# Elixir Administration Tools

## Release 4.0.0

*Elixir* Technology Pte Ltd

# Elixir Administration Tools: Release 4.0.0

*Elixir* Technology Pte Ltd

Published 2015

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1
# Elixir Administration Tools

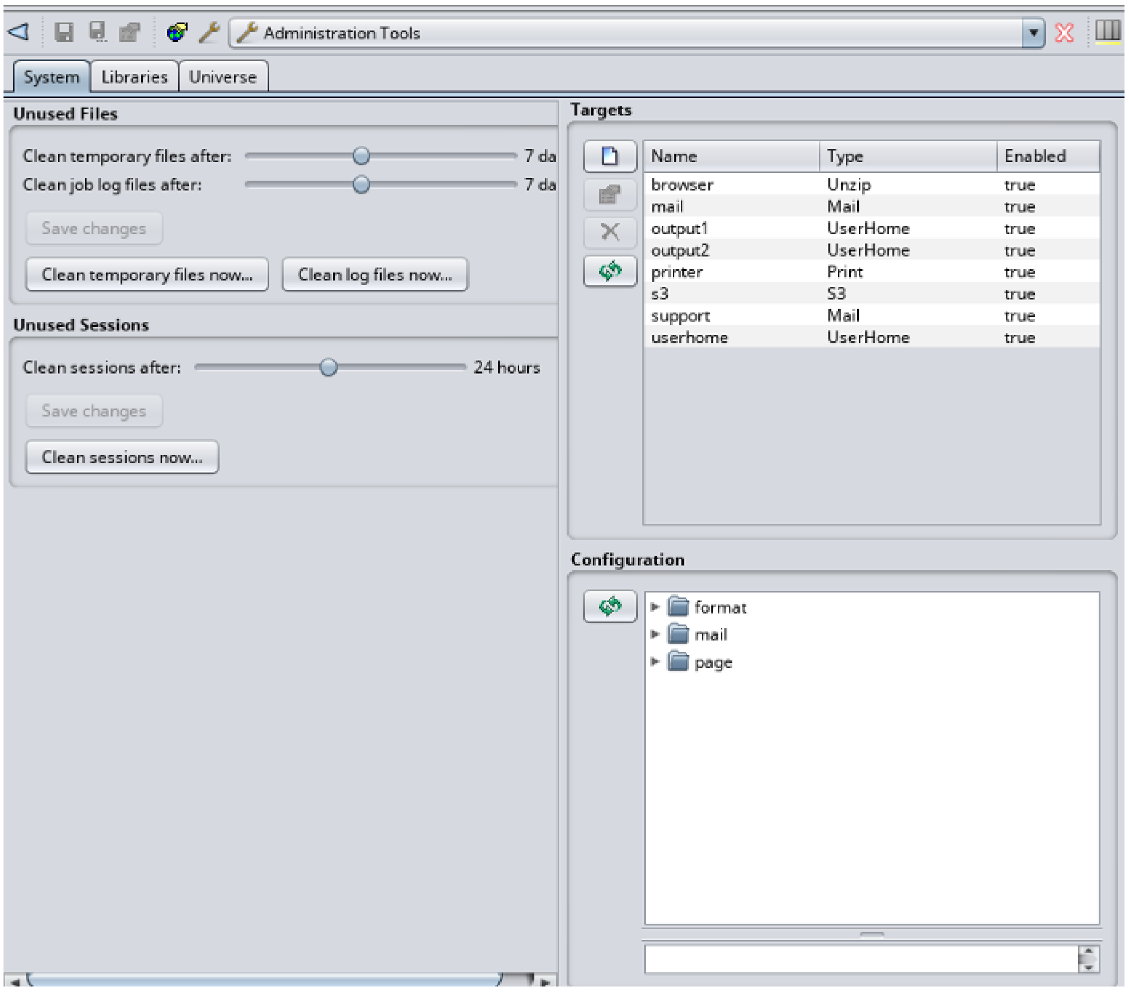
## Overview

On clicking the Admin Tools... button on the Action Bar, you will see the Administration Tools panel. These tools include System, Libraries, and Universe.

## System

The System page allows you to clean up unwanted log and temporary files, set targets and configure formats such as date and time.

**Figure 1.1. System**



| Name | Type | Enabled |
|---|---|---|
| browser | Unzip | true |
| mail | Mail | true |
| output1 | UserHome | true |
| output2 | UserHome | true |
| printer | Print | true |
| s3 | S3 | true |
| support | Mail | true |
| userhome | UserHome | true |

## Unused Files

Allows you to set the duration after which unused temporary and log files are automatically deleted.

The default duration is set to 7 days.

You can also elect to clean up the temporary and log files immediately, by clicking **Clean temporary files now** and **Clean log files now**.

## Unused Sessions

Allows you to set the duration after which unused sessions are automatically deleted.

The default duration is set to 24 hours.

You can also elect to clean up the unused sessions immediately, by clicking **Clean sessions now**.

## Targets

After a job finishes, a target may pick up the result. Different targets may be chained for further processing. Target processing is also a task run by the Job Engine.

### Target Constants

If you find yourself typing a string repetitively when configuring targets, you can define that string as a constant. Then you can refer to that constant in target property values like ${constant-name}. Target constants can be enabled/disabled. Only enabled constants can be used in target configurations. You can define multiple constants with the same name, but only one of them can be enabled at any time.

### Target List

- Target Creation/Update/Deletion
  All manipulations can be done in the Target Wizard. For most of the targets, configuration is simple. You need to provide a name, enable the target and provide values for required target properties.

- Status
  Targets can also be enabled/disabled. Only enabled targets can be used in RenderReport task. You can define multiple targets with the same name, but only one of them can be enabled. If you make one target enabled, the rest of the targets with the same name will be disabled automatically.

  Targets can be be available/unavailable to anonymous users. When the **Available to anonymous users** check box is selected, the target is accessible in anonymous mode. By default, only the "browser" target is available.

- Target Properties and Parameters
  A target requires certain properties such as file name, folder name, port number and so on.

  When editing a target, the Target Wizard lists all required properties for the target. An administrator has several options when configuring those property values:

  - Provision of an exact string value for some properties.

  - Reference to target constants.

  - Definition of parameters in some properties if they should be provided by end users. For example, the parameter in property "filename" can be defined as "${file#report}_${date}". End users should provide values for those parameters when invoking targets.

- Output Types for Report Rendering
  A target provides various output types such as CSV, HTML, PDF, Microsoft formats and so on.

  After the properties are set, the Target Wizard enables you to choose among different output types for report rendering purpose. You can choose multiple types at a time for convenience. You can also print the report directly.

**Table 1.1. Output Types for Report Rendering**

| | |
|---|---|
| CSV | Glint |
| HTML | Line Printer |
| Logical RML | Logical RML Tree |
| Microsoft DOCX | Microsoft Excel |
| Microsoft PPTX | ODF Spreadsheet |
| PCL | PDF |
| Postscript | Print |
| Rich Text Format | Simple HTML |
| XML | Zipped Bitmap |
| Zipped Jpeg | Zipped Png |
| Zipped Svg | Zipped Tiff |
| Zipped Wbmp | |

## Create New Target

When you create a new target using the Target Wizard, you should fill in values, for example parameters, into the property fields. Nested parameters are not supported. For example, you cannot specify a parameter like ${file##${reportname}}.

### JDBC Target

A JDBC Target allows reports to be written directly into a database. This is useful if you have some subsequent program to pick them up or otherwise act on them - for example a document management system. Each report is written as a record into a specific table in the database. The report data itself is stored as a BLOB. Before you can use the JDBC target, you need to set up a database with a table that has the correct schema to accept a report file. An example as shown:

```
CREATE TABLE JOBOUTPUT (
  id INTEGER NOT NULL GENERATED ALWAYS AS
   IDENTITY (START WITH 1, INCREMENT BY 1),
  name VARCHAR(256) NOT NULL,
  lastModified BIGINT NOT NULL,
  content BLOB NOT NULL,
  CONSTRAINT JOBOUTPUT_PK PRIMARY KEY(id) )
```

Once the database is setup, configuration will write into a table called JobOutput in the Derby database that is built into the Elixir Repertoire Server:

**Table 1.2. JDBC Target Configurations**

| Name | Value |
|---|---|
| Driver | org.apache.derby.jdbc.EmbeddedDriver |
| URL | jdbc:derby:/home/.../jdbctarget |
| Table | ${update table##JobOutput} |
| User | ${userid##Enter your userid} |
| Password | ${password##Enter your password} |
| Filename | ${filename##Job_Output_Report} |
| Overwrite | true |

## JMS Target

A JMS Target can be used for asynchronous messaging. JMS applications can use job messages as a form of managed request/response processing, to give remote feedback to the users on the outcome of their send operations and the fate of their messages. Examples of job messages are Exception, Expiration, Confirm on arrival (COA), Confirm on delivery (COD), etc.

**Table 1.3. JMS Target Configurations**

| Name | Value |
|---|---|
| Destination | RQueue |
| User | ${userid##Enter your userid} |
| Password | ${password##Enter your password} |
| Filename | ${filename##Pet_Store_User_Accounts_Report} |
| Reply required | true |
| Timeout | ${reply timeout in secs##30} |
| Reply success keyword | ${reply success keyword##OK} |
| Reply success pattern | ${reply success pattern##^.*OK.*$} |

## Mail Target

A Mail Target allows the output to be sent by email. There are a number of parameters to specify, but remember that you can use substitutions to avoid hard-coding those that you decide need to be flexible. The report will be sent as an attachment by email, so you can choose the render format you prefer.

**Table 1.4. Mail Target Configurations**

| Name | Value |
|---|---|
| To | ${to##sam@elixirtech.com} |
| SMTP host | elixir.aspirin |
| From | ${from##susan@elixirtech.com} |
| Cc | ${cc##bob@elixirtech.com} |
| Subject | ${subject##Pet Store User Accounts Report From Elixir Server} |
| Message | ${message##Your report is attached.} |
| Filename | ${filename##Pet_Store_User_Accounts_Report} |

## PDF Signer Target

PDF Signer Target is used when a PDF output needs to be signed digitally. It prints a "signature" in a PDF file when the file is rendered.

**Table 1.5. PDF Signer Target Configuration**

| Name | Value |
|---|---|
| Keystore | ${key.url##C:/Java/jre7/bin} |
| Keystore type | ${keystore-type} |
| Store encrypted password | false |
| Store password | ${key-password##keyStorePassword} |
| Key alias | ${key-alias} |
| Signer appearance | ${signer-appearance##self-signed} |
| Info reason | ${info-reason##Reason for signature} |
| Info location | ${info-location##bottom-left} |
| Sign width | ${sign-width##100} |
| Sign height | ${sign-height##60} |
| Sign page | sign-last-page |
| Sign image | ${signature-image-url##D:/image/logo.png} |
| Certification level | ${certificate-level##certificate-no-changes} |
| Filename | Specify file name here |

### PDF Signer Target Properties

**Signer appearance**

There are two types of signing mode. They are *self-sign* and *wincer-sign*.

*self-sign* signer can be generated using Java key generator (keytool.exe). An example of a signature command is as follows :

```
keytool -genkey -keyalg RSA -alias QAkey -keypass
mypassword
  -keystore keystore.ks -dname "cn=My Key Name, c=SG"
```

*wincer-sign* is recommended if higher security is required. Certificate obtained is installed to the web browser and needs to be exported to be used in PDF signing.

**Keystore type**

For *self-sign*, no keystore type is required. For *wincer-sign*, the keystore type will depend on the respective vendor. For example, a VeriSign keystore type will be `pks12`.

**Keystore**

The directory of the keystore is entered here. *repository* is prohibited for this parameter.

**Key alias**

This parameter is optional. It is to define the alias used by the keystore file.

**Store password**

*key-password* is the password entered when creating the keystore file. The password can either be in encrypted or unencrypted form.

**Store encrypted password**

This parameter holds a boolean value, `true` if the key password is encrypted and `false` if otherwise.

**Sign page**

This parameter is to specify the page that the signature is placed. *sign-no-page* will mean that the signature will not be visible. *sign-first-page* and *sign-last-page* implies that the signature will be placed on the first and last page of the PDF file respectively. If you wish to place the signature on a specific page, enter the page number as the value for the parameter. For example, *${sign-page##5}*, for placing the signature on the 5th page of the PDF document.

**Sign width and Sign height**

These two parameters are for user to specify the height and width of the signature rectangle size.

**Sign image**

The directory of the image that is to be used with the signature. When no value is entered for this parameter, there will still be a PDF Signature (self-signed) automatically generated.

**Info reason and Info location**

The reason for placing a signature in the PDF document and the position of the reason.

**Certification level**

| | |
|---|---|
| *certificate-not* | has no visible signature in the PDF document. The document is digitally counter signed. |
| *certificate-no-changes* | displays the signature in the PDF document and no changes can be made to the document. |
| *certificate-form-filling* | also displays the signature in the PDF document, but only the filling in of forms, signing and page adding is allowed for the document. |
| *certificate-form-filling-annotation* | displays the signature in the PDF document, but only commenting, form fill-in, signing and page adding is allowed. |

## Print Target

A Print Target allows you to send a report to a named printer. The only option is the name of the printer. If you have multiple alternate printers, you could use a separate target for each printer so that you could control access by different groups. In most cases, the printer names will be fixed and not include substitutions. You can also leave the printer name blank as it will route automatically to the default printer defined on the server.

### Table 1.6. Print Target Configurations

| Name | Value |
|---|---|
| Printer name | Canon iR C3220 PCL5c |

## Repository Target

A Repository Target writes the report to the filesystems. You can identify a target folder in the repository and provided it is writable, files will be written there. This works regardless of whether the target filesystem is of type local, secure or dbfs. You should use Repository Targets when you want to allow users to view the reports through their browser as the repository will automatically update to show the latest files.

During the configuration, the "folder" property must refer the "dir##" parameter to an existing target filesystem in the repository. The "folder##" parameter need not as it will create a new folder after its name if it is not found in the repository.

### Table 1.7. Repository Target Configurations

| Name | Value |
|---|---|
| Folder | ${dir##ElixirSamples}/${folder##repository} |
| Filename | ${filename##Pet_Store_User_Accounts_Report} |
| Overwrite | true |

## S3 Target

A S3 Target allows the report to be transferred to a user's Amazon S3 bucket. The available parameters are bucket name, access key, secret key, the folder name in which to transfer, the expiry time of the file (if any), the filename to name the transferred file and the presigned URL protocol.

### Table 1.8. S3 Target Configurations

| Name | Value |
|---|---|
| Available to Anonymous Users | If enabled, anonymous users can access the file. Else, only those users who have logged into S3 can access the file. |
| Bucket | The name of the S3 bucket in which the file is to be stored. |
| Access Key | Enter your Amazon access key. |
| Secret Key | Enter your Amazon secret key. |
| Folder | The folder in which to store the file. |
| Expiry | The number of days after which the file cannot be accessed anymore. |
| Filename | ${filename##Pet_Store_User_Accounts_Report} |
| Presigned URL Protocol | A signed URL is a URL which is valid for a specific time period. Its purpose is to share the pages that have time sensitive information, or when you want to share a file with specific people alone. Such information becomes invalid once the specified time period has passed. |

## SFTP Target

A SFTP Target allows the report to be transferred to a user's secured FTP Server. The available parameters are user, password, host, port, dir and filename. The port is optional and will default to 22, which is the default SFTP port, if not specified.

The parameter in "dir" property must be an existing directory found in the target ftp server.

### Table 1.9. SFTP Target Configurations

| Name | Value |
|---|---|
| Host | ${sftp host##domain_name.com} |
| Port | ${port##22} |
| Directory | ${dir##dept1}/${folder##Pet_Store} |
| Filename | ${filename##Pet_Store_User_Accounts_Report} |
| User | ${userid##Enter userid to access sftp client} |
| Password | ${password##Enter password to access sftp client} |

## Socket Target

A Socket Target sends the report to a program which is listening, typically on another machine. For example, a program can be written that listens on a company.com port 6000 and writes any data it receives to a database, or to a fax etc. It is up to the receiving program what it does with the data. The server opens a connection to the listening program, using the host and port information required by the socket target and streams the data across the network to the listening socket.

### Table 1.10. Socket Target Configurations

| Name | Value |
|---|---|
| Host | company.com |
| Port | 6000 |

## Unzip Target

This target enables you to specify a file name for unzipping purpose. You can enable/disable this target, and set this target available/unavailable to anonymous users.

## User Home Target

A Repository User Home Target writes the report to the invoking user's folder in the repository. Users can share jobs and they can keep separate output without overwriting one another. In the example, once the user "Jon" signs in to the repository and runs the job, the report will be written to /User/Jon/Pet_Store.

### Table 1.11. Repository User Home Target Configurations

| Name | Value |
|---|---|
| Folder | ${folder##Pet_Store} |
| Filename | ${filename##Pet_Store_User_Accounts_Report} |
| Overwrite | true |

# Configuration

This allows you to perform common configurations that will apply across all Ambience modules. For example, you can set the uniform formats for date, datepicker, time, timepicker and timestamp. When a date is displayed by Java, it will use the /Configuration/format/date/ settings. When JavaScript is used, it will use the /Configuration/format/datepicker settings by default, unless there is a local override. Web-based modules will reflect the changes quickly, while other modules may show the changes after they are restarted.

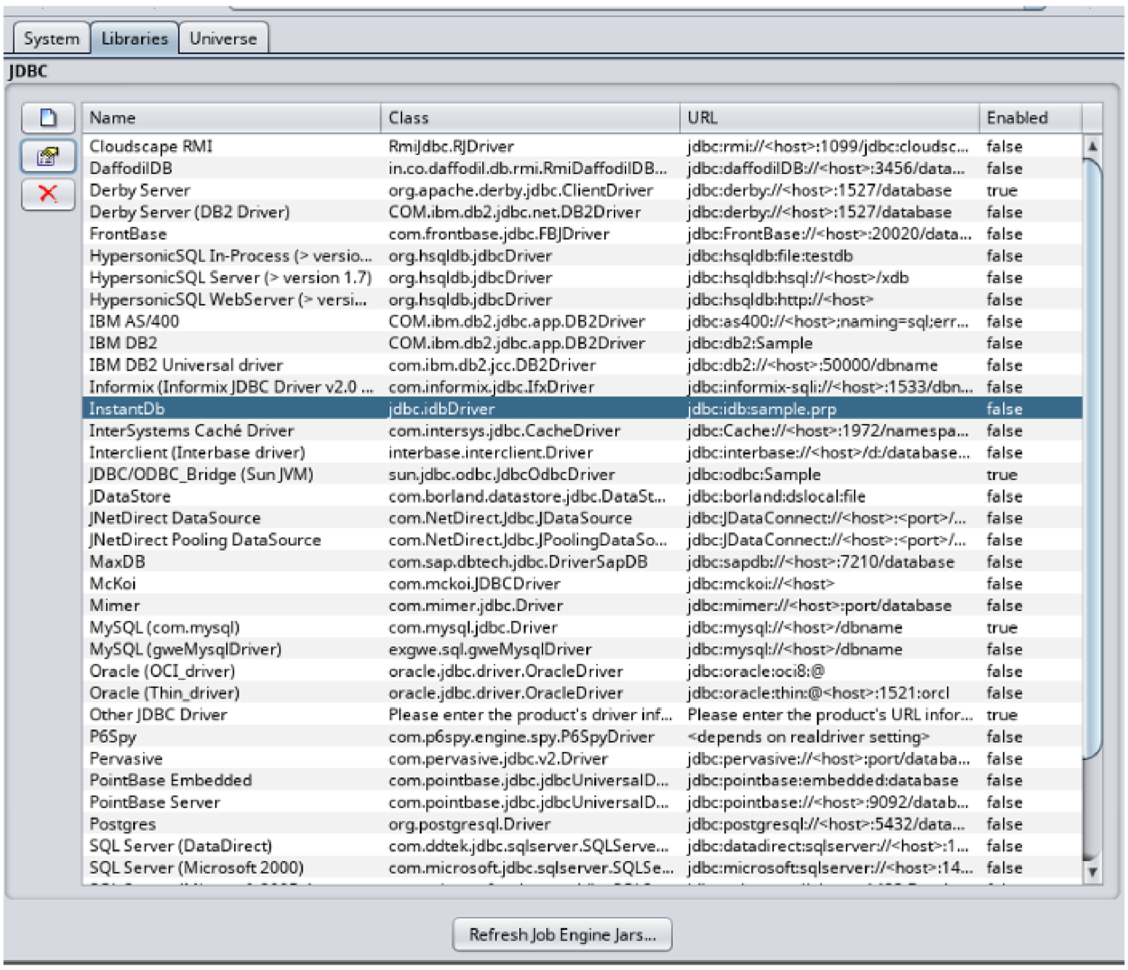
Right-click **Configuration** > **format** > **date**, **datepicker**, **time**, **timepicker** or **timestamp**, the following menu option will display:

- **Edit contents...:** This enables you to edit the contents of the current node. You can also validate the XML.

## Libraries

Libraries include configuration information for a full range of JDBC drivers. Repertoire 9 enables you to configure JDBC drivers more easily. Any JDBC drivers that require native components will usually require the Ambience system to be stopped and restarted. Currently, most drivers are JDBC-Net pure Java drivers or native protocol pure Java drivers. If there is no native code, then you can transfer the JAR files to the job engine machines by putting them in /Public/lib in the Repository Tree and clicking the Refresh Job Engine Jars... button. This will cause the job engines to stop at the end of their current task. The Process Managers will then start them up again, with the updated classpath (including any new JDBC drivers).

**Figure 1.2. Libraries**



| Name | Class | URL | Enabled |
|---|---|---|---|
| Cloudscape RMI | RmiJdbc.RJDriver | jdbc:rmi://<host>:1099/jdbc:cloudsc... | false |
| DaffodilDB | in.co.daffodil.db.rmi.RmiDaffodilDB... | jdbc:daffodilDB://<host>:3456/data... | false |
| Derby Server | org.apache.derby.jdbc.ClientDriver | jdbc:derby://<host>:1527/database | true |
| Derby Server (DB2 Driver) | COM.ibm.db2.jdbc.net.DB2Driver | jdbc:derby://<host>:1527/database | false |
| FrontBase | com.frontbase.jdbc.FBJDriver | jdbc:FrontBase://<host>:20020/data... | false |
| HypersonicSQL In-Process (> versio... | org.hsqldb.jdbcDriver | jdbc:hsqldb:file:testdb | false |
| HypersonicSQL Server (> version 1.7) | org.hsqldb.jdbcDriver | jdbc:hsqldb:hsql://<host>/xdb | false |
| HypersonicSQL WebServer (> versi... | org.hsqldb.jdbcDriver | jdbc:hsqldb:http://<host> | false |
| IBM AS/400 | COM.ibm.db2.jdbc.app.DB2Driver | jdbc:as400://<host>;naming=sql;err... | false |
| IBM DB2 | COM.ibm.db2.jdbc.app.DB2Driver | jdbc:db2:Sample | false |
| IBM DB2 Universal driver | com.ibm.db2.jcc.DB2Driver | jdbc:db2://<host>:50000/dbname | false |
| Informix (Informix JDBC Driver v2.0 ... | com.informix.jdbc.IfxDriver | jdbc:informix-sqli://<host>:1533/dbn... | false |
| InstantDb | jdbc.idbDriver | jdbc:idb:sample.prp | false |
| InterSystems Caché Driver | com.intersys.jdbc.CacheDriver | jdbc:Cache://<host>:1972/namespa... | false |
| Interclient (Interbase driver) | interbase.interclient.Driver | jdbc:interbase://<host>/d:/database... | false |
| JDBC/ODBC_Bridge (Sun JVM) | sun.jdbc.odbc.JdbcOdbcDriver | jdbc:odbc:Sample | true |
| JDataStore | com.borland.datastore.jdbc.DataSt... | jdbc:borland:dslocal:file | false |
| JNetDirect DataSource | com.NetDirect.Jdbc.JDataSource | jdbc:JDataConnect://<host>:<port>/... | false |
| JNetDirect Pooling DataSource | com.NetDirect.Jdbc.JPoolingDataSo... | jdbc:JDataConnect://<host>:<port>/... | false |
| MaxDB | com.sap.dbtech.jdbc.DriverSapDB | jdbc:sapdb://<host>:7210/database | false |
| McKoi | com.mckoi.JDBCDriver | jdbc:mckoi://<host> | false |
| Mimer | com.mimer.jdbc.Driver | jdbc:mimer://<host>:port/database | false |
| MySQL (com.mysql) | com.mysql.jdbc.Driver | jdbc:mysql://<host>/dbname | true |
| MySQL (gweMysqlDriver) | exgwe.sql.gweMysqlDriver | jdbc:mysql://<host>/dbname | false |
| Oracle (OCI_driver) | oracle.jdbc.driver.OracleDriver | jdbc:oracle:oci8:@ | false |
| Oracle (Thin_driver) | oracle.jdbc.driver.OracleDriver | jdbc:oracle:thin:@<host>:1521:orcl | false |
| Other JDBC Driver | Please enter the product's driver inf... | Please enter the product's URL infor... | true |
| P6Spy | com.p6spy.engine.spy.P6SpyDriver | <depends on realdriver setting> | false |
| Pervasive | com.pervasive.jdbc.v2.Driver | jdbc:pervasive://<host>:port/databa... | false |
| PointBase Embedded | com.pointbase.jdbc.jdbcUniversalD... | jdbc:pointbase:embedded:database | false |
| PointBase Server | com.pointbase.jdbc.jdbcUniversalD... | jdbc:pointbase://<host>:9092/datab... | false |
| Postgres | org.postgresql.Driver | jdbc:postgresql://<host>:5432/data... | false |
| SQL Server (DataDirect) | com.ddtek.jdbc.sqlserver.SQLServe... | jdbc:datadirect:sqlserver://<host>:1... | false |
| SQL Server (Microsoft 2000) | com.microsoft.jdbc.sqlserver.SQLSe... | jdbc:microsoft:sqlserver://<host>:14... | false |

## Configuring JDBC Drivers

To illustrate the JDBC driver configuration procedure, here are the steps:

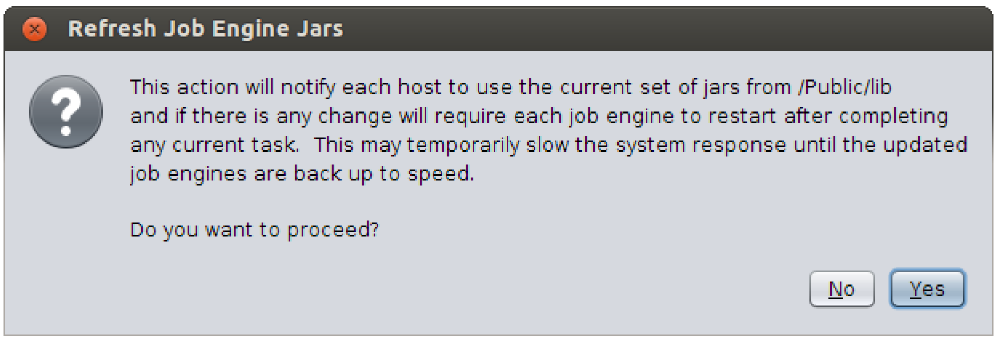
1. Click the Administration Tools... button on the workspace toolbar. Click the Libraries tab. The JDBC drivers display.

2. Find and select the desired driver from the list. Click the Edit button. The JDBC Driver dialog displays. Fill in the host machine name to replace <host> in the URL field. Leave the class information as it is.

3. Select the Enabled checkbox and click OK.

4. If there is no native code for the driver, download the driver’s JAR file from the vendor's Web site. Copy the driver's JAR file into the /Public/lib folder in the Repository Tree. Click the Refresh Job Engine Jars... button. A message may pop up, as shown in Figure 1.3, “Refresh Job Engine Jars”:

**Figure 1.3. Refresh Job Engine Jars**



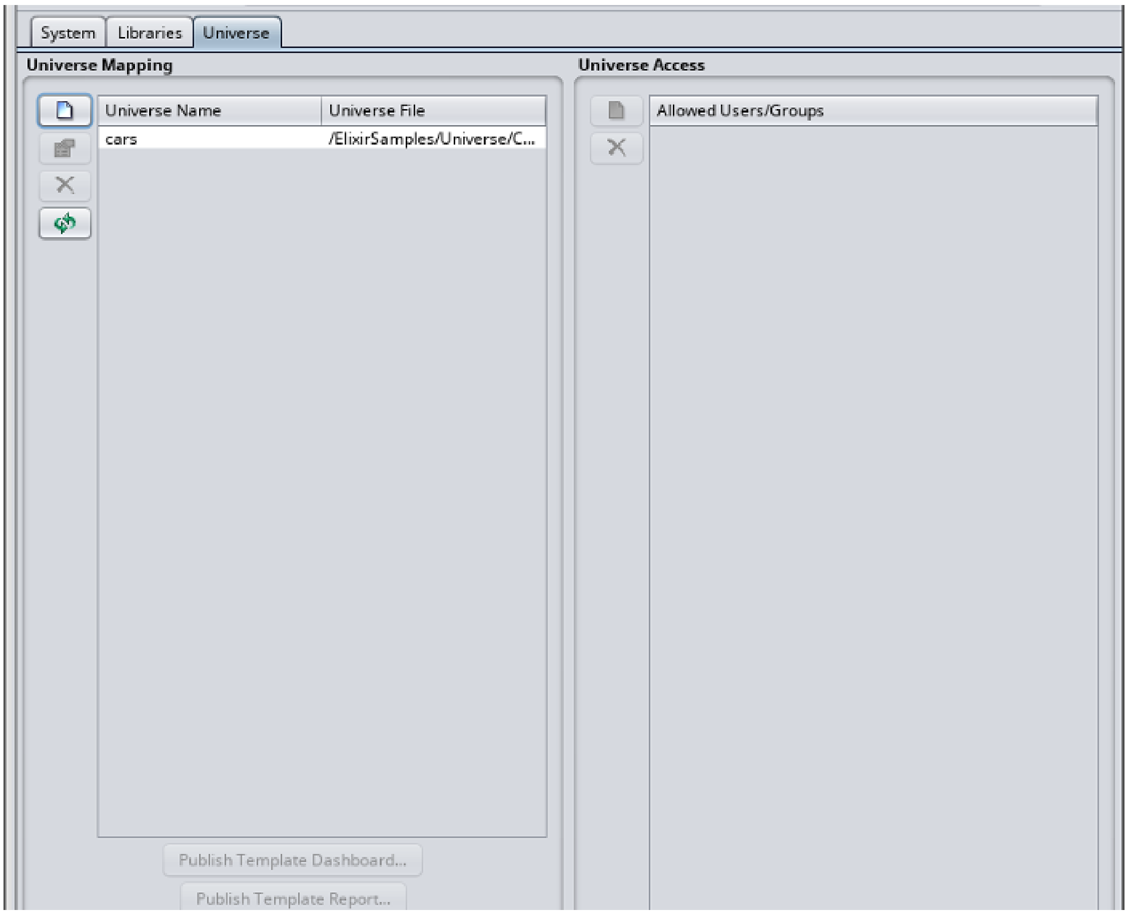
This action will notify each host to use the current set of jars from /Public/lib and if there is any change will require each job engine to restart after completing any current task. This may temporarily slow the system response until the updated job engines are back up to speed.

Do you want to proceed?

## Universe

Through the Universe interface, you can map a Universe name with its path, publish template dashboard or report, and define which users and groups will be able to access the Universe.

**Figure 1.4. Universe**



| Universe Name | Universe File |
|---|---|
| cars | /ElixirSamples/Universe/C... |

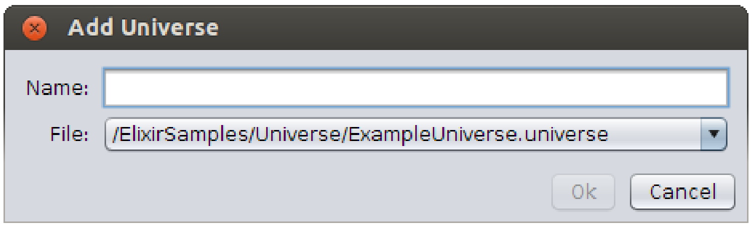
| Allowed Users/Groups |
|---|

## Universe Mapping

A Universe can exist anywhere in the Repository, but are only accessible if its path is mapped to a Universe name. Only administrators can set or alter the Universe mapping.

Complete the following steps to map a Universe:

**Figure 1.5. Add Universe**



Name:

File: /ElixirSamples/Universe/ExampleUniverse.universe

1. In the Administration Tools, click the **Universe** tab.
2. In the Universe Mapping pane, click the **Add Universe** button.
3. In the Add Universe window, enter a name and select the desired universe from the File list.
4. Click OK. The Universe name and path are successfully mapped.

### Note

Unwanted Universe(s) can be easily removed from the Universe Mapping pane by a mouse click.

For a Repository Universe, once its name and path have been mapped, the name will display as an option within Ad-hoc Dashboard. After setting the Primary Key, you can view the Ad-hoc Dashboard via the following URL. Fill in your domain, host machine and dashboard names to proceed:

```
http://<host>:8080/elx/do/<domain>/ei/accolades/
dashboard/Sample/
Demo/Ad%20Hoc/Dashboard/<dashboard>.dashboard?mode=file
```

You also have the option to publish template dashboard or report, as shown in the figures below:

**Figure 1.6. Publish Template Dashboard**

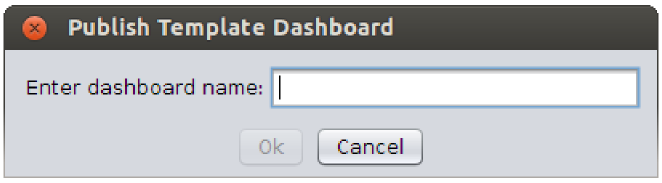
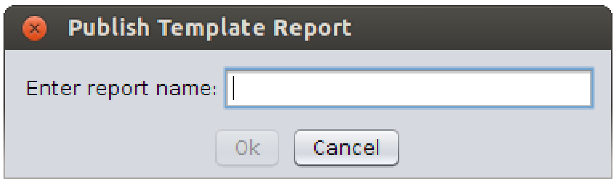


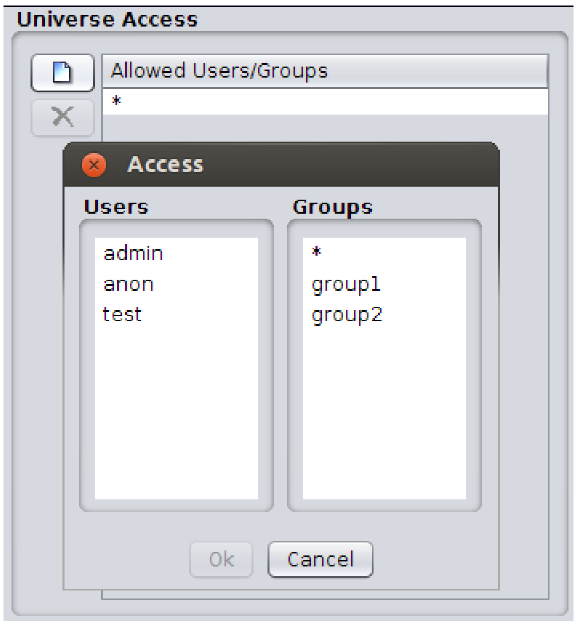
**Figure 1.7. Publish Template Report**



## Universe Access

Complete the following steps to control the access to the Universe:

**Figure 1.8. Universe Access**



1. In the Administration Tools, click the **Universe** tab.
2. In the Universe Access pane, click the **Add Access** button.
3. In the Access window, select the desired users/groups from the list.
4. Click OK. The access rights are successfully assigned.

Only users who can read or write the Universe file can open it to view and edit the contents. However, those users in the Access list can only use the Universe, without being able to open or edit the Universe file. The Universe service will access the file for those users.

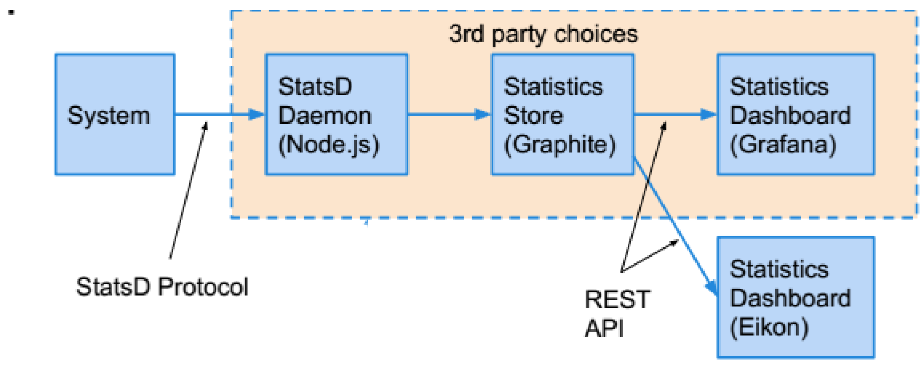
# Chapter 2
# System Monitoring With StatsD

## Overview of StatsD

You can install StatsD to monitor the performance of your system in real time.

Figure 2.1, “StatsD Monitoring Protocol” shows a simple flow chart that depicts the StatsD monitoring protocol.

**Figure 2.1. StatsD Monitoring Protocol**


3rd party choices
System
StatsD Daemon (Node.js)
Statistics Store (Graphite)
Statistics Dashboard (Grafana)
Statistics Dashboard (Eikon)
StatsD Protocol
REST API

StatsD is a fast, lightweight system monitor and can use many third party tools for storing the statistics it collects (e.g. Graphite) and for displaying the statistics as graphs (for e.g. using Grafana).

Examples of metrics that StatsD can monitor include:

- Number of report pages rendered
- Number of JDBC rows accessed
- Servlet response time
- Number of Job Engines
- System RAM and CPU loads

For more information on StatsD, see https://github.com/etsy/statsd [https://github.com/etsy/statsd/].

## Enabling StatsD

To enable StatsD monitoring, add the following line to `etc/application.conf`:

elixir.statsd.enabled = true

## Viewing Your System's Statistics

If you do not already have a StatsD monitoring solution, you can install a pre-packaged, easy to use, uncomplicated docker image that contains StatsD, Graphite, Grafana and a Kamon Dashboard, all set up, from https://github.com/kamon-io/docker-grafana-graphite.

If you want to run this image, ensure that your system is 64-bit.

To add the statistical parameters to the dashboard, and to view the graphs:

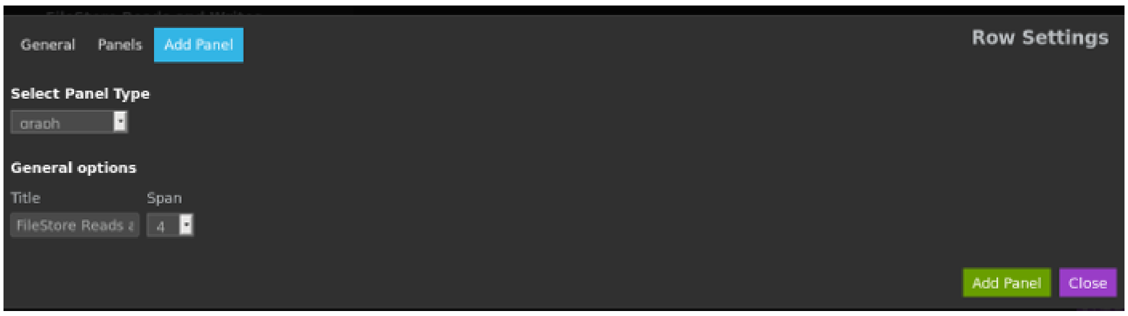
1. Start the docker image, open your browser and navigate to `localhost` to load the Grafana home page.

2. Create a new dashboard.

3. Click the `Configure Row` icon and then add a new graph panel.

4. Add a title for the panel as shown in Figure 2.2, “Adding a Graph Panel”.

**Figure 2.2. Adding a Graph Panel**



Row Settings

General Panels Add Panel

Select Panel Type: graph

General options

Title: FileStore Reads a

Span: 4

5. Click **Add Panel** to add the panel to the dashboard.

6. Click **Close** to return to the dashboard.

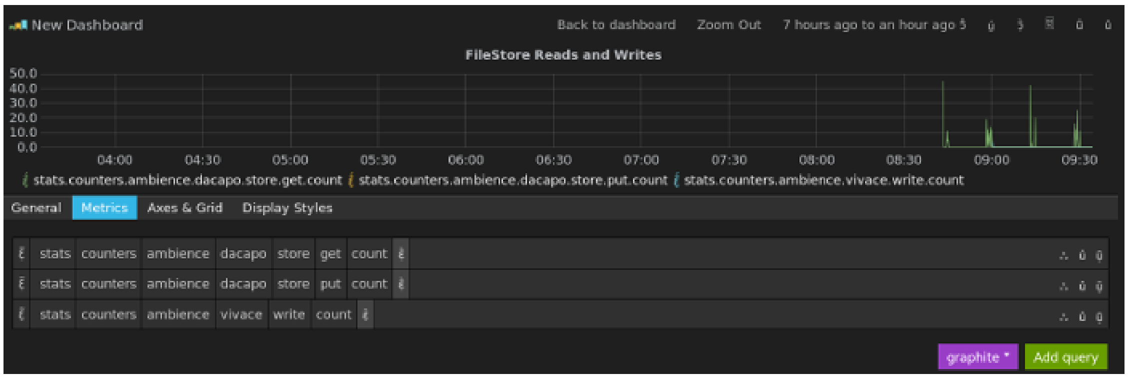
7. Click the `Expand Row` icon to expand your newly added row.

8. Click the row title and select `Edit`.

9. From the `Metrics` tab, add the statistic to be retrieved and displayed, as shown by the example in Figure 2.3, “Adding the Metrics”.

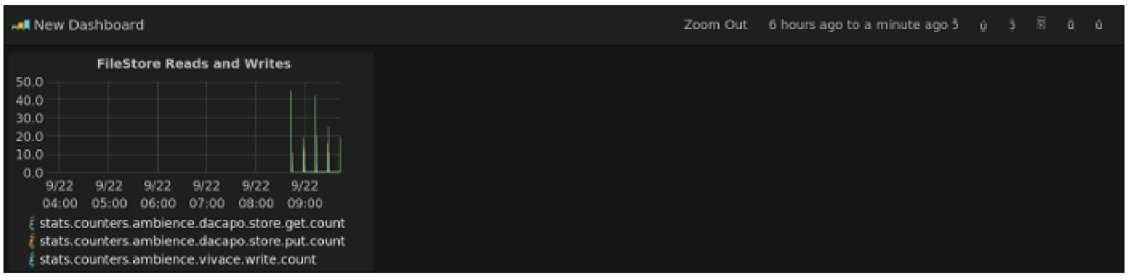
   Note that as you select each component of the statistic, the system filters and displays only the appropriate choices for the next component.

**Figure 2.3. Adding the Metrics**


New Dashboard
Back to dashboard
Zoom Out
7 hours ago to an hour ago
FileStore Reads and Writes
50.0
40.0
30.0
20.0
10.0
0.0
04:00
04:30
05:00
05:30
06:00
06:30
07:00
07:30
08:00
08:30
09:00
09:30
stats.counters.ambience.dacapo.store.get.count
stats.counters.ambience.dacapo.store.put.count
stats.counters.ambience.vivace.write.count
General
Metrics
Axes & Grid
Display Styles
stats counters ambience dacapo store get count
stats counters ambience dacapo store put count
stats counters ambience vivace write count
graphite
Add query

10. Click the `Back to Dashboard` link on the top, to return to your dashboard. The added statistic is displayed, as shown by the example in Figure 2.4, "Displaying the Added Statistic".

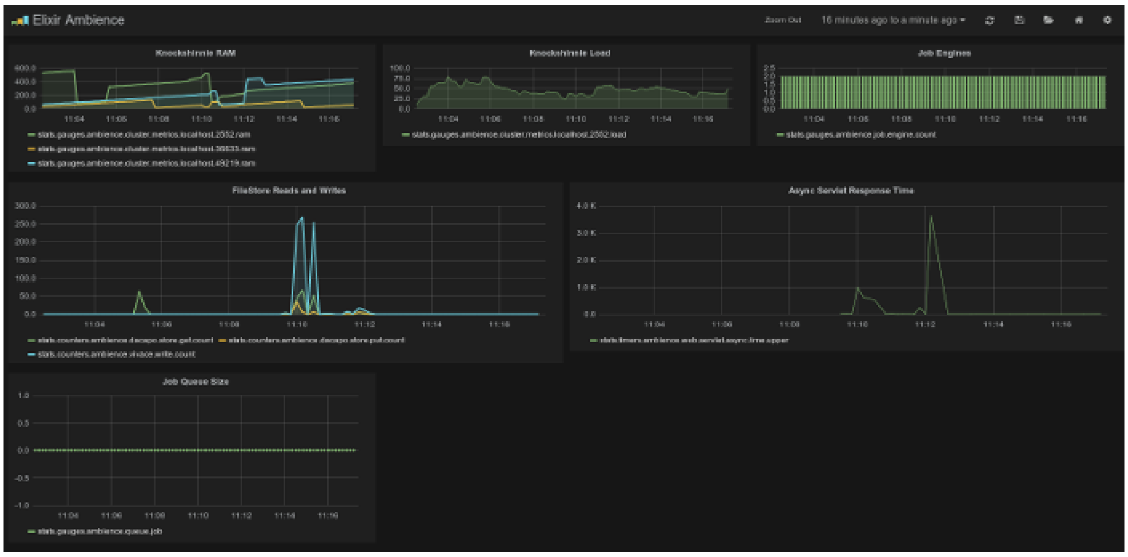
**Figure 2.4. Displaying the Added Statistic**


New Dashboard
Zoom Out
6 hours ago to a minute ago
FileStore Reads and Writes
50.0
40.0
30.0
20.0
10.0
0.0
9/22 04:00
9/22 05:00
9/22 06:00
9/22 07:00
9/22 08:00
9/22 09:00
stats.counters.ambience.dacapo.store.get.count
stats.counters.ambience.dacapo.store.put.count
stats.counters.ambience.vivace.write.count

11. Click **Add a Row** and repeat the process to add a new statistic to the dashboard.

An example of a completed dashboard is shown in Figure 2.5, "Completed Dashboard".

**Figure 2.5. Completed Dashboard**


Elixir Ambience
Zoom Out
16 minutes ago to a minute ago
Knockshinnie RAM
stats.gauges.ambience.cluster.metrics.localhost.2552.ram
stats.gauges.ambience.cluster.metrics.localhost.39533.ram
stats.gauges.ambience.cluster.metrics.localhost.49219.ram
Knockshinnie Load
stats.gauges.ambience.cluster.metrics.localhost.2552.load
Job Engines
stats.gauges.ambience.job.engine.count
FileStore Reads and Writes
stats.counters.ambience.dacapo.store.get.count
stats.counters.ambience.dacapo.store.put.count
stats.counters.ambience.vivace.write.count
Async Servlet Response Time
stats.timers.ambience.web.servlet.async.time.upper
Job Queue Size
stats.gauges.ambience.queue.job

## Statistics Listing

The following table lists the performance metrics that can be monitored.

**Table 2.1. Performance Metrics**

| Parameter | Description |
|---|---|
| stats.counters.ambience.dacapo.indexer.file | Counts the number of files that have been indexed. |
| stats.counters.ambience.dacapo.store.get | Counts the number of files that `dacapo` has read from the underlying file store. |
| stats.counters.ambience.dacapo.store.put | Counts the number of files that `dacapo` has written to the underlying file store. |
| stats.counters.ambience.vivace.write.count | Counts the number of files that `vivace` has written to the disk. |
| stats.counters.ambience.vivace.write.rate | Denotes the rate of change. For example, 10 nodes per second. |
| stats.gauges.ambience.cluster.metrics.localhost.[numbers].load | Measures the CPU load of the machine running JVM on port [numbers]. |
| stats.gauges.ambience.cluster.metrics.localhost.[numbers].ram | Measures the amount of RAM used by the machine running JVM on port [numbers]. |
| stats.timers.ambience.job.infowatcher.time.[metric] | Measures the time taken for the job engine to process each request. |

Counts are not cumulative - for example if six nodes are written in 10 seconds and then eight in the next ten seconds, you will get a chart showing 6, 8. Ten seconds is the default StatsD count interval.