



OpenbizFrameworkArchitecture

describe the architecture of Openbiz framework

Phase-Implementation

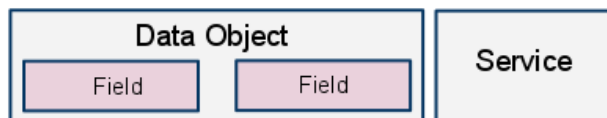
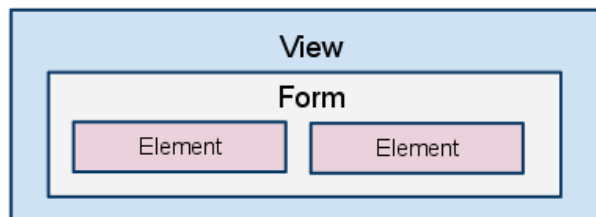
Updated Aug 9, 2011 by [rockys...@gmail.com](#)

Openbiz Architecture

The goal of Openbiz framework is to make design, development and maintenance of web applications quick and easy. The main innovation in Openbiz architecture is its metadata kernel. This means Openbiz objects are constructed based on description in their metadata files. Building an application means design and development of metadata files in most time. Due to the self-explanation nature of XML language, the application is easy to maintain. Meanwhile Openbiz is an extensible framework due to the extensible nature of XML.

Openbiz Core Objects

Any application can compose of two parts - back end and front end. The main business logic typically runs at back end, while the user interface is at front end. In Openbiz, back end will have Data object or Service object. The front end will have Form object and View object.



Data object

Data Object (aka "DO") is an unit of data. Openbiz DO maps database tables and relationship to the object. It encapsulates CRUD (create, read, update and delete) operation in the object as well as provide interface for advanced search.

A DO contains a set of Fields. In typical usage, a DO maps to database table(s), and a Field maps to a table column or a SQL expression.

Service

Service is an unit containing the implementation of business logic or a set of functions. Examples of Openbiz services are User Authentication Service and Email Service.

Form

Form is an unit of UI block that contains a set of related elements. It can be a standard HTML form, a HTML table with toolbar and paging bar, an list of images, and so on.

A Form contains a set of Elements which can be simple or advanced HTML controls. In typical usage, a Form maps to a DO, and an Element maps to a DO Field.

View

View is an actual web page. View is the container of Forms. Considering a View (web page) is a area of floor, a Form is the individual tile.

Core objects and metadata

The core Openbiz objects are defined by Openbiz metadata. The following excerpt lists a partial DO metadata. The meanings of the metadata will be discussed in later chapters.

```
<?xml version="1.0" standalone="no"?>
<BizDataObj Name="UserDO" class="BizDataObj" DBName="Default" Table="user" SearchRule="" SortRule="" OthersSQLRule="" IdGei
  <BizFieldList>
    <BizField Name="Id" Column="id" Type=""/>
    <BizField Name="username" Column="username" Type=""/>
    <BizField Name="password" Column="password" Type=""/>
    <BizField Name="email" Column="email" Type=""/>
  </BizFieldList>
</BizDataObj>
```

Openbiz reads the metadata and creates objects on the fly from ObjectFactory.

Architecture Features

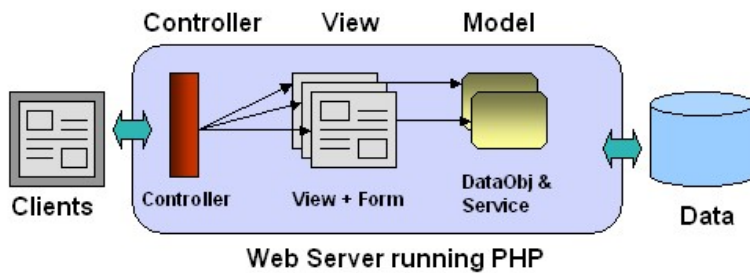
Multi-layer Object Oriented Design

OpenBiz is designed with multi-layer object oriented architecture. Openbiz application is modulated to 3 layers - Presentation layer, Business Logic layer and Data Integration layer. In OpenBiz,

- Presentation layer is implemented by Openbiz View and Form. Openbiz has additional javascript library that communicate with server side presentation objects.
- Business Logic layer is implemented by Openbiz DO as well as Openbiz Service.
- 3rd party package Zend_DB handles data integration layer.

Openbiz MVC

One of the key advantages of Openbiz is that it is a framework that follows the Model-View-Controller (MVC) design pattern. This makes Openbiz applications much more manageable because the presentation code (View) is cleanly separated from the application data and logic (Model). All user interactions with the application are handled by a front-end controller.



Comparing Openbiz MVC with existing MVC frameworks in market such as JSF and Struts, Openbiz is more close to JSF because both are component based architecture. Openbiz's View layer comprises components of View, Form and Element. These components are accessible during request processing.

Openbiz ORM

Openbiz implements necessary object/relational mapping (ORM) features to allow DO representing the data and relationships of database tables. The following section lists the features of Openbiz ORM.

Flexible mapping

- Support table to object mapping. Table-per-class, N tables to 1 class
- Support relationship mapping. Many to one, one to many, many to many, one to one.

Query facilities

- Support SQL like query language
- Support SQL functions and operators
- Support SQL aggregate functions
- Support group by, having and order by

Metadata facilities

- XML metadata describe the mapping

Database Abstraction

Openbiz DO can connect to all types of relational database supported by Zend_DB who provides the database abstraction layer on top of PDO and native database client. Developer can use Openbiz DO API for most database operations as well as Zend_DB API for advanced functions.

To connect to different types of database, user just needs to specify the database connection in application.xml under your application root folder. Openbiz DO will invoke the correct drivers to connect database server. Openbiz is currently support MySQL, MSSQL, Oracle, PostgreSQL, Sqlite and etc.

Each Openbiz DO can have its own Database reference. This features enable the multiple databases connections in one application or even in one web page.

Template Engine

Openbiz Form and View use Smarty template engine to render output by default. As Smarty is the most popular template practice, developers can easily learn how to render a Openbiz page.

In case of complex output that is hard to coded in Smarty template, Openbiz allows users to use PHP template. PHP templates will also give faster rendering speed than Smarty.

Code Structure

Openbiz core library code structure

```
openbiz_root/
```

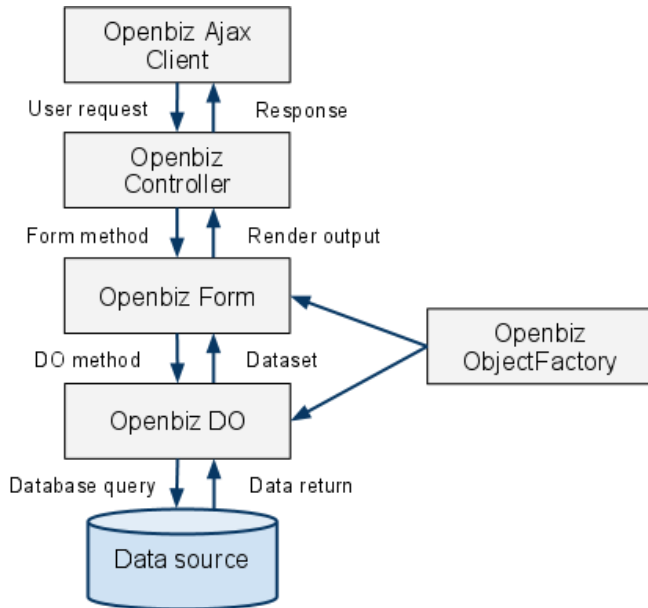
```

---bin/          (openbiz core php source)
-----data/      (data layer classes)
-----private/   (data layer non-public classes)
-----easy/      (new presentation layer classes)
-----element/   (html element classes)
-----service/   (openbiz core service classes)
-----util/      (utility helper classes)
---languages/    (languages files)
---metadata/     (openbiz metadata files)
-----service/   (service package)
---others/       (third party libraries)
-----smarty/    (smarty package)
-----zend/      (Zend Framework)

```

Execution Flow

The following diagram is a typical execution flow of a data query made by user in Openbiz application.



Openbiz and 3rd Party Libraries

Openbiz tried to leverage the best 3rd party libraries on the market. The key libraries are heavily used in Openbiz include:

- Zend Framework. Openbiz uses Zend Framework on
 - Database interactions
 - Language support
 - Data Validation
 - Email service
 - Cache management
 - JSON decode and encode
 - PHP template
- Smarty. Smarty is the main template engine used in Openbiz Form and View. For certain templates that need more complex rendering logic, Openbiz Form uses PHP template provided in Zend Framework.
- Javascript Libraries
 - Prototype. Openbiz Ajax client uses Prototype library for class inheritance and Ajax communication.
 - jQuery. jQuery is used in Openbiz Cubi for advanced UI elements

► [Sign in](#) to add a comment

[Terms](#) - [Privacy](#) - [Project Hosting Help](#)

Powered by [Google Project Hosting](#)