HCL Informix 15.0.0

InformixHQ

Contents

Chapter 1. InformixHQ Guide

InformixHQ is a modern web console for visualizing, monitoring, and managing your Informix server instances. It is purpose built for ease-of-use, scaling out, and optimizing DevOps needs. It provides critical performance management capabilities, monitoring how key performance metrics are changing over time and tracking how efficiently Informix is running your workload even when you've stepped away from your screen. Its monitoring system feeds directly into a customizable alerting system so you can be immediately alerted via email, Twilio, or PagerDuty whenever an issue occurs on one of your Informix database server instances. InformixHQ is designed to be scalable to efficiently manage and monitor as many Informix database server instances as you need. Moreover, it's a tool that can be shared by the DBAs, the app developers, the ops engineers, and management and accessed from any desktop, laptop, or mobile device. InformixHQ is the centralized hub for graphical monitoring, alerting, and administration of your Informix database servers.

What's new in InformixHQ

This topic includes information about new features in InformixHQ.

What's new in InformixHQ 3.0.3

- Administration on page 3
- Ease of use on page 4
- System Report Enhancements on page 4

What's new in InformixHQ 3.0.2

- Administration on page 4

What's new in InformixHQ 3.0.0

- Administration on page 4
- Ease of use on page 4

What's new in InformixHQ 3.0.3

Administration

- **Default Role Handling Enhancement**: Modifying a user's privilege from CONNECT to RESOURCE will no longer delete the default role unexpectedly.

Ease of use

- **Task Cleanup Enhancement**: The cleanup process now effectively deletes orphaned `ph_task` entries that does not have the corresponding records in `command_history`.
- **Space Usability Optimization** : InformixHQ 3.0.3 has been improved to preserve the search input under Tables & Indexes when any Optimize Space option is executed, eliminating the need to re-enter the search.
- **Schema Manager Enhancement**: InformixHQ 3.0.3 provides the ability to create an index when the database has a different owner.

System Report Enhancements

- Sequential scan is improved to properly populate pages when executed.
- An option has been added to show or hide system tables in the Table Extents report, allowing for easier focus on user-defined tables.

What's new in InformixHQ 3.0.2

Administration

- **SSL connection issue addressed**: InformixHQ 3.0.2 addresses an Informix server SSL connection failure issue (seen with IBM Java).

What's new in InformixHQ 3.0.0

Administration

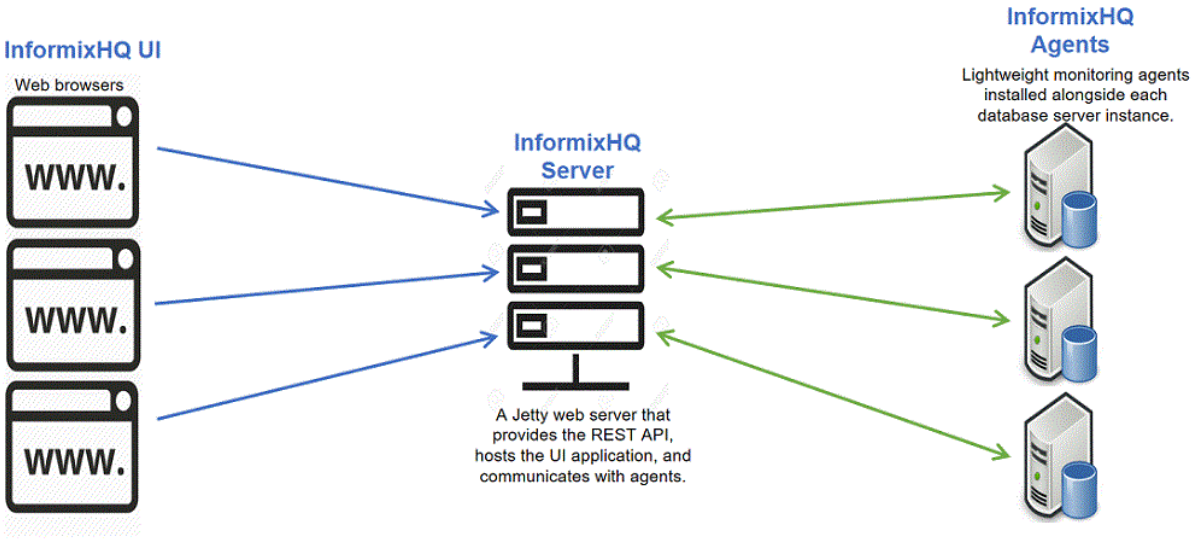
- **Java 11 or Higher Requirement**: InformixHQ 3.0.0 (both the server and agent) now mandates **Java 11** or higher for proper functionality.
- **Account Lockout Policy**: To enhance security, InformixHQ 3.0.0 now implements an account lockout policy. User accounts will be automatically locked after **5 consecutive failed login attempts**.

Ease of use

- **Support for Large Tables**: For Informix 15.0.0 or later, InformixHQ now supports the creation of **large tables** through the "**Create Table**" option. Additionally, users can view whether a table is categorized as **SMALL** or **LARGE** on the table information page.
- **Dropping Constraints**: InformixHQ now provides the ability to **drop constraints** directly from the user interface.
- **Server Refresh Functionality**: Users can now **refresh** specific servers or multiple servers simultaneously within InformixHQ. This feature allows loading the most recent information from the selected servers.

Architecture

InformixHQ consists of three distinct pieces that come together to give you a comprehensive monitoring and administering experience for Informix.



InformixHQ Server

- Java 11 based Jetty web server
- Monitors and administers multiple Informix database servers
- Connects directly to Informix databases servers to:
    - Gather live data from the system
    - Perform administration
- Connects to the InformixHQ agents regarding:
    - Monitored data
    - Alerts/events

InformixHQ Agent

- Lightweight Java 11 based monitoring agent
- Installed alongside each of your Informix database instances
- Only needs read access to database server
- Can perform native command execution to gather OS statistics as well as database statistics

InformixHQ User Interface (UI)

- Modernized web UI for monitoring, managing, visualizing, and assessing your Informix database servers

System Compatibility

Before you install InformixHQ, make sure that your computer meets the system requirements.

InformixHQ Prerequisites

The following table lists the software prerequisites for InformixHQ.

Software	Required Version
Informix Database Server	12.10 or higher
Java	11 or higher

Note:

In Debian environment, InformixHQ requires "haveged" service to be up and running. To start the service, follow these steps:

- sudo apt-get install haveged
- update-rc.d haveged defaults
- service haveged start

Java flavor

Requires Java version 11 or higher. InformixHQ supports all the flavors of Java version 11 on all platforms.

The following table lists the Java flavors and versions that InformixHQ has been tested with:

Java flavors	Tested on version	Tested on platform
IBM Java	17.0.6	AIX, Linux, Windows
Open JDK	17.0.12	Linux, Windows

Secured connection

1. **With Informix server**

   When connecting to Informix server having SSL connection in InformixHQ, following keystore formats are compatible:

Keystore type	Extensions
Java KeyStore	.keystore, .jks
PKCS KeyStore	.p12

   Note: GSKit version 8.0.55.17 available with Informix server bundle is used for creating p12 keystore.

   The recommended keystore is p12 created using GSKit.

2. **Within InformixHQ**

   Keystores supported for HTTPS connection:

   Case 1: Browser with InformixHQ Server.

   Case 2: InformixHQ Server and Agent.

Keystore type	Extensions
Java KeyStore	.jks

Supported Web Browsers

InformixHQ supports all the latest browsers. The following table lists the web browsers that InformixHQ has been tested with:

Web Browser	Version
Google Chrome	129.0.6668.90
Mozilla Firefox	126.0

Getting Started

This topic provides a brief tutorial to help you get started with InformixHQ.

Prerequisites

The following table lists the software prerequisites for InformixHQ.

Software	Required Version
Informix Database Server	12.10 or higher

Software	Required Version
Java	11 or higher

Starting InformixHQ

InformixHQ Server or Agent can be started using any of following methods:

- Java Command
- Script

Using Java Command

For more information about using Java Command, see Starting the InformixHQ Server on page 9 and Starting the InformixHQ Agent on page 10.

Using the Script:

The script to start the server or agent is in the form of *InformixHQ.bat* (Windows), *InformixHQ.sh* (Bash Shell for Linux) and *InformixHQ.ksh* (Korn shell for AIX) file on various operating system as per the requirement of the user.

Note: This script is indicative and users are free to modify/tune the scripts as per their requirements/environments.

Script Prerequisites

- Server and Agent jar file names should end with .jar and also must contain keyword "*informixhq*" in its name
- If using default filename option with scripts, jar and properties files with default names should be present in same folder as that of script
- User running the script must have read and write access to monitoring-server.log and monitoring-agent.log files
- If log files don't exist in the same folder from where HQ jar is running, HQ will create these files, so user running the script should have permissions to be able to create these files.
- Environment variable JAVA_HOME should be set correctly to JAVA 11 or higher and should be included in PATH variable.

This script support BASH shell on linux (.sh), KORN shell on AIX (.ksh) and command prompt on windows (.bat). This command starts/stops InformixHQ Server and Agent.

Syntax:

```
Command: InformixHQ
This command starts/stops InformixHQ Server and Agent.

Syntax:
InformixHQ startserver|startagent [encoding=<value>] [jvmmemx=<value>] [jarfile=<value>] [propfile=<value>]
InformixHQ stop <processid>
InformixHQ stopserver
InformixHQ stopagent [properties filename search string]
InformixHQ list [filename search string]
```

```
startserver          : Starts InformixHQ Server service
startagent           : Starts InformixHQ Agent service
stop                 : Stops InformixHQ Server/Agent service with processId
stopserver           : Stops InformiHQ Server process
stopagent            : Stops InformixQ Agent process (default all processes or filtered list if matching
 filename search string found)
list                 : Lists InformixHQ running processes (default all processes, filtered list if matching
 filename search string found)
encoding (Optional)  : Default value is utf-8
jvmmemx (Optional)   : JVM's default value will be used If not specified
jarfile (Optional)   : Default is informixhq-server.jar for startserver option and informixhq-agent.jar for
 startagent option.
                       For user provided filename, it must contain keyword 'informixhq' and it should end
 with .jar
propfile (Optional)  : Default is informixhq-server.properties for startserver option and
 informixhq-agent.properties for startagent option.
                       User can provide any custom name to properties file
```

For more information about using the script, see Starting the InformixHQ Server and the Agent on Windows on page 12 and Starting the InformixHQ Server and the Agent on Linux/AIX on page 16.

Starting the InformixHQ Server

This topic provides a brief tutorial to help you get started with InformixHQ Server.

1. Locate the *informixhq-server.jar* and the *monitoring-server.log4j.xml* file in the `$INFORMIXDIR/hq` directory of your Informix database server installation.
2. Create an InformixHQ server configuration on page 43 file. You can refer to the `$INFORMIXDIR/hq/monitoring-server-example.properties` file as an example.
3. Optionally, edit the *monitoring-server.log4j.xml* file to configure logging in the InformixHQ on page 32.

   Note: By default InformixHQ runs on 8080 port. For more information, see InformixHQ server configuration on page 43.

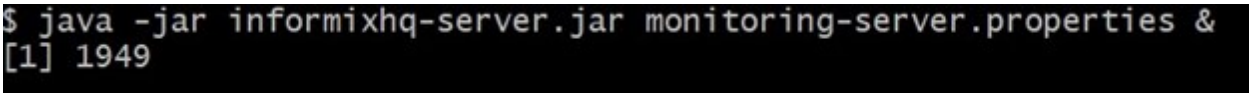
4. Start the InformixHQ server using the following command:

   ```
   java -jar informixhq-server.jar monitoring-server.properties &
   ```

   where **monitoring-server.properties** is the name of the InformixHQ server configuration file.

   Note: Staring from 14.10.xC4 onwards, you can start the server using an alternate method. For more information on this, see Starting the InformixHQ Server and the Agent on Windows on page and Starting the InformixHQ Server and the Agent on Linux on page .

   Sample output after starting the InformixHQ server with monitoring-server.log4j.xml:

When the InformixHQ server is started for the first time, a user with system administrator privileges for InformixHQ is created with the username and password as **admin**.

5. Using a web browser, go to the InformixHQ UI at `http://localhost:8080/` and login with the user and password **admin**. On first login, you will be prompted to reset your password.

   Note: We strongly recommend users to use a secured connection i.e. SSL while starting InformixHQ Server. To know how to configure it, see InformixHQ setup as Secured Server (SSL). on page 30

6. Once logged in, click **Add Server** to add an Informix server that you want to monitor.

Starting the InformixHQ Agent

This topic covers the two ways of starting the InformixHQ Agent.

To get the most out of InformixHQ, you should have an InformixHQ agent running for each Informix database server that you will be monitoring through the tool. While the InformixHQ agent is not required to view information about your database server in the InformixHQ UI, the agent is required if you want to gather monitoring data and configure alerts for that server.

There are two options for starting the InformixHQ agent. You can use the InformixHQ UI to automatically deploy and start the InformixHQ agent or you can manually start the InformixHQ agent on the command line.

Deploying and starting the InformixHQ agent automatically from the UI:

Before you deploy and start the InformixHQ agent automatically from the UI, you must meet the following prerequisites:

Prerequisites

1. SSH must be installed on the database server's host machine.
2. The *informixhq-agent.jar* file and the *informixhq-server.jar* file must be located in the same directory. For customized logging, the *monitoring-agent.log4j.xml* file must also be included in the same directory.

To deploy and start the InformixHQ agent automatically from the UI:

1. The Informix database server that the agent will be monitoring must first be defined in InformixHQ. If the database server has not been defined yet, in the UI, navigate to an InformixHQ group, click **Add Server** and define the server's connection properties.
2. Navigate to the Informix database server's **Setup** > **Agent** page and define a repository server and database. The repository database is where the monitored data will be stored.
3. From the server's **Setup > Agent** page, you can also try to deploy the agent with default configuration by clicking the **Deploy Agent** button. If the automatic deployment is successful, you are done. Otherwise, proceed to the next step.
4. Configure agent deployment:

- Username: remote user which will own and run the agent
- Password or Identity File/Passphrase: remote user's passphrase or remote user's identity file (e.g. private key) and its optional passphrase.
- Remote directory: Directory to deploy the agent files to. This directory will be created if it does not exist.

5. If SSL is enabled, keystore configuration may be required as well. If keystore configuration is not provided, InformixHQ will try to generate and deploy a keystore for you.
   - Keystore file: Location of an existing keystore on the remote machine (can be relative to the remote directory)
   - Keystore password: Password used to access the existing keystore
   - Keystore type: Existing keystore's type. Default is JKS
6. Click **Deploy Agent**
7. Once the agent is ready, use the InformixHQ UI in your web browser to configure the monitoring profile and alerts for this server.

Starting an InformixHQ agent manually on the command line:

1. The Informix database server that the agent will be monitoring must first be defined in InformixHQ. If the database server has not been defined yet, in the UI, navigate to an InformixHQ group, click **Add Server** and define the server's connection properties.
2. Navigate to the Informix database server's **Setup > Agent** page and define a repository server and database. The repository database is where the monitored data will be stored
3. Locate the *informixhq-agent.jar* and the *monitoring-agent.log4j.xml* file in the $INFORMIXDIR/hq directory of the InformixHQ database server installation.

   Note: Staring from 14.10.xC4 onwards, you can start the agent using an alternate method. For more information on this, see and Starting the InformixHQ Server and the Agent on Windows and Starting the InformixHQ Server and the Agent on Linux on page .

4. Copy the agent jar file and the log4j2 configuration file to the Informix database server host machine.
5. Create an agent configuration file.

   Sample agent configuration file:

```
## Key:         serverInstance.id
## Type:        integer
## Default:
## Description: Required. The ID of the Database Server defined in A&M Server
## that this Monitoring agent belongs to
#serverInstance.id=1

## Key:         server.host
## Type:        string
## Default:
## Description: Required. The host name or IP address on which the A&M
##   server is running.
#server.host=localhost
```

```
## Key:         server.port
## Type:        string
## Default:
## Description: Required. The port on which the A&M server is running.
#server.port=8080
```

Note: You can find the id of your Informix database server by navigating to the server's **Setup** page in the UI.

6. Optionally, edit the *monitoring-agent.log4j.xml* file to configure logging in the InformixHQ on page 32. If monitoring-agent.log4j.xml is not provided, default logging configuration is set and it gives warning as shown below.
7. Start the InformixHQ agent using the following command

```
java -jar informixhq-agent.jar monitoring-agent.properties &
```

where **monitoring-agent.properties** is the name of your InformixHQ agent configuration file.

Note: We strongly recommend users to use a secured connection i.e. SSL while starting InformixHQ agent. To know how to configure it, see InformixHQ Agent setup for secured connection (SSL) on page 31.

8. At this point the agent is ready and running. Use the InformixHQ UI in your web browser to configure the monitoring profile and alerts for this server.

Starting the InformixHQ Server and the Agent on Windows

This topic provides a brief tutorial about the script used to get started with InformixHQ Server and Agent on Windows.

1. Locate your *informixHQ.bat* for Windows (default path is %INFORMIXDIR%/hq).
2. Use command InformixHQ help to see sample syntax: e.g- InformixHQ.bat help.

Tip: *User can only run InformixHQ.bat for help menu*

Figure 1. InformixHQ help

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat
Command: InformixHQ
This command starts/stops InformixHQ Server and Agent.

Syntax:
    InformixHQ [startserver|startagent] [encoding=<value>] [jvmmemx=<value>] [jarfile=<value>] [propfile=<value>]
    InformixHQ [stop <processid>]
    InformixHQ [stopserver]
    InformixHQ [stopagent] [<properties filename search string>]
    InformixHQ [list] [<filename search string>]

    startserver             : Starts InformixHQ Server service
    startagent              : Starts InformixHQ Agent service
    stop                    : Stops InformixHQ Server/Agent with specified processid
    stopserver              : Stops InformiHQ Server process
    stopagent               : Stops InformixQ Agent process (default all processes or filtered list if matching filename search string found)
    list                    : Lists InformixHQ running processes (default all processes, filtered list if matching filename search string found)
    encoding (Optional)     : Default value is utf-8
    jvmmemx (Optional)      : JVM's default value will be used, if not specified
    jarfile (Optional)      : Default is informixhq-server.jar for startServer option and informixhq-agent.jar for startAgent option
                              For user provided filename, it must contain keyword 'informixhq' and it should end with .jar"
    propfile (Optional)     : Default is informixhq-server.properties for startServer option and informixhq-agent.properties for startAgent option
                              User can provide any custom name to properties file
```

3. Use command to start InformixHQ server with default file names. e.g- InformixHQ.bat startserver .

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat startserver
INFO: Please use list command to verify the process.
```

It will start a INFORMIXHQ Server with default values for all the parameters including jar file name *(informixhq-server.jar)* and properties filename *(informixhq-server.properties)*

4. Use command to start InformixHQ agent with default filenames. e.g- InformixHQ.bat startagent.

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat startagent
INFO: Please use list command to verify the process.
```

It will start a INFORMIXHQ Agent with default values for all the parameters including jar file name *(informixhq-agent.jar)* and properties filename *(informixhq-agent.properties)*.

5. List the Java processes running. It will list all the InformixHQ processes running with ProcessID e.g. – InformixHQ.bat list .

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat list
CommandLine                                                                    ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server-prod.properties  153308
java  -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server.properties       142952
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties    84388
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties         150032

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

6. Additionally user can filter the list by passing search keyword to the list command. e.g. – InformixHQ.bat list prod .

Note: It is recommended that properties file name should be unique which will help users to search specific processes.

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat list prod
CommandLine                                                                    ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server-prod.properties  153308
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties    84388

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

7. Stop the InformixHQ server or agent by providing processID to stop command. It will stop InformixHQ process with provided processID. e.g. –InformixHQ.bat stop 142952.

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat stop 142952
SUCCESS: The process with PID 142952 has been terminated.


C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat list
CommandLine                                                                    ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server-prod.properties  153308
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties    84388
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties         150032

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

8. stopserver command will stop the server processes. e.g. - InformixHQ.bat stopserver .

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat stopserver

INFO: This command has found 1 server process.
CommandLine                                                                  ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server-prod.properties  156572

Please Press:
0: exit
1: terminate
Please enter one of the above options [1/0]:1
SUCCESS: The process with PID 156572 has been terminated.
```

0. exit: It will exit from the command.

1. Terminate: It will terminate server process.

Note: Unless valid option is provided, it will not proceed further.

9. stopagent command will stop the agent processes. e.g. - InformixHQ.bat stopagent .

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat stopagent

INFO: This command has found 2 agent processes.
CommandLine                                                                ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties       145904
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties  125040

Please Press:
0: exit
1: terminate all
2: terminate one by one
Please enter one of the above options [1/2/0]:
```

0: exit - It will exit from the command.

1: Terminate all - It will terminate all the agent processes.

2: Terminate one by one - It will terminate agent processes one by one after user confirmation.

Note: Unless valid option is provided it will not proceed further.

Sample screenshot for 'terminate all processes' :

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat stopagent

INFO: This command has found 2 agent processes.
CommandLine                                                                ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties  84388
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties       150032

Please Press:
0: exit
1: terminate all
2: terminate one by one
Please enter one of the above options [1/2/0]:1
SUCCESS: The process with PID 84388 has been terminated.
SUCCESS: The process with PID 150032 has been terminated.
```

Sample screenshot for 'one-by-one process termination' :

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat stopagent

INFO: This command has found 2 agent processes.
CommandLine                                                                    ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties  149816
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties       153428

Please Press:
0: exit
1: terminate all
2: terminate one by one
Please enter one of the above options [1/2/0]:2

CommandLine                                                                    ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties  149816

Please press:
0: exit
1: terminate
2: move next
Please enter one of the above options [1/2/0]?:1
SUCCESS: The process with PID 149816 has been terminated.

CommandLine                                                                ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties  153428

Please press:
0: exit
1: terminate
2: move next
Please enter one of the above options [1/2/0]?:1
SUCCESS: The process with PID 153428 has been terminated.
```

Additionally, user can search specific agent process using stopagent command. e.g. - stopagent prod .

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat stopagent prod

INFO: This command has found 1 agent process.
CommandLine                                                                    ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties  153180

Please Press:
0: exit
1: terminate
Please enter one of the above options [1/0]:
```

Example

Following are sample commands where user can provide custom values (similar commands will work with agent)

- optional jar and properties filename to start InformixHQ server:

```
InformixHQ.bat startserver jarfile=informixhq-myinstance1.jar propfile=hqserver.properties
```

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat startserver jarfile=informixhq-myinstance1.jar propfile=hqserver.properties
INFO: Please use list command to verify the process.


C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat list
CommandLine                                                          ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-myinstance1.jar hqserver.properties  26864

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

- Use encoding and JVM value:

```
InformixHQ startserver Encoding=utf-8 JVMMemx=1004m
```

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ startserver Encoding=utf-8 JVMMemx=1004m
INFO: Please use list command to verify the process.


C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat list
CommandLine                                                                    ProcessId
java  -Xmx1004m -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server.properties  26448

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

- Use all input:

```
InformixHQ.bat startserver Encoding=utf-8 JVMMemx=1004m jarfile=informixhq-server.jar
 propfile=hqserver.properties
```

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat startserver Encoding=utf-8 JVMMemx=1004m jarfile=informixhq-server.jar propfile=hqserver.properties
INFO: Please use list command to verify the process.


C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat list
CommandLine                                                    ProcessId
java  -Xmx1004m -Dfile.encoding=utf-8 -jar informixhq-server.jar hqserver.properties  3668

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

Starting the InformixHQ Server and the Agent on Linux/AIX

This topic provides a brief tutorial about the script used to get started with InformixHQ Server and Agent on Linux/AIX.

1. Locate your *informixHQ.sh* for Linux and *InformixHQ.ksh* for AIX (default path is ${INFORMIXDIR}/hq).

   Note: Following examples are shown on bash shell, however those will run exactly same on korn shell with .ksh script.

2. Use command **InformixHQ help** to see sample syntax: e.g- InformixHQ.sh help .

   Tip: *User can only run InformixHQ.sh for help menu*

   Figure 2. InformixHQ help

```
[root@test-server hq]# ./InformixHQ.sh

Command: InformixHQ
This command starts/stops InformixHQ Server and Agent.

Syntax:
InformixHQ startserver|startagent [encoding=<value>] [jvmmemx=<value>] [jarfile=<value>] [propfile=<value>]
InformixHQ stop <processid>
InformixHQ stopserver
InformixHQ stopagent [properties filename search string]
InformixHQ list [filename search string]

startserver          : Starts InformixHQ Server service
startagent           : Starts InformixHQ Agent service
stop                 : Stops InformixHQ Server/Agent service with processId
stopserver           : Stops InformiHQ Server process
stopagent            : Stops InformixQ Agent process (default all processes or filtered list if matching filename search string found)
list                 : Lists InformixHQ running processes (default all processes, filtered list if matching filename search string found)
encoding (Optional)  : Default value is utf-8
jvmmemx (Optional)   : JVM's default value will be used If not specified
jarfile (Optional)   : Default is informixhq-server.jar for startserver option and informixhq-agent.jar for startagent option.
                       For user provided filename, it must contain keyword 'informixhq' and it should end with .jar
propfile (Optional)  : Default is informixhq-server.properties for startserver option and informixhq-agent.properties for startagent option.
                       User can provide any custom name to properties file
```

3. Use command InformixHQ startserver to start InformixHQ server with default file names. e.g- ./InformixHQ.sh startserver .

```
[root@test-server hq]# ./InformixHQ.sh startserver
INFO: Please use list command to verify the process.
```

It will start INFORMIXHQ Server with default values for all the parameters including jar file name (*informixhq-server.jar*) and properties filename (*informixhq-server.properties*).

4. Use command InformixHQ startagent to start InformixHQ agent with default filenames. e.g- ./InformixHQ.sh startagent

```
[root@test-server hq]# ./InformixHQ.sh startagent
INFO: Please use list command to verify the process.
```

It will start an INFORMIXHQ Agent with default values for all the parameters including jar file name (informixhq-agent.jar) and properties filename (informixhq-agent.properties).

5. List the Java processes running. It will list all the InformixHQ processes running with ProcessID e.g: ./InformixHQ.sh list.

```
[root@test-server hq]# ./InformixHQ.sh list

   PID USER     COMMAND
 182725 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server.properties
 183016 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server-prod.properties
 183230 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties
 183294 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

6. Additionally user can filter the list by passing search keyword to the list command. e.g. – InformixHQ.bat list prod .

Note: It is recommended that properties file name should be unique which will help users to search specific processes.

```
[root@test-server hq]# ./InformixHQ.sh list prod

   PID USER     COMMAND
 183016 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server-prod.properties
 183230 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

7. Stop the InformixHQ server or agent by providing processID to stop command. It will stop InformixHQ process with provided processID. e.g: ./InformixHQ.sh stop 183230 .

```
[root@test-server hq]# ./InformixHQ.sh stop 183230

SUCCESS: The process with PID 183230 has been terminated.

[root@test-server hq]# ./InformixHQ.sh list

   PID USER     COMMAND
 182725 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server.properties
 183016 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server-prod.properties
 183294 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

8. stopserver command will stop the server processes. e.g. - InformixHQ.bat stopserver .

```
[root@test-server hq]# ./InformixHQ.sh stopserver

INFO: This command has found 1 server process(es).

    PID USER     COMMAND
 183016 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server-prod.properties

Please Press:
0: exit
1: terminate

Please enter one of the above options [0/1]:1

SUCCESS: The process with PID 183016 has been terminated.
```

0. exit: It will exit from the command.

1. Terminate: It will terminate server process.

Note: Unless valid option is provided, it will not proceed further.

9. stopagent command will stop the agent processes. e.g. - InformixHQ.bat stopagent .

```
[root@test-server hq]# ./InformixHQ.sh stopagent

INFO: This command has found 2 agent process(es).

    PID USER     COMMAND
 184138 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties
 184281 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties

Please Press:
0: exit
1: terminate all
2: terminate one by one

Please enter one of the above options [0/1/2]:
```

0: exit - It will exit from the command.

1: Terminate all - It will terminate all the agent processes.

2: Terminate one by one - It will terminate agent processes one by one after user confirmation.

Note: Unless valid option is provided, it will not proceed further.

Sample screenshot for 'terminate all processes' :

```
[root@test-server hq]# ./InformixHQ.sh stopagent

INFO: This command has found 2 agent process(es).

    PID USER     COMMAND
 184138 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties
 184281 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties

Please Press:
0: exit
1: terminate all
2: terminate one by one

Please enter one of the above options [0/1/2]:1

SUCCESS: The process with PID 184138 has been terminated.
SUCCESS: The process with PID 184281 has been terminated.
```

Sample screenshot for 'one-by-one process termination' :

```
[root@test-server hq]# ./InformixHQ.sh stopagent

INFO: This command has found 2 agent process(es).

    PID USER     COMMAND
 184514 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties
 184564 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties

Please Press:
0: exit
1: terminate all
2: terminate one by one

Please enter one of the above options [0/1/2]:2

    PID USER     COMMAND
 184514 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties

Please press:
0: exit
1: terminate
2: move next

Please enter one of the above options [0/1/2]:1

SUCCESS: The process with PID 184514 has been terminated.

    PID USER     COMMAND
 184564 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties

Please press:
0: exit
1: terminate
2: move next

Please enter one of the above options [0/1/2]:1

SUCCESS: The process with PID 184564 has been terminated.
```

Additionally, user can search specific agent process using stopagent command. e.g. - stopagent prod .

```
[root@test-server hq]# ./InformixHQ.sh stopagent prod

INFO: This command has found 1 agent process(es).

    PID USER     COMMAND
 184564 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties

Please Press:
0: exit
1: terminate

Please enter one of the above options [0/1]:
```

Example

User can use different properties filenames (with appropriate informixServer.id for different servers) to run agents for different instances as follows

Following are sample commands where user can provide custom values (similar commands will work with agent)

- optional jar and properties filename to start InformixHQ server:

```
./InformixHQ.sh startserver jarfile=informixhq-myinstance1.jar propfile=hqserver.properties
```

```
[root@test-server hq]# ./InformixHQ.sh startserver jarfile=informixhq-myinstance1.jar propfile=hqserver.properties

INFO: Please use list command to verify the process.

[root@test-server hq]# ./InformixHQ.sh list
ProcessId  Name  Jar  Properties
112911 java informixhq-myinstance1.jar hqserver.properties

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

- Use encoding and JVM value:

```
./InformixHQ.sh startserver Encoding=utf-8 JVMMemx=1004m
```

```
[root@test-server hq]# ./InformixHQ.sh startserver encoding=utf-8 jvmmemx=1004m

INFO: Please use list command to verify the process.
```

- Use all input:

```
InformixHQ.bat startserver Encoding=utf-8 JVMMemx=1004m jarfile=informixhq-server.jar
 propfile=hqserver.properties
```

```
[root@test-server hq]# ./InformixHQ.sh startserver encoding=utf-8 jvmmemx=1004m jarfile=informixhq-server.jar propfile=hqserver.properties

INFO: Please use list command to verify the process.
```

```
[root@test-server hq]# ./InformixHQ.sh list
ProcessId  Name  Jar  Properties
114005 java informixhq-server.jar hqserver.properties

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

Exception Reporting

If options are not provided correctly then script will report error as follows

- If jarfile/propfile option is not specified

```
[root@test-server hq]# ./InformixHQ.sh startserver abc.jar
ERROR: The syntax of the commmand is incorrect.
Invalid Parameters :  [abc.jar,]

INFO: Use command InformixHQ.sh help to see the help.
```

- filename is not specified with jarfile/propfile option

```
[root@test-server hq]# ./InformixHQ.sh startserver jarfile
ERROR: The syntax of the commmand is incorrect.
Invalid Parameters :  [jarfile,]

INFO: Use command InformixHQ.sh help to see the help.
```

- incorrect filename provided

```
[root@test-server hq]# ./InformixHQ.sh startserver jarfile=abc.jar
ERROR: abc.jar doesn't exist.
```

- If specified or default files are not found also if custom filename is specified and file is not found in the folder, it will inform accordingly

```
[root@test-server hq]# ./InformixHQ.sh startagent jarfile=informixhq-inst1.jar propfile=agent-inst1.properties
ERROR: informixhq-inst1.jar doesn't exist.

ERROR: agent-inst1.properties doesn't exist.
```

- If IDS Server is not reachable by Agent. This is not related to script, however sometimes you might get exception if Server or Agent Java process encountered error. While starting agent process, if agent is not able to connect to IDS server specified in HQ then it might show error on screen as follows with message. Socket connection to server (Server IP:Port) failed. Check your server is reachable from this client on the host:port specified.

```
[root@test-server hq]# ./InformixHQ.sh startagent

INFO: Please use list command to verify the process.

[root@test-server hq]# Exception in thread "ServerStatusMonitor" com.zaxxer.hikari.pool.HikariPool$PoolInitializationException: Failed to initialize pool: Socket connection to server (192.1
68.150.130:16761) failed. Check your server is reachable from this client on the host:port specified.
        at com.zaxxer.hikari.pool.HikariPool.throwPoolInitializationException(HikariPool.java:576)
        at com.zaxxer.hikari.pool.HikariPool.checkFailFast(HikariPool.java:562)
        at com.zaxxer.hikari.pool.HikariPool.<init>(HikariPool.java:115)
        at com.zaxxer.hikari.HikariDataSource.<init>(HikariDataSource.java:81)
        at com.informix.hq.agent.InformixConnectionManager.newConnectionPool(InformixConnectionManager.java:88)
        at com.informix.hq.agent.InformixConnectionManager.getTargetConnection(InformixConnectionManager.java:130)
        at com.informix.hq.agent.InformixServerStatusMonitor.checkStatusOnJdbcConnection(InformixServerStatusMonitor.java:151)
        at com.informix.hq.agent.InformixServerStatusMonitor.run(InformixServerStatusMonitor.java:80)
Caused by: java.sql.SQLException: Socket connection to server (192.168.150.130:16761) failed. Check your server is reachable from this client on the host:port specified.
        at com.informix.jdbc.IfxSqliConnect.<init>(IfxSqliConnect.java:1525)
        at com.informix.jdbc.IfxDriver.connect(IfxDriver.java:169)
        at com.zaxxer.hikari.util.DriverDataSource.getConnection(DriverDataSource.java:136)
        at com.zaxxer.hikari.pool.PoolBase.newConnection(PoolBase.java:369)
        at com.zaxxer.hikari.pool.PoolBase.newPoolEntry(PoolBase.java:198)
        at com.zaxxer.hikari.pool.HikariPool.createPoolEntry(HikariPool.java:467)
        at com.zaxxer.hikari.pool.HikariPool.checkFailFast(HikariPool.java:541)
        ... 6 more
Caused by: com.informix.asf.IfxASFException: Socket connection to server (192.168.150.130:16761) failed. Check your server is reachable from this client on the host:port specified.
        at com.informix.util.IfxErrMsg.getLocIfxASFException(IfxErrMsg.java:585)
        at com.informix.asf.Connection.openSocket(Connection.java:1695)
        at com.informix.asf.Connection.connect(Connection.java:333)
        at com.informix.asf.Connection.<init>(Connection.java:327)
        at com.informix.jdbc.IfxSqliConnect.<init>(IfxSqliConnect.java:1250)
        ... 12 more
Caused by: java.net.ConnectException: Connection refused (Connection refused)
        at java.net.PlainSocketImpl.socketConnect(Native Method)
        at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
        at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206)
        at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:188)
        at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
        at java.net.Socket.connect(Socket.java:607)
        at com.informix.asf.Connection.openSocket(Connection.java:1661)
        ... 15 more
```

In such case, ensure, server added in InformixHQ server is configured correctly and is reachable and online.

Starting the InformixHQ Server and the Agent on Windows

This topic provides a brief tutorial about the script used to get started with InformixHQ Server and Agent on Windows.

1. Locate your *informixHQ.bat* for Windows (default path is %INFORMIXDIR%/hq).
2. Use command InformixHQ help to see sample syntax: e.g- InformixHQ.bat help.

> **Tip:** *User can only run InformixHQ.bat for help menu*

Figure 3. InformixHQ help

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat
Command: InformixHQ
This command starts/stops InformixHQ Server and Agent.

Syntax:
    InformixHQ [startserver|startagent] [encoding=<value>] [jvmmemx=<value>] [jarfile=<value>] [propfile=<value>]
    InformixHQ [stop <processid>]
    InformixHQ [stopserver]
    InformixHQ [stopagent] [<properties filename search string>]
    InformixHQ [list] [<filename search string>]

    startserver             : Starts InformixHQ Server service
    startagent              : Starts InformixHQ Agent service
    stop                    : Stops InformixHQ Server/Agent with specified processid
    stopserver              : Stops InformiHQ Server process
    stopagent               : Stops InformixQ Agent process (default all processes or filtered list if matching filename search string found)
    list                    : Lists InformixHQ running processes (default all processes, filtered list if matching filename search string found)
    encoding (Optional)     : Default value is utf-8
    jvmmemx (Optional)      : JVM's default value will be used, if not specified
    jarfile (Optional)      : Default is informixhq-server.jar for startServer option and informixhq-agent.jar for startAgent option
                              For user provided filename, it must contain keyword 'informixhq' and it should end with .jar"
    propfile (Optional)     : Default is informixhq-server.properties for startServer option and informixhq-agent.properties for startAgent option
                              User can provide any custom name to properties file
```

3. Use command to start InformixHQ server with default file names. e.g- InformixHQ.bat startserver .

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat startserver
INFO: Please use list command to verify the process.
```

It will start a INFORMIXHQ Server with default values for all the parameters including jar file name *(informixhq-server.jar)* and properties filename *(informixhq-server.properties)*

4. Use command to start InformixHQ agent with default filenames. e.g- InformixHQ.bat startagent.

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat startagent
INFO: Please use list command to verify the process.
```

It will start a INFORMIXHQ Agent with default values for all the parameters including jar file name *(informixhq-agent.jar)* and properties filename *(informixhq-agent.properties)*.

5. List the Java processes running. It will list all the InformixHQ processes running with ProcessID e.g. – InformixHQ.bat list .

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat list
CommandLine                                                                    ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server-prod.properties  153308
java  -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server.properties       142952
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties    84388
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties         150032

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

6. Additionally user can filter the list by passing search keyword to the list command. e.g. – InformixHQ.bat list prod .

Note: It is recommended that properties file name should be unique which will help users to search specific processes.

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat list prod
CommandLine                                                                    ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server-prod.properties  153308
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties    84388

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

7. Stop the InformixHQ server or agent by providing processID to stop command. It will stop InformixHQ process with provided processID. e.g. –InformixHQ.bat stop 142952.

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat stop 142952
SUCCESS: The process with PID 142952 has been terminated.


C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat list
CommandLine                                                                    ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server-prod.properties  153308
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties    84388
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties         150032

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

8. stopserver command will stop the server processes. e.g. - InformixHQ.bat stopserver .

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat stopserver

INFO: This command has found 1 server process.
CommandLine                                                                    ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server-prod.properties  156572

Please Press:
0: exit
1: terminate
Please enter one of the above options [1/0]:1
SUCCESS: The process with PID 156572 has been terminated.
```

0. exit: It will exit from the command.

1. Terminate: It will terminate server process.

Note: Unless valid option is provided, it will not proceed further.

9. stopagent command will stop the agent processes. e.g. - InformixHQ.bat stopagent .

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat stopagent

INFO: This command has found 2 agent processes.
CommandLine                                                                    ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties       145904
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties  125040

Please Press:
0: exit
1: terminate all
2: terminate one by one
Please enter one of the above options [1/2/0]:
```

0: exit - It will exit from the command.

1: Terminate all - It will terminate all the agent processes.

2: Terminate one by one - It will terminate agent processes one by one after user confirmation.

Note: Unless valid option is provided it will not proceed further.

Sample screenshot for 'terminate all processes' :

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat stopagent

INFO: This command has found 2 agent processes.
CommandLine                                                                    ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties  84388
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties       150032

Please Press:
0: exit
1: terminate all
2: terminate one by one
Please enter one of the above options [1/2/0]:1
SUCCESS: The process with PID 84388 has been terminated.
SUCCESS: The process with PID 150032 has been terminated.
```

Sample screenshot for 'one-by-one process termination' :

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat stopagent

INFO: This command has found 2 agent processes.
CommandLine                                                                    ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties  149816
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties       153428

Please Press:
0: exit
1: terminate all
2: terminate one by one
Please enter one of the above options [1/2/0]:2

CommandLine                                                                    ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties  149816

Please press:
0: exit
1: terminate
2: move next
Please enter one of the above options [1/2/0]?:1
SUCCESS: The process with PID 149816 has been terminated.

CommandLine                                                                ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties  153428

Please press:
0: exit
1: terminate
2: move next
Please enter one of the above options [1/2/0]?:1
SUCCESS: The process with PID 153428 has been terminated.
```

Additionally, user can search specific agent process using stopagent command. e.g. - stopagent prod .

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat stopagent prod

INFO: This command has found 1 agent process.
CommandLine                                                                    ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties  153180

Please Press:
0: exit
1: terminate
Please enter one of the above options [1/0]:
```

Example

Following are sample commands where user can provide custom values (similar commands will work with agent)

- optional jar and properties filename to start InformixHQ server:

```
InformixHQ.bat startserver jarfile=informixhq-myinstance1.jar propfile=hqserver.properties
```

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat startserver jarfile=informixhq-myinstance1.jar propfile=hqserver.properties
INFO: Please use list command to verify the process.


C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat list
CommandLine                                                          ProcessId
java  -Dfile.encoding=utf-8 -jar informixhq-myinstance1.jar hqserver.properties  26864

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

- Use encoding and JVM value:

```
InformixHQ startserver Encoding=utf-8 JVMMemx=1004m
```

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ startserver Encoding=utf-8 JVMMemx=1004m
INFO: Please use list command to verify the process.


C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat list
CommandLine                                                                    ProcessId
java  -Xmx1004m -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server.properties  26448

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

- Use all input:

```
InformixHQ.bat startserver Encoding=utf-8 JVMMemx=1004m jarfile=informixhq-server.jar
 propfile=hqserver.properties
```

```
C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat startserver Encoding=utf-8 JVMMemx=1004m jarfile=informixhq-server.jar propfile=hqserver.properties
INFO: Please use list command to verify the process.


C:\Program Files\IBM Informix Software Bundle\hq>InformixHQ.bat list
CommandLine                                                    ProcessId
java  -Xmx1004m -Dfile.encoding=utf-8 -jar informixhq-server.jar hqserver.properties  3668

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

Starting the InformixHQ Server and the Agent on Linux/AIX

This topic provides a brief tutorial about the script used to get started with InformixHQ Server and Agent on Linux/AIX.

1. Locate your *informixHQ.sh* for Linux and *InformixHQ.ksh* for AIX (default path is ${INFORMIXDIR}/hq).

   Note: Following examples are shown on bash shell, however those will run exactly same on korn shell with .ksh script.

2. Use command **InformixHQ help** to see sample syntax: e.g- InformixHQ.sh help .

   Tip: *User can only run InformixHQ.sh for help menu*

   Figure 4. InformixHQ help

```
[root@test-server hq]# ./InformixHQ.sh

Command: InformixHQ
This command starts/stops InformixHQ Server and Agent.

Syntax:
InformixHQ startserver|startagent [encoding=<value>] [jvmmemx=<value>] [jarfile=<value>] [propfile=<value>]
InformixHQ stop <processid>
InformixHQ stopserver
InformixHQ stopagent [properties filename search string]
InformixHQ list [filename search string]

startserver          : Starts InformixHQ Server service
startagent           : Starts InformixHQ Agent service
stop                 : Stops InformixHQ Server/Agent service with processId
stopserver           : Stops InformiHQ Server process
stopagent            : Stops InformixQ Agent process (default all processes or filtered list if matching filename search string found)
list                 : Lists InformixHQ running processes (default all processes, filtered list if matching filename search string found)
encoding (Optional)  : Default value is utf-8
jvmmemx (Optional)   : JVM's default value will be used If not specified
jarfile (Optional)   : Default is informixhq-server.jar for startserver option and informixhq-agent.jar for startagent option.
                       For user provided filename, it must contain keyword 'informixhq' and it should end with .jar
propfile (Optional)  : Default is informixhq-server.properties for startserver option and informixhq-agent.properties for startagent option.
                       User can provide any custom name to properties file
```

3. Use command InformixHQ startserver to start InformixHQ server with default file names. e.g- ./InformixHQ.sh startserver .

```
[root@test-server hq]# ./InformixHQ.sh startserver
INFO: Please use list command to verify the process.
```

It will start INFORMIXHQ Server with default values for all the parameters including jar file name (*informixhq-server.jar*) and properties filename (*informixhq-server.properties*).

4. Use command InformixHQ startagent to start InformixHQ agent with default filenames. e.g- ./InformixHQ.sh startagent

```
[root@test-server hq]# ./InformixHQ.sh startagent
INFO: Please use list command to verify the process.
```

It will start an INFORMIXHQ Agent with default values for all the parameters including jar file name (informixhq-agent.jar) and properties filename (informixhq-agent.properties).

5. List the Java processes running. It will list all the InformixHQ processes running with ProcessID e.g: ./InformixHQ.sh list.

```
[root@test-server hq]# ./InformixHQ.sh list

   PID USER     COMMAND
 182725 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server.properties
 183016 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server-prod.properties
 183230 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties
 183294 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

6. Additionally user can filter the list by passing search keyword to the list command. e.g. – InformixHQ.bat list prod .

Note: It is recommended that properties file name should be unique which will help users to search specific processes.

```
[root@test-server hq]# ./InformixHQ.sh list prod

   PID USER     COMMAND
 183016 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server-prod.properties
 183230 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

7. Stop the InformixHQ server or agent by providing processID to stop command. It will stop InformixHQ process with provided processID. e.g: ./InformixHQ.sh stop 183230 .

```
[root@test-server hq]# ./InformixHQ.sh stop 183230

SUCCESS: The process with PID 183230 has been terminated.

[root@test-server hq]# ./InformixHQ.sh list

   PID USER     COMMAND
 182725 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server.properties
 183016 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server-prod.properties
 183294 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

8. stopserver command will stop the server processes. e.g. - InformixHQ.bat stopserver .

```
[root@test-server hq]# ./InformixHQ.sh stopserver

INFO: This command has found 1 server process(es).

    PID USER     COMMAND
 183016 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-server.jar informixhq-server-prod.properties

Please Press:
0: exit
1: terminate

Please enter one of the above options [0/1]:1

SUCCESS: The process with PID 183016 has been terminated.
```

0. exit: It will exit from the command.

1. Terminate: It will terminate server process.

Note: Unless valid option is provided, it will not proceed further.

9. stopagent command will stop the agent processes. e.g. - InformixHQ.bat stopagent .

```
[root@test-server hq]# ./InformixHQ.sh stopagent

INFO: This command has found 2 agent process(es).

    PID USER     COMMAND
 184138 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties
 184281 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties

Please Press:
0: exit
1: terminate all
2: terminate one by one

Please enter one of the above options [0/1/2]:
```

0: exit - It will exit from the command.

1: Terminate all - It will terminate all the agent processes.

2: Terminate one by one - It will terminate agent processes one by one after user confirmation.

Note: Unless valid option is provided, it will not proceed further.

Sample screenshot for 'terminate all processes' :

```
[root@test-server hq]# ./InformixHQ.sh stopagent

INFO: This command has found 2 agent process(es).

    PID USER     COMMAND
 184138 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties
 184281 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties

Please Press:
0: exit
1: terminate all
2: terminate one by one

Please enter one of the above options [0/1/2]:1

SUCCESS: The process with PID 184138 has been terminated.
SUCCESS: The process with PID 184281 has been terminated.
```

Sample screenshot for 'one-by-one process termination' :

```
[root@test-server hq]# ./InformixHQ.sh stopagent

INFO: This command has found 2 agent process(es).

    PID USER     COMMAND
 184514 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties
 184564 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties

Please Press:
0: exit
1: terminate all
2: terminate one by one

Please enter one of the above options [0/1/2]:2

    PID USER     COMMAND
 184514 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent.properties

Please press:
0: exit
1: terminate
2: move next

Please enter one of the above options [0/1/2]:1

SUCCESS: The process with PID 184514 has been terminated.

    PID USER     COMMAND
 184564 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties

Please press:
0: exit
1: terminate
2: move next

Please enter one of the above options [0/1/2]:1

SUCCESS: The process with PID 184564 has been terminated.
```

Additionally, user can search specific agent process using stopagent command. e.g. - stopagent prod .

```
[root@test-server hq]# ./InformixHQ.sh stopagent prod

INFO: This command has found 1 agent process(es).

    PID USER     COMMAND
 184564 ajayana+ java -Dfile.encoding=utf-8 -jar informixhq-agent.jar informixhq-agent-prod.properties

Please Press:
0: exit
1: terminate

Please enter one of the above options [0/1]:
```

Example

User can use different properties filenames (with appropriate informixServer.id for different servers) to run agents for different instances as follows

Following are sample commands where user can provide custom values (similar commands will work with agent)

- optional jar and properties filename to start InformixHQ server:

```
./InformixHQ.sh startserver jarfile=informixhq-myinstance1.jar propfile=hqserver.properties
```

```
[root@test-server hq]# ./InformixHQ.sh startserver jarfile=informixhq-myinstance1.jar propfile=hqserver.properties

INFO: Please use list command to verify the process.

[root@test-server hq]# ./InformixHQ.sh list
ProcessId  Name  Jar  Properties
112911 java informixhq-myinstance1.jar hqserver.properties

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

- Use encoding and JVM value:

```
./InformixHQ.sh startserver Encoding=utf-8 JVMMemx=1004m
```

```
[root@test-server hq]# ./InformixHQ.sh startserver encoding=utf-8 jvmmemx=1004m

INFO: Please use list command to verify the process.
```

- Use all input:

```
InformixHQ.bat startserver Encoding=utf-8 JVMMemx=1004m jarfile=informixhq-server.jar
 propfile=hqserver.properties
```

```
[root@test-server hq]# ./InformixHQ.sh startserver encoding=utf-8 jvmmemx=1004m jarfile=informixhq-server.jar propfile=hqserver.properties

INFO: Please use list command to verify the process.
```

```
[root@test-server hq]# ./InformixHQ.sh list
ProcessId  Name  Jar  Properties
114005 java informixhq-server.jar hqserver.properties

INFO: In case, process is not listed after startserver/startagent command, please check the log files.
```

Exception Reporting

If options are not provided correctly then script will report error as follows

- If jarfile/propfile option is not specified

```
[root@test-server hq]# ./InformixHQ.sh startserver abc.jar
ERROR: The syntax of the commmand is incorrect.
Invalid Parameters :  [abc.jar,]

INFO: Use command InformixHQ.sh help to see the help.
```

- filename is not specified with jarfile/propfile option

```
[root@test-server hq]# ./InformixHQ.sh startserver jarfile
ERROR: The syntax of the commmand is incorrect.
Invalid Parameters :  [jarfile,]

INFO: Use command InformixHQ.sh help to see the help.
```

- incorrect filename provided

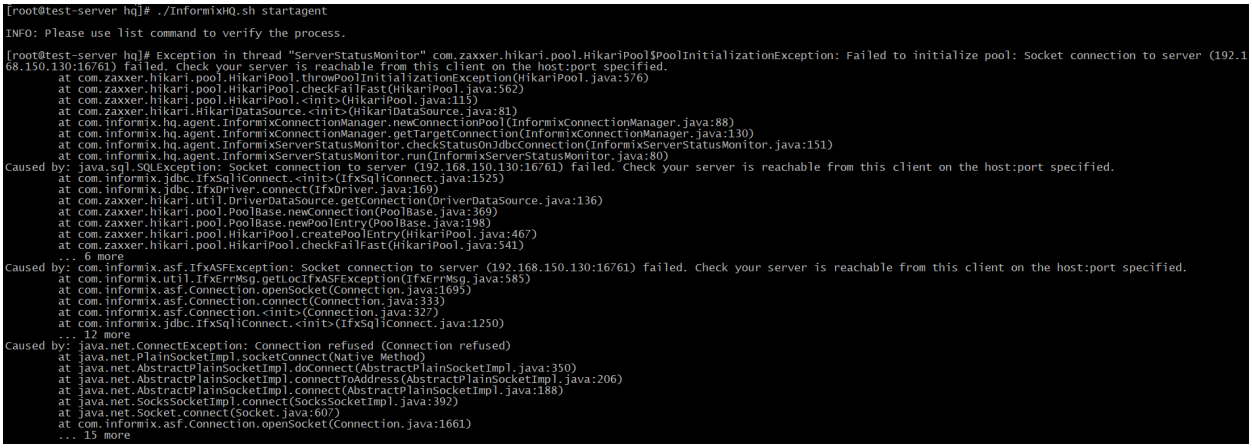
```
[root@test-server hq]# ./InformixHQ.sh startserver jarfile=abc.jar
ERROR: abc.jar doesn't exist.
```

- If specified or default files are not found also if custom filename is specified and file is not found in the folder, it will inform accordingly

```
[root@test-server hq]# ./InformixHQ.sh startagent jarfile=informixhq-inst1.jar propfile=agent-inst1.properties
ERROR: informixhq-inst1.jar doesn't exist.

ERROR: agent-inst1.properties doesn't exist.
```

- If IDS Server is not reachable by Agent. This is not related to script, however sometimes you might get exception if Server or Agent Java process encountered error. While starting agent process, if agent is not able to connect to IDS server specified in HQ then it might show error on screen as follows with message. Socket connection to server (Server IP:Port) failed. Check your server is reachable from this client on the host:port specified.



In such case, ensure, server added in InformixHQ server is configured correctly and is reachable and online.

InformixHQ in secure mode

InformixHQ – HTTPS Secured Connection:

InformixHQ setup as Secured Server (SSL)

About this task

This topic explains steps for setting up InformixHQ as Secured Server (SSL)

1. Create SSL keystore with self-signed certificate.
    - Use the following command to create jks keystore with self-signed certificate on HQ server at desired path:

      ```
      keytool -genkey -keyalg RSA -alias selfsigned -keystore keystore.jks
      ```

    - Modify -alias and -keystore values as needed.
    - User will be prompted to set keystore password, enter the password of your choice.
    - Remember to note the password as this will be needed at a later stage.
    - Password can also be set using argument -storepass in the keytool command given above.
    - User must type the HQ server hostname in response to keytool's first prompt in which it asks for first and last names. InformixHQ agent will check for this value while attempting a secured connection.
    - After this user will be asked a few more questions about unit, organization, locality, province, country. Provide values as per your choice, any values are fine.
    - Once all the values are entered, user will be asked for confirmation, type "yes".
    - Now, user will be asked password for certificate in keystore. If you wish to keep this password same as keystore (entered in step 1), press enter. If different password, remember to note it down for later use.

For further details on how to generate jks self-signed certificate, refer How to use keytool to create a server certificate and How to create a self signed certificate using Java Keytool

2. Note down SSL keystore path, filename just created.
3. Modify monitoring-server.properties to add/modify following properties:
   - httpPort=8080
   - httpsPort=8082 (HQ server port serving secured connections)
   - redirectHTTPtoHTTPS=true (if true, redirects http request to https)
   - ssl.keystore.file=<Keystore absolute path with file name>
   - ssl.keystore.password=<password>
   - #ssl.key.password=<password> (Use this only if you have set different password for certificate within keystore)
4. Now start InformixHQ server as usual.
5. Open a browser and enter the following url:

```
http://<HQserverhost>:8080 OR https://<HQserverhost>:8082
```

If redirectHTTPtoHTTPS is set to true, it will automatically direct http request to https port.

Note:

1) Internally InformixHQ server uses self-signed certificate to authenticate secured connection.

2) User can verify by providing incorrect values for **ssl.keystore.file** or **ssl.keystore.password** in parameter file **monitoring-server.properties**. After this, if user tris to access HQ server from a browser, it will respond with “connection refused.”

3) On a windows machine, add double slash for keystore path such as **D:\\SSL\\keystore\\key.jks**

InformixHQ Agent setup for secured connection (SSL)

About this task

This topic explains steps for setting up InformixHQ Agent for secured socket connection (SSL).

1. Create client keystore for HQ Agent connection:
   - Extract self-signed certificate from HQ server keystore using following command:

```
keytool -export -alias selfsigned -file server.cer -keystore keystore.jks
```

     - Same alias should be used as that of server keystore.
     - Keystore name would be server keystore.
     - This command will extract certificate to file **server.cer** (value provided for -file argument)
   - Copy this certificate to client machine (where HQAgent would be running).
   - Import this certificate to client keystore on HQAgent box using following command:

```
keytool -import -v -trustcacerts -alias selfsigned -file server.cer -keystore cacerts.jks
```

- Ensure alias is same as used in server keystore.
- This will ask to set keystore password, enter the password of your choice.
- Note the password as this will be needed at a later stage.

2. Note down client SSL keystore path, filename just created.
3. Modify **monitoring-agent.properties** to add/modify following properties:
    - serverInstance.id=<ID of the Informix Server defined in InformixHQ>
    - server.host=<InformixHQ server host>
    - server.port=<InformixHQ server port> (for above example, it would be 8082)
    - ssl.enable=true
    - ssl.keystore.file=<client keystore absolute path and file name>
    - ssl.keystore.password=<password>
    - ssl.keystore.type=jks
4. Start InformixHQ agent as usual.

Note:

1) In case, InformixHQ server and agent are running on the same machine, Server keystore can also be used as client keystore for Agent. This means the same Keystore path and password will be used in both monitoring-server.properties and monitoring-agent.properties files.

2) In case InformixHQ server and agent are running on different machines, users can also copy server keystore to agent machine and use the same keystore, instead of extracting certificate and creating new client keystore.

3) If any changes are done to keystore or properties file while HQ server or agent is running, ensure to restart InfomrixHQ server or Agent appropriately.

4) On windows machine, add double slash for keystore path such as **D:\\SSL\\keystore\\key.jks**

Logging in InformixHQ

This topic provides a brief tutorial on logging in InformixHQ.

The InformixHQ server and agent use the log4j2 library for logging. During InformixHQ installation, logging configuration files are provided with default configurations, *monitoring-server.log4j.xml* for InformixHQ server and *monitoring-agent.log4j.xml* `for` InformixHQ agent. By default, these files are present at $INFORMIXDIR/hq folder.

Users can customize default logging behavior by modifying *monitoring-server.log4j.xml* or *monitoring-agent.log4j.xml* file in the current HQ directory or classpath when starting the InformixHQ server or InformixHQ agent respectively.

Users can do customizations with logging levels (ERROR, WARN, INFO, or DEBUG), configure path for publishing logging information to various preferred destinations (such as a database, file, console, UNIX Syslog, etc.), configure filename format, configure rolling log file depending upon size, time or both, etc.

By default, the InformixHQ server and agent will log messages at INFO level to an*monitoring-server.log* file and an*monitoring-agent.log* file respectively. Different logging levels are explained in the sections given below.

If *monitoring-server.log4j.xml* or *monitoring-agent.log4j.xml* is not available while starting InformixHQ server or agent, all the logging configurations will be set to default by the application similar to monitoring-server.log4j.xml or monitoring-agent.log4j.xml files respectively.

Additionally, for InformixHQ agent, if *monitoring-agent.log4j.xml* file is available, logs for all the agents running from same directory will be appended to one agent log file (default name - monitoring-agent.log or file name as configured in *monitoring-agent.log4j.xml* file) and if *monitoring-agent.log4j.xml* file is NOT available, then application will create a separate log file for each agent process. The file name will be like monitoring-agent_1.log, monitoring-agent_2.log, … depending on number of agent processes running. The number in filename represents Informix server id added in Agent properties file.

LOGGERS

The <Logger> tags describe the different log levels that can be changed according to the Java packages of the source code.

Log levels are: OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE and ALL.

- **FATAL:** A log level which indicates that the application encountered an event or entered a state in which one of the crucial business functionalities is no longer working.
- **ERROR:** A log level which indicates that one or more functionalities are not working, preventing some functionalities from working correctly.
- **WARN:** A log level which indicates that an unexpected behaviour happened inside the application, but it is continuing its work and the key business features are operating as expected.
- **INFO:** A log level which indicates that an event happened, the event is purely informative and can be ignored during normal operations.
- **DEBUG:** A log level used for events considered to be useful during software debugging when more granular information is needed.
- **TACE:** A log level describing events showing step by step execution of your code that can be ignored during the standard operation.
- **ALL:** A log level which includes all logging levels.
- **OFF:** A log level which indicates that logging is turned off

Logging levels' summary for InformixHQ server and agent:

Logging level	FATAL	ERROR	WARN	INFO	DEBUG	TRACE	ALL
OFF							
FATAL	X						
ERROR	X	X					
WARN	X	X	X				
INFO	X	X	X	X			

Logging level	FATAL	ERROR	WARN	INFO	DEBUG	TRACE	ALL
DEBUG	X	X	X	X	X		
TRACE	X	X	X	X	X	X	
ALL	X	X	X	X	X	X	X

If we keep logging OFF then no logs will be printed on files or any other *medium.*

If we keep logging as Fatal, Error, Warn or Info - we will get a brief description of them and no stack trace for them will be printed on file or any other *medium*.

If we keep logging as Debug, Trace or ALL - we will get a brief description of them, and a detailed stack trace will be printed on a file or any other *medium*.

Here, the *medium* refers to a file, STDOUT and Console.

Example of a Logger:

```
<Loggers>
                <!-- The base logging level is set here -->
                <!-- You can choose from (TRACE, DEBUG, INFO, WARN, ERROR) -->
                <Root level="INFO">
                        <AppenderRef ref="FILE" />
                        <!-- <AppenderRef ref="CONSOLE" /> -->
                        <!-- <AppenderRef ref=" STDOUT " /> -->
                </Root>
```

AppenderRef: we are specifying the location of logs [a FILE/CONSOLE/STDOUT]

```
<!-- You can configure custom logging levels (TRACE, DEBUG, INFO, WARN,
                        ERROR) for any java package name -->
<Loggers>
<Configuration>
                <Logger name="h2database" level="WARN" />
                <Logger name="com.zaxxer.hikari" level="INFO" />
                <Logger name="com.zaxxer.hikari.pool.HikariPool" level="OFF" />
                <Logger name="com.zaxxer.hikari.HikariDataSource" level="OFF" />
</Loggers>
</Configuration>
```

There are a few external Java libraries used in InformixHQ for utilizing different functionalities provided by them. Logging levels for such libraries are defined and managed by respective vendors.

For example, in InformixHQ, description is printed for INFO level whereas external library might print different level of details at the same logging level. Hence, the logging `level` of some external packages are tuned through configuration file settings to suppress unnecessary information.

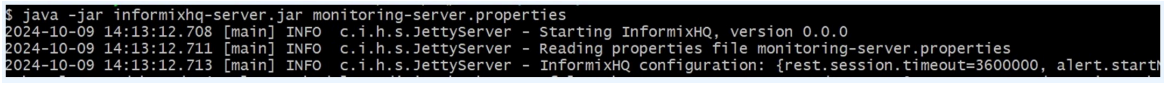
Note:

- It is recommended not to change the logging level for external packages mentioned above. If it is changed to a different logging level, unexpected logs may appear in log files.
- When InformixHQ fails to connect to Informix server, connection failure logs will be printed per hour. Once connection is established with Informix server, success log will be printed to the log file.
- <Configuration monitorInterval="5" status="FATAL">

  Log4j has the ability to automatically detect changes to the configuration file and reconfigure itself by using the attribute *monitorInterval*. If the monitorInterval attribute is specified on the configuration element and is set to a non-zero value, then the file will be checked the next time a log event is evaluated and/or logged and the monitorInterval has elapsed since the last check. The minimum interval is 5 seconds.

  The attribute status handles the level of internal Log4j events that should be logged to the console. This attribute is added to the Configuration element of monitoring-server-example.log4j.xml and monitoring-agent-example.log4j.xml files. You can change its level if you want to get the internal Log4j events logged on the console.

  Figure 5. Internal Log4j events



Similar behaviour is applicable for Agent log configurations as well.

Appenders:

Appender is primarily responsible for sending log messages to a certain output destination such as STDOUT, console, files. etc.

Note: `Only` one appender can be configured at a time. Also, reference specific to appender name should be given under logger section in tag(<AppenderRef>), we can add multiple AppenderRef.

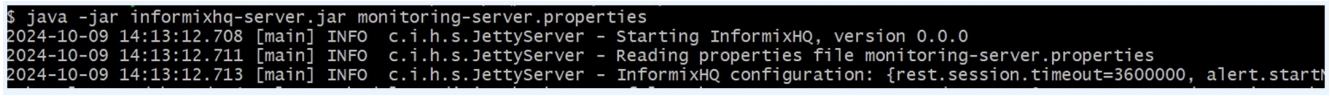
ConsoleAppender

The ConsoleAppender writes its output to either the console from where jar is run or as per the host variable configured (System.out or System.err with System.out being the default target) and writes in log file, too.

```
<Appenders>
  <Console name="STDOUT" target="SYSTEM_OUT">
    <PatternLayout pattern="%m%n"/>
  </Console>
</Appenders>
<Loggers>
  <Root level="INFO">
    <AppenderRef ref="STDOUT"/>
```

```
    </Root>
  </Loggers>
```

Figure 6. Internal Log4j events



Note: Only one appender can be configured at a time. Also, reference specific to appender name should be given under logger section in tag(<AppenderRef>).

RollingFile Appender

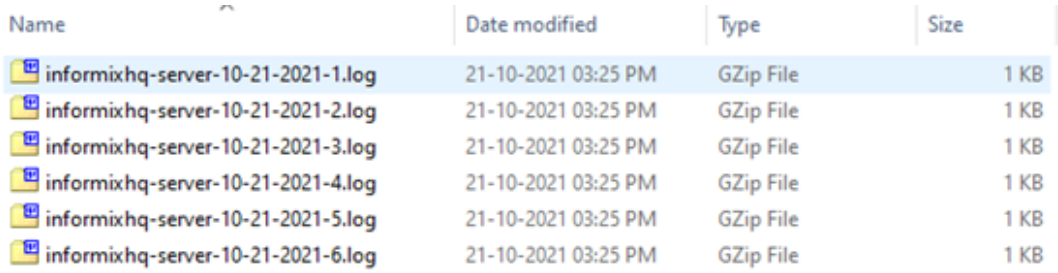
The RollingFileAppender is an OutputStreamAppender that writes to the file named in the fileName parameter and rolls the file over according to the TriggeringPolicy and the RolloverPolicy.

- **Triggering Policies**
  - **SizeBased Triggering Policy**

    Once the file reaches the specified size, the SizeBasedTriggeringPolicy causes a rollover. The size can be specified in bytes, with the suffix KB, MB or GB, for example 20MB. When combined with a time based triggering policy, the file pattern must contain a %i otherwise the target file will be overwritten on every rollover as the SizeBased Triggering Policy will not cause the timestamp value in the file name to change. When used without a time based triggering policy, the SizeBased Triggering Policy will cause the timestamp value to change.

    For illustration purpose we have set the size limit of 1 KB in following configuration snippet and screenshot displaying new file generated after configured size.

```
<Appenders>
      <RollingFile name="FILE" fileName="logs/monitoring-server.log"
 filePattern="logs/$${date:yyyy-MM}/informixhq-server-%d{MM-dd-yyyy}-%i.log.gz">
         <PatternLayout>
            <Pattern>%d %p %c{1.} [%t] %m%n</Pattern>
         </PatternLayout>
         <Policies>
            <SizeBasedTriggeringPolicy size="1 KB" />
         </Policies>
      </RollingFile>
   </Appenders>
   <Loggers>
      <Root level="INFO">
         <AppenderRef ref="FILE" />
      </Root>
   </Loggers>
```



Name	Date modified	Type	Size
informixhq-server-10-21-2021-1.log	21-10-2021 03:25 PM	GZip File	1 KB
informixhq-server-10-21-2021-2.log	21-10-2021 03:25 PM	GZip File	1 KB
informixhq-server-10-21-2021-3.log	21-10-2021 03:25 PM	GZip File	1 KB
informixhq-server-10-21-2021-4.log	21-10-2021 03:25 PM	GZip File	1 KB
informixhq-server-10-21-2021-5.log	21-10-2021 03:25 PM	GZip File	1 KB
informixhq-server-10-21-2021-6.log	21-10-2021 03:25 PM	GZip File	1 KB

- **TimeBased Triggering Policy**

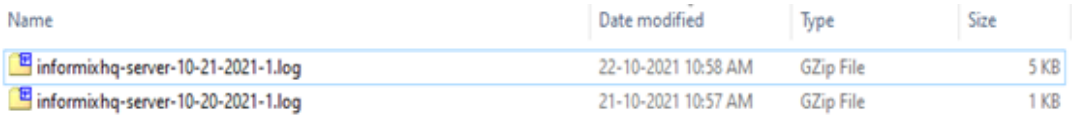
The TimeBasedTriggeringPolicy causes a rollover once the date/time pattern no longer applies to the active file. This policy accepts an interval attribute which indicates how frequently the rollover should occur based on the time pattern and a modulate boolean attribute. The default value of interval is 1. Following snippet shows configuration to create a new logfile everyday.

Note: To create a new log file everyday, configure filePattern parameter as {MM-dd-yyyy} whereas to create a new log file hourly, configure filePattern parameter as {MM-dd-yyyy-HH}.

Parameters of TimeBasedTriggeringPolicy:

Parameter Name	Type	Description
interval	integer	How often a rollover should occur based on the most specific time unit in the date pattern. For example, with a date pattern with hours as the most specific item and increment of 4 rollovers would occur every 4 hours. The default value is 1.
modulate	boolean	Indicates whether the interval should be adjusted to cause the next rollover to occur on the interval boundary. For example, if the item is hours, the current hour is 3 am and the interval is 4 then the first rollover will occur at 4 am and then next ones will occur at 8 am, noon, 4pm, etc.

```
<Appenders>
<RollingFile name="FILE" fileName="logs/monitoring-server.log"
 filePattern="logs/$${date:yyyy-MM}/informixhq-server-%d{MM-dd-yyyy}-%i.log.gz">
         <PatternLayout>
            <Pattern>%d %p %c{1.} [%t] %m%n</Pattern>
         </PatternLayout>
         <Policies>
            <TimeBasedTriggeringPolicy interval="1" modulate="true"/>
  </Policies>
      </RollingFile>
</Appenders>
<Loggers>
      <Root level="INFO">
         <AppenderRef ref="FILE" />
      </Root>
   </Loggers>
```



Name	Date modified	Type	Size
informixhq-server-10-21-2021-1.log	22-10-2021 10:58 AM	GZip File	5 KB
informixhq-server-10-20-2021-1.log	21-10-2021 10:57 AM	GZip File	1 KB

- **Composite Triggering Policy**

The CompositeTriggeringPolicy combines multiple triggering policies and returns true if any of the configured policies return true. The CompositeTriggeringPolicy is configured simply by wrapping other policies in a Policies element.

For example, the following XML fragment defines policies that rollover the log when the JVM starts, when the log size reaches twenty megabytes, and when the current date no longer matches the log's start date.

```
<Appenders>
<RollingFile name="FILE" fileName="logs/monitoring-server.log"
 filePattern="logs/$${date:yyyy-MM}/informixhq-server-%d{MM-dd-yyyy}-%i.log.gz">

      <PatternLayout>
        <Pattern>%d %p %c{1.} [%t] %m%n</Pattern>
      </PatternLayout>
      <Policies>
  <SizeBasedTriggeringPolicy size="20 MB" />
  <TimeBasedTriggeringPolicy />
</Policies>
    </RollingFile>
  </Appenders>
  <Loggers>
    <Root level="INFO">
      <AppenderRef ref="FILE"/>
```

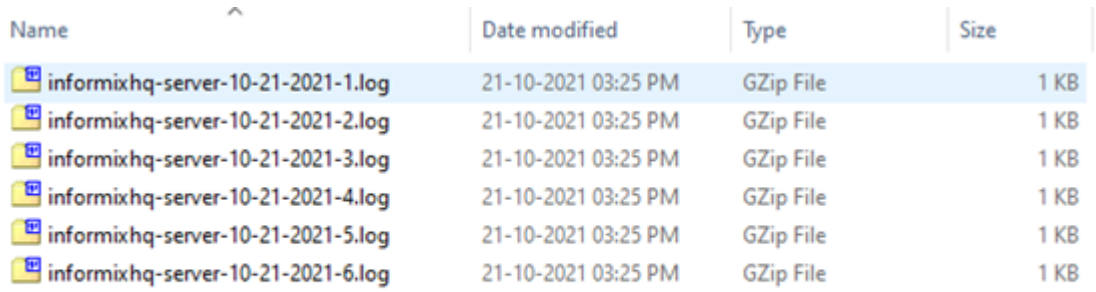
```
    </Root>
  </Loggers>
```

- **Default Rollover Policy**

  The default rollover takes both date/time pattern and an integer specified in filePattern Attribute in RollingFileAppender. If the pattern contains an integer, it will be incremented on every rollover. If the date/time pattern is present, it will be replaced with current date and time values. If the file pattern ends with ".gz", ".zip", ".bz2", ".deflate", ".pack200", or ".xz", the resulting archive will be compressed using the compression scheme that matches the suffix. Default Rollover Policy needs to have at least one triggering policy configured.

  This example shows a rollover strategy that will keep up to 20 files before removing them.

```
<Appenders>
      <RollingFile name="FILE" fileName="logs/monitoring-server.log"
 filePattern="logs/$${date:yyyy-MM}/informixhq-server-%d{MM-dd-yyyy}-%i.log.gz">
         <PatternLayout>
            <Pattern>%d %p %c{1.} [%t] %m%n</Pattern>
         </PatternLayout>
         <Policies>
            <SizeBasedTriggeringPolicy size="1 KB" />
         </Policies>
         <DefaultRolloverStrategy max="20" />
      </RollingFile>
   </Appenders>
   <Loggers>
      <Root level="INFO">
         <AppenderRef ref="FILE" />
      </Root>
   </Loggers>
```



Note:

- We can specify pattern layout as per given format. For more information, see pattern layout.
- There are many more appenders available in Log4j2 framework. For more information, see log4j2 documentation.

Handling Account Lockout

This topic provides details to manage account lock out situation.

Overview

This document provides guidelines for handling situations where accounts are locked due to incorrect login attempts. It outlines the steps required to unlock accounts and provides instructions for managing multiple accounts under various scenarios.

General Lockout Policy

- **Lockout Threshold**: Accounts will be locked automatically after 5 consecutive failed login attempts.
- **Purpose**: This measure enhances security by preventing unauthorized access to accounts through brute-force attacks.

Scenario 1: Managing Locked Admin and User Accounts

Situation

If an account is locked (admin or user) due to multiple incorrect login attempts, but other admin accounts are active, follow these steps to unlock the locked account:

1. **Login with an active admin account:**
    - Use the credentials of an admin account that is not locked.
2. **Access User Management:**
    - Navigate to the **User Management** section under **System Settings**.
3. **Find and unlock the locked account:**
    - Locate the locked account in the list of users.
    - Change the account status from "Locked" to "Unlocked".

Scenario 2: All Admin Accounts Locked

Situation

In a rare case where all admin accounts are locked due to multiple incorrect login attempts, follow these steps to regain access and unlock accounts:

1. **Restart the InformixHQ web application:**
    - Restart the application to reset the lockout state for the admin account.
2. **Login with standard admin credentials:**

- Use the standard admin credentials to log in.
- You will be prompted to set a new password upon first login after the restart.

3. **Set a new password:**
   - Follow the prompts to create and confirm a new password.

InformixHQ Concepts

This topic covers some of the conceptual aspects of InformixHQ.

Group

InformixHQ provides the ability to create groups of servers to make them easier to manage and monitor. InformixHQ's groups are based on a hierarchy. The base “root” group is the top level group for all InformixHQ groups and servers. From this root group, you can add as many servers and sub-groups as you desire, nesting them to whatever level makes sense for your organization.

You can define monitoring and alerting profiles for groups, simplifying the task of managing monitoring for all of your database servers.

Agent

The InformixHQ agent is a lightweight Java based program that is designed to run alongside each Informix database server, gathering data about the performance of the system. The data gathered by the agent is fully configurable and is defined by the list of **sensors** in the server’s **monitoring profile**. The data gathered by the agent is stored in a **repository database**.

Repository Database

A repository database holds information collected by the InformixHQ agent about the Informix database server. The repository database can either be local to the database server that is being monitored or it can be on a remote Informix instance. You can define a common repository database shared by multiple Informix database server instances, or you can define a separate repository database for each monitored instance.

The repository database must be an Informix database and must exist before the agent is started. You must define the database server to be used as a repository in the InformixHQ UI, using the **Add Server** action on any group dashboard, before it can be defined as a repository.

The repository database is a production environment, as such, it must be deployed within a licensed Informix instance using almost any Informix edition (except Developer Edition due to its non-production license restriction) including Innovator-C. Be aware that there are scalability and other functional limitations to Innovator-C instances including a lack of H/A failover technologies, not covered by other edition S&S agreements, and an instance storage limitation. Using Innovator-C to host the repository database for many monitored servers, may require segmenting the monitored instances across multiple repository databases in multiple Innovator-C environments to stay within the size limit.

To define a repository database for a particular Informix database server, go to the server's **Setup** page in the UI and click on the **Agent** tab.

Sensor

A sensor defines a metric or set of metrics for the agent to gather. An example is the “DBSpace Usage” sensor that gathers metrics on used and free space for all database server spaces.

Monitoring Profile

A monitoring profile defines the list of sensors that the agent runs to gather data about and Informix database server instance or about its host operating system.

For each sensor in a monitoring profile, you can configure the frequency at which that sensor will run and how long that sensor’s data will be kept in the repository database.

Monitoring profiles can be configured for groups as well as servers. InformixHQ uses the concept of inheritance for determining a particular server's or group's monitoring profile. All servers and groups inherit monitoring profile information from its parent group in the hierarchy. Servers or groups can also disable or override the configuration of any sensors inherited from a parent group.

Alert

An alert defines a condition that should trigger an alerting incident in InformixHQ. An example would be an alert defined for when the Informix database server goes offline.

Alerting profile

An alerting profile defines the set of alerts configured for a particular server or group. Alerting profiles follow an inheritance model similar to monitoring profiles.

Alerting incident

While the monitoring data is collected by the InformixHQ agent, it is the InformixHQ server that is tasked with evaluating alerting conditions. As data is collected by the agent, the InformixHQ server evaluates each new incoming data point against the alert definitions in the server's alerting profile. Any data point that meets an alerting condition triggers an alerting incident.

Alerting incidents are shown in the InformixHQ UI for that server. Alerting incident counts are also aggregated and highlighted on the group dashboards. Alerting incidents can be acknowledged as read or even deleted from the Incidents page in the UI. Users of InformixHQ can configure their own alerting preferences to automatically receive alerting incidents directly via email, Twilio, or PagerDuty.

InformixHQ Server

The InformixHQ server is a Java 11 based Jetty web server. The server is the heart of InformixHQ. It manages the monitoring profiles for all instances, communicates with agents, handles all alerting activities, hosts the web UI, and provides the REST services that the web UI depends on.

InformixHQ Server Configuration

A properties file is required to run the InformixHQ server. When starting the InformixHQ server, you can pass the properties file name as part of the start command. Otherwise, InformixHQ will look for a properties file named `monitoring-server.properties` in the classpath.

An example configuration file documenting all supported InformixHQ server configuration properties can be found in `$INFORMIXDIR/hq/monitoring-server-example.properties`.

- Optional configuration properties
  - alert.numberConditionCheckThreads on page 44
  - alert.startNumberAlertSendThreads on page 43
  - dataSource.IFX_ISOLATION_LEVEL on page 44
  - hostname on page 44
  - httpPort on page 44
  - httpsPort on page 44
  - h2.encrypt.algorithm on page 44
  - h2.encrypt.enable on page 44
  - h2.encrypt.password on page 45
  - pool.connectionTimeout on page 45
  - pool.maximumPoolSize on page 45
  - pool.minimumIdle on page 45
  - pool.idleTimeout on page 45
  - redirectHTTPtoHTTPS on page 45
  - rest.session.timeout on page 45
  - ssl.keystore.file on page 45
  - ssl.keystore.password on page 46
  - ssl.key.password on page 46
  - user.password.algorithm on page 46
  - user.password.maxAge on page 46
  - user.password.minLength on page 46
  - user.password.requireLowerCase on page 46
  - user.password.requireNumber on page 46
  - user.password.requireSpecialCharacterFromSet on page 46
  - user.password.requireUpperCase on page 46

alert.startNumberAlertSendThreads

Configures the number of threads in the thread pool that processes and dispatches alert notifications (by email, Twilio, Pager Duty, etc.) when an alerting incident occurs. The default number of threads is 4.

alert.numberConditionCheckThreads

Configures the number of threads in the thread pool that checks whether alerting conditions have been violated whenever new monitoring data comes in. The default number of threads is 4.

dataSource.IFX_ISOLATION_LEVEL

Specifies the isolation level to set on JDBC connections to the various Informix database servers. The default value is 1.

hostname

The host name of the InformixHQ server. The host name determines the network adapter or interface that the InformixHQ server binds the server socket to.

The default value is an empty string. When set to an empty string, the InformixHQ server will bind to all available network interfaces on the host machine.

httpPort

The HTTP port to run the InformixHQ server on. This port will serve both the InformixHQ web UI and the InformixHQ REST API. Set this value to -1 to disable the HTTP protocol for InformixHQ. If httpPort is set to -1, make sure that httpsPort is set to something other than -1.The default value is 8080.

httpsPort

The HTTPS port to run the InformixHQ server on. This port will serve both the InformixHQ web UI and the InformixHQ REST API.

Set this value to -1 to disable the HTTPS protocol for InformixHQ. If httpsPort is set to -1, make sure that httpPort is set to something other than -1.

If httpsPort is something other than -1, you must set the **ssl.keystore.file** and **ssl.keystore.password** properties, and potentially also the **ssl.key.password** property if your key password is different from the keystore password.

The default value is -1 indicating that HTTPS is disabled by default.

h2.encrypt.algorithm

Sets the algorithm for H2 database file encryption. The encryption algorithms supported by H2 are AES, XTEA, and FOG. The default value is AES.

h2.encrypt.enable

Controls whether the H2 database file which holds InformixHQ server's internal metdata is encrypted. If you set this property to true, you must also set the h2.encrypt.password property. The default value is false.

h2.encrypt.password

Sets the password to use for H2 database file encryption. If h2.encrpyt.enable is set to true, you must set the password for encryption.

pool.connectionTimeout

Specifies the number of milliseconds to wait for a JDBC connection to an Informix database server to be established before it times out. The default value is 5000 (5 seconds).

pool.idleTimeout

Specifies the number of milliseconds that a JDBC connection can be idle in the connection pool before it is closed. The default value is 60000 (1 minute).

pool.maximumPoolSize

The maximum number of JDBC connections in each connection pool. The InformixHQ server will maintain a connection pool for each Informix database that it needs to connect to. The **pool.maximumPoolSize** puts a cap on the total number of open JDBC connections that can be established to each database.

The default value is 5.

pool.minimumIdle

The minimum number of idle JDBC connections in each connection pool. The InformixHQ server will maintain a connection pool for each Informix database server that it needs to connect to. Setting **pool.minimumIdle** to zero indicates that all JDBC connections in the connection pool should be closed when they have been sitting idle for longer than the **pool.idleTimeout** threshold. Setting **pool.minimumIdle** to a positive integer indicates the number of connections that should be kept open in the connection pool even when they exceed the **pool.idleTimeout**. The default and recommended value is 0.

redirectHTTPtoHTTPS

If set to true, HTTP traffic to InformixHQ will automatically be redirected to HTTPS. This will include web socket communication between the InformixHQ server and agent. If this value is set to true, you will be required to configure SSL in your agent configuration properties.

The default value is false.

rest.session.timeout

Specifies the number of milliseconds that a REST session can be idle before it is closed. The default value is 3600000 (60 minutes).

ssl.keystore.file

The path to the keystore file that contains the certificate to use for network encryption. This property must be set if **httpsPort** is set to something other than -1. Use Java keytool tool to create a Java keystore and a certificate.

ssl.keystore.password

The password to unlock the keystore file for network encryption. If this property is not set and the HTTPS is configured, you will be prompted on the command line to enter the keystore password when starting the InformixHQ server.

ssl.key.password

The password to unlock the entry into the keystore. The default value is no password, which means to use the keystore password. If the entry into the keystore requires a password that is different from the keystore password, set this property to the entry password.

user.password.algorithm

Sets the algorithm for InformixHQ login password. The encryption algorithms supported by InformixHQ are SHA-1, SHA-256, SHA-384, SHA-512. The default value is SHA-256.

user.password.maxAge

Controls the maximum age (in days) of a user password. User passwords that are older than the max age will be considered as expired. Setting this property to zero, which is the default value, specifies that user passwords never expire. Setting this property to a value greater than zero specifies the maximum age (in days) of a user password before it expires. A user will start receiving notifications in the InformixHQ UI when the difference between the current date and the password expiration date is less than or equal to 15 days.

user.password.minLength

Controls the minimum length for a user password. The default value is 8.

user.password.requireLowerCase

Controls whether user passwords are required to include at least one lowercase character. The default value is true.

user.password.requireNumber

Controls whether user passwords are required to include at least one number. The default value is true.

user.password.requireSpecialCharacterFromSet

Controls whether user passwords are required to include at least one special character. An empty string indicates that no special characters are required. Setting this value to “!@#$%^&*()” would require user passwords to include at least one of those characters. The default value is an empty string.

user.password.requireUpperCase

Controls whether user passwords are required to include at least one uppercase character. The default value is true.

InformixHQ UI

This topic highlights important aspects of the InformixHQ user interface. The InformixHQ UI is run by the Jetty web server that is embedded in the InformixHQ server. The default URL for the InformixHQ server is `http://localhost:8080/` but the InformixHQ server configuration file can be used to change the port and configure whether https is enabled.

Adding Servers and Groups

After logging in to InformixHQ for the very first time, you will be taken to a group dashboard for the root group. This dashboard will initially contain zero servers and zero groups.

Adding Servers

To add servers, click the **Add Server** button to add connection information for your Informix database server instances.

When adding a server to InformixHQ, you can provide two sets of credentials:

- Monitoring credentials
- Admin credentials

The **monitoring credentials** are used by the InformixHQ server whenever a user navigates to any part of the UI that issues a REST query which needs to gather data from the live Informix database server instance. The **monitoring credentials** are also used by the agent whenever it gathers data from your database server. The **admin credentials** are used whenever a user in the UI requests that an administration action be performed on the database server, for example create a dbspace, edit an onconfig parameter, or deploy an agent.

Required privileges for the monitoring user	• CONNECT access to the sysmaster database • CONNECT access to the sysadmin database
Optional privileges for the monitoring user	• select privileges on the sysconblock, syssqltrace, syssqltrace_info, syssqltrace_hvar, and syssqltrace_iter tables in the sysmaster database. • select privileges on the ph_task, ph_run, ph_alert, ph_threshold, ph_bg_jobs, and ph_bg_jobs_results tables in the sysadmin database. **Note:** Providing these optional privileges to the monitoring user enables the UI to show information

	about the server's the storage pool, sql tracing, auto update statistics, and scheduler tasks.
Required privileges for the admin user	• Part of the DBSA group. • CONNECT access to the sysadmin database. • Privilege to run SQL Admin API commands. • select/insert/update/delete privileges on ph_task, ph_run, ph_alert, ph_threshold, ph_bg_jobs, and ph_-bg_jobs_results tables in the sysadmin database. • Execute privilege on the following functions: ◦ exectask(lvarchar) ◦ exectask(lvarchar,lvarchar) ◦ exectask(integer) ◦ exectask(integer,lvarchar) If this server is used as a repository server storing monitoring data for one or more instances, the admin user must also have: • RESOURCE access to the repository database.

It is required that you provide monitoring credentials when adding your Informix database server information to InformixHQ. Admin credentials need only be provided if you want to use InformixHQ to run administrative actions on your database server or if you want the InformixHQ server to automatically deploy your agents.

SSL Connections

If your database supports or requires SSL connections you can setup SSL using the connection properties on the **Add Server** page. Specify the following connection parameters:

SSLCONNECTION	`true`
SSL_TRUSTSTORE	Absolute or relative path to the truststore/keystore file from the perspective of the directory from which the agent and the InformixHQ server start. The truststore/keystore file must be present where both InformixHQ server and InformixHQ agent are running
SSL_TRUSTSTORE_PASSWORD	Password to unlock the truststore/keystore file for both InformixHQ server and InformixHQ agent

Advanced Connection Properties

You can specify any of the advanced JDBC connection parameters when you setup your connection to alter the behavior of the underlying connections InformixHQ makes to the Informix database servers. For more information, see Informix environment variables with the Informix JDBC Driver

Adding Groups

From the dashboard, you can also click the **Add Group** button to create groups of servers to make it easier to monitor and manage multiple servers.

Related information

InformixHQ Agent Setup on page 123

Exploring Groups

On a group dashboard, you will see all of the servers and sub-groups that belong to the group. For each sub-group, the UI will show a card indicating the number of servers in the group (the first number is the number of servers directly in the group; the number in parenthesis is the total number of servers under it in the hierarchy). It will also show if and how many unread incidents exist for servers within the group.

There will also be a card for each server in the current group indicating the server and agent status. Since the agent monitors server status information, the server status will be unknown if the agent is offline.

From the group dashboard, you can use the **Add Server** and **Add Group** buttons to add servers or groups to the hierarchy. Each server or group also has additional actions to rename, edit connection information (for servers only), move, or delete that object. To drill down on any server or group, click on the server or group card.

From the group dashboard, you can also access the monitoring and alerting profiles for the group by selecting those menu items from the left-hand sidebar. If you are a System Administrator, you will also see a link for granting/revoking permissions on the group.

Exploring Informix Database Servers

Clicking on any server card from a group dashboard will take you to the server's dashboard. The dashboard includes sections on server status and any alerting incidents that have occurred, as well as information on high availability, threads, storage performance metrics, host memory and CPU usage, etc. If the agent is not running or sensors are not enabled, you will see the latest measurement for these metrics queried directly from the live database server. If the corresponding sensors are running, the server dashboard will show the recent history of each of these metrics graphically to give you a visual indication of how the database server is performing.

Note: If you see a triangular exclamation icon in the top right of any dashboard, that is an indication that the dashboard uses sensors that do not exist in the server's monitoring profile. Clicking on that icon will open a pop-up detailing the sensors used by the dashboard and providing a one-click button to enable all of those sensors.

For a particular database server, use the left-hand sidebar menu to continue to explore the server.

Configuring Monitoring

After clicking on a server or group from the dashboard, you can use the Monitoring link on the left-hand sidebar menu to configure the monitoring profile for that server or group. The Monitoring page lists the sensors currently configured in the server or group's monitoring profile.

Clicking the **Add Sensors** button will open a pop-up displaying the list of sensors not yet configured for this server or group. You can add custom sensors to this list from the Sensor Management page.

Click the **Edit** (pencil) icon to modify the run interval or data retention interval.

For servers or groups that inherit sensors from parent groups, there will also be a list of Inherited Sensors. The buttons next to the inherited sensors allow you to disable or override properties of the inherited sensors.

Configuring Alerting

Alerts in InformixHQ can be fully customized. You can configure not only what you want to be alerted on, but also the threshold or condition that should trigger that alert.

Alerts defined for a group are automatically inherited by the child groups and child servers of that group. Defining alerts at the group level simplifies the process of managing alerts for multiple servers.

To create alerts:

1. Click on a server or a group from the dashboard and then select the **Alerting** link on the left-hand sidebar menu. The **Alerting** page lists the alerts configured in the server or group's alerting profile.
2. Click the **Add Alert** to open a form that guides you through the process of defining an alert. Provide an alert name to identify it.
3. Select what you want to alert on: Informix server status, agent status, or data from a sensor metric.
4. Define the alerting condition. For example, alert me when the Informix server status is offline or alert me when the number of sessions connected to Informix is greater than 200.

For servers or groups that inherit alerts from parent groups, there will also be a list of Inherited Alerts. The buttons next to the inherited alerts allow you to disable inherited alerts. While you cannot override parent alerts like you can do for sensors (due to the complexities of alerts), there is a clone button provided to make it easy to clone and modify an inherited alert if changes are required.

Once alerts are defined, the InformixHQ server will evaluate the applicable alerting condition for each new data point collected by the InformixHQ agent in the process of monitoring your database server. If an alerting condition is met, an

alerting incident is created. Alerting incidents are automatically displayed in the InformixHQ UI, both in the dashboard and the **Incidents** pages for the server and for the parent group(s) it belongs to.

You can enable alert notifications to be sent through email, PagerDuty, Twilio, or a custom alerting script. To enable alerting notifications, a system administrative user for InformixHQ must enable the desired alerting notification service on the System Settings->Alerting Configuration on page 113 page. For certain alerting notification services, including email and Twilio, each individual user who wants to receive alerting notifications must enable it for their user on the My Settings->Alerting Configuration on page 116 page where they will provide user specific settings like their email address or phone number.

Custom Dashboards

Creating a custom dashboard

You can create custom dashboard of sensor metric or custom query for one or more Informix database servers.

Dashboards can be created for a single server or for multiple servers. Multi-server dashboards can have up to 5 different servers. All dashboards will be saved in the current group. This allows you to open and view the dashboard for different database servers within that group, although you have the option of saving a default server or set of servers for any particular dashboard.

To create a custom dashboard:

1. Click on **Dashboards** from the sidebar menu for a server or a group.
2. Click the **New Dashboard** button.
3. Select a server or set of servers which will allow you to preview the dashboard as you build it.
4. Add one or more dashboard panels.

   Each dashboard panel can be defined to graph one or more sensor metrics from the repository. You can also define a graph based on custom query.

   You can graph multiple metrics from multiple different sensors on the same graph. For multi-server dashboards, you can choose to associate each panel to a single server or you can graph metrics for each of your servers in the same graph.

   You can customize the graph types (bar, pie, line and table), attributes of the left and right y-axes, colors and labels of graphed metrics.
5. Once you add graph/data source in the panel, it mandates to save the panel to apply the changes. You can also change the name of the panel by clicking on the pencil icon.
6. Arrange your dashboard panels. Panels can be re-sized and moved to different positions to create a custom layout for your dashboard.

   You can view and save your dashboard based on the timestamp selection also. Once you select your desired timestamp, the save button will be enabled/disable based on the action you performed with the timestamp selection.

   As you make changes to your dashboard throughout the process of defining and editing, it is automatically saved.

Viewing a custom dashboard

To view a custom dashboard:

1. Go to the **Dashboards** page of the group or server within the group.
2. Select the desired dashboard from the list of available dashboards.
3. Optionally, select or change the servers to show on the dashboard.

   If the selected dashboard has a set of default servers defined, it will open with the defined servers. Once opened however, you can change the servers shown in the dashboard by clicking on the server drop-down at the top of the dashboard.

   If the dashboard has no default servers defined, your context within InformixHQ will determine which servers get loaded into the dashboard when it opens. If you open a dashboard from the context of a server, then that dashboard will be automatically loaded for the current server. If you open a dashboard from the context of a group, then you will be prompted to select one or more servers from that group to show on a dashboard.

Importing a custom dashboard

To import a custom dashboard:

1. Go to the **Dashboards** page of the group or server within the group.
2. Click the **Import** button and select the JSON file that has been exported from Dashboards page in InformixHQ. New dashboard will be created with the same configuration that has been provided while exporting dashboards.

Exporting a custom dashboard

To export a custom dashboard:

1. Go to the **Dashboards** page of the group or server within the group.
2. Select the desired dashboard from the list of available dashboards.
3. Click the **Export** button. JSON file will be downloaded to system's Downloads folder.

Custom Reports

About this task

InformixHQ provides many built-in reports to monitor Informix server effectively. To provide further flexibility, there is an option of creating customized reports to meet specific needs of the users. Using the Custom Reports option, users can create a report from any query that they want to analyse the data for. Once such custom report is created, this report is made available in System Reports, from where users can easily access the report and analyse the data.

There are two options to create custom reports in InformixHQ:

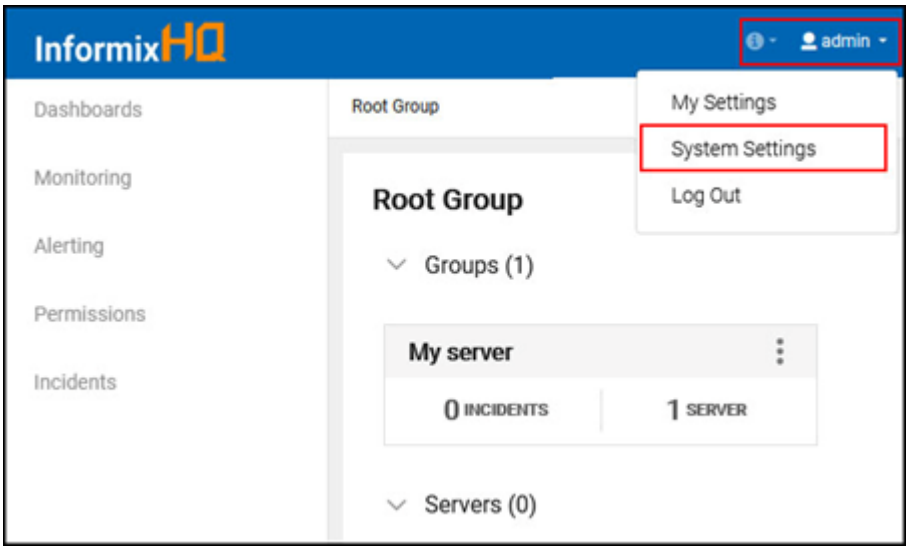
1. System Settings -> Report Management
2. Server Page -> System Reports

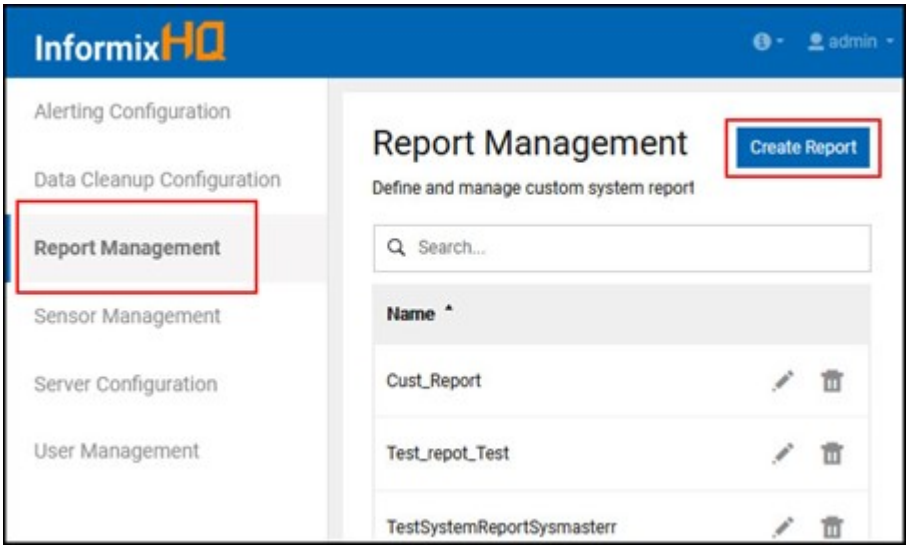
Note:

1. Custom report can ONLY be created by InformixHQ user with system administrator privileges.
2. Normal InformixHQ user (without admin privileges) created by admin user, will NOT be able to create new custom report. However, such user will be able to run custom report and view data.

To create a custom report:

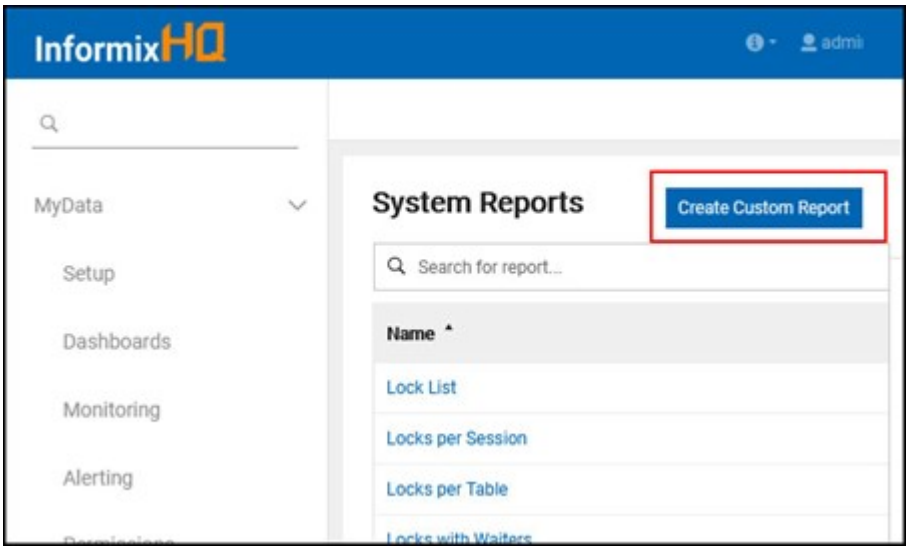
1. Login to InformixHQ as user 'admin' (any user with administrative privilege in InformixHQ)
2. Navigate to **System settings -> Report Management -> Create Report**



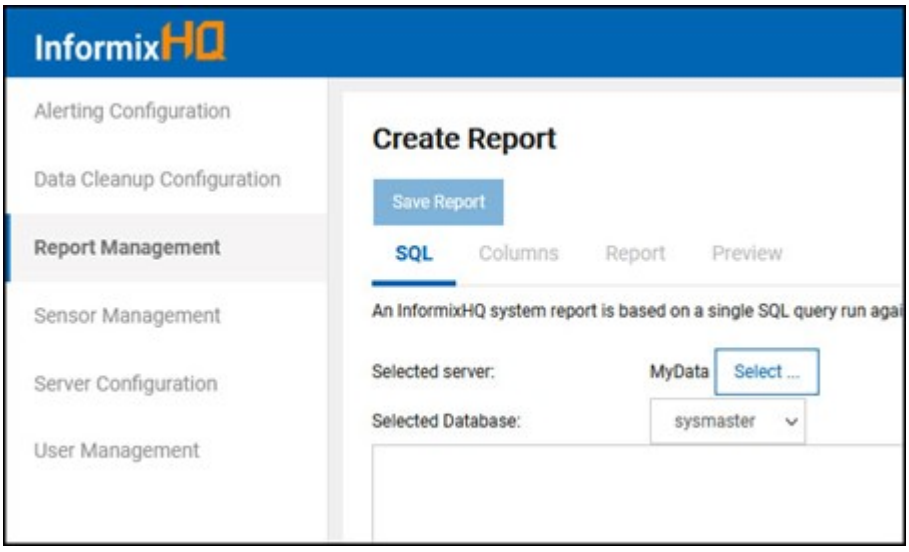


3. Alternatively, navigate to **Open Server page -> System Reports** from side menu.

4. Click on **Create Custom Report** button on **System Reports** page.

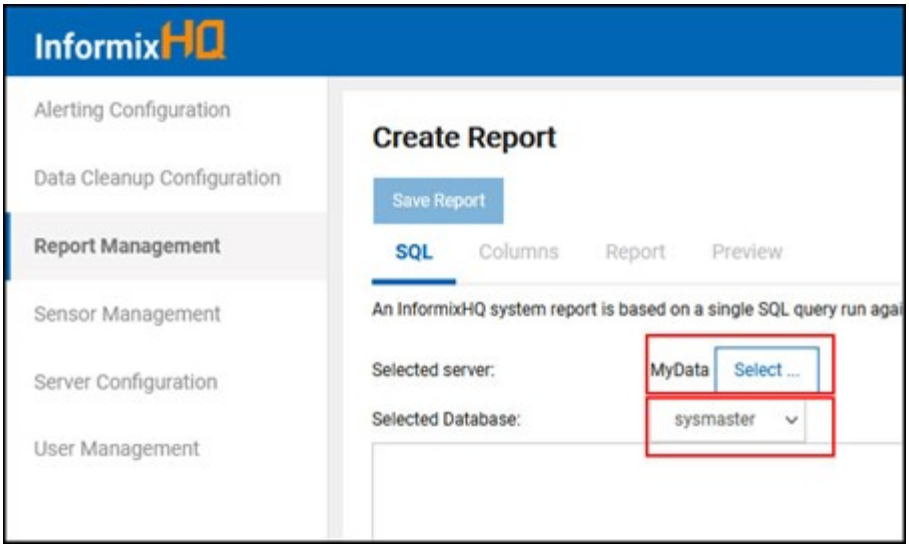


5. After clicking on **Create Custom Report** button, user will navigate to **Create Report** page.

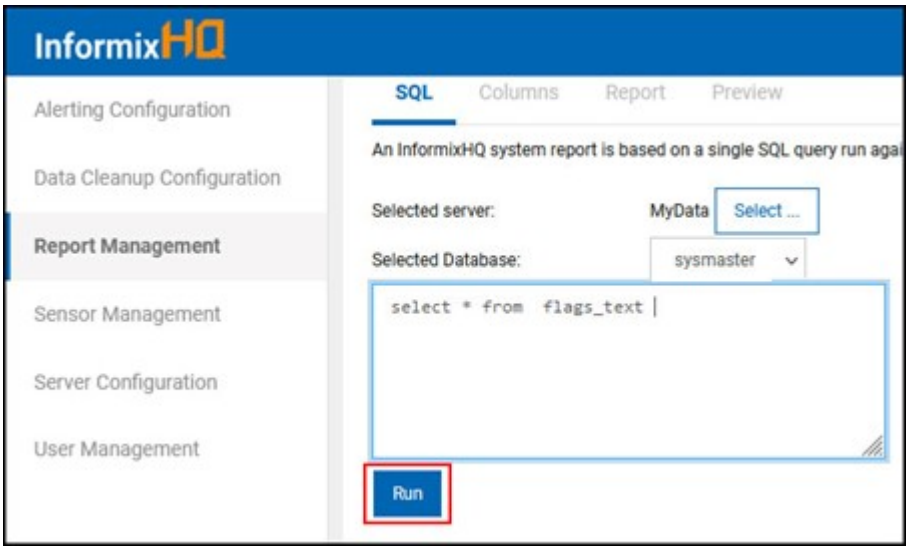


6. In SQL tab, select server from **Selected Server** dropdown & **Selected Database** from dropdown.

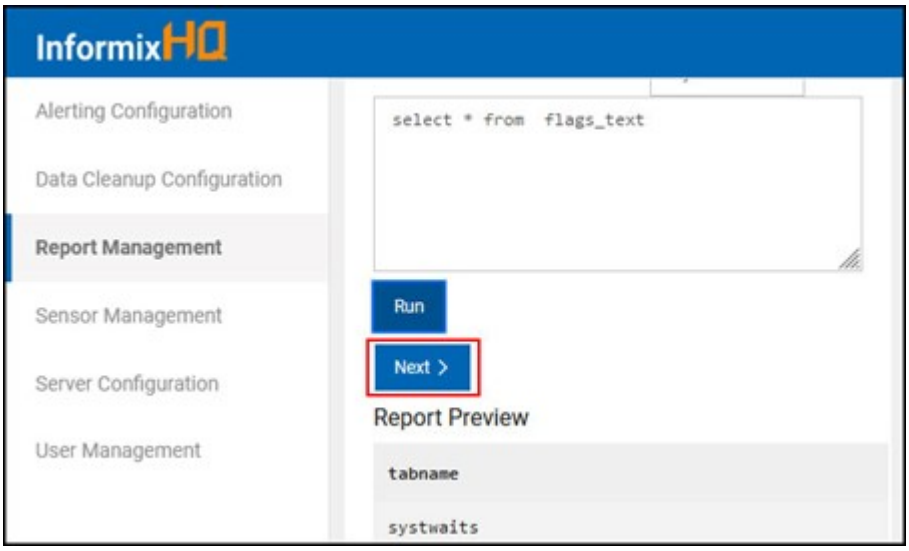
Note: While creating a custom report, user needs to select server and database to validate and run query to create the report.



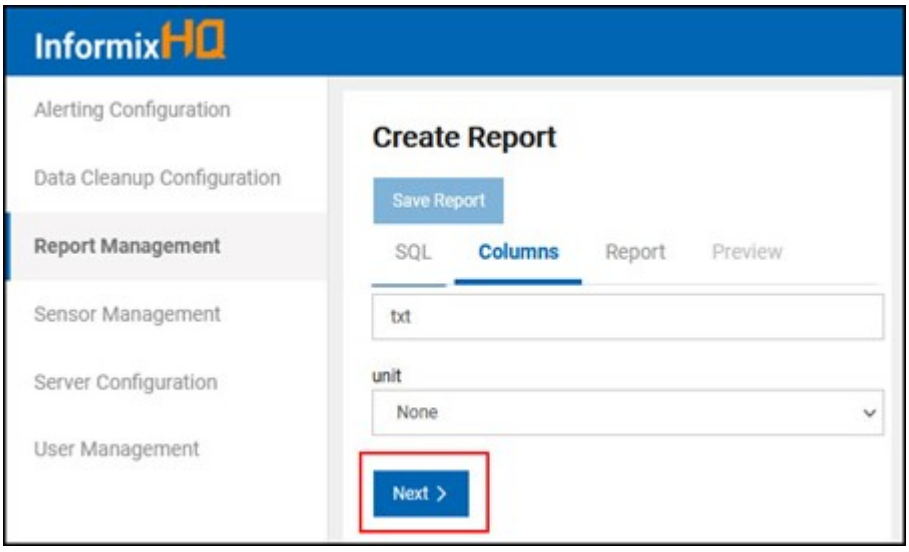
7. Type customized query in text area & click on the **Run** button.



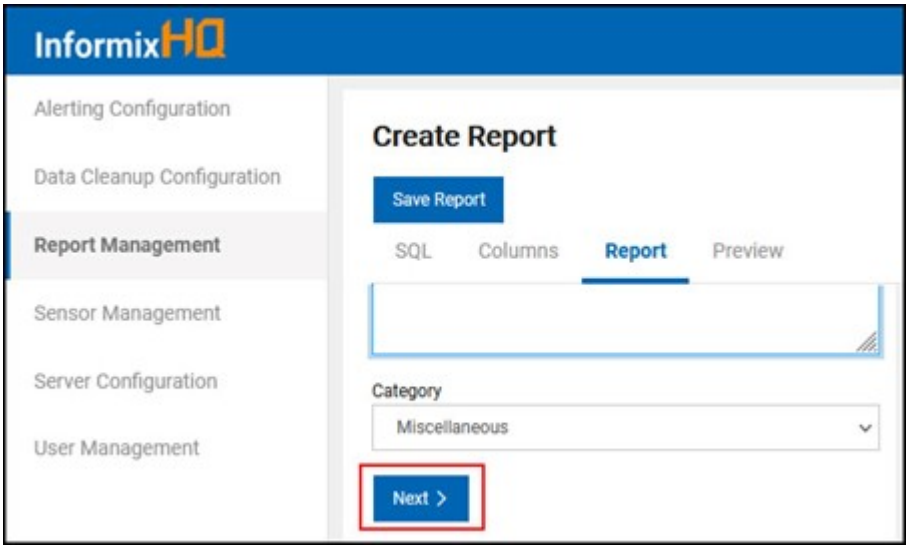
8. Output of the query will be shown after clicking **Run** button. Validate query result & click on the **Next** button.



9. On clicking **Next** button, screen will navigate to **Columns** tab.



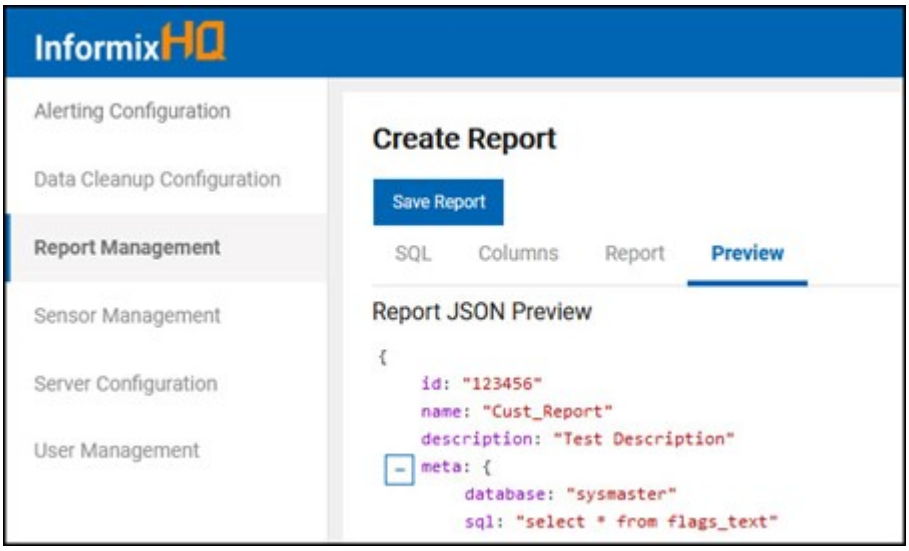
10. In **Columns** tab, there is an option for users to select specific column to show in a report or **Show ALL** option to include all the columns in a report.
11. Also, there is an option to rename columns (the way user wants them in a report) and select unit as desired. Click **Next** button.
    - **: Name** - The display name of this column.
    - **: Unit** - The unit determines how this column will be displayed in the report. A byte value is converted in human readable value (KB, MB, GB etc.). A unit time value (seconds since the unit epoch) will be converted in timestamp.
12. On clicking **Next** button, you will navigate to **Report** tab.



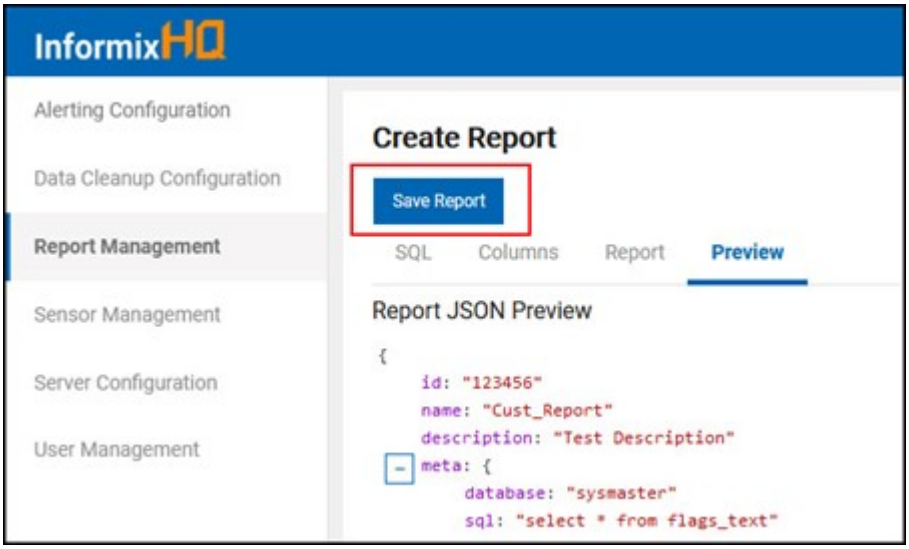
13. From **Report** tab, add ID, Name, Description & select **Category** from dropdown.
    - **ID** - The ID to uniquely identify this report. Can only use lowercase characters, digits, and single underscores (e.g.: my_custom_report)
    - **Name** - The display name of this report

- **Description** - Add detailed description of purpose of the report
- **Category** – Select report category e.g. lock , storage, miscellaneous, etc.

14. On clicking **Next** button, screen will navigate to **Preview** tab. **Preview** tab will show report query and columns in JSON format.



15. View and verify the changes on **Preview** tab, then click on **Save Report** button. If user wishes to change anything, navigate back to respective tabs, do the correction and come back to the **Preview** tab.

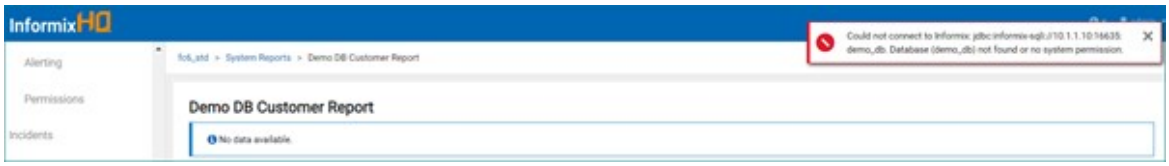


16. On clicking **Save Report** button, you will navigate back to **System Reports** page.
17. Find newly created custom report on **System Reports** page.

Note:

- Though the report was created by selecting one server, once custom report is successfully created, this report will be available for all the servers added in InformixHQ.
- In case, the database used to create custom report does not exist (or doesn't have permissions) on another server and user tries to run custom report on such server, InformixHQ will report an error similar to this:

  "Could not connect to Informix: jdbc:informix-sqli://<server>:<port>:<database>. Database (<database>) not found or no system permission."

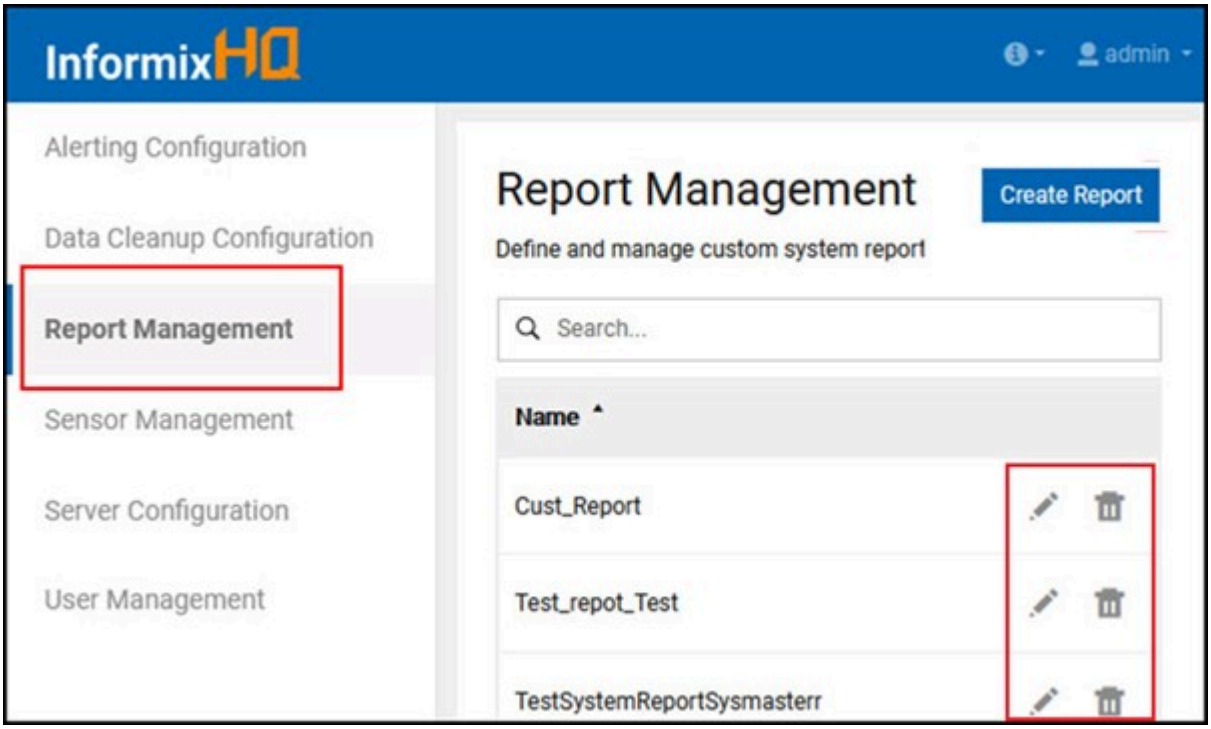


Editing and deleting custom reports

About this task

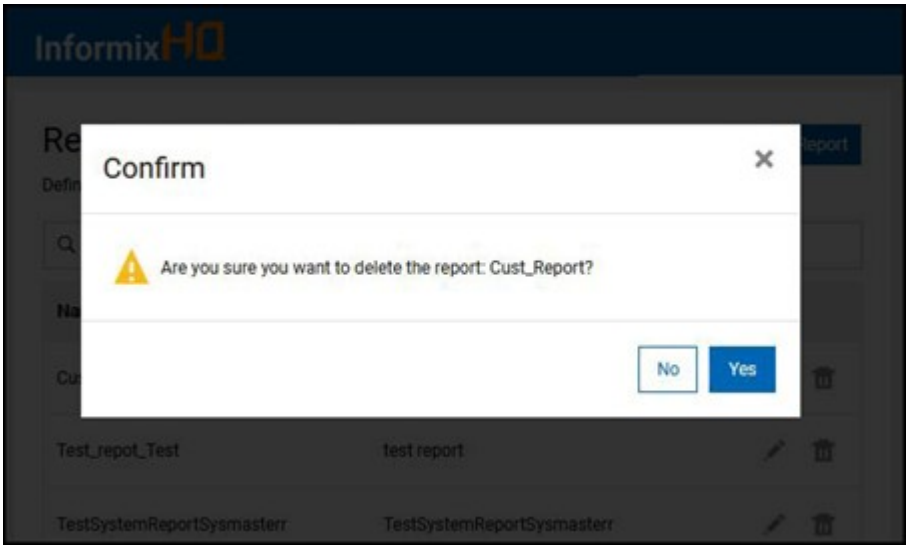
It is very easy to modify or delete custom reports.

1. Similar to create report, admin user needs to login into InformixHQ and navigate to Report Management. **System Settings-> Report Management**, where user can find list of all the custom reports created. **Edit** and **Delete** icons are available for each of the custom reports.



2. If user chooses to edit, same navigation will have to be followed as that of **Create Report** as detailed out above.

3. If user chooses to delete, there will a confirmation pop-up, if clicked yes, report will be deleted.



Sensors

Sensors in InformixHQ

Sensor : A sensor defines a metric or set of metrics for the agent to gather. An example is the “DBSpace Usage” sensor that gathers metrics on used and free space for all database server spaces.

Sensors are provided in InformixHQ to collect the data for metric which users want to monitor over a period. Once Sensor is configured, and InformixHQ agent is configured and Online, Sensor data will be collected at specified interval and stored in Repository database in Informix Server.

List of InformixHQ built-in sensors

Sensors in InformixHQ

InformixHQ provides many built-in sensors to monitor Informix server performance. User can configure a sensor from the list of built-in sensors or create a custom sensor based on a specific requirement. User can also monitor all the built-in sensors' list or custom sensors by adding them to the Dashboard or by creating alerts for them, and a few sensors can be monitored in HQ screens, which are mentioned in the table given below.

By default, there are no sensors added to InformixHQ for an instance of Informix Server. Users need to add sensors as per the monitoring requirements for each of the Informix servers defined in HQ.

When sensors are added to InformixHQ, corresponding tables in the repository database are created to store the sensor's data. These tables are named as **<s>_<informix_server_id>_<sensor_name>**.

For example, if an AF files sensor is added to a server instance with the Informix server ID "1," (Informix server ID can be found on the server setup page or in the URL), then the table name in the repository database will be *"s_1_af_files,"* where "s" stands for sensor.

All the information regarding sensor tables and their storage details (like the number of rows, space utilization, size, etc.) is available on the Tables & Indexes page under the Storage menu.

Use the following steps to view the sensor data storage information:

1. Go to **Storage -> Table & Indexes** page in InformixHQ.
2. Select **Dbspace -> select repository dbspace**.
3. Select **Database -> Select repository database from dropdown**.
4. All the storage details regarding sensor will be shown in a table format.

Serial No.	Sensor Name	Type	Description	Sensor data availability on the InformixHQ screen
1	AF files	files	Monitors for the presence of AF (assert fail) files in the Informix database server's DUMPDIR.	• Generated AF files' count can be seen on Server details page under status.
2	Informix Server Status	ifx_status	Monitors the status of the Informix database server (online, offline, etc.)	
3	Operating System CPU	cmd	Monitors CPU usage on the operating system	• On Server details page • On System Resource page
4	Operating System CPU per Core	cmd	Monitors CPU usage per core on the operating system	
5	Operating System Disk I/O	cmd	Monitors Disk I/O activity on the operating system	• On System Resource page
6	Operating System Disk Utilization	cmd	Monitors operating system disk usage for the storage devices used by the Informix database server.	• On Server details page • On Storage page

7	Operating System Memory	cmd	Monitors memory usage on the operating system	• On Server details page • On System Resource page • On Memory page under System Resource
8	Operating System Network I/O	cmd	Monitors network i/o activity on the operating system	• On System Resource page
9	Session Statistics	sql	Monitors the number of sessions connected to the Informix database server and the average and maximum amount of memory used by each session	• On Server details page
10	Checkpoint	sql	Monitors checkpoint statistics for the Informix database server	• On Server details page • On Checkpoints page under Performance
11	Memory	sql	Monitors memory usage by the Informix database server	• On Memory page under System Resource
12	Memory Segments	sql	Monitors the memory segments used by the Informix database server	
13	Buffer and Disk I/O	ifx_diskio	Monitors buffer and disk I/O activity on the Informix database server	• On Server details page
14	Chunk Writes	sql	Monitors chunk writes by the Informix database server	
15	Foreground Writes	sql	Monitors foreground writes by the Informix database server	• On Server details page
16	LRU Writes	sql	Monitors LRU writes by the Informix database server	
17	Sequential Scans	sql	Monitors the number of sequential scans performed on the Informix database server	• On Server details page
18	Thread Counts	sql	Monitors the number of Informix database server threads, keeping track of thread counts for each state (running, ready, waiting on mutex, etc.).	• On Server details page

19	Virtual Processors	sql	Gathers statistics about each virtual processor (vp) class.	
20	DBSpace Usage	sql	Monitors the space usage for dbspaces on the Informix database server	• On Storage page
21	Backups per dbspace	sql	Monitors the time of the last backups run for each dbspace on the Informix database server	• On Backups page
22	Online Log	log	Monitors the Informix database server's log file.	• In Online log, under Agent data
23	High Availability Workload	sql	Monitors the cpu workload % for each server in a high availability cluster. This sensor should be run on the primary server of a cluster.	
24	High Availability Lagtime	sql	Monitors the lagtime in seconds for each server in a high availability cluster. This sensor should be run on the primary server of a cluster.	
25	High Availability Connection Status	sql	Monitors the connection status for each secondary server in a high availability cluster. This sensor can be run on any server of a cluster.	
26	High Availability Logical Log Rate	sql	Monitors logical logs of a high availability cluster. This sensor should be run on the primary server of a cluster.	
27	High Availability Approximate Log Backlog	onstat	Monitors the approximate log backlog in seconds for each secondary server in a high availability cluster. This sensor should be run on the primary server of a cluster.	
28	High Availability Transaction Latency	onstat	Monitors the transaction latency in seconds for each secondary server in a high availability cluster. This sensor should be run on each secondary server of a cluster.	
29	High Availability Apply Rate	sql	Monitors the apply rate for each secondary server in a high availability cluster. This sensor should be run on each secondary server of a cluster.	

30	SLA Connections	sql	Monitors the number of connections for each SLA. This sensor should be run on those servers which are being managed by Connection Manager.

Note:

1. “sql”’ type of sensors collect database level information and run over jdbc connection with sysmaster database of monitored server.
2. “ifx_status” and “ifx_diskio” types of sensors collect database related information from server and runs over jdbc connection with sysmaster database of monitored server.
3. “cmd”, “files” and “log” types of sensors collect OS level information and run as command on host OS.
4. "Onstat" type of sensor collects data from onstat command and monitors based on that data.
5. The InformixHQ Agent can be started in two modes:
   - **Local mode:** It means that the InformixHQ Agent is running on a host machine for Informix Server, where all types of sensors can be run i.e. server-level sensors (sql and ifx) as well as OS-level sensors (cmd, files, and log).
   - **Remote mode:** It means running the InformixHQ Agent from another machine that is not an Informix Server host machine, where only server-level sensors (sql and ifx) can be run due to a limitation that OS-level sensors (cmd, files, and log) can not log data since data is collected by executing commands on the host OS for these types of sensors. For more information about Remote mode, see InformixHQ Remote Agent mode on page 127

Sensor Table Size Calculation

Sensor data can accumulate rapidly, especially in environments where multiple sensors are collecting data at high frequencies. Sensor data often varies in structure and content based on type of sensor. Calculating the size of a sensor table helps users in storage estimation. Calculating the size of row involves understanding the typical columns and their potential data types.

Note: The overall size used by sensor tables is totally dependent on how many sensors are activated by the user and the frequency at which those are executed.

Table structure:

- **Timestamp:** Records the time when the sensor reading is taken.
- **Data column:** It varies widely based on the type of sensor and the parameters being used to measure the data.

For example, in case of Chunk Writes sensor, parameters are chunkwrites_10K, chunkwrites_2K. Checkpoint sensor parameters are cp_time, ckpt_logid, physused, logused, etc.

- **Primary Key:** It stores primary key (a few sensors include primary key).

Data types and size:

- **Timestamp:** Type - BIGINT, Size - 8 bytes
- **Data:** BSON documents up to 4 KB are stored in-row.

  Type - BSON, Size - 4096 bytes + 4 bytes (4 KB) = 4100 bytes

  It takes 4 bytes extra to store BSON data. JSON and BSON documents up to 4 KB are stored in-row. Documents that are greater than 4 KB in size, are stored in the sbspace that is associated with the table, or the default sbspace if the table does not have a designated sbspace. The maximum size of a JSON or BSON document is 32 KB.

- **Primary Key:** Stores character strings of varying length (up to 255 bytes).

  Type - VARCHAR, Size - 255 bytes

  The maximum length of a VARCHAR column is the defined length (255) plus 1 byte to store the length of the actual value.

Calculate row size using SQL query:

1. Query to get row size of Checkpoint sensor:

   ```
   SELECT rowsize FROM systables WHERE tabname = 's_1_checkpoint';
            4108 bytes
   ```

2. Query to get row size of Chunk Writes sensor:

   ```
   SELECT rowsize FROM systables WHERE tabname = 's_1_chunkwrites’;
            4108 bytes
   ```

Note: The table name used to store sensor data is dynamically generated based on InformixHQ database server id.

Estimating row size:

Following example explains row size calculation:

If two sensors are added , "Chunk Writes" and "Checkpoint", for 1 hour (60 min) and the run interval is 15 minutes, database will have 4 records for each sensor after 60 minutes.

- Row size calculation without primary key:

  ```
  row size = timestamp 8 bytes + data (4096 bytes + 4 bytes) = 4108 bytes per row
  ```

Hence, Table size for 1 hr (60 min) = 4 * 4108 = 16432 bytes

- Row size calculation with primary key:

```
Row size = timestamp 8 bytes + data (4096 bytes + 4 bytes) + primary key (255 bytes + 1 byte) = 4364
 bytes per row
```

Table size for 1 hr (60 min) = 4 * 4364 = 17456 bytes (in case of primary key)

Note: In case of a primary key in the table, row size is 4364 bytes.

Creating sensors from InformixHQ built-in sensors

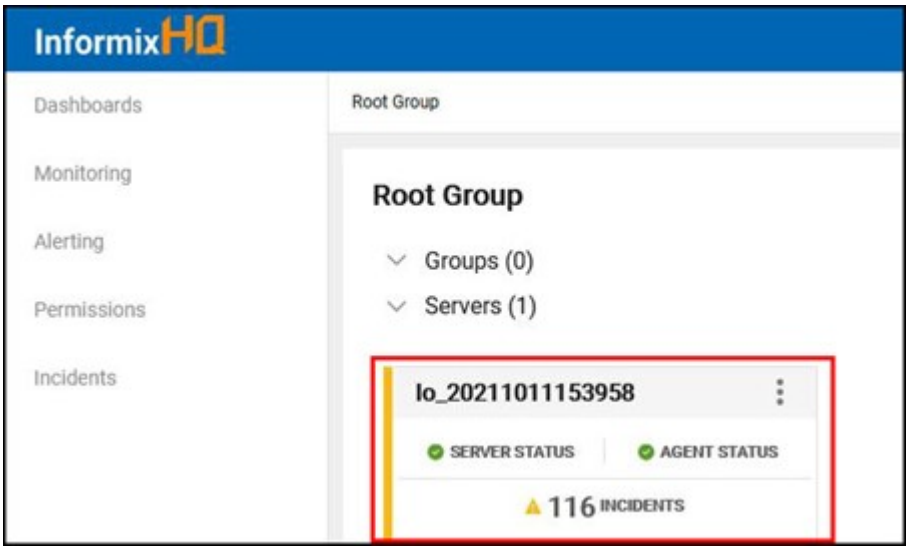
Using InformixHQ, sensors can be added in two ways:

Adding a sensor for a specific Informix server

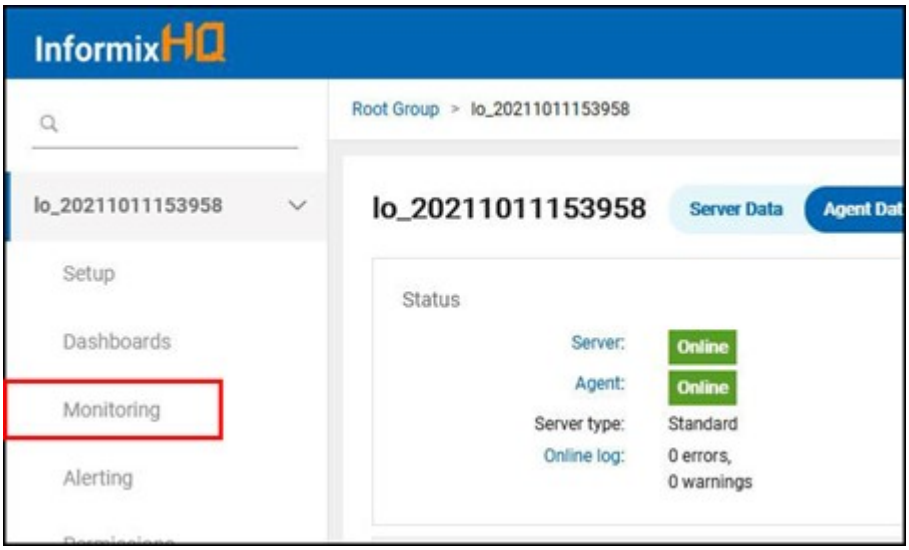
About this task

Follow given steps to add a sensor for a specific Informix server:

1. Click on **Server card** to navigate to **Server Dashboard** page.



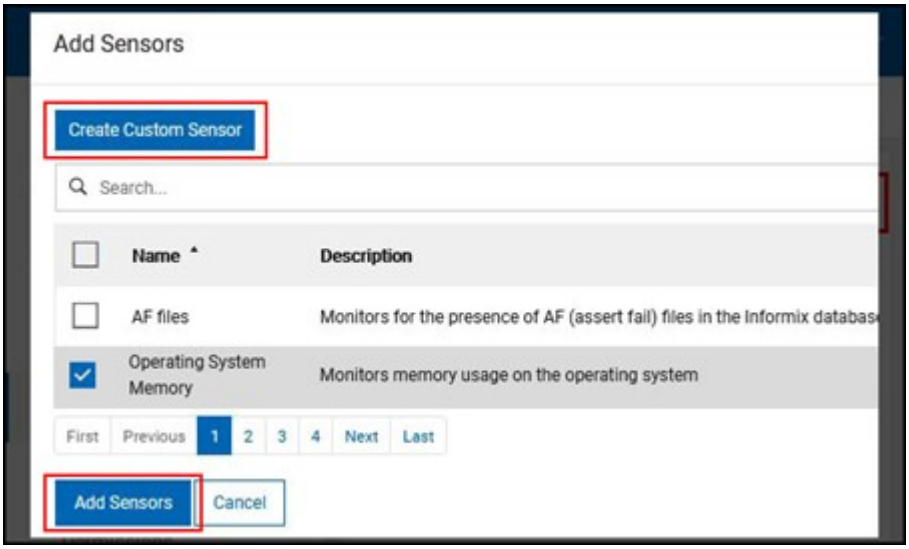
2. From the **Server Dashboard** page, click on **Monitoring tab** from the menu.



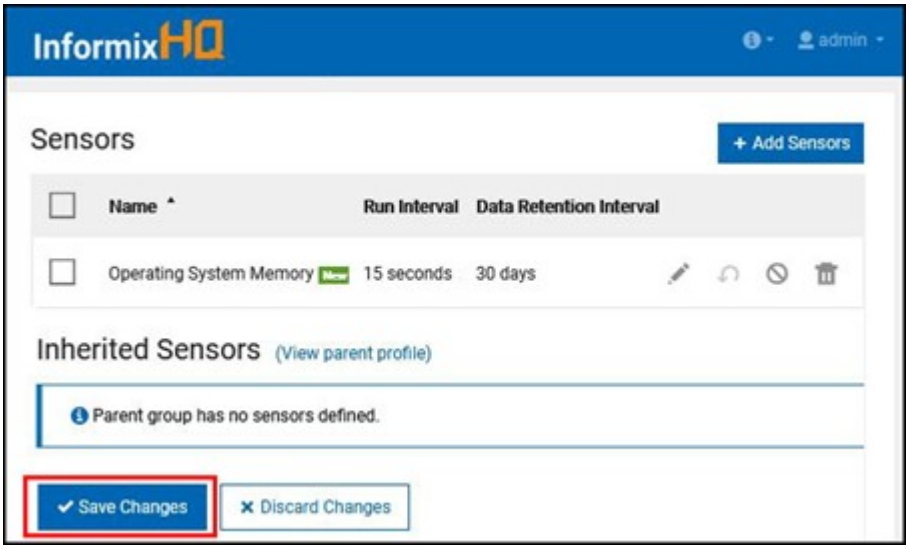
3. Click on **+ Add sensors** button.

4. By clicking on **+ Add sensors** button, a model appears with available built-in sensors which is by default provided by InformixHQ .

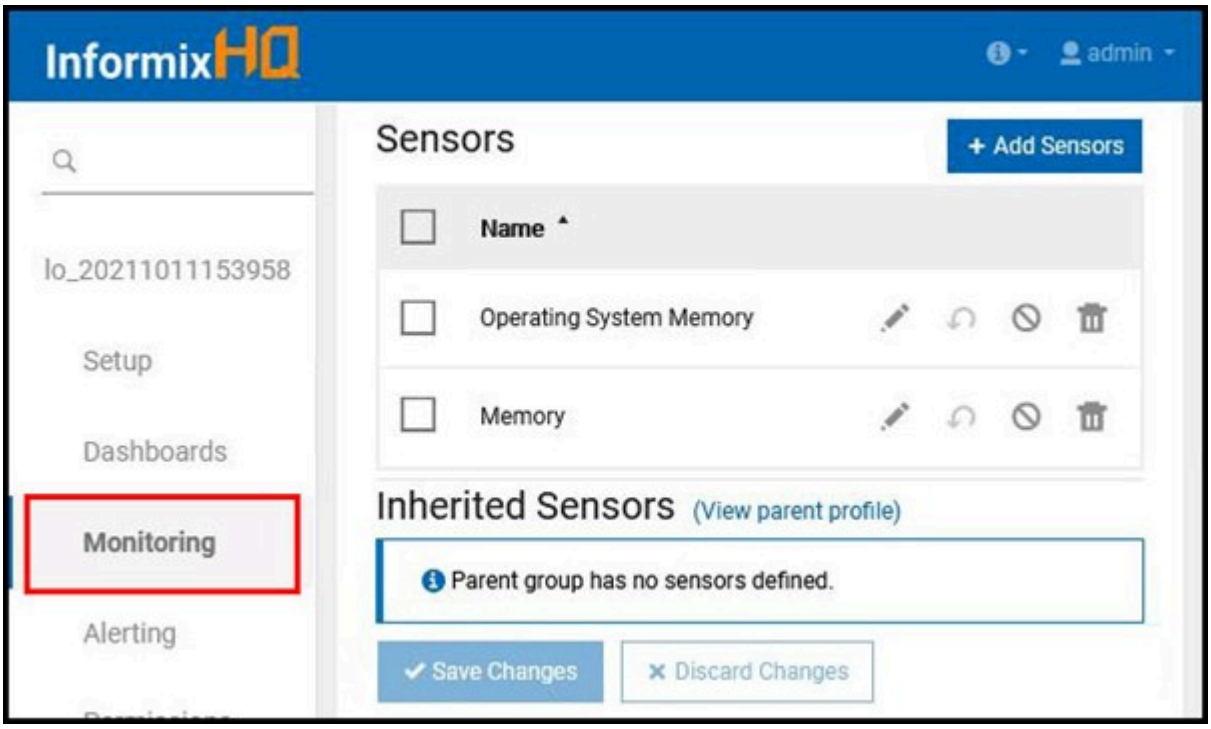
Note: User can add multiple sensors based on the requirements or can also add custom sensors by clicking on **Create Custom Sensor** button. For more details on creating custom sensors, refer Creating custom sensors.



5. Select sensors from existing sensor list to monitor your Informix server performance from the model window & then click on **Add Sensors** button.
6. Once sensors are added based on the requirements, click on **Save Changes** button to save all added sensors.



7. After saving changes, on **Monitoring** tab, user can view all added sensors.



Note: On **Monitoring** tab, user can view all added sensors & custom added sensors if any sensors already exist for selected database or root group.

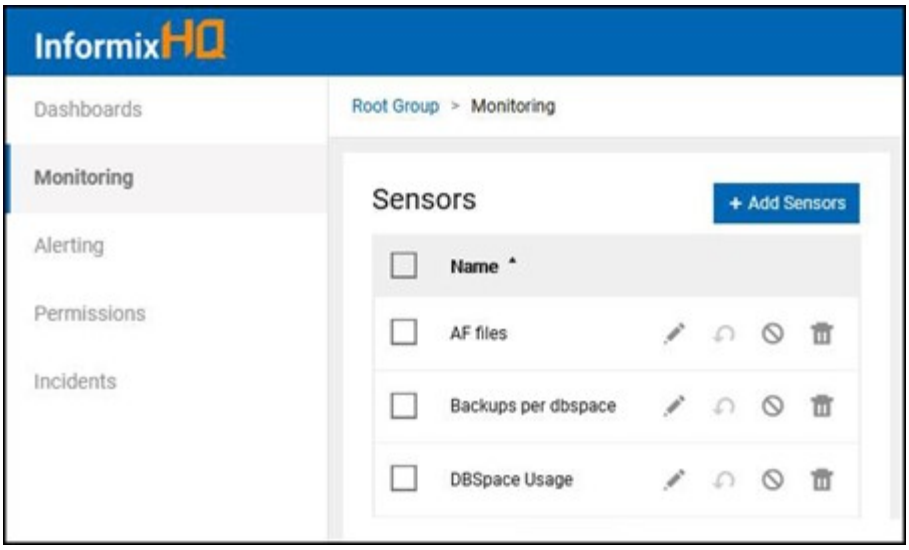
Adding a sensor on a root group

About this task

Follow given steps to add a sensor on a root group:

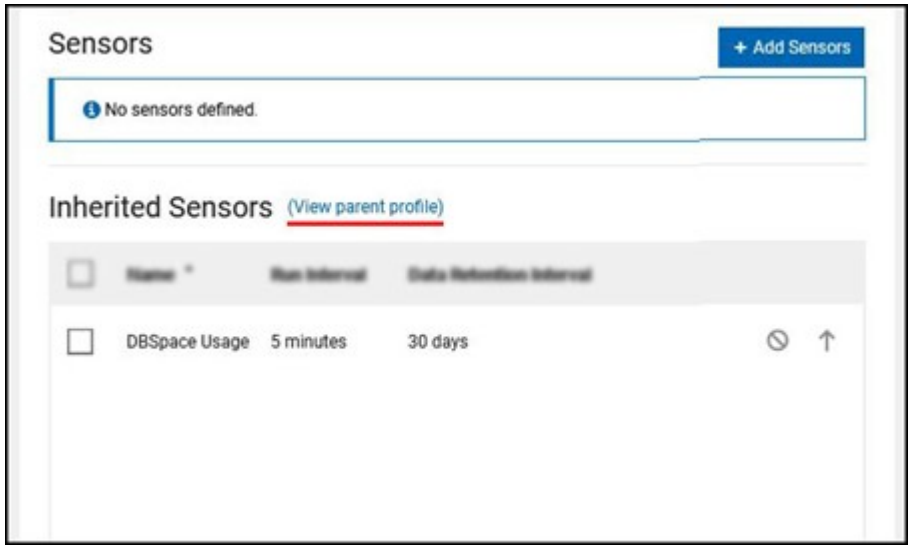
1. To add a sensor for all available Informix servers at root group, click on **Monitoring** tab and then click on **+ Add Sensors** button from root group.

2. All added sensors can be viewed from **Monitoring** tab for root group.



Note: Once sensors are added in root group, all root group sensors will be available for each of the Informix server in InformixHQ.

3. To view all available sensors for root group from a specific server , click on **View parent profile**. **View parent profile** link will navigate to **Monitoring** page on root group.



Overriding a sensor for a specific Informix Server

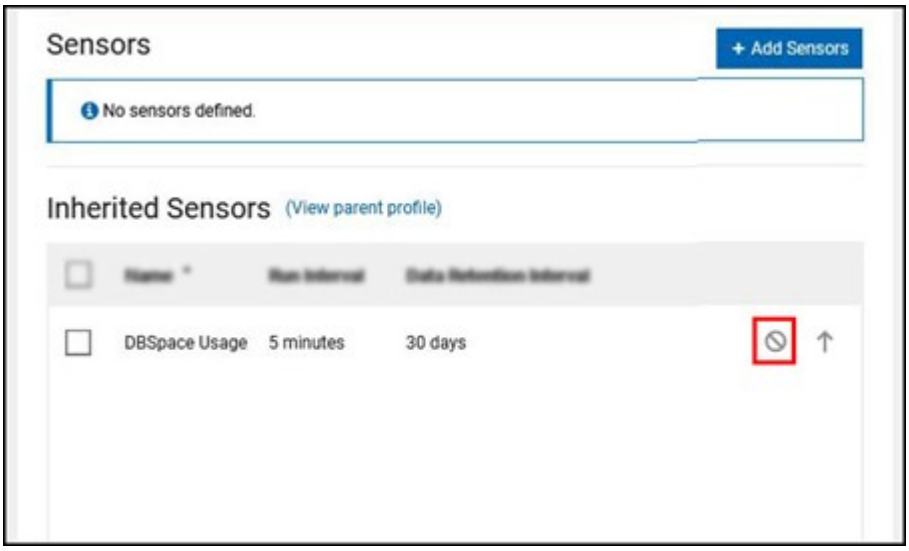
About this task

Based on the requirements

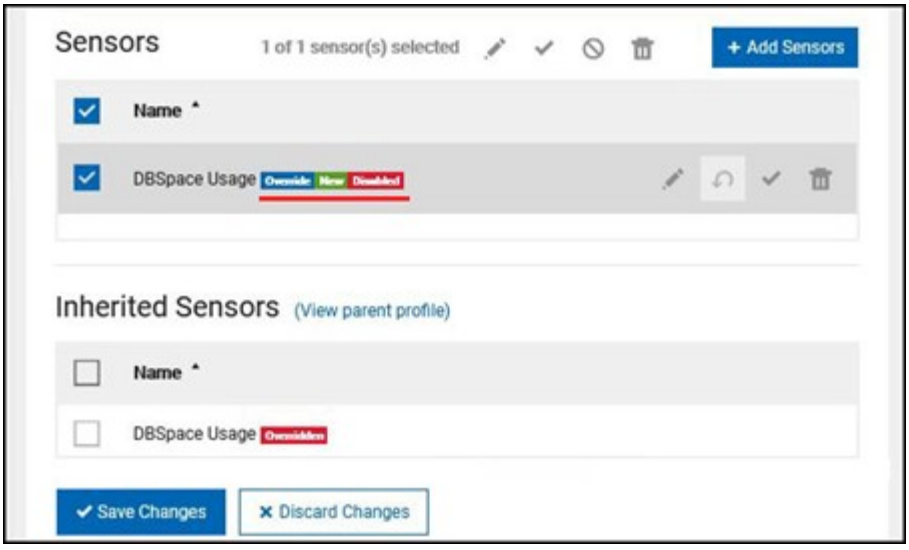
1. sensors can be inherited from root group
2. sensors can be overridden for a specific Informix server

1. Select server card as per the requirement.
2. Click on **Monitoring** tab menu.

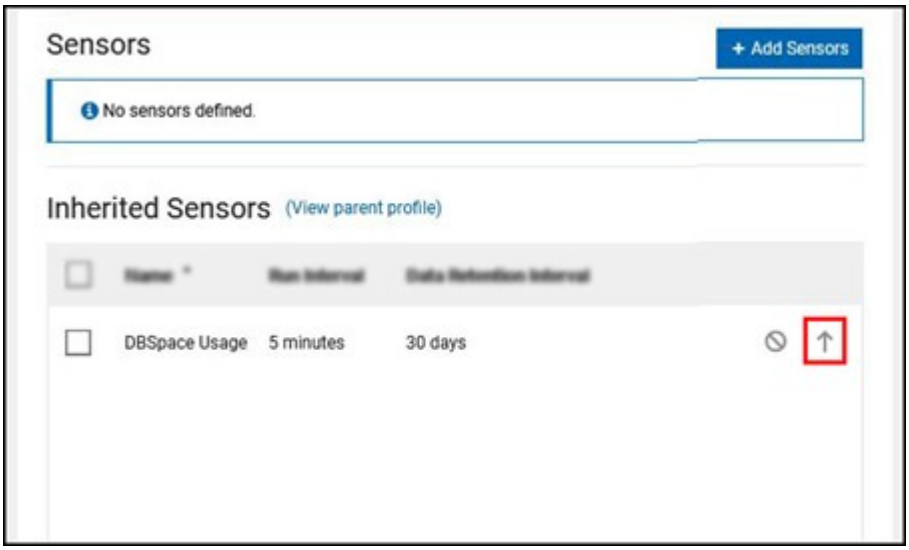
Note: User can view all available sensors for root group in **Inherited Sensors** section. User can also view sensors in **Sensors** section, if sensors already existed for a specific Informix server.



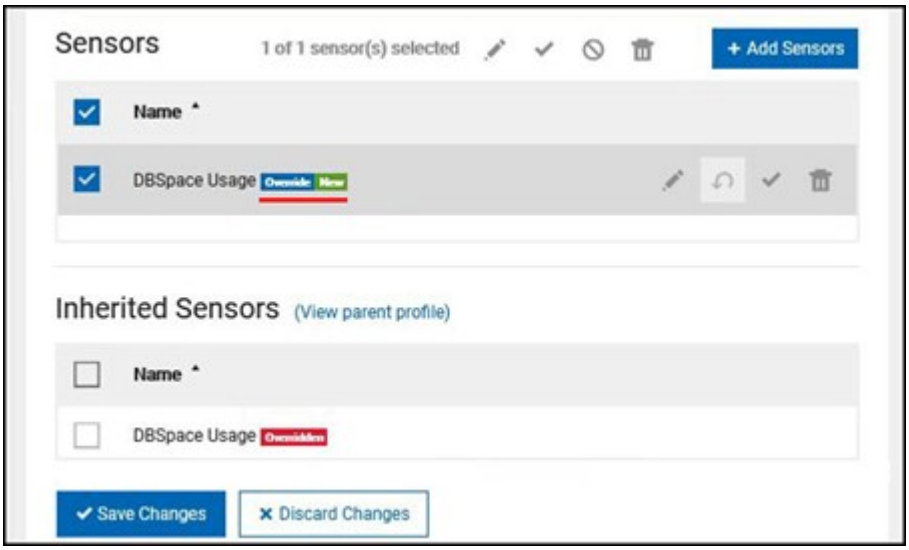
3. Clicking on **Override & Disable icon** will make inherited sensor override for your specific server with disabled.



4. Clicking on **Override** icon will make inherited sensor override for your specific server with enable state.



5. After clicking on **Override** icon,



Note: All override sensors checkboxes are disabled after overriding for a specific server in **Inherited Sensors** section. If user deletes any inherited sensors from **Sensors** section, those sensors will be available in the inherited sensors again in enable state.

Modifying sensors

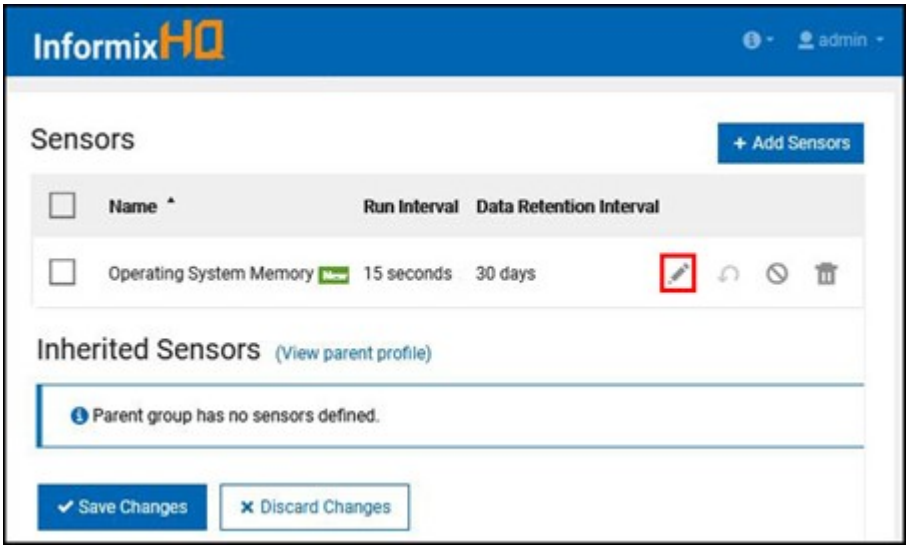
Using InformixHQ, user can take following actions related to the sensors:

Editing sensors

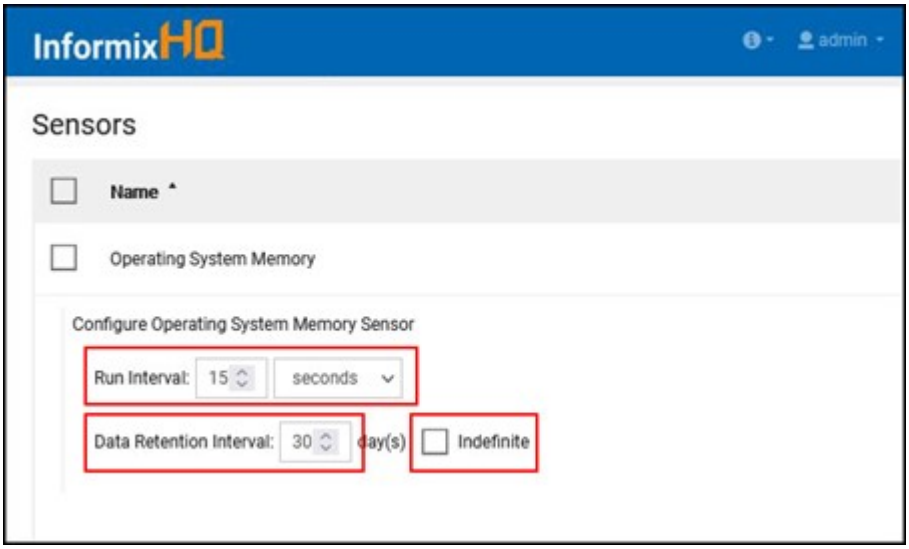
About this task

To edit an added sensor follow the given steps:

1. Click on **Pencil** icon.



2. Default values for parameters are provided by the InformixHQ. User can modify those as needed.
    - **Run Interval :** A Sensor will run after specified time period (based on the selection), e.g. seconds, minutes, hours, days, weeks, months.
    - **Data Retention Interval Days:** InformixHQ will retain collected sensor data for specified number of days from the date of collection, after which data will be deleted or
    - **Indefinite :** InformixHQ will retain data forever without deleting it.

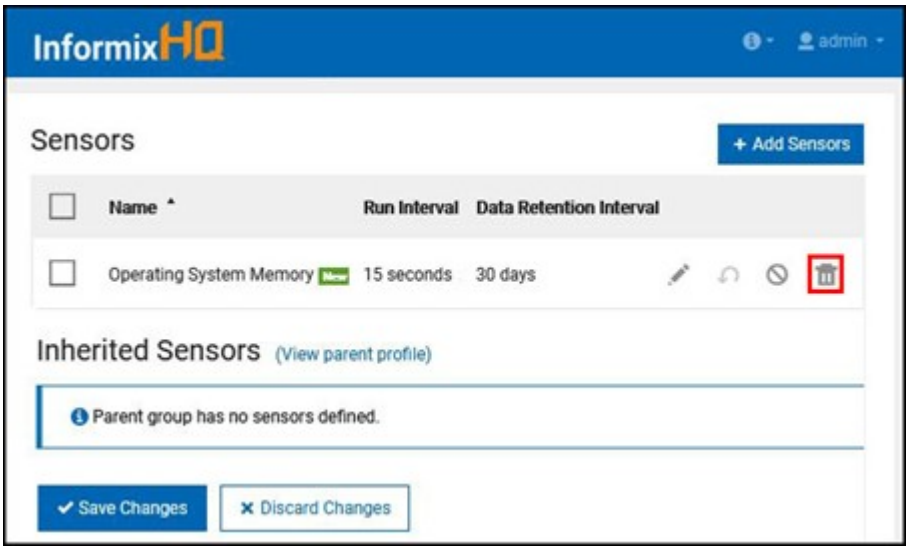


Deleting sensors

About this task

To delete an added sensor follow the given steps:

Click on **Delete** icon. User can delete multiple sensors also.

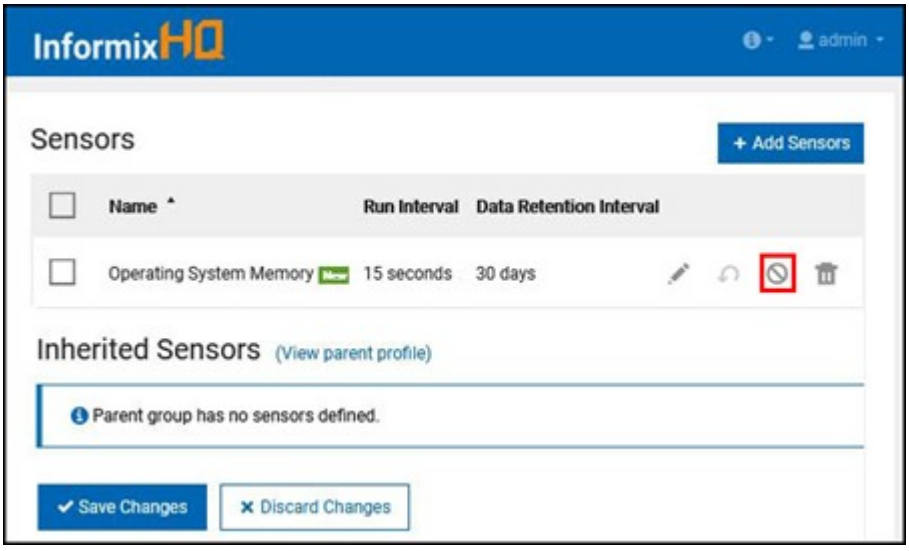


Enabling/Disabling sensors

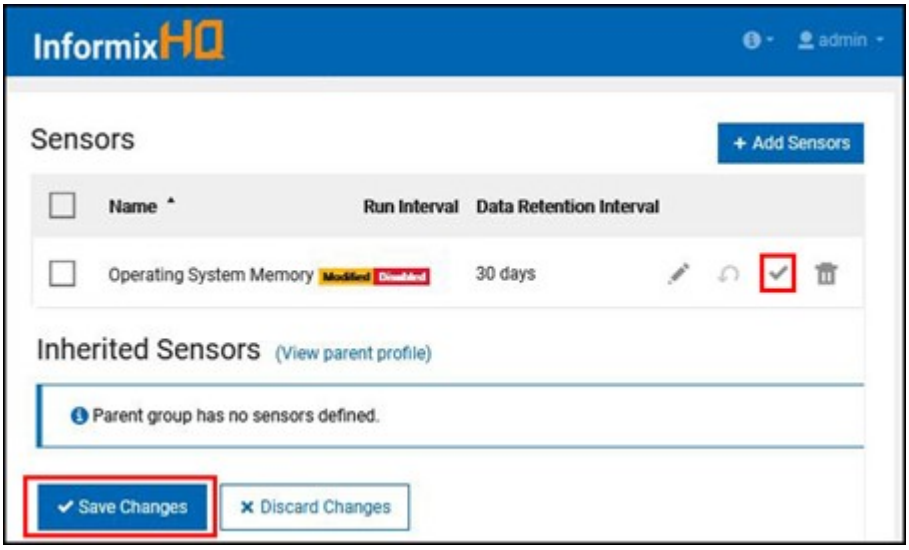
About this task

Follow the given steps for enabling/disabling an added sensor:

1. Click on **enable/disable** icon. Check images given below carefully to notice different icons based on enable/disable actions. Based on the modified state, InformixHQ notifies the user by putting tag next to sensor name.
2. Save changes after modification by clicking on **Save Changes** button. **New** tag next to the sensor name indicates a newly added sensor as shown below.



3. Similarly, **Modified** tag indicates a modified sensor.

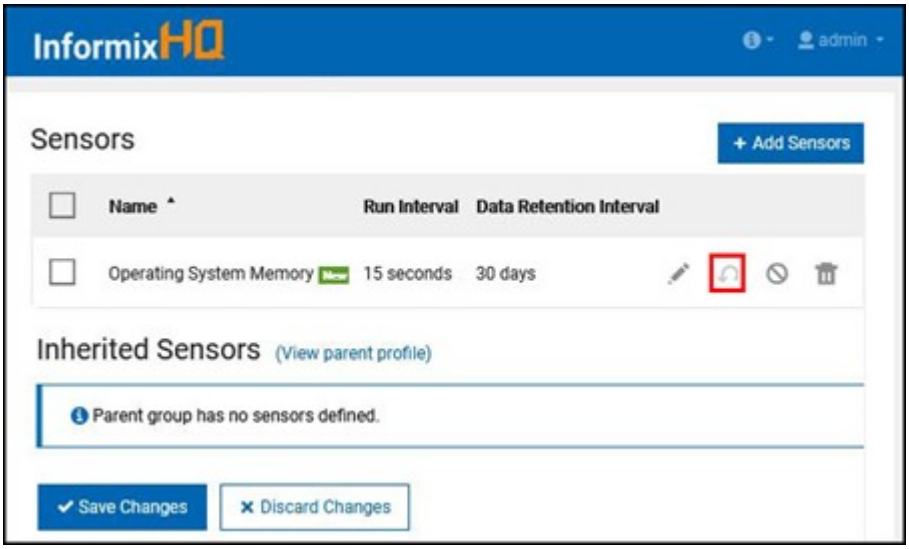


Undo editing done on sensors

About this task

To revert back to the default InformixHQ sensor parameters, follow the given steps:

Click on **Undo edits** icon.



Creating custom sensors

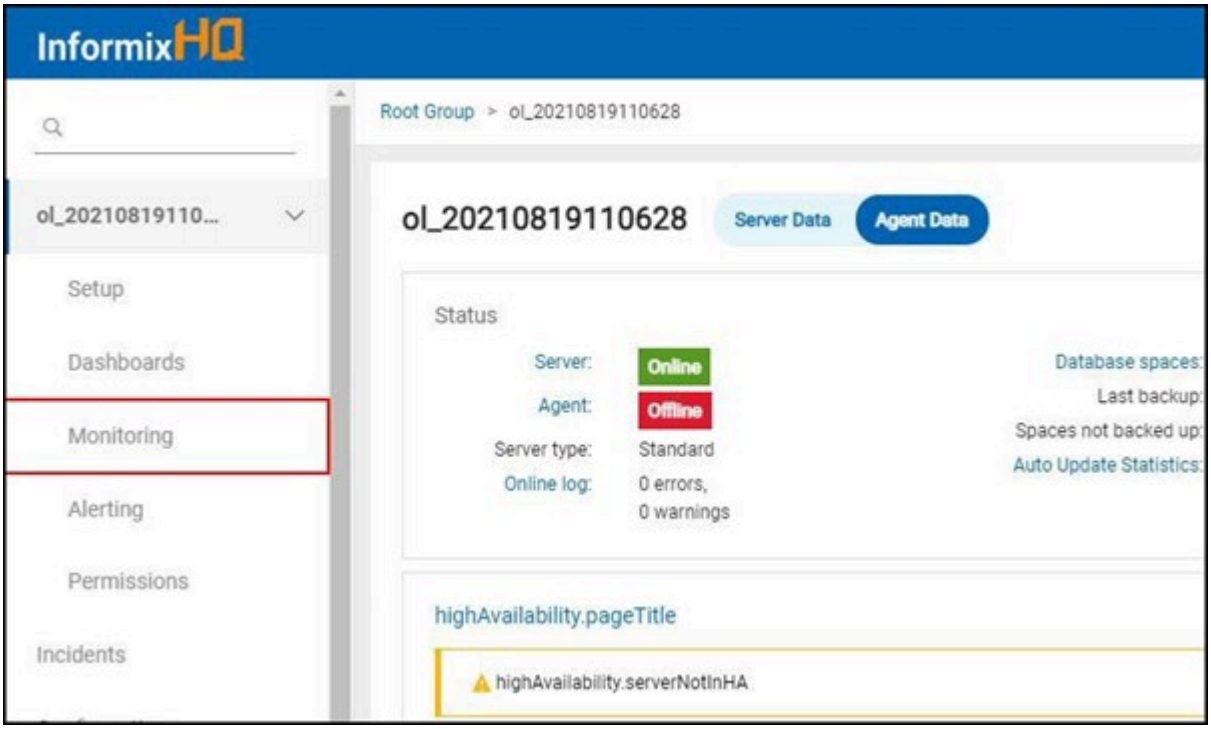
This topic explains how to create a custom sensor using InformixHQ. A custom sensor can be created in the following two ways:

Creating a custom sensor using Monitoring option

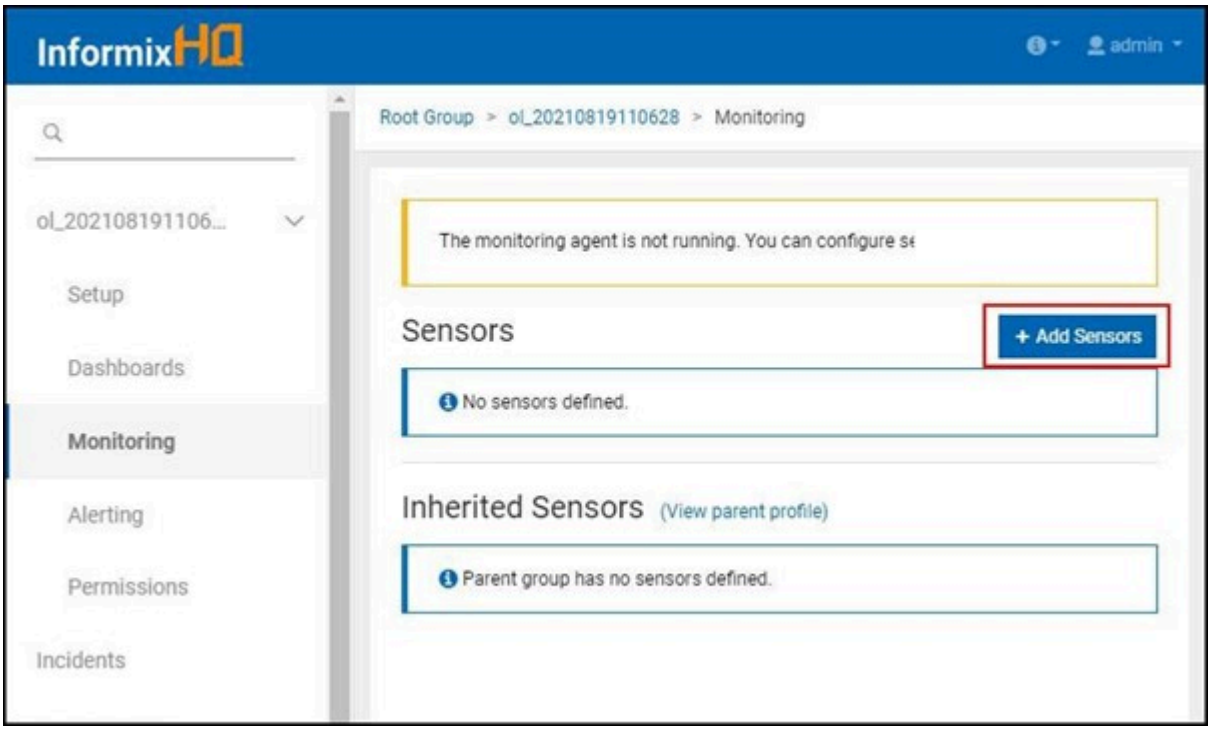
About this task

This topic explains how to create a custom sensor using InformixHQ Monitoring option.

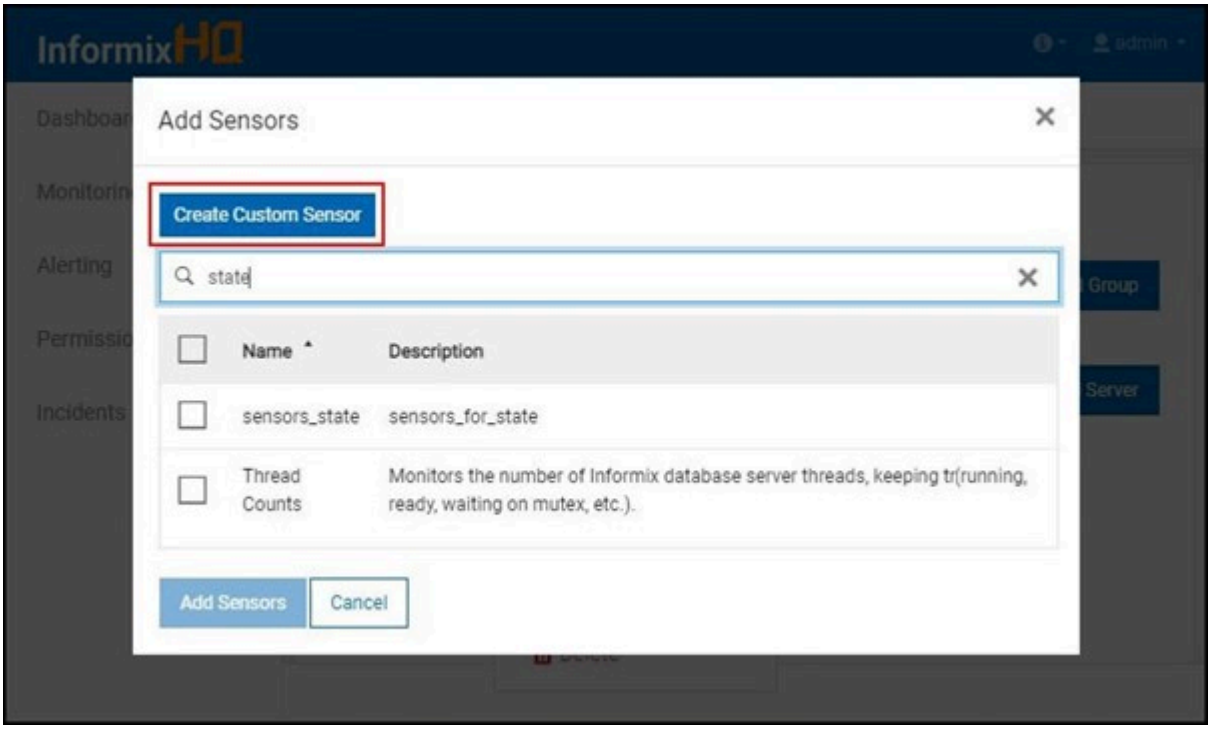
1. Select the desired server from InformixHQ.
2. Click on **Monitoring** option from side menu.



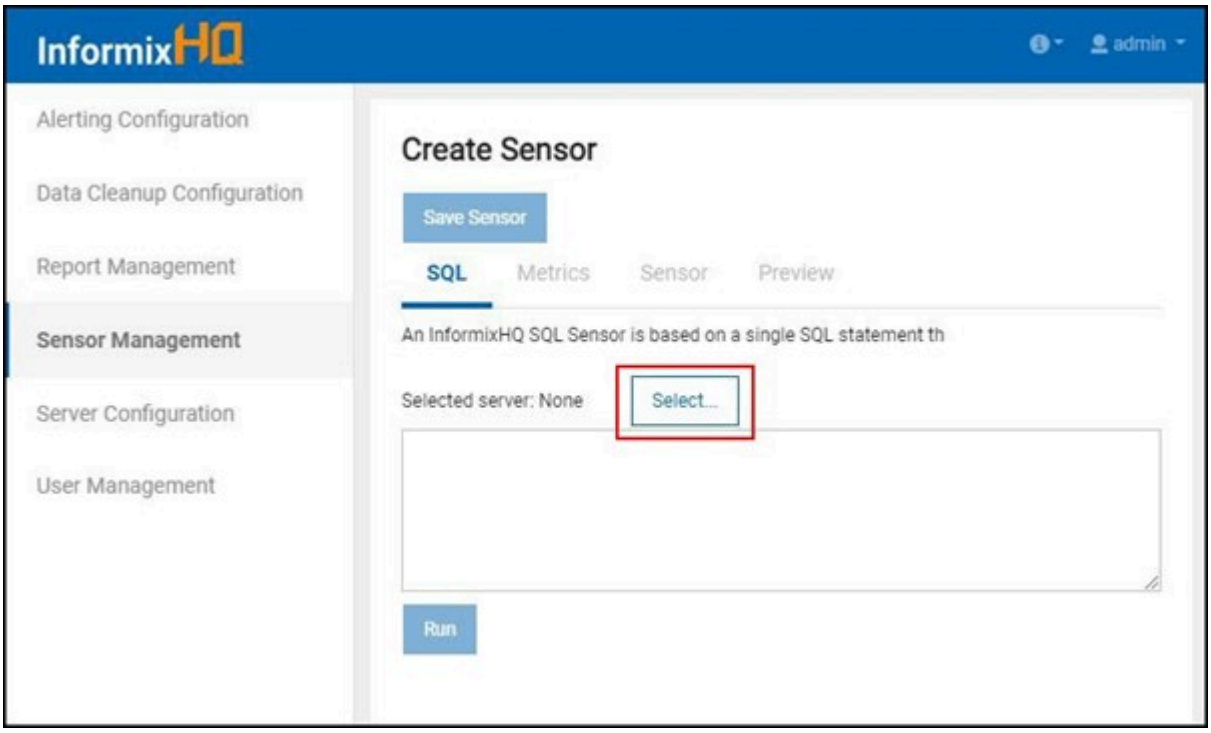
3. Click on **+ Add Sensors** button.



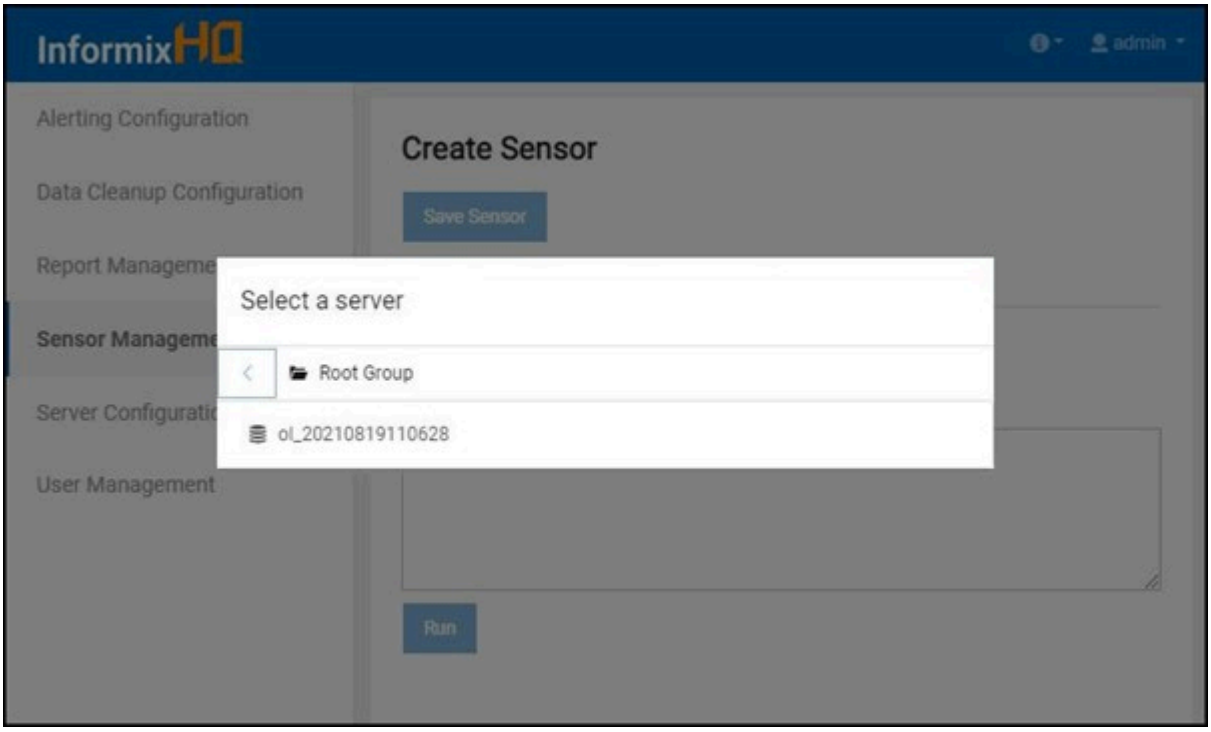
4. **Add Sensors** pop up appears. For creating a custom sensor, click on **Create Custom Sensor** button.



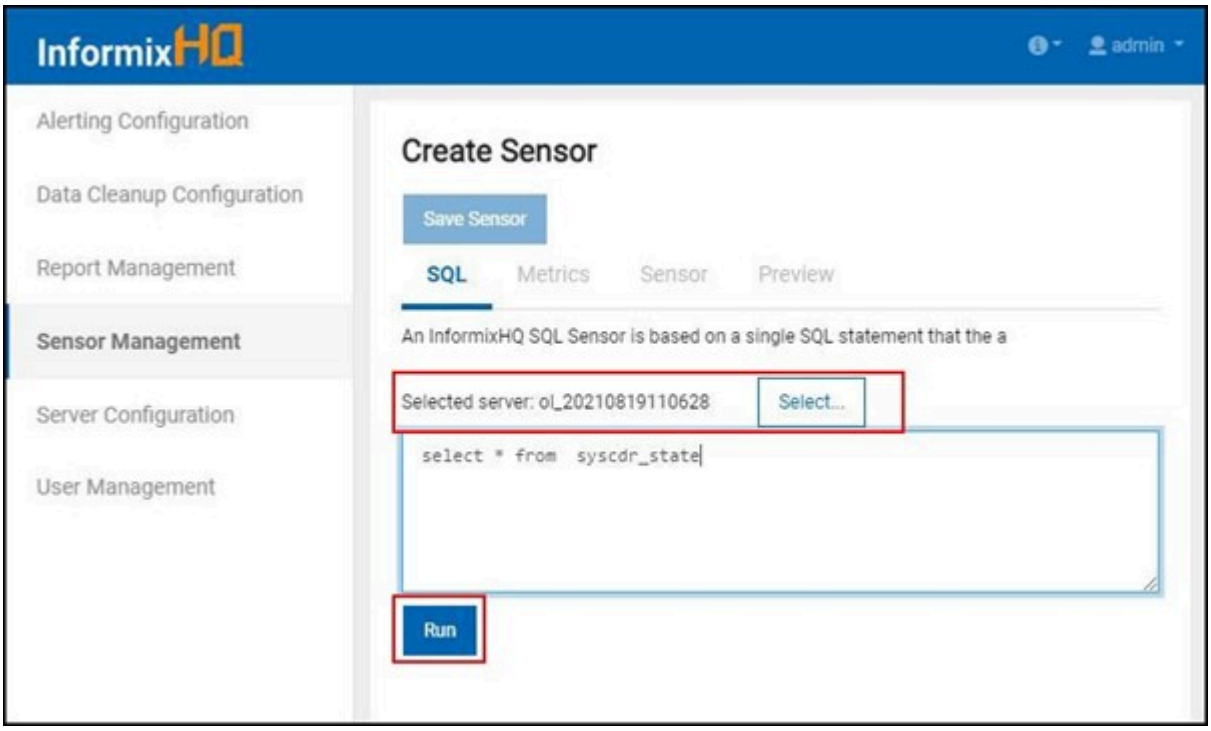
5. **Create Custom Sensor** button will navigate user to **Create Sensor** setup page.



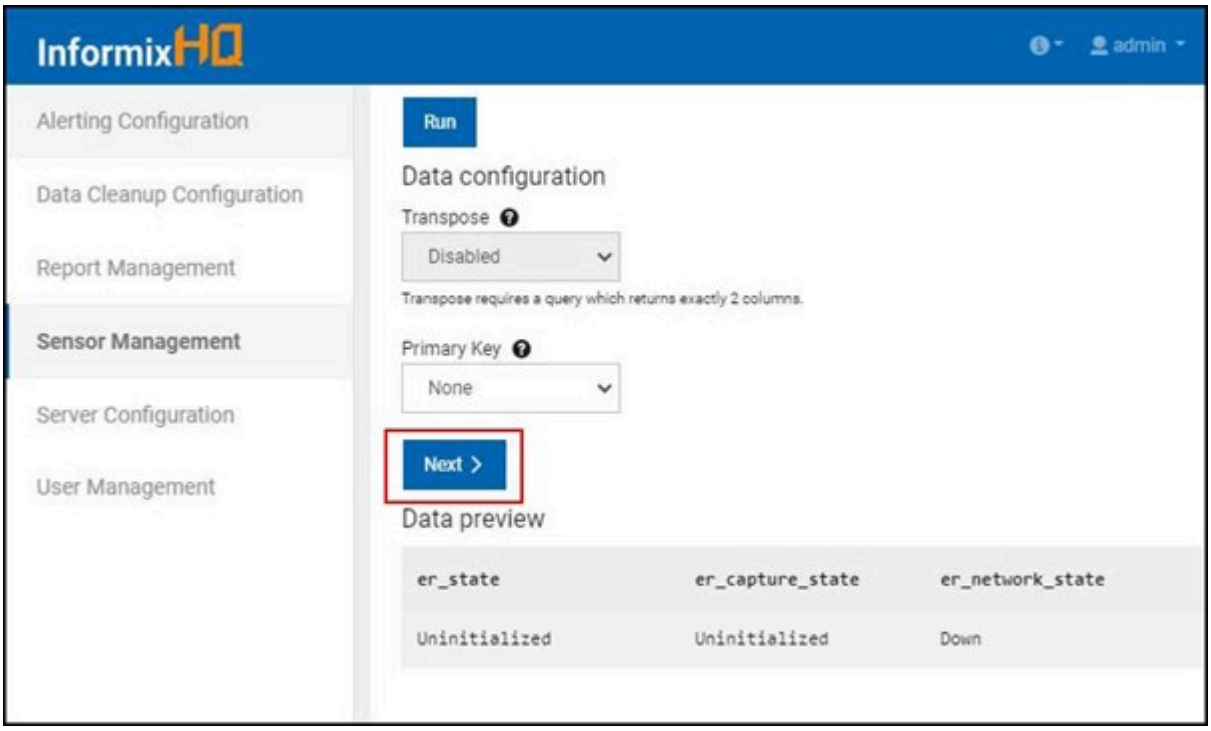
6. Click on the **Select** button to view the list of servers.



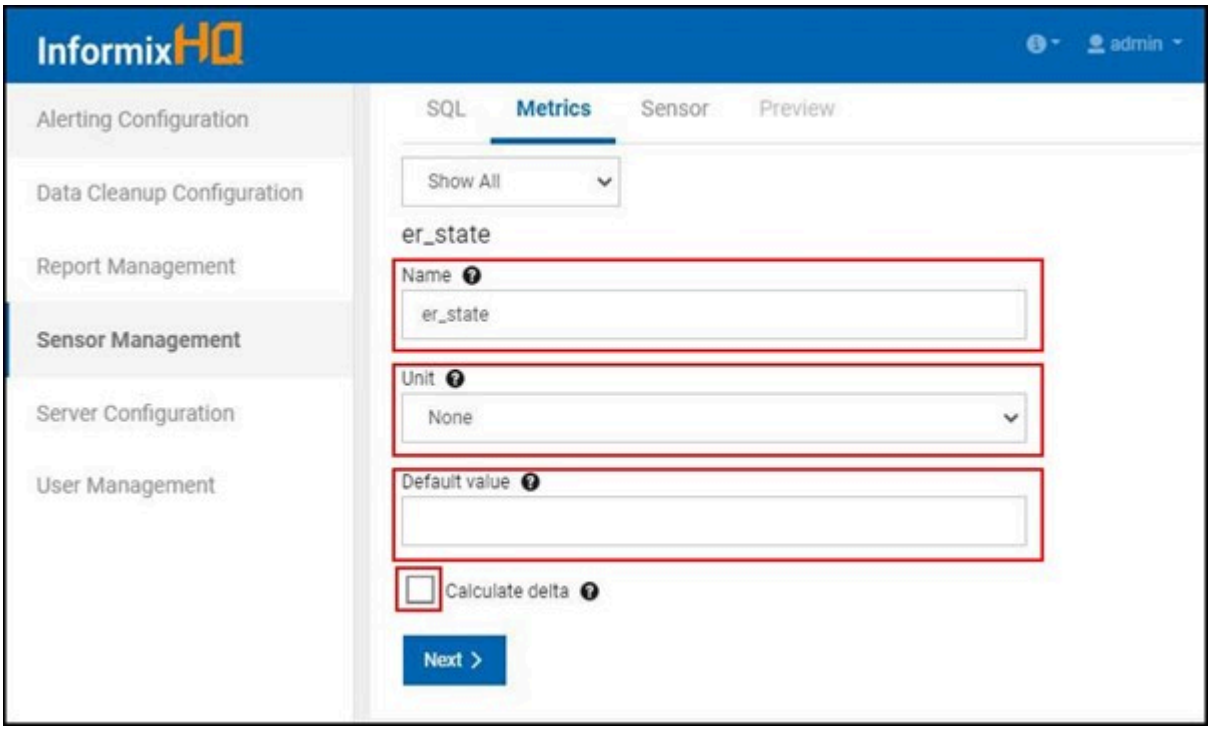
7. Select the desired server from the server list.
8. After selecting the desired server , write a query in text area & click on **Run** button to execute the query.



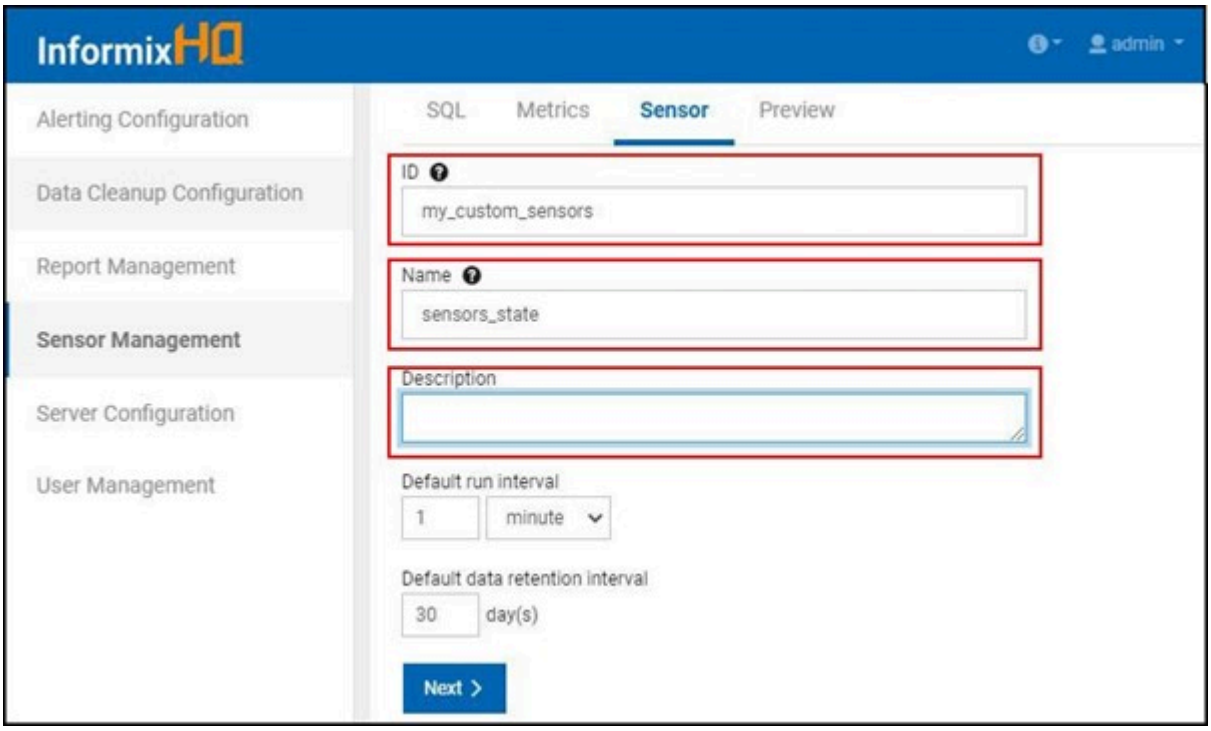
9. Clicking on **Run** button will show query data preview.
    - **Transpose :** Transpose (or pivot) will transform rows into columns based on the selected columns.
    - **Primary Key :** If a query returns multiple rows describing multiple objects (e.g.: dbspaces), the primary key will be used to uniquely identify each object. Null and duplicate values will be ignored.



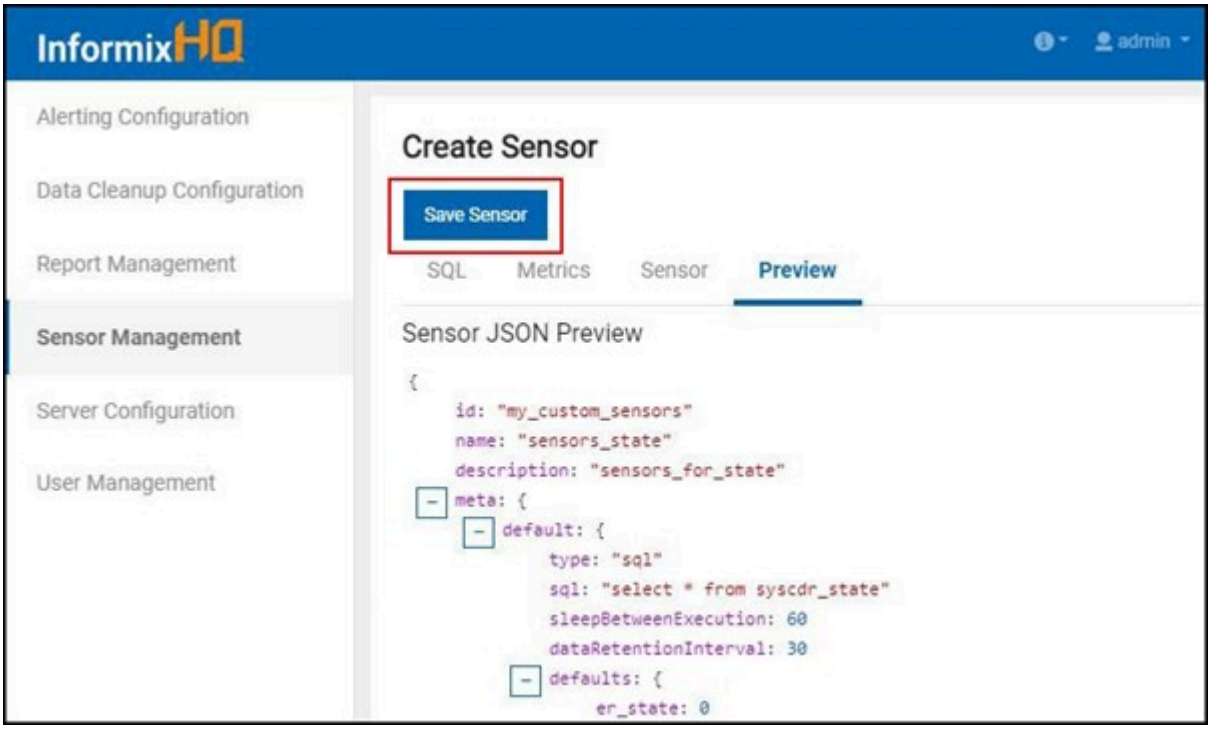
10. Click on **Next** button to navigate to **Metrics** tab. Metrics tab defines metrics for your columns. Change values as per the requirements.
    - **Name :** The display name of metric.
    - **Unit :** The unit determines how will this metric be displayed by default.
    - **Default value :** The default value of this metric if the query returns a null or invalid value. If blank, an invalid value will be ignored.
    - **Calculate delta :** If enabled, the sensor will calculate and store the difference per second between the latest reading and the previous reading.



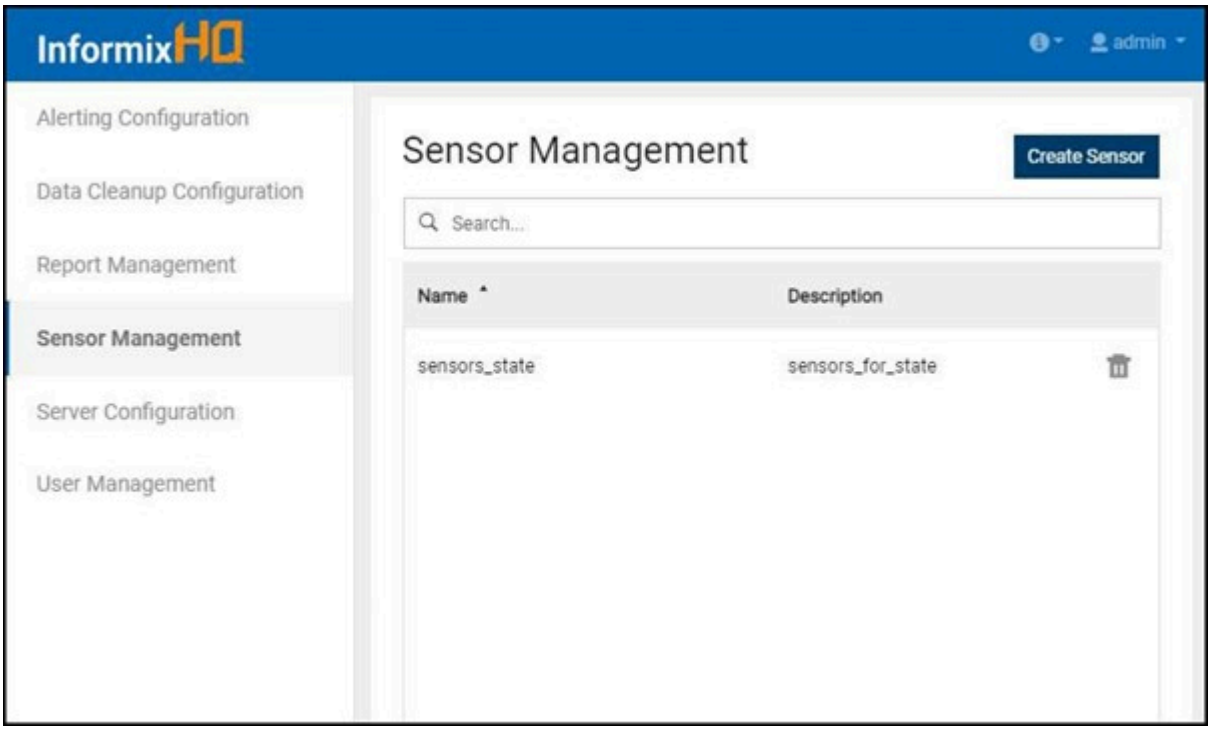
11. After making the changes click on **Next** button to navigate to the **Sensor** tab. **Sensor** tab provides all the related details for a sensor as per the requirements.
    - **ID :** The ID to uniquely identify this sensor. Can only use lowercase characters, digits, and single underscores (e.g.: my_custom_sensors).
    - **Name :** The display name of this sensor.
    - **Descriptions :** Add detailed description of the sensor.
12. User can change **Default run interval** and **Data retention interval** as per the requirements.



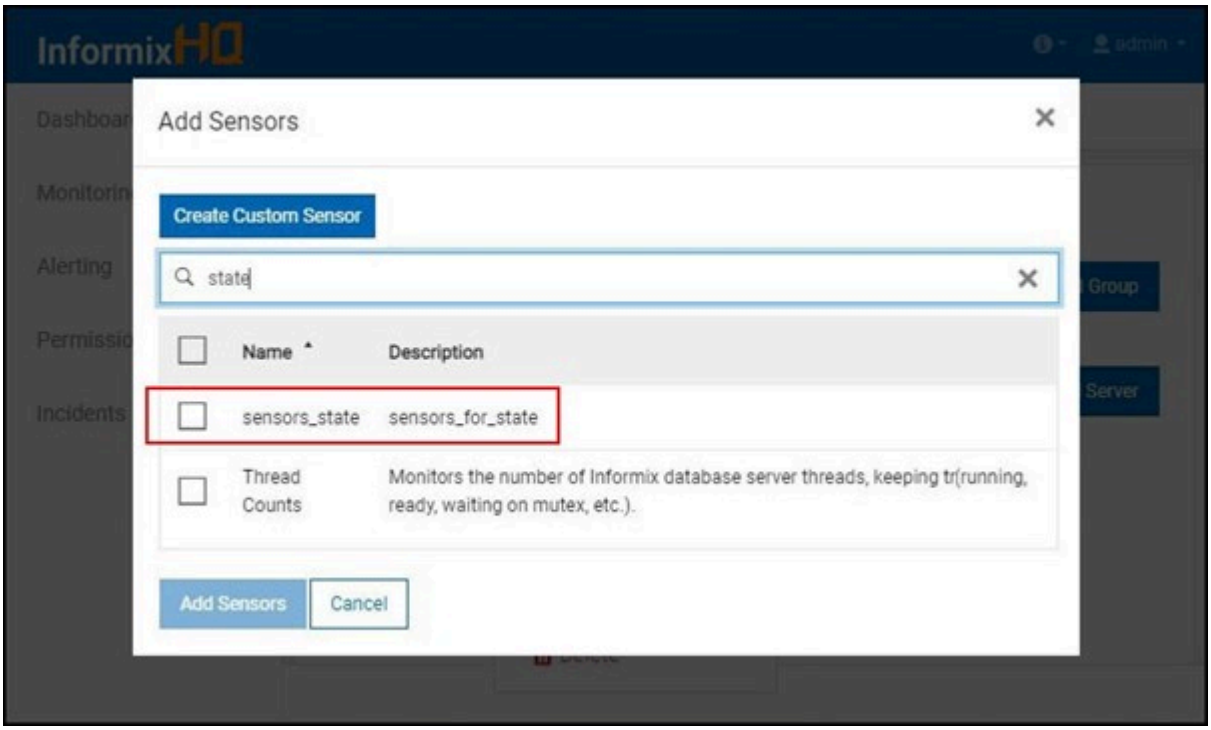
13. Click on **Next** button to navigate to **Preview** tab. Preview tab shows query preview for a custom sensor.



14. View your **Sensor JSON Preview**.
15. Click on **Save Sensor** button and **Confirm** to create a custom sensor.
16. After confirming, user can view list of all custom sensors created for a selected server on **Sensor management** page.



17. User can view recently added custom sensors in the sensor list from **Monitoring** page using **+ Add Sensors** model pop up.

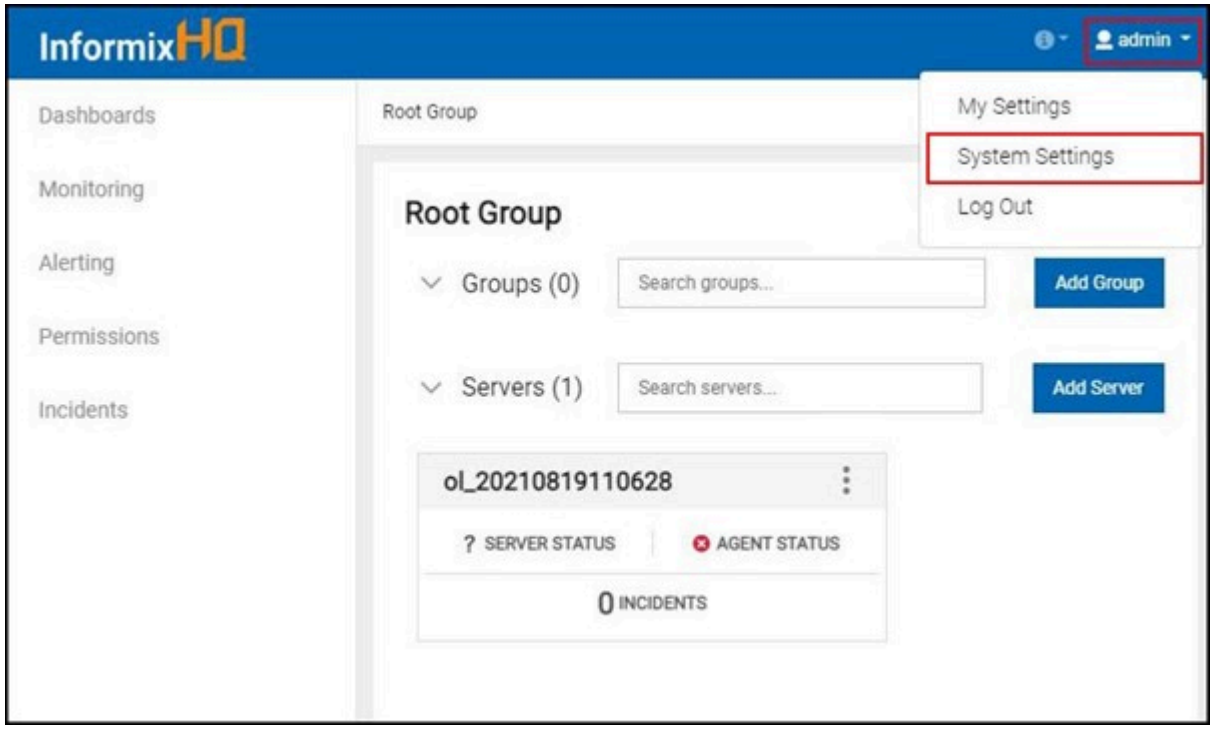


Creating a custom sensor from admin menu

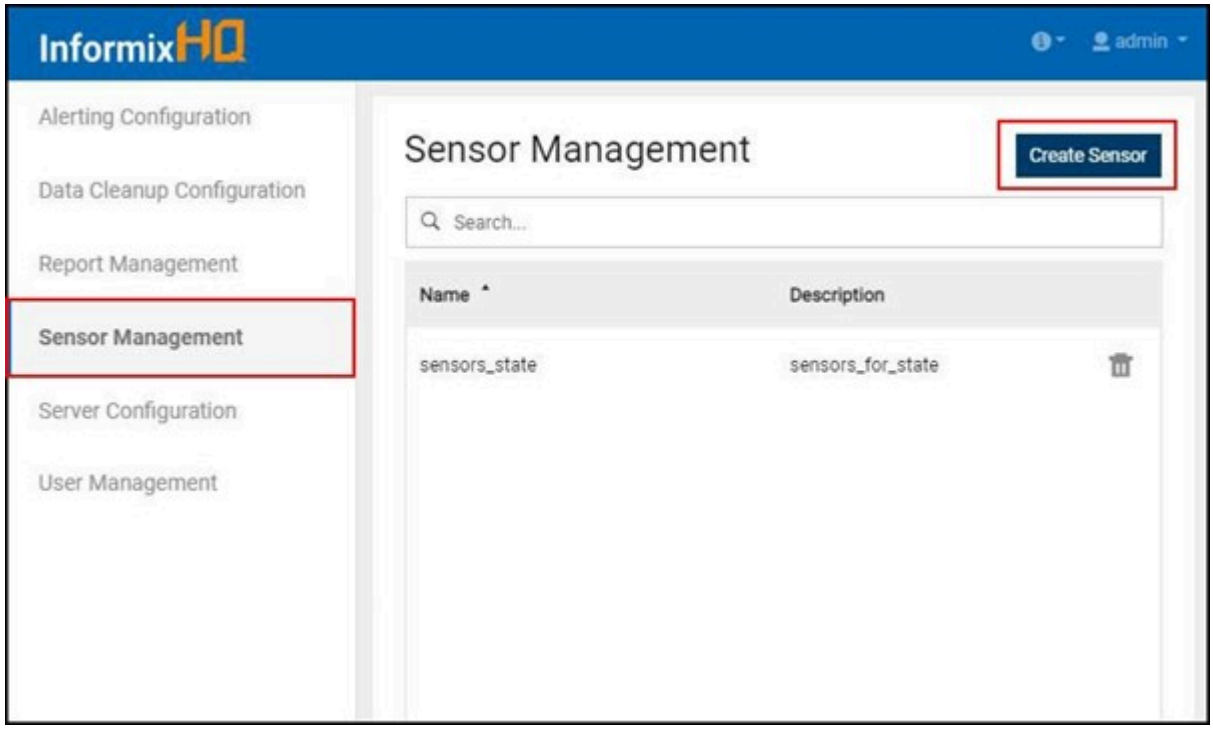
About this task

This topic explains how to create a custom sensor using InformixHQ **admin** menu.

1. Click on **admin** dropdown.
2. Click on **System Settings**.



3. Click on **Sensor Management.**
4. Click on **Create Sensor**.



Schema Manager

The **Schema Manager** page allows you to browse and view detailed information about the various tables and indexes in each of your databases.

Use the **Schema Manager** to:

- View detailed information about Databases
- View detailed information about Tables
- Create a Database
- Create a Demo Database
- Drop a Database
- Create a Table
- Drop a Table
- Create an Index
- Delete an Index

Viewing Database Information

You can view detailed information about database using different tabs like Stored Procedures, Sequences, User Defined Types, Data Blades etc.

To view database information:

1. Go to the **Schema Manager** page in InformixHQ.
2. Select any table from the list.
3. Click on each tab to view the information like Stored Procedures, Sequences, User Defined Types, Data Blades.

Viewing Table Information

You can view detailed information about tables using different tabs like Indexes, References, Constraints, Triggers etc

To view information about table:

1. Go to the **Schema Manager** page in InformixHQ.
2. Select any table from the list.
3. Click on each tab to view the information like Indexes, References, Constraints, Triggers.

Creating a Database

To create a database:

1. Go to the **Schema Manager** page in InformixHQ.
2. Select any database from the list.
3. Click **…** and – Select **Create Database**

4. Enter the required details.
5. Click the **Finish** button. The new database will be created.

Creating a Demo Database

To create a demo database:

1. Go to the **Schema Manager** page in InformixHQ.
2. Select any database from the list.
3. Click **…** and – Select **Create Demo Database**
4. Enter the required details.
5. Click the **Create** button. The demo database will be created.

Dropping a Database

To drop a database:

1. Go to the **Schema Manager** page in InformixHQ.
2. Select the database you want to drop.
3. Click **…** and – Select **Drop Database**
4. Click **Yes** to confirm. The database will be dropped.

Creating a Table

About this task

This topic explains the steps to create a table using InformixHQ UI.

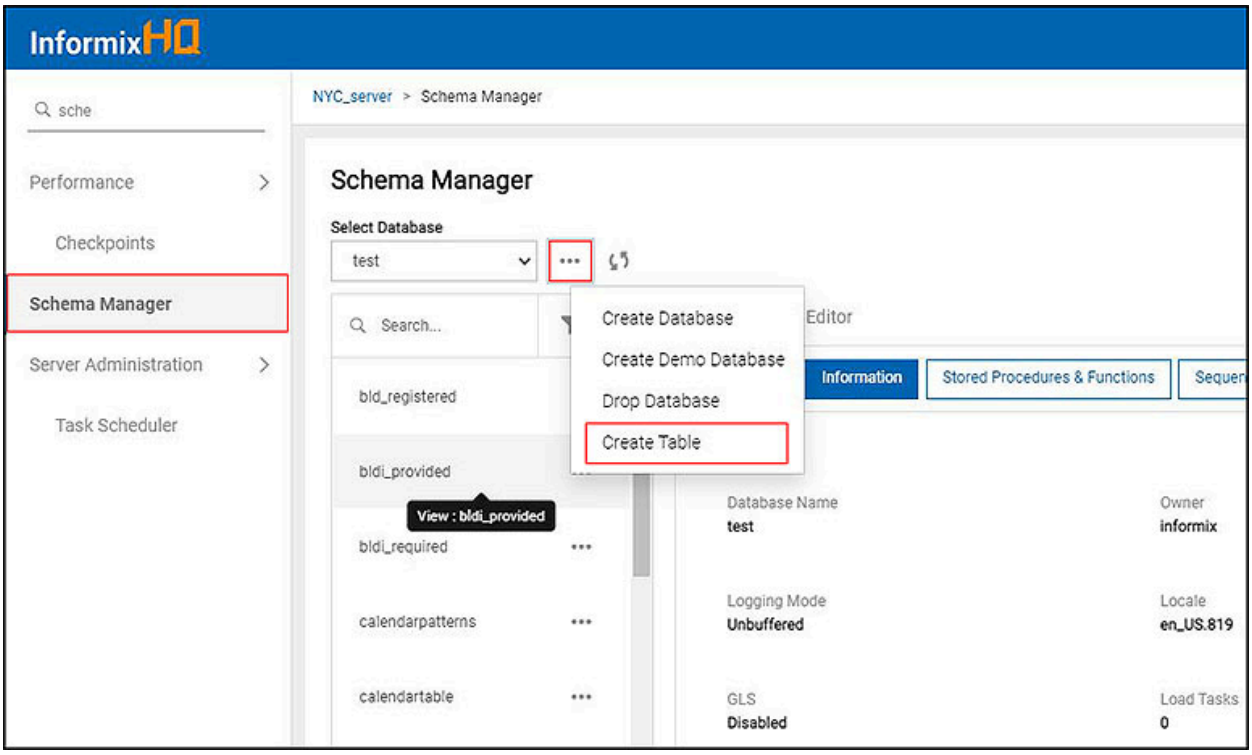
There are five types of tables:

1. Standard table
2. Raw table
3. External Fixed table
4. External Delimited table
5. External Informix table

Standard & Raw table types are almost similar & three external table types are almost similar. This topic explains how to create both table types.

To create a table, follow the steps given below:

1. Go to the **Schema Manager** in InformixHQ.
2. Select desired database and click on menu option (3 dots) next to **Select Database** dropdown.
3. Click on **Create Table** option from dropdown to create a table



To create a Standard or Raw type table:

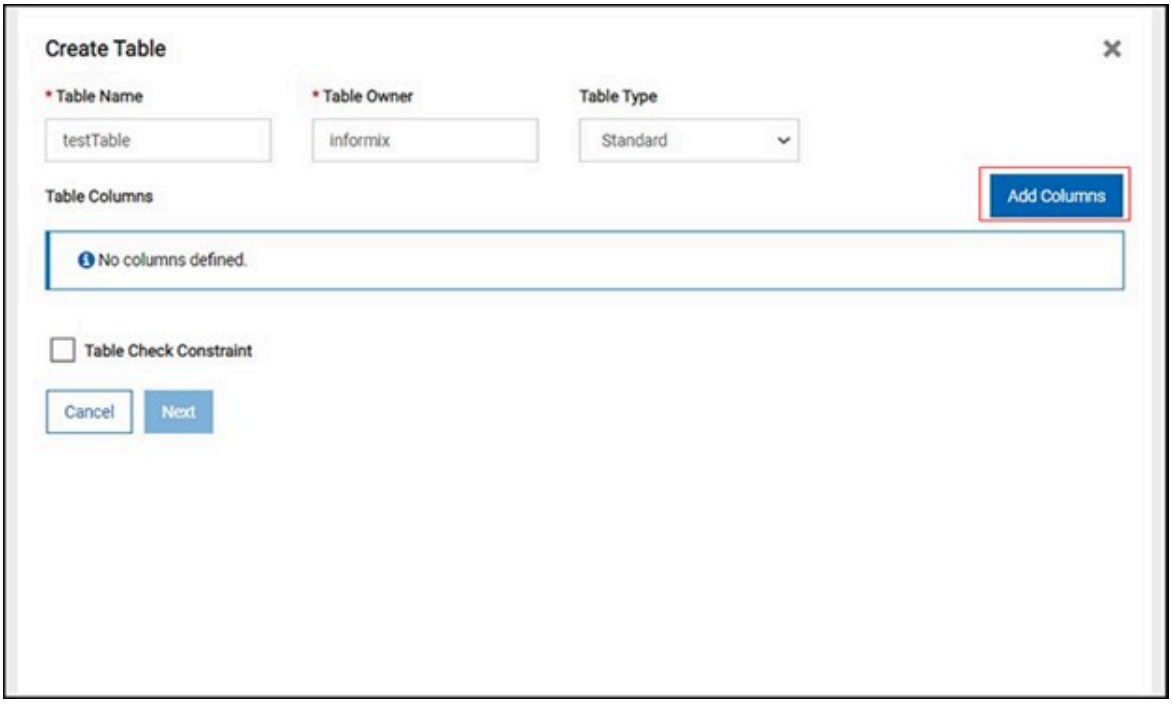
a. Enter mandatory fields such as **Table Name** and **Table Owner**.

b. Select Standard(default) or Raw type from **Table Type** dropdown.

c. Click on **Add Column** for adding columns to the table.

d. To cancel **Create Table** operation click on **Cancel** button.

e. **Next** button will be enabled after user is done adding columns to the table.



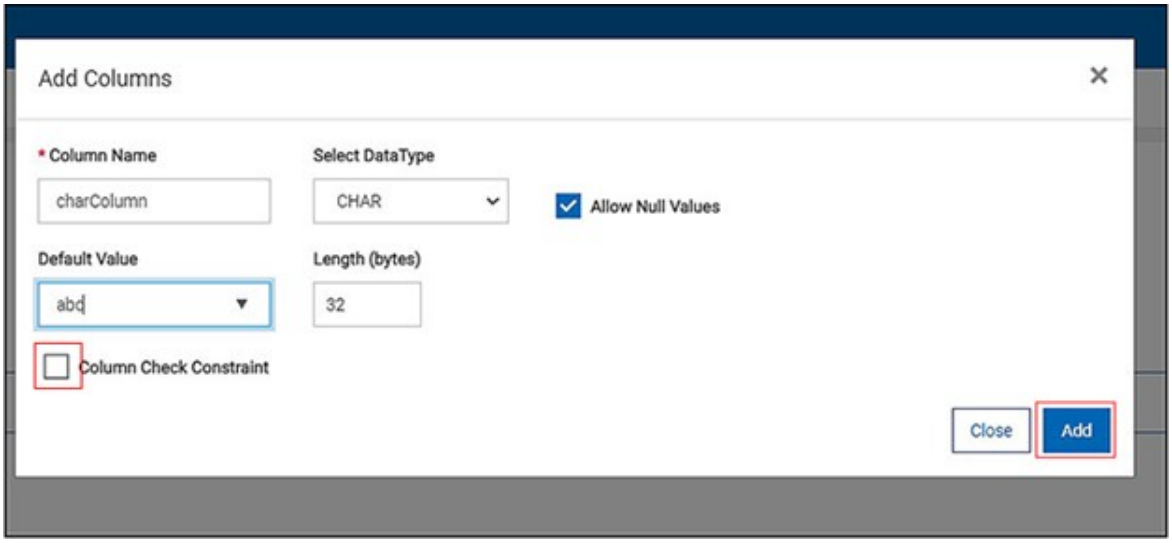
a. By clicking on **Add Columns** button pop up will appear to add column details.

b. Enter values for all mandatory fields(*).

c. To give a constraint on any column, click on **Column Check Constraint** checkbox.

d. Multiple columns can be added using the same pop up.

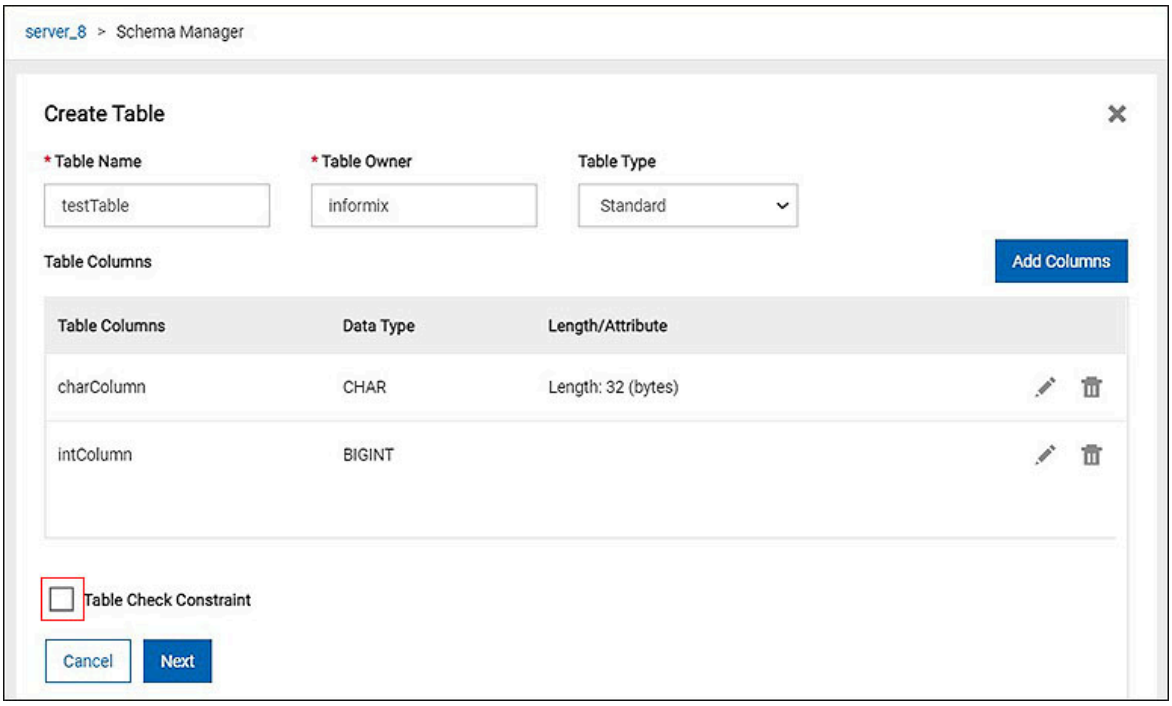
e. To go back to the main screen, click on **Close** button or **cross** icon in the upper right corner.



a. Once columns are added, user can view, edit, delete any of the columns.

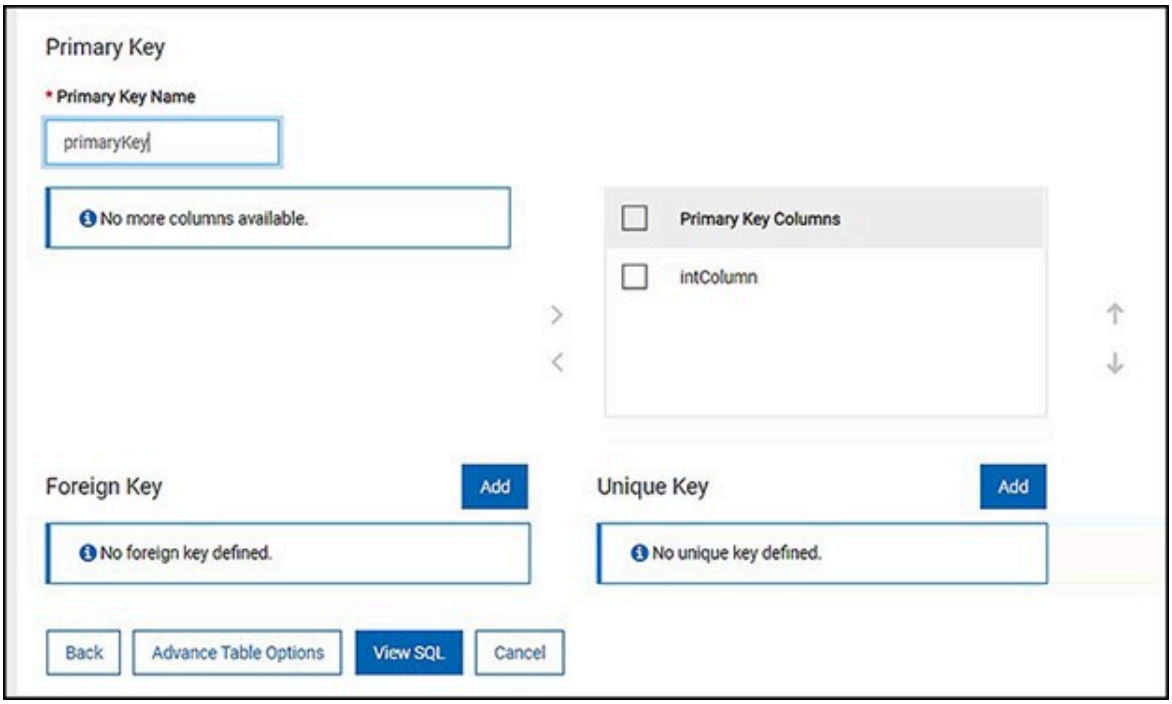
b. Table level constraint can be added on this screen by clicking on **Table Check Constraint** checkbox.

c. Once details related to the columns are finalized, clicking on **Next** button takes user to add constraint page.



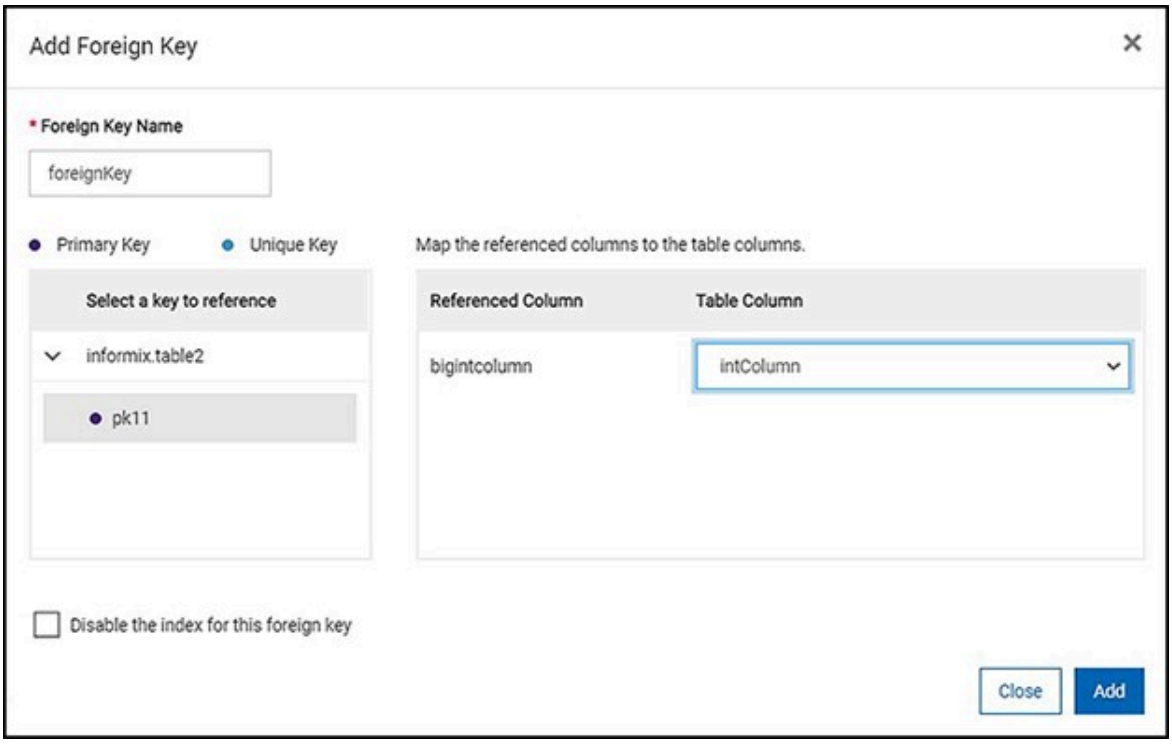
a. This screen is for adding a constraint like primary key, foreign key, unique key to a table.

b. To assign primary key to a table, give **Primary Key Name** & select **Primary Key Columns**.



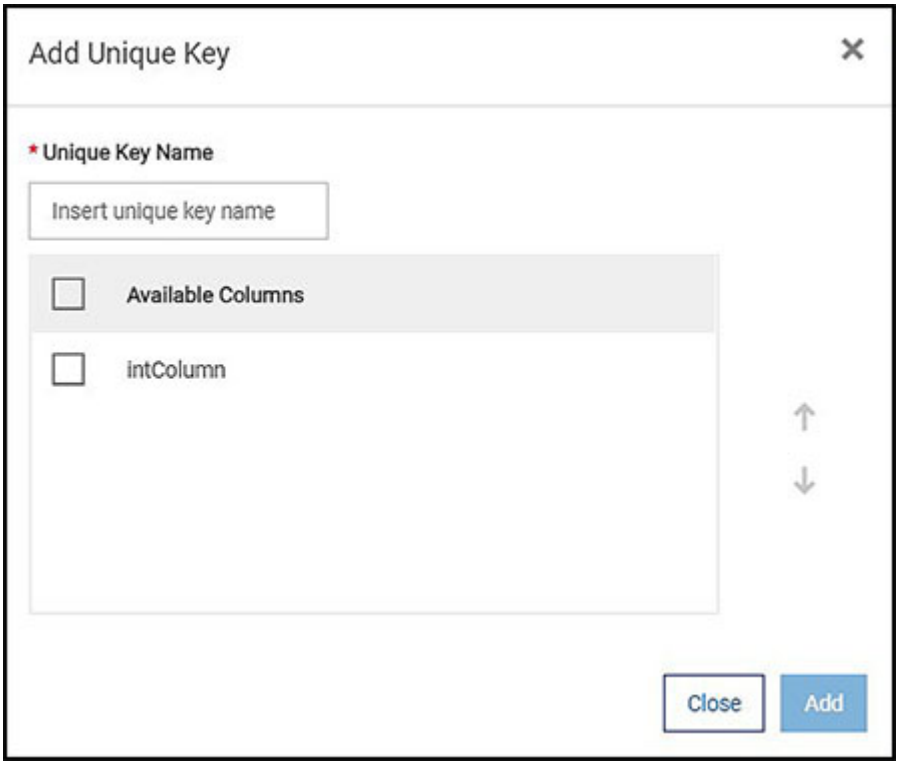
a. To add Foreign key constraint, click on **Add** button in Foreign key section.

b. To assign a foreign key for a table, give **Foreign Key Name** & map **Referenced Column** with **Table Column**.

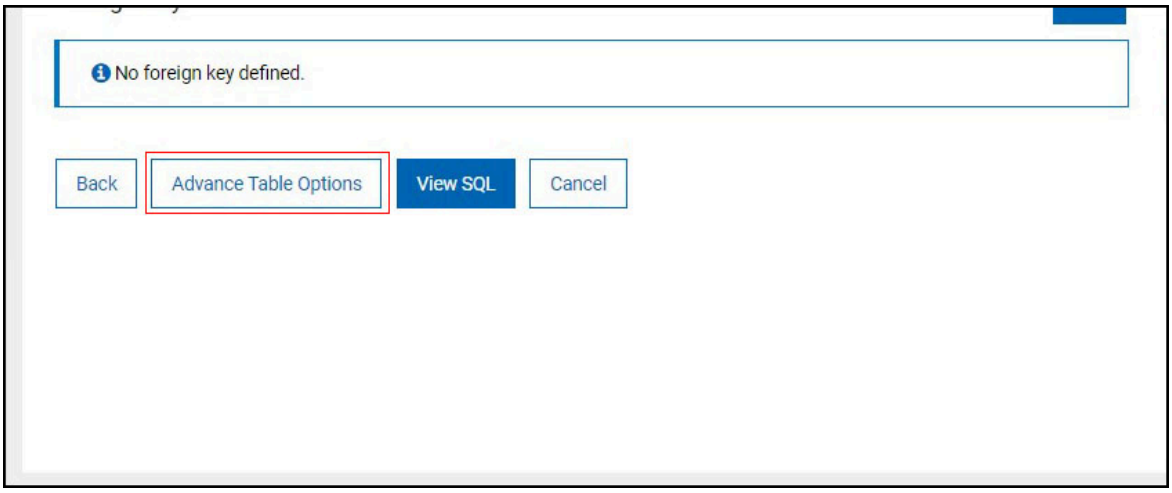


a. To add a Unique key constraint, click on **Add** button in **Unique key** section.

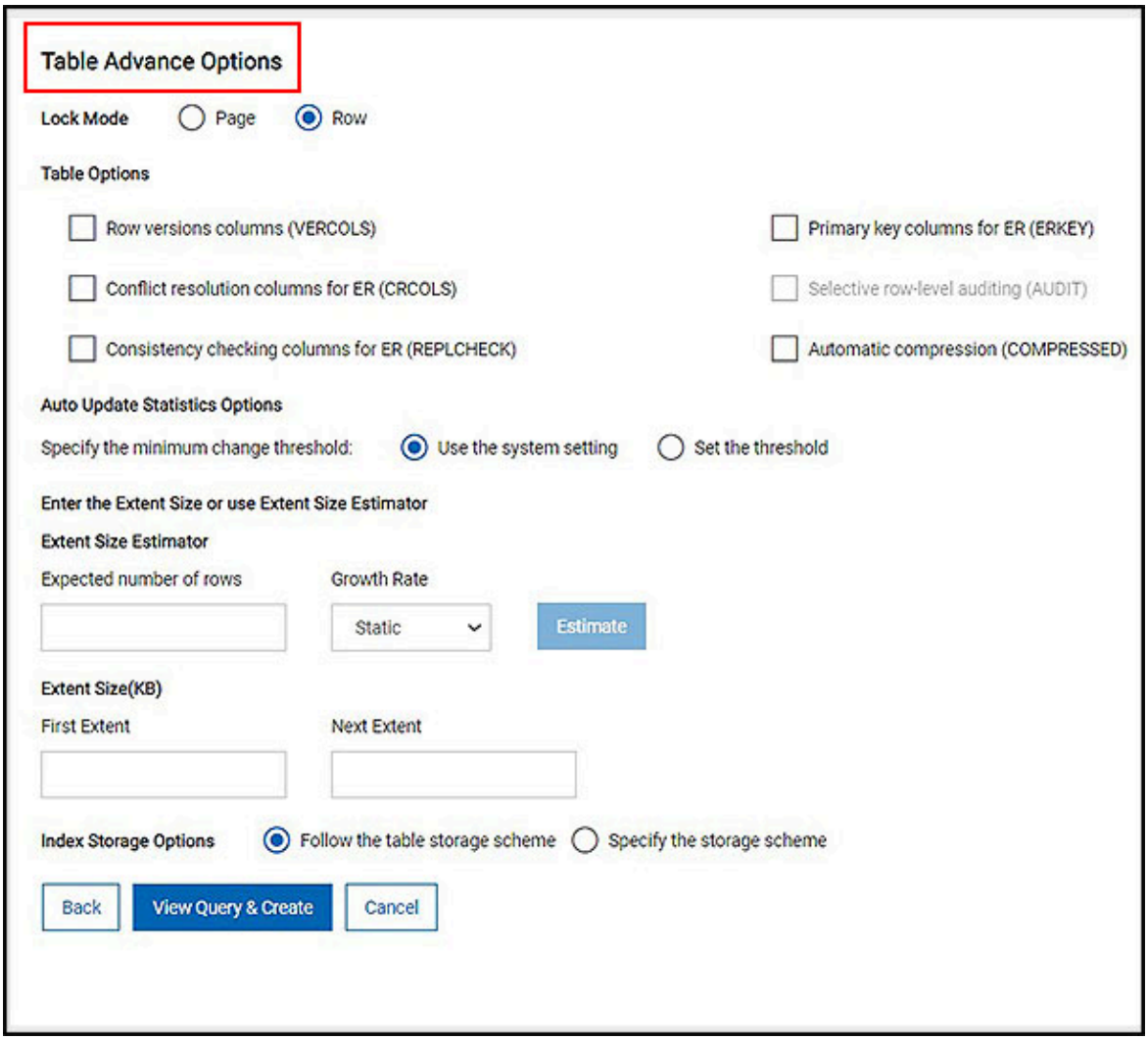
b. To assign a unique key for a table, give **Unique Key Name** & select **Available Columns** from the list. This will enable the **Add** button.



a. Once constraints are added user can go to either **View SQL**or **Advance Table Options**



a. Modify advance table level options using the screen given below. For example, changing lock mode, storage scheme, update statistics, etc.



a. Once advance table option are set, click on **View Query & Create** button to view SQL query for creating the table.

b. After clicking on **View Query & Create** button from Advance Table Options or on **View SQL**button from **Add Constraints** screen, user will be able to view 'create table' query as shown in the screen given below.

c. Click **Create** button to create the table.

d. If table is created successfully, information status message will be shown and user will be taken back to **Schema Manager** page.

e. If table creation fails, error status message is displayed and all the create table related queries will be rolled back

f. To go back to modify any properties click **Back** button and to cancel the operation of create table click on **Cancel** button.

External table type:

To create an External type table:

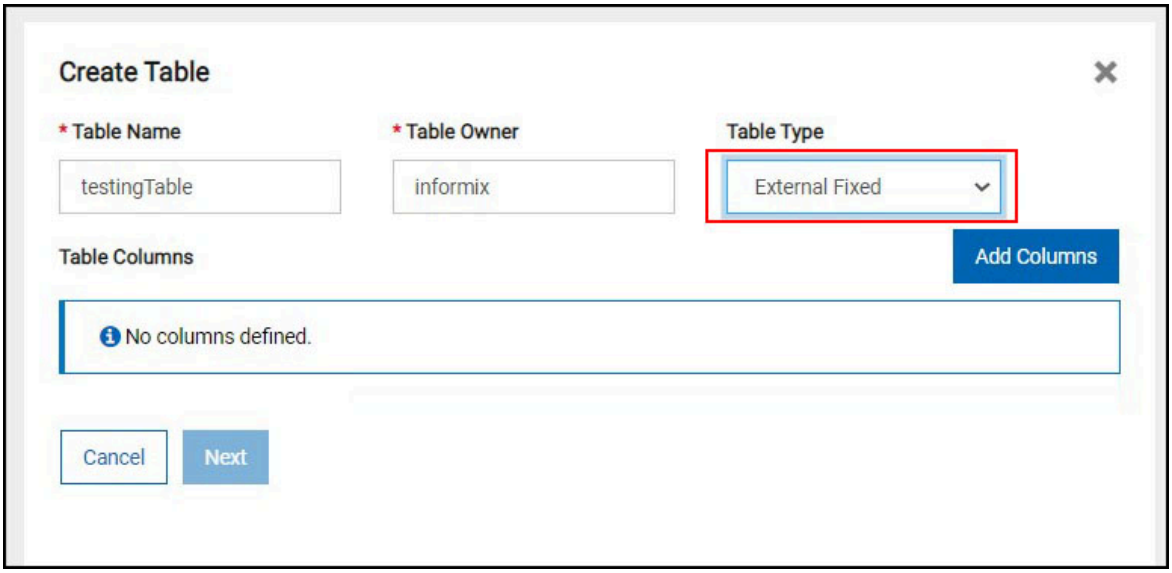
a. Enter mandatory fields such as **Table Name** and **Table Owner**.

b. Select one of **External Fixed**, **External delimited**, **External Informix** table type from **Table Type** dropdown.

c. Click on **Add Columns** for adding columns to the table.

d. To cancel **Create Table** operation, click on **Cancel** button.

e. **Next** button will be enabled once user adds columns to the table.

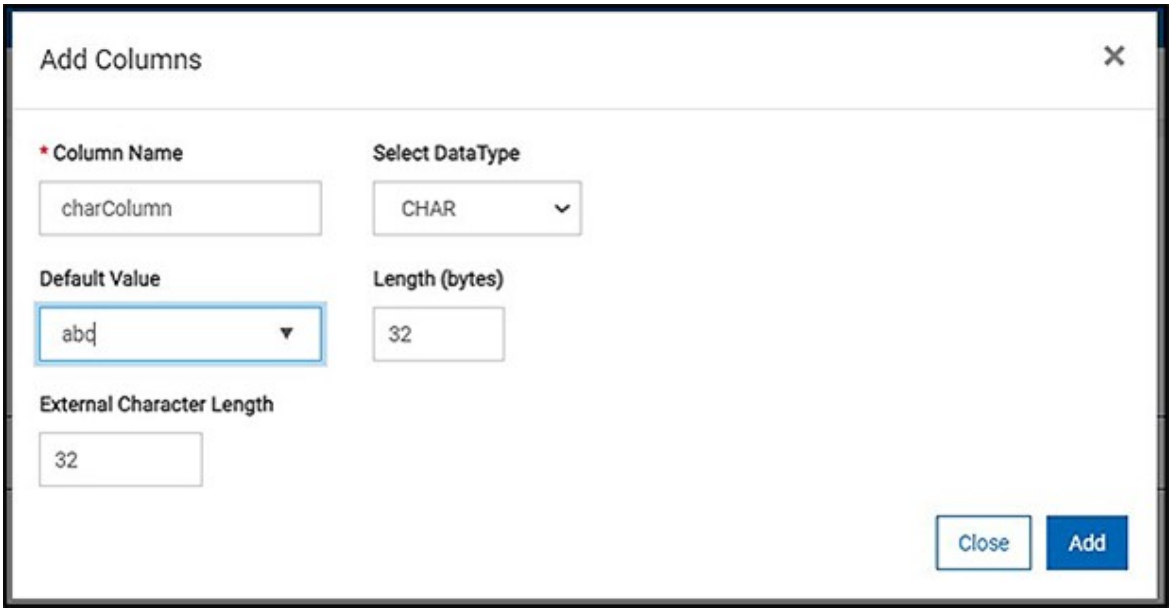


a. By clicking on **Add Column** button, a pop up will appear to add column details.

b. Enter all the mandatory field values(*).

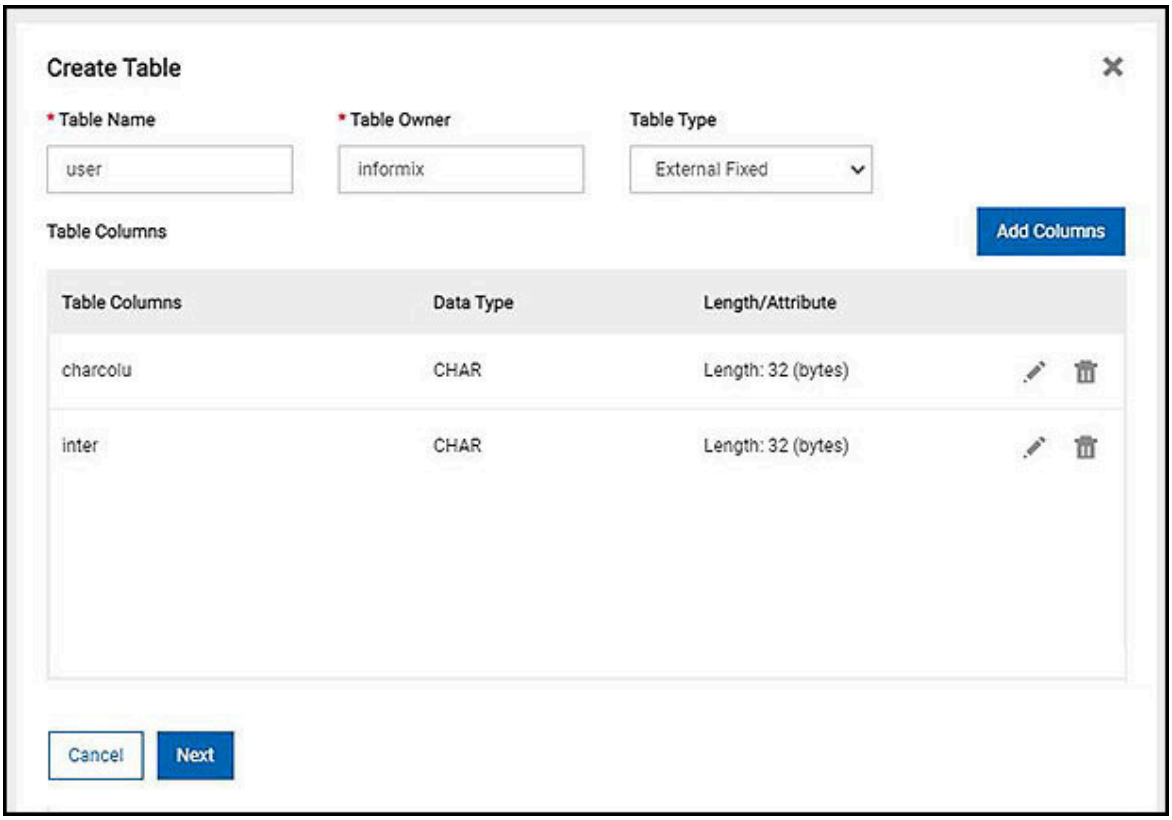
c. User can add multiple columns using the same pop up.

d. To go back to the main screen, click on **Close** button or **Cross** icon.

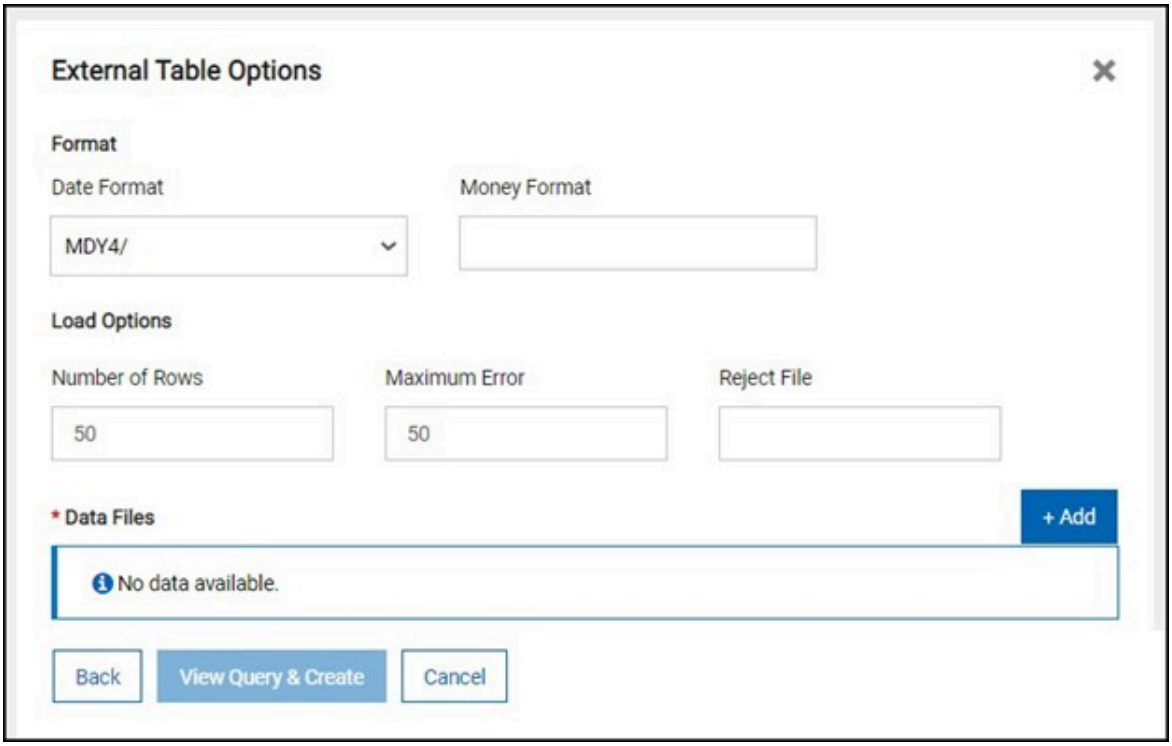


a. Once columns are added, user can view, edit , delete any of the columns.

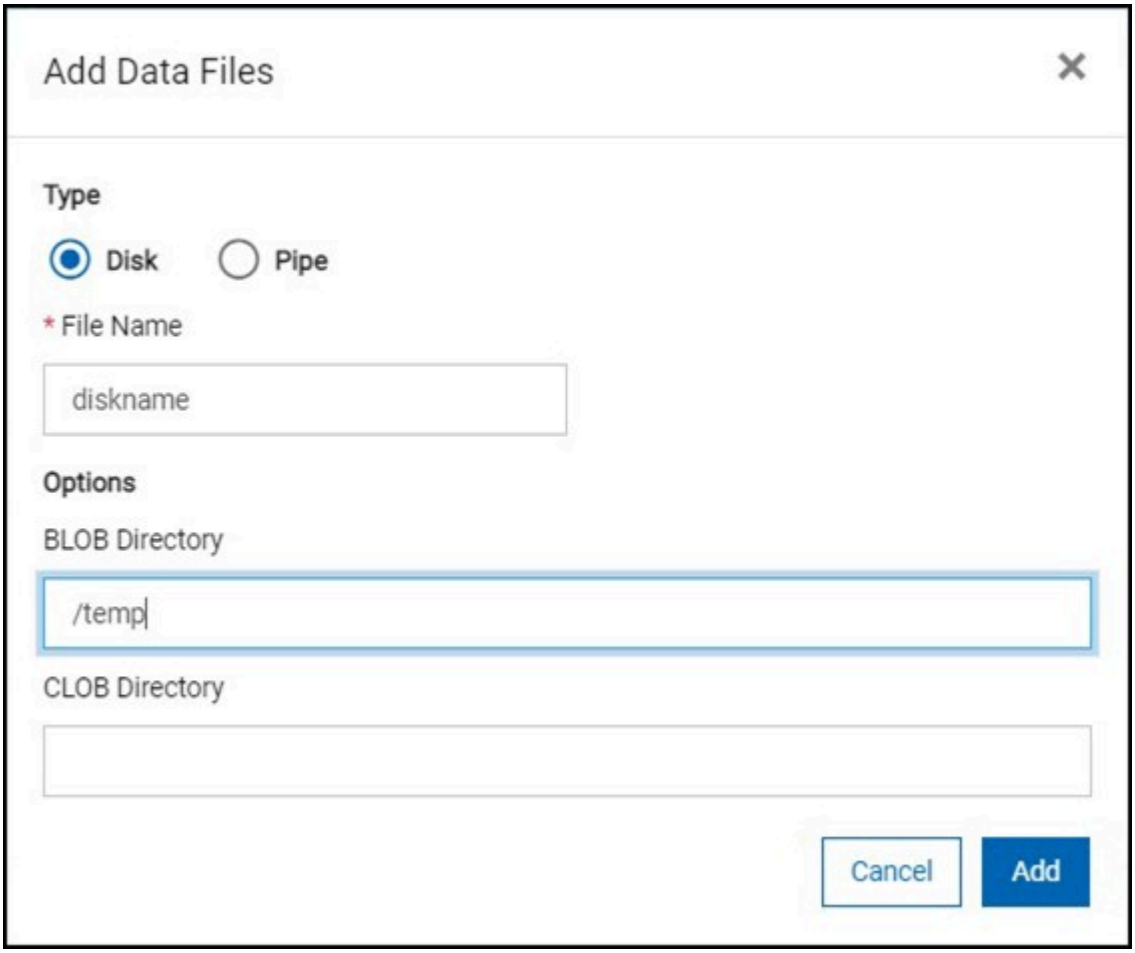
b. Once column details are finalized click on **Next** button to go to **External Table Options**



a. Provide information for external table options & add mandatory data file by clicking on **+Add** button.



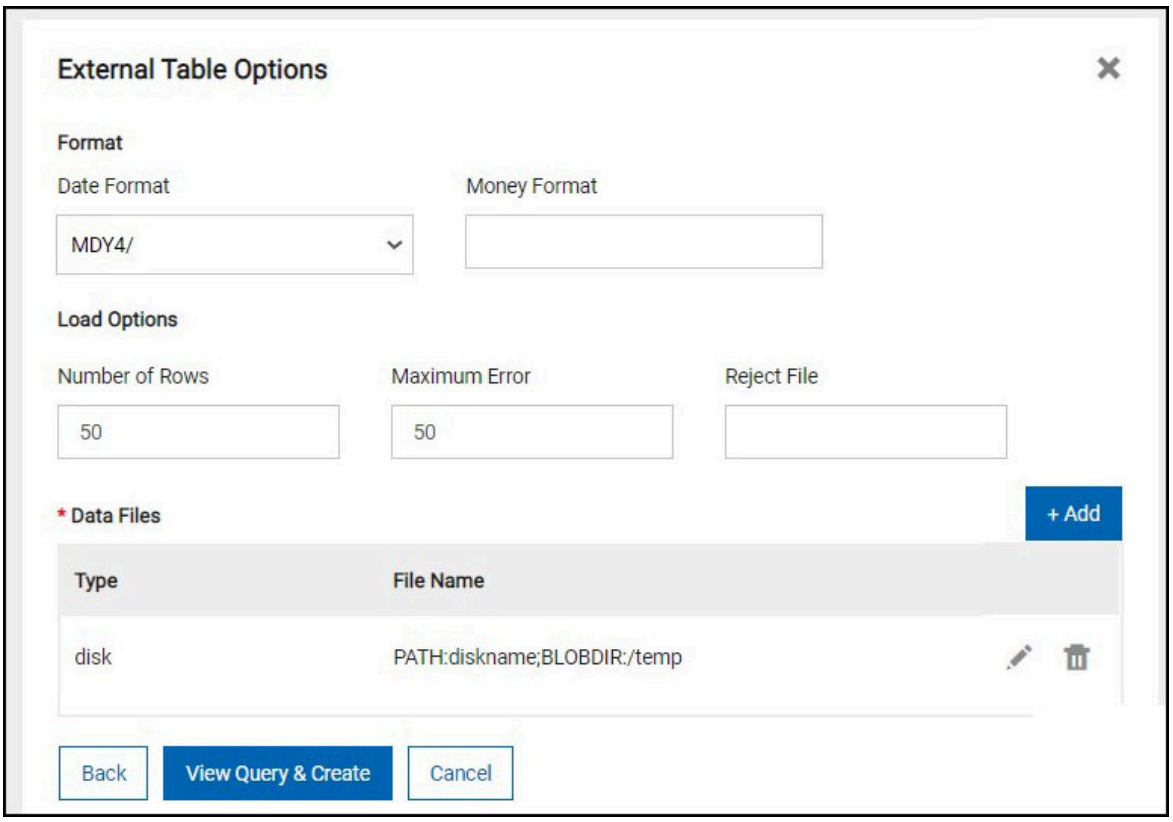
a. Following pop up is used to add data files to an external table.



a. Once data files are added, user can view, edit , delete any of the data files.

b. Once External table options are finalized click on **View Query & Create** button to view SQL query for creating the table.

c. User can either go **Back** or **Cancel** the operation using respective buttons.



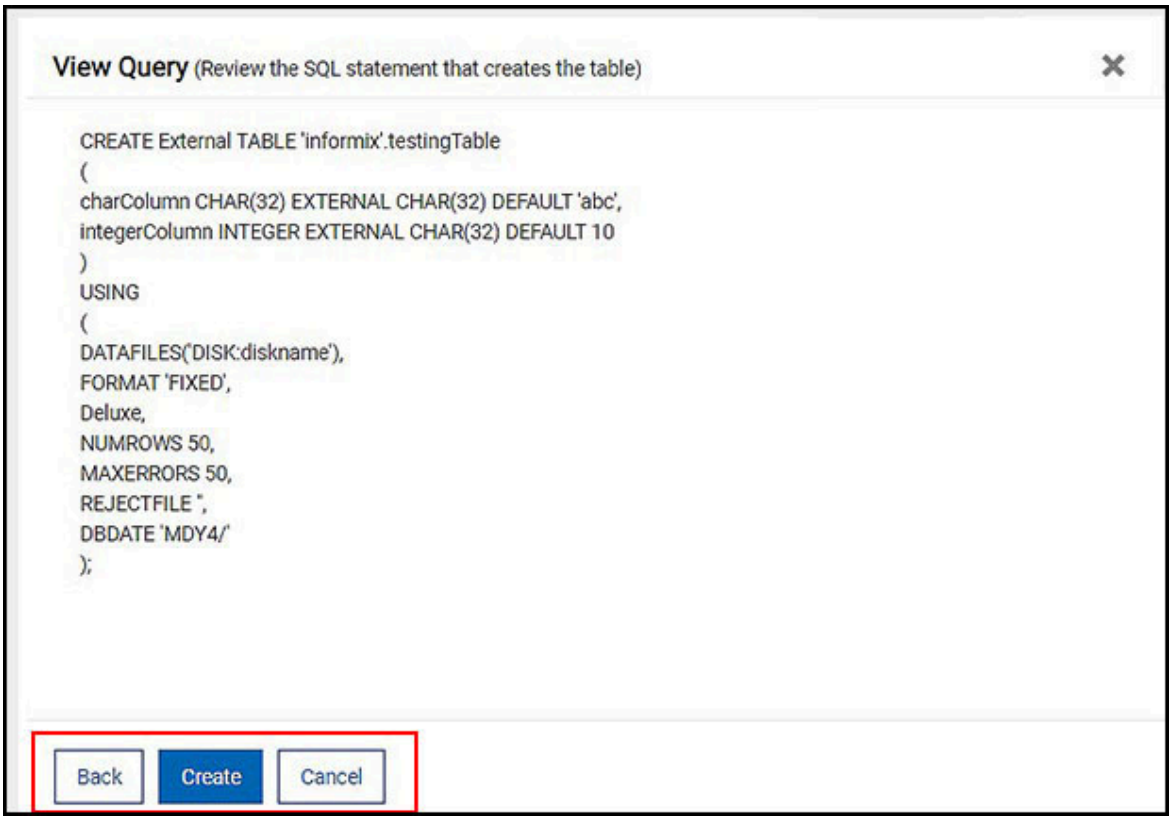
a. After clicking on **View Query & Create** button from the external table option, user will be able to view create table query as shown in the screen below.

b. Click **Create** button to create the table.

c. If table is created successfully information status message will be shown and user will be taken back to **Schema Manager** page.

d. If table creation fails, error status message will be displayed and all the create table queries will be rolled back.

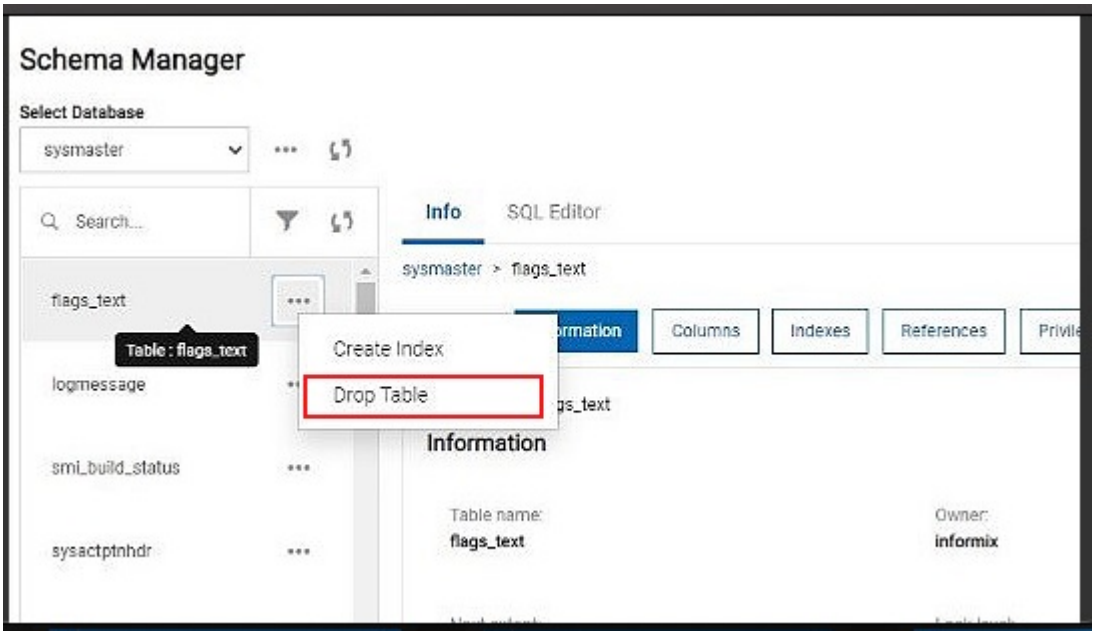
e. To go back to modify any properties, click **Back** button and to cancel the operation of create table click on **Cancel** button.



Dropping a Table

1. Click on **Schema Manager** in InformixHQ.
2. Select a desired database from **Select Database** dropdown which contains the table to be dropped.
3. From the table list shown, locate the table to be dropped.
4. Click on menu option (3 dots) next to this table and select **Drop Table** from the dropdown menu to drop a table.

5. Confirm action on pop over (confirmation pop up for dropping a table).



Creating an Index

To create an index:

1. Go to the **Schema Manager** page in InformixHQ.
2. Select the table.
3. Click **…** and – Select **Create Index**
4. Enter the required details.
5. Click the **View SQL** button to review the SQL statement.
6. Click the **Create** button, index will be created.

Note: You can also **enable** or **disable** the index.

Deleting an Index

To delete an index:

1. Go to the **Schema Manager** page in InformixHQ.
2. Select the table for which you want to delete the index.
3. Click the **Indexes** tab.
4. Select the index you want to delete.
5. Click the **Delete** icon.
6. Click **Yes** to confirm. Index will be deleted.

SQL Editor

Why use the SQL Editor?

The SQL editor allows user to run Structured Query Language (SQL) statements in a InformixHQ for the Informix database server. User may use SQL editor to retrieve, create, change, or delete data in a database, if user has database permissions to perform these actions. In short, it creates a way for managing and manipulating data in the database.

Note: SQL Editor supports single query execution only. Execution of multiple queries and other editor features will be considered for forthcoming releases.

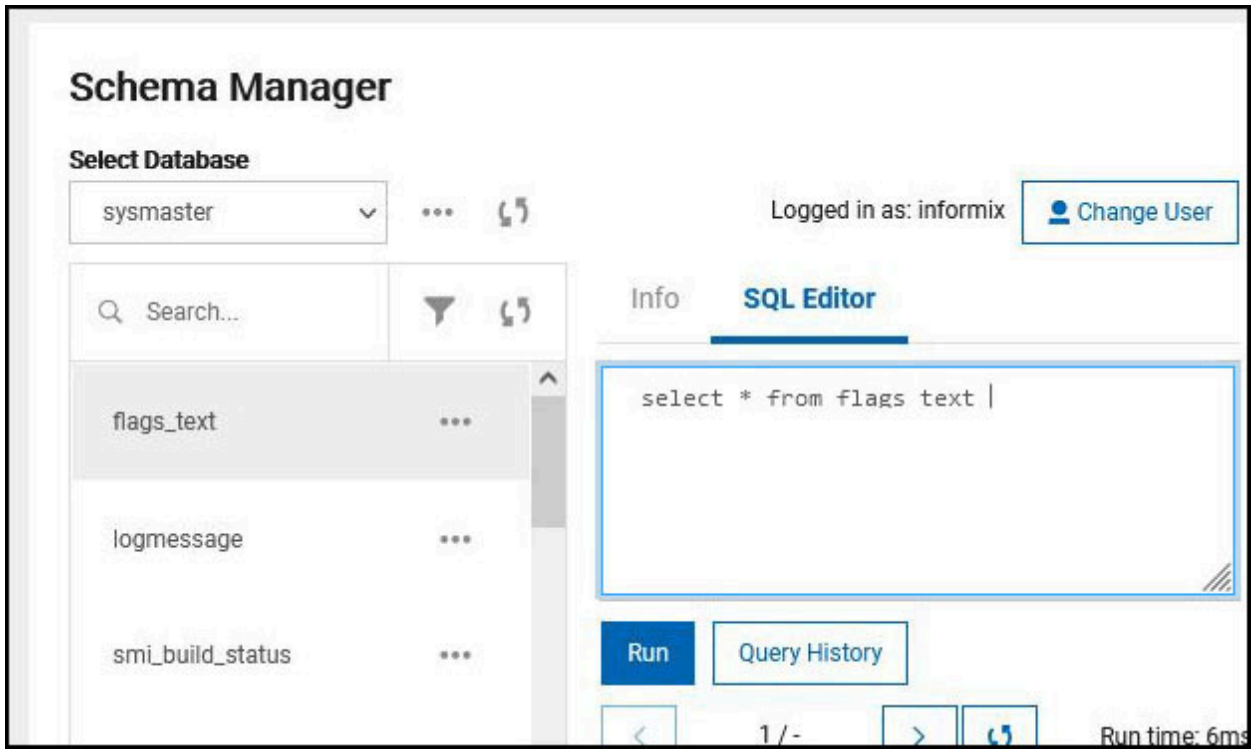
How to use the SQL Editor?

Follow the steps given below to navigate to the SQL editor:

- Select server in informixHQ on which you have to run SQL query.
- Select **Schema Manager** from side menu.

- Once user clicks on **Schema Manager** menu, **Schema Manager** page shows below tabs:

1. **Info :** Tab will show all the details (Properties) for selected database & table.
2. **SQL Editor :** Tab allows user to run SQL query for selected database & table.

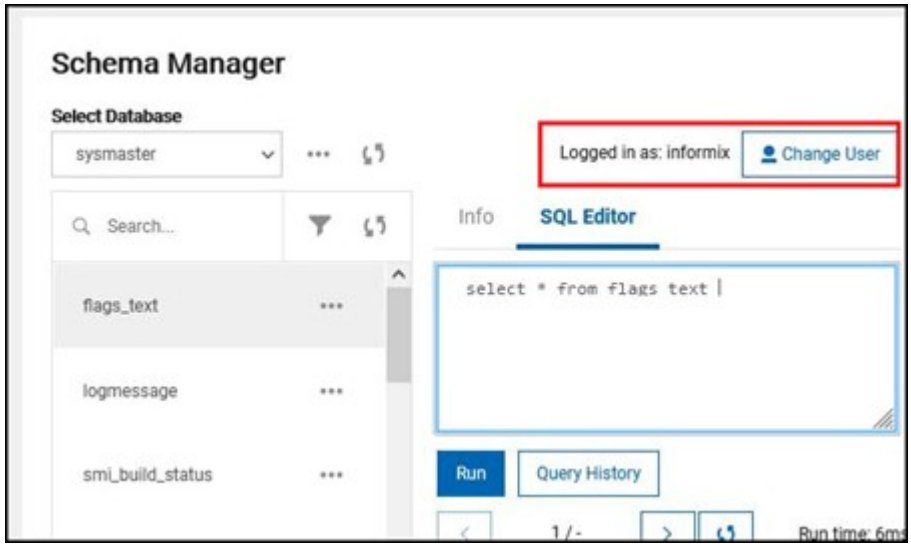


Changing logged in user

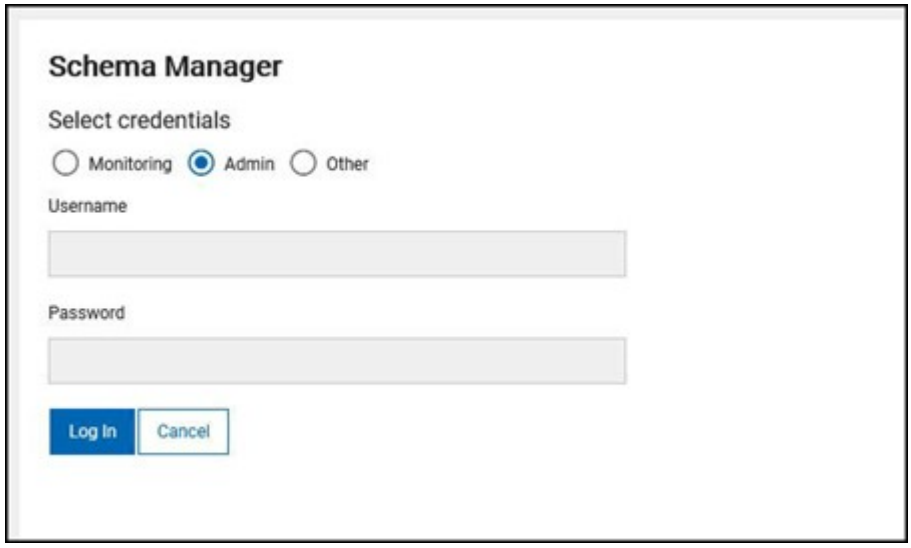
About this task

This topic explains steps for changing logged in as user.

1. User can change 'logged in as' by clicking on **Change User** button.



2. Click on **Change User** button to navigate to the **Schema Manager** screen where user can select credentials.

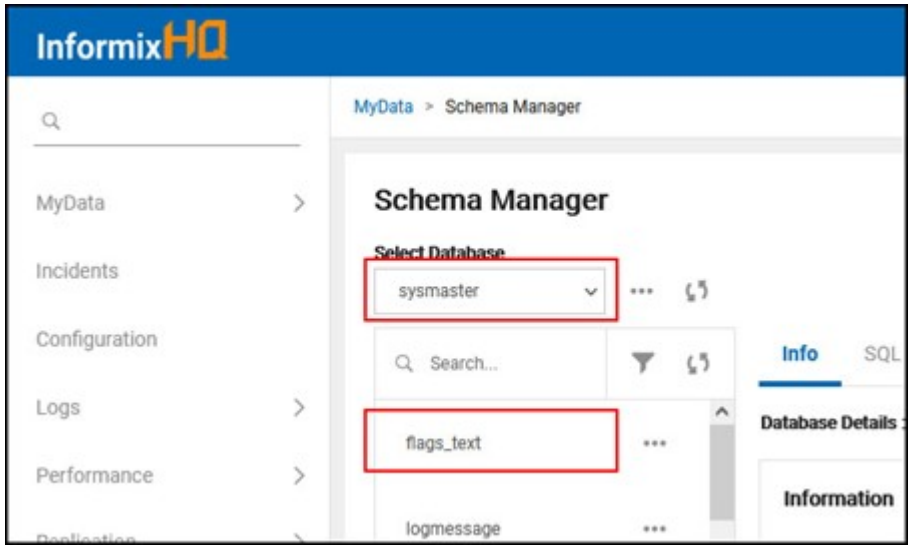


Using SQL Editor to view table records

About this task

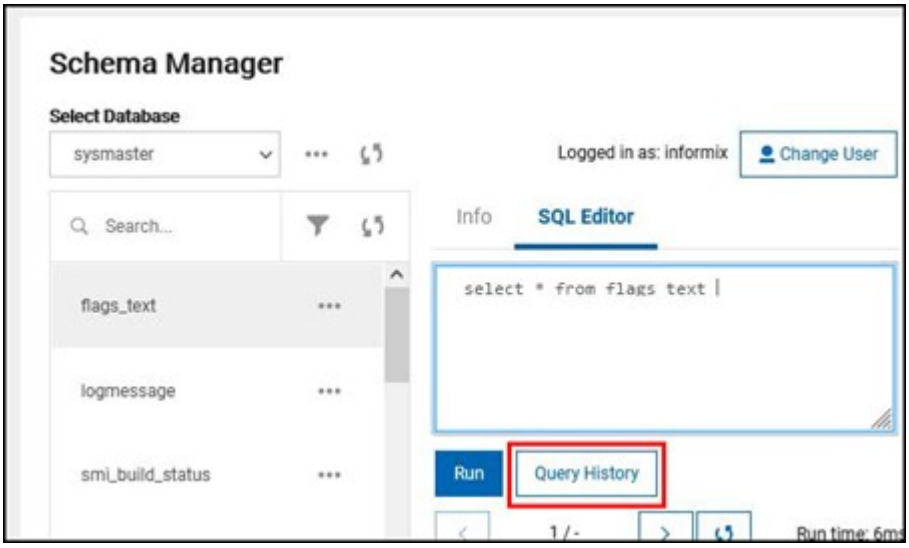
This topic explains how to use SQL Editor for viewing table records and SQL execution plan.

1. Select the required database & table from the **Schema Manager** page.

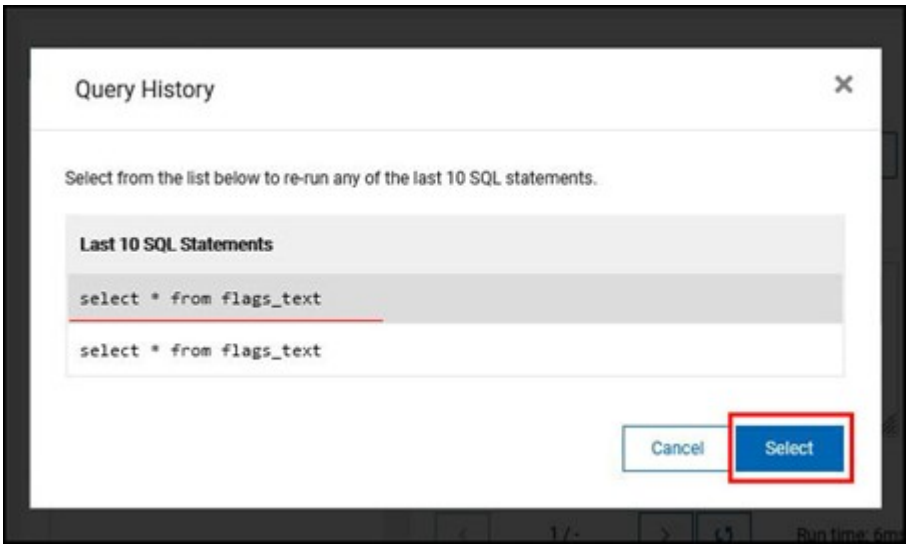


2. On **Info** tab, InformixHQ will show properties for database & table based on user's selection.
3. Click on **SQL Editor** tab, write the query in the text area and click on **Run** button.

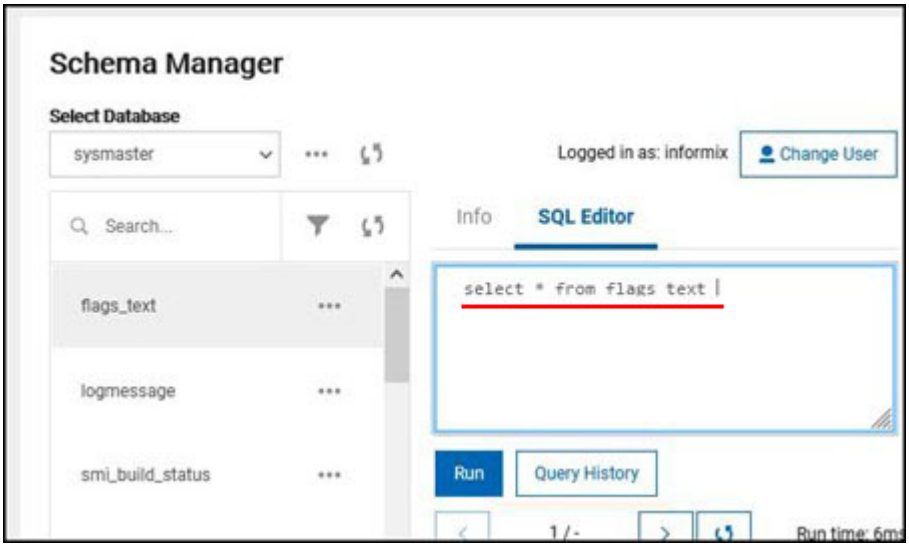
4. Clicking on **Run** button, will show results in the following two tabs:
    a. **Result :** Tab will show all the records available for the given query.
    b. **Execution Plan :** Tab will show SQL query execution plan in a tree format if plan is available for the given query.
5. To view last 10 queries, click on **Query History** button.



6. Clicking on **Query History** button, model will appear to select a query and run (last 10 queries).



7. Select query from **Query History** model and click on **Select** button. Selected query will appear in SQL Editor in text area.



Note:

a. InformixHQ saves last 10 executed queries for each database.

b. Saved queries will clear from history once user logs out from InformixHQ.

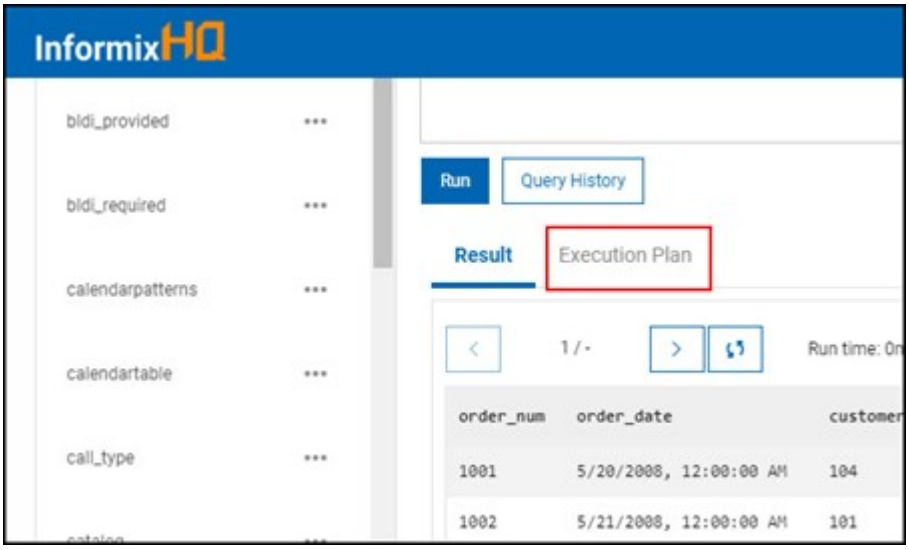
Using SQL Editor to view SQL execution plan

About this task

New feature of viewing query execution plan is now available in InformixHQ v1.6.0 onwards.

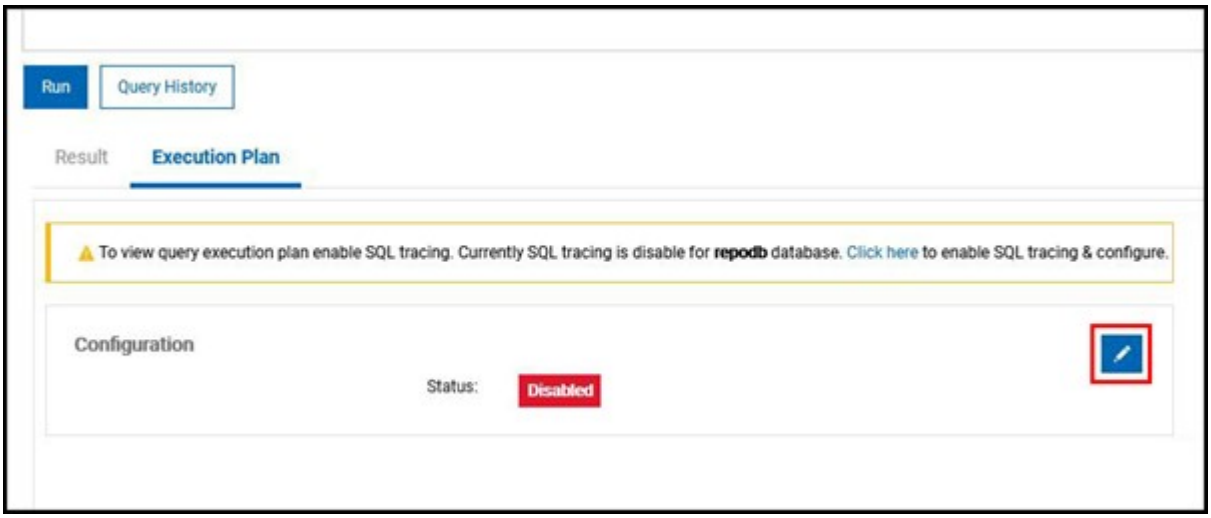
- **Pre-requisite:** For Informix server to generate execution plan, **SQL tracing should be enabled**. Once execution plan is generated by server, it will be available in the **Execution plan** tab of the SQL editor.

1. To view SQL execution plan, click on **Execution Plan** tab.

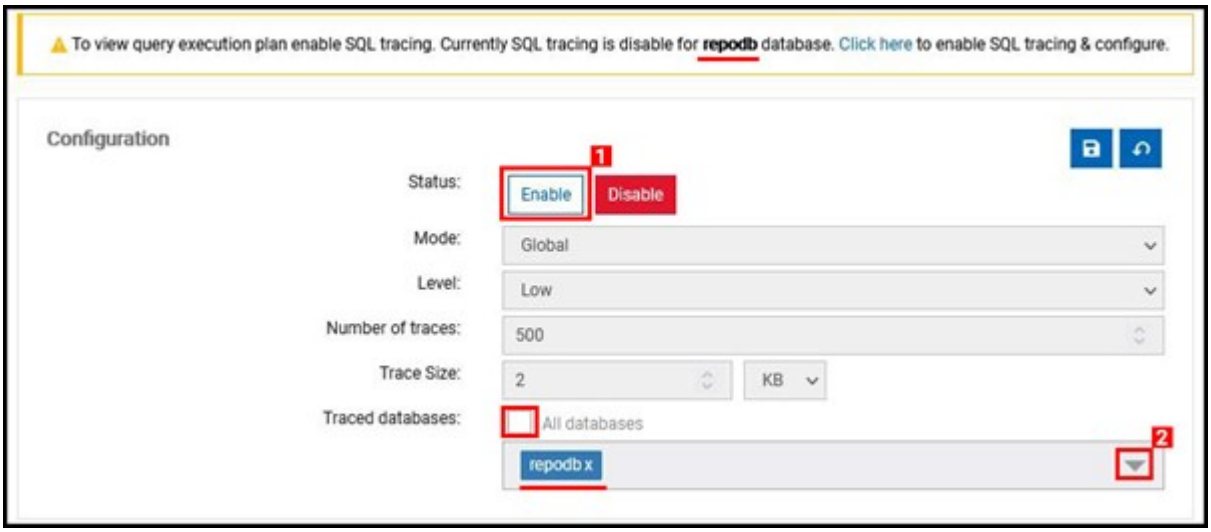


2. If SQL tracing is disabled, user can enable SQL tracing from **Execution Plan** tab also. Click on **Click here** option from **Execution Plan** tab.

3. Clicking on Click here, shows SQL tracing Configuration option. Click on **Edit Configuration icon** (Pencil icon).

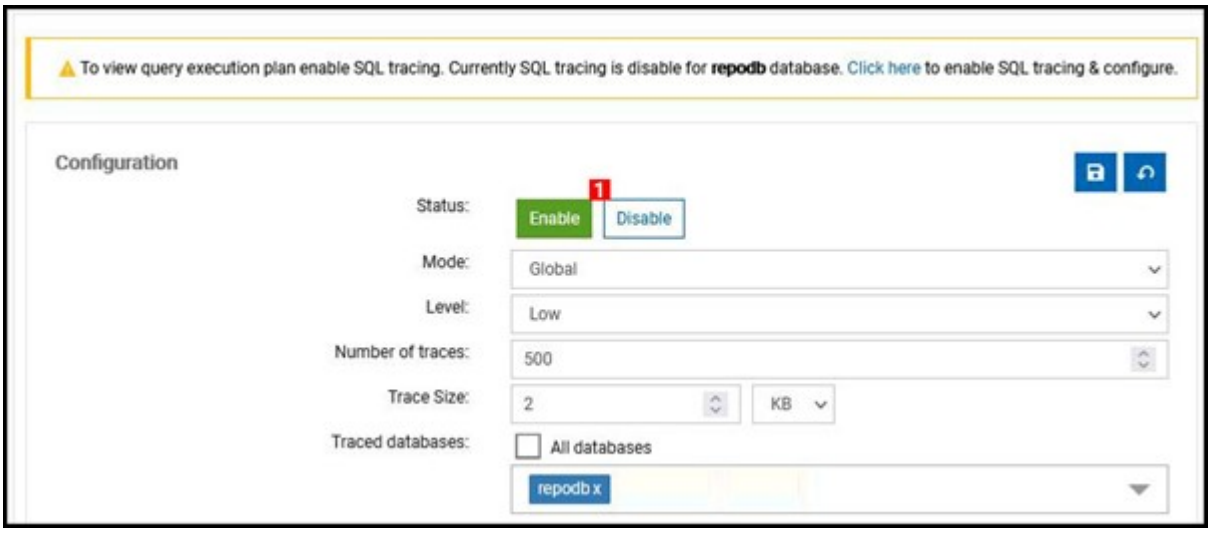


4. Clicking on **Edit Configuration** icon shows below SQL tracing option to configure.

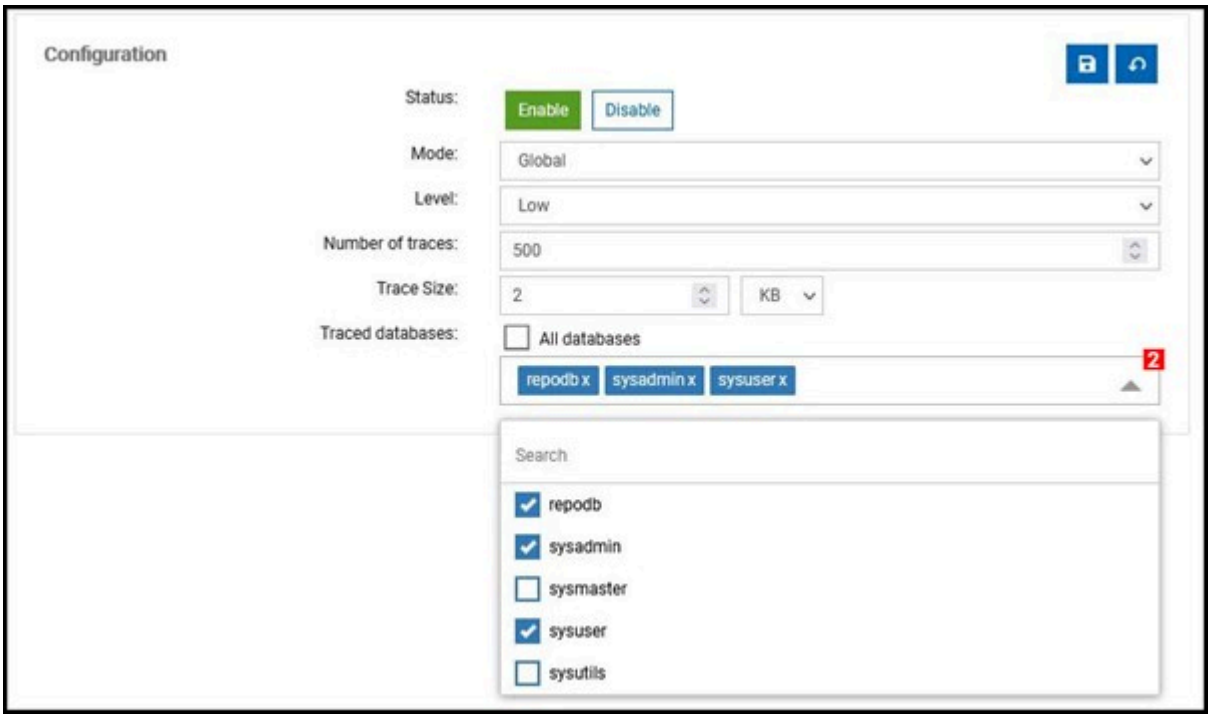


Note: Any selected database from the **Schema Manager** can be added in the **traced databases** list. User can remove selected database from the list by clicking **Enable** button.

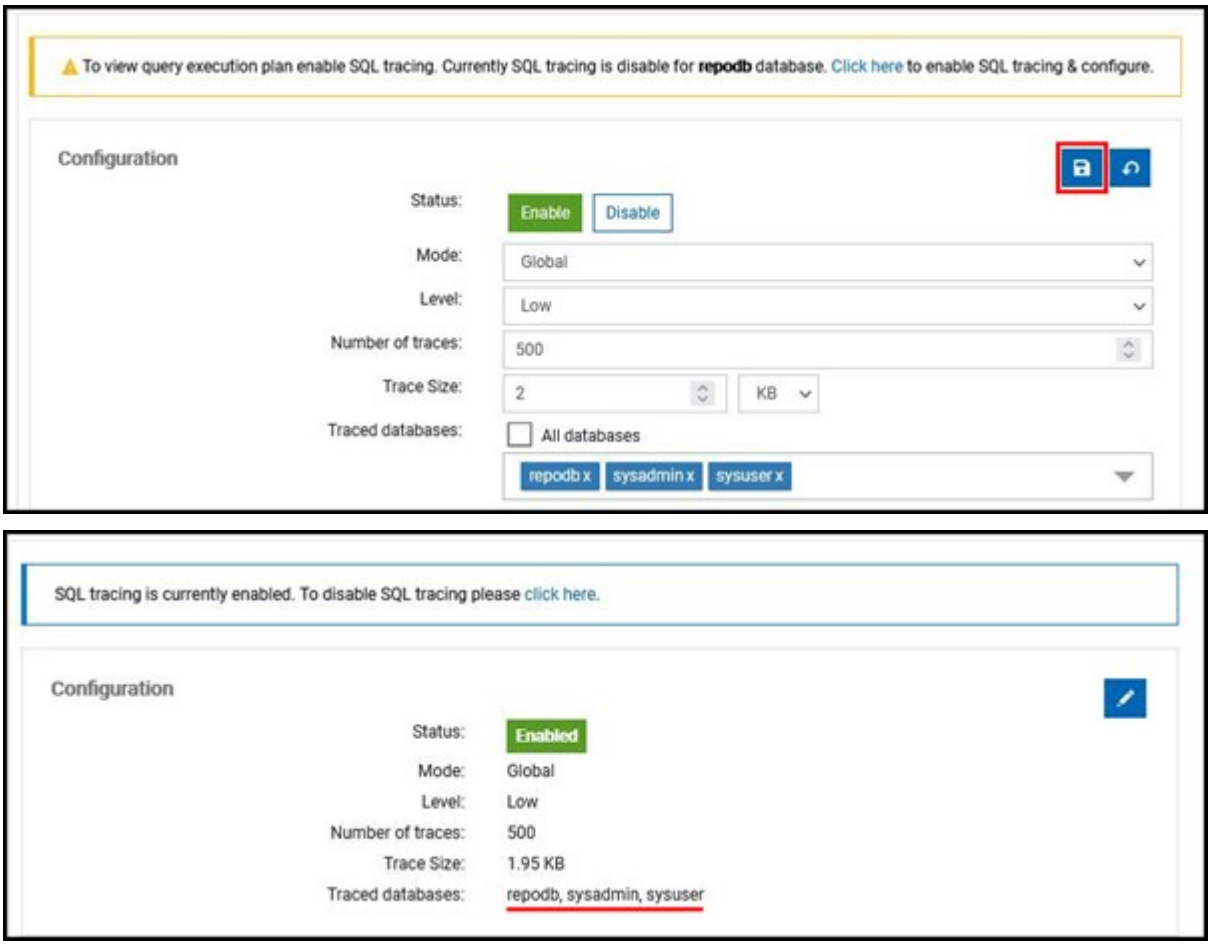
5. For enabling SQL Tracing on for selected databases or for all databases, click on **Enable**.



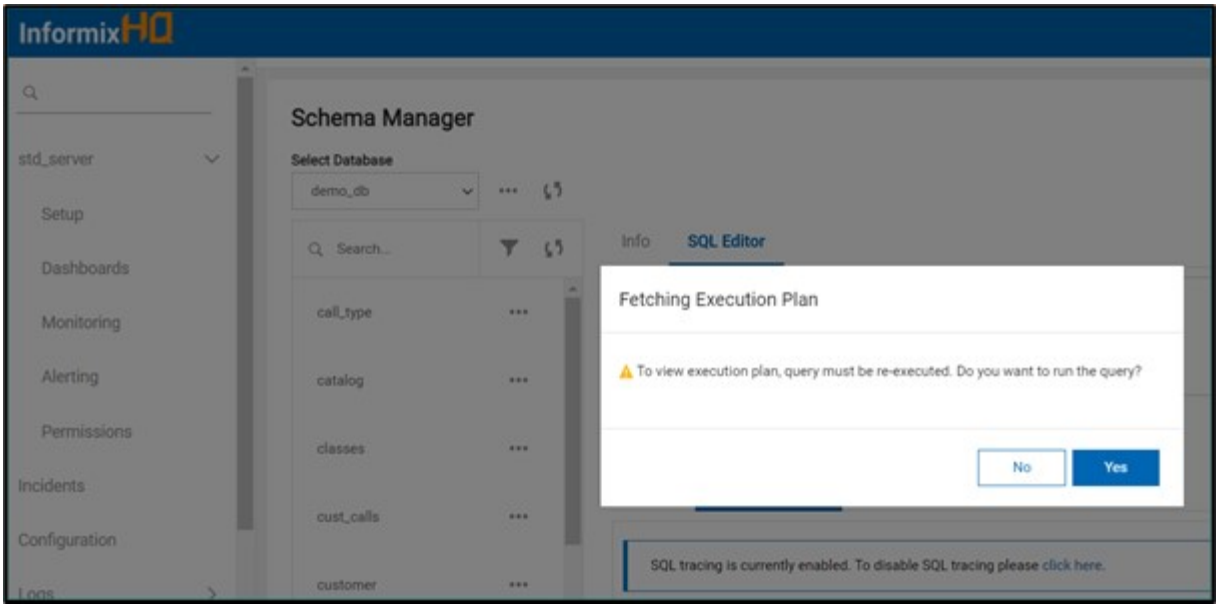
6. To turn on SQL tracing for the selected databases, click on the **databases drop-down** & select databases from the list OR select all databases based on the requirement.



7. Click on **Save** icon to save the configuration after changes are made.



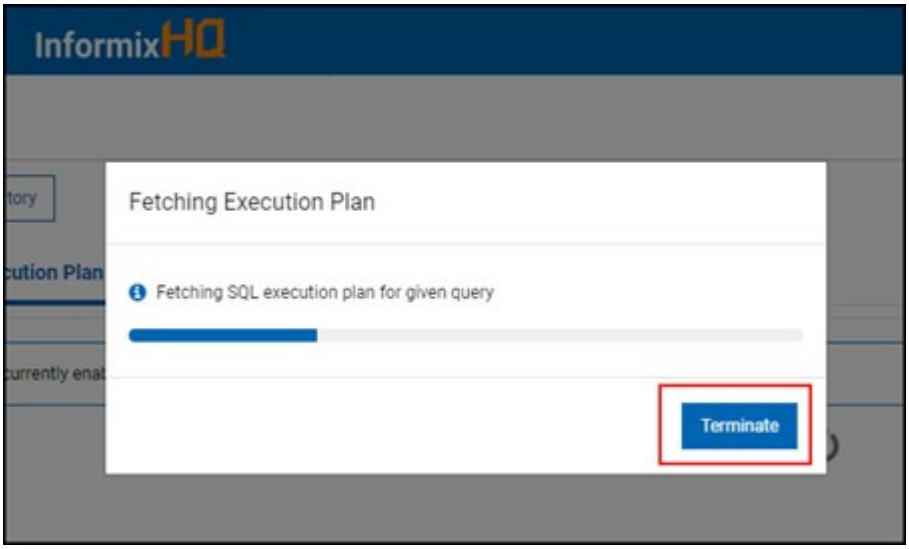
8. If SQL tracing is enabled from **Execution plan** tab, query needs to be re-executed for Informix server to generate execution plan. InformixHQ will ask user for the same, once SQL tracing is enabled form execution tab.



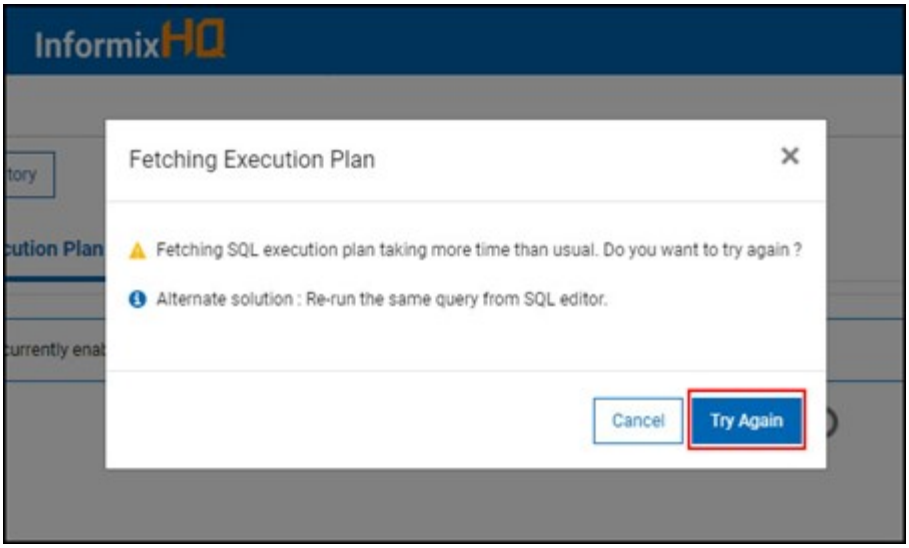
- If user chooses **Yes**, query will be re-executed and plan will be displayed if available.
- If user chooses **No**, then user needs to re-run the query to view query execution plan.

Note: If execution plan is readily available, InformixHQ will fetch and display the query plan for given query. InformixHQ will try fetching SQL execution plan within first 10 seconds. If SQL execution is not shown within couple of tries, it is recommended to run the query from the SQL Editor.

9. User can terminate fetching SQL execution plan by clicking on **Terminate** button.



10. User can retry fetching SQL execution plan by clicking on **Try Again** button if InformixHQ is unable to fetch SQL execution plan.



11. Once InformixHQ fetches SQL execution plan successfully, result will be shown in the **Execution Plan** tab.

Connection Manager

. The **Connection Manager** page allows you to visualize and manage CM unit, SLA and FOC for any CM.

Use the **Connection Manager** to:

View all connection units

1. Go to the **Connection Manager** page in InformixHQ.
2. Click on any CM row. All CM units will be displayed.
3. Click on any CM unit name. CM unit detail page will be displayed.

View/add/modify/delete SLA within connection units

1. Go to the **Connection Manager** page in InformixHQ.
2. Click on any CM row. All CM units will be listed.
3. Click on any CM unit name. SLA table will be displayed.
4. Click the **Modify/Delete** button on any row to edit or delete the existing SLA.
5. Click the **Add SLA** button to create new SLA.

View/add/modify FOC within connection units

1. Go to the **Connection Manager** page in InformixHQ.
2. Click on any CM row. FOC details will be displayed in the Cluster CM unit detail page.
3. Click the **Modify** button to Modify FOC values. Use the Slider to enable or disable FOC.
4. To change the FOC order, click the **Click here** button in CM unit page.

View or get alerts for number of SLA connections

1. Go to the **Monitoring** page in InformixHQ.
2. Click the **+Add Sensor** button.
3. Search for SLA connections. The existing SLA Connections sensor details will be displayed in the Add sensor page.

InformixHQ Server Settings

Users with System Administrator privileges, including the initial admin user created when InformixHQ is started for the very first time, will have a **System Settings** link in the drop-down menu shown when they click on the user icon in the top right corner of the application title bar. This link is used for making changes to the global InformixHQ Configuration.

The InformixHQ Configuration page includes settings for:

- Alerting Configuration – enable and configure InformixHQ to be able to send alerts to users via email, Twilio, or PagerDuty
- Data Cleanup Configuration – configure the schedule and settings for when InformixHQ runs its repository data cleanup jobs
- Sensor Management – create and manage custom SQL sensors for monitoring your Informix database servers
- Server Configuration – configure system-wide settings for the InformixHQ server, including the REST SQL session timeout.
- User Management – add users and edit their InformixHQ privileges

Configuring Alerting Notification

System administrative users for InformixHQ must enable which alerting notification services the InformixHQ server should use when an alerting incident occurs. If desired, you can configure and enable multiple alerting notification services.

InformixHQ supports the following alerting notification services, all of which can be configured and enabled on the **System Settings > Alerting Configuration** page.:

- **Email**

  InformixHQ can be configured to send emails through an external SMTP server when alerting incidents occur. To enable email notifications, the system administrative user must provide a SMTP server and port to use. Optionally, you can provide a user and password for authenticating to that SMTP server and a ***from email address*** that InformixHQ should use when sending alerting notification emails.

  Email notifications must first be enabled at the system level by the system administrative user. Then each individual InformixHQ user who wants email notifications must enable it for their email address on their My Settings->Alerting Configuration page.

- **Twilio**

  InformixHQ can be configured to send alerting incidents through Twilio. To enable Twilio notifications, the system administrative user must provide the Twilio account SID, authorization token, and phone number to send alerts from.

Twilio notifications must be enabled at the system level by the system administrative user. Then each individual InformixHQ user who wants Twilio notifications must enable it for their phone number on their My Settings->Alerting Configuration page.

- **Pager Duty**

  InformixHQ can be configured to send alerting incidents through Pager Duty. Pager Duty alerts are enabled globally by the system administrative user of InformixHQ. To enable Pager Duty notifications, the system administrative user must provide a PagerDuty service key.

  Pager Duty alerts do not need to be enabled by each individual user of InformixHQ. When Pager Duty alerting notifications are enabled by the system administrative, all alerting incidents that occur in InformixHQ will be sent to the specified Pager Duty service key.

  InformixHQ sends PagerDuty notifications through REST using Pager Duty's Events API v1.

- **Run Script**

  The InformixHQ server can be configured to run a local script whenever an alerting incident occurs. Script notifications in InformixHQ provide an extensible way to integrate InformixHQ's alerting with any alerting mechanism used by your organization.

  Script notification is enabled globally by the system administrator on the **System Settings > Alerting Configuration** page. When script notification is enabled, the InformixHQ server will run the specified script whenever an alerting incident occurs on any server or group managed by InformixHQ.

  Before the InformixHQ server runs your script, it will set the following environment variables to contain information about the alerting incident that occurred:
    - ALERT_ID – id of the alerting incident
    - ALERT_TIMESTAMP – timestamp of the alerting incident
    - ALERT_SUMMARY – summary text describing the alerting incident
    - ALERT_MESSAGE – detailed message describing the alerting incident
    - SERVER_ID – id of the Informix server on which the alerting incident occurred
    - SERVER_ALIAS – alias of the Informix server on which the alerting incident occurred
    - GROUP_ID – id of the parent group containing the Informix server on which the alerting incident occurred
    - GROUP_NAME – name of the parent group containing the Informix server on which the alerting incident occurred
    - EVENT_URL – an url link to view the alerting incident in InformixHQ

  A sample use case for the "Run Script" alerting service would be, suppose your organization uses a third party alerting service that InformixHQ does not have native support for. That service requires you to POST a JSON document to a specific URL to generate an alert. You can write a script that reads in the environment variables set by InformixHQ, reformat that data into a JSON document as required, and then use curl to send a REST POST request to your organization's alerting service.

Creating Custom Sensors

Use the **System Settings > Sensor Management** page to create and manage custom SQL sensors. Custom sensors allow you to define data to be collected by the InformixHQ agent. Data from custom sensors can be graphed on a custom dashboard. You can also use custom sensors to define alerting conditions in InformixHQ.

Each custom SQL sensor is based on a single SQL query to be run by the agent on the sysmaster database of the Informix server being monitored. Any data that can be returned by a SQL query against sysmaster can be monitored by the agent.

Only System Administrative users of InformixHQ can define custom SQL sensors. Once defined, custom sensors can be added to any server or group's monitoring profile.

To define a custom SQL sensor, go to the **System Settings > Sensor Management** page and click **Create Sensor**.

For each custom sensor, define the following:

- **SQL**
  - Define the **SQL query** to be run against the sysmaster database on the Informix server being monitored.
    - Select a sample server to run the query against to preview the data.
  - Optionally, specify the **Transpose** option to have the agent transpose (or pivot) the data returned by your SQL query based on a selected column.
  - Optionally, specify a **Primary Key** column for your query.
    - If your query returns multiple rows describing multiple objects (e.g. dbspaces), use the primary key column to define the unique identifier for each object.
- **Metrics**
  - Define a metric for each column returned by your query. Each metric will have:
    - **Name**: A display name for the metric which will be used to display this metric's data in the InformixHQ UI.
    - **Unit**: Optional. Defining the unit as percentage or bytes will direct the InformixHQ UI to format the sensor data values before they are displayed in the UI.
    - **Default Value**: Optional. A default value to be used for the metric if the query returns null for that column.
    - **Calculate Delta**: Optional. If enabled, the agent will store the difference per second between the latest reading and the previous reading.
- **Sensor Metadata**
  - A unique **id** for the sensor
  - A display **name** for the sensor
  - An optional **description** of the sensor
  - A default **run interval**
  - A default **data retention interval**

User Settings

Each user will have a Settings link in the drop-down menu shown when they click on the user icon in the top right corner of the application title bar. This link is where users can configure their own personal settings.

The user settings page includes links for:

- Alerting Configuration – enable alerts for your username and choose how to get alerts (email or Twilio). This requires that the InformixHQ administrator has configured the corresponding alerting services.
- Changing your password

Configuring User Alerting Notification

InformixHQ provides two alerting notification options that can be enabled for each individual user: email and Twilio. To enable these notification services, the system administrative user for InformixHQ must enable email and/or Twilio notifications at the system level on the InformixHQ System Settings > Alerting Configuration page. Then each individual user must enable email or Twilio notifications for their user on the **My Settings->Alerting Configuration** page, providing their email address or phone number respectively.

InfromixHQ also supports Pager Duty and Run Script based alerting notifications, but these are configured globally in InformixHQ and are not enabled for each individual user.

Users and Permissions

Only users with System Administrator privileges can add or delete users or modify their privileges. The initial **admin** user that is created when the InformixHQ server starts for the first time has System Administrator privileges. Additional users with System Administrator privileges can be created.

To add a user, delete a user, or modify their InformixHQ system privileges and permissions, a System Administrator user should click on their user icon in the top right corner of the title bar and go to the **System Settings > User Management** page.

When adding a user, you can optionally make the new user a System Administrator if you want them to manage users and configure InformixHQ. System Administrators automatically have full access (**Read**, **SQL**, and **Admin** permissions) on all servers and groups. When creating a new non-System Administrator, you have the option to grant **Read, SQL,** and/or **Admin** permissions for that user to any servers and groups that have been added to InformixHQ.

Read permissions allow a user to view information about a server in the UI, **Admin** permissions allow a user to execute administrative actions to make changes to a server, and **SQL** permissions allow a user to run SQL queries against that database server on the **Schema Manager** page.

Note: These three permissions are mutually exclusive. Granting **Admin** permissions does not automatically include **Read** permission.

In order to run administrative actions on an Informix database server, a user must have **Admin** permissions on that server and admin credentials must have been provided when adding that server to InformixHQ. You can review and edit server

credentials by going to the server's **Setup** page or by going to the group dashboard, selecting the server's card, and clicking **Edit**.

Permissions for a server or a group in InformixHQ are inherited. If a user is granted **Read** or **Admin** privilege on a group, the same privilege will be granted on all servers and sub-groups within that group.

While the **System Settings > User Management** page allows you to view and edit the complete permissions for each individual user of InformixHQ, there is also a **Permissions** page specific to each server or group that allows to you to see (and edit) all users who have permissions to that particular server or group. You can access this page by navigating to a particular server or group and clicking on **Permissions** in the side-bar menu.

HDR Secondary Operations

Introduction

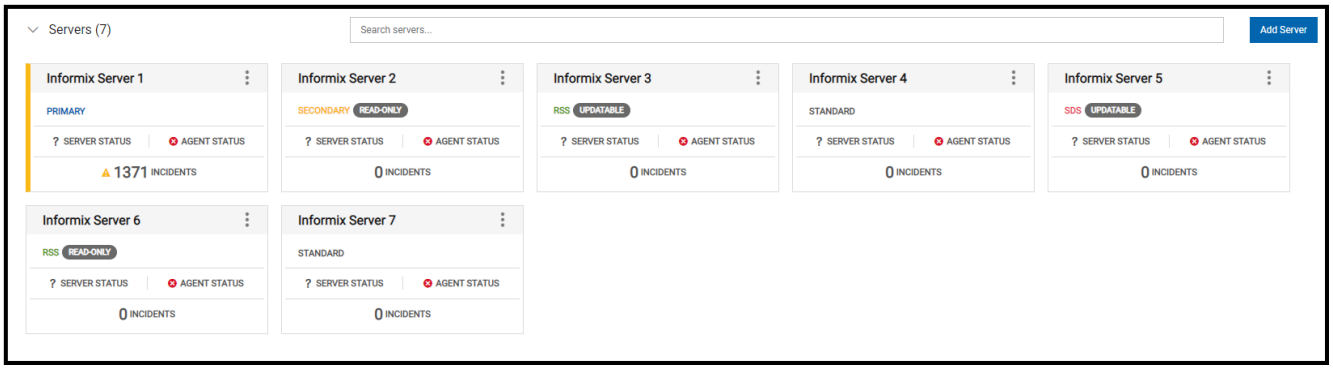
HDR cluster consists of Primary and Secondaries. InformixHQ treats every server added as individual server. Depending upon server type in cluster, Informix Server allows or restricts certain acions on specific server types.

From 14.10.XC7 (InformixHQ v.1.6.0) onwards, InformixHQ has handled all such operations on secondary servers in the cluster.

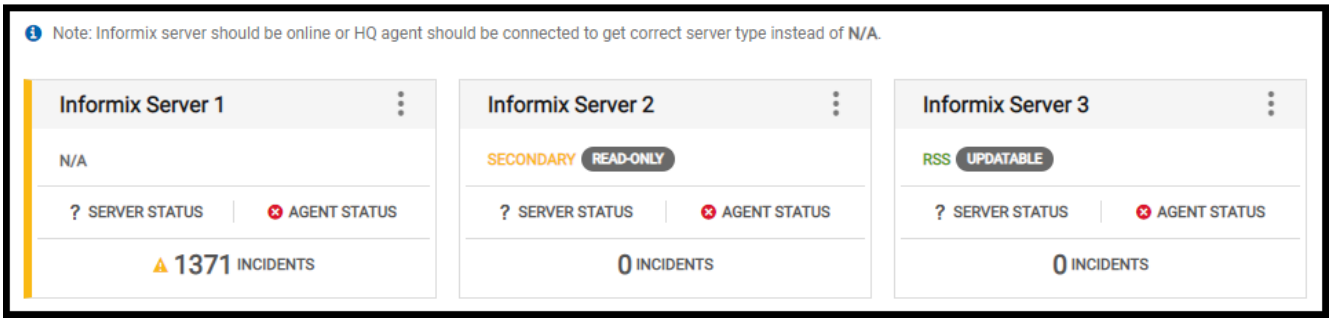
User needs to add primary as well as secondaries in InformixHQ as different servers.

HDR Secondary in Informix HQ

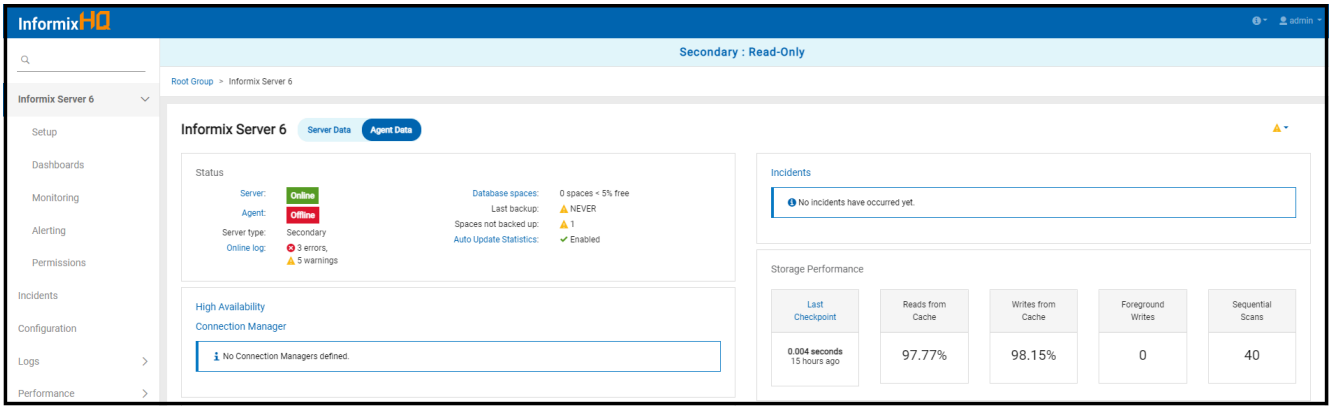
Server type is available on server card with different colour for each type. Servers apart from Primary & Standard types have “Updatable” or “Read-Only” access mode.



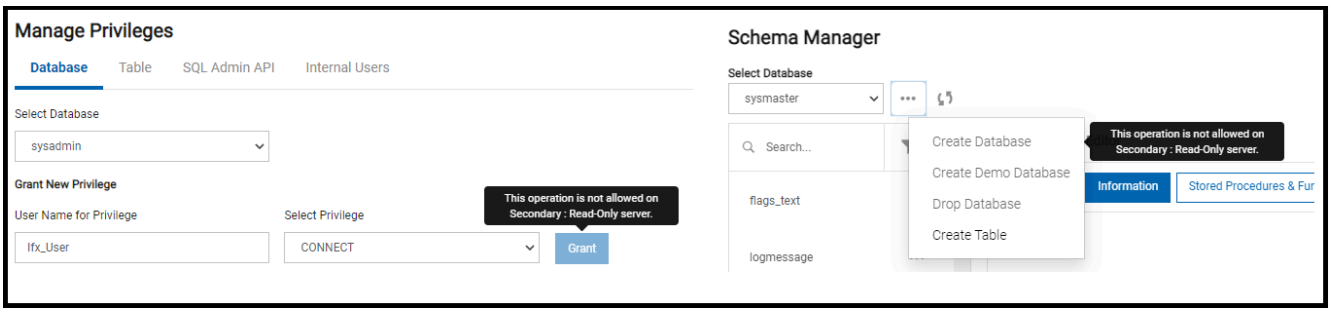
N/A will be shown for the server that is offline since InformixHQ can not determine the type of the Informix server if unable to connect with it in the first place. It will be updated in scenarios mentioned in Server types and access mode update scenarios on page 119.



HDR banner on each page is shown for servers apart from Primary & Standard types. HDR banner contains server type and server access mode.



Operations that are not allowed on HDR Secondary are disabled in UI with a pop-up message that shows server type and access mode. For more details, refer HDR Secondary operations allowed/disallowed on page 118.



HDR Secondary operations allowed/disallowed

Following table lists the HDR operations allowed/disallowed based on different types of the Informix servers and its access modes in InformixHQ.

Operation	Primary	Secondary		RSS		SDS	
		Updatable	Read-only	Updatable	Read-only	Updatable	Read-only
Run Checkpoints, Automatic checkpoints ON/OFF	Yes	No	No	No	No	No	No
Privileges: SQL Admin API -> Grant, Edit and Revoke Privileges	Yes	Yes	No	Yes	No	Yes	No
Privileges: Databases -> Grant, Edit and Revoke	Yes	Yes	No	Yes	No	Yes	No
Privileges: Tables -> Grant, Edit and Revoke	Yes	Yes	No	Yes	No	Yes	No
SQL Tracing: Enable, Suspend, Resume and Disable	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Memory-> Low Memory Manager Edit	Yes	Yes	No	Yes	No	Yes	No
Schema Manager-> Create, Drop Database	Yes	Yes	No	Yes	No	Yes	No
Virtual Processor-> Add and Drop VPs	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Spaces-> Create, Edit, Delete, Add Chunk and Extend Space	Yes	No	No	No	No	No	No
Chunks->Add, Edit, Delete, and Extend Chunk	Yes	No	No	No	No	No	No
Pools->Add, Edit, and Delete Entry	Yes	Yes	No	Yes	No	Yes	No
Backup: Configuration-> Edit and Delete	Yes	No	No	Yes	Yes	No	No
Backup: Status-> Run Level 0,1,2 backup	Yes	No	No	Yes	Yes	No	No
Recovery Logs-> Move, Add, Delete and Switch Log	Yes	No	No	No	No	No	No
Session-> Kill Session	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Note: Above table is based on InformixHQ v1.6.0. Some permissions may vary depending on specific Informix server version.

Server types and access mode update scenarios

Following are the cases when Informix server type and access mode are updated in UI on root page in InformixHQ:

- On running InformixHQ server post v1.5.0 --> When InformixHQ server is started (run HQ Jar) post version 1.5.0, it connects and fetches Informix server types and access mode of all the Informix servers which are added in InformixHQ and are online.
- InformixHQ Agent is connected --> If Informix server is online and InformixHQ Agent is connected to it. Agent fetches Informix server type and access mode and following are the cases for it:

1. Case 1 - Via JDBC connection: If InformixHQ agent is running on a remote host other than Informix server.
2. Case 2 - Via onstat - command : If InformixHQ agent is running on the same host where Informix server is running.

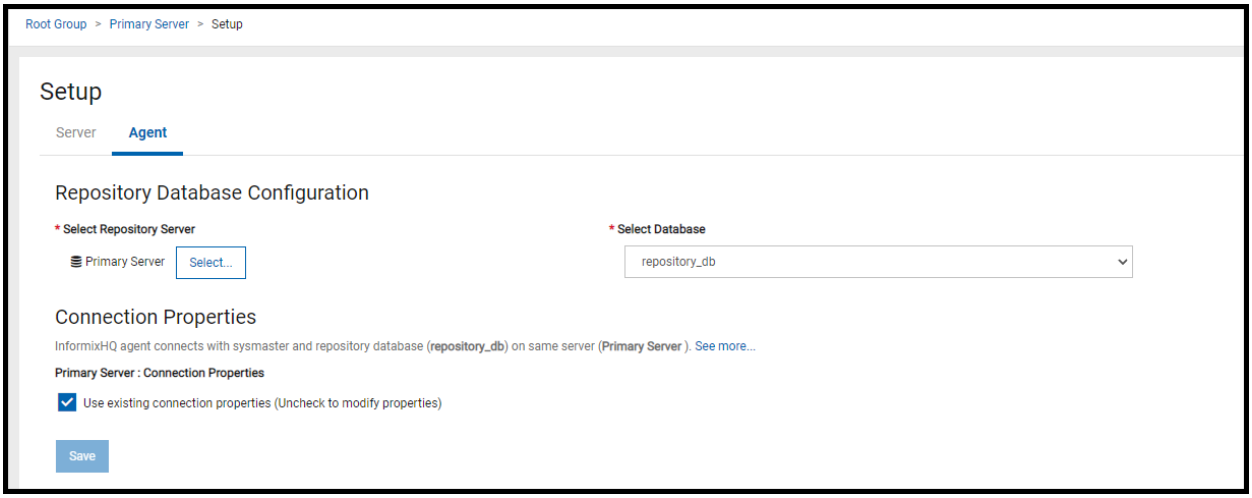
- Open server card:-> When user opens server information page and Informix server is online.

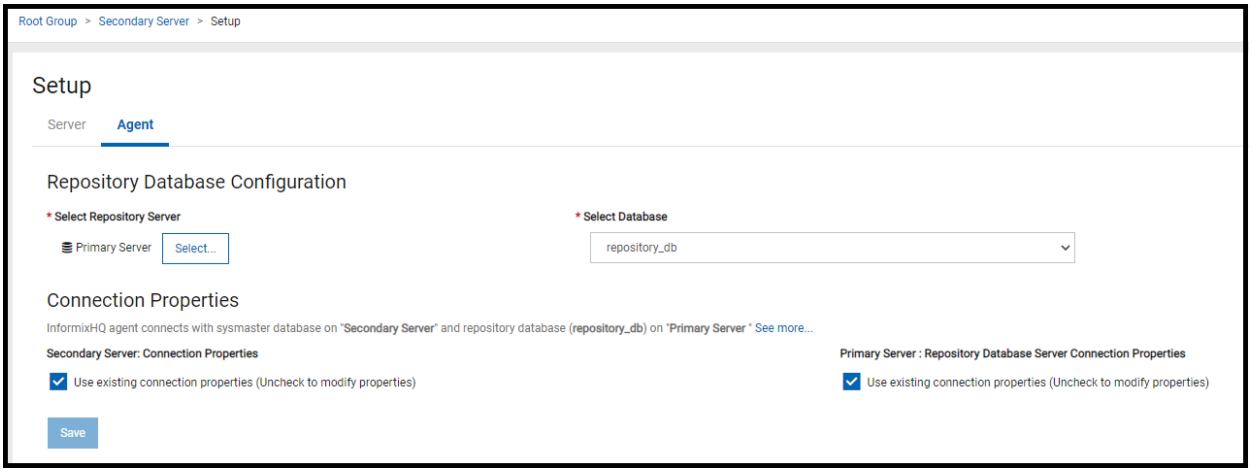
Failover scenarios in InformixHQ

User needs to add both primary and secondary servers in InformixHQ to monitor them separately. In case of failover InformixHQ Agent will not be able to connect to failed server and there will be no monitoring data collection. User needs to take corrective action as explained below to continue with HQ Agent data collection.

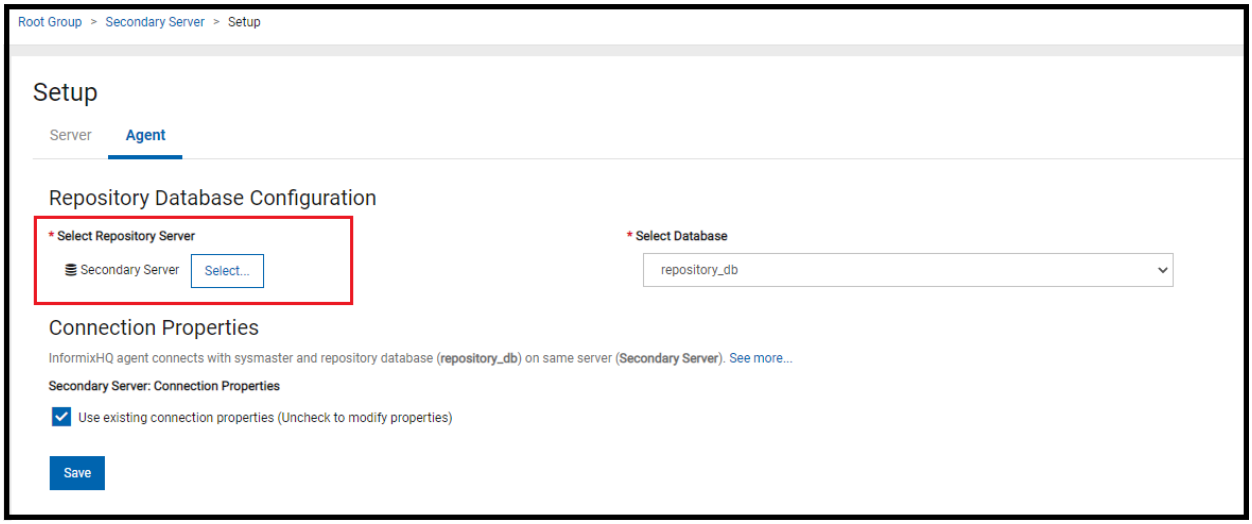
1. Case 1 - Primary & Secondary servers have same repository database on primary server:

   In this case if primary server fails/is offline, user has to manually change the repository database server configuration in secondary server which is added in InformixHQ. For example:





   As primary server is offline, changed repository server to secondary.



2. Case 2 - Primary & Secondary servers have same repository database on secondary(updatable) server:

   In this case if secondary server fails/is offline, user manually has to change the repository database server configuration in the primary server which is added in InformixHQ. This case is just vice versa of case 1 which is described above.

3. Case 3 - Primary & Secondary servers have their respective repository database on their server itself:

   In this case each server has its own repository database, so user does not need to do any configuration changes in HQ agent setup page in case of failover.

InformixHQ Server Shutdown

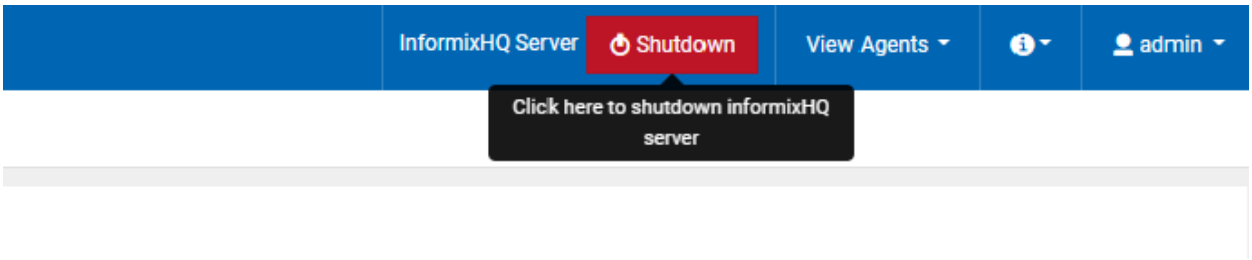
About this task

Using **InformixHQ Server Shutdown** button is an easy way to shut down the InformixHQ server directly from the InformixHQ UI instead of command prompt.

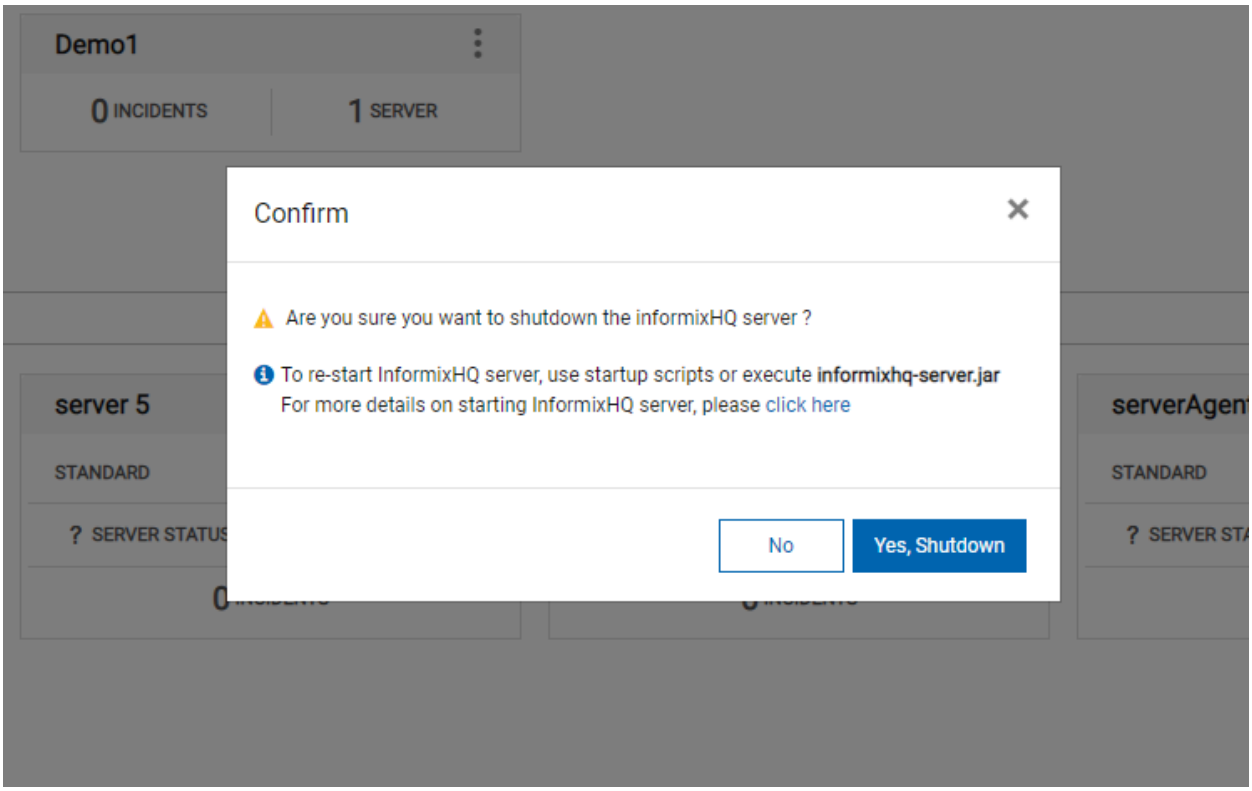
Use the following steps to shut down InformixHQ server:

1. New option to shut down InformixHQ server is available in the top right corner from InformixHQ UI.

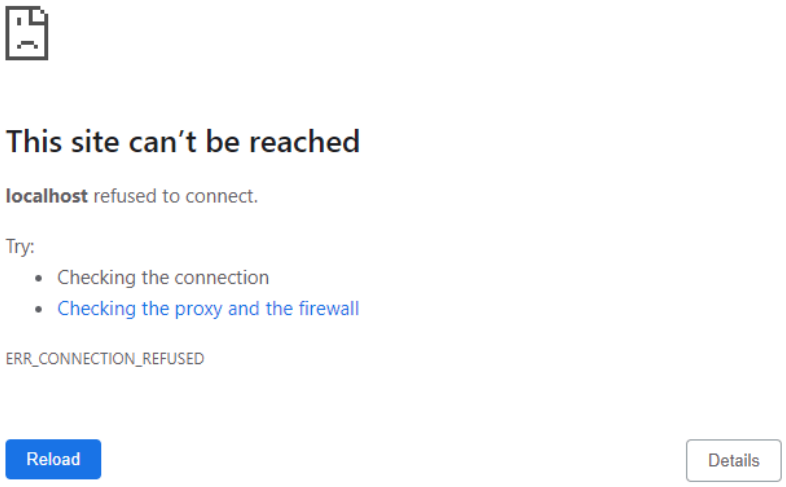
   Note: This **Shutdown** button is only available for InformixHQ system administrator role.



2. Once user clicks on the **Shutdown** button, InformixHQ will show a pop up to confirm the shutdown action.



3. Clicking on **No** button will cancel the InformixHQ shutdown request and user will remain logged in InformixHQ.
4. Clicking on **Yes, shutdown** button will shut down the InformixHQ server.
5. As InformixHQ server URL will not be accessible anymore, browser will show **refused to connect** message as shown below.



InformixHQ Agent

The InformixHQ agent is a lightweight Java 11 based monitoring agent. It should be installed on the host machine for each Informix database server that you want monitored by InformixHQ.

The agent runs alongside the database server, gathering database statistics through JDBC connections to sysmaster. The InformixHQ agent only needs read access to the database server.

By running the agent directly on the Informix server host machine, the agent is also able to gather and monitor OS statistics which can be just as critical in evaluating and tuning the Informix database server performance metrics.

InformixHQ Agent Setup

This topic explains how to configure InformixHQ agent from InformixHQ UI.

Select repository database

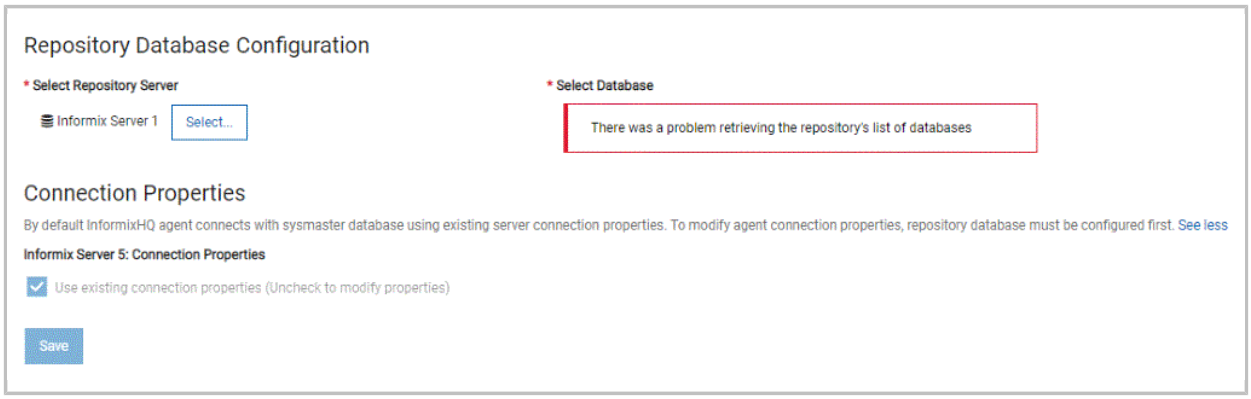
Repository server is the server which contains Repository database, which will be used to store all the monitoring data collected by InformixHQ agent. Without selecting a repository database, user is not allowed to save agent setup changes. Repository database server can be any Informix server specified in InformixHQ User Interface.

TIP: User should create a dedicated database to store metrics to be captured by InformixHQ agent. A new database can easily be created from Schema Manager on page 85 page. Similarly metrics can be defined by adding Sensors from Motitoring page.

Following are two scenarios if repository database is not available for any reason:

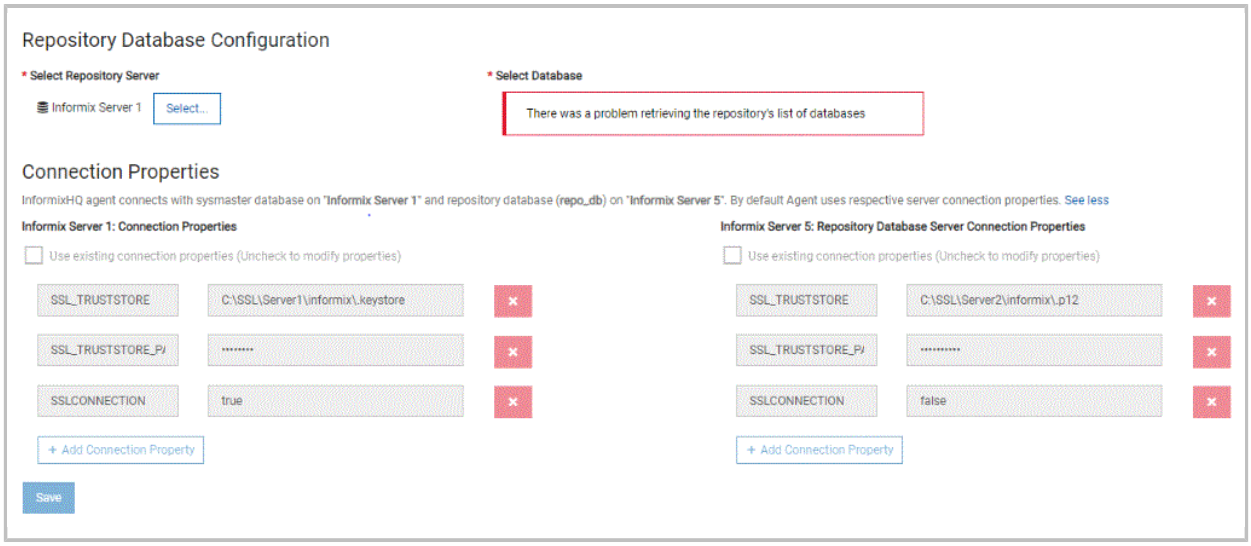
1. ***InformixHQ Agent is not configured yet:***

   In this scenario, InformixHQ agent configuration is not set yet by the user, probably user is setting up InformixHQ agent for the first time. InformixHQ agent connects with monitoring server using existing Informix server connection properties.



2. ***InformixHQ Agent is already configured:***

In this scenario, previously saved InformixHQ agent configuration will be shown in read only mode. Basically, whenever user runs agent jar, it will connect using already saved configuration.



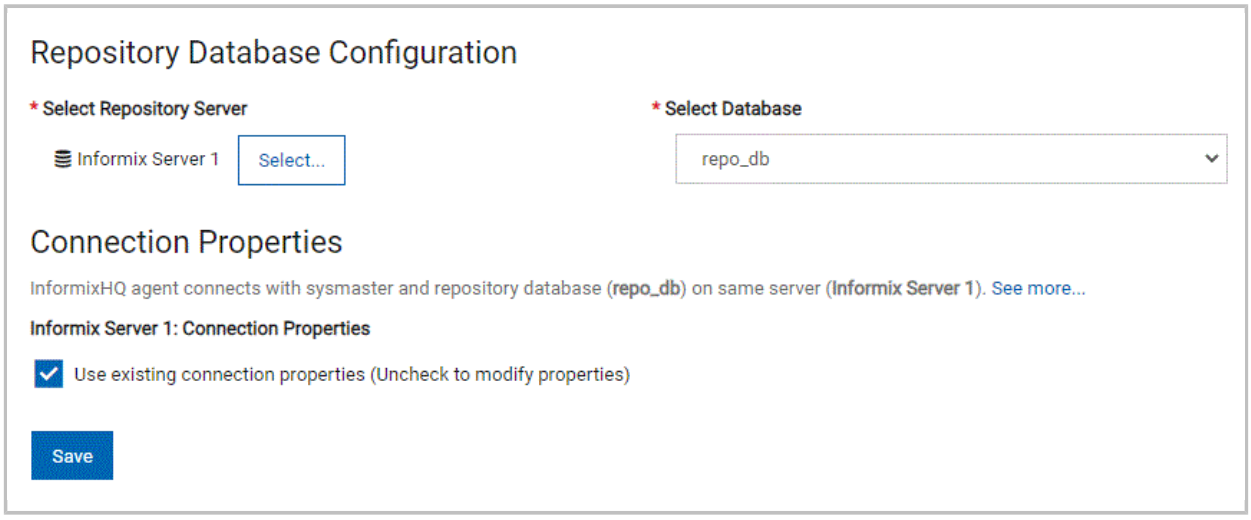
Add Connection Properties

Separate connection properties can be specified for the InformixHQ agent. Users can specify connection properties for monitoring server and repository database server, respectively. Following are two cases while specifying connection properties.

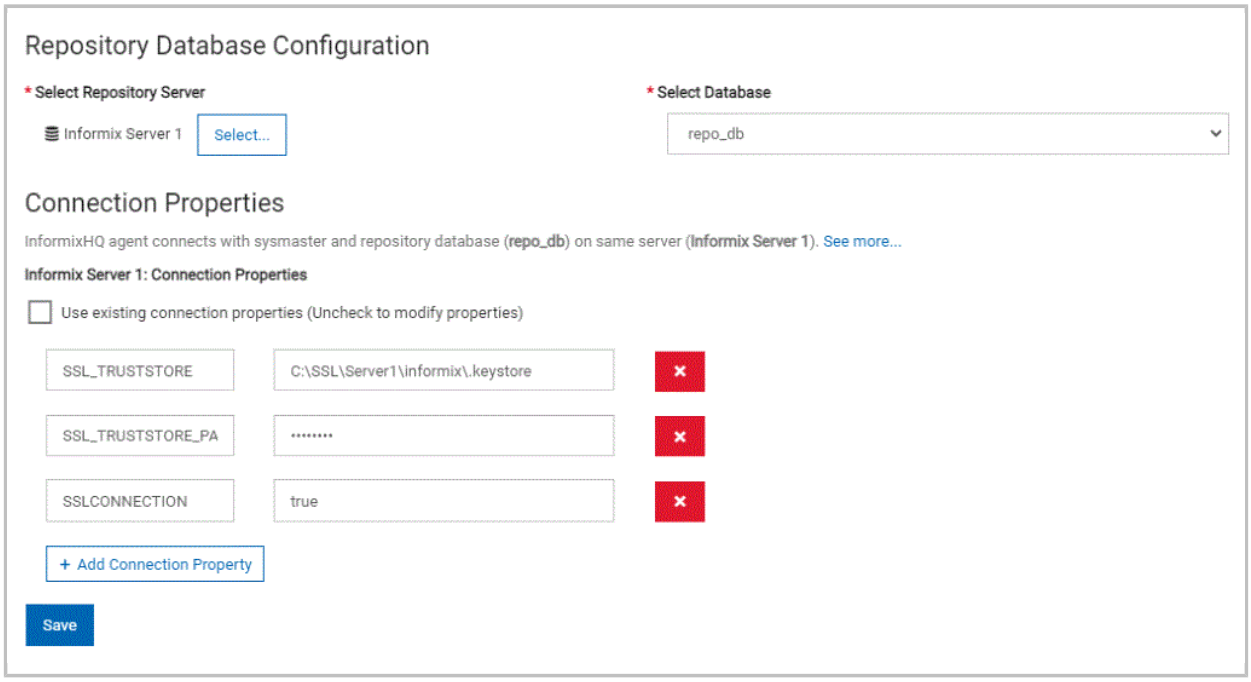
1. ***Repository database is located on monitoring Server:***

   In this scenario, InformixHQ agent will connect with monitoring server and repository database using the same connection properties.

   By default, InformixHQ agent will use Informix server connection properties (from Informix server setup page) to connect with monitoring server and repository database. (When checkbox is checked)



If needed user can add/modify InformixHQ agent connection properties by unchecking the checkbox as follows:

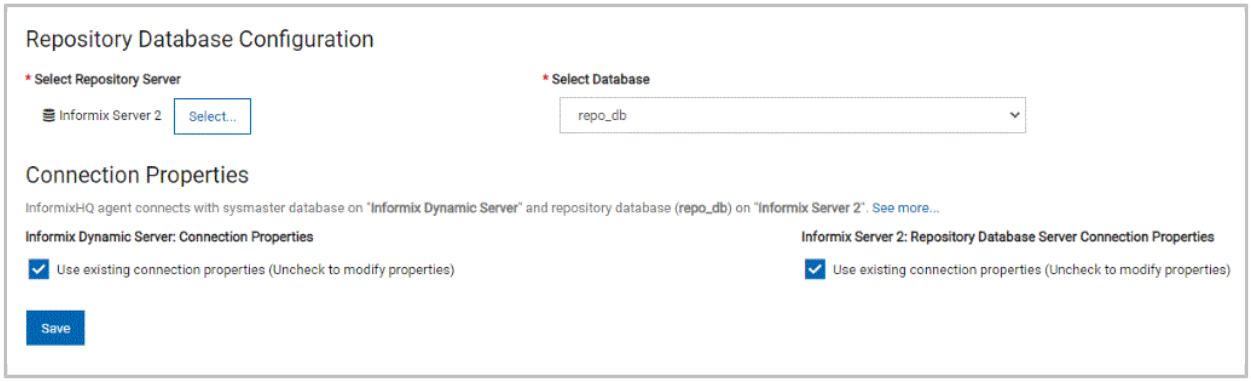


This will be specifically useful for providing different SSL keystore path for InformixHQ agent.

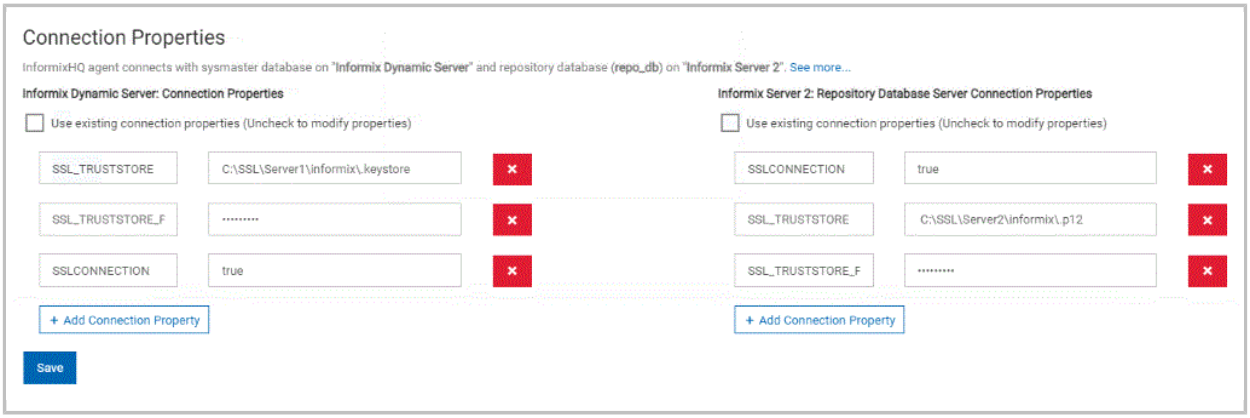
2. ***Repository database server is located on a different server:***

   In this scenario, InformixHQ agent will connect with monitoring server and repository database using different connection properties.

   By default, InformixHQ agent will use connection properties of respective Informix servers to connect with monitoring server and repository database, respectively. (When checkbox is checked)



If needed user can add/modify InformixHQ agent connection properties for monitoring server and repository database separately by unchecking respective checkboxes as follows:



This will be specifically useful for providing different SSL keystore path for monitoring server and repository database, respectively. See (Compatibility matrix for Java with SSL Keystore format on page 126).

In both scenarios, if checkbox is unchecked, at least one connection property should be added to enable save button.

Note:

If Repository database is not configured or could not be connected due to any reason, then user will not be able to add/modify agent connection properties.

If already configured Repository server is removed from InformixHQ, then such server will not be visible in agent setup page under "Select Repository server". In this scenario, all the custom agent connection properties previously saved, will reset and user needs to re-enter all the custom connection properties in Agent Setup page once new repository server is selected.

Compatibility matrix for Java with SSL Keystore format

Following are keystore formats supported based on Java providers:

Java Provider	Type
IBM Java(v11)	JKS
Oracle Java(v11)	JKS, PKCS

Related information

Schema Manager on page 85

Secure sockets layer protocol

InformixHQ Remote Agent mode

This topic explains InformixHQ Remote Agent mode and its limitations.

What is Remote Agent mode?

Remote agent mode refers to a setup where the InformixHQ agent and the Informix database server are running on different machines.

Limitations with Remote Agent mode:

- **Incomplete Metrics:**

  When the InformixHQ agent operates remotely, it lacks access to the Informix database server machine to run some sensors and processes. This setup results in unavailable metrics, such as operating system-specific data and file system metrics, etc. For example, the InformixHQ Agent can not show **Monitor CPU**, **Memory usage** or **Disk I/O** metrics which are essential for monitoring.

  Note: Upon connecting to the remote agent, user will see the following log message inside the remote agent log file:

  "When running the agent in remote mode, some metrics will not be available, for example "operating system or file metrics".

- **Sensors:**

  In remote agent mode, while creating the dashboard, if user adds any of the sensors listed below on dashboard, no data for these sensors will be shown on the dashboard. The list of sensors given below is a sample list. Users may add many other sensors while creating the dashboard.
  - AF Files
  - Backups per DBspace
  - Online log
  - Operating System CPU
  - Operating System CPU per core
  - Operating System Disk I/O
  - Operating System Disk Utilization
  - Operating System Memory
  - Operating System Network I/O
- **System Resources:**

The system's resource monitoring displays a list of charts. However, upon connecting to the remote agent, the charts show no data. Below is the list of charts which do not have data to show.

- Overall CPU usage
- OS Memory
- Disk I/O Activity
- Network I/O – Receive
- Network I/O Transmit

- **Memory:**

  Inside **System Resource->Memory** submenu, **Agent Data** toggle shows data related to the memory. However, upon connecting to remote agent, the charts do not have data to show for **OS Memory** and **Database Shared Memory**.

InformixHQ Agent Configuration Parameters

A properties file is required to run the InformixHQ agent.

When starting the agent, you can pass the properties file name as part of the start command. Otherwise, the agent will look for a properties file named **monitoring-agent.properties** in the classpath.

An example configuration file documenting the supported InformixHQ agent configuration properties can be found at `$INFORMIXDIR/hq/monitoring-agent-example.properties`.

- Required configuration properties
  - serverInstance.id on page 129
  - server.host on page 129
  - server.port on page 129
- Optional configuration properties
  - dataSource.IFX_ISOLATION_LEVEL on page 129
  - pool.connectionTimeout on page 129
  - pool.idleTimeout on page 129
  - pool.maximumPoolSize on page 129
  - pool.minimumIdle on page 129
  - ssl.enable on page 129
  - ssl.keystore.file on page 130
  - ssl.keystore.password on page 130
  - target.informixdir on page 130
  - target.onlinelog on page 130
  - target.ping.frequency on page 130
  - user.password.minLength on page 130
  - user.password.requireLowerCase on page 130
  - user.password.requireUpperCase on page 131
  - user.password.requireNumber on page 131
  - user.password.requireSpecialCharacterFromSet on page 131

serverInstance.id

The id of the Informix database server in InformixHQ. You find the server's id on the server's Setup page in the InformixHQ UI.

server.host

The host name on which the InformixHQ server is running.

server.port

The port on which the InformixHQ server is running.

dataSource.IFX_ISOLATION_LEVEL

Specifies the isolation level to set on JDBC connections to the target and repository Informix database servers.

The default value is 1 (DIRTY READ).

pool.connectionTimeout

Specifies the number of milliseconds to wait for a JDBC connection to the target or repository Informix database server to be established before it times out. The default value is 5000 (5 seconds).

pool.idleTimeout

Specifies the number of milliseconds that a JDBC connection can be idle in the connection pool before it is closed.

The default value is 60000 (1 minute).

pool.maximumPoolSize

The maximum number of JDBC connections in each connection pool. The InformixHQ agent will maintain a connection pool for the target database server and another a connection pool for the repository database server. The **pool.maximumPoolSize** puts a cap on the total number of open JDBC connections that can be established to each database. The default value is 3.

pool.minimumIdle

The minimum number of idle JDBC connections in each connection pool. The InformixHQ agent will maintain a connection pool for the target database server and another a connection pool for the repository database server. Setting **pool.minimumIdle** to zero indicates that all JDBC connections in the connection pool should be closed when they exceed the **pool.idleTimeout**. Setting **pool.minimumIdle** to a positive integer indicates the number of connections that should be kept open in the connection pool even when they exceed the **pool.idleTimeout**.

The default and recommended value is 0.

ssl.enable

Whether SSL should be enabled to secure web socket communication between the agent and the InformixHQ server. Set this value to true if the InformixHQ server port specified in **server.port** is an HTTPS port.

Note: if **redirectHTTPtoHTTPS** is set to true in the InformixHQ server configuration file, you must set this value to true in the agent configuration file.

If **ssl.enable** is set to true, you must also configure the **ssl.keystore.file** and **ssl.keystore.password** configuration properties.

The default value is false.

ssl.keystore.file

The path to the keystore file that contains the certificate to use for encrypting web socket communication between the agent and the InformixHQ server. This property must be set if **ssl.enable** is set to true.

ssl.keystore.password

The password to unlock the keystore file for used for encrypting web socket communication between the agent and the InformixHQ server.

This property must be set if **ssl.enable** is set to true.

target.informixdir

Optionally, specify the directory on the local machine where Informix is installed. If left empty, the agent will query the server for the INFORMIXDIR property. This property is used by sensors that gather data from onstat or other Informix utilities

target.onlinelog

Optionally specify the path to the **online.log** file for the target Informix database server.

If this is left empty and the Online Log monitoring sensor is enabled for this server, the agent will lookup the **online.log** file path by querying the database server.

There is no default value.

target.ping.frequency

Specifies the interval, in seconds, between pings to the target database server to see if it is still online. When the agent is running, it will regularly monitor whether the target database server is online or offline. This property controls the duration between these checks.

The default value is 1, indicating to check the server status every second.

user.password.minLength

Controls the minimum length for a user password. The default value is 8.

user.password.requireLowerCase

Controls whether user passwords are required to include at least one lowercase character. The default value is true.

user.password.requireUpperCase

Controls whether user passwords are required to include at least one uppercase character. The default value is true.

user.password.requireNumber

Controls whether user passwords are required to include at least one number. The default value is true.

user.password.requireSpecialCharacterFromSet

Controls whether user passwords are required to include at least one special character. An empty string indicates that no special characters are required. Setting this value to “!@#$%^&*()” would require user passwords to include at least one of those characters. The default value is an empty string.

InformixHQ View Agents and Shutdown

About this task

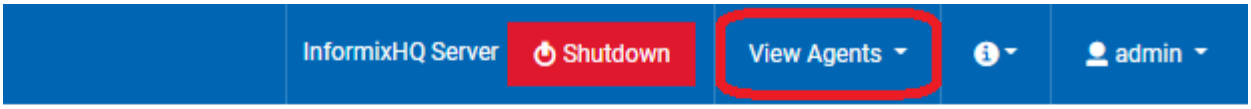
View Agents feature in InformixHQ displays the list of all the Agents' information and their status at one place, depending on user privilege to specific instance/server added in InformixHQ. This feature helps users to see consolidated view of all the agents.

Note:

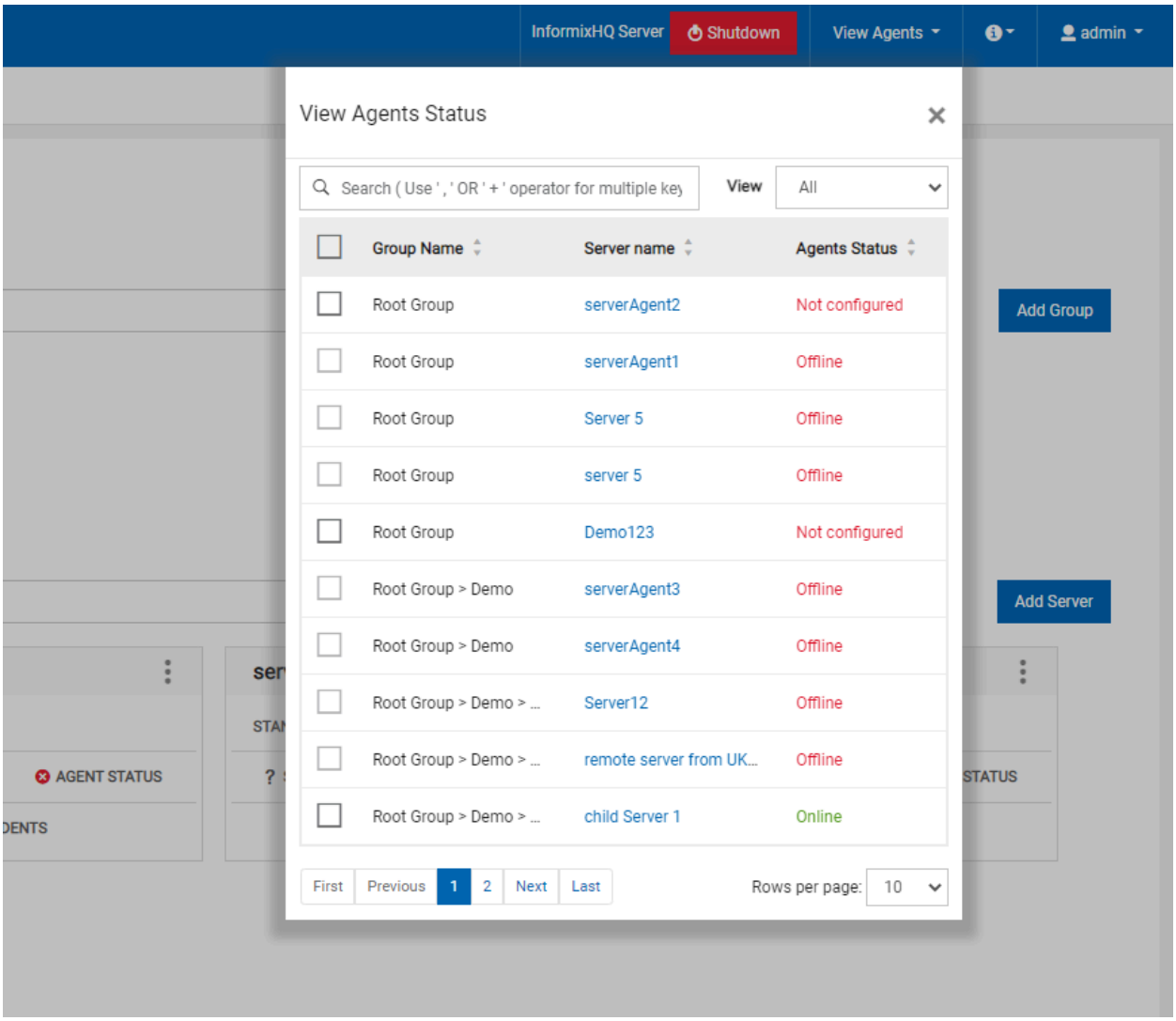
- If users have admin access to Informix server in InformixHQ, those users can initiate shutdown agent from this list.
- If users have only read access to Informix server in InformixHQ, those users can only view agent status in this list.
- If users don’t have any privilege to specific instances, Agent status for such instances won’t be visible to those users.

Use the following steps to shut down an InformixHQ agent:

1. **View Agents** drop-down option is available in InformixHQ top right corner.

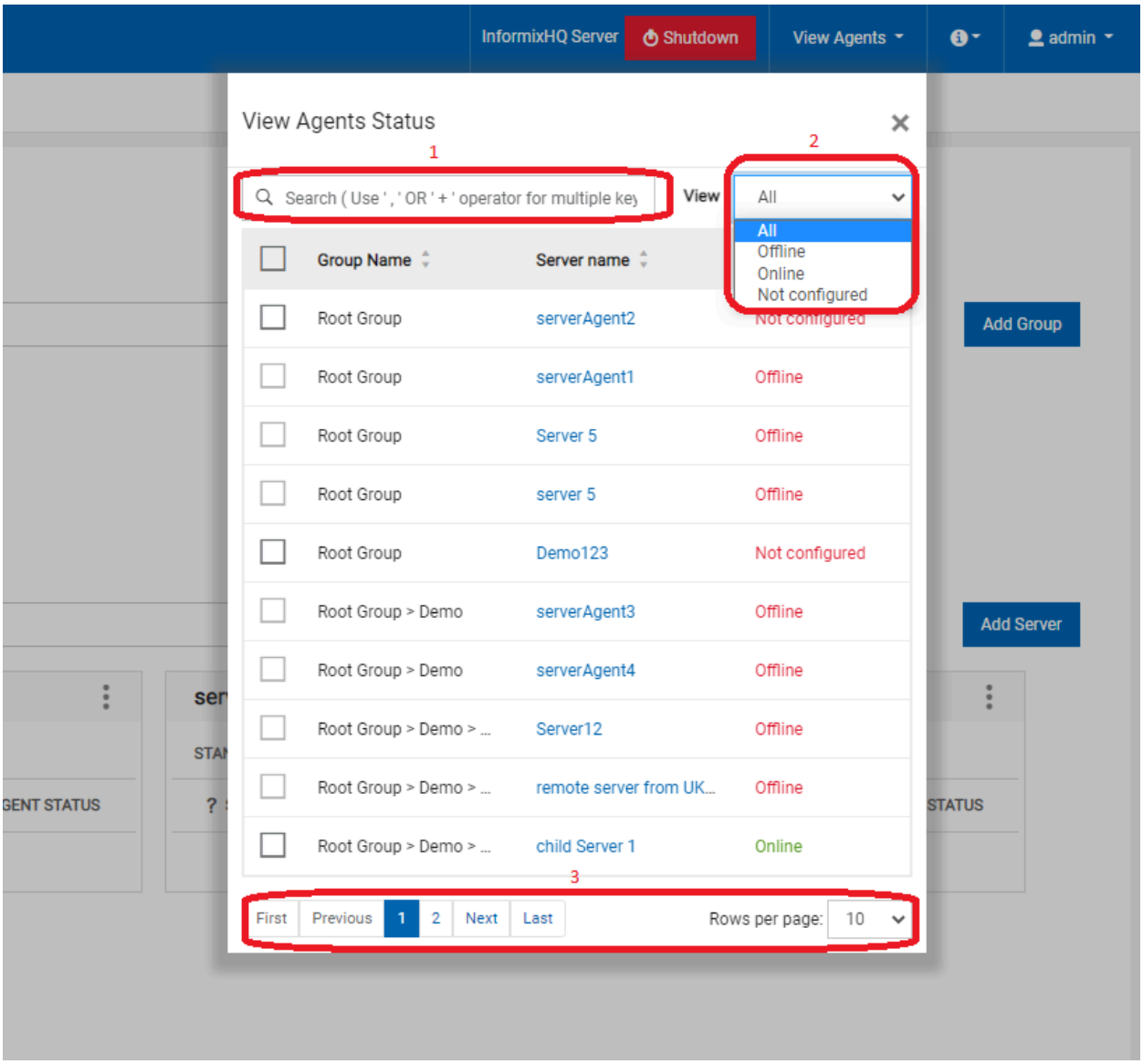


2. Clicking on **View Agents** will display the list of Agents with their respective status.

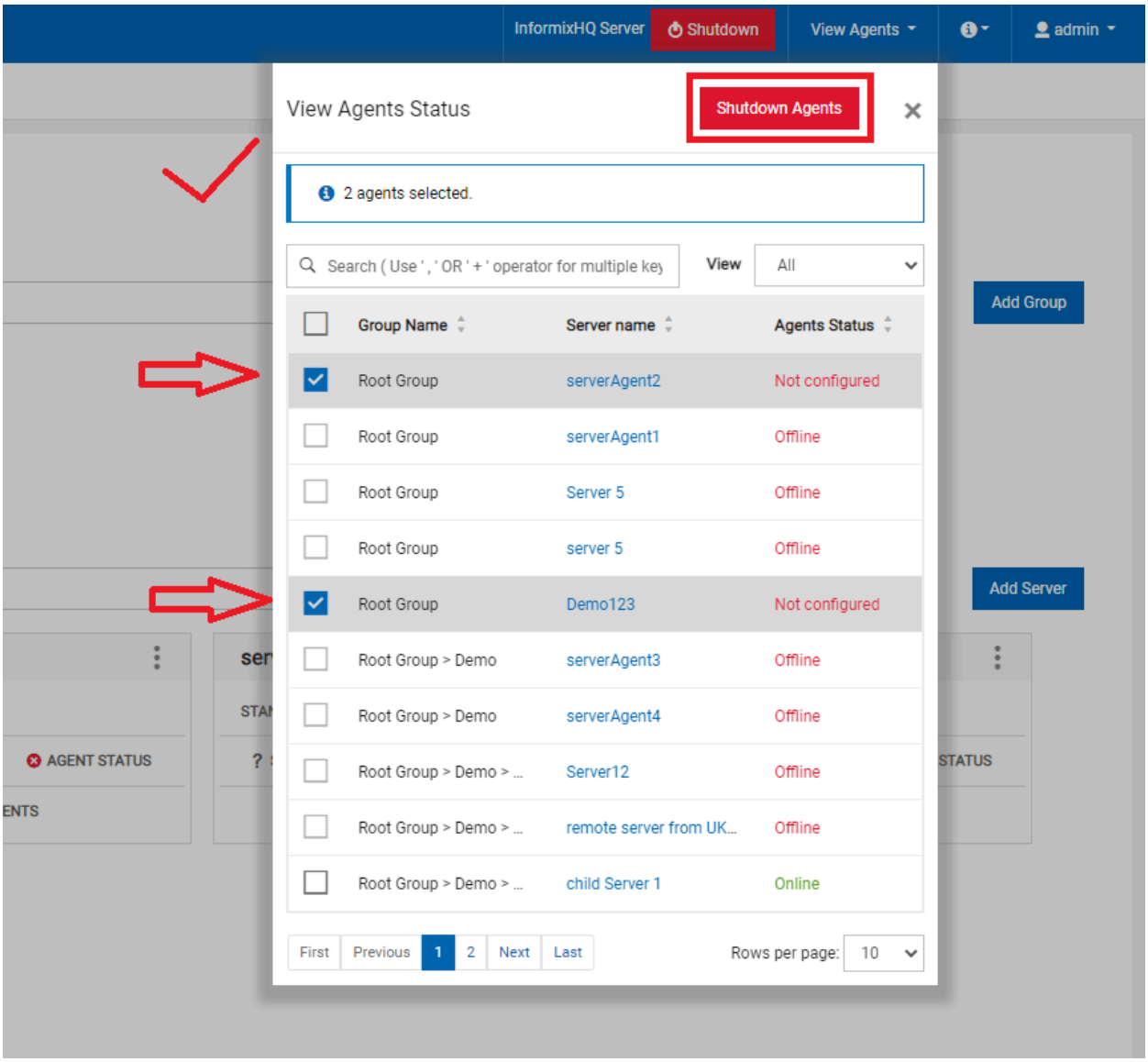


3. **View Agents** also displays the information about filters, pagination, and sorting columns availability.

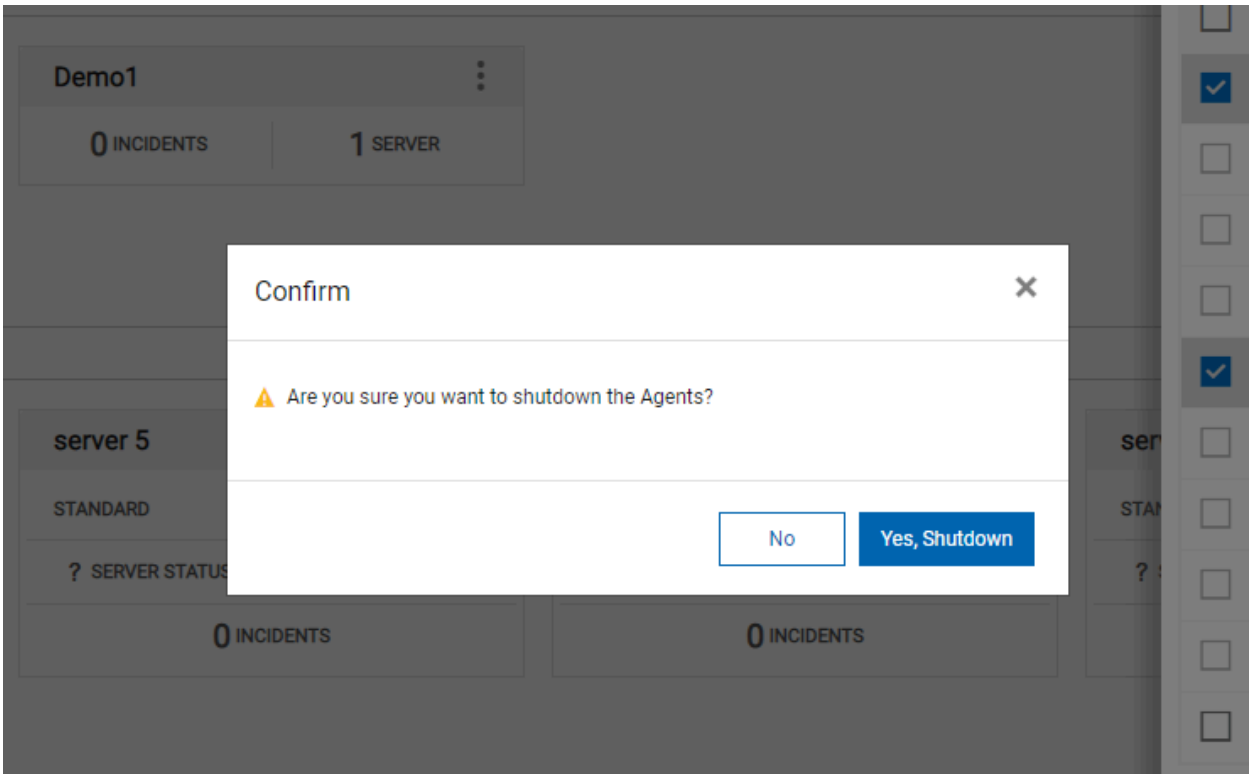
    - **Searching:** search for the data available using mentioned operators/key words.
    - **View Selection:** Select the drop-down values to filter out the data.
    - **Pagination and Page limit:** Move from one page to another page and count of records availability per each page with page limit.



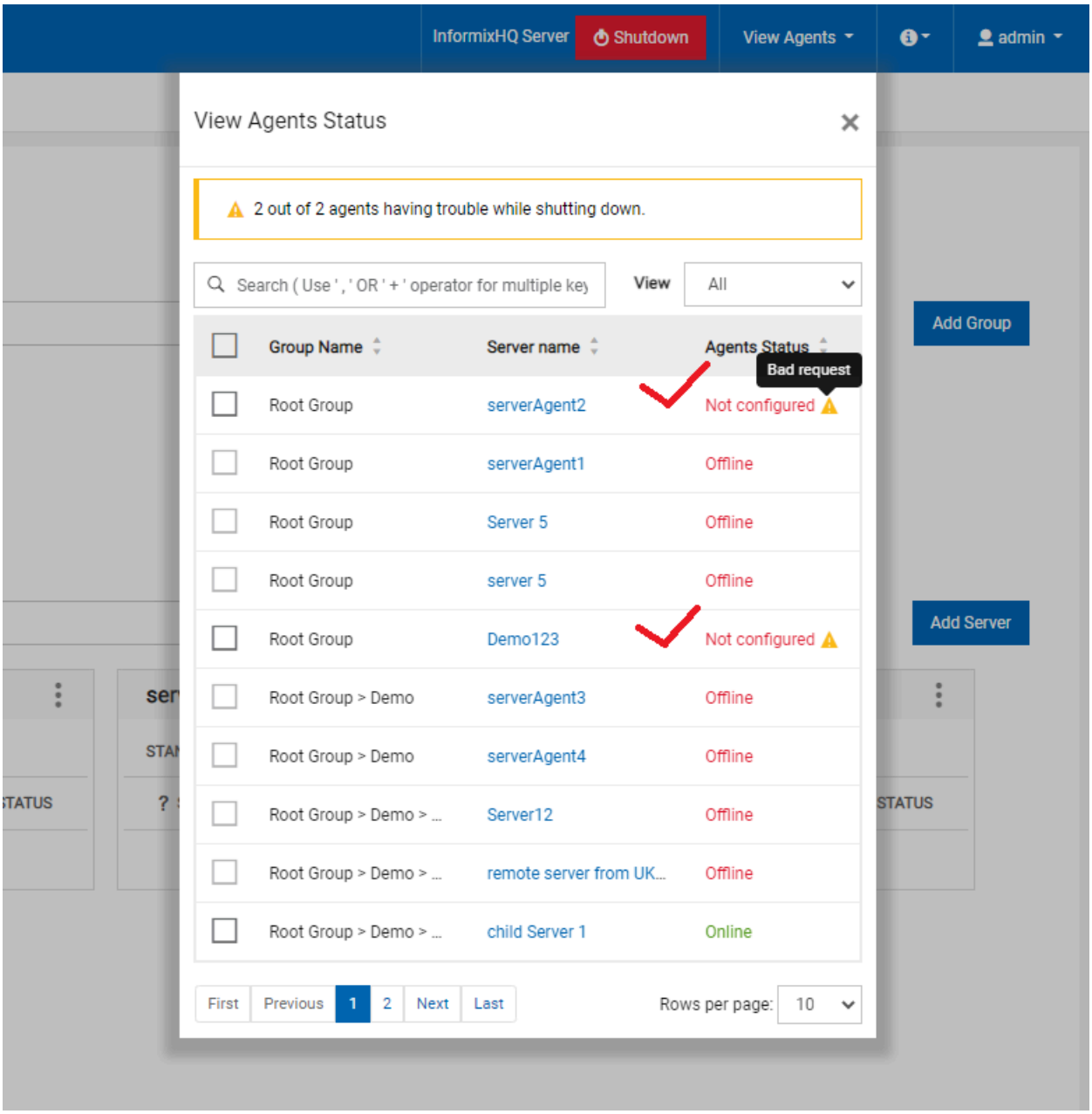
4. Once any agent record is selected, **Shutdown Agent** button will be displayed (applicable only for users with admin access to that instance) and the count of selected records is also displayed as an information message.



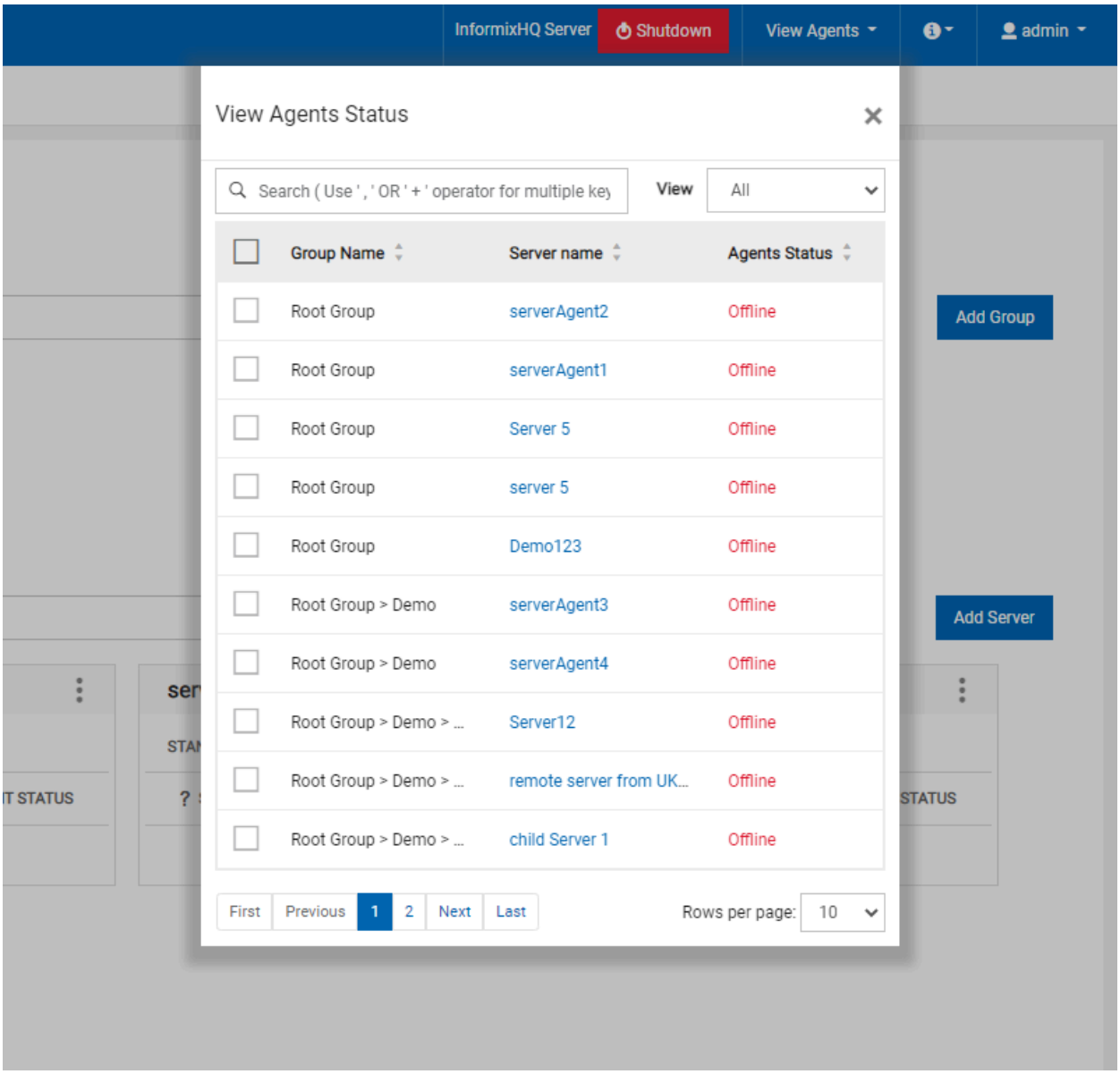
5. On clicking **Shutdown Agents** button, confirmation pop-up will be displayed. **Yes, shutdown** will shut down the InformixHQ agent and **No** will cancel the InformixHQ Agent shutdown request and it will be remain on the same page.



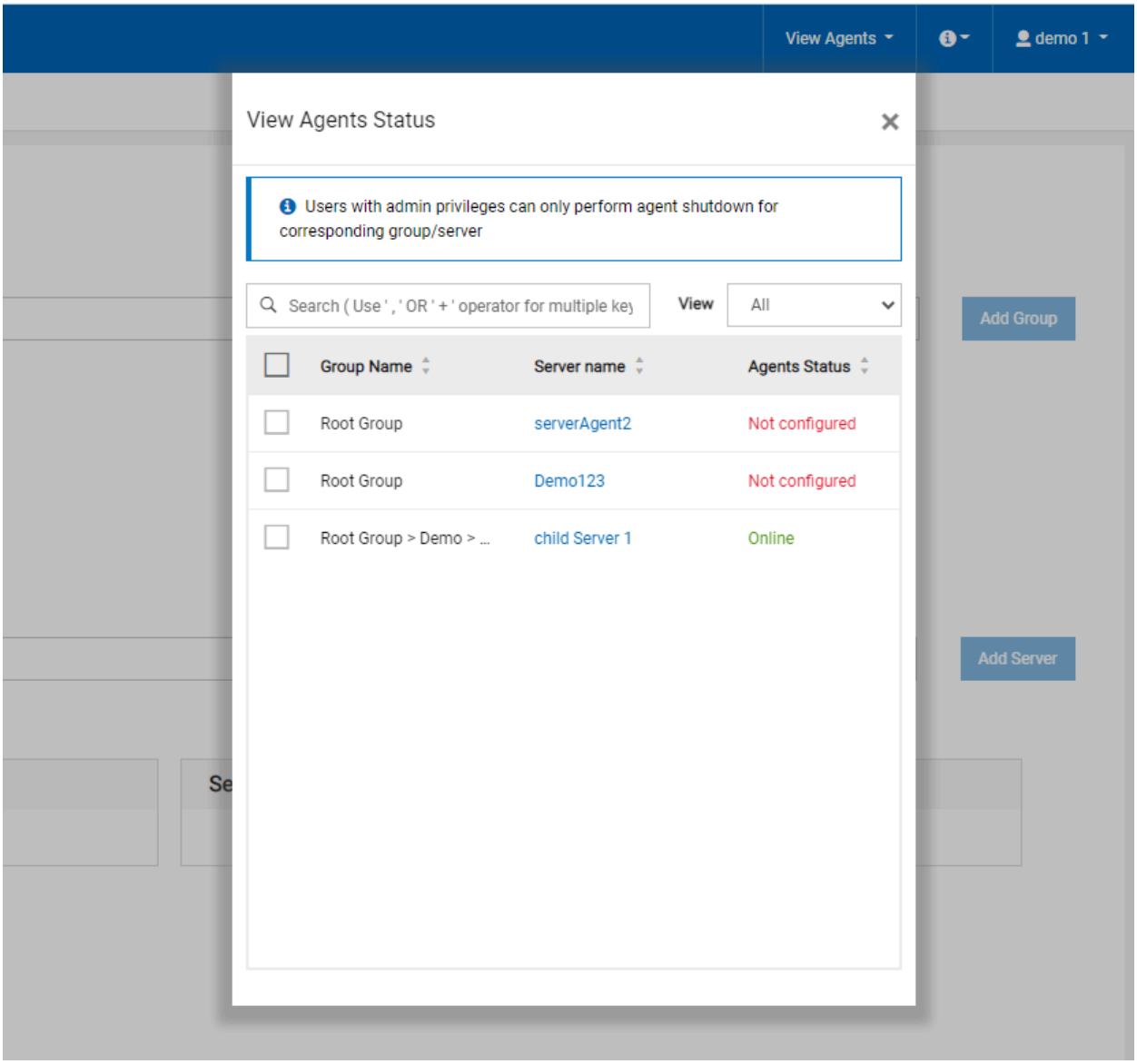
6. If any error occurs while shutting down Agent, **View Agents Status** will show warning messages with warning icon.



7. If requested agents shut down successfully, then those agents will be available in offline mode.


InformixHQ Server
Shutdown
View Agents
admin
View Agents Status
Search (Use ' , ' OR ' + ' operator for multiple key
View
All
Group Name
Server name
Agents Status
Root Group
serverAgent2
Offline
Root Group
serverAgent1
Offline
Root Group
Server 5
Offline
Root Group
server 5
Offline
Root Group
Demo123
Offline
Root Group > Demo
serverAgent3
Offline
Root Group > Demo
serverAgent4
Offline
Root Group > Demo > ...
Server12
Offline
Root Group > Demo > ...
remote server from UK...
Offline
Root Group > Demo > ...
child Server 1
Offline
First
Previous
1
2
Next
Last
Rows per page:
10
Add Group
Add Server

8. If a user has only read access, that user will only view the agents but will not be able to select agent to shutdown from **View Agents**.



H2 Database

This topic provides a brief tutorial on what is H2 Database and how you can migrate your old version of h2db data to new version of h2db in InformixHQ.

What is H2 Database?

H2 Database is a light weight java database. InformixHQ uses H2 Database as an embedded database where it stores all the details related to InformixHQ users, informix servers' connections, connection properties, repository database name, alerting configurations, etc.

Where can you find h2db.mv.db in InformixHQ?

h2db is located in the directory from where you are running informixhq-server.jar. If you are running informixhq-server.jar for the first time, InformixHQ will create the h2db.mv.db file.

Default location of informixhq-server.jar and h2db.mv.db is $INFROMIXDIR/hq .

H2 Database Migration

Why is h2db migration required?

InformixHQ 2.1.0 has upgraded its H2 Database version due to the following security vulnerabilities present in the older versions of H2 Database:

- CVE-2021-42392
- CVE-2021-23463
- CVE-2022-23221

Between H2 Database version 1.4.192 and the latest H2 Database version 2.1.214, there have been considerable changes and direct upgrade of H2 Database is not possible. For existing users, H2 Database (InformixHQ 1.6.3 or lower) is created with version 1.4.192, hence data needs to be migrated to 2.1.214 manually. Two options for this manual migration are explained in this tutorial.

For new users who are using InformixHQ for the first time (version 2.1.0 or higher), there is no impact, and no upgrade is needed as InformixHQ will create new H2 database with the latest version.

What is the impact if you have not migrated to the new version of H2 Database?

If users using InformixHQ 1.6.3 or lower try to move to InformixHQ 2.1.0 or above, they will not be able to start InformixHQ until the data is migrated to the newer version of H2 Database. If users try to run InformixHQ 2.1.0 or above without this migration done, they will get the following error in the InformixHQ log file:

```
"org.h2.mvstore.MVStoreException: The write format 1 is smaller than the supported format 2 [2.1.214/5]"
```

Users will also get a console message providing steps of migration along with a documentation link for reference.

For new users who are using InformixHQ for the first time (version 2.1.0 or higher), there is no impact and no upgrade is needed as InformixHQ will create new H2db with the latest version.

Note:

1. If existing users start InformixHQ (2.1.0 or higher) using java -jar command as shown below, InformixHQ will not be started and the information given below is shown to the users on console.

```
java -jar informixhq-server.jar informixhq-server.properties
Unable to start InformixHQ server as H2db version is older and needs upgrade. Please run below
 command to upgrade H2db to newer version.
```

```
Please note:

1) Path of the old InformixHQ server jar (version 1.6.3 or lower) should be known, and should be
 supplied where ${OLDHQDIR} is referenced in
 the command below.

2) If ${OLDHQDIR} jar path is not available, then this upgrade needs to be completed using a few
 manual steps. For details of this migration,
 refer Manual steps on page 141.

3) Optional second parameter -> Path of the properties file. If any H2db configuration properties
 are set previously, then this should be
supplied where ${PROPERTIES_DIR} is referenced in the command below.

4) Current directory should contain old h2db.mv.db file.

5) ${NEWHQDIR} referenced below is the current InformixHQ (2.1.0 or higher) jar path. Default
 informixhq-server.jar path is {INFORMIXDIR}/hq/

Migration command:

1) java -cp ${NEWHQDIR}/informixhq-server.jar com.informix.hq.server.h2.upgrade.H2Upgrade
 ${OLDHQDIR}/informixhq-server.jar
OR
2) java -cp ${NEWHQDIR}/informixhq-server.jar com.informix.hq.server.h2.upgrade.H2Upgrade
 ${OLDHQDIR}/informixhq-server.jar
 ${PROPERTIES_DIR}/informixhq-server.properties
```

2. If users start InformixHQ using startup scripts, InformixHQ will not be started and there will be no information on the console. Users need to check the hq server logs for details.

H2 Database Migration Steps

There are two ways user can do h2db migration:

1. Old InformixHQ (1.6.3 or lower) where old jar file should be available:

In this case, user needs to simply run one command for upgrade and data migration:

```
java -cp {NEWHQDIR}/informixhq-server.jar com.informix.hq.server.h2.upgrade.H2Upgrade
 {OLDHQDIR}/informixhq-server.jar
OR
java -cp {NEWHQDIR}/informixhq-server.jar com.informix.hq.server.h2.upgrade.H2Upgrade
 {OLDHQDIR}/informixhq-server.jar
{PROPERTIES_DIR}/informixhq-server.properties
```

Note:

- Path of the old InformixHQ server jar(version 1.6.3 or lower) should be known, and should be supplied where {OLDHQJAR} is referenced in the command above.
- If {OLDHQJAR} jar path is not available, then this upgrade needs to be completed using few manual steps. Please refer section 2 on page 141 below.

- Optional second parameter -> Path of the properties file. If any H2 Database properties (encrypt.enable, encrypt.algorithm, encrypt.password) is set previously, then this should be supplied where {PROPERTIES_DIR} is referenced in the command above.
- Current directory should contain old H2 Database file (h2db.mv.db).
- {NEWHQJAR} referenced above is current InformixHQ (2.1.0 or higher) jar path. Default InformixHQ jar path is {INFORMIXDIR}\hq\
- Ensure old h2db.mv.db has correct file permission for InformixHQ jar to access the same.
- After running above command with required parameters successful output will be:

```
Backup of old h2db file is completed successfully, now old h2db file name is h2db_old.mv.db.
Alter process executed successfully.
Data export process executed successfully.
Data import process executed successfully.
Clean-up for import and export process completed successfully.
```

2. If Old InformixHQ (1.6.3 or lower) jar path NOT available:

In this case, manual steps given below need to be followed for H2 Database Upgrade and migration:

- **Prerequisites for manual upgrade:**

  In order to migrate from H2 Database 1.4.192 to H2 Database 2.1.214, both versions of H2 Database are needed.

  Download the jars version 2.1.214 (h2-2.1.214.jar) and version 1.4.192(h2-1.4.192.jar) from H2 site or from maven repo https://mvnrepository.com/artifact/com.h2database/h2

  Ensure that the downloaded H2 jar files have appropriate file permissions for migration process. If any of the commands given below fails, migration can be started again with backup taken in step 1 below.

Software	Required Version
h2-1.4.192.jar	1.4.192
h2-2.1.214.jar	2.1.214
Java	1.8

- **Procedure:**

  Follow these steps in order to migrate the H2 Database:

  1. **Backup existing database**

     Find the existing h2 database (h2db.mv.db).

     For InformixHQ, database can be found at <installation_Path>/explore/server/h2db.mv.db

     By default, InformixHQ installation path is $INFORMIXDIR/hq

H2 Database file name is h2db.mv.db.

Make a copy of h2db.mv.db to some other directory for backup before starting the migration.

2. **Create a SQL File migrate.sql**

   Create an SQL file migrate.sql with the following ALTER TABLE statements.

   Save the file in the same folder where h2db.mv.db file is present.

```
ALTER TABLE IF EXISTS users ALTER COLUMN ID SET NOT NULL;
ALTER TABLE IF EXISTS users ADD PRIMARY KEY (ID);
ALTER TABLE IF EXISTS informix_servers ALTER COLUMN ID SET NOT NULL;
ALTER TABLE IF EXISTS informix_servers ADD PRIMARY KEY (ID);
ALTER TABLE IF EXISTS informix_server_groups ALTER COLUMN ID SET NOT NULL;
ALTER TABLE IF EXISTS informix_server_groups ADD PRIMARY KEY (ID);
ALTER TABLE IF EXISTS alerting_incidents ALTER COLUMN ID SET NOT NULL;
ALTER TABLE IF EXISTS alerting_incidents ADD PRIMARY KEY (ID);
```

3. **Run the Script migrate.sql and Export data in Zip file**

   Open a command line window in the path where H2 db file is present and run the java command. Replace <h2_path> with the path where h2-1.4.192.jar is present.

   - If h2db encryption is **NOT** enabled in InformixHQ properties file:

```
java -cp <h2_path>/h2-1.4.192.jar org.h2.tools.RunScript -url jdbc:h2:./h2db -script
 migrate.sql
java -cp <h2_path>/h2-1.4.192.jar org.h2.tools.Script -url jdbc:h2:./h2db -script h2db.zip
 -options compression zip
```

   - If h2db encryption is **Enabled** in InformixHQ properties file:

     If **h2.encrypt.enable**, **h2.encyrpt.password, and h2.encrypt.algorithm** parameters are set in InformixHQ server properties file, then the commands will be modified as shown below.

     Example: If h2.encrypt.enable=true, h2.encyrpt.password=password123, and h2.encrypt.algorithm= AES, h2 database URLs should be modified as shown below:

     (Add a space after the password, since the format is <file_password> space <user_password>)

```
java -cp <h2_path>/h2-1.4.192.jar org.h2.tools.RunScript -url "jdbc:h2:./h2db;CIPHER=AES"
 -password "password123 "
-script migrate.sql
java -cp <h2_path>/h2-1.4.192.jar org.h2.tools.Script -url "jdbc:h2:./h2db;CIPHER=AES"
 -password "password123 " -script h2db.zip
 -options compression zip
```

4. **Rename altered H2db**

   Rename the existing h2db.mv.db database.

```
mv h2db.mv.db h2db_old.mv.db
```

5. **Create new H2 Database (2.1.214) and restore data**

Run following java command to create new H2 Database file (h2db.mv.db) and to restore data from zip file created in step 3. Replace <h2_path> with the path where h2-2.1.214.jar is present.

- If h2db encryption is **NOT** enabled in InformixHQ properties file:

```
java -cp <h2_path>/h2-2.1.214.jar org.h2.tools.RunScript -url jdbc:h2:./h2db -script
 h2db.zip -options compression zip FROM_1X
```

- If h2db encryption is **Enabled** in InformixHQ properties file:

  If **h2.encrypt.enable**, **h2.encyrpt.password, and h2.encrypt.algorithm** parameters are set in InformixHQ server properties file, then the commands will be modified as shown below.

  Example : If h2.encrypt.enable=true, h2.encyrpt.password=password123, and h2.encrypt.algorithm= AES, h2 database URLs should be modified as shown below:

  (Add a space after the password, since the format is <file_password> space <user_password>)

```
java -cp <h2_path>/h2-2.1.214.jar org.h2.tools.RunScript -url "jdbc:h2:./h2db;CIPHER=AES"
 -password "password123 " -script h2db.zip
-options compression zip FROM_1X
```

Once all the above steps are completed without any error, data is successfully migrated to h2db.mv.db.

6. **Cleanup Migration files (Optional)**

   At this point h2db.zip , h2db_old.mv.db, and migrate.sql files can be deleted.

Frequently asked questions (FAQs) about InformixHQ

These topics provide short answers to some frequently asked questions about InformixHQ.

High level architecture and functionality

This topic provides answers to some frequently asked questions about high level architecture and functionality.

What is the difference between the InformixHQ server and agent?

The InformixHQ server is a Java 11 based Jetty web server that hosts both the web user interface (UI) portion of the tool and the REST web services. The InformixHQ server also connects directly to the Informix database server instances to gather live data and run administration commands, manages connections to all agents, and evaluates and dispatches alerts when new monitored data comes in, among other things. You only need to run a single instance of the InformixHQ server to manage and monitor all of your Informix database server instances.

The InformixHQ agent is a lightweight Java program that runs alongside each of your Informix database server instances and gathers monitoring data. The agent is intended to be installed on the same host machine as the Informix database server that you want monitored by InformixHQ, which allows it to also gather operating system statistics about the host machine. Unlike the InformixHQ server, you will have one instance of the InformixHQ agent running for each Informix database server that you want the tool to monitor.

For information, see InformixHQ Architecture on page 4.

Is it necessary to start the agent to use InformixHQ?

The agent is not required to use InformixHQ. However, it is important to note that the agent process is the one responsible for gathering all of the monitoring data. Therefore, if you choose not to connect the agent, the tool will not be monitoring your Informix database servers when you close your web browser. You will not be able to see graphs in the UI of how various performance metrics are trending over time and you will not be able to configure alerting conditions if you are not using the agent.

I have multiple Informix database instances running on the same host machine. Do I need one agent per host or one agent per database server?

There is a one-to-one relationship between the InformixHQ agent and the Informix database server. Each agent monitors just one Informix database server instance. If you have multiple Informix database server instances on the same host and you want to monitor each of them, then you will need to have multiple agents on that same host machine, one for each database server instance.

Getting Started

This topic provides answers to some frequently asked questions on getting started with InformixHQ.

Where can I get InformixHQ?

InformixHQ is available as part of the Informix database server installation for versions 12.10.xC13 or higher.

For information, see Getting Started on page 7.

What should I do before upgrading to the latest version of InformixHQ?

Before upgrading to the latest version, it is suggested to make a backup copy of the old database file i.e. a backup copy of h2db.mv.db file. This is required in case if you ever want to revert to the previous version of InformixHQ with previous InformixHQ jar.

Where can I find sample configuration files for the server and agent?

Sample configuration files for both the InformixHQ server and agent are available in the **$INFORMIXDIR/hq** directory of your Informix database server installation.

For information, see InformixHQ Server Configuration on page 43 and InformixHQ Agent Configuration on page 128.

How can I configure the logging for the server or agent? How do I change the logging level?

InformixHQ uses the log4j2 library for logging. By default, the InformixHQ server and agent will log messages at INFO level to an monitoring-server.log file and an monitoring-agent.log file respectively.

You can customize the logging behavior by providing a monitoring-server.log4j.xml file in the current directory or classpath when starting the InformixHQ server or an monitoring-agent.log4j.xml file in the current directory or classpath when starting the InformixHQ agent. Use these Log4j configuration files to change the logging level (ERROR, WARN, INFO, or DEBUG), change the log file location, or enable rolling window logging. For more information, see log4j2 and Logging in InformixHQ on page 32.

How do you stop the server and how do you stop the agent?

You can stop both the InformixHQ server and agent by terminating the java process that is running them.

Monitoring and the Repository Database

This topic provides answers to some frequently asked questions on monitoring and the repository database in InformixHQ.

Where is the monitored data stored?

The monitored data is stored in the repository database as defined in the InformixHQ UI. The repository database must be an Informix database, but it can be located wherever you choose. You can store the monitored data within the same Informix instance that is being monitored or you can choose to use a central repository database to store all of the monitored data for all of your Informix database server instances.

Does the repository database need to exist ahead of time? Do I need to run any DDL statements to initialize the repository database?

The repository database must exist before you define it as a repository database. You define a repository database in the InformixHQ UI on a server's **Setup > Agent** page.

You do not need to run any DDL statements to initialize the schema for the repository database. The agent will automatically create the tables it needs to store the monitored data.

If I add a new database server instance to a group and start an agent for that new server, do I have to do anything to enable all of the group's sensors on the new server?

No, this happens automatically. Sensors defined in a group's monitoring profile are automatically applied to all database servers within the group. When a new server is added to the group, it will automatically inherit all of the sensors enabled in the group's monitoring profile. No additional step is needed to make this happen. The same thing applies to a group's alerting profile.

How is the row size and table size for sensor data calculated in InformixHQ?

Sensor data often varies in structure and content based on type of sensor. Calculating the size of a row involves understanding the typical columns and their potential data types. For details, refer Sensor Table Size Calculation on page 64.

Security

This topic provides answers to some frequently asked questions on InformixHQ security.

How can I configure HTTPS and/or SSL for InformixHQ?

To use the Secure Sockets Layer (SSL) protocol to encrypt communication with InformixHQ, you will need a keystore and certificate. You can use the method that best fits your environment for creating the keystore and certificate, for example Java keytool, OpenSSL, or even the IBM Global Security Kit (GSKit).

- **Configuring HTTPS in the InformixHQ server**

  Once you have the keystore, secure the InformixHQ web user interface and REST API by configuring HTTPS in the InformixHQ server. To configure HTTPS in the InformixHQ server, in your InformixHQ server properties file, set the httpsPort on page 44, ssl.keystore.file on page 45, and ssl.keystore.password on page 46 properties and potentially also the ssl.key.password on page 46 property if your key password is different from the keystore password.

  Additionally, if you want to disable HTTP access to the InformixHQ so that all communication to and from the InformixHQ server uses HTTPS, set the httpPort on page 44 to -1 in your properties file. If instead you would like

the InformixHQ server to automatically redirect all HTTP traffic to the HTTPS port, set the redirectHTTPtoHTTPS on page 45 property to true.

Figure 7. Sample InformixHQ server properties file with HTTPS enabled

```
# configure ports
httpPort=-1
httpsPort=8088
redirectHTTPtoHTTPS=true


#--------------------------------------------------------------------------
# SSL keystore configuration
#--------------------------------------------------------------------------

ssl.keystore.file=/opt/informixhq/mykeystore.jks
ssl.keystore.password=myStorePassword

## Key:          ssl.key.password
## Type:         string
## Default:
## Description: The password to unlock the entry into the keystore. The default
##    value is no password, which means to use the keystore password. If the
##    entry into the keystore requires a password that is different from the
##    keystore password, set this parameter to the entry password.
#
#ssl.key.password=
```

- Once you have HTTPS enabled in the InformixHQ server, you must configure your InformixHQ agents to encrypt their web socket communication with the InformixHQ server. If you use the **Deploy Agent** button in the UI to have the InformixHQ server automatically deploy the agent, it will automatically configure the agent to use SSL if the InformixHQ server has HTTPS enabled.

  If you are starting your agents manually to enable SSL, set the ssl.enable on page 129 property to true in your agent configuration file and then set the ssl.keystore.file on page 130 property, the ssl.keystore.password on page 130 property.

Figure 8. Sample InformixHQ server properties file with SSL enabled

```
# host and port of the InformixHQ server
server.host=localhost
server.port=8088

# The id of the Informix database server as defined in InformixHQ
serverInstance.id=1

# SSL configuration
ssl.enable=true
ssl.keystore.file=/opt/informixhq/mykeystore.jks
ssl.keystore.password=myStorePassword
```

How can I encrpyt the internal H2 database that the InformixHQ server uses?

The InformixHQ server creates an H2 database to store its internal metadata. The H2 database file, **h2db.mv.db** will be created in the directory where you start the InformixHQ server. It will store information about the groups and servers you define in the tool (including the database server connection credentials), the monitoring and alerting profiles, and alerting incidents.

You can configure encryption for this H2 database file by setting the following properties in your InformixHQ server configuration file.

```
h2.encrypt.enable=true
h2.encyrpt.password=some_password
```

Optionally, you can also set the h2.encrypt.algorithm property if you want to set the encryption algorithm to something other than AES.

Note: If you want to encrypt the H2 database, you must set these properties the first time you start the InformixHQ server when the H2 database is first created and initialized. You cannot change your H2 encryption configuration after the H2 database has been created. If you want to encrypt an H2 database that has already been created, you can use H2's ChangeFileEncryption tool as described in http://www.h2database.com/html/features.html#file_encryption or you can delete your h2db.mv.db file and have the InformixHQ server recreate it from scratch the next time you start it.

How can I configure InformixHQ to use SSL when connecting to my database server?

If your database supports or requires SSL connections, you can setup SSL using the connection properties on the **Add Server** page when adding the server or on the server's **Setup** page after it is created.

You must add the following connection properties in order to use SSL on InformixHQ's JDBC connections to your database server:

```
SSLCONNECTION=true
SSL_TRUSTSTORE=/path/to/truststore
SSL_TRUSTSTORE_PASSWORD=password
```

The truststore/keystore file that you specify must be present both where InformixHQ server is running as well as the machine where the InformixHQ agent is running.

For more information, see Adding Severs and Groups on page 47.

Users and Permissions

This topic provides answers to some frequently asked questions on InformixHQ users and permissions.

What's the difference between monitoring/admin credentials and Read/SQL/Admin permissions?

The monitoring and admin credentials are used by the InformixHQ server itself as part of the JDBC connection whenever it needs to connect to the database server. The Read, SQL, and Admin permissions are those that are assigned to users in the tool and therefore control what users can see and do in the UI.

- **Monitoring and admin credentials**

  When you click **Add Server** from a group dashboard page to add a new Informix database server to InformixHQ, you are asked to provide the InformixHQ tool with not only the host and port, but also user and password information that will be used when establishing a JDBC connection to that database server instance.

  The monitoring credentials are used by the InformixHQ server whenever it needs to query for data to be displayed in the UI or by the InformixHQ agent data it monitors your database server instance.

  The admin credentials are used whenever a user in the UI requests that an administration action be performed on the database server, for example creating a dbspace, editing an onconfig parameter, or deploying an agent. The user provided for the admin credentials should be a DBSA and needs access to the sysadmin database and permissions to run SQL Admin API commands on the database server.

  The required privileges for the monitoring and admin users can also be found in Adding Severs and Groups.

  It is required that you provide monitoring credentials when adding your Informix database server information to InformixHQ. Admin credentials need only be provided if you want to use InformixHQ to run administrative actions on your database server or if you want that server to be used as a repository database.

- **Read, SQL, and Admin permissions**

  The Read, SQL, and Admin permissions are the permissions assigned to the users that are created in InformixHQ itself. These permissions determine what kinds of access that a user has on various Informix servers and groups in the UI and also which REST services that user is authorized to run. Therefore, these permissions control what each user can see or do in the tool.

  Read permission provides the ability to view information about a server. SQL permission provides the ability to run any SQL query or statement (including DML) against the database server on its Schema Manager page in the UI. Admin permission provides the ability run administrative actions on the server, for example creating a dbspace or editing and onconfig parameter. Read, SQL, and Admin permissions are mutually exclusive, so be sure to grant each user the full set of permissions that they will need.

  Read, SQL, and Admin permissions can be granted to InformixHQ users by a System Administrator user in InformixHQ on the **System Settings > User Management** page or on the **Permissions** page for any server or group.

What privileges are required on the Informix database server?

The required privileges for the monitoring and admin users can be found in Adding Severs and Groups.

Is there any relationship between the users I create in InformixHQ and OS users?

There is no relationship between the users you create in InformixHQ and the operating system users on the host machine it is running on. Users that are created in InformixHQ are users that are specific to the InformixHQ tool. They have no relationship to operating system users or even Informix database server users.

Index