

# BizDock components

BizDock is a play framework application which is divided into multiple components. This structure is motivated by the need to speed-up the compilation times (play framework is based on a 100% compiled model, including the views and the links). Each BizDock component is a maven project and a git repository.

## Functional components

Here are the 3 main “functional components”:

- The **app-framework** (<https://github.com/theAgileFactory/app-framework>) : this is a play project (mavenized) which contains the “non business” functions and services. In theory, one may consider reusing the framework code for building another kind of business application. The framework is the part of the application which is supposed to be quite stable.
- The **maf-datamodel** (<https://github.com/theAgileFactory/maf-desktop-datamodel>) : this is a play project (mavenized) which contains the business objects of the application (portfolio, actors, milestone, etc.)
- The **maf-desktop-app** (<https://github.com/theAgileFactory/maf-desktop-app>) : this is the play project which depends on the two previous ones and which implements the BizDock business logic (DAO, controllers and views).

## Data evolution components

In addition to these functional components, BizDock also requires 2 “data components” which are managing the database evolutions using the [MyBatis migration maven plugin](#):

- The **dbmdl-framework** (<https://github.com/theAgileFactory/dbmdl-framework>)
- The **maf-dbmdl** (<https://github.com/theAgileFactory/maf-dbmdl>)

These projects basically contain a description of the database changes (a list of SQL scripts) which are run against the database before the deployment of the code.

- The location of the scripts for the **maf-dbmdl** component is <https://github.com/theAgileFactory/maf-dbmdl/tree/master/src/main/resources/repo/scripts>.
- The location of the scripts for the **dbmdl-framework** component is <https://github.com/theAgileFactory/dbmdl-framework/tree/master/src/main/resources/repo/scripts>.

The database contains a table (named `__ADM_CHANGELOG`) which records the scripts which have been successfully launched. Thanks to these scripts, the database is kept consistent with the code. This also ease the deployment in various environments.

## Use of Maven

We use maven for the dependency management. Play framework normally rely on activator and SBT (Scala Build Tool). We are thus “encapsulating” the SBT build with maven using the Nanoko plugin (<http://nanoko-project.github.io/maven-play2-plugin/maven/release/>). From a build point of view this allow to compile a play application like any maven project.

In development mode, however, we are using activator for the “maf-desktop-app” compilation.

Please refer to the development environment description.

From:

<https://help-online.bizdock.io/> - **BizDock**

Permanent link:

[https://help-online.bizdock.io/doku.php?id=open-source:bizdock\\_components](https://help-online.bizdock.io/doku.php?id=open-source:bizdock_components)

Last update: **2016/03/12 18:21**

