Foglight® 8.3.0

Federation Field Guide

© 2026 Quest Software Inc.

ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software Inc.

The information in this document is provided in connection with Quest Software products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest Software products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST SOFTWARE ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest Software makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest Software does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.
Attn: LEGAL Dept.
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our website (https://www.quest.com) for regional and international office information.

Patents

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at https://www.quest.com/legal.

Trademarks

Quest, the Quest logo, and Join the Innovation are trademarks and registered trademarks of Quest Software Inc. For a complete list of Quest marks, visit https://www.quest.com/legal/trademark-information.aspx. "Apache HTTP Server", Apache, "Apache Tomcat" and "Tomcat" are trademarks of the Apache Software Foundation. Google is a registered trademark of Google Inc. Android, Chrome, Google Play, and Nexus are trademarks of Google Inc. Red Hat, JBoss, the JBoss logo, and Red Hat Enterprise Linux are registered trademarks of Red Hat, Inc. in the U.S. and other countries. CentOS is a trademark of Red Hat, Inc. in the U.S. and other countries. Fedora and the Infinity design logo are trademarks of Red Hat, Inc. Microsoft, .NET, Active Directory, Internet Explorer, Hyper-V, Office 365, SharePoint, Silverlight, SQL Server, Visual Basic, Windows, Windows Vista and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. AIX, IBM, PowerPC, PowerVM, and WebSphere are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Java, Oracle, Oracle Solaris, PeopleSoft, Siebel, Sun, WebLogic, and ZFS are trademarks or registered trademarks of Oracle and/or its affiliates in the United States and other countries. SPARC is a registered trademark of SPARC International, Inc. in the United States and other countries. Products bearing the SPARC trademarks are based on an architecture developed by Oracle Corporation. OpenLDAP is a registered trademark of the OpenLDAP Foundation. HP is a registered trademark that belongs to Hewlett-Packard Development Company, L.P. Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. MySQL is a registered trademark of MySQL AB in the United States, the European Union and other countries. Novell and eDirectory are registered trademarks of Novell, Inc., in the United States and other countries. VMware, ESX, ESXi, vSphere, vCenter, vMotion, and vCloud Director are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. Sybase is a registered trademark of Sybase, Inc. The X Window System and UNIX are registered trademarks of The Open Group. Mozilla and Firefox are registered trademarks of the Mozilla Foundation. "Eclipse", "Eclipse Foundation Member", "EclipseCon", "Eclipse Summit", "Built on Eclipse", "Eclipse Ready" "Eclipse Incubation", and "Eclipse Proposals" are trademarks of Eclipse Foundation, Inc. IOS is a registered trademark or trademark of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries. Apple, iPad, iPhone, Mac OS, Safari, Swift, and Xcode are trademarks of Apple Inc., registered in the U.S. and other countries. Ubuntu is a registered trademark of Canonical Ltd. Symantec and Veritas are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. OpenSUSE, SUSE, and YAST are registered trademarks of SUSE LCC in the United States and other countries. Citrix, AppFlow, NetScaler, XenApp, and XenDesktop are trademarks of Citrix Systems, Inc. and/or one or more of its subsidiaries, and may be registered in the United States Patent and Trademark Office and in other countries. AlertSite and DéjàClick are either trademarks or registered trademarks of Boca Internet Technologies, Inc. Samsung, Galaxy S, and Galaxy Note are registered trademarks of Samsung Electronics America, Inc. and/or its related entities. MOTOROLA is a registered trademarks of Motorola Trademark Holdings, LLC. The Trademark BlackBerry Bold is owned by Research In Motion Limited and is registered in the United States and may be pending or registered in other countries. Quest is not endorsed, sponsored, affiliated with or otherwise authorized by Research In Motion Limited. Ixia and the Ixia four-petal logo are registered trademarks or trademarks of Ixia. Opera, Opera Mini, and the O logo are trademarks of Opera Software ASA. Tevron, the Tevron logo, and CitraTest are registered trademarks of Tevron, LLC. PostgreSQL is a registered trademark of the PostgreSQL Global Development Group. MariaDB is a trademark or registered trademark of MariaDB Corporation Ab in the European Union and United States of America and/or other countries. Vormetric is a registered trademark of Vormetric, Inc. Intel, Itanium, Pentium, and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries. Debian is a registered trademark of Software in the Public Interest, Inc. OpenStack is a trademark of the OpenStack Foundation. Amazon Web Services, the "Powered by Amazon Web Services" logo, and "Amazon RDS" are trademarks of Amazon.com, Inc. or its affiliates in the United States and/or other countries. Infobright, Infobright Community Edition and Infobright Enterprise Edition are trademarks of Infobright Inc. POLYCOM®, RealPresence® Collaboration Server, and RMX® are registered trademarks of Polycom, Inc. All other trademarks and registered trademarks are property of their respective

owners.

Legend

WARNING: **A WARNING icon indicates a potential for property damage, personal injury, or death.**

CAUTION: **A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.**

IMPORTANT NOTE, **NOTE**, **TIP**, **MOBILE**, or **VIDEO:** An information icon indicates supporting information.

Foglight Federation Field
Guide Foglight Version- 8.3.0

Contents

Federation overview **5**
Use cases 5
Capabilities 6
Known issues 7
Requirements 8
Scalability expectations 9

Managing federation **10**
Understanding federation communication channels 10
Topology synchronization 10
Metric pull on demand 11
Alarm service communication channel 12
Planning for federation 13
Communication ports 13
HA considerations 14
Configuring a federated environment 14
Setting up a federated environment 14
Federation configuration changes 16
Security settings 17
Query limitations 18
Custom script queries 19
Upgrading cartridges on a federated system 19
Selectively disabling federation services 19
Deploying cartridge components 20
Troubleshooting 21
Frequently asked questions 21

About Us **27**
We are more than just a name 27
Our brand, our vision. Together. 27
Contacting Quest 27
Technical support resources 27

1

Federation overview

Federation is a Foglight® feature that addresses the needs of customers who monitor large-scale environments that are naturally partitioned into logical units. Each partition is traditionally served by one Foglight Management Server instance and its distributed clients/agents.

This document presents an overview of the federation feature and provides guidelines for configuring Federation Masters and Federated Children.

The intended audience for this document is the Quest SC/PSO (System Consultant/Professional Services Organization) personnel working in the field.

This section describes a number of cases where the Foglight federation feature is used. In addition, a list of federation capabilities, known issues, and requirements are provided.

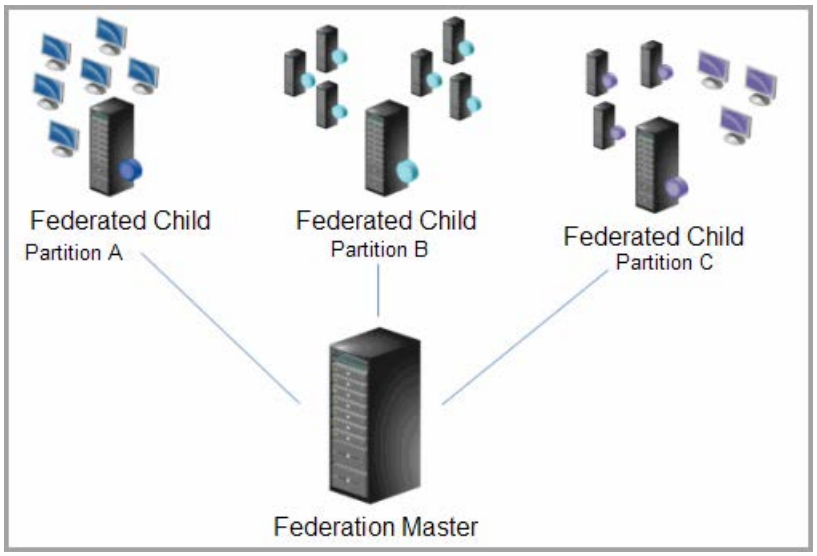
- Use cases
- Capabilities
- Known issues
- Requirements
- Scalability expectations

Use cases

Federation enables the use of one Foglight® Management Server to operate, while using other Management Servers as the data source. As illustrated in the following figure, a Foglight user can visualize the data from one or more stand-alone Management Servers (the Federated Children) from a single, central Management Server (the Federation Master). This allows users to meet both local and central monitoring requirements.

NOTE: It is not possible to perform administration of Federated Children from the Federation Master.

Figure 1. Federation use cases



Federation should be used to address the following customer environment issues:

- In large environments, where customers need to manage reports, view alarms, and track service levels of items which run on more than one Management Server.

  For example:

  - A customer has an environment consisting of 3000 Oracle databases, and a requirement to create a single report that summarizes activity across all 3000 instances. In this case, a Federation Master can be used to pull the data from two or more Management Servers (Federated Children).
  - A customer has a large environment (hundreds of JVMs and hundreds of databases) with one common message bus, which is a key component of every application. In this case, the Federation Master can be used to create application/service models based on topology objects from more than one Federated Child. Service levels, in this case, are calculated on the Federation Master.

- In an environment where initially a single Management Server is set up, but where customers must see technology views that cross several Management Servers.

  For example:

  - Due to purchasing constraints, the business divisions purchase Management Servers for their applications only. As a result, the technical management structure does not align with the business organization. The database administrators and system administrators need to have a unified view of the technology (for example, databases and hosts), across business divisions. To address this problem, the existing servers must be federated.

- In an environment where customers require scalability improvements on a single Management Server.

  For example:

  - If a customer is nearing the scalability limit, or is experiencing slow performance, a Federation Master can be added to remove the UI (user interface) processing burden on the primary Management Server. As a result, this increases the ability of the primary server to process the rules, derived metrics, and calculations.

Capabilities

Federation Masters have the following capabilities:

- Unifying and periodically synchronizing the model between more than one Foglight® Management Server.

> **IMPORTANT:** If an object is deleted from a Federated Child but still exists on another Federated Child, the object is not deleted from the Federation Master. However, if an object is deleted from a Federated Child and does not exist on any other Federated Child, the object and all references to it are deleted from the Federation Master.

- Serving as the UI, by pulling data on request, from one or more other Management Servers.

  > **IMPORTANT:** The Federation Master does not store data. It queries data, on demand, from the Federated Children.

- Building new applications/services that use elements crossing more than one Management Server.
- Processing and storing time-driven derived metrics by gathering the required data at run time from one or more Management Servers.
- Processing and performing actions on time-driven rules by gathering the required data at run time from one or more Management Servers.

The federation feature allows users to:

- See a complete view of all topologies combined in the Foglight browser interface. Once the information is centralized, this information is available regardless of whether the Federated Children are running or not.
- See data in charts as if it were present on the Federation Master (pulled on demand).

  > **IMPORTANT:** If one or more of the Federated Children are not available, the affected data charts will be empty (“No data” may be displayed). Metrics and complex observations (pulled data) are not stored or cached on the Federation Master.

- Go to the **Alarms Viewer** and see all current alarms that are reported by any Federated Child.

  > **IMPORTANT:** In Foglight 5.7.x, users can control the alarms’ refresh period by setting the MaxAlarmUpdateDelay parameter in the *federation.config* file. For more information, see Alarm service communication channel.

- Define services and add services, categories, and objects from Federated Children.

  > **IMPORTANT:** If an object is deleted from a Federated Child but still exists on another Federated Child, the object is not deleted from the Federation Master. However, if an object is deleted from a Federated Child and does not exist on any other Federated Child, the object and all references to it are deleted from the Federation Master.

- Write time-driven rules and derived metrics that operate on data across Federated Children.
- Acknowledge and clear all alarms originating from any of the Federated Children.

  > **IMPORTANT:** Some predefined rules are unavailable on the Federation Master.

Known issues

Foglight® 8.3.0 has the following known issues with respect to federation:

- Administration of Federated Children from a single Federation Master is not supported.
- In most cases, data-driven rules are ineffective in the Federation Master, because this server does not receive data from agents. Only data-driven rules that rely on internal Management Server metrics work in the Federation Master.
- If the source event originates on the Federation Master, then associated event-driven rules work well. If the source event originates on a Federated Child, associated event-driven rules do not fire.

- Agents are not able to connect to Federation Masters directly. If an agent is configured to attempt a direct connection, it will receive errors similar to the errors that agents normally receive when the server is not running.
- Running the Federation Master in High Availability (HA) mode is not supported. Only the Federated Children can be run in High Availability mode.
- Only a single layer of federation is supported (that is, no federation of Federation Masters is provided).
- There is a potential time-delay for agents.

  IMPORTANT: There is no time delay for data once the topology model is synchronized.

- Custom dashboards are not propagated from Federated Children to the Federation Master. Customers must manually push custom dashboards to the Federation Master if they wish to use them.
- Foglight for Automation is not supported in a Foglight federation environment. This limitation will be addressed in a future release. For more information, see the *Foglight Release Notes*.
- When Federation Children have Dynamic Managed Components with the same name, the Federation Server does not merge their queries, displaying only the components from the first synced Federation Child.

Requirements

Federation has the following requirements:

- A server must be dedicated as a Federation Master.
- The minimum hardware needed by a federated environment to run Foglight® follows the recommendations for any Management Server. It varies widely, depending on a number of factors, including the number of agents being used, the persistence and data-rollup policies, agent configuration settings, and platform. For detailed information, see the *Foglight System Requirements and Platform Support Guide*.
- Identical cartridges must be installed on the Federation Master and the Federated Child(ren).
- Version compatibility: The Federation Master is compatible with Federated Children that are running a supported version which is the same as or earlier than the Federation Master.
- The maximum number of Federated Children in a federated environment is 20.

Scalability expectations

The deciding factor for scalability is the unified topology monitored by all of the Federated Children. The Federation Master keeps an internal representation of this global view. The memory consumption for the Federation Master is 1:4 (that is, four Federated Children are handled by one Federation Master).

To satisfy memory address space, sufficiently large environments may require a 64-bit VM to run the Federation Master.

2

Managing federation

Federation is an advanced deployment option that requires careful consideration. At a minimum, it requires dedicated hardware for a Federation Master, a plan for propagating and unifying models, and an established workflow to execute the administration of the servers.

This section describes the use of the Federation Master and Federated Children, and how to configure the federated environment. It also provides troubleshooting information.

- Understanding federation communication channels
- Planning for federation
- Configuring a federated environment
- Troubleshooting
- Frequently asked questions

Understanding federation communication channels

Topology is retrieved from all servers periodically and then stored in the federation database. When the UI requests topology data, it is retrieved from the local federation database.

Three “channels” of communication exist between the Federated Children and the Federation Master: Topology synchronization, Metric pull on demand, and the Alarm service communication channel.

Topology synchronization

The Federation Master periodically retrieves topology information from Federated Children and merges it into the federation model—a unified model that represents the environment as a whole (see Model union rules on page 11). Because the model in the Federation Master is a union of the other servers, all of the normal UI components in Foglight® work as if they were on a non-federated server.

Topology information is retrieved in small chunks in previous releases, which causes the synchronization process takes significant time to complete in large environments for each synchronization cycle. The Management Server, starting with version 5.9.1, has improved the synchronization methodology to implement the synchronization in an incremental way. This means the initial synchronization process still takes much time to complete in version 5.9.1 (or later) but in the subsequent synchronization cycles, the topology object is to be synchronized from Federation Children only when its version varies from its copy existing on Federation Master. Consider for example the topology object is automatically refreshed every 5 minutes (For more information about how to change the interval of topology object refresh, see Step 9 in the Setting up a federated environment procedure). The Management Server checks topology objects on Federation Children after the specified 5 minutes, to see if they are different from the copies on Federation Master. If a topology object’s version gets changed, the Manager Server starts synchronizing the changed objects other than copying all topology objects on Federation Children.

The Federation Master automatically synchronizes certain topology types with Federated Children. For example: property, metric and observation definitions are synchronized, but type and property annotations are not. Topology types are now pulled from Federated Children at the beginning of each synchronization cycle.

The extent of the topology to be synchronized is controlled by the `TopologyQueries` section in the *federation.config* file (the default is `everything`).

The maximum acceptable difference in system time between Federated Children and the Federation Master (in milliseconds) is controlled by the `MaxSystemTimeDifference` parameter in the *federation.config* file (the default is 60000 milliseconds, or one minute). Large system time differences can lead to inconsistencies in metrics and alarm data on the Federation Master. Foglight does not provide time synchronization services, therefore, system time among servers in a federation must be kept synchronized by other means, for example by using the Network Time Protocol (NTP).

NOTE: The measured time difference may include network delays. Therefore, setting the MaxSystemTimeDifference parameter to a value too small may result in false positive warnings. The recommended value for this setting is 5000 or less.

Model union rules

Objects that exist on two or more Federated Children are merged when pulled by the Federation Master. The union of these object instances includes all contained objects of each respective remote instance.

The following is an example of the model union rules for a given service object defined in two Federated Children:

```
Federated Child 1
   Service MyService
      Service Backend
      Service Frontend
         Host 1
         Host 2
         Host 3
Federated Child 2
   Service MyService
      Service Production
Federation Master
   Service MyService
      Service Production
      Service Backend
      Service Frontend
         Host 1
         Host 2
         Host 3
```

Metric pull on demand

When you request data for a particular metric through the Foglight® browser interface (for example, a time-driven rule or derivation, or a script), the Federation Master searches the Federated Children for the requested data, and retrieves the data for the specified time period.

Federation supports merging metrics from several sources:

- If more than one Federated Child provides data for the same metric, then the metric values are retrieved and aggregated across all those servers.
- If the Federation Master and some Federated Children provide data for the same metric (for example, if the metric is re-derived in the Federation Master), then the local data from the Federation Master is used.

The same set of servers is used for servicing subsequent requests for a certain period of time unless the state of the distributed data service is reset in the Federation Master (for example, by a change in the *federation.config* file). This period of time is (by default) set to three minutes, and is configurable via the instruction:

```
-Dfoglight.distributed.persistence.cache.ttl=<milliseconds>.
```

When the UI sends a new request, which comes three minutes after the original request time, the Federation Master re-determines the source of the metric by contacting all Federated Children again and by checking the local database for data.

NOTE: The “original request time” is the time when the Federation Master has first identified which Federated Child provided a particular metric.

Alarm service communication channel

Alarm data used in the Federated Child uses a communication channel that is separate from metric data. While metrics pull data directly from the Federated Children, an intermediate alarm service actually gathers and queues alarm data for use by the Federation Master. Therefore, when the Federation Master requests alarm data, it is actually gathering that data from the alarm service queue, rather than from each of the Federated Children directly. The alarm service is used to help prevent excessive queries of the Federated Children, in order to calculate time-driven rules and derived metrics (on the Federated Child) that depend on alarm data.

The impact of this is that the refresh interval of the alarm service dictates how long it takes for Federated Child alarms to appear on Federation Masters.

In Foglight® 8.0.0, the refresh interval is configurable. This leads to delays like the one illustrated in the following example.

Example:

1. At 11:00:00 the Federation Master runs the query and pulls back data on EJB X from Federated Child Y.
2. At 11:00:10 Federated Child Y detects a FATAL condition on EJB X.
3. At 11:00:30 someone viewing the Federation Master data does not see the state of the object as FATAL.

   IMPORTANT: The “state of the object” is derived from alarms, therefore, it is affected by the way you see alarms and topology objects.

4. At 11:02:10 the Federation Master refreshes the alarm state for EJB X, and the FATAL condition shows up in the object.

The alarm service refresh time can be adjusted.

In Foglight 8.0.0, the standard method recommended for configuring the alarm refresh rate is by setting the `MaxAlarmUpdateDelay` parameter in the *federation.config* file. This property specifies the maximum delay (in seconds) that is allowed in the Federation Master before it checks all Federated Children for alarm changes.

The Federation Master does not run periodic checks; it only retrieves alarm data from the Federated Children, on demand. The following two factors control when the retrieval operation occurs:

- The value set for the `MaxAlarmUpdateDelay` parameter in the *federation.config* file—this parameter is programmable and has a default value of 50 (seconds).
- The value set for the internal alarm cache—this parameter can be adjusted using the command:

  `-Dfoglight.uiquery.alarm.cache.duration.millis =<milliseconds>`

  and has a default value of 15 minutes (therefore, it is recommended to set the `MaxAlarmUpdateDelay` to a value less than 15 minutes).

  IMPORTANT: Changing this setting is not recommended.

Alarm data is refreshed on the Federation Master when the all of the following conditions occur:

- A request for that data is generated (for example, from a rule or from the UI).
- The last refresh was done more than `MaxAlarmUpdateDelay` seconds ago.
- The alarm data has been changed on the Federated Children.

Planning for federation

CAUTION: Before setting a more rapid refresh rate, it is prudent to consider the volume of alarms and potential performance impact of frequent alarm queries on the Federated Children.

NOTE: Federation is an advanced feature, which requires a dedicated Management Server. Contact Quest PSO to design and implement a federation strategy.

When planning to deploy the federation feature, customers must respect the following practical considerations:

- No agents may report directly to the Management Server.

  If customers require monitoring of the Management Server OS/host/database, they must do it from a Federated Child.

- Consideration must be given to service names.

  Services on Federated Children sharing the same name are joined in the Federation Master (for example, if on Federated Child A, a user has created a service `Foo`, and on Federated Child B, another user has created a service `Foo`, then both services are merged in the Federation Master to service `Foo`, which contains any logical components included in either A or B).

- Connectivity.
- Scalability.
- Service models and where they will be built.
- Administration of the Federation Master.

  Administration does not propagate in any way (Federated Child to Federation Master, Federation Master to Federated Child).

- Cartridge versions (the Federation Master must have a super set of all installed cartridges in all Federated Children).
- Co-locating all Management Servers is essential in order to ensure low network latency between them. Placing Management Servers in different data centers could affect dashboards on the Master Server as a result of WAN latency. Preventing a network lag between Federation Master and Child servers is therefore critical:
  - **10 ms or lower** in most cases leads to optimal results.
  - **10 ms to 35 ms** might be acceptable, depending on the complexity and frequency of browser interface requests.
  - **35 ms and higher** is likely to result in significant slowness

Communication ports

To ensure that the federation setup works correctly, the Federation Master must open a certain set of connections to each Federated Child. When a firewall exists between the Federation Master and the Federated Children, the TCP 4444 port must be opened on the Federation Children.

Federation connections are always established in the Federation Master to Federated Child direction. Once a connection is established, data transfer is bidirectional.

The Federation Master uses the RMI 1099 port during the connection setup process. In the *server.config* file of each Federated Child, specify the *server.federation.port* property to identify this port number.

IMPORTANT: Starting in version 5.9.2, Foglight® for Java EE Technologies no longer supports Nexus Federation. Without Federation, all application servers that could possibly participate in servicing the same request (through RMI or other remote calls) should be connected to the same Nexus.

HA considerations

High Availability (HA) mode enables you to install multiple servers, where one server is the primary server and the others are standby servers. If the primary server stops responding, then the responsibility is taken over by one of the standby servers.

Only the Federated Children can be run in High Availability mode. Running a Federation Master in HA mode is currently not supported.

In a federated environment with two or more Federated Children working in HA mode, the Management Server automatically chooses the primary Federated Child for data synchronization.

Configuring a federated environment

The Federation Master is a Foglight® Management Server that has been configured to incorporate the data of federated instances. A Federation Master can **not** be utilized as any other Management Server (for example, no connected clients, agents, standard rule analysis).

When installing a Management Server, federation is by default disabled. When you install a Management Server, the server does not know if it is going to operate as a Federation Master or as a normal server. The server identifies this only on startup, by checking the `server.federation` variable in the *server.config* file. If this variable is set to `TRUE`, the server makes a number of changes that are necessary for it to run efficiently (given a certain load).

A Federation Master must always be started with `server.federation` variable set to `TRUE`. If a Federation Master starts up, even just one time, as a normal server, it makes modifications to the server that deplete its performance as Federation Master.

CAUTION: If you accidentally start the server with this variable set to FALSE, you must then either restore from a backup (if you have one), or completely re-install the Federation Master and database.

When upgrading a Federation Master, you must first verify that the `server.federation` variable was not changed to `FALSE` before you restart the server.

IMPORTANT: Starting in version 5.9.2, the Cartridge for Java EE Technologies no longer supports Nexus Federation. Without Federation, all application servers that could possibly participate in servicing the same request (through RMI or other remote calls) should be connected to the same Nexus.

Setting up a federated environment

To set up a federated environment:

1. Install and configure all the Management Servers that you want to include as Federated Children in the federated environment. For information about how to install a Management Server, refer to the *Foglight® Installation and Setup Guide* set.
2. Install all the cartridges required on these Federated Children. For information about how to install a Foglight cartridge, refer to the cartridge's documentation set.
3. Perform a fresh installation of the Management Server that you want to include as Federation Master in the federated environment. For information about how to install a Management Server, refer to the *Foglight Installation and Setup Guide* set.

   IMPORTANT: Re-configuring an ordinary Management Server to Federation Master is not currently supported.

4. Enable the federation mode by setting the `server.federation` variable to `TRUE`, in the Federation Master's *server.config* file.

```
# server.federation is a boolean parameter that indicates whether the
# server should start in the "federation" mode. The default value is false.

server.federation = true;
```

5 Start the Federation Master.

6 Install all the cartridges required on the Federation Master.

> **IMPORTANT:** The Federation Master must have a super set of all cartridges installed in all Federated Children.

7 In the *server.config* file of each Federated Child, identify the port number allocated for communicating with the Federation Master (that is, identify the `server.federation.port` property).

The following is an example of defining the port number for three Federated Children, in their respective *server.config* files:

- Federated Child 1: `server.federation.port = "1099";`
- Federated Child 2: `server.federation.port = "1099";`
- Federated Child 3: `server.federation.port = "1099";`

8 Add the port number of each Federated Child in the federation to the *federation.config* file of the Federation Master.

If the Federated Children are being run in HA mode, add URLs for all Federated Children in the cluster (primary and stand-by) to the *federation.config* file.

All other Management Servers in the environment are made Federated Children by virtue of this inclusion.

> **IMPORTANT:** By default, the server checks the federation.config file every minute for changes. The changes are applied immediately, without rebooting the server.

The following is an example of how to add to the *federation.config* file the port numbers for the three Federated Children specified in Step 7.

```
# *** FederationURLs ***

FederationURLs = (

      "rmi://child1.example.com:1099",

      "rmi://child2.example.com:1099",

      "rmi://child3.example.com:1099",

);
```

9 In the *federation.config* file, edit the topology auto refresh period, if required. The default provided is 300 seconds (5 minutes):

```
# *** TopologyRefreshPeriod ***

# This property specifies the period in seconds for topology refresh

# operations. The initial topology refresh re-fetches all relevant

# topology objects from all federated children and merges them into

# the local topology model. The subsequent refresh only fetches topology

# objects which version gets changed.

# The default value is 300, that is 5 minutes.

TopologyRefreshPeriod = 300;
```

> **IMPORTANT:** You can also control the size of the federation topology model by editing the list of TopologyQueries. This is an advanced configuration setting (by default, the server pulls everything), which requires careful consideration of dependencies among topology objects. Configuring this subset incorrectly does not break the server, but some UI views may not work properly or may have missing data.

Federation configuration changes

If you are about to upgrade the Management Server to 5.9.1 from any version of 5.7.5.x, refer to Table 2: Federation configuration changes for the detailed configuration changes about Federation environment.

Table 2. Federation configuration changes

	Configuration file	Version 5.7.5.x	Version 5.9.1	Note
Port change	*<FMS_home>/config/server.config*	The following ports used in 5.7.5.x have been removed in 5.9.1. • *server.jndi.rmi.port* • *server.jndi.jnp.port* • *server.jrmp.invoker.port* • *server.pooled.invoker.port* • *server.unified.invoker.port*	The Management Server only requires *server.federation.port*, as JBoss Server has been removed in 5.9.1.	The value of *server.federation.port* is the value of *server.jndi.jnp.port.*

URL configuration change	*<FMS_home>/config/federation.config*	The following illustrates the implementation of URL configuration in 5.7.5.x.
		`# *** JndiURLs ***`
		`#`
		`# This list contains JNDI provider URLs for federated servers.`
		`...`
		`# The default value is JndiURLs = ();`
		`#`
		`JndiURLs = (`
		`);`
			The following illustrates the implementation of URL configuration in 5.9.1
			`# *** FederationURLs ***`
			`#`
			`# This list contains RMI provider URLs for federated servers.`
			`...`
			`# The default value is FederationURLs = ();`
			`#`
			`FederationURLs = (`
			`);`	

Table 2. Federation configuration changes

	Configuration file	Version 5.7.5.x	Version 5.9.1	Note
Security settings change	*<FMS_home>/config/federation.config*	The following illustrates the implementation of **SecurityToken** used in 5.7.5.x. *# *** SecurityToken **** *#* *# A token can be specified here, to enable token-based authentication. Once the master and children* *...#* *# Example:* *# SecurityToken = "PutYourFederation SecretTokenHere";* *#*	The implementation of **SecurityToken** has been removed in 5.9.1. Instead, the Management Server provides the SSL configuration which offers the more advanced security settings for Federation environment. For more information, see To configure an SSL for communication between the Federation Master and a Federated Child: on page 18.	For more information, see Security settings on page 17.
Topology sync change	*<FMS_home>/config/federation.config*	The following illustrates the implementation of topology objects sync in 5.7.5.x. *# *** TopologyRefreshPeriod **** *#* *...* *# The default value is 1800, that is 30 minutes.* *#* *TopologyRefreshPeriod = 1800;*	The following illustrates the implementation of topology objects sync in 5.9.1. *# *** TopologyRefreshPeriod **** *#* *...* *# The default value is 300, that is 5 minutes.* *#* *TopologyRefreshPeriod = 300;*	The default value of ***TopologyRefreshPeriod*** in *federation.config* has changed to 5 minutes. For more information, see Topology synchronization on page 10.

Security settings

The security settings of Federated Children are not shared with the Federation Master, each is its own security domain with user authentication settings, users, groups and role assignments.

Operations that require authentication (such as clearing alarms), when performed on the Federation Master, are authenticated on the Federation Master and executed on a respective Federation Child based on a trusted relationship. This trust is established by a security sockets layer (SSL) for Federation that must be configured on both the Federation Master and the Federated Child. Failing to configure an SSL for Federation prevents the authentication of Federation Master actions on the Federated Child, such as acknowledging or clearing alarms.

NOTE: Configuration of **SecurityToken** through the *federation.config* file (adopted in Foglight 5.7.5.8 and before) has been removed in Foglight 5.9.1. Therefore, configuring an SSL for Federation is required when you upgrade the Federation environment to version 5.9.1, if your Federation environment configures **SecurityToken** before.

NOTE: By default, federation SSL is **not** configured on federation environment, so that operations that require authentication (such as clearing alarms on Federation Master) will not allowed. Other operations (such as topology synchronization, observation fetching and alarm fetching which are the main operations for federation) still work properly on a normal socket. If federation SSL is configured on federation environment, all operations will work on an SSL.

To configure an SSL for communication between the Federation Master and a Federated Child:

1. Open the Foglight JMX Console on the Federation Master. Click **service=FederationConfig** under the *com.quest.nitro.fed* service.

   The *FederationConfig service* dashboard opens. Locate the *generateFederationKeyStores()* function, and then click **Invoke** to generate *Keystore* and certificate files. The generated *Keystore* and certificate files for Federation Master and Federation Child are saved under the *<Foglight HOME>/config* directory on the Federation Master.

2. Open the *federation.config* file on the Federation Master and locate the following lines:

   ```
   # KeyStore = "./config/federation-master.keystore";
   ```

   ```
   # KeyStorePwd = "q171ede14ed29f0c1ee9dc65f698d8e6d";
   ```

3. Uncomment the line by removing the `#`.
4. Save the *federation.config* file.
5. Copy the *federation-child.keystore* file of Federation Master to the *<Foglight HOME>/config* directory on the Federation Child.
6. Open the *server.config* file on Federation Child and locate the following lines:

   ```
   # server.federation.ssl = true;
   ```

   ```
   # server.federation.keystore = "./config/federation-child.keystore";
   ```

   ```
   # server.federation.keystore.pwd = "q171ede14ed29f0c1ee9dc65f698d8e6d";
   ```

7. Uncomment the line by removing the `#`.
8. Save the *server.config* file.
9. Restart the Federation Child Foglight server.
10. Repeat Step 5 through Step 9 for each Federated Child that is connected to Federation Master.

NOTE: Make sure to configure all Federation Child servers to establish the SSL communication with the Federation Master. This means Federation Child servers must enable or disable the Federation SSL at the same time.

Query limitations

The following is an example of a topology object query that limits the objects pulled by the server:

```
TopologyQueries = (
"!JavaEETopologyObject",
"!Host : name = 'example.com' ",
);
```

In general, when limiting the scope of topology queries, some interface views may not work properly or have missing data. If the federation topology must be limited, then the queries must be manually modified with respect to the desired topology scope and the cartridges involved. Customizing queries after the initial topology pull is less

likely to cause errors. The benefit would be shorter topology refresh cycles, which can be configured to run more frequently (to refresh only the “interesting” part of the model).

Custom script queries

When deploying cartridges on a federated environment, by default, the following components are not deployed on the Federation Master:

- Monitoring Policies (for example, Rules, Derived Metrics)
- Agent Adapters
- CDTs (Canonical Data Transformation) and Agent Adapters Extensions

Custom (Groovy) script queries running on a standalone Foglight® Management Server do not show on the Federation Master after the *.car* file has been successfully deployed. This is because the `federateDeployType` attribute of query components in cartridges is set to `DEFAULT_DEPLOY` by default.

To allow custom script queries defined in cartridge components to appear on the Federation Master:

- Set the `federateDeployType` component to `ALWAYS_DEPLOY`.

NOTE: This procedure is valid for all cartridge components.

Upgrading cartridges on a federated system

To upgrade a cartridge on a federated system:

1. Disable the topology synchronization in the Federation Master by commenting out all queries in the *federation.config* file.
2. Upgrade the cartridges on the Federated Children.
3. Upgrade the cartridges on the Federation Master.
4. Enable the topology synchronization, by editing the queries in the *federation.config* file.

Selectively disabling federation services

NOTE: This is an advanced configuration procedure; it is not part of your basic federation setup process. Use it only if necessary.

Certain areas of the federation functionality can be disabled independently of each other with the help of Java system properties, as presented in the following table.

Table 3. Disabling Federation services

Java System Property	Value	Result
`server.federation.remote.alarms.disabled`	true	Disables downloading alarms from Federated Children. The Federation Master continues to show local alarms.
`server.federation.remote.properties.disabled`	true	Disables remote calls during dynamic properties evaluation. Any dynamic properties that require information from Federated Children have null values.
`server.federation.remote.data.disabled`	true	Disables retrieving observations (including metrics) from Federated Children. The Federation Master continues to show local metrics (for example, local derived metrics).

Table 3. Disabling Federation services

Java System Property	Value	Result
`server.federation.remote.topology.disabled`	true	Disables topology synchronization. This option is useful only after an initial topology synchronization has been performed.
`server.federation.type.synchronization.disabled`	true	Disables topology type synchronization between Federated Children and the Federation Master.

These properties must be set only on the Federation Master. To take effect, any changes to these properties require a server restart using the `-D` option. For example, to disable remote calls during dynamic properties evaluation, restart the server using the option:

```
-Dserver.federation.remote.properties.disabled=true
```

Deploying cartridge components

NOTE: This is an advanced configuration procedure; it is not part of your basic federation setup process. Use it only if necessary.

Cartridges consist of components of various types. Each component is deployed and enabled individually. Whether a particular component is enabled in a Management Server depends on the server type (Federation Master or Federated Child) and on the federation deployment type.

You can control whether components are actually deployed by overriding the default federation deployment type during the cartridge build process. If no explicit value is provided for the federation deployment type, the server uses the default values.

The federation deployment type of a component is set in the *federateDeployType* cartridge manifest entry for that component. The cartridge-building ant task (<*car*>) allows you to set that attribute by using the *federateDeployType* (optional) parameter on cartridge sub-elements.

The following table lists possible values of the *federateDeployType* parameter.

Table 4. *federateDeployType* parameter values

Value	Effect
<not specified>	Same as `DEFAULT_DEPLOY`.
`DEFAULT_DEPLOY`	The server uses the default value based on the type of the cartridge component.
`ALWAYS_DEPLOY`	The component is deployed on the Federation Master and on the Federated Children.
`FEDERATE_ONLY_DEPLOY`	The component is deployed on Federation Masters, but not on Federated Children.
`NON_FEDERATE_ONLY_DEPLOY`	The component is deployed only on Federated Children, not on Federation Masters.

If a cartridge component has the `DEFAULT_DEPLOY` federation deployment type, the Management Server uses a pre-defined set of rules (based of the type of the component) to decide whether to deploy the component or not. The following table presents the cartridge components that are **not** deployed by default on Federation Masters.

Table 5. Cartridge components not deployed by default on Federation Masters

Component Type	Examples of artifacts that may be deployed in components of this type
Agent Configuration	Agent configuration (agent properties).
Adapter Extension Config	Agent adapter libraries, upgrade packages.
Installers	Agent packages and installers.

Table 5. Cartridge components not deployed by default on Federation Masters

Component Type	Examples of artifacts that may be deployed in components of this type
Monitoring Policy	Rules, derivations, registry variables, scripts, baseline computation parameters.
Agent-Adapter	Agent adapters (for example, Foglight Agent Manager Adapter).

Troubleshooting

This section presents a list of known federation-related issues:

- Foglight® Management Server federation mode fails if the Federated Child runs on a Red Hat 5 environment and it does not have a DNS entry added.

  Federation connections are established in two phases. In phase one, the initial connection is made to the address specified in the federation URL. In phase two, another connection is made to the address returned from the first communication phase. That second address is used by the Federated Child to identify itself. In this case, this name is not valid for other servers, which leads to connection failures.

  Workaround: Add DNS entries for all Federated Children. Alternatively, bind each Federated Child to a specific IP address (as opposed to the default host name); for details, see “Binding the Foglight Management Server to an IP Address” in the *Foglight Installation and Setup Guide* set.
- Agent hangs while trying to reconnect to a federated environment.
- On the Federation Master, the Java EE Requests dashboard can show requests that are not active in the time range.
- Topologies in the Federated Children are merged with the topology in the Federation Master.
- `java.lang.IllegalArgumentException: null object name` errors appear in the Federation Master log file when a Federated Child is shut down.
- Cartridge files (*.car*) installed on a Federation Master must be the exact same versions as the cartridge files installed on the Federated Children.
- Editing the rule definition of an alarm from the Federated Child is currently disabled.

Frequently asked questions

This section presents a list of the most frequently asked questions related to Foglight® federation:

- What versions of Foglight Management Server work together in a federation?
- Do multiple versions of a cartridge in a federation work?
- How much data is stored on the Federation Master?
- Where is the data stored in a federated environment?
- Can the customer mix repository types between Federated Child(ren) and a Federation Master? For example, can you federate an environment using three Management Servers with Oracle databases, with a Management Server (with a MySQL database) as a Federation Master?
- Is it possible to mix Foglight installations on Windows and Unix systems with Oracle and MySQL databases in a federated environment?
- Is the Federation Master able to consolidate data in its dashboards from separate Management Servers?
- Does the Federation Master have any High Availability capabilities, to prevent it from being a single point of failure?
- Can all federation members be a High Availability installation?

- Can a Federation Master also be a regular server at the same time?
- What is the repository sizing requirement for the Federation Master? If the Federation Master is pulling data from individual Foglight Management Servers, does it mean the amount of data kept in the Federation Master repository is low?
- What is federated in 8.0.0? Is it only metrics and topology objects? What about things like agent parameters?
- Is ASP (Agent Startup Parameters) data shared between the Federation Master and Federated Children?
- Is the status shown on the Federation Master browser interface in real time? What is the minimum configurable refresh interval?
- Is a persistent connection maintained between the Federation Master and the Federated Child?
- Are there any concerns regarding the time synchronization between the Federation Master and Federated Children?
- How can you force a topology refresh, to ensure that the Federation Master is up-to-date?
- If a user exists on the Federation Master, but not on the Federated Child, and that user acknowledges an alarm, will the username on the Federated Child understand to take the name from the Federation Master?
- How quickly should the Federation Master reflect changes on the Federated Children?
- Can the Federation Master continue to monitor the environment if the link between the Federation Master and Federated Children goes down?
- What happens when a value in the federation.config file is overridden at runtime, then, while the Management Server is still running, the federation.config file is modified again, changing the value of the property originally overridden?
- What is stored in the federation database?

What versions of Foglight Management Server work together in a federation?

For a complete list of versions supported in this and previous versions of the Federation Master, see Requirements.

For optimal results, your Federation Master and Federated Child(ren) should be the same version.

NOTE: If a Federated Child is an earlier version than your Federation Master, performance may not be optimal.

Do multiple versions of a cartridge in a federation work?

This theory is under Quality Assurance testing and has not been validated yet. Therefore, this release imposes a strict rule that the cartridge versions need to be the same.

How much data is stored on the Federation Master?

The Federation Master stores about 15% of what a Management Server stores.

NOTE: This value is an estimate. The data ratio may change over time.

Where is the data stored in a federated environment?

The Federation Master receives data on demand from its clients. The data is only cached in the memory (not stored in the Federation Master's repository), which ensures that similar retention policies are used for the Federation Master and Federated Children.

Can the customer mix repository types between Federated Child(ren) and a Federation Master? For example, can you federate an environment using three

Management Servers with Oracle databases, with a Management Server (with a MySQL database) as a Federation Master?

Yes, running different databases between Federated Child(ren) and a Federation Master is a supported configuration.

Is it possible to mix Foglight installations on Windows and Unix systems with Oracle and MySQL databases in a federated environment?

Mixing different platforms between Federation Master and Federated Children is allowed and supported. The only requirement is to install the same Management Server version on all servers that are part of the federated environment. The Federation Master does not store a lot of data, so having the Federation Master running with an embedded or external MySQL database is supported as well.

Is the Federation Master able to consolidate data in its dashboards from separate Management Servers?

Yes, it is able to do so. The Federation Master synchronizes all of the models across all Federated Children. When a dashboard is deployed, it operates against the federated model. For data, the Federation Master handles sending queries to each of the Federated Children at the time the query is made. Therefore, you can use a browser on the Federation Master to display all the information. The Federation Master does not have a heavy data load because it sends queries to each of the Federated Children.

Does the Federation Master have any High Availability capabilities, to prevent it from being a single point of failure?

A Management Server can act as either a Federated Child or a Federation Master—the code base is the same, just the configuration is different.

A Management Server has built-in High Availability mode, and the Management Server cluster can perform a hot failover. Agents have a propagation mechanism that allows them to redirect to the hot failover server in High Availability mode.

In federation, multiple High Availability Federated Children can be added to the list of servers in *federation.config*. If High Availability makes one Federated Child become the primary server of a cluster, the Federation Master automatically receives data from this new primary server.

NOTE: If the Federation Master has the addresses of all Federated Children (both primary and standby) configured in federation.config, it is able to redirect requests appropriately (some data may be unavailable for short periods of time when Federated Children terminate).

Can all federation members be a High Availability installation?

Federated Children can run in High Availability mode, Federation Masters cannot.

CAUTION: Customers are advised to configure nothing on the Federation Master until it can run in High Availability mode. Failing to do so may result in data loss.

Can a Federation Master also be a regular server at the same time?

No, the Federation Master does not accept agent connections (see details on the earlier FAQ Does the Federation Master have any High Availability capabilities, to prevent it from being a single point of failure?).

[Hide the following information until we have a formula for calculating the repository sizing]

What is the repository sizing requirement for the Federation Master? If the Federation Master is pulling data from individual Foglight Management Servers,

does it mean the amount of data kept in the Federation Master repository is low?

Size of the Federation Master database is based upon the size of the federated model, not the amount of collected data from Federated Children. The repository size is generally smaller for the Federation Master.

What is federated in 8.0.0? Is it only metrics and topology objects? What about things like agent parameters?

All the functionality in the Foglight browser interface is federated, with the exception of the administrative functions (for example, the agent management).

Is ASP (Agent Startup Parameters) data shared between the Federation Master and Federated Children?

The agent property data (previously known as ASP data) is not shared between Federated Children and the Federation Master:

- The Federation Master gathers and stores a unified topology (for example, host names, database names, service models) model from the Federated Children.
- At the database layer, the Federation Master is only configured to talk to its repository (not to the repositories of the Federated Children).
- All agent administration activities are specific to the instance to which they are bound (for example, the Federated Child agent properties cannot be altered from the Federation Master).
- The Federation Master queries the Federated Child on periodic intervals (for state and topology updates), and on demand for operational UI requests.
- The Federation Master persists:
  - Unified topology
  - Time-driven derived metrics from the Federation Master
  - Alarms generated on the Federation Master from time-driven rules

Is the status shown on the Federation Master browser interface in real time? What is the minimum configurable refresh interval?

The status shown on the Federation Master browser interface is as up-to-date as the status shown on any other Management Server. The Federation Master pulls data from the Federated Children on request (for example, when a user clicks a dashboard), or at the refresh interval that is associated with a given dashboard/view.

WCF (Web Component Framework) lets you configure refresh intervals. The minimum refresh interval is one second.

NOTE: Alarms are subject to some delay due to caching (although they are generally pulled on demand), as described in section Understanding federation communication channels.

Is a persistent connection maintained between the Federation Master and the Federated Child?

No, it is not. New connections are created as needed. If there is no activity for a while, older connections may be closed.

Are there any concerns regarding the time synchronization between the Federation Master and Federated Children?

Yes, this is a concern. Ideally the time must be kept synchronized via some system service (such as NTP). If the time on the two types of servers is not synchronized, requests for data are made based on the Federation Master

time, but serviced based on the Federated Child time (for example, data shown as 11:30 may in fact be 11:45, or 11:20).

How can you force a topology refresh, to ensure that the Federation Master is up-to-date?

To force a topology refresh:

1. Set `TopologyRefreshPeriod=1` (in the *federation.config* file).
2. Wait for one minute.
3. Reset `TopologyRefreshPeriod` to its old value.
4. Wait for the refresh to finish.

IMPORTANT: Refresh time depends on the environment complexity.

To force the topology refresh from the JMX:

- Invoke `refreshAll()` on the *TopologySynchService MBean*.

IMPORTANT: No configuration changes are necessary in this case.

If a user exists on the Federation Master, but not on the Federated Child, and that user acknowledges an alarm, will the username on the Federated Child understand to take the name from the Federation Master?

If you have set up token exchange security between the Federation Master and the Federated Child, the username on the Federated Child will understand to take the name from the Federation Master. For instructions on how to set up token exchange security, see Security settings.

If you have not set up token exchange security between the Federation Master and the Federated Child, the acknowledge operation fails immediately, and a security error appears on the Federation Master. Permissions are not enforced for alarm clear or acknowledge operations, just user authentication (the same as in the non-federation case).

How quickly should the Federation Master reflect changes on the Federated Children?

Topology changes made on the Federated Children are visible on the Federation Master with a delay defined by the `TopologyRefreshPeriod` parameter in the *federation.config* file. Alarms are treated differently than topology changes, and their refresh frequency is defined by the `MaxAlarmUpdateDelay` parameter in the *federation.config* file.

Can the Federation Master continue to monitor the environment if the link between the Federation Master and Federated Children goes down?

When the link between the Federation Master and a Federated Child is interrupted, then the Federation Master has no access to data on that Federated Child. If several Federated Children are connected, and one link to one of them goes down, the Federation Master continues to receive data from the connected servers.

What happens when a value in the federation.config file is overridden at runtime, then, while the Management Server is still running, the federation.config file is modified again, changing the value of the property originally overridden?

The Management Server reloads the *federation.config* periodically, and all changes take effect upon reload (by default, once a minute).

You can override any property in the *server.config* using the standard system properties methods that are recognized by the common code. If a property from *server.config* is overridden via a system property, then the override holds even if the configuration is modified at runtime.

What is stored in the federation database?

The only thing stored on the Federation Master is the combined topology. All alarms and data are pulled on demand (that is, they are not stored). The Federation Master-side derived metrics and local alarms are stored as usual.

About Us

We are more than just a name

We are on a quest to make your information technology work harder for you. That is why we build community-driven software solutions that help you spend less time on IT administration and more time on business innovation. We help you modernize your data center, get you to the cloud quicker and provide the expertise, security and accessibility you need to grow your data-driven business. Combined with Quest's invitation to the global community to be a part of its innovation, and our firm commitment to ensuring customer satisfaction, we continue to deliver solutions that have a real impact on our customers today and leave a legacy we are proud of. We are challenging the status quo by transforming into a new software company. And as your partner, we work tirelessly to make sure your information technology is designed for you and by you. This is our mission, and we are in this together. Welcome to a new Quest. You are invited to Join the Innovation™.

Our brand, our vision. Together.

Our logo reflects our story: innovation, community and support. An important part of this story begins with the letter Q. It is a perfect circle, representing our commitment to technological precision and strength. The space in the Q itself symbolizes our need to add the missing piece—you—to the community, to the new Quest.

Contacting Quest

For sales or other inquiries, visit www.quest.com/contact.

Technical support resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at https://support.quest.com.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request.
- View Knowledge Base articles.
- Sign up for product notifications.
- Download software and technical documentation.
- View how-to-videos.
- Engage in community discussions.
- Chat with support engineers online.
- View services to assist you with your product.