LUX 2.8 Upgrade Guide

Last Update: 17 JUN 19

1 Prerequisites

1.1 Recommended Minimum Prerequisite Hardware for 2.8

Hardware requirements for LUX may vary greatly based on customer requirements. Please consult ICG personnel for additional information with regards to hardware allocation recommendations.

LUX Component	Required Hardware
Engine Server	16 CPU / 32GB RAM / 50GB SSD
User Interface Server	4 CPU / 8GB RAM / 50GB HDD
Database Server	8 CPU / 16GB RAM / 2TB SSD
Alert Batch Analytic Server (optional)	16 CPU / 32GB RAM / 50GB HDD

1.2 Required Prerequisite Software for 2.8

None of the prerequisite software supported versions have been changed for the 2.8 release. As such, existing installations of required prerequisites should not need to be updated.

Software	Version
CentOS or RHEL Linux	7.x
JDK	1.8.x
MongoDB	3.6.x

1.3 Optional Prerequisite Software for 2.8

Software	Version	Alternative to:
ActiveMQ	5.3+	Engine REST Alerter

LUX can be configured to use ActiveMQ for the alerting component of the installation if necessary to match requirements. Please contact ICG personnel for configuration information.

1.4 Prerequisite Software Alterations for 2.8

Software	Version	Replaced by:	Version
MySQL	5.7.x	MongoDB	3.6.x

Beginning in LUX 2.8.0 MySQL is no longer required, as all data that was previously stored in MySQL has now been configured for storage in MongoDB. Migration of data from MySQL to MongoDB is covered in Section 3.2.

2 Pre-Upgrade Steps

2.1 Stopping the LUX & LUXEngine Services

Validate that both LUX & LUX Engine services have been stopped gracefully before proceeding further in the upgrade process.

1. Stop LUX Engine & LUX UI :
    a. On the luxengine VM, `sudo service luxengine stop`
    b. On the lux ui VM, `sudo service lux stop`

2.2 Backing Up Configuration & Alert Data

When performing an RPM or YUM upgrade, existing configuration for the UI & Engine should be preserved as `.rpmsave` files. However, it is still highly recommended for Administrators to back up all configuration for both the UI & Engine components of LUX before upgrading LUX versions.

2.2.1 Backing Up Engine Configuration

The LUX Engine has multiple configuration files & directories that should be backed up prior to upgrading to LUX 2.8.0.

Within `/usr/local/lux/engine/`

1. `license/lux.lic`
2. `bin/set-env.sh`

Within `/usr/local/lux/engine/EngineMain/data/`

1. `plugins/*`
2. `plugin_data/*`
3. `conf/*`
    a. This directory contains virtually all Engine properties and `.xml` configuration files.

Within `/etc/init.d/`:

1. `luxengine`

2.2.2 Backing Up UI Configuration

The LUX UI has multiple configuration files & directories that should be backed up prior to upgrading to LUX 2.8.0.

Within **`/usr/local/lux/ui/`**

1. **`certs/`**
2. **`conf/`**
3. **`config/`**
4. **`forms/`**
5. **`mapserver/`**
6. **`stores/`**
7. **`templates/`**

Within **`/usr/local/lux/ui/webapps/lux/WEB-INF/`**

1. **`spring/`**
2. **`classes/`**

Within **`/usr/local/lux/ui/webapps/lux/`**

1. **`mapserver/`**
2. **`templates/`**

Within **`/etc/init.d/`**:

1. **`lux`**

2.2.3 Consider Backing up MongoDB

ICG recommends backing up any MongoDB data prior to performing updates to the LUX system. The following command shows an example of using MongoDump to write MongoDB data to file.

1. Backup the LUX DB within MongoDB.
    a. `sudo mkdir /var/backups/mongoLUXbackups`
    b. `sudo mongodump --db lux --out /var/backups/mongoLUXbackups/`date +"%m-%d-%y"`

2. If needed due to failed LUX 2.8 Upgrade:
    a. Restore MongoDB LUX DB: `sudo mongorestore --db lux --drop /var/backups/mongoLUXbackups/`**`<<DATE>>`**`/lux/`

3 LUX UI Installation & DB Alterations

LUX UI v2.8 includes two database alteration tools that are necessary prior to formal configuration of the UI installation. (UI configuration discussed in Section 4.1). These are the MySQL Data Migration tool discussed in Section 3.2.1 and the Alert Transformation Tool discussed in Section 3.2.2.

3.1 User Interface (UI) Initial Installation

Use YUM to install the LUX UI if your environment is not on a disconnected LAN (see 3.1.2). If your environment is not able to access the internet, or ICG YUM repositories, use the RPMs provided in your LUX distribution software package (see 3.1.1).

3.1.1 Install LUX packages from RPM

If you are **not** using YUM to install the LUX UI, you will need to install JSVC manually. If you are using YUM to install the LUX UI, skip to 4.1.2.

1. Install JSVC
    a. `sudo yum install jsvc`
    b. If this does not run, download and try the following:
        i. On Centos 6.x: `sudo rpm -Uvh jakarta-commons-daemon-jsvc-1.0.1-8.9.el6.x86_64.rpm`
        ii. On Centos 7.x: `sudo rpm -Uvh apache-commons-daemon-jsvc-1.0.15-11.fc24.x86_64.rpm`
2. `sudo rpm -Uvh lux-ui-<VERSION>-noarch.rpm`

3.1.2 Install LUX package from YUM

Install the LUX UI Package:

1. `sudo yum upgrade lux-ui`

3.2 MySQL Data Migration & Alert Transformation

As LUX 2.8 no longer requires MySQL, all data within the MySQL DB will need to be migrated into MongoDB. Additionally, by default, LUX 2.8 alerts are formatted differently. As such, existing alerts in MongoDB will need to be transformed into a new format in order to be visualized in the LUX UI.

3.2.1 Run the Migrator Tool

Included in the LUX UI rpm is a MySQL to MongoDB migration tool. By default it will pull the existing location for MySQL & MongoDB installations from the UI's **`db-rules.properties`** and **`mongo.properties`** configuration files.

1. Verify that configuration in the DB properties files is still correct after upgrading LUX UI. You will need to copy configuration from the .rpmsave files to the new properties files.
    a. **`cp /usr/local/lux/ui/webapps/lux/WEB-INF/classes/mongo.properties.rpmsave /usr/local/lux/ui/webapps/lux/WEB-INF/classes/mongo.properties`**
    b. **`cp /usr/local/lux/ui/webapps/lux/WEB-INF/classes/db-rules.properties.rpmsave /usr/local/lux/ui/webapps/lux/WEB-INF/classes/db-rules.properties`**
2. Navigate to the Migrator's directory on the LUX UI VM.
    a. **`cd /usr/local/lux/ui/tools/migrator`**
3. Execute the Migrator shell script.
    a. **`./migrator.sh`**
    b. **NOTE: There is a known issue that may cause incorrect error messages regarding Rule History to appear during migration. These may be disregarded.**
4. Verify that data was migrated successfully by checking all newly created collections in MongoDB.
    a. Example Mongo Commands:
        i. **`db.projects.findOne();`**
        ii. **`db.users.findOne();`**
        iii. **`db.userGroups.findOne();`**
        iv. **`db.dataGroups.findOne();`**
        v. **`db.rules.findOne();`**
        vi. **`db.watchboards.findOne();`**

3.2.2 Alert Transformation Tool

Included in the LUX UI rpm is an Alert Transformation tool. By default it will pull the existing MongoDB installation location from the UI's **`mongo.properties`** configuration file. Note that this may take a number of hours depending on the volume of alerts in Mongo.

1. Navigate to the Transformation tool's directory on the LUX UI VM.
    a. **`cd /usr/local/lux/ui/tools/migrator`**
2. Execute the alertFormatConverter shell script.
    a. **`./alertFormatConverter.sh`**

While the alerts are being transformed to the new format, you may continue following the Upgrade Guide. Executing the steps in the upcoming sections will not affect the alert transformation script.

3.2.3 MongoDB Index Alterations

Due to alert format changes, existing indices within Mongo will need to be replaced with newly configured indices.

1. Login to MongoDB.
    a. `mongo -ulux -plux lux`
    b. `use lux;`
2. Drop all indices.
    a.
```
db.getCollectionNames().forEach(function (d) {
   db[d].dropIndexes();
});
```

3.2.4 MongoDB Index Creation

As the existing indices were dropped in Section 3.2.3, new indices will need to be generated using the contents of Appendix 1.

1. Login to MongoDB:
    a. `mongo -ulux -plux lux`
    b. `use lux;`

Copy and paste all index creation commands from Appendix 1 into the Mongo shell:

1. Example of a single AOI index: `db.aoi.createIndex({"name" : 1, "id" : 1});`

4 LUX UI & Engine Upgrade

4.1 LUX UI Configuration

The following section discusses generalized configuration of LUX UI after an RPM upgrade. However, as a LUX installation may be highly configured on a per-system basis, there may be additional configuration that will need to be restored outside the scope of this guide.

1. Restore: `/etc/init.d/lux`

2. If not using ActiveMQ and instead using the default REST alerter, delete jms.xml
    a. `/usr/local/lux/ui/webapps/lux/WEB-INF/spring/jms.xml`

3. Restore:
`/usr/local/lux/ui/webapps/lux/WEB-INF/classes/spring-context.properties`

4. Restore MongoDB Configuration for the UI:
    a. `/usr/local/lux/ui/webapps/lux/WEB-INF/classes/mongo.properties`

5. Restore certs for the UI:
    a. `/usr/local/lux/ui/certs`

6. Configure **Optional** Information for the LUX UI:
    a. Modify or restore the following file with the desired timeline values for the UI:
        i. `/usr/local/lux/ui/webapps/lux/config/timeline/timerangeselector.json`
    b. Add Installation details, such as customer name, LUX serial number, and license expiry dates to `/usr/local/lux/ui/webapps/lux/WEB-INF/classes/license.properties`
    c. Add UI Installation name and validate product version in: `/usr/local/lux/ui/webapps/lux/WEB-INF/classes/version.properties`
    d. Add the LUX Engine Version to: `/usr/local/lux/ui/webapps/lux/WEB-INF/classes/engine.properties`

7. `sudo service lux start`
    a. **NOTE:** LUX UI will automatically generate a set of new indices for MongoDB when the application is started for the first time. Depending on the volume of existing data in MongoDB, this may take multiple hours to complete.
8. `sudo chkconfig lux on`
9. Check the UI `catalina.out` log for errors. Rectify any found and restart LUX.
    a. `less +F /usr/local/lux/ui/logs/catalina.out`
10. Navigate to the LUX UI Webpage
    a. The default login page URL will be as follows:
    b. URL: https://<HOSTNAME>/lux/

4.2 LUX Engine Upgrade

Either install the engine from RPM files provided from ICG using the instructions in section 4.2.1, or install directly from ICG's YUM repository using the instructions in 4.2.2. You may need to execute the following instructions with sudo or as root.

4.2.1 Install packages from RPM

1. Install the LUX Engine Package:
    a. `sudo rpm -Uvh lux-engine-<VERSION>.noarch.rpm`
2. Restore a valid `/usr/local/lux/engine/license/lux.lic` license file from backup.
3. Restore `/etc/init.d/luxengine` from backup.
4. `chkconfig luxengine on`

4.2.2 Install packages from YUM

1. Install the LUX Engine Package:
    a. `sudo yum install lux-engine`
2. Restore a valid `/usr/local/lux/engine/license/lux.lic` license file from backup.
3. Restore `/etc/init.d/luxengine` from backup.
4. `chkconfig luxengine on`

4.2.3 Restore LUX Engine Configuration from Backup

Within `/etc/init.d/`, restore:

1. `luxengine`

Within `/usr/local/lux/engine/`, restore:

1. `license/lux.lic`
2. `bin/set-env.sh`

Within `/usr/local/lux/engine/EngineMain/data/`, restore:

1. `plugins/*`:
    a. Specifically, only restore plugins that are specific to your LUX installation and would not be included in a commercial LUX software release.
2. `conf/*`
    a. This directory contains virtually all Engine properties and `.xml` configuration files. Administrators should restore all configuration pertinent to their previous LUX Engine installation.
    b. Configuration files & directories within `conf/` to pay special attention towards:
        i. `engine.properties`
        ii. `lux.properties`
        iii. `analytics.xml`
        iv. `enrichments.xml`
        v. `ingesters/*`
        vi. `enrichments/*`
        vii. `analytics/*`

4.2.4 Starting and stopping LUX Engine

1. To start LUX Engine: `sudo service luxengine start`
2. To stop LUX Engine: `sudo service luxengine stop`

4.2.5 Install Engine Admin Console

1. If using the LUX UI, verify that the Admin Console .war file exists on the UI host (**`/usr/local/lux/ui/webapps/AdminConsole.war`**).
    a. The **`AdminConsole.war`** file is included on the Engine VM in most installations under **`/usr/local/lux/engine/AdminConsole.war`**, and can be copied to the UI VM.
2. If using the Engine as standalone, verify that an installation of Apache Tomcat exists on the Engine host and copy the **`/usr/local/lux/engine/AdminConsole.war`** file to the **`$CATALINA_HOME/webapps/`** directory.
3. Login to the LUX Engine Administration Console
    a. The default EAC URL will be as follows:
    b. URL: `https://<HOSTNAME>/AdminConsole/`

Appendices

Appendix I: Mongo Index Creation

```
// Default Mongo TTL set to 14 days, which matches the Default LUX UI time slider.
db.alerts.createIndex({"createdDate" : 1}, {"expireAfterSeconds": 1209600, "background": true});

db.alerts.createIndex({"projectId" : 1}, {"background": true});
db.alerts.createIndex({"rule.id" : 1, "createdDate": -1}, {"background": true});

db.aoi.createIndex({"name" : 1, "id" : 1});
db.aoi.createIndex({"id" : 1, "name" : 1});
db.aoi.createIndex({"lastModified" : 1});
db.aoi.createIndex({"id" : 1,"name":1,"source":1,"lastModified":1,"managed":1,"classification":1,"version":1,"aoiType":1,"description":1,"tags":1,"dynamic":1});

db.change.createIndex({"objectId" : 1});
db.change.createIndex({"date" : 1});
db.change.createIndex({"objectType" : 1, "date":1});
```

```
db.datastores.createIndex( { "name": 1 }, { unique: true } );

db.forms.createIndex( { "name": 1 }, { unique: true } );

db.mapconfigs.createIndex( { "name": 1 }, { unique: true } );

db.streams.createIndex({"streamName" : 1});

db.users.createIndex( { "name": 1 }, { unique: true } );

db.groups.createIndex( { "name": 1 }, { unique: true } );

db.datagroups.createIndex( { "name": 1 }, { unique: true } );

db.rules.createIndex( { "project": 1 });
db.rules.createIndex( { "status": 1 });
```