Table of Contents

1. Building and running OFBiz using Docker

OFBiz includes a Dockerfile which can be used to build a container image for running OFBiz. The container image is built based on the sources already available in the build content, i.e. it only uses sources that you have already downloaded.

If you want to include any plugins in your container image, you must download those plugins before running the build. (See the plugin documentation in README.adoc).

1.1. Quickstart

Follow these instructions to qet started building and running OFBiz using Docker.

1.1.1. Build the OFBiz container image

From the sources directory (i.e. the directory containing DOCKER.md), run

```
docker build --tag ofbiz-docker .
```

1.1.2. Run the OFBiz container

Run the following command:

```
docker run -it -e OFBIZ_DATA_LOAD=demo --name ofbiz-docker -p 8443:8443 ofbiz-docker
```

This will start an instance of the ofbiz-docker container, publish port 8443 to localhost, load the OFBiz demo data, and then run the OFBiz server.

Once start up completes, you can access OFBIZ at https://localhost:8443/partymgr

1.2. Other container building options

The OFBiz `Dockerfile` defines a multi-stage build. The default container image produced by the build is named `runtime` and consists of an uninitialized OFBiz installation.

During the first run of a container based on the runtime image, seed or demo data can be loaded and an admin user created.

The runtime container is the default target of a docker build and can be created with commands similar to:

```
docker build --tag ofbiz-docker .
```

or

```
docker build --target runtime --tag ofbiz-docker .
```

The Dockerfile also defines another stage, `demo`, which produces a container image with demonstration data already loaded. This container image may be useful in cases where containers are frequently run and destroyed and the time taken to load demo data is becoming noticeable.

To build a container image pre-loaded with demo data, run:

```
docker build --target demo --tag ofbiz-docker .
```

1.3. Container runtime options

The container's behaviour at runtime is controlled via environment variables, 'hook' scripts and XML entity import files. These items will only be applied to a container during its first run. Flags stored in `/ofbiz/runtime/container_state` will prevent the repeated application of these items during subsequent starts of the container.

Use of environment variables, hook scripts and XML entity import files are manged by the `docker-entrypoint.sh` script.

1.3.1. Environment variables

Environment variables are used in `docker-entrypoint.sh` to control configuration options for the OFBiz container.

Environment variable	Default value	Description
OFBIZ_SKIP_INIT	*empty*	Any non-empty value will cause the docker-entrypoint.sh script to skip any initialisation steps.
OFBIZ_ADMIN_USER	admin	Sets the username of the OFBIZ admin user.
OFBIZ_ADMIN_PASSWORD	ofbiz	Sets the password of the OFBIZ admin user.
OFBIZ_DATA_LOAD	seed	Determine what type of data loading is required. *none*: No data loading is performed. *seed*: Seed data is loaded. *demo*: Demo data is loaded.

Environment variable	Default value	Description
OFBIZ_HOST		Specify the hostname used to access OFBiz. If empty then the default value of host-headers-allowed from framework/security/config/security.properties is used.
OFBIZ_CONTENT_URL_PREFIX		Used to set the content.url.prefix.secure and content.url.prefix.standard properties in `framework/webapp/config/url.properties`.
OFBIZ_ENABLE_AJP_PORT	*empty*	Enable the AJP (Apache JServe Protocol) port to allow communication with OFBiz via a reverse proxy. Enabled when this environment variable contains a non-empty value.
OFBIZ_SKIP_DB_DRIVER_DOWNLOAD	*empty*	Any non-empty value will cause the docker-entrypoint.sh script to skip downloading of any database drivers.
OFBIZ_DISABLE_COMPONENTS	plugins/birt/ofbiz-component.xml	Commas seperated list of paths to ofbiz-component.xml files of the components that should not be loaded.

1.3.2. Hooks

At various steps of initialisation, the `docker-entrypoint.sh` script will check for 'hook' scripts in various directories.

Users of the container can mount files into these directories and influence the OFBiz startup process.

Only script files with filename extension `.sh` will be processed. If the file is executable, it will be executed, otherwise it will be sourced.

Directory	Step
`/docker-entrypoint-hooks/before-config-applied.d`	Scripts processed before configuration, such as modifications to property files, are applied.
`/docker-entrypoint-hooks/after-config-applied.d`	Scripts processed after configuration modifications have been applied.

Directory	Step
`/docker-entrypoint-hooks/before-data-load.d`	Scripts processed before data loading is executed. Could be used to apply modifications to data files.
`/docker-entrypoint-hooks/after-data-load.d`	Scripts processed after data loading is executed.

1.3.3. Data files

During the data loading step - but after either seed or demo data has been loaded - directory `/docker-entrypoint-hooks/additional-data.d` will be checked to see if any files are present.

If files are present then the load-data functionality in OFBiz will be executed, specifying the `/docker-entrypoint-additional-data.d` directory as a data source. Any `.xml` files in this directory will be treated as a data source and will be imported by the entity engine.

This functionality can be used to pre-load OFBiz with user-specific data, such as a chart of accounts.

1.3.4. Database

By default the OFBiz container will use an internal Derby database, storing database related files in the /ofbiz/runtime volume.

Use of an external database can be configured through environment variables.

Derby

To use the embedded Derby database, ensure all database related environment variables are unset.

PostgreSQL

To use a Postgres database set the `OFBIZ_POSTGRES_HOST` environment variable.

Environment variable	Default	Description
OFBIZ_POSTGRES_HOST	*unset*	Hostname of the PostgreSQL database server.
OFBIZ_POSTGRES_OFBIZ_DB	ofbiz	Name of the *ofbiz* database.
OFBIZ_POSTGRES_OFBIZ_USER	ofbiz	Username when connecting to the ofbiz database.
OFBIZ_POSTGRES_OFBIZ_PASSWORD	ofbiz	Password when connecting to the ofbiz database.
OFBIZ_POSTGRES_OLAP_DB	ofbizolap	Name of the *olap* database.
OFBIZ_POSTGRES_OLAP_USER	ofbizolap	Username when connecting to the olap database.
OFBIZ_POSTGRES_OLAP_PASSWORD	ofbizolap	Password when connecting to the olap database.

Environment variable	Default	Description
OFBIZ_POSTGRES_TENANT_DB	ofbiztenant	Name of the *tenant* database.
OFBIZ_POSTGRES_TENANT_USER	ofbiztenant	Username when connecting to the tenant database.
OFBIZ_POSTGRES_TENANT_PASSWORD	ofbiztenant	Password when connecting to the tenant database.

The docker-entrypoint.sh script will download a JDBC driver to access the PostgreSQL server and place the script in the `/ofbiz/lib-extra` volume. If you wish to skip this step then set the OFBIZ_SKIP_DB_DRIVER_DOWNLOAD environment variable to a non-empty value. This would be useful if you have already placed a suitable database driver in the `/ofbiz/lib-extra` volume.

1.4. Examples of running the OFBiz container

```
docker run -it -p 8443:8443 ofbiz-docker
```

Launch the OFBiz container, load the seed data, create the administrator user with name `admin` and password `ofbiz`, listen on port 8443 for connections to `localhost`.

Users can access OFBiz at https://localhost:8443/partymgr

The docker container will remain attached the terminal. Interrupting the container, i.e. pressing Ctrl-C, will trigger a graceful shutdown of the container.

```
docker run -it -e OFBIZ_DATA_LOAD=demo -p 8443:8443 ofbiz-docker
```

Launch the OFBiz container, load the demo data, listen on port 8443 for connections to `localhost`.

The demo data includes the administrator user with name `admin` and password `ofbiz`.

```
docker run -it -e OFBIZ_DATA_LOAD=seed -e OFBIZ_ADMIN_USER=localadmin -e
OFBIZ_ADMIN_PASSWORD=TTTTT -p 8443:8443 ofbiz-docker
```

Launch the OFBiz container, load the seed data, create the administrator user with name `localadmin` and password `TTTTT`, listen on port 8443 for connections to `localhost`.

```
docker run -it -v 'C:\ofbiz-framework\add-data':/docker-entrypoint-additional-data.d
-p 8443:8443 ofbiz-docker
```

Example of running on Windows.

Launches the container with default seed data and administrator user.

After data is loaded, any `.xml` files in directory `C:\ofbiz-framework\add-data` are imported by the

OFBiz entity engine.

```
docker run -it -p 8443:8443 ofbiz-docker
```

Launch the OFBiz container, load the seed data, create the administrator user with name `admin` and password `ofbiz`, listen on port 8443 for connections to `localhost`.

Users can access OFBiz at https://localhost:8443/partymgr

The docker container will remain attached the terminal. Interrupting the container, i.e. pressing Ctrl-C, will trigger a graceful shutdown of the container.

```
docker run -it -e OFBIZ_OPTS="-
agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=*:5005" -p 8443:8443 -p
5005:5005 ofbiz-docker
```

Creates a debuggable instance of OFBiz, listening on port 5005.