeled every time with BigFix:

**Agent**

A computer on which the BigFix client is installed

**Console**

The BigFix console

**Client**

The BigFix client

**Server**

The BigFix server

**Relay**

The BigFix relay
Chapter 2. BigFix Site Administrator and Console Operators

In BigFix there are two basic classes of users:

**The Site Administrator**

The Site Administrator is responsible for installing and maintaining the BigFix software, and to run administrative tasks that globally affects the environment such as site-level signing keys management. There is only on Site Administrator for a BigFix environment. For more information, see [The Site Administrator (on page 22)](#).

**The Console Operators**

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

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

The Site Administrator

The site administrator has the following primary responsibilities:

**Obtaining and securing the Action Site Credentials**

To install BigFix, the site administrator must generate a private key, receive a license certificate from HCL, and create a masthead with the digital signature and configuration information. This is a special key and must be used only for site-level tasks such as:
• Setting global system options
• Editing Mastheads
• Administering Distributed Server Architecture (DSA)

Preparing the Server

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

Installing the various components

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

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

Maintaining the Server

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

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

The Site Administrator cannot:

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

There are two types of Console operators:

**Master Operators (MO)**

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

**Operators or Non-Master Operators (NMO)**

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

By default the Console operators cannot:

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

These and other permissions can be set by a master operator in the Permissions area of the Details tab of the operator's description. For more information about operators rights, see [Mapping authorized activities with permissions (on page 29)](#).

Different ways to define a Console Operator

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

- You can add single operators at any time by selecting the **Tools > Create Operator** item or by right clicking in the operators work area and selecting **Create Operator** as described in [Adding Local Operators (on page 25)](#).
• If you are using Active Directory or a generic LDAP, you can add previously defined users by selecting the **Tools > Add LDAP Operator** item or by right clicking in the operators work area and selecting **Add LDAP Operator** as described in **Adding LDAP Operators (on page 48)**.
• You can also associate an LDAP group to an existing role, in this way, with just one click, you add an operator for each user specified in the LDAP group and you associate that operator to the role. For more information about this capability, see **Associating an LDAP group (on page 51)**.

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

### Adding Local Operators

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

To add a local operator perform the following steps:

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

![Add User dialog](image)

2. Enter the **Username** of the person you want to designate as a publisher or operator.
3. Create a **Password** and retype it for confirmation. When you give the keys to your operators, they can change their passwords if they want.
4. Click **OK**. The **Console Operator** window opens.
5. From the **Details** tab, assign operator permissions.
You can control the default settings by using the `defaultOperatorRolePermissions` option in Advanced Options of the BigFix Administrative tool. For details, see List of advanced options (on page 26).

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Explicit Permissions</th>
<th>Effective Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master Operator</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Show Other Operators’ Actions</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Stop Other Operators’ Actions</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Can Create Actions</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Can Lock</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Can Send Refresh to Multiple Computers</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Can Submit Queries</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Custom Content</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Unmanaged Assets</td>
<td>Show All</td>
<td>Show All</td>
</tr>
</tbody>
</table>

where:

**Master Operator**

Specifies if the operator is a Master operator or not.

**Show Other Operator's Actions**
Specifies if the operator can see the actions submitted by other operators.

ていました: An operator with the **Show Other Operators' Actions** permission can see the action only in the following cases:

- If he is the owner of the action.
- If another operator submitted the action on at least one of his administered computers, and this computer is administered by both operators. In this case, the information is available only when the computer reports back the data to the BigFix server.

**Stop Other Operator's Actions**

Starting from BigFix Platform V9.5 Patch 5, you can specify if the operator can stop the actions submitted by other operators. To stop an action, triggered but not expired yet, the roles or the computer management rights assigned to the operator must be either identical or a superset of the roles or the computer management rights assigned to the operator who submitted the action. To use this capability, the operator must have also the **Show Other Operator's Actions** permission set to 'Yes'.

**Can Create Actions**

Specifies if the operator can create actions.

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

**Can Lock**

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

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

**Can Submit Queries**

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

**Custom Content**

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

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

**Unmanaged Assets**

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

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

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

<table>
<thead>
<tr>
<th>Restart and Shutdown</th>
<th>Explicit Permissions</th>
<th>Effective Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Post-Action Behavior</td>
<td>Allow Restart and Shutdown</td>
<td>Allow Restart and Shutdown</td>
</tr>
<tr>
<td>Action Script Commands</td>
<td>Allow Restart and Shutdown</td>
<td>Allow Restart and Shutdown</td>
</tr>
</tbody>
</table>

Depending on the configuration that you set for a specific operator for shutdown and restart, the radio button in the Take action panel might be disabled for that operator. This configuration has no effect on actions with type other than BigFix Action Script. You can also set permissions to access the BigFix user interfaces.
6. From the **Administered Computers** tab, you see the list of computers that this operator can manage. This list is populated after the computers that satisfy the criteria specified in the **Computer Assignments** tab report back their information to the BigFix server.

7. From the **Assigned Roles** tab, select the roles to apply to this operator.

8. From the **Sites** tab, assign the sites you want this operator to have access to.

9. From the **Computer Assignments** tab, specify the properties that must be matched by the computers that the operator can manage. For master operators, all the computers are assigned.

10. From the **WebUI Apps** tab, specify the WebUI Applications that the operator is allowed to access.

11. To save the changes click **Save Changes**.

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

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

### Mapping authorized activities with permissions

The following table shows which activities you can, cannot or could, under specific conditions, allow an operator to do by assigning permissions in the Details tab of the Operator Definition. For more information about operator's specific permissions, see *Adding Local Operators* (on page 25).

**Table 1. Mapping of authorized activities with operator permissions**

<table>
<thead>
<tr>
<th>Activities</th>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Fixlet Sites</td>
<td>No</td>
</tr>
<tr>
<td>Activities</td>
<td>Operator</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Change Client heartbeats</td>
<td>No</td>
</tr>
<tr>
<td>Create Fixlets</td>
<td>If Custom Content is set to YES</td>
</tr>
<tr>
<td>Create Tasks</td>
<td>If Custom Content is set to YES</td>
</tr>
<tr>
<td>Create Analyses</td>
<td>If Custom Content is set to YES</td>
</tr>
<tr>
<td>Create Baselines</td>
<td>If Custom Content is set to YES</td>
</tr>
<tr>
<td>Activate/Deactivate Analyses</td>
<td>Administered</td>
</tr>
<tr>
<td>Take Fixlet/Task/Baseline Action</td>
<td>Administered</td>
</tr>
<tr>
<td>Take Custom Action</td>
<td>If Custom Content is set to YES and Can Create Actions is set to YES</td>
</tr>
<tr>
<td>Stop Actions</td>
<td>Administered</td>
</tr>
<tr>
<td>Manage Administrative Rights</td>
<td>No</td>
</tr>
<tr>
<td>Manage Global Retrieved Properties</td>
<td>No</td>
</tr>
<tr>
<td>View Fixlets</td>
<td>Administered</td>
</tr>
<tr>
<td>View Tasks</td>
<td>Administered</td>
</tr>
<tr>
<td>View Analyses</td>
<td>Administered</td>
</tr>
<tr>
<td>View Computers</td>
<td>Administered</td>
</tr>
<tr>
<td>View Baselines</td>
<td>Administered</td>
</tr>
<tr>
<td>View Computer Groups</td>
<td>Administered</td>
</tr>
<tr>
<td>View Unmanaged Assets</td>
<td>Administered</td>
</tr>
<tr>
<td>View Actions</td>
<td>Administered</td>
</tr>
<tr>
<td>Make Comments</td>
<td>Administered</td>
</tr>
<tr>
<td>View Comments</td>
<td>Administered</td>
</tr>
<tr>
<td>Globally Hide/Unhide</td>
<td>No</td>
</tr>
<tr>
<td>Locally Hide/Unhide</td>
<td>Yes</td>
</tr>
<tr>
<td>Use Wizards</td>
<td>If Custom Content is set to YES</td>
</tr>
<tr>
<td>Remove computer from database</td>
<td>If Can Create Actions is set to YES</td>
</tr>
</tbody>
</table>
## Operators and analysis

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

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

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

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

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

There are several tabs to choose from:

- **Details**: Describes the operator by name and type and lets you select a login type. This is also where you can view and alter operator permissions.
- **Administered Computers**: Presents a list of computers that are currently assigned to the selected console operator.
- **Issued Actions**: Presents a list of actions that have been issued by the selected console operator.
• **Assigned Roles:** Displays the currently assigned roles, and lets you reassign them.

• **Sites:** Displays the sites currently assigned to this operator, and lets you reassign them.
  
  If the site is a custom site, you can also set Read/Write/Owner permissions.

• **Computer Assignments:** Lists the properties that must be matched by the computers that the operator can manage. If you specify a property to be matched, any time a computer is changed to match that property, it is added to the list of computers assigned to the operator. On the other hand, if a computer is changed not to match that property, that computer is removed from the list.

  This tab is available only for not-master operators.
Chapter 3. Integrating with LDAP

You can add Lightweight Directory Access Protocol (LDAP) associations to BigFix. That allows you and other users to log in to the console using those credentials. The same advantage applies also to Web Reports.

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

After you completed the steps to integrate with one of these two types of LDAP, you can associate LDAP users or groups to BigFix Console operators or roles as described in Adding LDAP Operators (on page 48) and Associating an LDAP group (on page 51).

Integrating with a Generic LDAP

Configure the integration with a Generic LDAP by adding an existing LDAP domain to the console as follows:

1. From the Tool menu, select Add LDAP Directory or right click in the work area and then select Add LDAP Directory. The Add LDAP Directory dialog appears.
2. Provide a name and from the Type pull-down, make sure **Generic LDAP Server** is selected. Note that no **global catalog** option is available on generic LDAP servers.

3. Fill in the information pertaining to your LDAP installation. Under **Server**, enter the host name or IP Address of the server.

4. Enter the port number, typically 636 if you are using Secure Sockets Layer (SSL).

5. Enter the base distinguished name (**Base DN**), of the form `dc=example,dc=com`.

6. Click the button to **connect anonymously** or to **use credentials**. If you choose to connect using credentials, enter your **User DN** and **password**.

7. Click **Test** to ensure you have entered your information correctly and a connection can be made to your LDAP.

8. If you want to include user or group filters, click the **Show advanced settings** link. After specified, all further LDAP searches will be subject to the appropriate filter.

9. Click **Add** to complete the LDAP setup.

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

**Integrating with Active Directory**
You can use Microsoft Active Directory (AD) to handle authentication on BigFix. That allows you and other users to log in to the console using your Active Directory credentials, taking advantage of your existing authentication policies. The same advantage applies also to Web Reports.

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

**Integrating the Windows server with Active Directory**

To add an existing Active Directory to the console, follow these steps:

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

   ![Add LDAP Directory](image)

2. Provide a name for the Active Directory and from the Type pull-down, make sure **Microsoft Active Directory** is selected.
3. Under **Server**, enter the host name, IP Address or fully qualified domain name of the server.

4. To access an entire Active Directory forest, click **This is a global catalog server**.

5. Click the button to **connect as the root server service user** or to **use credentials**. If you choose to connect using credentials, enter your Active Directory **Username** and **Password**.

6. Click **Test** to make sure you have entered your information correctly and a connection can be made to your Active Directory server.

7. Click **Add** to complete the Active Directory setup.

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

To add an existing Active Directory running over SSL, you must perform the following steps:

1. Select **Generic LDAP Server** as server type.
2. If the server is a global catalog server, specify as port number 3269.

3. Click the **Show advanced settings** link. The user filter and group filter options are displayed:
4. Enter UserPrincipalName in Login attribute.

Note: The UserPrincipalName attribute cannot be one of the following formats: domain/user, domain.com/user, or user.

5. Enter (objectClass=user) in User filter
6. Enter (objectClass=group) in Group filter.
7. Click Use the following credentials to connect to the directory server and enter your Active Directory Username and Password.
8. Click Test to ensure you have entered your information correctly and a connection can be made to your Active Directory server.
9. Click Add to complete the Active Directory setup.

Your Active Directory Server is now configured and available for use in the console.
Integrating the Linux server with Active Directory

Configuring Kerberos authentication

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

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

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

Preliminary Checks

Before running the integration between the BigFix server running on a Red Hat Enterprise Linux 6 or Linux 7 system and the Active Directory server, ensure that:

- The DNS host names of both the Red Hat Enterprise Linux 6 or Linux 7 system and the Active Directory server are resolved correctly, by performing the following steps on the Red Hat Enterprise Linux 6 system:
  1. Open the file `/etc/host` and ensure that both DNS host names are specified as fully qualified domain names.
  2. Open the file `/etc/sysconfig/network` and ensure that the host name of the Red Hat Enterprise Linux 6 or Linux 7 system is specified as fully qualified domain name.
- The time between the Active Directory and the Linux BigFix server is synchronized. If needed, you can synchronize the time service on the Red Hat Enterprise Linux 6 or Linux 7 system and the Active Directory server with the time source server, by performing the following steps:
1. In the file `/etc/ntp.conf` on the Red Hat Enterprise Linux 6 or Linux 7 system, replace the following lines:

```plaintext
server hostname
```

with:

```plaintext
server time_source_server_name
```

where `time_source_server_name` is the server hostname or IP address of the time source server used to synchronize the time.

2. When DNS lookups are not reliable, configure the Red Hat Enterprise Linux systems to perform DNS lookups from the Active Directory server by editing the `/etc/resolv.conf` file as follows:

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

3. Activate the change on the Red Hat Enterprise Linux 6 or Linux 7 system by:

   - Stopping the `ntp` daemon:

```plaintext
service ntpd stop
```

   - Updating the time:

```plaintext
ntpdate Red_Hat_server_IP
```

   - Starting the `ntp` daemon:

```plaintext
service ntpd start
```

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

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

where `time_source_server_name` specifies the list of DNS names or IP addresses for the NTP time source with which the Linux server synchronizes. For example,
you can specify time.windows.com as the NTP time server. When you specify multiple peers, use a space as the delimiter and enclose the names of the peers in quotation marks.

5. On the Active Directory server, run the following command to ensure that the time is synchronized with the time source server

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

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

```
    chkconfig ntpd on
```

**Installing the NSS and PAM libraries**

Ensure that the following NSS and PAM packages are installed:

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

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

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

**Configuring Authentication**

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

- `system-config-authentication` graphical tool
- `authconfig` command-line tool

**Using the system-config-authentication graphical tool**

To configure the authentication with the system-config-authentication tool, perform the following steps:
1. Run the `system-config-authentication` graphical tool to define LDAP as the user account database for user authentication.

2. In **Identity & Authentication**, from the **User Account Database** drop-down list, select **LDAP**. Selecting the **LDAP** option allows the system to be configured to connect to the Windows Active Directory domain using LDAP with Kerberos authentication.

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

4. In **LDAP Server** specify the address of the LDAP server such as `ldap://winserver.tem.test.com`

5. In **Authentication Method** select **Kerberos password**.
6. Configures the realm for the Kerberos server in **Realm**, such as `TEM.TEST.COM`. Ensure you enter the Realm name in uppercase.

7. Specify the **Key Distribution Center** *(KDC)* in **KDCs** for issuing Kerberos tickets, for example, `winserver.tem.test.com`.

8. Specify the administration servers running `kadmind` in the **Admin Servers**, such as `winserver.tem.test.com`.

9. Click **Apply**.

For more information about how to use this tool, see [Launching the Authentication Configuration Tool UI](#).

**Using the authconfig command-line tool**

To update all of the configuration files and services required for system authentication, you can run the `authconfig` command-line tool, as shown in the following example:

```bash
authconfig --enableldap --ldapserver=ldap://winserver.tem.test.com:389
--ldapbasedn="dc=tem,dc=test,dc=com" --enablekrb5
--krb5realm TEM.TEST.COM --krb5kdc winserver.tem.test.com:88
--krb5adminserver winserver.tem.test.com:749 --update
```

where:

--- **--enableldap**

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

--- **--ldapserver**

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

--- **--ldapbasedn**

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

--- **--enablekrb5**
Enables the Kerberos password authentication method.

---krb5realm

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

---krb5kdc

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

---krb5adminserver

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

---update

Applies all the configuration settings.

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

Modifying the local LDAP name

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

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

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

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

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

3. Restart the local LDAP name service daemon:

```
service nslcd restart
```

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

**Configuring the NSS and PAM libraries**

To use the LDAP database to authenticate users on a Linux system edit the `/etc/nsswitch.conf` and change `passwd`, `shadow` and `group` entries from the SSSD daemon (`sss`) to LDAP:

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

to LDAP (`ldap`):

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

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

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

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

For additional information on RedHat integration see [Integrating Red Hat Enterprise Linux 6 with Active Directory](#).
Integrating the server with Active Directory

Integrating the BigFix server with Active Directory

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

2. Provide a name for the Active Directory and from the Type pull-down, make sure Microsoft Active Directory is selected.
3. Under Server, enter the host name, IP Address or fully qualified domain name of the server.
4. To access an entire Active Directory forest, click This is a global catalog server.
5. Click the button to connect as the root server service user or to use credentials. If you choose to connect using credentials, enter your Active Directory Username and Password.
6. Click Test to make sure you have entered your information correctly and a connection can be made to your Active Directory server.
7. Click Add to complete the Active Directory setup.
Note: When you add an LDAP Server as **Microsoft Active Directory**, ensure that on the LDAP server you have defined the **UserPrincipalName** attribute corresponding to the **User logon name** of each user. This attribute value is used on the BigFix Console for each user authentication.

Adding LDAP Operators

You can create accounts for operators to access the console by using an existing Active Directory or LDAP account. When you select this option, an operator with the same name as the one specified in the LDAP directory, is added to the operators node in the Domain Panel on the BigFix console. These operators can then log in as usual, using one of the following notations:

- username
- username@domain
- domain\username

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

Note:

Starting from version 9.2.6 for accesses to Web UI and Web Reports, and from version 9.5 for accesses to the Console, you can integrate BigFix with SAML V2.0 to provide BigFix LDAP operators with:

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

For more information, see [Enabling SAML V2.0 authentication for LDAP operators](#) (on page 53).

To add an LDAP operator, complete the following steps:
1. Ensure that the needed Active Directory or LDAP directory is added to the BigFix environment.

2. Click the **Tools > Add LDAP Operator** menu item or right click in the work area and then select **Add LDAP Operator**. The Add LDAP User dialog appears.

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

4. When you find the user to add as LDAP operator, select it and click **Add**. The Console Operator panel opens.
5. From the Details tab assign operator permissions.

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

You can also set permissions to access the BigFix Console and REST API.

6. The Administered Computers tab lists the computers managed by this operator.

7. From the Assigned Role tab, select the roles that you want to assign or unassign this operator to.

8. From the Sites tab, assign the sites that you want this operator to have access to or unassign them.

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

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

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

### Associating an LDAP group

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

To add such a group perform the following steps:

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

![Create Role](image)

Enter a name for your group and click **OK**.
3. The **Role** panel appears.
Click the **LDAP Groups** tab.

4. Select the LDAP group that you want to assign to this role and click **Assign LDAP Group**.

5. To save the changes click **Save Changes**.

When you assign an LDAP group to a role, any user from that group can then log in to the console. Only those users who actually log in will be provisioned with accounts and thus end up in the list of operators. This avoids the creation of unnecessary accounts. Operators are granted the highest privileges resulting from the sum of all their roles and permissions. For instance, if a user has access to computer set A and sites X from role 1, and computer set B and sites Y from role 2, they will have permissions for Sites X and Y across both computer sets A and B.
Chapter 4. Enabling SAML V2.0 authentication for LDAP operators

Starting from Version 9.5.5, BigFix supports SAML V2.0 authentication via LDAP-backed SAML identity providers. After configuration, SAML V2.0 support enables:

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

What Is SAML 2.0

The OASIS Security Assertion Markup Language (SAML) is a standard that uses an XML-based framework to describe and exchange security information between online entities. SAML 2.0 supports:

**Web-Based Single Sign-On**

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

**Identity federation**

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

This type of sharing helps to reduce identity management costs.

Federated identity implements FIPS 201 to define a US Government-wide interoperable identification credential, known as the Personal Identity
Verification (PIV), for controlling physical access to federal facilities and logical access to federal information systems.

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

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

**Web services and other industry standards**

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

**How SAML works**

The SAML specification defines three parties:

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

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

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

For more information about SAML V2.0 use case scenarios, see [SAML V2.0 Overview](#).
Which BigFix user interfaces integrate with SAML V2.0

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

How BigFix integrates with SAML V2.0

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

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

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

• If you want to use the SAML authentication for Web Reports and for the BigFix console only, and you do not need to use it with any WebUI application, you can start the WebUI in SAML-only mode. This SAML configuration allows you to minimize resource consumption. For more information about how to set up this configuration, see Enabling the WebUI in SAML-Only Mode.

• If you want to use the SAML authentication for all the BigFix user interfaces, including the full set of WebUI components, or the WebUI ETL process, follow the instructions provided in WebUI Installation Checklist if are using BigFix Version 9.5.5 or later.

If the BigFix environment uses one LDAP server as a user repository, user provisioning is not affected by this integration, and administrators continue to define operators and roles to authorize them to use BigFix services. If your BigFix environment operators are defined on more than one LDAP server, read carefully the information provided in Assumptions and requirements (on page 56).

Integration with SAML 2.0 maintains existing audit scenarios and includes SAML-authenticated user entries in the server_audit.log file.
See the following sample use case:

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

Assumptions and requirements

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

- BigFix supports SAML V2.0 authentication with an SAML V2.0-compliant identity provider such as Active Directory Federation Services (ADFS).
- The SAML V2.0 authentication is restricted to:
  - Only one SAML IdP backed by one or more LDAP directories. If you already defined multiple LDAP servers as user repositories in your BigFix environment, be aware that, after enabling SAML authentication, only the users and the groups managed by the selected IdP will still be known to the BigFix environment. In this case, ensure that your IdP environment is correctly configured so that the SAML IdP (ADFS or ISAM) can authenticate users from the different LDAP environments that you want to use as the user repository.
  - Identity providers using SHA256 as secure hash algorithm.
  - Web Reports servers connecting to only one data source (Root server) and configured with SSL.
• To configure and use SAML authentication, you must have the WebUI installed. If you are using the WebUI solely for providing SAML authentication for Web Reports and the BigFix console, you can start the Web UI in SAML-only mode to reduce resource consumption. For information about how to start the Web UI in SAML-only mode, see https://help.hcltechsw.com/bigfix/9.5/webui/WebUI/Admin_Guide/c_saml_2_0.html?hl=saml%2C2.0.

• In DSA architecture, the configuration is replicated to replica DSA servers. However, the replica does not enable WebUI for SAML on non-primary DSA's, because multiple WebUI configuration is not supported.

• When running Web Reports, if SAML is enabled, the check on the referrer is not performed. You can use the setting _HTTPServer_Referrer_CheckEnabled to enable or disable the referrer check. The referrer is an optional header of the HTTP protocol. It identifies the address of the web page (that is the URI or IRI) that linked to the resource being requested. For information about how BigFix manages the referrer check, see List of settings and detailed descriptions (on page ).

What changes from the BigFix user's perspective

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

After enabling SAML authentication for LDAP users:

LDAP operators:

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

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

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

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

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

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

**Local non-LDAP operators:**

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

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

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

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

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

After SAML is configured and enabled only local non-LDAP users will be able to log in using API; the 4-eyes authentication approvers must be local accounts.
How to configure BigFix to integrate with SAML 2.0

Before configuring the integration, ensure that:

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

The overall configuration comprises two parts:

- The configuration of the SAML identity provider for explicit two-factor authentication, which is under the responsibility of the identity provider administrator. For what concerns this part, ensure that:
  - The redirect URLs are added to the relying party trust indexed, with binding HTTPS_POST, and in this format:
    - https://<WebUI_server>/saml (for the Web UI server, assuming that it listens on port 443)
    - https://<Web_Reports_server>:8083/saml (for each Web Reports server, assuming that they listen on port 8083)
    - https://<Bigfix_server>:52311/saml (for the BigFix Console)

  **Note:** If the identity provider is ADFS, the redirect URLs must be added, as SAML Assertion Consumer Endpoints, in the Endpoints tab inside the ADFS Relying Party Trust properties.
  - In the Identity Provider configuration, the login setting must be set for FORMS login.
  - If you plan to use the smart card authentication, ensure that the Identity Provider is correctly configured to use multi factor authentication. For example, if you use ADFS, ensure that at least one between Certificate Authentication and Windows...
Authentication, if you want to use the Windows Integrated Authentication, is enabled in the Global Authentication Policy configuration.

- For Active Directory user authentication, set the identity provider Claim Rules as follows:

  **Attribute store:**
  
  Active Directory

  **Mapping of LDAP attributes to outgoing claim types:**
  
  - LDAP Attribute: User-Principal-Name
  - Outgoing Claim: Name ID

- The configuration to allow the BigFix server to use SAML authentication, which is a Master Operator (MO) and Web Reports administrator responsibility. Complete these steps to accomplish this task:
  
  1. Configure LDAP with Active Directory in the BigFix Console. For more details, see [Integrating the Windows server with Active Directory](on page 36).
  2. Define LDAP operators. For more details, see [Adding LDAP Operators](on page 48).
  3. Define Web Reports LDAP operators in the Web Reports user management pages.
  4. Access the Administration page to configure the integration with SAML 2.0:
     
     a. Log in to the Web UI server:
        
        - If the Web UI listens on port 443: `https://<WebUI_server>`
        - If the Web UI listens on a port that is different than 443: `https://<WebUI_server>:<webui_port_number>`

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

   **Entry Point:**

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

   **Signing Certificate:**

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

   **Issuer:**

   Enter the Identity Provider Identifier in a textual format, for example "BigFix". If you are configuring ADFS configuration, this value must match the ADFS Relying Party Identifier setting.

6. After filling in all the fields, click **Enable**.

7. If WebUI is installed on a separate remote server, set the `_WebUI_AppServer_Hostname` key of the BigFix server computer to the hostname, fully qualified domain name (FQDN) or IP address of the computer where the WebUI is installed (the WebUI Server computer), ensuring that it matches the WebUI certificate subject name.

8. If you want to enable the use of smart cards as SAML authentication method, set on the WebUI Server computer the `_WebUIAppEnv_SAML_AUTHNCONTEXT` setting to one of the following two values:
° urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient if the Identity Provider is set to use the Transport Layer Security (TLS) cryptographic protocol.

° urn:federation:authentication:windows if the Identity Provider is set to use Integrated Windows Authentication (IWA).

Then restart the WebUI.

9. Restart the BigFix root server.
10. Restart the BigFix Web Reports services.

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

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

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

**Note:** After completing these steps, to prevent errors when logging on to the BigFix console, ensure that you set for the _BESDataServer_AuthenticationTimeoutMinutes configuration setting a value, specified in minutes, bigger than 5 minutes.
Chapter 5. Using multiple servers (DSA)

Here are some of the important elements of multiple server installations:

- Depending on the platform where you plan to install the additional server, you can follow the procedures described in Installing Additional Windows Servers (DSA) (on page [153]) or Installing Additional Linux Servers (DSA) (on page [458]).
- Servers communicate on a regular schedule to replicate their data. To review the current status and adjust the replication interval, see Managing Replication (DSA) on Windows systems (on page 67) or Managing Replication (DSA) on Linux systems (on page 68).
- When each server is ready to replicate from the other servers in the deployment, it calculates the shortest path to every other server in the deployment. Primary links are assigned a length of 1, secondary links 100, and tertiary links 10,000. Links that resulted in a connection failure the last time they were used are considered to be non-connected.
- When an outage or other problem causes a network split, it is possible for a custom Fixlet or a retrieved property to be modified independently on both sides of the split. When the network is reconnected, precedence goes to the version on the server with the lowest Server ID.
- If multiple copies of Web Reports are installed, they operate independently. Each Web Report server can connect to the server that is most convenient, because they all contain equivalent views of the database.
- By default, server 0 (zero) is the master server. The BigFix Administration Tool on Windows and the BESAdmin command on Linux only allow you to perform certain administrative tasks (such as creating and deleting users) when connected to the master server.

Disaster Server Architecture (DSA)

The following diagram shows a typical DSA setup with two servers. Each Server is behind a firewall, possibly in a separate office, although it is easy to set up multiple servers in a single
office as well. The servers must have high-speed connections to replicate the BigFix data (generally LAN speeds from 10 to 100Mbps are required). The BigFix servers communicate over ODBC and HTTP protocols.

In case of a failover, the specific configured relays automatically find the backup server and reconnect the network. For more information about the relay configuration see Configuring relay failover (on page 65).
Configuring relay failover

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

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

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

If Message Level Encryption is enabled and clients are set using **Task: BES Client Setting: Encrypted Reports**, move the BigFix server encryption key to the secondary BigFix DSA server. This enables the BigFix DSA server to process reports from encrypted BigFix clients during normal operations or in the event of an outage on the primary BigFix server.
Copy the encryption key (.pvk) from the BigFix server directory:

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

to the DSA secondary server.

### Managing Replication (DSA) on Windows systems

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

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

### Changing the replication interval on Windows systems

On Windows systems if you have multiple servers in your deployment, you can schedule when each one replicates. The default is five minutes, but you can shorten the time for greater recoverability or increase it to limit network activity:

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

By default, server 0 (zero) is the master server. The Administration Tool allows you to perform certain administrative tasks (such as creating and deleting users) only when you are connected to the master server. If you want to switch the master to another server, you must set the deployment option `masterdatabaseServerID` to the other server ID. Here is how:

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

After the value has successfully replicated to the new server, it become the master server. If a server suffers a failure while it is the master, another server must be made the master server by direct manipulation of the `ADMINFIELDS` table in the database. The details of this are beyond the scope of this guide, but broadly speaking, you might use a tool like SQL Enterprise Manager to view and alter the `ADMINFIELDS` table. Set the variable name `masterDatabaseServerID` to the value you want.

Managing Replication (DSA) on Linux systems

To install additional Linux servers, follow the procedure described in Installing Additional Linux Servers (DSA) (on page 68).

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

Changing the replication interval on Linux systems
On Linux systems if you have multiple servers in your deployment, you can schedule when each one replicates. The default is five minutes, but you can shorten the time for greater recoverability or increase it to limit network activity:

To change the replication interval, perform the following steps:

1. From the `/opt/BESServer/bin` command prompt, start the command line:
   ```
   ./iem login --server=servername:serverport --user=username
   --password=password
   ```

2. From the `/opt/BESServer/bin` command prompt, run the following command:
   ```
   ./iem get replication/server/0 > /appo/replicationServer0.xml
   ```

3. In the `/appo/replicationServer0.xml` file, edit the following keyword:
   ```
   <ReplicationIntervalSeconds>300</ReplicationIntervalSeconds>
   ```
   to change the value in seconds of the replication interval. Using longer replication intervals means that the servers replicate data less often, but have more data to transfer each time.

   ```xml
   <?xml version="1.0" encoding="UTF-8"?>
   <BESAPI xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="BESAPI.xsd">
   replication
   /server/0">
   <ServerID>0</ServerID>
   <URL>http://nc926068.romelab.it.ibm.com:52311</URL>
   <DNS>nc926068.romelab.it.ibm.com</DNS>
   <ReplicationIntervalSeconds>300</ReplicationIntervalSeconds>
   replication
   /server/0/link/3">
   <SourceServerID>0</SourceServerID>
   ```
4. Upload the modified file by running the following command:

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

Switching the master server on Linux systems

By default, server 0 (zero) is the master server. To switch the master to another server, set the deployment option `masterDatabaseServerID` to the other server ID as follows:

1. From the `/opt/BESServer/bin` command prompt, start the command line:
To switch the master server to another master server:

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

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

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

3. In the `/appo/switchmaster.xml` file, add or edit the following keyword and its value:

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

4. Upload the modified file by running the following command:

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

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

If a server suffers a failure while it is the master, another server must be made the master server by direct manipulation of the ADMINFIELDS table in the database.

**Schema next tables regenerated during upgrade**

When upgrading DSA environments both on Windows and Linux systems to BigFix Version 9.5 Patch 13 from older Versions, the schema next tables, used by the WebUI component,
are regenerated starting from the legacy database tables if the replication process did not complete successfully.

This behavior affects only the DSA environments on which you enabled the DisableReplicationOfNextTables registry setting (on Windows) or the DisableReplicationOfNextTables configuration setting (on Linux). This setting prevents the replication of the schema next tables in your environment.

**Before the upgrade**

If you enabled the DisableReplicationOfNextTables registry setting (on Windows) or the DisableReplicationOfNextTables configuration setting (on Linux), to save the amount of time required by the schema next tables regeneration, it is strongly recommended that you perform a clean up of the environment using the BigFix Administration Tool tab named Clean Up. For details, see Clean Up (on page ). This operation will reduce the amount of time required for the upgrade.

**After the upgrade**

The DisableReplicationOfNextTables registry/configuration setting is automatically disabled.
Chapter 6. Server object IDs

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

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

- ActionID
- FixletID
- ID
- ContentID
- RoleID

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

Before the current implementation, the maximum number of available object ids per server was 16,777,215, and consequently the maximum number of available DSA servers was 256.

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

|x__|___y__|____________z____________|

where:

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

In this way, the number of available object ids was increased to 1,627,389,951 and consequently the number of available DSA servers was decreased to 32.

This solution has the advantage of keeping the 32-bit object id so to avoid any backward compatibility issue.
Chapter 7. Customizing HTTPS for Gathering

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

To enable the HTTPS protocol, you must set a client keyword.

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

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

Enabling HTTPS

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

On the BigFix Server:

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

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

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

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

The default value for this setting is 2.

In the airgapped environment:

Launch the `Airgap` command as follows:

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

```
Airgap -usehttps
```

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

```
Airgap -no-usehttps
```

The server uses the protocol defined in the URL.

## Validating HTTPS certificates

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

The Windows default path is:

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

The Linux default path is:

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

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

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

2. **On the Server:**

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

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

`Airgap -cacert <path>`

where `<path>` is the path of the saved set of trusted certificates.
Chapter 8. Configuring secure communication

BigFix automatically enables the Secure Socket Layer (SSL) protocol by using self-signed certificates to ensure secure communication between your Web Reports or Rest API server and all users that access it. If you don’t want to use the provided private key and the self-signed certificate complete the following steps:

1. Generate a private key and a certificate signing request (CSR) for a CA signed certificate. For additional information on the private key and certificate format see Private key and certificate format (on page 86).

   The advantage of using an external CA is that root certificates of known public CAs are imported by default into modern web browsers.

   ☢️ **Important:** You can use the private key and the certificate generated for BigFix Inventory also for Web Reports only if the private key is not password protected.

   For additional information on how to get these files see Creating private keys and certificates (on page 89) and Signing certificates (on page 90).

2. Copy the files to a folder of your choice on the Web Reports or Rest API server.
3. Configure the Web Reports server or REST API server as described in Customizing HTTPS on Web Reports (on page 78) and Customizing HTTPS on REST API (on page 83).

   ☢️ **Note:** You can also configure Web Reports or Rest API to work with Hyper Text Transport Protocol Secure (HTTPS) manually without using the console. For additional information, see Configuring HTTPS manually on Windows systems (on page ) and Configuring HTTPS manually on Linux systems (on page ) for Web Reports, and Customizing HTTPS manually on Windows systems (on page 84) and Configuring HTTPS manually on Linux systems (on page ) for Rest API.

4. Depending on which component you are setting HTTPS, restart the corresponding service, **BESWebReports** for Web Reports and **BES Server** for Rest API:
Web Reports

- On Windows, open Services, select BESWebReports and on the Action menu, click Restart.
- On Linux, run from the prompt: `service beswebreports restart` or `/etc/init.d/beswebreports restart`.

Rest API

- On Windows, open Services, select BESServer and on the Action menu, click Restart.
- On Linux run from the prompt: `service besserver restart` or `/etc/init.d/besserver restart`.

Customizing HTTPS on Web Reports

If you have a trusted SSL security certificate and key from a certificate authority, you can configure the BigFix Web Reports computer to use this certificate and key to enable trusted connections.

Complete the steps to accomplish the following tasks:

- Specify that you are using a secure communication.
- Specify where the SSL certificate and private key files are located.
- Define the HTTPS port number, listening for HTTPS connections and redirecting the client to HTTPS on the SSL port.

1. From the BigFix console, select the Computers tab.
2. Right click on the computer on which Web Reports runs and click Edit Computer Settings.
3. Look for the `_WebReports_HTTPServer_UseSSLFlag` setting. If it exists, do not create a second one, but edit its value to 1 to enable HTTPS. If it does not exist, add it.
Important: If you combined the private key file with the certificate file, skip the following step and set only the `_WebReports_HTTPServer_SSLCertificateFilePath` setting.

4. Look for the `_WebReports_HTTPServer_SSLPrivateKeyFilePath` setting. If it exists, do not create a second one, but edit its value to the full path name of the private key (.pvk file) which contains the private key for the server. The private key must not have a password. If it does not exist, add it.

5. Look for the `_WebReports_HTTPServer_SSLCertificateFilePath` setting. If it exists, do not create a second one, but edit its value to the full path name of the .pem file which might contain both the certificate and private key for the server, or only the certificate. If it does not exist, add it.

Ensure that the .pem file is in standard pem file format.

The certificate is supplied by the server to connecting clients (browsers) and they present a dialog to the user containing information from the certificate. If the certificate meets all of the trust requirements, then the browser connects without any interventions by the user. If the certificate does not meet the trust requirements of the browser, then the user is prompted with a dialog that asks if it is OK to proceed with the connection, and giving them access to information about the certificate. A trusted certificate is signed by a trusted authority (such as Verisign), contains the correct host name, and is not expired.
Note: These settings are stored in the registry under the key `HKLM/Software/WoW6432Node/BigFix/EnterpriseClient/Settings/Client` of the Web Reports computer.

6. Look for the `_WebReports_HTTPServer_PortNumber` setting. If it exists, do not create a second one, but edit its value to the port number you would like to use. If it does not exist, add it:

![Add Custom Setting](image)

7. When SSL is enabled define the forwarding port with the following settings:
   - `_WebReports_HTTPRedirect_Enabled` to 1
   - `_WebReports_HTTPRedirect_PortNumber` to the port listening for HTTP connection and redirecting the client to HTTPS.

8. To require TLS12 for web browser requests, look for `_WebReports_HTTPServer_RequireTLS12`. If it exists, do not create a second one, but edit its value to 1. The Web Reports component always uses TLS 1.2 when communicating with the BigFix server, regardless of local settings or settings of the masthead.

   Important: Use of a TLS with a version earlier than 1.2 is deprecated.

9. Restart the BES Web Reports Server service:
   - On Windows, open Services, select BESWebReports and on the Action menu, click Restart.
   - On Linux run from the prompt: `service beswebreports restart` or `/etc/init.d/beswebreports restart`

You can also set the secure communication using a manual procedure as described in Customizing HTTPS manually on Linux systems (on page 82) and Customizing HTTPS manually on Windows systems (on page 81).
Customizing HTTPS manually on Windows systems

When you have a trusted SSL certificate (a .pem file), place it on the computer running Web Reports (usually the server) and follow these steps:

1. Run `regedit` and locate `HKEY_LOCAL_MACHINE\Software\BigFix\EnterpriseClient\Settings\Client` for x32 systems and `HKEY_LOCAL_MACHINE\Software\Wow6432Node\BigFix\EnterpriseClient\Settings\Client` for x64 systems.

   You need to add or modify subkeys for the HTTPS flag, for the location of the SSL certificate, for the HTTPS port number, and for the redirection to HTTPS.

2. Create a subkey of `Client` called `_WebReports_HTTPServer_UseSSLFlag` (if it does not yet exist).

3. Create a string value (reg_sz) for the key `_WebReports_HTTPServer_UseSSLFlag` called `value` and set it to 1 to enable HTTPS.

4. Create a subkey of `Client` called `_WebReports_HTTPServer_SSLCertificateFilePath` (if it does not yet exist).

5. Create a string value (reg_sz) for the key `_WebReports_HTTPServer_SSLCertificateFilePath` called `value` and set it to the full path name of the SSL certificate (cert.pem).

6. Create a subkey of `Client` called `_WebReports_HTTPServer_PortNumber` (if it does not yet exist).

7. Create a string value (reg_sz) for the key `_WebReports_HTTPServer_PortNumber` called `value` and set it to the port number you want to use (typically 443).

8. Create a subkey of `Client` called `_WebReports_HTTPRedirect_Enabled` (if it does not yet exist).

9. Create a string value (reg_sz) for the key `_WebReports_HTTPRedirect_Enabled` called `value` and set it to 1 to enable the browser redirection to HTTPS.

10. Create a subkey of `Client` called `_WebReports_HTTPRedirect_PortNumber` (if it does not yet exist).

11. Create a string value (reg_sz) for the key `_WebReports_HTTPRedirect_PortNumber` called `value` and set it to the number of the port listening for HTTP connection and redirecting the client to HTTPS.

12. Restart the `BESWebReports` service.
Customizing HTTPS manually on Linux systems

When you have a trusted SSL certificate (a .pem file), place it on the computer running Web Reports and customize the keywords in the applicable file:

- `besclient.config` - if a client is installed together with Web Reports
- `beswebreports.config` - if only Web Reports is installed

To define the port number you want to use:

```plaintext
[Software\BigFix\EnterpriseClient\Settings\Client\_WebReports_HTTPServer_PortNumber]
value = 443
```

To define the full path name of the SSL certificate (cert.pem):

```plaintext
[Software\BigFix\EnterpriseClient\Settings\Client\_WebReports_HTTPServer_SSLCertificateFilePath]
value = /tmp/CERT/cert.pem
```

To enable HTTPS:

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

To enable client redirection from an HTTP connection to an HTTPS connection:

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

To define the number of the port listening for the HTTP connection and redirecting the Client to HTTPS:

```plaintext
[Software\BigFix\EnterpriseClient\Settings\Client\_WebReports_HTTPRedirect_PortNumber]
```
Customizing HTTPS on REST API

If you have a trusted SSL security certificate and key from a certificate authority, you can configure the BigFix root server to use this certificate and key to enable trusted connections. After you have completed the configuration, connections from the REST API and console use this trusted certificate.

Complete the steps to accomplish the following tasks:

• Specify that you are using a secure communication.
• Specify where the SSL certificate and private key files are located.

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

   ![Add Custom Setting](image)

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

![Add Custom Setting](image)

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

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

6. To require TLS12, look for _BESRelay_HTTPServer_RequireTLS12. If it exists, do not create a second one, but edit its value to 1.

⚠️ Note: The REST API component always uses TLS 1.2 when communicating with the BigFix server, (regardless of local settings or settings of the masthead).

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

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

Customizing HTTPS manually on Windows systems
If you have a trusted SSL security and a key from a certificate authority (.pem file), you can configure the computer running REST API (usually the server) to customize trusted connections. After you have completed the configuration, connections from the Rest API and console use this trusted certificate. Complete the following steps:

1. Run `regedit` and locate `HKEY_LOCAL_MACHINE\Software\BigFix\EnterpriseClient\Settings\Client` for x32 systems and `HKEY_LOCAL_MACHINE\Software\Wow6432Node\BigFix\EnterpriseClient\Settings\Client` x64 systems.

   You need to add or modify subkeys for the HTTPS flag, and for the location of the SSL certificate.

2. Create a subkey of Client called `_BESRelay_HTTPServer_UseSSLFlag` (it might already exist).

3. Create a string value (reg_sz) for the key `_BESRelay_HTTPServer_UseSSLFlag` called `value` and set it to 1 to enable HTTPS.

4. Create a subkey of Client called `_BESRelay_HTTPServer_SSLCertificateFilePath` (it might already exist).

5. Create a string value (reg_sz) for the key `_BESRelay_HTTPServer_SSLCertificateFilePath` called `value` and set it to the full path name of the SSL certificate (cert.pem).

6. Restart the BES Root Server service.

Customizing HTTPS manually on Linux systems

If you have a trusted SSL security certificate and key from a certificate authority (.pem file), you can configure the BigFix root server to use this certificate and key to enable trusted connections. After you have completed the configuration, connections from the REST API and console use this trusted certificate.

This procedure describes how you can configure the BigFix root server on Linux systems to use a certificate to enable trusted connections through the REST API and BigFix console.

1. Save the file `em.pem`, containing both the certificate and the private key, in a protected area of the file system, where it can be accessed by the BigFix `besserver` process, for example, `/etc/opt/BESServer/em.pem`
2. Edit the `/var/opt/BESServer/besserver.config` file, adding the following two entries, and using `/etc/opt/BESServer/em.pem` as an example:

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

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

3. Stop and restart the BigFix root server.

**Private key and certificate format**

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

**Private key format**

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

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

**X509 certificate format**

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

```plaintext
BEGIN CERTIFICATE-----
```
If you received the root certificate (ca_root.crt) in addition to the intermediate certificate, combine them as follows:

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

**Single file (private key with certificates) format**

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

- Private key and primary certificate:

```
-----BEGIN CERTIFICATE-----
<<primary certificate: certificate.crt>>
-----END CERTIFICATE-----
-----BEGIN PRIVATE KEY-----
<<private key: base64 string from private.key>>
-----END PRIVATE KEY-----
```
• Private key, primary certificate and intermediate certificate:

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

-----BEGIN CERTIFICATE-----
<<intermediate certificate: base64 string from ca_intermediate.crt>>
-----END CERTIFICATE-----

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

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

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

-----BEGIN CERTIFICATE-----
<<intermediate certificate: base64 string from ca_intermediate.crt>>
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
<<root certificate: base64 string from ca_root.crt>>
-----END CERTIFICATE-----

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

If your file has DER-encoded or other formats, you can convert it to the PEM format, for example by using openSSL.
Creating private keys and certificates

To improve security, create your own private key and a certificate instead of using the self-signed ones that are available in BigFix Web Reports by default. You can use openSSL to create a private key and a certificate signing request (CSR) that can be transformed into a certificate after it is signed by a certificate authority (CA).

This procedure is valid for all operating systems that support openSSL.

If you are generating an encrypted private key in the pkcs8 format, add the following line to the `installation_dir/jre/lib/security/java.security` file:

```bash
security.provider.10=org.bouncycastle.jce.provider.BouncyCastleProvider
```

Then, restart the BigFix Web Reports server.

1. Open the command line.

2. Create a new private key.

   ```bash
   openssl genrsa -out key_name.key key_strength -sha256
   ```

   For example, `openssl genrsa -out private_key.key 2048 -sha256`

   Where:

   - **key_name**
     
     File name for your new private key.
   
   - **key_strength**
     
     Key strength, measured in bits. The maximum value that you can use for BigFix Web Reports is 2048 bits.

3. Create a certificate signing request (CSR). The request is associated with your private key, and is later transformed into a certificate.

   ```bash
   openssl req -new -key path_to_private_key.key -out csr_name.csr
   ```

   For example, `openssl req -new -key private_key.key -out CSR.csr`

   Where:
**path_to_private_key**

Path to your private key.

**csr_name**

File name for your certificate signing request (CSR).

After you run the command, you are asked to provide information that helps your users to identify your certificate and ensure that it can be trusted. The following excerpt from the command line is filled in with sample information:

```
Country Name (2 letter code) [XX]: US
State or Province Name (full name) [-]: New York
Locality Name (eg, city) [Default City]: New York
Organization Name (eg, company) [Default Company Ltd]: IBM
Organizational Unit Name (eg, section) [-]: Software
Common Name (eg, your name or your server's hostname) [-]: webreports.ibm.com
Email Address [-]: webreports@ibm.com
```

After completing these steps, two files are created: your private key (.key) and the certificate signing request (.csr). You must now sign the request to transform it into the certificate. For information about how to create a private certificate authority (CA) to sign the request, see [Signing certificates](on page 90).

### Signing certificates

Your certificate signing request (CSR) must be signed by a certificate authority (CA) to transform it into a certificate that can be uploaded to BigFix Web Reports. You can use the openSSL cryptographic library to create a private CA and sign your request.

There are other ways to sign the request aside from using a private CA. You can also use the CA of your organization or send the request to internationally trusted CAs, such as Entrust or Verisign. The certificates of these CAs are often trusted by default and do not display any warnings in the browser. Warnings might be displayed if you use a private CA.
1. Create a private certificate authority (CA) and a certificate for it.

   a. Create a private CA. This step creates a private key (.key) and a request (.csr) similar to those that you created in Creating private keys and certificates (on page 89).

      ```
      openssl req -new -newkey rsa:key_strength -nodes -out CA_csr_name.csr -keyout CA_key_name.key -sha256
      ```

      For example, `openssl req -new -newkey rsa:2048 -nodes -out CA_CSR.csr -keyout CA_private_key.key -sha256`

      Where:

      **key_strength**

      Key strength, measured in bits. The maximum value that you can use for BigFix Web Reports is 2048 bits.

      **CA_csr_name**

      File name for the certificate signing request (CSR). The certificate authority (CA) requires a separate request.

      **CA_key_name**

      File name for the private key. The certificate authority (CA) requires a separate private key.

   b. Create a certificate for your private CA. This step creates a certificate (.arm) that you can use to sign your CSR.

      ```
      openssl x509 -signkey path_to_CA_key.key -days number_of_days -req -in path_to_CA_csr.csr -out CA_certificate_name.arm -sha256
      ```

      For example, `openssl x509 -signkey CA_private_key.key -days 90 -req -in CA_CSR.csr -out CA_certificate.arm -sha256`

      Where:
key_strength

Key strength, measured in bits. The maximum value that you can use for BigFix Web Reports is 2048 bits.

path_to_CA_csr

File name for the certificate signing request (CSR) that you created for the certificate authority (CA).

path_to_CA_key

File name for the private key that you created for the certificate authority (CA).

number_of_days

Number of days for the new certificate to be valid.

CA_certificate_name

File name for the certificate of your CA. This certificate is used to sign your CSR.

2. Use the CA certificate to sign the certificate signing request that you created in Creating private keys and certificates (on page 89).

```bash
openssl x509 -req -days number_of_days -in path_to_csr.csr -CA path_to_CA_certificate.arm -CAkey path_to_CA_key.key -out new_certificate.arm -set_serial 01 -sha256
```

For example, `openssl x509 -req -days 90 -in CSR.csr -CA CA_certificate.arm -CAkey CA_private_key.key -out certificate.arm -set_serial 01 -sha256`

Where:

number_of_days

Number of days for the new certificate to be valid.

path_to_csr

Path to certificate signing request (CSR) that you want to sign.
**path_to_CA_certificate**
Path to certificate that you created for the certificate authority (CA).

**path_to_CA_key**
Path to the private key that you created for the certificate authority (CA).

**new_certificate**
File name for the new certificate that is created from your certificate signing request (CSR). You upload this certificate together with your private key to BigFix Web Reports.

You signed your certificate signing request and obtained a new certificate. You can now enable SSL in BigFix Web Reports and upload your private key and the certificate. These files replace the self-signed certificate that is already available in BigFix Web Reports, and thus ensure secure communication.
Chapter 9. Downloading files in air-gapped environments

In air-gapped environments, to download and transfer files to the main BigFix server, use the Airgap utility and the BES Download Cacher utility.

Overview

In an air-gapped environment where a secure network is physically isolated from insecure networks, such as the public Internet or an insecure local area network, and the computers on opposite sides of the air gap cannot communicate, to download and transfer files to the main BigFix server, you can use the Airgap utility and the BES Download Cacher utility.

Note: The Airgap utility does not support a configuration where the clients are air-gapped separately from the main BigFix server. The clients must be air-gapped together with the main BigFix server to be able to gather across the network from the main BigFix server.

Starting from BigFix Version 9.5.5, you have two different modes to work in an air-gapped environment. The "Extraction usage" mode, that was already available before Version 9.5.5, and the new "Non-extraction usage" mode.

Non-extraction usage overview

The "Non-extraction usage" mode is available only starting from BigFix Version 9.5.5.

Airgap might need to work without extracting any information from the BigFix server because in some places a rule forbids to extract any information in a secure network and move to an external network, such as Internet. To satisfy these requirements, the Airgap tool can now work without creating any Airgap request.

You can use the Airgap tool in three different ways:

Gather site contents
1. Run the Airgap tool on the internet facing computer to gather license information and create a site list file, which contains information related to the sites that you have licensed.
2. Edit the site list file and change the flags to specify the sites that you want to gather contents from.
3. Run the Airgap tool on the internet facing computer to gather license information and site contents as specified by the site list file into the Airgap response.
4. Move the Airgap response to the BigFix server.
5. Run the Airgap tool on the BigFix server to load the Airgap response into the BigFix server.
1. Run the Airgap tool on the internet facing computer to gather license information and create a site list file, which contains information related to the sites that you have licensed.
2. Edit the site list file and change the flags to specify the sites that you want to gather contents from, and the sites from which you want to download referenced files.
3. Run the Airgap tool on the internet facing computer to gather license information and site contents as specified by the site list file into the Airgap response, and then download the files referenced by the Fixlets.
4. Move the Airgap response and the downloaded files to the BigFix server.
5. Run the Airgap tool on the BigFix server to load the Airgap response into the BigFix server, and copy the downloaded files to the cache folder of the BigFix server.

Gather site contents and download files selectively
1. Run the Airgap tool on the internet facing computer to gather license information and create a site list file, which contains information related to the sites that you have licensed.

2. Edit the site list file and change the flags to specify the sites that you want to gather contents from, and sites from which you want to download referenced files.

3. Run the Airgap tool on the internet facing computer to gather license information and site contents as specified by the site list file into the Airgap response, and then create a file list file, which contains information about the referenced files.

4. Edit the file list file to specify the files that you want to download.

5. Run the Airgap tool on the internet facing computer to download the files as specified by the file list file.

6. Move the Airgap response and the downloaded files to the BigFix server.

7. Run the Airgap tool on the BigFix server to load the Airgap response into the BigFix server, and copy the downloaded files to the cache folder of the BigFix server.
Extraction usage overview

In this mode, the Airgap tool extracts information from the BigFix server. You run the Airgap tool starting from the BigFix server by performing the following steps:

1. Run the Airgap tool on the BigFix server to create the Airgap request.
2. Move the Airgap request to the internet facing computer.
3. Run the Airgap tool on the internet facing computer to gather the license information and the site contents into the Airgap response.
4. Move the Airgap response to the BigFix server.
5. Run the Airgap tool on the BigFix server to load the Airgap response into the BigFix server.
In this mode, Airgap gathers the contents of the site, but not the files. To download the files referenced by the Fixlets, such as the patch modules, run the BES Download Cacher utility by performing the following steps:

1. Locate the site masthead files for the sites you want to download files for, and copy the site masthead files to the computer with internet access.
2. On the internet facing computer, run the BES Download Cacher utility for each site masthead file to download files referenced from the site that the site masthead file represents.
3. Move the downloaded files to the cache folder of the BigFix server.

Requirements

When your BigFix server is installed in an air-gapped environment where a secure network is physically isolated from insecure networks, such as the public Internet or an insecure local area network, and the computers on opposite sides of the air gap cannot communicate, you need a workstation that has access to the public Internet to download Fixlet site contents using the Airgap tool, and to download files referenced in the Fixlet action scripts. This workstation cannot be a BigFix server or a BigFix relay.
The Airgap tool is platform dependent, but the `AirgapRequest.xml` (for extraction usage only) and `AirgapResponse` files are not. For the workstation that has access to the public Internet, you can use different operating systems available for the BigFix server.

Depending on sites gathered, the `AirgapResponse` file can be larger than 4GB. Your workstation must have enough free disk space to save the Airgap tool, the `AirgapResponse` file, and the files to download.

To run the Airgap tool on Windows computers, you must have the following libraries and files installed:

```plaintext
BESAirgapTool.exe
libBEScrypto.dll
libBEScryptoFIPS.dll
msvc90.dll
msvcp90.dll
msvcr90.dll
Microsoft.VC90.CRT.manifest
cabundle.crt
```

You can get all the above files by downloading a compressed file (Airgap Tool) from the Utilities page.

To run the Airgap tool on Linux computers, you must have the following files installed:

```plaintext
Airgap
Airgap.sh
libBEScrypto.so
libBEScryptoFIPS.so
cabundle.crt
```

If DB2 is not installed on the Linux computer that has access to the public Internet, to run the Airgap tool you must have installed the HCL Data Server Client or HCL Data Server Runtime Client using the `db2setup` command. The DB2 instance must be created with user `db2inst1`. 
Using the Airgap tool

Non-extraction usage

The "Non-extraction usage" mode is available only starting from BigFix Version 9.5.5.

The Airgap command line interface can gather site information without having to access the BigFix server and can optionally download files without passing through a download cacher.

With the non-extraction usage, the Airgap tool can download the files specified in Fixlets from download sites like Windows that do not require to authenticate. When you need to download files from sites that require to authenticate with an userid and password, or to download files not specified by prefetch or download commands in Fixlets, as in the case of patch modules for AIX, CentOS, HP-UX, RedHat, Solaris or SUSE, you must use a download cacher.

As a prerequisite for the following procedure, ensure that you have the files required for the Airgap tool to run.

**On Windows**

You can download the required files from the Airgap Tool page.

**On Linux**

Access the BigFix server computer, open the /opt/BESServer/bin folder and run this command:

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

Where `directory` is a folder of your choice.

Move to the directory containing the output generated by the above command, locate the file named `airgap.tar` and decompress it. Delete the `AirgapRequest.xml` file from the directory, copy all the other files portable drive.

To gather site information without accessing the BigFix server, complete the following steps:
1. Create a site list

Run the tool on a workstation that has access to the public Internet specifying the license serial number, the email address used to register your license, and the name of the file in which the tool lists the sites for your license. You must have writing access for the folder where the Airgap tool is located. Enter the following command:

**On Windows operating systems:**

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

**On Linux operating systems:**

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

Where:

- **mail_address**
  
  Is the mail address that you specified in your license; if it does not match, the Airgap tool fails. Option `-email` can be used only together with option `-createSiteList`.

- **proxy**
  
  Option used when the workstation that has access to the public Internet can connect only by a proxy server. In this case,
after the -proxy option, specify the hostname and port of the proxy server in the form hostname:port. If the proxy is an authenticating proxy, add also the userid and password in the form userid:password@hostname:port.

-usehttps
When this option is specified, "https" is used to contact the license server. Use option -cacert to specify a path in which to put the file ca-bundle.crt if you want to use a different folder from that in which the Airgap tool runs. The file ca-bundle.crt is used to validate the server certificate when you use the -usehttps option, or when the URL in the Fixlet begins with "https".

-cacert
This option can only be used together with option -usehttps.

-othersites
Use this option if your license is entitled to AllowOtherSites, to include sites of your choice to your site list. Create a folder, copy in it all the masthead files (*.efxm files) related to your mastheads not included in your license, and specify the name of this folder with option -othersites when you create a site list.

After running the tool, a file is created with the name that you specified as site_list_filename.

Note: The site list file, once created, can be used until you change the license, or HCL adds a new site to the existing license. If you delete the site list file for any reason, you can create it again with the same command, as the history of downloaded files is maintained as long as the license serial number does not change.

2. Edit the site list file
Each line of the file created in step 1 contains three pieces of information separated by a double colon:

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

You can edit only the `flag` parameter, that can have one the following values:

A

Site contents are gathered when a newer site version is available and stored in the `AirgapResponse` file, and used for downloading files or creating a file list.

R

Site contents are always gathered and stored in the `AirgapResponse` file regardless of the version of the site, and used for downloading files.

G

Site contents are gathered when a newer site version is available and stored in the `AirgapResponse` file, but not used for downloading files or creating a file list.

Q

Site contents are always gathered and stored in the `AirgapResponse` file regardless of the version of the site, but not used for downloading files or creating a file list.

D

Site contents are not gathered, but are used for downloading files or creating a file list. This flag is useful when you want to keep the current contents of a site without updating it and download files to run Fixlets at your current site. This option is valid only when the site contents have already been gathered.

N
Site is ignored, but site information is kept in the file for future reference.

**Note:** When you create a site list file, the default values for the BES Support and Web UI Common components are set to G. If you are not interested in the Web UI component, modify the default Web UI Common value from G to N. The default values for the other components are set to N. At the first run after installing the BigFix server, the license information, the BES Support and the Web UI Common components must be gathered. Only after moving this first Airgap response generated on the workstation that has access to the public Internet to the BigFix server, you can enable the other components that you can access from the License Overview dashboard of the console and continue with the process. Be sure to enable the required components other than default before gathering.

3. Gather site contents and create the Airgap response file

After you have edited the flags in the site list file, run the Airgap tool again to complete one of the following site operations:

**a. Gather site contents**

To gather site contents for sites with flag A or R or G or Q, run the following command:

**On Windows operating systems:**

```
BESAirgapTool.exe -site site_list_filename
```

**On Linux operating systems:**

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

On completion, you have created the Airgapresponse file.

**b. Gather site contents and download files**
To gather site contents for sites with flag A or R or G or Q, and download files referenced by Fixlets on sites with flag A or R or D, run the following command:

**On Windows operating systems:**

```
BESAirgapTool.exe -site site_list_filename -download
[-cache cache_name]
```

**On Linux operating systems:**

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

where `cache_name` is the folder path where to store the downloaded files. On completion, you have created the `Airgapresponse` file and downloaded the files to the `cache_name` folder.

c. **Gather site contents and download files selectively**

To gather site contents for sites with flag A or R or G or Q, and create a list of files referenced by Fixlets on sites with flag A or R or D, run the following command:

**On Windows operating systems:**

```
BESAirgapTool.exe -site site_list_filename
-createFileList referenced_list
```

**On Linux operating systems:**

```
./Airgap.sh -site site_list_filename
-createFileList referenced_list
```
On completion, you have created the Airgapresponse file and the file list with the name specified in referenced_list.

In all cases, site contents gathered for sites with flag A or R or G or Q are put in the AirgapResponse file. When you run the Airgap tool for the first time, all sites with flag A or R or G or Q are gathered. For subsequent times, the contents of sites with flag A or G are gathered only if either they have not been previously gathered or a newer site version is available. For sites with flag R or Q, contents are always gathered.

Optionally, you can also specify the following options:

-usehttps

License information and site contents are gathered using "https". For case "b. Gather site contents and download files", all urls beginning with "http" are forced to use "https". Note that some urls in Fixlets begin with "https" and some patch sites might redirect requests to urls beginning with "https".

-proxy [user:password@]hostname:port

Used when the workstation that has access to the public Internet can connect only through a proxy server. In this case, after the -proxy option, specify the host name and port of the proxy server in the format hostname:port. If the proxy is an authenticating proxy, add also the user ID and password in the format userid:password@hostname:port.

-cacert crt_filename

To specify a path in which to put the file ca-bundle.crt if you want to use a different folder from that in which the Airgap tool runs. The file ca-bundle.crt is used to validate the server certificate when you use the -usehttps option, or when the url in the Fixlet begins with "https". The option -cacert can only be used together with option -usehttps.

-timeout timeout_seconds
This option is available starting from V9.5.7. It specifies a http timeout interval in seconds. Values range from 30 to 3600. The default value is 30. In the event you get the error "HTTP Error 28: Timeout was reached" while using a proxy, try also to use option -usehttps as it makes proxy to work in tunneling mode and that might help avoiding timeouts.

For cases b and c, you can also use other options to reduce the number of files to download or to gather in the file list. These filtering options select Fixlets that refer to files, not the files themselves. For example, when you specify last 5 days, it means files referenced by Fixlets modified in the last 5 days, not files added or changed by vendors in the last 5 days. To create a list of possible values for filtering options, run the following command:

**On Windows operating systems:**

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

**On Linux operating systems:**

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

The list of available values is limited to the following options: -fcategory, -fcve, -fproduct, -fseverity, -fsource, and -fsourceid. The following options are available for filtering:

- **-fcategory**
  Fixlet category property.

- **-fcve**
  To specify the CVE (Common Vulnerabilities and Exposures) id associated with a security patch.
--fdays
To select Fixlets whose last modified date falls within a specified number of days from the date you run the command.

--fproduct
To specify the product name to which the Fixlet is applicable, such as Win2008 or Win7. This information is not shown in the Console. This option is available only for sites related to patches for Windows operating systems.

--fseverity
To specify the severity that a vendor associates with a security patch.

--fsource
Provider of file, such as BigFix, Adobe, or Microsoft.

--fsourceid
Identification specified by the provider.

--includeCorrupt
To include Fixlets marked as Corrupted, that are excluded by default when this option is not specified.

--includeSuperseded
To include Fixlets marked as Superseded, that are excluded by default when this option is not specified.

When multiple filter conditions are specified, only Fixlets that satisfy all conditions are selected. For options --fsource, --fsourceid, --fcve, --fcategory, and --fseverity, you can specify multiple comma-separated values, for example: --fseverity "Critical, Important". When you use commas to separate values, or values contain spaces, enclose parameters in double quotes, as in the previous example. Note that values are case sensitive.
4. Edit the file list

Applicable only to case c. **Gather site contents and download files selectively** of step 3.

With `-createFileList` option, you create a file that contains a list of files. Each line of the list contains pieces of information separated by a double colon:

```plaintext
flag::site_name::Fixlet_id::site_url::
size::hash_value::hash_algorithm
```

For example:

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

You can edit only the `flag` value, changing it to `Y` to download the file, or to `N` to not download the file.

5. Run the tool on the Internet facing workstation to download files

Applicable only to case c. **Gather site contents and download files selectively** of step 3.

After editing the file list in step 4, to download only the files with flag `Y` in the file list, run the Airgap tool by issuing the following command:

**On Windows operating systems:**

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

**On Linux operating systems:**

```bash
./Airgap.sh -file file_list_filename -download -cache cache_foldername
```
[-proxy [user:password@]hostname:port] [-usehttps]
[-cacert crt_filename]

where cache_foldername is the folder path where to store the downloaded files. The files already in the cache folder are not downloaded again.

6. Move the Airgap response file to the BigFix server and run the Airgap tool on the BigFix server

Copy in a portable drive the AirgapResponse file, and the file list that you have created in step 3 or the downloaded files that you collected in step 5, and transfer them to the BigFix server computer. Make sure that the AirgapResponse file is in the same folder as the Airgap tool, and run it by issuing the following command:

**On Windows operating systems:**

BESAirgapTool.exe

**On Linux operating systems:**

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

This imports the response file with the Fixlet content and license updates into your deployment.

⚠️ **Note:** The Airgap tool passes site contents in the response file to the GatherDB component of your BigFix server, and the GatherDB component imports site contents. For sites other than WebUI sites, you can monitor the import progress in the DebugOut of the GatherDB component (default name GatherDB.log).

Copy the downloaded files also into the BigFix server cache folder. The cache folder default location is:

**On Windows operating systems:**

%PROGRAM FILES%\BigFix Enterprise\BES Server\wwwrootbes\bfmirror\downloads\sha1
On Linux operating systems:

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

Repeat these steps periodically to keep updated the Fixlet content in the main BigFix server. Join the new Fixlet mailing list to receive notifications when Fixlets are updated. Always make sure that the Airgap tool version is compatible with the version of the BigFix server installed.

Usage tips:

1. Unzip the exact same version of the AirgapTool used in Step 1 into a directory on the BigFix root server.
2. Copy the airgapresponsefile into this same directory.
3. Run BESAirgapTool.exe with no options.
   The contents of the airgapresponsefile is imported in to the directory. If you downloaded any files at Step 5, then copy those files in to the SHA1 directory on the root server as well. This might be necessary because the Airgap tool downloads files and names them with their SHA256 values.

Note: You do not need to rename the SHA256 value as its SHA1 value after pasting it to the SHA1 directory.

Optional actions:

Check if all required files have been downloaded

To check if you have downloaded all the files required for the Fixlet you are planning to apply, use option -checkfixlet when you run the Airgap tool. For example:

On Windows operating systems:

BESAirgapTool.exe -site site_list.txt -checkfixlet
-fdays 100 -fseverity Critical -cache
MyCache
On Linux operating systems:

```bash
./Airgap.sh -site site_list.txt -
checkfixlet
-fdays 100 -fseverity Critical -cache
MyCache
```

For Fixlets satisfying the specified filtering conditions, the tool checks the downloaded history and contents of destination folder, and if there are still files to download, Fixlet names and urls are displayed.

**Files to be downloaded manually**

Some files referenced by Fixlets might not be downloaded because they can be obtained only by contacting the vendor support center, or because the download site requires that you explicitly accept the license terms and this action cannot be automated for legal reasons. In these cases, the involved files have the download url containing the string `MANUAL_BES_CACHING_REQUIRED` and must be downloaded manually. To create a list of these files, use option `-createmanuallist` as in the following example:

**On Windows operating systems:**

```bash
BESAirgapTool.exe -site site_list.txt -
createmanuallist
manual_list -fseverity Critical
```

**On Linux operating systems:**

```bash
./Airgap.sh -site site_list.txt -
createmanuallist
manual_list -fseverity Critical
```
You can also use the `--checkmanual` option to check if your destination folder contains all the files that must be manually downloaded, as in the following example:

**On Windows operating systems:**

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

**On Linux operating systems:**

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

**Reset history**

The Airgap tool keeps a history of downloaded files. Even if you move all the downloaded files from your public Internet facing workstation to the BigFix server, this history is maintained and files previously downloaded are not downloaded again to save time and disk space. If you deleted part or all of your previously downloaded files and you need them again, you can use the `--resync` option. This option clears the download history and checks the files in the folder specified with `--cache` option. Note that the newly-created download history is based only on the files contained in the folder specified with the `--cache` option.

**Changing license**

If you want to manage another license, you must erase the history of gathered sites and downloaded files. To complete this action, use the `--force` option as in the following example:

**On Windows operating systems:**

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

On Linux operating systems:

./Airgap.sh -serial serial_number -email mail_address -
createSiteList site_list_filename -force

Miscellaneous options

By default, the Airgap tool simultaneously downloads two files. You can change the number of files to download concurrently by specifying a number after the -download option. This number can range from 1 to 8. For example, to download 3 files at the same time, specify -download 3. Note that you need a larger bandwidth when downloading more than 2 files simultaneously.

When the url specified in a Fixlet begins with "https", or if you specify the -useHttps option, the Airgap tool tries to verify that the server specified in the url has an appropriate SSL Server Certificate. If, for any reason, you want to skip this check and avoid a download failure when the Airgap tool cannot verify the server certificate, use the -noverify option. With this option, the Airgap tool does not validate any server certificate and you must check that your workstation translates correctly host names by checking your DNS.

To have the Airgap tool to print more information than usual, use the -verbose option.

Working with multiple BigFix servers
If you want to use the same public Internet facing workstation for several BigFix servers, like a test server and a production server, create a folder for each server, copy the Airgap tool in each folder, and work with each folder separately. You can share the same site list among the different folders, but each server keeps its own history in its folder. When using multiple Airgap tools with different servers, you can also share a cache folder to download only once files that are common to different servers, but you must ensure to run only one instance of the Airgap tool at the same time.

In case you need to gather set of sites, load them to your test server, then perform tests with the gathered sites and load the tested sites, not the latest ones, to your production server, you can load one AirgapResponse file to multiple BigFix servers when they are licensed for the same products (like BigFix Lifecycle, BigFix Compliance, etc.). When you intend to load one AirgapResponse file to multiple BigFix servers, it is recommended to gather only sites enabled on all of your BigFix servers.

**Note:** At the first run after installing the BigFix server, the license information, the BES Support, and the Web UI Common components must be gathered for each installation. For this step, an AirgapResponse file must be created for each BigFix server because license information is unique to each serial number.

If you want to update the license information of a particular BigFix server without changing version on any site, you can create an AirgapResponse file that contains only license information by running the Airgap tool with a site file containing no lines or with site files where all sites have the flag \textbf{N}. Run the following command:
On Windows operating systems:

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

On Linux operating systems:

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

Enabling WebUI in air-gapped environments

To install the WebUI in air-gapped environments, perform the following steps:

1. Gather the latest BES Support and WebUI Common sites, and download the required files to install the WebUI Service. Load them to your BigFix server.
2. Install the WebUI Service by using the task "Install HCL BigFix WebUI Service" in BES Support site.
3. After the installation completes, wait for the activation of a WebUI Service (on Windows operating systems) or process (on Linux operating systems) on the WebUI targeting system. The WebUI initialization has started; wait for its completion. Initialization usually completes in few minutes, but it is suggested to wait 30 minutes or more before proceeding with step 4.
4. Gather all the latest WebUI sites and load them to your BigFix server. You can gather WebUI sites before running the task to install the WebUI service, but you can load them only after the WebUI initialization has completed.
**Important:** If you have a BigFix 9.5.7 fresh installation, to make the WebUI sites available, you must complete the following steps:

1. Install the WebUI and run the Airgap tool
2. Wait a few minutes for the WebUI initialization to complete
3. Rerun the Airgap tool.

To make Fixlet content and product license updates available in the isolated network, the utility must be transferred from a computer with internet connectivity using the following steps:

**On Windows operating systems**

1. **Run on the BigFix server**
   
   From the BigFix server installation directory, double-click `BESAirgapTool.exe` or run it from the command line without any parameters, a Graphical User Interface opens.

   Provide a destination folder for the Airgap tool to store its site request and all the files it requires to run. After the Airgap tool finishes copying the files, copy the entire folder to a portable drive.

2. **Move the Airgap request and run on the internet facing computer**

   Bring the portable drive to a computer with Internet connectivity. You must have the rights to write in the folder where the `BESAirgapTool.exe` is located. Enter the folder and run the Airgap tool by double-clicking `BESAirgapTool.exe` or invoking it from the command line.

   Optionally, you can also specify the following command line parameters:

   `-usehttps`
   
   All urls beginning with "http" are forced to use "https" to gather license information and site contents. Note that some urls in Fixlets begin with "https" and some patch sites might redirect requests to urls beginning with "https".

   `-proxy [user:password@]hostname:port`
This option is available only starting from BigFix Version 9.5.5. Used when the workstation that has access to the public Internet can connect only through a proxy server. In this case, after the -proxy option, specify the host name and the port of the proxy server in the format hostname:port. If the proxy is an authenticating proxy, add also the user ID and the password in the format userid:password@hostname:port. In extraction usage, when a proxy server is configured in the client registry settings or in the Internet Explorer settings for the current user and the -proxy option is not specified, the proxy settings are used as in earlier versions of the Airgap tool. When you use the -proxy option, the specified values are used regardless of other settings.

-cacert <full_path_to_ca-bundle.crt_file>

To specify a path in which to store the file ca-bundle.crt, if you want to use a different folder from that where the Airgap tool runs. The file ca-bundle.crt is used to validate the server certificate when you use the -usehttps option, or when the URL in the Fixlet begins with "https". The option -cacert can only be used together with the -usehttps option.

A Graphical User Interface opens. The Airgap tool will download all files required by the Airgap request in the same folder as BESAirgapTool.exe. This exchanges the Airgap request file for an Airgap response file. Copy the Airgap response file to a portable drive.

3. Move the Airgap response to the BigFix server and run the Airgap tool on the BigFix server

Take the portable drive back to the BigFix server computer and run the BESAirgapTool.exe again by double-clicking BESAirgapTool.exe or invoking it from the command line without any parameters. Ensure that you are running it logged on as a user that:

- Has Administrator privileges.
- Has the database permissions necessary to add content to the BFEnterprise database.

A Graphical User Interface opens.
This imports the Airgap response file with the Fixlet content and license updates into your deployment.

The Airgap tool creates temporary files in the folder specified by the TEMP environment variable. If you want to use a different folder for temporary files, set the TEMP environment variable to that folder before you run the BESAirgapTool.exe.

To update the Fixlet content on the main BigFix server, repeat these steps periodically. You can join the new Fixlet mailing list to receive notifications when Fixlets are updated.

Ensure that the Airgap tool version is compatible with the installed BigFix server version.

**On Linux operating systems**

1. **Run on the BigFix server**
   
   Ensure that on the Linux computer, the Airgap tool is located in the same path where you installed the BigFix server. The default path is /opt/BESServer/bin. Open the Linux Terminal, and enter the following commands to create a tar file named airgap.tar, containing the AirgapRequest.xml file based on the BigFix database information:

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

   Where:

   **-remotedir directory**

   Runs Airgap to generate the request file in the specified folder.

2. **Move the Airgap request and run on the internet facing computer**

   Copy the airgap.tar file to a portable drive, and extract the airgap.tar file content by issuing the following command:

   ```
   # tar -xf airgap.tar
   ```
Ensure that your system has an environment variable named `LD_LIBRARY_PATH` set to the path of the folder containing the DB2 library `libdb2.so.1`. Ensure that the `Airgap.sh` and `AirgapRequest.xml` files are in the same folder and that you have writing rights to that folder. Run the `Airgap.sh` command.

Optionally, you can also specify the following command line parameters:

- **-usehttps**

  All urls beginning with "http" are forced to use "https" to gather license information and site contents. Note that some urls in Fixlets begin with "https" and some patch sites might redirect requests to urls beginning with "https".

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

  Used when the workstation that has access to the public Internet can connect only through a proxy server. In this case, after the `-proxy` option, specify the host name and the port of the proxy server in the format `hostname:port`. If the proxy is an authenticating proxy, add also the user ID and the password in the format `userid:password@hostname:port`.

- **-cacert <full_path_to_ca-bundle.crt_file>**

  To specify a path in which to store the file `ca-bundle.crt`, if you want to use a different folder from that where the Airgap tool runs. The file `ca-bundle.crt` is used to validate the server certificate when you use the `-usehttps` option, or when the URL in the Fixlet begins with "https". The option `-cacert` can only be used together with the `-usehttps` option.

This exchanges the Airgap request file for an Airgap response file. Copy the Airgap response file to a portable drive.

If you receive the following error message when running the Airgap tool:

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

Create and export the `LD_LIBRARY_PATH` variable by running the command:
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/your/path/"

Where:

/your/path

Is the path of the folder containing the DB2 library `libdb2.so.1`

3. **Move the Airgap response to the BigFix server and run the Airgap tool on the BigFix server**

Connect the portable drive back to the BigFix server computer and run the `Airgap.sh` command. This imports the response file with Fixlet content and license updates into your deployment.

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

Optionally, you can also specify the following option:

- **temp directory**

  The Airgap tool creates temporary files under the `/tmp` directory, but in the event you do not have enough space left in it, you can use this option to specify a different folder where you have enough space.

  Note that the `Airgap.sh` and `AirgapRequest.xml` files must be in the same folder.

  To update the Fixlet content on the main BigFix server, repeat these steps periodically. You can join the new Fixlet mailing list to receive notifications when Fixlets are updated.

  Ensure that the Airgap tool version is compatible with the installed BigFix version.

**Transferring downloaded files**

Deploying Fixlets on the main BigFix server requires downloaded patches and other files from the Internet. You can use the Airgap tool in extraction usage for gathering site contents and in non-extraction usage for downloading files (you can ignore the AirgapResponse file generated in non-extraction usage). As an alternative, you can use the BES Download Cacher utility. This utility helps to:
• Download and transfer files to the main BigFix server.
• Download patch contents in a Fixlet site or single file downloads from an URL.

You can download the current utility from http://software.bigfix.com/download/bes/util/BESDownloadCacher.exe. To see the list of available options run BESDownloadCacher.exe /?. If the BigFix server or an BigFix relay is installed on the system where you run the BES Download Cacher utility, the -x utility parameter is optional because the utility detects relevant local BES settings and reuse them as defaults.

Some sites require additional steps to download content from patch vendors that restrict access. For additional information see the following Knowledge documents that describe using a tool to manually download patches for Solaris, Red Hat Enterprise Linux, SuSE Linux Enterprise, and AIX.

These sites require a three step process:

1. Run the BESAirgapTool.exe to download Fixlets and Tasks for each site.
2. Run the BES Download Cacher utility to download any site tools from BigFix.
3. Run the download tool for each vendor to download patch contents.

**Transferring all files from Fixlet sites**

To transfer files from Fixlet sites, perform the following steps:

1. Locate the .efxm file for the site from which you want to gather downloads, for example, BES Asset Discovery.efxm.
2. Run the BES Download Cacher utility with the following command:

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

   **Note:** This might take a very long time because it downloads every file referenced in the Fixlet site and puts the files into the downloads folder. If the files already exist in the downloads folder, they are not re-downloaded. Files are named with their sha1 checksum.
3. When the download finishes, copy the contents of the downloads folder (just the files, not the folder) into the sha1 folder on the main BigFix server. The default location for the sha1 folder is:

- **On Windows systems:** %PROGRAM FILES%\BigFix Enterprise\BES Server wwwrootbes\bfmirror\downloads\sha1
- **On Linux systems:** /var/opt/BESServer/wwwrootbes/bfmirror/downloads/sha1

The BigFix server uses these files instead of trying to download them from the Internet.

**Note:** If you run the BES Download Cacher utility later, you can look at the modification time of the files to see which files are the newest. Using this method, you transfer only the newest files to the Main BigFix server instead of copying every file each time.

You might need to increase the size of the cache on the main BigFix server so that it does not try to delete any files from the cache. Run the BES Download Cacher utility to increase the size of the cache with the following command:

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

The default size of the cache is 1024 MB.

**Note:** Use the `-c` option only when the BigFix server or a relay is installed on the system where you run the BES Download Cacher utility. If no BigFix component is installed, cache has no limit.

After the files are cached in the BigFix server sha1 folder, they are automatically delivered to the BigFix relays and BigFix clients when you click an action in the Fixlet message that references a downloaded file. If the file is not cached, the BigFix console gives you a status of *Waiting for Mirror Server* after you deploy an action. For additional information about how the BigFix cache works, see [How does the TEM Server and TEM Relay cache work](#).

**Transferring a single file**

To transfer a single file from a Fixlet site, perform the following steps:

1. Run the BES Download Cacher utility with the following command:
BESDownloadCacher.exe -u http://www.mysite/downloads/myplugin.exe -x downloads

2. When the download finishes, copy the contents of the downloads folder (just the file, not the folder) into the sha1 folder on the main BigFix server. The default location for the sha1 folder is:

- **On Windows systems:** %PROGRAM FILES%\BigFix Enterprise\BES Server\wwwrootbes\bfmirror\downloads\sha1
- **On Linux systems:** /var/opt/BESServer/wwwrootbes/bfmirror/downloads/sha1

You might need to increase the size of the cache on the main BigFix server so that it does not try to delete any files from the cache. Run the BES Download Cacher utility to increase the size of the cache with the following command:

BESDownloadCacher.exe -c 1024

The default size of the cache is 1024 MB.

**Note:** Use the `-c` option only when the BigFix server or a relay is installed on the system where you run the BES Download Cacher utility. If no BigFix component is installed, cache has no limit.

After the files are cached in the BigFix server sha1 folder, they are automatically delivered to the BigFix relays and BigFix clients when you click an action in the Fixlet message that references a downloaded file. If the file is not cached, the BigFix console gives you a status of "Waiting for Mirror Server" after you deploy an action. For additional information about how the BigFix cache works see [How does the TEM Server and TEM Relay cache work?](#).

**Log files**

The Airgap tool produces two types of log files: normal log files and debug log files.

Normal log files record the messages you see in the command window so you can check Airgap tasks, such as sites gathered on a specific date. Debug log files are intended for the HCL Support team. The naming convention for normal log files is:
On Windows operating systems:

BESAirgapTool_yyyy-mm-dd.log

On Linux operating systems:

Airgap_yyyy-mm-dd.log

where \textit{yyyy-mm-dd} is the date when the file has been created. Starting from V9.5.7, files older than 30 days are deleted.

The debug log file is \textit{AirgapDebugOut.txt}. Starting from V9.5.7, this file contains only information of the last day and older log files are renamed to \textit{AirgapDebugOutyyyyymmdd.txt}, where \textit{yyyyymmdd} is the date when the file has been created; files older than 10 days are deleted. The Airgap tool can write more information to the debug log file by using the verbose option \textit{-verbose}. 
Chapter 10. Getting client information by using BigFix Query

The BigFix Query feature allows you to retrieve information and run relevance queries on client workstations from the WebUI BigFix Query Application or by using REST APIs.

Use the BigFix Query feature to:

- Quickly collect data from clients without impacting BigFix environment performance.
- Run your query in relevance language on targets identified using an applicability relevance or on a set of target agent IDs.
- Show the collected results in the WebUI Query Application, optionally paging them. The results displayed are updated periodically as new values are received from clients.
- Test relevance expressions on a few selected clients before rolling out to production.

This guide contains the information about how to configure BigFix for using BigFix Query. Additional information is available clicking the following links:

- BigFix Query section of the WebUI User's Guide
- Query (on page ) in List of settings and detailed descriptions (on page )

BigFix Query requirements

The clients that are targeted by BigFix Query requests must satisfy specific conditions.

The following requirements must be satisfied to run BigFix Query on clients:

- The client can receive UDP notifications. The BigFix Query feature does not support components that are connected to the BigFix server through proxies or firewalls.
- BigFix V9.5 Patch 2 or later must be installed on the client machine and on all the intermediate relays that must be passed through to reach the client.
BigFix Query restrictions

Some restrictions apply when using the BigFix Query feature.

The following limitations affect the use of the BigFix Query feature:

- The feature is available only for BigFix Lifecycle or BigFix Compliance Version 9.5 Patch 2 or later versions.
- The feature does not support requests requiring the agent context.
- If you configured your environment in a Disaster Server Architecture (DSA), be aware that:
  - The information about BigFix Query is not replicated among the multiple servers.
  - Each server can run BigFix Query requests only on the clients that connect either directly or through relays to the server where the query is submitted.

Who can use BigFix Query

BigFix Query requests can be run by Master Operators and Non-Master Operators. Specific permissions must be set to allow operators to use this feature.

To access the WebUI Query application from the WebUI toolbar:

The user must have, at operator or role level, the effective permission on the query WebUI Application set to Allowed, for example:

As an alternative, you can see which permissions are assigned to the users on the WebUI Applications in the working area of the WebUi Apps Domain.
The **WebUI Apps** Domain is available under **All Contents** after you enable the WebUI.

For more information about how to access the WebUI Query Application, see [How to run BigFix Query from the WebUI (on page 131)](#).

**To run BigFix Query requests and see their results:**

Master Operators can run queries by default. A Non-Master Operator must have, at operator or role level, the **Can Submit Queries** permission set to **Yes** in the **Details** tab:

![WebUI Query Application](image)

The default value of the **Can Submit Queries** permission for Non-Master Operators is **No**.

For more information about operator permissions and roles, see [Adding Local Operators (on page 25)](#).
How to run BigFix Query from the WebUI

You can access the BigFix Query on the WebUI user interface by selecting **Content -> Query**.

The Query panel opens:

For information about using this feature from the Query panel, see [WebUI Enablement](#).

How BigFix manages BigFix Query requests

A BigFix Query request is processed in a sequence of customizable steps.

The following picture shows the internal flow of a BigFix Query. Each step lists which variables you can configure to tune how BigFix Query requests and responses are managed.
1. The operator that logged on to the WebUI submits a request from the BigFix Query Application.

**What can you customize for this step?**

You can decide to run this step as an operator that is not a master operator. In this case ensure that either the operator permissions or the
permissions specified in the role assigned to the operator contain the **Can Submit Queries** value set to **Yes**.

**Note:** If you are using the REST API to manage the query, be aware that only the operator issuing the query can see its responses.

2. The submitted request is propagated through the relay hierarchy to the target clients using dedicated memory queues on each relay. This ensures that the request quickly reaches the targets without impacting normal BigFix processing. If a target or a child relay does not answer within a given amount of time, then it is no longer requested to answer.

**What can you customize for this step?**

From the BigFix Console you can customize, for the server and for each relay, how the memory queues are cleaned up:

**How often the cleanup task is run.**

The default value is 10 minutes and the name of the setting is `_BESRelay_Query_RemovalTask_`.

**How long a request can stay in the queue before being deleted by the cleanup task.**

The default value is 60 minutes and the name of the setting is `_BESRelay_Query_MinTime_`.

**The maximum size of the memory queue dedicated to BigFix Query requests.**

Before running the cleanup task, BigFix checks if the size of this memory queue exceeds the maximum size specified. If it exceeds, when the cleanup task runs, it removes the entries in the queue until the size of the queue returns within the threshold. The default value is 100 MB and the name of the setting is `_BESRelay_Query_MemoryLimit_`.

For more information about these settings, see Query (on page ).
3. When the request reaches the target client parent relay, the relay informs the client, using the UDP protocol, that there is a new request to process, and, in turn, the agent retrieves the request.

4. For each responsive target, the client passes the query to the local QnA to run the query and return the results.

What can you customize for this step?

From the BigFix Console you can customize for the client:

**How long can the QnA process a query issued by a Master Operator before the request timeout elapses**

The default value is 60 seconds and the name of the setting is _BESClient_Query_MOMaxQueryTime_.

**How long can the QnA process a query issued by a Non Master Operator before the request timeout elapses**

The default value is 10 seconds and the name of the setting is _BESClient_Query_NMOMaxQueryTime_.

**How long the QnA waits for new queries to process before stopping.**

The default value is 600 seconds and the name of the setting is _BESClient_Query_IdleTimeout_.

**How much CPU is used by the QnA process running the query.**

You can limit the CPU used by the QnA process by defining time slots during which the QnA runs. By default the QnA processing the query runs for 10 milliseconds and then sleeps for 480 milliseconds, which corresponds to a CPU usage lower than 1-2 %, and the name of the settings that define this behavior are _BESClient_Query_WorkTime_ and _BESClient_Query_SleepTime_.

For more information about these settings, see Query (on page ).
**Note:** These settings are not considered when running the QnA tool connected as local user to the client system.

5. When the agent receives the response from the QnA, it creates a report containing the response and delivers it to the parent relay in parallel the other reports.

6. The report is delivered back to the server through the relay hierarchy. On each relay the report is stored in a memory queue while waiting to be delivered to the parent relay. If the parent relay is not available, the report waits in the queue and is delivered as soon as the parent relay becomes available again. The same criteria for encryption and signing used for normal reports are applied also to these reports.

**What can you customize for this step?**

From the BigFix Console you can customize for each relay:

- **The maximum size of the memory queue dedicated to BigFix Query results.**

  Before running the cleanup task, BigFix checks if the size of this memory queue exceeds the maximum size specified. If it exceeds, when the cleanup task runs, it removes the entries in the queue until the size of the queue returns within the threshold. The default value is 100 MB and the name of the setting is `_BESRelay_Query_ResultsMemoryLimit`.

  For more information about this setting, see Query (on page ).

7. When the server receives the result, it stores it in a dedicated queue from where a dedicated FillDB thread gets the data to store it in the database. In this way, the normal processing on the BigFix server is not impacted.

  The database stores, for a specified time period, both the BigFix Query request and its responses, that can be used, for example, to be filtered, displayed, or to create reports. On a timely basis the BigFix Query Application checks the database for updates and updates the displayed results accordingly.

**What can you customize for this step?**
From the BigFix Administrative Tool you can customize on the server:

**For how long the BigFix Query requests are stored in the database before being deleted.**

The default value is 1440 hours (60 days) and the name of the advanced option is `queryHoursToLive`.

**For how long the BigFix Query responses are stored in the database before being deleted.**

The default value is 4 hours and the name of the advanced option is `queryResultsHoursToLive`.

**How many requests and responses, for which `queryHoursToLive` or `queryResultsHoursToLive` elapsed, are deleted at a time from the database.**

The default value is 100000 entries and the name of the advanced option is `queryPurgeBatchSize`.

For more information about these advanced options, see Advanced Options (on page ).

For information about how to edit computer settings, see Edit Settings for Computer (on page ).
Chapter 11. **Persistent connections**

Starting from Patch 11, the capability to establish persistent connections was added to the product.

**Clients behind firewall or NAT**

Firewalls or NAT might prevent the BigFix Query function from working properly because the UDP notification, with which a parent relay delivers the query to the child clients, cannot usually reach the clients. Unlike other product functions, the BigFix Query cannot take advantage of client polling to overcome this restriction in the downstream communications.

This restriction is overcome by establishing a persistent TCP connection between the parent relay and at least one of its child clients. The persistent connection, which is always initiated by the client, is used by the relay to send UDP notifications to all clients in the same subnet of the persistently connected client (PCC).

**Overview**

The following picture displays the persistent TCP connection established between client and relay, and the UDP notifications sent from the PCC to other clients of the same subnet:
Enabling persistent connections on the relay

1. Log in as a master operator to the BigFix Console.
2. Locate and right-click the relay computer. Select Edit Computer Settings...
3. Add the following setting to the computer:

   _BESRelay_PersistentConnection_Enabled = 1

4. Restart the relay process for the setting to become effective.

   **Note:** When adding this setting to a relay computer having a Version prior to 9.5 Patch 11, the behavior of its child clients remains unchanged.

   **Note:** This setting is not effective on the BigFix server computer.

Enabling persistent connections on the client

1. Log in as a master operator to the BigFix Console.
2. Locate and right-click the client computer. Select Edit Computer Settings...
3. Add the following setting to the computer:

   _BESClient_PersistentConnection_Enabled = 1

Establishing a persistent connection

After being enabled, a persistent TCP connection between a client and its parent relay is normally established at the next registration of the client.

When the next registration occurs, the relay on which the client is registering checks whether the client is eligible to open a persistent connection, based on the overall number of persistent connections that the relay is already handling, and their partition by subnet. If the client is eligible, the relay notifies it accordingly. The client, then, waits for 60 seconds. If the client does not receive a test UDP notification from the relay within this time interval, it eventually opens the persistent connection.

If the client fails when establishing the persistent connection, it will retry opening the persistent connection after 3 minutes, up to a maximum of 4 attempts in total.
The persistent connection can generally be closed and then established again every time the client performs a new registration, provided that all prerequisites are still satisfied. The persistent connection might also terminate when either the client or the relay must handle restart and shutdown operations.

**Communicating on the persistent connection**

**Directly:**

If the relay must send a UDP notification to a persistently connected client (PCC), it uses the persistent connection to send it directly to the target client.

**Served by another client of the same subnet:**

If the relay must send a UDP notification to a client in a subnet served by a PCC, the relay sends the notification and the target client information (hostname/IP address stored during the registration phase) to the PCC. The PCC reads the notification and sends it through UDP to the target client. The target client processes the notification normally, and sends back a reply directly to the relay, as usually. If there is more than one PCC available, within the same subnet, that can serve the client, the relay sends the notification to one PCC only, not to all available PCCs.

**Managing persistent connections**

You can manage persistent connections by configuring a few settings. For details, see Persistent TCP connections (on page 139).
Chapter 12. **Relays in DMZ**

Starting from Patch 13, the capability to establish a persistent TCP connection between the parent relay in the more secure zone and its child relay inside the DMZ network was added to the product. This allows you to manage systems in a demilitarized zone (DMZ network).

In an environment where a relay in DMZ reports to a parent relay within its intranet network, it can be assumed that all communications between intranet and DMZ pass through a firewall that does not allow any upstream communication. In this case, any attempt for the child relay in the DMZ to initiate communication with its parent relay will fail.

This restriction is overcome by establishing a persistent TCP connection between the parent relay and its child relay inside the DMZ. The persistent connection is always initiated by the parent relay. The communication cannot be initiated by the child relay due to network restrictions.

**Overview**

The following picture displays the persistent TCP connection established between parent relay and child relay:

---

**Hypothesis:** the BigFix Parent Relay is outside the DMZ and the BigFix Child Relay is inside the DMZ network.
In the picture are displayed:

- In green: The persistent TCP connection established between the parent relay located in the more secure zone and the child relay located in the demilitarized zone.
- In yellow and black: The line of the demilitarized zone (DMZ network).

### Enabling persistent connections on both parent and child relay

**On a child relay where the BigFix client was not registered on the BigFix server yet**

1. Log in to the BigFix Console.
2. Run the **Relays in DMZ: Enable Parent Relay and set Child Relay List Fixlet** on the parent relay computer:

   ❗️ **Note:** Before running the Fixlet, you must specify in the text field of the Description tab the list of child relays allowed.

3. Manually install the BigFix client on the child computer. For more details, see Installing the client manually (on page 113).
4. Manually install the BigFix relay on the child computer by downloading the appropriate package depending on your operating system from the following web site: [http://support.bigfix.com/bes/release/](http://support.bigfix.com/bes/release/)

   ❗️ **Note:** In a typical scenario, run the Fixlet first on the parent relay and then manually configure the child relay.

5. On the child computer, ensure that the client and relay processes are stopped.
6. On a Windows child relay, add the `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\EnterpriseClient\Settings\Client\_BESRelay_DMZ_ChildEnable` key to the Windows registry and set its string REG_SZ value to 1.
7. On a Linux child relay, if the `besclient.config` file does not already exist, make a copy of the file named `besclient.config.default` located in the `/var/opt/BESClient/` directory and rename it into `besclient.config`. Manually edit the `besclient.config` by adding the following new section:
8. Restart first the relay process.
9. At least one minute after restarting the relay process, restart the client process.

**Note:** If your parent relay was configured as an authenticating relay, it might be necessary to temporarily disable the relay authentication to allow the child relay to register successfully for the first time. Enable again the relay authentication after your child relay was registered successfully.

**On a child relay where the BigFix client was already registered on the BigFix server**

1. Log in to the BigFix Console.
2. Run the **Relays in DMZ: Enable Parent Relay and set Child Relay List Fixlet** on the parent relay computer:

   **Note:** Before running the Fixlet, you must specify in the text field of the Description tab the list of child relays allowed.
3. Run the **Relays in DMZ: Enable Child Relay Fixlet** on the child relay computer:

   **Note:** In a typical scenario, run the Fixlet first on the parent relay and then on the child relay.
4. Both Fixlets will restart the relay process.

**Establishing a persistent connection**

The parent relay will try to open a socket to the child relay at port 52311.

The child relay can "grab" the socket used by the parent to communicate with it and keep it alive by sending ping messages periodically. At the same time, the child relay will start to listen on a different port such as 52312 only on its loopback address, this will be used to forward all the traffic through the socket opened by the parent that was previously grabbed.
All requests coming to the child relay that must be propagated upstream (for example during the registration of a client below the child relay or for reporting purposes) will be internally routed to the loopback address to be sent to the parent relay within the intranet.

**Communicating on the persistent connection**

To achieve the requirement, the parent relay initiates a communication with its own child relay and keeps the connection standing and persistent to, later on, use it from the child relay to the parent relay when upstream communication is needed by the child relay.

**Managing persistent connections**

You can manage the Relays in DMZ persistent connections by configuring a few settings. For details, see Relays in DMZ (on page 143) in BigFix Configuration Settings (on page 106).
Chapter 13. **Peer to peer mode**

Starting from Patch 11, the BigFix client includes a new feature named PeerNest, that allows to share binary files among clients located in the same subnet.

While PeerNest is applicable in many use cases, we can consider as an example the use case of a branch office connected to the data center through a slow link. With earlier BigFix versions, the suggested configuration required a relay to be installed in the branch office in order to download and cache large payloads. With PeerNest, the BigFix clients can share downloaded binaries and therefore reduce the number of communications going outside of the branch office even if a relay is not installed locally.

When PeerNest is enabled, the BigFix clients are able to self optimize the download of the binaries required to execute actions. When multiple clients are executing actions requiring the prefetch of a binary file, they check with their peers if the file is already cached in the subnet. If the binary was not cached yet, the clients can elect one of them as responsible for downloading from the relay and then sharing with the peers. In this way, multiple clients generate on the relay the download load of a single client, because only one clients downloads from the relay and then shares the download with the peers.

![Note:](note.png)  The BigFix clients are able to self optimize the download of the binaries required to execute actions only when the hash of the file is specified inside the `prefetch` statement.

**Requirements to use this mode**

PeerNest requires UDP communication to be enabled, in order to allow the BigFix clients to communicate with each other. If you are not allowing UDP communications and/or you are not opening the 52311 UDP port on your clients, you will not be able to leverage this new feature.

PeerNest also requires that the TCP port 52311 is open among peers, in order to allow the BigFix clients to download files from a peer. If this port is not open, the clients will not be able to download files from peers. It is recommended to set PeerNest in passive mode (using the `_BESClient_PeerNest_IsPassive` configuration setting) on clients that cannot open this port.
PeerNest also requires that the subnet supports multicasting. You will not be able to take advantage of this new feature on subnets that do not support multicasting functions.

Additionally, PeerNest requires an increased disk storage space.

**Optimal efficiency to use this mode**

The efficiency of the PeerNest feature is optimal in a subnet where all endpoints hosting the BigFix client use the same version of IP protocol. When this condition is not met, the same file might be downloaded from the BigFix relay by more than one client, and the time needed for all involved BigFix clients to be aligned getting the file cannot be minimized.

**Enabling PeerNest**

To enable the PeerNest feature, set to 1 the following configuration setting on the client:

```
_BESClient_PeerNest_Enabled = 1
```

The client enables all the PeerNest feature in order to locally optimize the download of binaries.

This configuration setting requires a restart of the client to be effective.

**PeerNest flows**

After enabling the PeerNest mode, the file download flow is as follows:

1. While executing an action that requires a download, the clients broadcasts over the subnet a UDP message in order to check if the file is available. The broadcast is executed on the subnet broadcast address, on port 52311, which is the agent UDP port.
2. Depending on the answer that the client receives:
   - One or more peers already have the file: The client chooses randomly one of the peers that have the file available and starts the download.
   - One peer is downloading the file from the relay: The client goes to sleep mode and waits for the download to complete.
   - No other peer is downloading the file: The client adds the file to a download pending list. It checks with the relay for the file availability. If the relay has it available, the client starts the file download.
3. After completing the download, the client notifies its peers that the download is now available. The notification is performed sending a multicast UDP message over the subnet on port 52311.

**Troubleshooting scenario 1**

On BigFix clients hosted on an Operating System that has the Internet Protocol version 6 (IPv6) disabled or not configured:

If you want to use the PeerNest feature, you must:

1. Set on these clients the \_BESClient\_Comm\_IPCommunicationsMode configuration setting as follows:
   
   \_BESClient\_Comm\_IPCommunicationsMode = OnlyIpv4

2. Restart the clients for the changes to take effect.

**Troubleshooting scenario 2**

On BigFix clients that have an active polling set using the \_BESClient\_Comm\_CommandPollEnable and \_BESClient\_Comm\_CommandPollIntervalSeconds configuration settings:

If you want to use the PeerNest feature, you must not configure these clients to be "passive" PeerNest agents. Do not enable on them the \_BESClient\_PeerNest\_IsPassive configuration setting. Otherwise, depending on the timing of the polling, multiple clients in a subnet can download the same binary, without sharing it.

**Configuring the clients**

You can configure the clients for peer to peer mode by using a few settings. For details, see [Peer to peer mode (on page 144)](#) in BigFix Configuration Settings (on page ).
Chapter 14. Archiving Client files on the BigFix Server

You can collect multiple files from BigFix clients into an archive and move them through the relay system to the server. This allows the BigFix Administrator to automatically log data from specific managed computers.

To do this, a new component called the Archive Manager has been added to the BigFix Client which can collect files periodically or on command. It passes the resultant compressed tar-ball to the Upload Manager on the BigFix Client. The Upload Manager has an input directory that queues the files for uploading.

The Upload Manager performs one upload operation at a time, moving the data in manageable chunks to reduce network traffic. It sends these chunks to the nearest BigFix relay or server, where the PostFile program reassembles the chunks back into the original file.

PostFile then passes the file up the chain, to the next BigFix relay or to its ultimate destination at the BigFix server. It again uses the Upload Manager to slice the file into chunks and send them on to the next PostFile program in the hierarchy. When the file finally arrives at the BigFix server, it is saved in a special directory location based on the ID of the client computer. Along the way, both the Upload Manager and the PostFile program can alter the chunk size or throttle the upload speed to smooth out network traffic.

For information about configuration settings related to these components, see Archiving client files (on page )

**Note:** When it encounters an unregistered BigFix Client, the Upload Manager pauses. This can happen for a variety of reasons, including a downed network, a busy server, or a disconnected client. As soon as the BigFix client can register with the BigFix system again, it restarts the Upload Manager and continues from where it stopped.
Archive manager settings

A typical archive is a collection of logs and configuration files that are compiled regularly and posted to the server. There are many settings available to help you customize your logging needs.

For details about the configuration settings related to this component, see Archive Manager (on page____) in BigFix Configuration Settings (on page____).

Creating a Custom Action

You can create custom actions that can post attributes about the BigFix client to an archive file.

To create a custom action:

1. Start the BigFix Console.
2. Select the Computers tab.
3. From the filter/list, select the set of computers that you want to target for the action.
4. Select Take Custom Action from the Tools menu.
5. Select the Action Script tab.
6. Enter the desired Action Script in the text box provided.

Archive Manager

Archive Manager is a component of the BigFix Client that can collect files periodically or on command. It passes the resultant compressed tar-ball to the Upload Manager on the BigFix Client.

For details about the configuration settings related to this component, see Archive Manager (on page____) in BigFix Configuration Settings (on page____).

Archive Manager internal variables

These are the internal variables of the Archive Manager component:
__BESClient_ArchiveManager_LastArchive

The Archive Manager updates this setting whenever it posts an archive. The value of the setting is the secure hash algorithm (sha1) of the file that was posted.

__BESClient_ArchiveManager_LastIntervalNumber

The BigFix Client updates this setting whenever it posts an archive. It represents the interval number from 1970 to the time when the archive was last collected. If the interval is a day long (the default), then the setting indicates the number of days from 1970 to the day when it created the last archive. It is calculated such that when the interval number changes, it is time to create a new archive.

The value is also offset by a time corresponding to the computer id, to stagger the collecting of archives.

Archive Manager Index File Format

During the building of the archive, the Archive Manager creates an index containing metadata about the archive. This is a sample index from an archive with a single file:

```
MIME-Version: 1.0
Content-Type: multipart/x-directory2; boundary="==="
Unique-ID: 1077307147
Archive-Size: 105
SendAll: 0
Date: Wed, 17 Mar 2004 02:23:01 +0000
FileSet-(LOG): c:\temp\log\newfile.log

---

URL: file:///c:/temp/log/newfile.log
NAME: (LOG)newfile.log
SIZE: 105
```
Upload Manager

The Upload Manager coordinates the sending of files in chunks to the Post File program. You can throttle the upload dataflow to conserve bandwidth. The file system uses 64-bits, sufficient for file sizes of up to $2^{64} - 1$ bytes in length.

For details about the configuration settings related to this component, see Upload Manager (on page 154) in BigFix Configuration Settings (on page 186).

PostFile

The PostFile program receives the chunks of files posted by the Upload Manager and appends them to its own copy of the file. The Upload Manager specifies the range of bytes being posted and the sha1 of the file, which is used as the filename. For details about the configuration settings related to this component, see Post File (on page 288) in BigFix Configuration Settings (on page 300).

These parameters are appended to the URL as in the following example:

```
postfile.exe?
sha1=51ee4cf2196c4cb73abc6c6698944cd321593007&range=1000,1999,20000
```

Here the sha1 value identifies the file, and the range in this case specifies the second 1,000 byte chunk of a 20,000 byte file.

When PostFile receives a chunk of the file it first checks to make sure it is the correct segment. If so, it appends the posted data to its local copy of the file. It returns the size of this file, as well as the current chunk size and throttle BPS settings.
PostFile has to handle several BigFix clients feeding into it at the same time. To balance that load, it adjusts the throttle rate. The effective throttling rate is determined by dividing the limiting PostFile rate by the number of concurrently uploading files.

For example, if PostFile has a throttle setting of 100 KBPS and 50 clients are simultaneously uploading files, the throttle value returned to each client would be adjusted to 2 KBPS. By setting custom throttle values to specific BigFix relays, you can efficiently deal with any bottlenecks in your network.

PostFile stores the partially uploaded files in the Upload Manager's buffer directory with an underscore in front of them (the Upload Manager does not upload files that begin with underscore). When PostFile receives the last chunk of the file, it calculates the sha1 of the file and checks that it matches the sha1 parameter in the URL. If so, it removes the leading underscore.

The Upload Manager can then upload the file to the next relay up the hierarchy (or any other server, if so specified).

PostFile determines whether or not the Upload Manager is running. If not, PostFile assumes that it has reached its root server destination. It renames the uploaded file, extracts the files from the archive, and deposits them in a subfolder of the Upload Manager's buffer directory.

The program calculates the subfolder path using a modulus of the computer ID. This has the effect of spreading out file directory accesses and preventing an overpopulation of any single directory.

For example, the path to file "log" from computer ID1076028615 is converted to the path "BufferDir/sha1/15/1076028615/log" where 15 is the remainder modulo 100 (the lower two digits) of the id.

If the uploaded file is a valid BigFix archive and is successfully extracted, then the original uploaded file is deleted.

Resource Examples

Example 1
In this example, we want to collect all the files in the `c:\log` folder and all the `.ini` files in the `c:\myapp` folder once an hour. Send up only the differences and don't send the archive if it exceeds 1,000,000 bytes in size. To set this up, create the following settings in the BigFix Console:

```
_BESClient_ArchiveManager_FileSet-(Log) = c:\log
_BESClient_ArchiveManager_FileSet-(Ini) = c:\myapp\*.ini
_BESClient_ArchiveManager_OperatingMode = 1
_BESClient_ArchiveManager_Interval_Seconds = 3600
_BESClient_ArchiveManager_SendAll = 0
_BESClient_ArchiveManager_MaxArchiveSize = 1000000
```

**Example 2**

In this example, we want the same set of files as above, but we also want to collect some useful attributes (retrieved properties) from the client computer. A custom action can generate these attributes and trigger an archive when it completes. It uses the same settings as above, but sets the operating mode to 2 to enable the `archive now` action command:

```
_BESClient_ArchiveManager_OperatingMode = 2
```

You can then create a custom action, specifying the attributes you want to collect. For example, to append the operating system, computer name, and DNS name to the log file, create a custom action like this:

```
appendfile {"System:" & name of operating system}
appendfile {"Computer:" & computer name}
appendfile {"DNS name:" & dns name}
delete "c:\log\properties.log"
copy __appendfile "c:\log\properties.log"
archive now
```

The `appendfile` command creates a temporary text file named `__appendfile`. Each time you invoke the command, it appends the text you specify to the end of this temporary file.
The **delete** and **copy** commands clear out the old log file (if any) and copy the __appendfile to the log. This has the effect of creating a new properties.log file. The **archive now** command immediately creates an archive, as long as the OperatingMode is set to 2.

You can then target this action to any subset of BigFix Clients, using whatever scheduling you choose. Using variations on this scheme, you could perform a full archive once a week, in addition to nightly differences.
Chapter 15. Maintenance and Troubleshooting

If you are subscribed to the Patches for Windows site, you can ensure that you have the latest upgrades and patches to your SQL server database servers. This means that you must install the client on all your computers, including the server and console computers. In addition, you might want to take advantage of these other tools and procedures:

- If you have the SQL Server installed, you should become familiar with the **MS SQL Server Tools**, which can help you keep the database running smoothly.
- It is standard practice to back up your database on a regular schedule, and the BigFix database is no exception. It is also wise to run the occasional error-check to validate the data.
- If you start to notice any performance degradation, check for fragmentation. BigFix writes out many temporary files, which might create a lot of disk fragmentation, so defragment your drive when necessary. Regular maintenance also involves running the occasional error-check on your disk drives.
- The BigFix **Diagnostics Tool** performs a complete test on the server components and can be run any time you experience problems. For additional information see the **BigFix Installation Guide**.
- Check the **BigFix Management** domain often. There are a number of Fixlets available that can detect problems with any of your BigFix components. This can often prevent problems before they ever affect your network.
- Check the BigFix Knowledge Base at [HCL BigFix Support site](https://www.hcl.com/support/bigfix). This site is continually updated, and if you cannot find an existing knowledge-base article about your question, you can find information about how to submit a question to HCL Software Support.
- Add relays to improve the overall system performance and pay close attention to them. Healthy relays are important for a healthy deployment.
- Review the **Deployment Health Checks** dashboard in the **BigFix Management** domain for optimizations and failures.
- Set up monitoring activities on the servers to notify you in the event of a software or hardware failure, including:
- Server powered off or unavailable
- Disk failure
- Event log errors about server applications
- Server services states
- FillDB buffer directory data back-up situations

Monitoring relays health

BigFix allows you to monitor your client and relay setups to ensure they are working optimally. Before deploying a large patch, you might want to check the status of your relays to guarantee a smooth rollout.

Here are some suggestions for monitoring your relay deployment:

- Click the BigFix Management domain and the Analyses node and activate the relay Status analysis. This Analysis contains a number of properties that give you a detailed view of the relay health.
- Click the Results tab for the analysis to monitor the Distance to relay property in the relay status analysis to see what is normal in your network. If your topology suddenly changes, or you notice that some of your clients are using extra hops to get to the server, it could indicate the failure of a relay.
- Try to minimize the number of clients reporting directly to the server because it is generally less efficient than using relays. You can see which computers are reporting to which relays by studying this analysis.

Relay and Server diagnostics

To monitor your BigFix environment setup and status and to complete actions on your clients.

You can use the following diagnostics functions to get information about your server and your relay settings and to complete actions on your clients. Starting from V9.5.6, the relay diagnostics page is disabled by default and can be protected by a password when enabled; for more information, see: Relay diagnostics (on page ).
To access the diagnostics, open a browser and type in the address field:

```
http://<computer_name>:52311/rd
```
or

```
http://<computer_name>:52311/RelayDiagnostics
```

where:

- `<computer_name>`
  
  Is the address of the workstation where the server or the relay that you want to check is installed.

The diagnostics page is divided in the following sections:

**Relay or Server Diagnostics**

In this section, you can gather information about your environment settings. Click the + sign to expand the different types of setting and see their values.

**Note:** The entry Query Settings refers to BigFix Query processing. For more information about this function, see [Getting client information by using BigFix Query](on page 128).

**Relay Status Information**

In this section, you can view information about the cache used on the relay in the queues dedicated to FillDB and toBigFix Query requests and results.

- **FillDB File Size Limit**
- **FillDB File Counter Limit**
- **Timeout for queries in queue** displays how long the BigFix Query requests can stay in the queue before being removed.
- **Size of queries in queue** shows the size of the cache that is used on the relay to store the BigFix Query requests.
- **Size of results in queue** shows the size of the cache that is used on the relay to store the BigFix Query results.
If you click the **Empty Query Queues** buttons, the queues that store the BigFix Query requests and results in the relay cache are cleaned up.

**Console user information**

In this section, you can check whether an user is authorized to access BigFix. This section is available only when you access the server diagnostic.

Click **Check User Authorization** and type the user's credentials to verify whether that user is granted access to the BigFix console without the need to actually log in with those credentials.

**Site gathering information**

In this section, you can collect information related to your environment sites.

- Click **Gather Status Page** to get information about site gathering status.
- Click **Gather All Sites** button to gather the latest version of site contents.
- In **Fixlet Site Requests**, you can collect information about different types of requests related to a site. Select the type of request, the URL of the site in the list provided, whether you want to use CRC or not and then click **Submit**.

**Client register**

In this section, you can perform requests either for a single computer or for all the computers in your environment.

- Click **Get Computer ID** button to know the computer ID of your relay.
- In **Single Computer Requests**, you can choose different types of requests related to a single computer by selecting one of the requests in the list and by clicking the **Submit** button. Depending on the request type, you might need to fill one or more text fields. The needed fields are automatically enabled.
- In **All Computer Requests**, you can select different types of requests related to all the computers in your environment by selecting one of the
requests in the provided list and clicking the **Submit** button. Depending on the request type, you might need to specify the **Action ID**, if enabled.

**Download information**

In this section, you can collect information about the downloads that are run on the system.

- Click **Download Status Page** to get information about downloads active on your server or relay.
- Click **Download Status Text Page** to get information, in xml language, about downloads active on your server or on your relay.
- In **Download Requests**, you can collect information about a specific action for a specific site by providing the **Action ID** and the **Site URL** in the related fields. Click **Gather Download Request** button to run your request.

**Virtualized environments and virtual machines**

To run your operating system in multiple virtual machines.

In BigFix you can run your operating system in multiple images to benefit from the ability to share hardware and software resources. This is true especially on HCL z Systems where, within the z/VM environment, Linux images benefit from the reliability, availability, and serviceability of IBM z Systems servers and from internal high-speed communications. z/VM offers an ideal platform for consolidating Linux workloads on a single physical server where you can run hundreds to thousands of Linux images.

In BigFix design, the BESClient agent works in a loop checking the activity to run based on the contents of its directory `<BESClient_installation_path>/__BESData`. These activities, together with a large number of concurrent virtual machines as it is common in z/VM environments, might result in a 100% CPU usage. To avoid this problem and control the CPU assignment to processes, use the configuration settings described in **CPU Usage** (on page ).

Some useful parameters are **_BESClient_Resource_WorkIdle** and **_BESClient_Resource_SleepIdle**, that have default values of 10 and 480 milliseconds.
respectively, to control the CPU consumption by balancing the amount of work with the amount of idle time; with the default values, this means about 2% of work for each virtual machine. You can change these values if you need to have a lower percentage; the negative side in this instance is that the BigFix client becomes slower when a new activity must be processed. By setting new values, you can take account of the number of virtual machines and avoid the overall CPU being 100% busy.

With other parameters you can set your agents to remain quiet during a part of the day and become active for the remainder of the day; during the quiet period the CPU consumption is almost 0%. The parameters that control this behavior are 
_BESClient_Resource_QuietEnable, _BESClient_Resource_QuietStartTime, and _BESClient_Resource_QuietSeconds. For example, by setting the following values:

```
_BESClient_Resource_QuietEnable=1
_BESClient_Resource_QuietSeconds=43200
_BESClient_Resource_QuietStartTime=07:00
```

the agent enters quiet mode at 07:00 AM each day, remains in this state for 43,200 seconds, that is for 12 hours, and wakes up at 07:00 PM. During quiet mode, the agent uses almost 0% of CPU time and does not process activities.

Other useful parameters to control the amount of time a client stays in sleep mode, especially suitable when there are battery low power problems or the need to reduce CPU utilization, are _BESClient_Resource_PowerSaveEnable and _BESClient_Resource_PowerSaveTimeout0. Good results can be obtained by setting them to 1 and 30 respectively; in this instance, clients remain in sleep mode for 30 minutes between each activity cycle. When sleep mode is active, the command polling is paused.

You might find useful also these parameters: _BESClient_Resource_PowerSaveTimeout1, _BESClient_Resource_PowerSaveTimeout2, _BESClient_Resource_PowerSaveTimeout3, _BESClient_Resource_PowerSaveTimeout4 and, _BESClient_Resource_PowerSaveTimeout5.

For a full description of all of these parameters and many more, see the configuration settings in the link listed previously.
Related reference

• List of settings and detailed descriptions (*on page*)

Related information

• CPU Usage (*on page*)
Chapter 16. Support

For more information about this product, see the following resources:

- Knowledge Center
- BigFix Support Center
- BigFix Support Portal
- BigFix Developer
- IBM BigFix Wiki
- HCL BigFix Forum
Notices

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

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

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

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

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

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

HCL TECHNOLOGIES LTD. PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.
This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. HCL may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

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

© (your company name) (year).

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

**Trademarks**

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

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

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

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

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

Terms and conditions for product documentation

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

Applicability

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

Personal use

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

Commercial use

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

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

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

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

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