

Moab Insight

Administrator Guide 9.0.3

January 2017



© 2017 Adaptive Computing Enterprises, Inc. All rights reserved.

Distribution of this document for commercial purposes in either hard or soft copy form is strictly prohibited without prior written consent from Adaptive Computing Enterprises, Inc.

Adaptive Computing, Cluster Resources, Moab, Moab Workload Manager, Moab Viewpoint, Moab Cluster Manager, Moab Cluster Suite, Moab Grid Scheduler, Moab Grid Suite, Moab Access Portal, and other Adaptive Computing products are either registered trademarks or trademarks of Adaptive Computing Enterprises, Inc. The Adaptive Computing logo and the Cluster Resources logo are trademarks of Adaptive Computing Enterprises, Inc. All other company and product names may be trademarks of their respective companies.

Adaptive Computing Enterprises, Inc.

1712 S. East Bay Blvd., Suite 300

Provo, UT 84606

+1 (801) 717-3700

www.adaptivecomputing.com



Scan to open online help

Welcome	1
Chapter 1 Moab Insight Overview	2
Chapter 2 Insight Conceptual Information	4
The Insight Archiver Sample Rate	4
The Insight Pruning Policy	4
Configuration	5
Chapter 3 Customizing Insight	9
Writing A Report	9
Adding A New Table, View, Or Index To The Schema	10
Tuning Insight For Your System	10
Tuning The Insight Archiver Sample Rate	12
Tuning Your PostgreSQL Server	13
Changing The Pruning Policy	13
Configuring Reliable Message Delivery	17
Troubleshooting	17

Welcome

Welcome

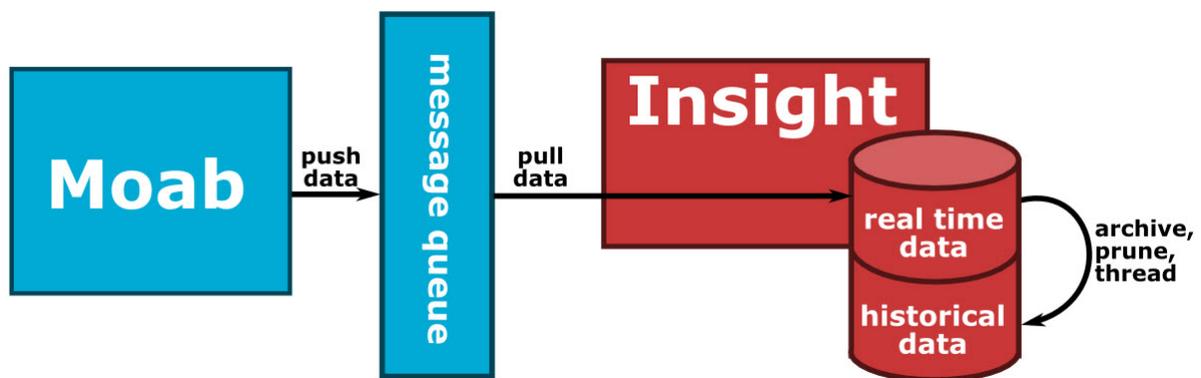
Welcome to *Moab Insight 9.0.3 Administrator Guide*.

This guide is intended as a reference for system administrators.

Chapter 1 Moab Insight Overview

Insight is an optional component of the Moab HPC Suite, Enterprise or Basic Edition, that lets you create, analyze, and report dashboards that represent the current and historical state of your cluster. It collects the data that Moab emits on its message queue. The message queue is efficient and reliable and gracefully tolerates disconnections or restarts on either side.

You should install Insight if you want to do either of the following: 1) Use Moab Viewpoint 9.0.3 or 2) Run reports and analyze events within the cluster using standard relational database tools such as Crystal Reports.



Moab produces a large, coherent snapshot of its object model at the end of each scheduling iteration and event-based updates as they occur; for instance, whenever a job starts or finishes. Insight constantly collects this data and writes it to the database. In the background, it runs an "archiver." Every so often, the archiver takes a sample of the current state of the cluster and copies it to historical tables. Insight periodically prunes the database to eliminate data that has become stale or expired. When Moab and Insight experience periods of disconnection for any reason, Moab uses a buffer to temporarily save that data. You can configure the sampling, pruning, and the buffer to suit the needs of your environment.

Associated tasks

The following sections describe how to configure, customize, and use Insight.

- [Installing Moab Insight](#)
- [Writing a Report](#)
- [Adding a New Table, View, or Index to the Schema](#)
- [Tuning the Insight Archiver Sample Rate](#)
- [Changing the Pruning Policy](#)
- [Configuring Reliable Message Delivery](#)

References

The following sections contain detailed information about the use of Insight.

- [The Insight Archiver Sample Rate](#)
- [The Insight Pruning Policy](#)
- [Troubleshooting](#)

Chapter 2 Insight Conceptual Information

This chapter provides conceptual information about Insight's features and functions.

In this chapter:

- [The Insight Archiver Sample Rate](#)
- [The Insight Pruning Policy](#)
- [Configuration](#)

The Insight Archiver Sample Rate

The sample rate is the rate at which Insight archives snapshots of statistical, usage, and trend data in the cluster. You can make the sample rate as granular as once per minute or as relaxed as once per hour. To maximize the usefulness of samples in statistical analysis and dashboards, ensure that your sample rate divides evenly into 10- and 60-minute intervals. Valid values include 2 minutes, 5 minutes, 10 minutes, 20 minutes, and 30 minutes, but values such as 3 minutes and 15 minutes are invalid.

Usually when you configure a more granular sample rate, Insight streams more data to the database and the reports and dashboards contain greater detail. However, if Moab has a long scheduling iteration, sampling may provide little benefit. This is because Moab emits a great deal of the data only once per cycle.

As you create samples, the database grows larger and running queries places a greater burden on the RAM and processors of the Insight machine. Note, however, that because Insight has a retention and pruning policy, it does not retain all samples forever. For more information, see [Changing the Pruning Policy](#).

Related Topics

[Tuning the Insight Archiver Sample Rate](#)

[The Insight Pruning Policy](#)

The Insight Pruning Policy

A pruning or retention policy tells Insight how much data to retain. It is stored as a structure in the Insight configuration file that you define in terms of units at varying granularity. For example, a simple pruning policy could dictate that

Insight should do the following: retain minute-by-minute samples of the cluster for one week and consider data older than that expired and prune it to avoid overwhelming the capacity of the database.

The pruning policy is the set of rules applied to `job_samples`, `node_samples`, and `reservation_samples` tables in the database. The pruner uses these rules to identify which historical records can be removed and which ones should be retained. The aim of the pruner is to retain a coherent historical view at the cluster and, at the same time, take care of the DB size growth.

The pruner policy rules are written in terms of granularity (i.e. the sampling rate or how often the records can appear) and the time interval to which the given granularity is applied.

During the last hour, the samples are retained with one-per-minute granularity; during the last day, with one-per-ten-minutes granularity; during the last week, with one-per-hour granularity; during the last month, with one-per-six-hours granularity; and finally, indefinitely, with one-per-day granularity.

The default insight pruning policy is complex and says the following: keep minute-by-minute samples back to the beginning of the previous hour; keep 10-minute samples back to the beginning of the previous day; keep hourly samples back to the beginning of the previous week; keep 6-hour samples back to the beginning of the previous day; and keep daily samples forever.

Related Topics

[Changing the Pruning Policy](#)

[The Insight Archiver Sample Rate](#)

Configuration

These properties can be modified by setting the appropriate values in the `/opt/insight/etc/config.groovy` file.

[Configuration reference](#)

For all possible values that can be set, please see the Grails reference guide. For project specific settings (usually the only ones you'll need to change), you may set the following properties:

Property	Type	Default	Description
<code>archiver.jobs.expireAfterSeconds</code>	Integer	63244800	The number of seconds a completed job will remain in the Insight Mongo database before being pruned.

Property	Type	Default	Description
archiver.jobSamples.expireAfterSeconds	Integer	63244800	The number of seconds a job sample will remain in the Insight Mongo database before being pruned.
archiver.jobStateJournal.expireAfterSeconds	Integer	63244800	The number of seconds a job state journal record will remain in the Insight Mongo database before being pruned.
archiver.nodes.expireAfterSeconds	Integer	63244800	A node that has not been updated in this number of seconds will be pruned from the Insight Mongo database.
archiver.nodesAuxiliary.expireAfterSeconds	Integer	63244800	A node auxiliary record that has not been updated in this number of seconds will be pruned from the Insight Mongo database.
archiver.nodeSamples.expireAfterSeconds	Integer	63244800	The number of seconds a node sample will remain in the Insight Mongo database before being pruned.
archiver.reservations.expireAfterSeconds	Integer	63244800	A reservation that has not been completed in this number of seconds will be pruned from the Insight Mongo database.
archiver.systemSamples.expireAfterSeconds	Integer	63244800	The number of seconds a system sample will remain in the Insight Mongo database before being pruned.
health.stale.threshold.seconds	Integer	300	Insight tables are considered stale if they have not been updated within this many seconds.
insight.port	Integer	5568	Port on which Insight listens for control messages (start, stop, sync, health).

Property	Type	Default	Description
insight.skip.database.migration	Boolean	false	If true, Insight will skip migrating database data from PostgreSQL into MongoDB when connected to a database from an older version of Insight.
moab.dataPort	Integer	5574	Port on which Insight listens for incoming messages from Moab.
moab.mongo.databaseName	String	moab	The MongoDB database name to use for Moab.
moab.mongo.host	String	127.0.0.1	The MongoDB host to use for Moab.
moab.mongo.password	String	secret4	(Optional) The password to use when connecting to MongoDB for Moab.
moab.mongo.port	Integer	27017	The MongoDB port to use for Moab.
moab.mongo.username	String	insight_user	(Optional) The username to use when connecting to MongoDB for Moab.
mongo.databaseName	String	insight	The MongoDB database name to use for Insight.
mongo.host	String	127.0.0.1	The MongoDB host to use for Insight.
mongo.password	String	secret4	(Optional) The password to use when connecting to MongoDB for Insight.
mongo.port	Integer	27017	The MongoDB port to use for Insight.
mongo.username	String	insight_user	(Optional) The username to use when connecting to MongoDB for Insight.

Property	Type	Default	Description
moab.reliabilityPort	Integer	5575	Port used for reliable message delivery between Moab and Insight.
messageQueue.secretKey	String	-	Base64-encoded AES key used to encrypt messages sent from Moab to Insight.

Chapter 3 Customizing Insight

This chapter provides individual procedures when using Insight, including configuration and troubleshooting.

In this chapter:

- [Writing a Report](#)
- [Adding a New Table, View, or Index to the Schema](#)
- [Tuning Insight for Your System](#)
- [Tuning the Insight Archiver Sample Rate](#)
- [Tuning your PostgreSQL Server](#)
- [Changing the Pruning Policy](#)
- [Configuring Reliable Message Delivery](#)
- [Troubleshooting](#)

Writing a Report

To write a report

1. Choose a reporting tool such as Crystal Reports, Stonefield, Cognos, Jasper.
2. Define a JDBC connection with the Postgres database. For more information, see "[Connecting to the Database](#)" in the PostgreSQL documentation. It is recommended that you create a new read-only user for the PostgreSQL `moab_insight` database for all reporting tools.
3. Enter your query. For example:

```
SELECT job.state, count(*) AS count, "user".name AS user_name
FROM job INNER JOIN "user" ON job.user_id = "user".credential_id
GROUP BY job.state, "user".name;
```

i The above query is just an example and may not be applicable for your configuration.

Related Topics

[Adding a New Table, View, or Index to the Schema](#)

Adding a New Table, View, or Index to the Schema

To add a new table, view, or index to the schema

1. Open PostgreSQL and switch to the `moab_insight` database.

```
$ sudo -u postgres psql -d moab_insight
```

2. Run the `CREATE TABLE` or `CREATE VIEW` command, ensuring that your table or view name begins with `x_`. This convention tells Insight that your table or view is an extension that it should protect on upgrade and that it should not directly manage it.

```
> CREATE TABLE x_my_table
```

Related Topics

[Writing a Report](#)

Tuning Insight for Your System

Insight is by default configured for smaller systems in order to prevent undesired memory or CPU usage. This section provides information to tune your configuration to keep pace with Moab Workload Manager.

Available Hardware

We recommend configuration based on the hardware available to Insight. The parameters listed below should be uncommented and tuned in the `/opt/insight/etc/config.groovy` configuration file.

- `messageQueue.workerCount = <processor count * 4> // i.e. 64 for a 16 core machine`
- `messageQueue.parserCount = 4 // Smaller systems may use less`
- `messageQueue.workerQueueCapacity = <workerCount * 25> // i.e. 1600 for a workerCount value of 64`
- `jdbc.c3p0.maxPoolSize = <workerCount + 6> // i.e. 70 for a workerCount value of 64`

i Please note that the `maxPoolSize` configuration parameter must be less than the maximum connections allowed by PostgreSQL or else errors will occur. This configuration parameter may be changed in PostgreSQL if it needs to be increased.

- `jdbc.c3p0.maxStatements = <workerCount * 512> // i.e. 32768 for a workerCount value of 64`

Job Archiving Schedule

Insight will periodically move the records of jobs that have completed into the `jobs` and `job_state_journal` collections of MongoDB. This keeps job-related database tables small enough that inserts and queries do not become slow. By default, job archiving will start every day at midnight and will archive all jobs that have been completed (or canceled) for seven or more days. This includes data in tables related to jobs (for example, job reservations and job state transitions will also be archived). Running and idle jobs will not be archived (irregardless of the job date).

If it appears that Insight is not keeping pace with Moab, you may want to configure the archiver to run more aggressively. This is most likely to be the case *only* if you have a large cluster with high numbers of jobs. One indication that you may want to do this is that the data in Insight is out of date by several RM poll intervals. See `RM POLLINTERVAL` in the *Moab Workload Manager Administrator Guide* to specify the interval between RM polls.

To Customize the Archiver Settings

Add or adjust the following parameters in the `/opt/insight/etc/config.groovy` configuration file.

Parameter	Default	Description
<code>archiver.jobArchiving.maxAgeDays</code>	7	Jobs that have been completed longer than this many days will be archived.
<code>archiver.jobArchiving.schedule</code>	"0 0 0 * * *" // Every day at midnight	Job archiving will happen according to this schedule, which is described in this javadoc .

For example:

```
archiver.jobArchiving.maxAgeDays = 3
archiver.jobArchiving.schedule = "0 0 */6 * * *" // Every 6 hours
```

i For archiving purposes, the job's age is the number of days it has been completed rounded down to the nearest integer value. For example, if a job has been completed for 3 1/2 days the archiver will consider the job's age to be 3.

Java Runtime Environment

Insight should be run on Oracle® Java® Runtime Environment (JRE) version 8.

Normally Insight will fail to start and will log an error to `/opt/insight/log/insight.log` if run on an unsupported JRE. The following is a sample error message:

```
java.lang.Exception: Unsupported Java Virtual Machine (OpenJDK 64-Bit Server VM): see
the Requirements section of the Insight Installation Guide.
```

This verification check prevents you from accidentally using a JRE which is known to have problems with Insight. However, there may be times when you want to force Insight to run on an unsupported JRE. This might happen if a supported JRE is not available for your platform. In these cases you tell Insight to skip the JRE verification check by setting the `insight.skip.jre.verification` parameter in the `/opt/insight/etc/config.groovy` configuration to "true". For example:

```
insight.skip.jre.verification = true
```

i By default, Insight sets its `JAVA_HOME` environment variable to `/user/java/latest`. To use a different JRE, edit `/opt/insight/etc/insight.conf` and change the value of `JAVA_HOME`.

Tuning the Insight Archiver Sample Rate

Context

Insight always captures the current snapshot of jobs; however, you can configure it to track statistics, usage, and trends in a more granular or relaxed fashion according to your needs.

To tune the Insight sample rate

Open the Insight configuration file (`/opt/insight/etc/config.groovy`) and uncomment and change the default **archiver.schedule** parameter.

- Default

```
// Archiver configuration - the archiver creates samples or historical data from
current state data
//archiver.schedule = "0 */1 * * * *" // Every minute
```

- Example

```
// Archiver configuration - the archiver creates samples or historical data from
current state data
archiver.schedule = "0 */10 * * * *" // Every 10 minutes
```

Schedule value

The schedule value is a cron-like string. It is a string that contains six space-separated fields representing second, minute, hour, day, month, and

weekday. You can give month and weekday names as the first three letters of the English names.

Example Patterns

0 0 * * * * – the top of every hour of every day

*/10 * * * * * – every ten seconds

0 0 8-10 * * * – 8:00, 9:00 and 10:00 every day

0 0/30 8-10 * * * – 8:00, 8:30, 9:00, 9:30 and 10:00 everyday

0 0 9-17 * * MON-FRI – on the hour from 9:00 to 5:00 on weekdays

0 0 0 25 12 ? – every Christmas at midnight

Related Topics

[The Insight Archiver Sample Rate](#)

[Changing the Pruning Policy](#)

Tuning your PostgreSQL Server

Adaptive Computing recommends you follow the PostgreSQL team's official documentation for tuning your server. Their tuning documentation can be found [here](#).

The following parameters are especially important to tune for use with Moab Insight:

- shared_buffers
- effective_cache_size
- checkpoint_segments
- checkpoint_completion_target
- work_mem

Changing the Pruning Policy

The Insight data retention and pruning behavior control how large the database grows and how far back in time it stretches.

You can configure the pruning policy via configuration files (`/opt/insight/etc/config.groovy` and `/opt/insight/etc/config.d/`).

In this topic:

- [Pruning the PostgreSQL Database on page 14](#)
- [Pruning the MongoDB Database on page 16](#)

Pruning the PostgreSQL Database

The following is a sample pruning configuration for the PostgreSQL database:

```
pruner.schedule.samples = "15 */1 * * * *"
pruner.schedule.acl = "25 */1 * * * *"
pruner.policy = [
  // Last hour => Granularity 1 per minute
  ['interval': ['unit': "HOUR", 'count': 1], 'granularity': ['unit': "MINUTE",
'each': 1]],

  // Then last day => Granularity 1 per 10 minutes
  ['interval': ['unit': "DAY", 'count': 1], 'granularity': ['unit': "MINUTE",
'each': 10]],

  // Then last week => Granularity 1 per hour
  ['interval': ['unit': "WEEK", 'count': 1], 'granularity': ['unit': "HOUR",
'each': 1]],

  // Then last month => Granularity 1 per 6 hours
  ['interval': ['unit': "MONTH", 'count': 1], 'granularity': ['unit': "HOUR",
'each': 6]],

  // Then (forever) => Granularity 1 per day
  ['interval': ['unit': "INFINITY"], 'granularity': ['unit': "DAY", 'each': 1]]
]
```

You can find the pruner policy within the policy section. The policy consists of several blocks enclosed with square brackets []. Each item contains two subsections: interval and granularity. The intervals described in this policy are applied one after another, beginning from the current moment. The order is important.

The `unit` fields can have one of the following values:

- MINUTE
- HOUR
- DAY
- WEEK
- MONTH
- YEAR
- INFINITY

`each` and `count` fields are unsigned integers.

The pruner policy is loaded and validated when Insight starts. If the given policy is invalid, Insight will fail to start.

Pruning Rules

Here is the list of rules which the pruning policy must satisfy:

- The policy cannot be empty. You must have at least one item with the interval and granularity set.
- When you go from the current time to the past, the granularity must not get smaller (for example, "Now I want 1 item per 10 minutes, but after an hour let it be 1 item per minute."). This would cause historical data loss, which is considered erroneous.
- The sample rate cannot be arbitrary.
- The set of valid granularity radices are: one minute, 10 minutes, hour, day, week.
- The set of valid sample rates is any number N that is a factor of all granularities greater than N and a multiple of all granularities less than N . So 2 is a valid sample rate, because 2 minutes is factor of all granularities greater than 2 and a multiple of 1; however, 3 is not a valid sample rate, because it doesn't divide evenly into 10.

One month is a valid granularity because it is divided by week radix and there is no greater radix it should divide.

20 minutes is a valid granularity because it is divided by 10 minutes radix and the greater radix – hour – is divided by 20 minutes.

11 minutes is not a valid granularity.

Pruning Schedule

In the sample configuration provided above there is a `schedule` field. This field controls the schedule when the corresponding pruner background thread gets invoked. This schedule impacts directly how often the sample data is aligned to the pruning policy.

The schedule value is a cron-like string. It is a string that contains six space-separated fields representing second, minute, hour, day, month, and weekday. You can give month and weekday names as the first three letters of the English names.

Example Patterns

0 0 * * * * – the top of every hour of every day

*/10 * * * * * – every ten seconds

0 0 8-10 * * * – 8:00, 9:00, and 10:00 every day

0 0/30 8-10 * * * – 8:00, 8:30, 9:00, 9:30, and 10:00 everyday

0 0 9-17 * * MON-FRI – on the hour from 9:00 to 5:00 on weekdays

0 0 0 25 12 ? – every Christmas at midnight

Pruning the MongoDB Database

To keep the size of the Insight MongoDB database from growing too large, each piece of data in MongoDB is given an expiration date.

After the data has been around longer than its expiration date, it will be purged. By default the time to live for a piece of data in MongoDB is two years (63244800 in seconds). However, this can be adjusted for each collection by adjusting the following parameters in the `/opt/insight/etc.config.groovy` file:

Parameter	Default (in seconds)	Description
<code>archiver.jobSamples.expireAfterSeconds</code>	63244800	Job samples whose age (based on the <code>sampleDate</code> field) is older than this value will be deleted.
<code>archiver.jobStateJournal.expireAfterSeconds</code>	63244800	Job state journal entries whose aged (based on the <code>timestamp</code> field) is older than this value will be deleted.
<code>archiver.jobs.expireAfterSeconds</code>	63244800	Jobs that have been completed longer than this value (based on the <code>completedDate</code> field) will be deleted.
<code>archiver.nodeSamples.expireAfterSeconds</code>	63244800	Node samples whose age (based on the <code>sampleDate</code> field) is older than this value will be deleted.
<code>archiver.nodes.expireAfterSeconds</code>	63244800	Nodes that have not been reported by Moab Workload Manager more than this number of seconds ago will be deleted to save disk space. (based on the <code>lastUpdatedDate</code> field)
<code>archiver.nodes.expireAfterSeconds</code>	63244800	Nodes that have not been reported by Moab Workload Manager more than this number of seconds ago will be deleted to save disk space. (based on the <code>lastUpdatedDate</code> field)
<code>archiver.nodesAuxiliary.expireAfterSeconds</code>	63244800	Supporting information about nodes (e.g. generic resources) that have not been reported by Moab Workload Manager more than this number of seconds ago will be deleted to save disk space. (based on the <code>lastUpdatedDate</code> field)
<code>archiver.reservations.expireAfterSeconds</code>	63244800	Reservations that ended longer than this time ago will be deleted to save disk space. (based on the <code>endDate</code> field)

Related Topics

[The Insight Pruning Policy](#)[Tuning the Insight Archiver Sample Rate](#)

Configuring Reliable Message Delivery

Context

Moab and the Insight daemon gracefully handle disconnects or restarts. If you restart Moab, you do not have to restart Insight. To handle instances of disconnect, or downtime, Moab stores all data it attempts to send to Insight in memory and on disk. Once Moab generates enough data to meet the maximum storage size, it begins to delete the oldest data and make room for the new data. The size of the data on the disk is two times the maximum storage size. By default, the maximum storage size is 1 GB with 2 GB max on disk. You can customize the storage size for your unique environment.

To configure reliable message delivery

1. Open the Moab configuration file on the Moab head node and set the **INSIGHTSTORESIZE** and **INSIGHTSTOREDIR** configuration parameters. See Appendix A: Moab Parameters in Moab Workload Manager for parameter information.

```
[moab]$ vi /opt/etc/moab
...
INSIGHTSTORESIZE 2048 # 2 GB store size with 4 GB on disk
INSIGHTSTOREDIR /tmp/insight_store
...
```

If **INSIGHTSTOREDIR** is a relative path, the Moab home directory is prepended to it. It uses the given path if it is an absolute path.

2. Restart Moab in order for the new configuration parameters to take effect.

Related Topics

[Changing the Pruning Policy](#)[Tuning the Insight Archiver Sample Rate](#)

Troubleshooting

This topic contains information on troubleshooting and resolving issues.

- [liquibase.exception.LockException: Could not acquire change log lock](#)
- [Unable to view older job information after upgrading Insight](#)

- [Job state is completed yet job has a null completion time](#)
- [Remove node or job from Insight](#)
- [Sync Insight database with Moab database](#)

liquibase.exception.LockException: Could not acquire change log lock

Cause

This usually happens when Insight is prematurely terminated the first time it is run after being installed. The termination would have occurred during or right after Insight was setting up the database schema.

1. The main symptom of this problem is that whenever you start Insight you'll see the following error in the `/opt/insight/log/insight.log` and Insight will terminate abruptly several minutes after starting:

```
java.lang.RuntimeException: liquibase.exception.LockException: Could not acquire
change log lock.  Currently locked by geminst02 (fe80:0:0:0:f816:3eff:fe12:44a9%2)
since 12/9/14 9:24 AM
at com.ace.insight.data.service.DbInitService.validateDbConsistency
(DbInitService.java:130) ~[insight-8.1.jar:8.1]
at
com.ace.insight.data.service.DbInitService$$FastClassBySpringCGLIB$$91c2bfb8.invoke
(<generated>) ~[spring-core-4.0.3.RELEASE.jar:8.1]
at org.springframework.cglib.proxy.MethodProxy.invoke(MethodProxy.java:204) ~
[spring-core-4.0.3.RELEASE.jar:4.0.3.RELEASE]
at
org.springframework.aop.framework.CglibAopProxy$$CglibMethodInvocation.invokeJoinpoi
nt(CglibAopProxy.java:711) ~[spring-aop-4.0.3.RELEASE.jar:4.0.3.RELEASE]
```

2. If the error displays, check the `DATABASECHANGELOGLOCK` table for entries where the locked column is set to 't'.

```
[root]# sudo su - postgres
[postgres]$ psql -d moab_insight_reference -c "select * from
DATABASECHANGELOGLOCK;"
 id | locked |          lockgranted          |          lockedby
-----+-----+-----+-----
  1 | t      | 2014-12-09 09:24:29.538-07 | geminst02
(fe80:0:0:0:f816:3eff:fe12:44a9%2)
(1 row)
```

Prevention

To prevent this avoid doing anything that might terminate the Insight process until you see the following lines appear in the `insight.log`:

```
2014-12-11T17:28:57.920+0400    main    INFO    com.ace.insight.app.ApplicationListenerBean    0    The application has been started.
Insight version: master
2014-12-11T17:28:57.950+0400    main    INFO    com.ace.insight.app.Application    0    Started Application in 139.173 seconds (JVM running for 142.189)
```

These lines are an indication that Insight has set up the database and released the locks. The service can be shut down at this point.

Troubleshooting Methods

Method 1

Use this method first. This method is safe to use regardless of whether you have data in your moab_database as it will not disturb this moab_insight database.

Remove the lock line in the DATABASECHANGELOGLOCK table.

```
[postgres]$ psql -d moab_insight_reference -c "DELETE FROM DATABASECHANGELOGLOCK where id>0;"
[postgres]$ exit
[root]# service insight start
```

If the issue still exists, follow the instructions for Method 2.

Method 2

This method assumes you do **not** have data in the database that you need to keep. This is likely only going to be the case if you have just barely installed Insight.



This method will delete all data in your moab_insight database. If you **do** have data in the database, and you have already tried Method 1, do not use this method. Contact your Adaptive support representative.

Remove the Insight database and restart Insight.

1. Run the following:

```
[root]# service insight stop #Make sure Insight is not running
[root]# su - postgres -c "dropdb moab_insight"
[root]# su - postgres -c "dropdb moab_insight_reference"
[root]# su - postgres -c "createdb -O moab_insight moab_insight"
[root]# su - postgres -c "createdb -O moab_insight moab_insight_reference"
[root]# service insight start
```

2. Wait until you see the started message in the insight.log. Example:

```
2014-12-11T17:28:57.920+0400    main    INFO    com.ace.insight.app.ApplicationListenerBean    0    The application has been
started. Insight version: master
2014-12-11T17:28:57.950+0400    main    INFO    com.ace.insight.app.Application    0    Started Application in 139.173 seconds (JVM running for 142.189)
```

3. If you are using Viewpoint, you must manually grant SELECT permissions to the mws user.

```
[root]# su - postgres -c "psql -d moab_insight -c 'GRANT SELECT ON ALL TABLES IN
SCHEMA public TO mws;'"
```

Unable to view older job information after upgrading Insight

Cause

Beginning the 9.0.2 release, you can choose to postpone the upgrade of the Insight database when upgrading Insight. This option provides for a faster upgrade of Insight, however it does not enable access to stored database information until the database is upgraded.

Troubleshooting Method

View the `/opt/insight/log/insight.log` file and confirm that the database was not updated. If the upgrade occurred, at the end of the file, you will see output similar to the following:

```
2016-06-28T06:25:13.120-0600    main    INFO
com.ace.insight.data.service.dbinit.DbUpgradeService 0 Database has been upgraded to
current version
```

If the database was not updated, do the following:

1. Set the following in the `/opt/insight/etc/config.groovy` file.

```
insight.skip.database.migration = false
```

2. Restart Insight.

Job state is completed yet job has a null completion time

Cause

If you notice that a job's state is completed, yet the job has a null completion time, it is likely that Insight did not process a job end event. For example:

```
[root]# su - postgres
[postgres]$ psql -d moab_insight -c "select job_id,job_name,job_state,wallclock_
seconds,job_start_datetime,completion_datetime from workload_view where job_
name='Moab.411'"
job_id | job_name | job_state | wallclock_seconds | job_start_datetime |
completion_datetime
-----+-----+-----+-----+-----+-----
403    | Moab.411 | COMPLETED | 3600 | 2015-01-19 10:30:00.000-07 |
```

Troubleshooting Method

To rectify this data inconsistency it is possible to insert a job end event into the Insight database manually. Do the following:

1. Notice what the UTC offset is. The UTC offset represents the difference in hours between local time and UTC time. There are many ways of determining this, but one way is to look at the trailing digits after the final dash in the `job_start_datetime`. Since the `job_start_datetime` in the

example above is "2015-01-19 10:30:00.000-07", the UTC offset is "07".

- Determine the job's completion time. Since the database does not have it, we first try querying checkjob on the machine running Moab Workload Manager. This may or may not be the same machine where your PostgreSQL database is.

```
[root]# checkjob Moab.411 | grep Completion
Completion Code: 0 Time: Mon Jan 19 11:28:00
```

If checkjob reports "ERROR: invalid job specified: Moab.411" this means that Moab has already purged the record of the job. You'll have to use the events file. Navigate to the /opt/moab/stats folder and search for JOBEND events relating to Moab.411. This will also be on the machine running Moab Workload Manager.

```
[root]# cd /opt/moab/stats
[root]# grep -r 'Moab.411' * | grep JOBEND | cut -c1-100
events.Mon_Jan_19_2015:11:28:00 1421692080:372 job      Moab.411      JOBEND
0      1      bob
```

- Create a primary key for the row you wish to insert. To do this create a row in the job_state_journal_id table and set the id in your newly created row to DEFAULT. If Moab and PostgreSQL are running on different machines, be sure to return to the machine running PostgreSQL. For example:

```
[postgres]$ psql -d moab_insight -c "INSERT INTO job_state_journal_id(id) VALUES
(DEFAULT) RETURNING id"
 id
----
 522
(1 row)

INSERT 0 1
```

- In the example above the primary key that was generated is 522. Now that we have a primary key, we can use it to manually insert the missing job end event into the job_state_journal table. We know from the events file that the job completed on January 19, 2015 at 11:28 AM. We also know from our earlier work that the UTC offset is 07. The date we wish to insert will be "2015-01-19 11:28:00.000-07".

```
[postgres]$ psql -d moab_insight -c "INSERT INTO job_state_journal(id,job_
id,state,timestamp_datetime) VALUES (522,403,'COMPLETED','2015-01-19 11:28:00.000-
07')"
INSERT 0 1
```

Remove node or job from Insight

To remove a node or job from the Insight PostgreSQL and MongoDB databases use:

```
/opt/insight/utils/delete-objects-from-db.py
```

i Run `delete-objects-from-db.py -h` for more usage information.

Sync Insight database with Moab database

i This procedure lets you force a sync of the Insight databases with the Moab database. For example, if Insight has been down for more than two days.

To temporarily stop all traffic to Insight and sync all jobs in the Insight database with the jobs in the Moab database use:

```
/opt/insight/utils/sync.py
```