

Elixir Repository

Release 3.5.0



Elixir Technology Pte Ltd

Elixir Repository: Release 3.5.0

Elixir Technology Pte Ltd

Published 2014

Copyright © 2014 Elixir Technology Pte Ltd

All rights reserved.

Java is a trademark or registered trademark of Oracle Corporation in the United States and other countries. Microsoft and Windows are trademarks of Microsoft Corporation.

Table of Contents

1. About Elixir Repository	1
Overview	1
Starting the programs	1
Common Ambience File Types	2
2. Using Elixir Repository	3
Working with filesystems	3
Add	3
Export...	3
Import...	3
Backup filesystem	4
Restore filesystem	5
Retain versions...	5
Working with sub-directories	5
Add	5
Export...	6
Import...	6
Delete Children	6
Working with files	6
Open External...	6
Save Local...	6
Recent Files	6
Properties	7
Tags	7
Versions	8
Save a version	8
Delete a version	8
Revert to a version	9
Access Control	9
Assign read access right	9
Assign write access right	9
Remove write access right	9
Remove read access right	9
Clone access right	9
Recycle Bin	10
Delete	10
Restore	10
Delete permanently	10
Empty Recycle Bin	10
File Stores	10
Create a filesystem	10
Create a filesystem snapshot	10
Compact a store	11
Repository Synchronization	11

List of Figures

2.1. Incremental Backup from Day 2 to Day 3	4
2.2. Incremental Backup from Day 1 to Day 3	4
2.3. Full + Incremental Backup	5

List of Tables

1.1. Default Usernames and Passwords	1
2.1. Before and After Import	3
2.2. File Properties	7
2.3. Tags	7
2.4. Keywords and Results	8
2.5. Versions	8

Chapter 1

About Elixir Repository

Overview

The Repository is a central storing hub for information that can be accessed and modified by authenticated users and system engines, regardless of their location. It is a valuable tool for Ambience to keep records of all versions of files, which are modified by different parties. The Repository enables you to backup or restore filesystems, export or import directories, retain filesystem versions, control file versions, control the access rights and more. These features empower the authenticated users and system engines to use the complete features of Ambience without complex hardware configuration. It allows different parties to edit and share the same set of files without being conflicted.

The Repository feature is available for use both in the Repository tool and the Repertoire tool. The Repository feature in the Repertoire tool allows you to create and edit Connection Pool, DataSource, JavaScript, HTML and a few more supported types of file within Repertoire. When you need to create new files and conveniently edit them within one tool, use the Repository feature in the Repertoire tool. Differently, the individual Repository allows you to store files and directories, instead of directly editing them. You will be able to open files externally, which is the same behaviour with opening unsupported types of file using the Repertoire tool. When you only need to save a copy of the files and directories, use the individual Repository.

Using the Repository instead of a database as the storing hub brings the following benefits:

- The user will be able to use the same scheduled backup/restore mechanism as the rest of the Repository.
- Version control makes it easier for the user to track changes.
- Tagging enables the user to differentiate files for easier recognition and searching.
- Searching makes it possible to find out useful information more quickly and accurately.

Starting the programs

To start the Repertoire and Repository programs after installation, open the `/opt/elixir/desktop/` directory and double-click the icons.

Alternatively, you can start Repertoire and Repository with the following command lines:

```
/opt/elixir/repertoire/elx-repertoire.sh  
/opt/elixir/repository-gui/elx-repository-gui.sh
```

The default usernames and passwords are shown in the following table, which can be changed to your own. Please note that if you forget the altered values, they cannot be recovered.

Table 1.1. Default Usernames and Passwords

Program	Username	Password
Repertoire and Repository	admin	sa

Common Ambience File Types

Connection Pool: A Connection Pool allows connections to JDBC databases and connection properties to be shared amongst multiple datasources. Connection Pools are described in the *Elixir Data Designer User Manual*.

CSS: CSS is used to create and edit a Cascading Style Sheets file. It can define the styles of a report, including background, fonts, header, footer and more.

DataSource: The DataSource is the principal building block in the Elixir Ambience solution. External data may be wrapped as a DataSource and multiple DataSources may be merged and processed by a Composite DataSource. Each kind of DataSource is described in the *Elixir Data Designer User Manual*.

HTML: HTML is used to create and edit a Hypertext Markup Language file. HTML can be imported into Dashboard and embedded in Frame or Page.

JavaScript: Elixir Ambience is an extensible tool, allowing users familiar with Java or JavaScript to integrate the tool with their code or customize the tool to meet their own requirements. Many parts of the tool allow JavaScript to be embedded, however if the same scripts are required in multiple locations it is preferable to create a common JavaScript file in the repository and then to import it into each DataSource, Report or Dashboard that requires it. This editor is discussed further in *Elixir Repertoire User Manual*.

Job: Job is used to add tasks that are intended to run together.

Map: Map is usually created to represent geographic areas and adds new dimension to data visualisation. More information about Map can be found in *Elixir Dashboard User Manual*.

Perspective: A Perspective holds the definition for a dashboard - a set of interactive views, including tables, charts, reports and data cubes. Dashboards are stored in Perspective Markup Language (an XML syntax) with a file extension .pml. Perspectives are described in the *Elixir Dashboard Designer User Manual*.

Report Template: A Report Template holds the specification for a report, including the layout of report components and reference to the datasources needed to provide the data during rendering. Reports are stored in Report Markup Language (an XML syntax) with a file extension .rml. Report Templates are described in the *Elixir Report Designer User Manual*.

Safe: A Safe is an encrypted text file, used for holding valuable text-based information, such as database passwords etc. The Safe is described in *Elixir Repertoire User Manual*.

Text: A simple text editor/viewer is included for reviewing log files, CSV data etc.

Universe: A Universe is an interface to data. The user can add more than one Universe to handle higher-volume data processing. Universes hold many tables. There are two types of Universes: Repository and JDBC. In the Repository Universe, each table maps to a DataSource. Universes can exist anywhere in the Repository, but are only accessible when mapped to a Universe name. All modules that use Universe reference the universe name instead of the path. This enables you to easily switch between different Universes by re-mapping. Universe mapping is a feature only available for administrators.

XML: XML is used to create and edit an Extensible Markup Language file. It can be imported into DataSources and pass the data.

Chapter 2

Using Elixir Repository

Working with filesystems

Add

Using this menu item several kinds of file or a folder can be added. Each will invoke a specialized wizard to guide you through the creation process. You can add different types of file into a filesystem.

The actual set of file types that each installation supports will vary with the tool options and configuration chosen. For a review of common file types, please see [the section called “Common Ambience File Types”](#).

Important: To be able to add files, you have to use the Repository in the Repertoire tool.

Export...

Instead of manually transferring data every time after resetting the Repository to the default, you can use the export and import tools.

The export action will copy only the latest version of all files referenced. Filesystems can be exported from the Repository and saved as a TGZ or ZIP file. To export a filesystem in the Repository to the designated location, right-click the filesystem and select **Export...** Switch to the target location, type the TGZ or ZIP file name and click **Save**.

Import...

The import action will create a new version for the referenced files with the same name, if the contents are different from the current version. To import the designated TGZ or ZIP file to a filesystem in the Repository, right-click the target filesystem and select **Import...** Switch to the location where the TGZ or ZIP file is stored, select it and click **Open**.

If you are importing a TGZ or ZIP file to replace an existing directory, referenced files might be overwritten. You can trace an overwritten file in the version history. Refer to the following table as an example. Before the import action, Folder X has two files A and B. The TGZ or ZIP file contains information on a previous status of Folder X, including File A* with different contents, as well as files C and D. After the import action, you will find that Folder X contains File A* with a new version created for it. The previous File A is also recorded in the version history. Although File B is not contained in the TGZ or ZIP file, it remains in Folder X. Files C and D are incrementally added.

Table 2.1. Before and After Import

Before Import	Contents in the TGZ or ZIP file	After Import
Folder X: File A, File B	Folder X: File A*, File C, File D	Folder X: File A*, File B, File C, File D

Backup filesystem

The Repository in Ambience is different from the Repository in earlier versions. It allows you to store and edit the complete set of filesystems at the server end. However, for archiving purpose, you have the option to back up the filesystems either fully or incrementally on the designated location. When you back up the filesystem, all versions of the files referenced will be backed up. The backup file format should be TGZ or ZIP.

Full Backup: The entire filesystem will be backed up to the designated location. Right-click the filesystem and select **Backup filesystem**. Select **Full...**, which indicates everything in the current filesystem will be backed up. Specify the target folder URL in the Backup dialog and click **Backup**.

Incremental Backup: Only the changes in the filesystem, compared to a backup action on a start date, will be backed up to the designated location. Right-click the filesystem and select **Backup filesystem**. Select **Incremental...**, which indicates only the changes in the current filesystem will be backed up. Specify the date to start from and the target folder URL in the Backup dialog and click **Backup**.

Note

If you select a different date to start from, you will receive different incremental backup files. See the following figures for comparison. When you are performing an incremental backup on Day 3:

- If the start date is Day 2, only the changes between Day 2 and Day 3 will be reflected in the incremental backup file.
- If the start date is Day 1, changes between Day 1 and Day 3 will be reflected in the incremental backup file.
- If there are multiple versions of referenced files on the same start date, only the latest version will be backed up.

Figure 2.1. Incremental Backup from Day 2 to Day 3

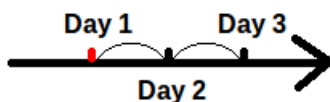
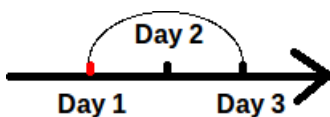
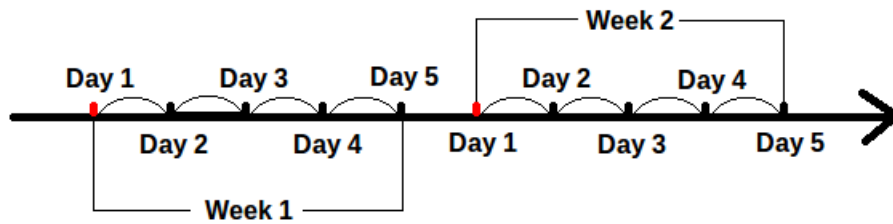


Figure 2.2. Incremental Backup from Day 1 to Day 3



You have the option to always use **Full Backup** for simplicity, or use **Full + Incremental Backup**. It is recommended that you take the **Full + Incremental Backup** approach and set intervals to regularly backup the filesystem. If you have sufficient disk space, the full backup interval can be shorter, for example once a week, which makes incremental backups easier and faster. If you have moderate disk space, the full backup interval can be longer, for example once a month, which saves disk space but needs more efforts on incremental backups. See [Figure 2.3, “Full + Incremental Backup”](#) for example. Company A performs a full backup on the first day of each week (Day 1). The incremental backup interval depends on the full backup interval and the frequency of changes that are happening to the filesystem. Company A decides to perform an incremental backup at the end of each work day (except Day 1).

Figure 2.3. Full + Incremental Backup

Restore filesystem

You can restore the filesystem from one or more backup files. When you restore the filesystem, all version information backed up in the TGZ or ZIP file will be restored as well.

Full Restore: The entire filesystem will be restored from the full backup file. To fully restore the backup file in TGZ or ZIP format from the designated location, right-click the filesystem and select **Restore filesystem**. Select the **Full...** option. Specify the source file URL in the Restore dialog and click **Restore**.

Incremental Restore: Only the changes in the filesystem will be restored from the incremental backup file. To incrementally restore the backup file in TGZ or ZIP format from the designated location, right-click the filesystem and select **Restore filesystem**. Select the **Incremental...** option. Specify the source file URL in the Restore dialog and click **Restore**.

According to your backup approach, you have the option to always use **Full Restore**, or use **Full + Incremental Restore**. Take Company A for example and refer to [Figure 2.3, “Full + Incremental Backup”](#). Assume that the designated filesystem is crashed on Day 5. How does Company A restore to the status on Day 4? The correct steps include performing a full restore from the backup file on Day 1 of the week, and then performing several incremental restores from Day 2 to Day 4 backup files. Please note that the incremental restores must follow the chronological order. If the order is incorrect, the changes in the filesystem might be messed up.

Retain versions...

You have the option to set how many versions Ambience will keep in track for a filesystem, either within a certain number of version count or recent days. This feature makes it more convenient for you to personalize the limit of version numbers, instead of keeping a huge number of unnecessary versions. It reduces the data storage at the server end and reduces the time to look into the entire version history. To customize the version number limit, right-click the desired filesystem in the Repository. Select **Retain Versions...**, and the Retain Versions dialog opens. In the dropdown list, select **count** or **days**. Specify the value in the field. For example, setting the **count** value to “0” means keeping only the current version, while setting the **days** value to “100” means keeping the versions within the recent 100 days.

Working with sub-directories

Add

Using this menu item several kinds of file or a folder can be added. Each will invoke a specialized wizard to guide you through the creation process. You can add the same types of file with those for a filesystem.

Important: To be able to add files, you have to use the Repository in the Repertoire tool.

Export...

The export and import tools support not only the entire filesystem, but also any sub-directory.

The export action will copy only the latest version of all files referenced. Sub-directories can be exported from the Repository and saved as a TGZ or ZIP file. To export a sub-directory in the Repository to the designated location, right-click the sub-directory and select **Export...** Switch to the target location, type the TGZ or ZIP file name and click **Save**.

Import...

The import action will create a new version for the referenced files with the same name, if the contents are different from the current version. To import the designated TGZ or ZIP file to a sub-directory in the Repository, right-click the target sub-directory and select **Import...** Switch to the location where the TGZ or ZIP file is stored, select it and click **Open**.

Delete Children

The Delete Children action provides the option to select exactly which children you want to delete or keep. In the Delete Children dialog, select **All**, **None** or **Invert** to specify which files or folders will be affected. Click **Delete selected files** to delete those you have chosen.

Working with files

Open External...

A copy of this file will be externally opened by a default program according to the file type if this menu item is selected.

Save Local...

While using Ambience, you need to transfer files between the designated location and the Repository. For example, setting up your Repository requires adding files into the tree structure, and keeping a local copy of a file requires saving it to a location.

Save a file: To save a file from the Repository to the designated location, right-click and select **Save Local...** Switch to the target directory, specify the file name and click **Save**.

Add a file: To add a file from the designated location, drag and drop the file to the Repository Tree.

Recent Files

Ambience enables you to quickly locate a recent file without manually unfolding the tree and looking into the nodes, which might be time-consuming.

Locate a recent file: Click the **Recent Files** tab, right-click the target file and select **Show in Repository Tree**. The target file will be automatically located.

Clear all recent files: You have the option to delete the file browsing history. Click the **Recent Files** tab, right-click anywhere in the tab page and select **Clear All Recent Files**. The file browsing history will be erased, while the files still remain in the store.

Properties

The properties of files in the Repository Tree are shown in the **Properties** tab, including file name, file location, file size, last access time and last modification time. See the following table for an example:

Table 2.2. File Properties

Name	Value
name	Sales.ds
folder	/User/Brian
size	3297 bytes
access	2013-05-20 17:05:21
modified	2013-05-19 09:32:46

Tags

Tagging a file helps you differentiate it from other similar files and search for it according to customized keywords.

Tag a file: Select the desired file in the tree view of the Repository, click the **Tags** tab and type the tag name in the text field below the tree view. A tag can begin with **<username>#**, followed by text without any space. These tags can only be added and removed by the specific user. For example, **jon#todo** is a private tag which only jon can add and remove. Similarly, **sales#2013** is a private tag which anyone in the sales team can add and remove. A tag such as **important** can be added and removed by anyone. Everyone can see the tags even if tags are private, and tags that do not begin with **<username>#** can be edited by anyone. With private tags, however, the ability to add and remove is restricted to those with matching credentials.

Note

Directories cannot be tagged.

Search for a file: Type the keywords in the text field above the tree view. Keywords are case-sensitive, and wildcard characters are not supported. If you need to specify a certain file type, add the filter **“ext:xxx”** at the end of the keywords separated by a space, where **“xxx”** stands for the file extension, such as **“ds”**, **“rml”**, **“txt”** and more. Please note that the filter **“ext:xxx”** must be used with some keywords instead of being used alone, because it means searching for all files in that type and there might be a huge number of files to process. You can add multiple **“ext:xxx”** to search for files that match any of the given extensions. Click the **Search...** button. The file name and last modification time will be shown if any file is found. When you select a search result, the tab pages at the bottom still work, such as **Versions**, **Tags** and **Properties**. Click the **Refresh Tree** button to return to the tree view.

The following tables show an example, in which you tag the files first, and then type different keywords to search. You will receive different results depending on the tags and keywords.

Table 2.3. Tags

File	File extension	Tag
A	txt	first second third
B	ds	first sample
C	rml	second sample

Table 2.4. Keywords and Results

Keyword	Result
first	A, B
second	A, C
sample	B, C
sample ext:rml	C

Versions

Ambience regards the change of contents as the signal of a new version. In the meantime, change of access rights or tags is regarded as the signal of the same version. Please note that if you log in with a new user name and choose to transfer a copy of file into the Repository, it is regarded as a new version even if the contents are the same with the previous version. The only difference between these two versions is the ownership, as shown in the third record in the following table. This feature is designed for recording the complete version history for all the authenticated users and system engines.

Table 2.5. Versions

User	Length (Bytes)	Timestamp
admin	0	2011-09-01 16:51:43
admin	50	2011-09-01 16:52:06
test	50	2011-09-01 16:55:37
admin	70*	2011-09-01 16:58:15
admin	10	2011-09-01 17:00:52
admin	70	2011-09-01 17:01:29

*Right-click and revert to this version.

Before viewing the previous versions, make sure that you are granted the read access right. It helps you to prevent the outlet of important information. Similarly, you need to be granted the write access right before editing the files. For more details, refer to the **Controlling the access rights** section. Ambience enables you to edit the files directly. All you need are a properly configured host machine and a network connection.

Save a version

You have the option to save either the current or a previous version, which enables you to look into that version for details. To save a version on the host machine, click the **Versions** tab, right-click the version and select **Save Local**.

Delete a version

If you want to remove unnecessary versions from the records, select the file, click the **Versions** tab, right-click those versions and select **Delete...** Deleting the unnecessary versions reduces the time and effort of managing a huge number of version records.

If you delete a file, its version records are still in the store. When you restore a deleted file from the Recycle Bin, the previous versions will also be restored. If you permanently delete a file, the version records will be cleared. However, be cautious to use this feature and make sure you do not delete the versions with significant changes.

Revert to a version

One important purpose of recording the versions is that you can date back to any previous version and revert to it. Undoing the changes you don't need is extremely useful for changes that happen back and forth, which saves a lot of repeated efforts. By clicking the **Versions** tab, right-clicking a previous version and selecting **Revert...**, the previous version becomes the most current version. To better serve your tracking needs, the more recent versions are still kept in the records. [Table 2.5, "Versions"](#) shows what happens after you choose to revert. Please note that the record in bold indicates the result of the revert action.

Access Control

Because the files in the Repository are shared among a variety of different users and system engines, the security of information must be protected. To ease your concern, the Repository allows you to assign read and write access rights in a convenient way and keep the confidential information in proper place.

Assign read access right

To assign read access right to another user account, be sure that you have write access right to the target file or directory. Right-click the target file or directory, click the **Access Control** tab, click the **Add** button. The Clone Access Rights dialog opens, indicating that these access rights are defined at an upper-level directory. Click **Yes** to clone them to the current file for editing, or click **No** to select the path where they are defined. In the Add Entry dialog, select the **User** or **Group** radio button according to your needs. Select a user name or group name and click **OK**.

Assign write access right

Repeat the steps of assigning read access right, but select the **Allow write access as well** check box before clicking **OK**. Alternatively, you can assign read access right, select the user name or group name from the **Reader** list in the **Access Control** tab, and then select the check box in the **Writer** list.

The steps above apply to users and groups that already exist. To create users and groups and configure their properties, refer to the *Working with the Domain Manager*.

Remove write access right

Select the user name or group name from the **Reader** list in the **Access Control** tab, and then deselect the check box in the **Writer** list.

Remove read access right

Select the user name or group name from the **Reader** list in the **Access Control** tab, and then click the **Remove** button.

Clone access right

In a filesystem where you select one access entry and click "-", you will receive a message, saying that these access rights are defined at certain directory level, which is the parent access control list; click **yes** to clone them to the current file for editing, or **no** to select the path where they are defined.

If you click **yes**, you will get a copy of the permissions at the directory level where these access rights are defined. If you click **no**, you will revert back to the parent access control list.

Recycle Bin

Recycle Bin in the Repository temporarily stores removed items, allowing you to either restore them to the original place or permanently discard them.

Delete

To delete a redundant file or directory, locate it in the Repository Tree, right-click and select **Delete**. This operation removes the file or directory from the tree and places a copy in the Recycle Bin.

Restore

In case a file was accidentally deleted, you can always restore it if the file is still in the Recycle Bin. Right-click its record in the Recycle Bin and select **Restore File**.

Delete permanently

If you want to permanently delete a file or directory, right-click its record in the Recycle Bin and select **Delete File**.

Empty Recycle Bin

If you want to clear all deleted files and directories, right-click a random record in the Recycle Bin and select **Empty Recycle Bin**. Proceed with caution. These actions are not reversible.

File Stores

The **File Stores** tab displays only when you have the administrator privileges.

Create a filesystem

A filesystem is a logical file location, which reads and writes from a single store. Click the **File Stores** tab. Right-click a random record and select **Create FileSystem....** Type the name in the Create FileSystem dialog and click **OK**. Your new filesystem will display in the Repository Tree.

A file store is a physical file location. It might be on your local disk, on a server, or in an Amazon cloud. One file store may support multiple filesystems. To create a file store, you should have the domain administrator privileges. Refer to *Working with the Domain Manager*.

Create a filesystem snapshot

Administrators have an option to create a snapshot of a filesystem. Different from backup and restore, which keeps a physical copy of the filesystem state typically for long-term use, the snapshot enables the store to hold filesystem information typically for short-term use.

Follow the syntax below to create a snapshot of a filesystem:

```
./elx-snapshot-create.sh <domain> <user> <password> <snapshot-name>  
<filesystem>
```

The following shows an example:

```
cd /opt/elixir/dacapo/  
./elx-snapshot-create.sh eno admin sa mysnap ElixirSamples
```

After some changes have been made, for example, adding files, removing files, editing file contents and changing access rights, you can restore the snapshot using the following command lines, which will undo all changes since the snapshot is taken. Be sure to use the consistent snapshot name:

```
cd /opt/elixir/dacapo/  
./elx-snapshot-restore.sh eno admin sa mysnap ElixirSamples
```

Compact a store

Compact will remove physical data from storage which is no longer referenced. For example, if you have three files with the same contents, we only store the contents once on disk - all three logical files share it. If you delete one of the logical files, the other two still share the physical data. If you delete both of those logical files, then the physical data is now no longer referenced and can be compacted.

To remove unreferenced data and release storage room, click the **File Stores** tab. Right-click the target store and select **Compact**. A message pops up, indicating that the the store is successfully compacted. Click **OK**.

Note

If a filesystem is compacted, a snapshot that depends on the removed files cannot be successfully restored.

Repository Synchronization

Repository Synchronization refers to the synchronization between a local directory and a remote directory within the Repository. According to your configuration of direction, Ambience enables you to drag and drop files into one end and automatically reflects the folder content changes in the other end.

Different from the main features of Ambience, the Repository Synchronization feature needs some manual configurations from the user before it starts to function. In this section, "server" refers to the machine running the Ambience suite, and "client" refers to the end user's machine.

Prerequisite: The server is already set up and running.

Complete the following steps on the client:

1. Install two .deb files using the following command lines:

```
sudo dpkg --install elx-arch_2.5-1_all.deb  
sudo dpkg --install elx-dacapo-sync_2.5-1_all.deb
```

Note

If client and server are the same machine, it means you already have `elx-arch` installed.

2. Edit the configuration file using the following command line:

```
sudo emacs /etc/elixir/dacapo-sync/sync.xml
```

Note

You have the option to use any editor according to your preference. For example, you can replace `sudo emacs` with `sudo gedit`.

The default configuration file contains the following code segment:


```
<?xml version="1.0"?>

<sync-config initial-delay="120">
<!-- 120 gives system two minutes to get started -->

  <server url="http://knockshinnie:8080/elx" domain="eno"
  username="admin" password="9ILS4lhvu8HjaYFjZy6KjQ==">
    <local path="/home/jon/DaCapo"/>
    <remote path="/User/admin"/>
    <action interval="300" direction="both" change-trigger=
    "true"/>
    <!-- direction may be up, down or both -->
  </server>
</sync-config>
```

You have to edit the following values:

- The `server url` value needs to reflect your machine.
- Change the `domain` value as appropriate.
- Choose the username and password for authentication. The password must be encrypted (the default value here is "sa"). Please use the encryption tool at `/opt/elixir/dacapo-sync/encrypt.sh` to encrypt your password.

The following command lines show how to use the encryption tool. When running the command, please replace "sa" with your own password:

```
username@/opt/elixir/dacapo-sync: ./encrypt.sh sa
Encrypted value: 9ILS4lhvu8HjaYFjZy6KjQ==
```

- Choose the local directory that you want files to be mirrored to or from. All child directories will be mirrored too.
- Choose the Repository folder that you want files to be mirrored to or from. All child folders will be mirrored.
- Set the frequency of synchronization. The unit is second, hence 300 means 5 minutes. Set the direction as well.
 - a. If the direction value is set to "up", client files will be sent to the server. Any files added on the server will not be replicated to the client. Files deleted on the client will not be deleted on the server.
 - b. If the direction value is set to "down", server files will be sent to the client. Any files added on the client will not be replicated to the server. Files deleted on the server will not be deleted on the client.
 - c. If the direction value is set to "both", the system will ensure the files are fully synchronized in both directions. In this mode, the system recognizes deletes on either the client or server side, and replicates them on the other.

Note

You can have as many `<server>` sections within the `<sync-config>` as you need. Therefore you can mirror multiple folders independently.

3. Edit another configuration file using the following command line:

```
sudo emacs /etc/init.d/elx-dacapo-sync
```

This script starts or stops the Repository Synchronization service. By default it is set to run as the "elixir" user:

```
#the user that will run the script  
USER=elixir
```

You need to change `elixir` to your own (linux) account name. Then the service will run as your (linux) account name and ensure that files have your permissions.

4. If the service is running as your (linux) account name, it won't have permissions to write logs to `/var/log/elixir`. Please choose one of the following options:

- Option 1: Assign yourself permissions to write to the log folder, by running the following command lines:

```
sudo chown elixir:USERNAME /var/log/elixir  
sudo chmod g+w /var/log/elixir
```

- Option 2: Change the logging folder in `/etc/elixir/dacapo-sync/log4j.properties`, to point to a folder that you have write access to:

```
log4j.appender.ROLLINGFILE.File=/var/log/elixir/dacapo-sync.log
```

5. Start the Repository Synchronization service by running the following command line:

```
sudo service elx-dacapo-sync start
```

If you make further changes to the configuration files, e.g. adding new directories you want to mirror, please restart the Repository Synchronization service to have them read:

```
sudo service elx-dacapo-sync restart
```

If you want to stop or even remove the service, please run the following commands:

```
sudo service elx-dacapo-sync restart  
sudo dpkg --remove elx-dacapo-sync
```