

Installation Documentation

map.apps with MD2

Project Seminar

Model-driven Mobile Development

(University of Münster)

15th December 2014

Contents

1	Prerequisites	1
2	Installing a map.apps Development Environment	1
2.1	Basic map.apps Development Setup	1
2.2	Deploy MD2 Runtime Bundles	1
3	Creating Symbolic Links for Generated Apps	2
3.1	Reference a Generated App in the Development Project	2
3.2	Jetty: Allow Serving of Symbolically Linked Files	2
4	Running a Generated Backend	2
4.1	Configure Eclipse to Manage GlassFish	2
4.2	Start GlassFish and Add the Backend	3

1 Prerequisites

The following software is required prior to the next steps:

- map.apps 3.1.0
- Eclipse IDE for Java and DSL Developers (e.g. Version Kepler)
- NetBeans EE (e.g. Version 8.0)
- Apache Tomcat 7.0 with running map.apps runtime
- GlassFish 4.+

2 Installing a map.apps Development Environment

2.1 Basic map.apps Development Setup

1. Extract the `sampleProjRemote` project from the map.apps distribution and load it in NetBeans.
2. In its `pom.xml`, set the `mapapps.remote.base` directive to the URL where the map.apps runtime is installed.
3. Right-click the project, select “Custom” and select “Goals...” to create a new Maven goal.
4. Open the `nbactions.xml`.

Describe configuration

Describe set property for Jetty port 9090

2.2 Deploy MD2 Runtime Bundles

1. Copy the MD2 runtime bundles into the `bundles/` directory of the project:

```
md2_formcontrols
md2_local_store
md2_location_service
md2_runtime
md2_store
onlinestatus
```

2. Copy a generated MD2 app into the project or use a symlink to reference apps from another location (see Section 3.1).

3 Creating Symbolic Links for Generated Apps

3.1 Reference a Generated App in the Development Project

By using a symlink, a running NetBeans instance will automatically notice changes to the generated app. Consequently, if the Jetty server is running, the newly generated app will automatically be published and made available in the browser.

1. Open a terminal, and navigate to the map.apps NetBeans project directory (e.g. the extracted sampleProjRemote from step 1 in Section 2.1).
2. Navigate to the apps/ directory using `cd src/main/js/apps`.
3. Create a symbolic link using an appropriate command (where `<PROJECT_NAME>` is the name of your MD2 Project in Eclipse, `<ECLIPSE_PROJECT_LOCATION>` is its location, and `<APP_NAME>` is the name of the generated app(s)):
 - a) Windows:

```
mklink /j <APP_NAME> <ECLIPSE_PROJECT_LOCATION>\<PROJECT_NAME>\src-gen\<PROJECT_NAME>.mapapps\<APP_NAME>
```
 - b) Linux / OS X:

```
ln -s <ECLIPSE_PROJECT_LOCATION>/<PROJECT_NAME>/src-gen/<PROJECT_NAME>.mapapps/<APP_NAME> <APP_NAME>
```
4. Repeat step 3 for every generated app that you would like to be refreshed automatically.

3.2 Jetty: Allow Serving of Symbolically Linked Files

On Linux/OSX, Jetty by default does not serve symbolically linked files due to security concerns. To override this setting (which is not recommended in a production environment), put the `jetty-web.xml` file into the folder `/src/test/webapp/WEB-INF/` of your map.apps project.

4 Running a Generated Backend

4.1 Configure Eclipse to Manage GlassFish

Get
GlassFish
sup-
port, add
GlassFish
server

4.2 Start GlassFish and Add the Backend

1. Within the GlassFish installation directory, navigate to `glassfish/bin/`.
2. Run the `asadmin` utility (Windows: Double-click on `asadmin.bat`, Linux/OS X: Open a terminal in that directory and run `./asadmin`).
3. In the GlassFish administration utility, type `start-database` to start the Derby database for the backend.
4. Start Eclipse and import the generated project `<PROJECT_NAME>.backend`.
5. In the Project Explorer tab, right-click the project in Eclipse, choose “Properties”, and navigate to “Targeted Runtime”.
6. Deselect all runtimes and click “Apply”.
7. Select the item “GlassFish 4.0” and click “Apply”.
8. Correct JRE-related build path problems, if any, by resorting to the default JRE.
9. Confirm by clicking “OK”.
10. In the Servers tab, right-click the “GlassFish 4.0” entry, and choose “Add/Remove”.
11. Add the backend project to the server.
12. Start the server.