DARWINO

Developers Guide

Table of Contents

Darwino Installation Guide

Welcome to Darwino!

This guide will show you how to install a development environment to create Darwino based applications.

To get started, go to Setting Up page.

Installation

Setting Up Your environment

Darwino is a flexible development platform that supports IDE or command line based development. The platform itself is built using Apache Maven. Maven is also the recommended build tool for Darwino applications, but other systems can also be used, including Gradle, Eclipse PDE, etc. The following instructions assume that you are using:

- Eclipse J2EE as your IDE
- Maven as your build tool.
- NodeJS and NPM for JavaScript client development

Unless specified otherwise, all the installation instructions assume a Microsoft Windows environment, with the software being installed on `c:\Darwino`. However, most of the instructions work identically on other platforms such as Apple macOS or Linux.

Darwino currently supports the following target platforms:

- Web applications through a Java EE server
- Android applications
- Apple iOS applications
- OSGi environments, like Eclipse rich client or IBM Domino

Notes: Based on Apple requirements, an Apple Mac computer running macOS is required for developing Apple iOS applications. Currently, only Android applications can be developed and tested using Microsoft Windows or Linux.

Darwino Installation

In order to develop with the Darwino Studio, the following components should be installed:

- Java JDK: The minimum version is Java 7, but it is advised to use the latest one (Java 8 as of today)
- Eclipse for J2EE or other IDE
- Web application server is required to run the web applications. Although any servlet container 2.5+ would work, these instructions show how to install/run the demo projects using Apache TOMCAT
- Database server: These instructions document the use of Postgres 9.4+ database server. Although other databases may be used, the demo applications are pre-configured to work with PostgreSQL. (Note that a future version of Darwino might use a embedded database and thus this step won't be necessary.)

Also, a few configuration actions has to be taken:

- Configure the Maven settings to point to Darwino repository
- Configure the NPM settings to point to Darwino repository
- Configure the Darwino beans and properties

Note that since Darwino 1.5.1, the Darwino Eclipse studio features a configuration module that makes the installation process easier. It is highly recommended to use this helper.

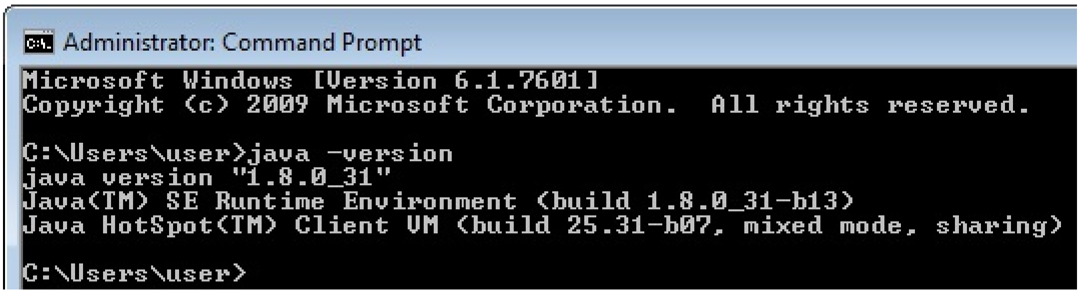
The following instructions describe the installation of each component.

Installing the Java JDK

If you already have Java JDK 7 or higher installed (Java 8 is highly recommended), you can skip this step.

Afull Java JDK is required by the maven tools to compile the Java. AJava JRE is not sufficient.

The Java SDK can be downloaded from the Oracle web site: http://www.oracle.com/technetwork/java/javase/downloads/index.html. An IBM JDK will also work. Once installed, make sure that the development environment variable JAVA_HOME is pointing to you Java environment. Make also sure that the JVM is available from the command line (Terminal on macOS or Linux) by using the command `java -version`.

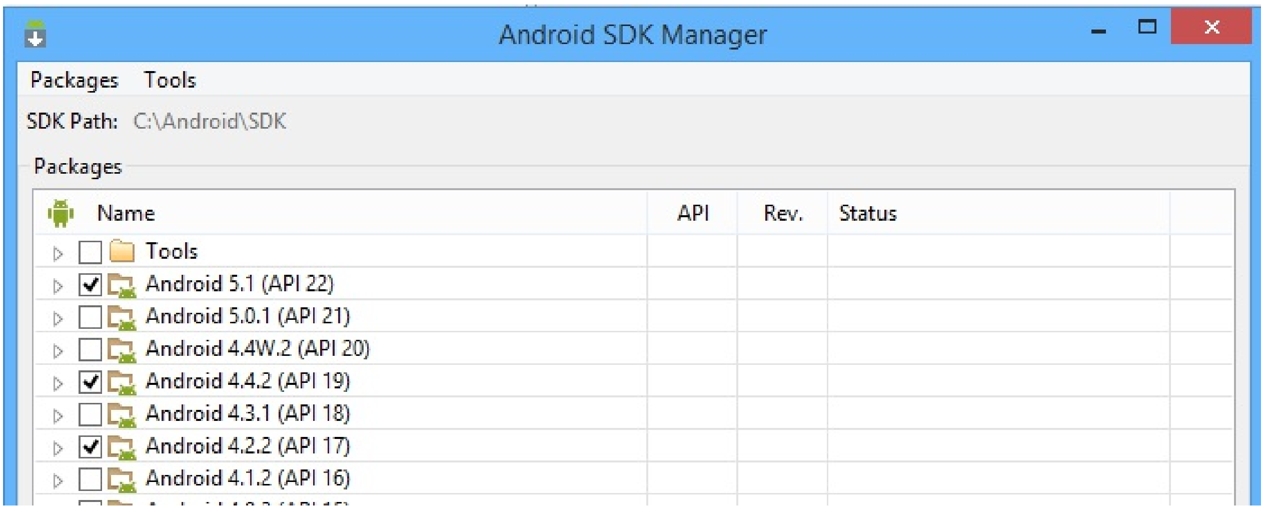


```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\user>java -version
java version "1.8.0_31"
Java(TM) SE Runtime Environment (build 1.8.0_31-b13)
Java HotSpot(TM) Client VM (build 25.31-b07, mixed mode, sharing)

C:\Users\user>
```

Installing the Android application development tools

Developing for the Android platform requires the Android Software Development Kit to be installed. It is available from: http://developer.android.com/sdk/index.html#Other. The 'SDK Tools only' package is sufficient if you plan to use Eclipse or Maven from the command line.

Using the SDK manager Android SDK manager, select and install at least the entire "Android 4.2.2 (API 17)" & "Android 4.4.2 (API 19)" folders.



SDK Path: C:\Android\SDK

Packages

Name	API	Rev.	Status
☐ Tools			
☑ Android 5.1 (API 22)			
☐ Android 5.0.1 (API 21)			
☐ Android 4.4W.2 (API 20)			
☑ Android 4.4.2 (API 19)			
☐ Android 4.3.1 (API 18)			
☑ Android 4.2.2 (API 17)			
☐ Android 4.1.2 (API 16)			

Additionally, set the path to the Android SDK root (e.g. `c:\Android\SDK`) in an environment variable named `ANDROID_HOME`. Also do not forget, when installing maven, to add the android libraries to your local repository, as described in the documentation on installing Maven. The Studio configuration module will also report an error if this is not properly defined.

Note: using the stock Android emulator, it is likely that the applications will be too slow to provide a great developer experience. This is particularly true if you launch the emulator from a virtual machine. It is then better to run the demo applications on real hardware or in a faster emulator/virtual machine. See: http://blog.riand.com/2014/08/running-android-apps-for-development.html

Installing the iOS application development tools

As stated earlier, developing for iOS requires an Apple Mac computer running OS X. To compile, run in a simulator or deploy the application, the Apple Xcode development environment is required. It is available freely from the Mac App Store (https://itunes.apple.com/us/app/xcode/id497799835?mt=12).

PostgreSQL

Postgresql is a free and popular relational database featuring well suited JSON extensions for Darwino. The instructions bellow show how to install Postgreql for development and test purposes. Other databases, like IBM DB2 10.5 FP8 or MS SqlServer 2016, are also supported.

See: Configuring PostgreSQL

Installing Eclipse for J2EE

We advise developers to use Eclipse as the development IDE. Moreover, starting with Darwino 1.5.1, the Darwino Eclipse plugins feature an installation validation module that checks for the pre-requisites, and helps fixing the potential issues.

If you plan to use Eclipse, or already do, please ensure that you first ensure that you follow the [instructions] (Installing Eclipse) for downloading and configuring Eclipse, **and the Darwino Studio Eclipse add-on component**. These instructions assume that you will use the Eclipse for J2EE version (Not Eclipse for Java).

NPM Installation

NPM should be configured to point to the Darwino repository for Darwino libraries, identified by @darwino.

You should first install NodeJS and NPM following the instructions here: https://www.npmjs.com/get-npm

Then you should configure NPM to connect to the Darwino private repository. See: Configuring NPM

Manual configuration

These steps are only required if they were not achieved through the Studio configuration module.

Maven Configuration

Maven should be configured to point to the Darwino repository for Darwino libraries. If you are using Eclipse, please see the following instructions after installing Eclipse: Setting Up Maven.

(If you are using the command line tools, then you should first install maven (latest recommended) and configure it as described in the instructions.)

Apache TOMCAT

Apache Tomcat is a free, easy to use, Java servlet container. The instructions bellow explain how to configure it for development and test purposes. Any servlet 3.0 compliant web server can also be used.

See: Configuring TOMCAT

Note that TOMCAT can also be launch from Maven. Most of the demo projects, as well as the new application wizard, now integrate the TOMCAT Maven plugin. As a result, TOMCAT can be launched using a maven commend without going though a full installation: `mvn tomcat7:run`

IBM Websphere Liberty

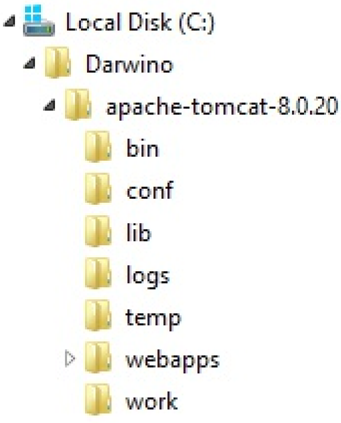
Alternatively, you can use IBM Websphere Liberty as the application server, on premises and on the cloud through IBM Bluemix.

See: Configuring IBM Websphere Liberty

If you plan to use Tomcat as the development web application server, then you have to both install it in the file system and eventually configure Eclipse to use it.

Installing Apache Tomcat

Download the latest Tomcat server (8.x or 8.5x) from the Apache Web Site: https://tomcat.apache.org/download-80.cgi. Just unzip the server under your installation directory:



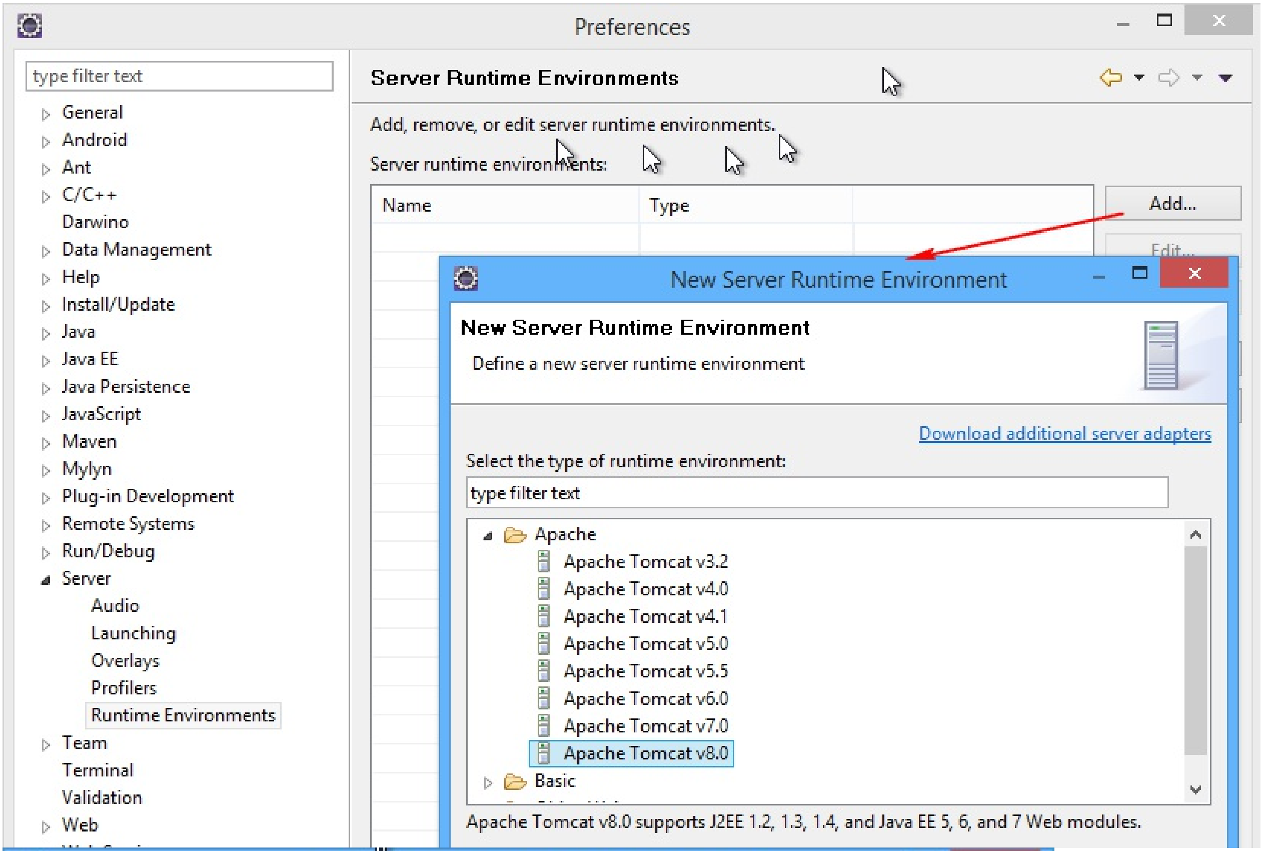
```
Local Disk (C:)
  Darwino
    apache-tomcat-8.0.20
      bin
      conf
      lib
      logs
      temp
      webapps
      work
```

If you plan to use the demo applications, then you need to add some demo users with roles in your tomcat environment. Add the following content to `{tomcat install dir}\conf\tomcat-users.xml` :

```
<role name="user"/>
  <user password="darwino" roles="user" username="atinov"/>
  <user password="darwino" roles="user" username="amass"/>
  <user password="darwino" roles="user" username="aboucher"/>
  <user password="darwino" roles="user" username="acalder"/>
  <user password="darwino" roles="user" username="agardner"/>
  <user password="darwino" roles="user" username="bchapot"/>
  <user password="darwino" roles="user" username="blemercier"/>
  <user password="darwino" roles="user" username="bchris"/>
  <user password="darwino" roles="user" username="bbright"/>
  <user password="darwino" roles="user" username="larmatti"/>
  <user password="darwino" roles="user" username="lbros"/>
  <user password="darwino" roles="user" username="mdavis"/>
  <user password="darwino" roles="user" username="pcollins"/>
  <user password="darwino" roles="user" username="rjordan"/>
```

Configuring Eclipse with Tomcat

Eclipse J2EE comes with a set of tools called WTP, which allows the configuration and use of application servers. To configure Apache TOMCAT, you have to create a new 'Runtime Environment' from the Eclipse 'Window->Preferences...':



Server Runtime Environments

Add, remove, or edit server runtime environments.

Server runtime environments:

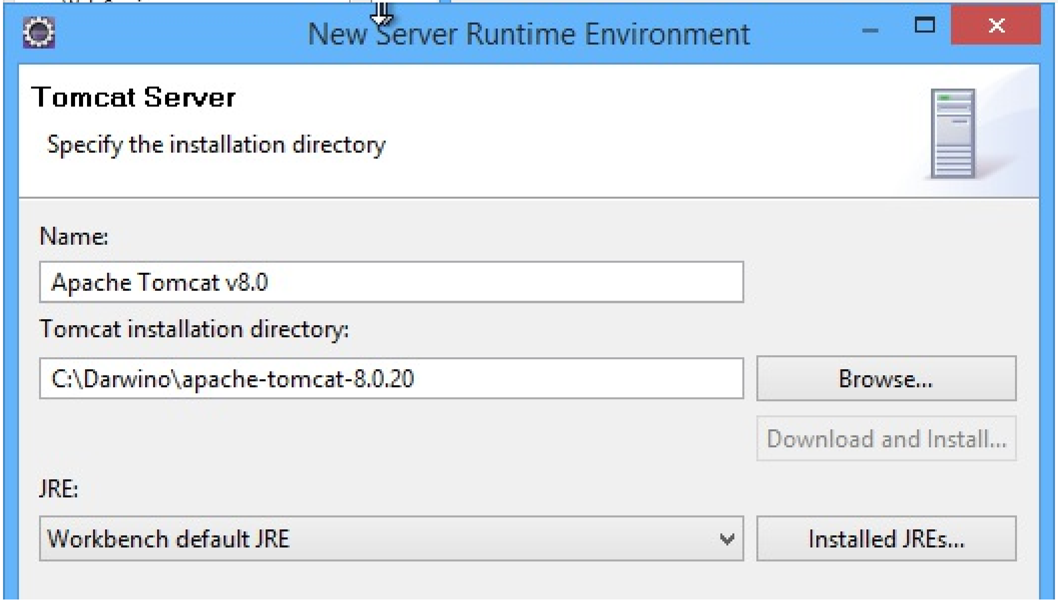
New Server Runtime Environment

Define a new server runtime environment

Download additional server adapters

Select the type of runtime environment:

Apache Tomcat v8.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5, 6, and 7 Web modules.



Tomcat Server

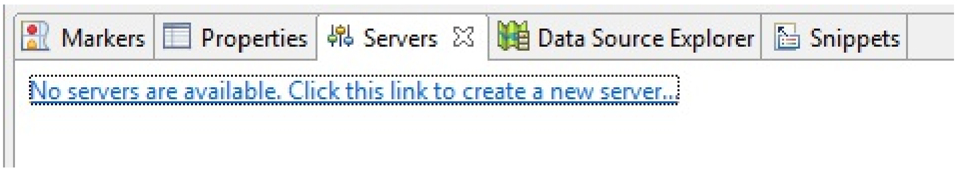
Specify the installation directory

Name: Apache Tomcat v8.0

Tomcat installation directory: C:\Darwino\apache-tomcat-8.0.20

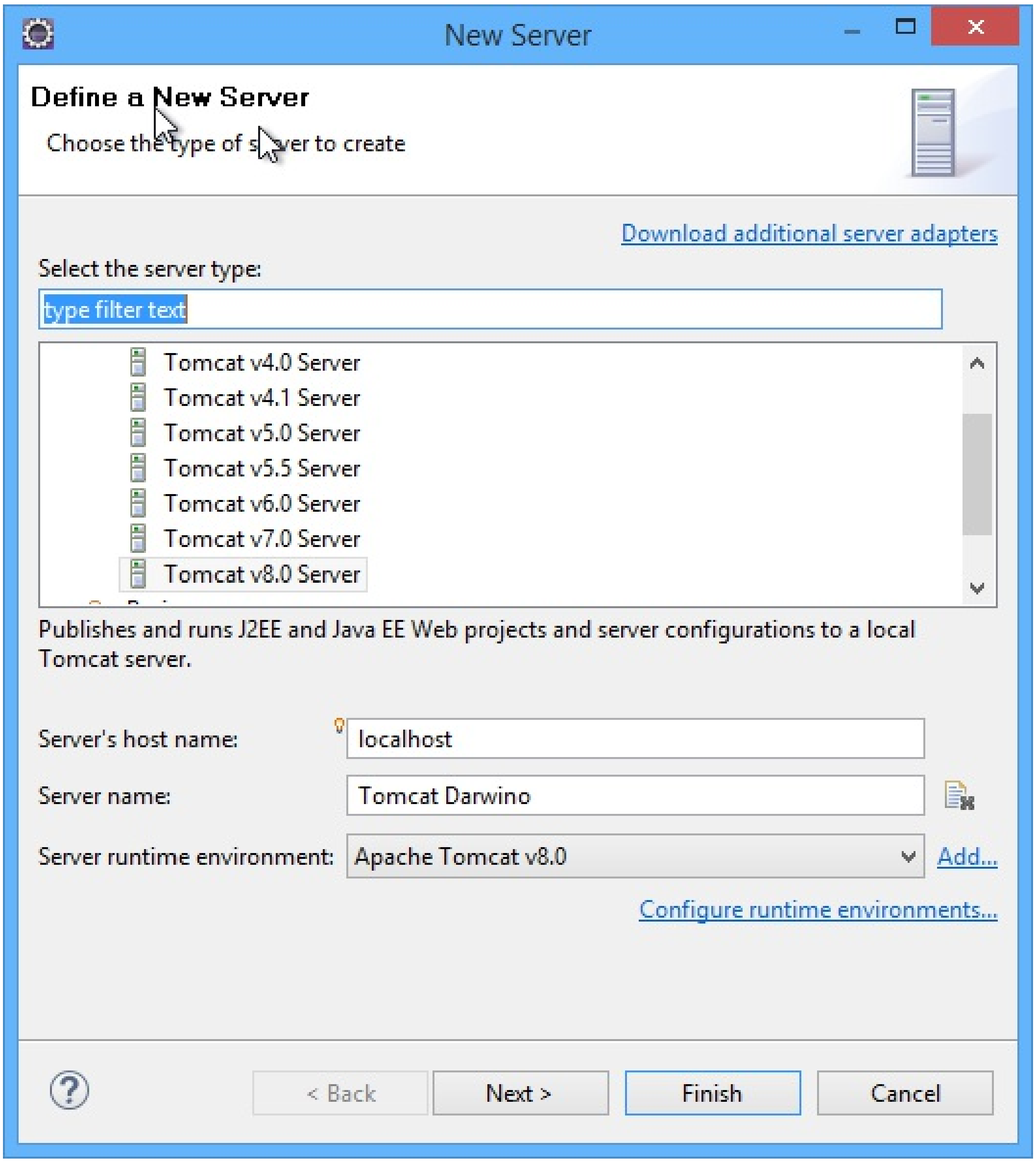
JRE: Workbench default JRE

Once the runtime environment is configured, you should create a new server. For this, you should either make visible the Eclipse 'Servers' view (Window->Show View...Servers) or switch to the J2EE perspective (Window->Open Perspective...Java EE). In the server view, create a new server by hitting the link bellow. If the view displays a list instead, right click and select 'New->Server...'



No servers are available. Click this link to create a new server...

Select your Tomcat environment and eventually change the server name, and hit 'Finish'



New Server

Define a New Server

Choose the type of server to create

Download additional server adapters

Select the server type:

type filter text

- Tomcat v4.0 Server
- Tomcat v4.1 Server
- Tomcat v5.0 Server
- Tomcat v5.5 Server
- Tomcat v6.0 Server
- Tomcat v7.0 Server
- Tomcat v8.0 Server

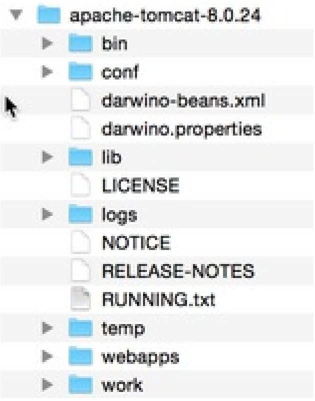
Publishes and runs J2EE and Java EE Web projects and server configurations to a local Tomcat server.

Server's host name: localhost

Server name: Tomcat Darwino

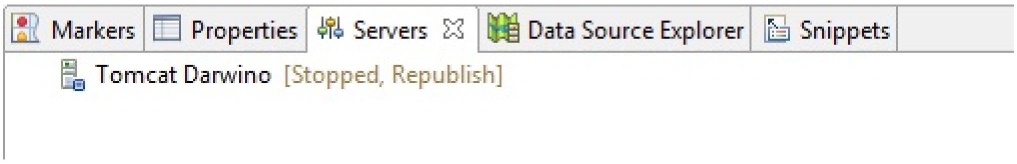
Server runtime environment: Apache Tomcat v8.0

Download the darwino-beans.xml and darwino.properties files to the Tomcat install directory.



```
apache-tomcat-8.0.24
    bin
    conf
    darwino-beans.xml
    darwino.properties
    lib
    LICENSE
    logs
    NOTICE
    RELEASE-NOTES
    RUNNING.txt
    temp
    webapps
    work
```

Now you have your server ready for the Darwino applications

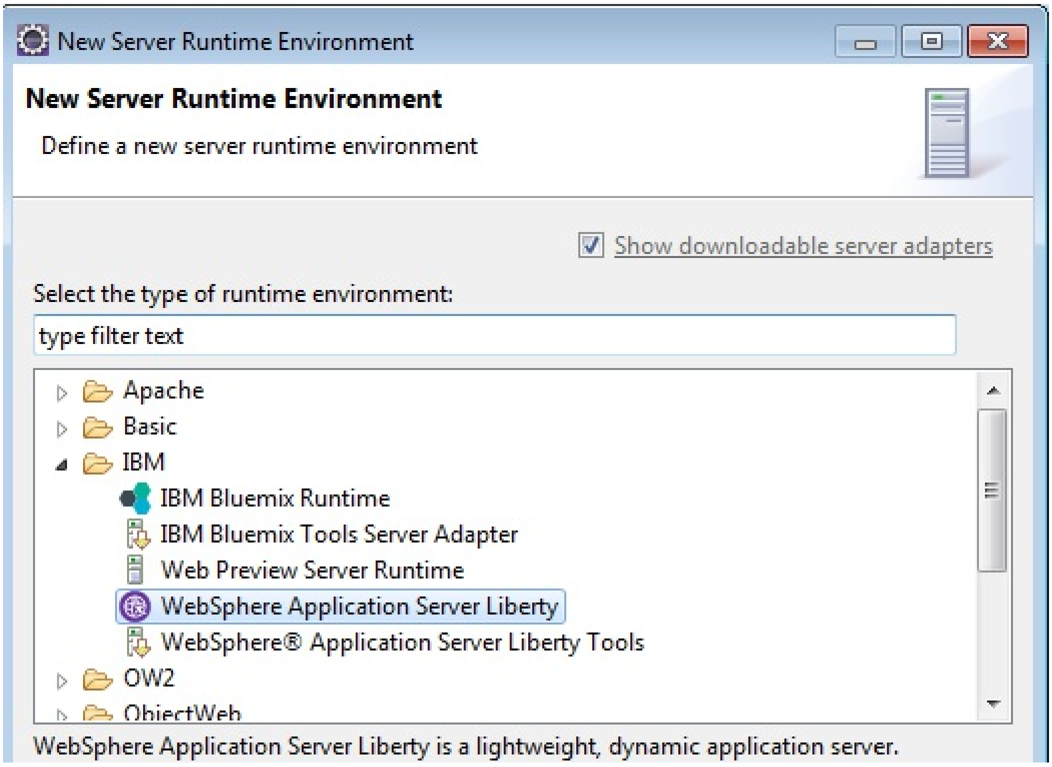


Tomcat Darwino [Stopped, Republish]

If you plan to use IBM Websphere Liberty as the development web application server, then you have to install it and configure the Eclipse accordingly. The easiest is to drop the install site to the Eclipse toolbar, as explained here: https://developer.ibm.com/wasdev/getstarted/

Configuring Eclipse with IBM Websphere Liberty

Eclipse J2EE comes with a set of tools called WTP, which allows the configuration and use of application servers. To configure WAS Liberty, you have to create a new 'Runtime Environment' from the Eclipse 'Window->Preferences...':



New Server Runtime Environment

New Server Runtime Environment

Define a new server runtime environment

☑ Show downloadable server adapters

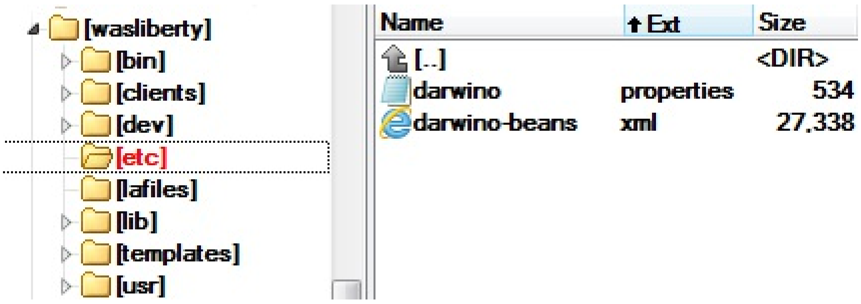
Select the type of runtime environment:

type filter text

- ▷ Apache
- ▷ Basic
- ◢ IBM
  - IBM Bluemix Runtime
  - IBM Bluemix Tools Server Adapter
  - Web Preview Server Runtime
  - WebSphere Application Server Liberty
  - WebSphere® Application Server Liberty Tools
- ▷ OW2
- ▷ ObjectWeb

WebSphere Application Server Liberty is a lightweight, dynamic application server.

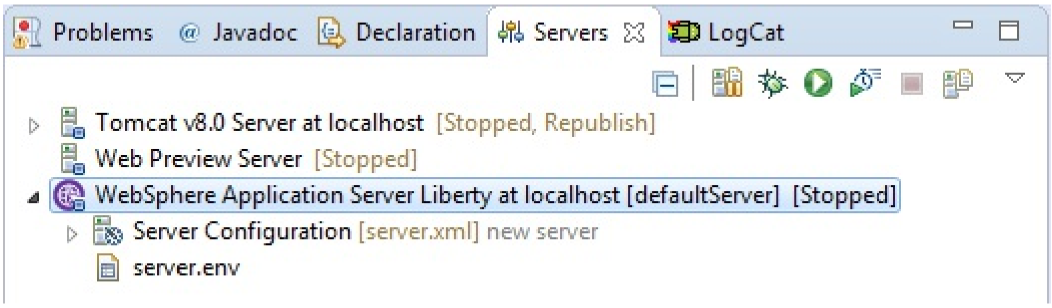
Download the darwino-beans.xml and darwino.properties files to the Webphere Liberty install directory, under /etc (if the optional directory does not exist, just create it).



- ◢ [wasliberty]
  - ▷ [bin]
  - ▷ [clients]
  - ▷ [dev]
  - [etc]
  - [lafiles]
  - ▷ [lib]
  - ▷ [templates]
  - ▷ [usr]

Name	↑Ext	Size
[..]		<DIR>
darwino	properties	534
darwino-beans	xml	27,338

Now you have your server ready for the Darwino applications



- ▷ Tomcat v8.0 Server at localhost [Stopped, Republish]
- Web Preview Server [Stopped]
- ◢ WebSphere Application Server Liberty at localhost [defaultServer] [Stopped]
  - ▷ Server Configuration [server.xml] new server
  - server.env

PostgreSQL is a free, open source database with great capability. It is particularly suited to Darwino because, since 9.4, it features a native JSON data type with query capabilities. Moreover, it comes with an integrated full text search engine.

Installing PostgreSQL

If you just started with PostgreSQL, the we strongly advise you to run one of the predefined installers. See: http://www.postgresql.org/download/. Please, choose the latest version, or at least version 9.4.

By default, the pre-packaged Darwino configuration files (see: TOMCAT installation) look for a PostgreSQL database on the local host using the user/password and port below:

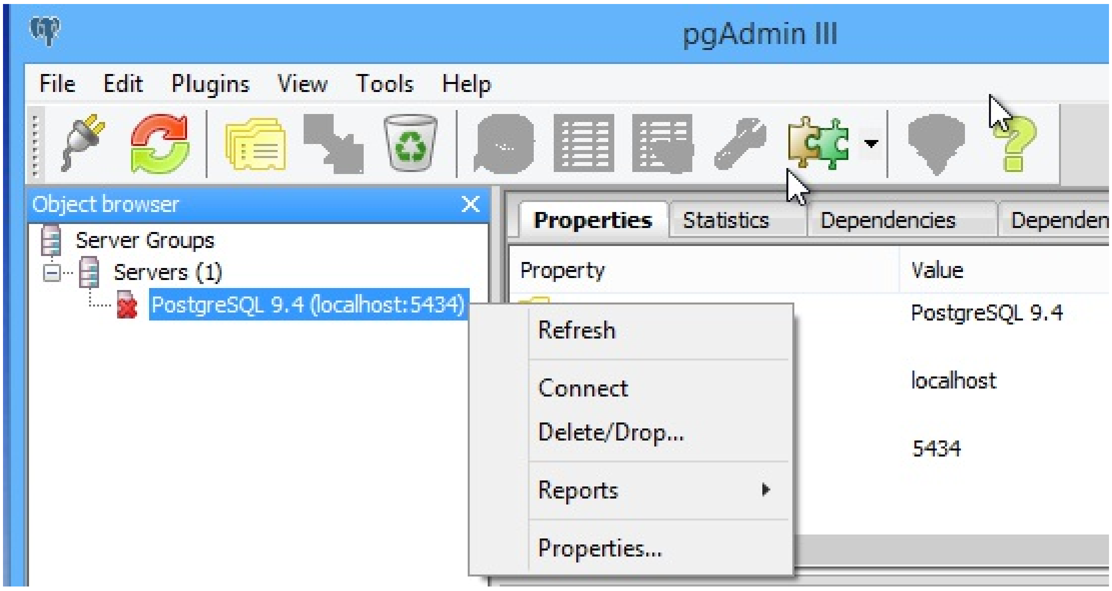
- Port: 5432 *(default)*
- Username: `postgres`
- Password: `postgres`

Configuring PostgreSQL for the Demo Application

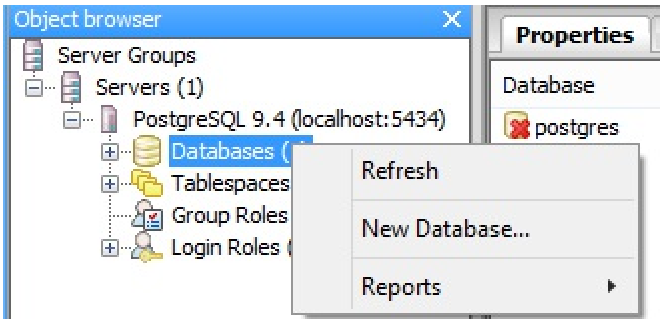
The demo projects use a specific database called `dwodemo`, which should be created by either using the PostgreSQL command line tools, or the easiest UI admin 'pgAdmin III', installed by the PostgreSQL installer.

Here are the steps:

1. Launch pgAdmin III
2. Right click on the server and select 'Connect'. If asked, enter the password 'postgres'

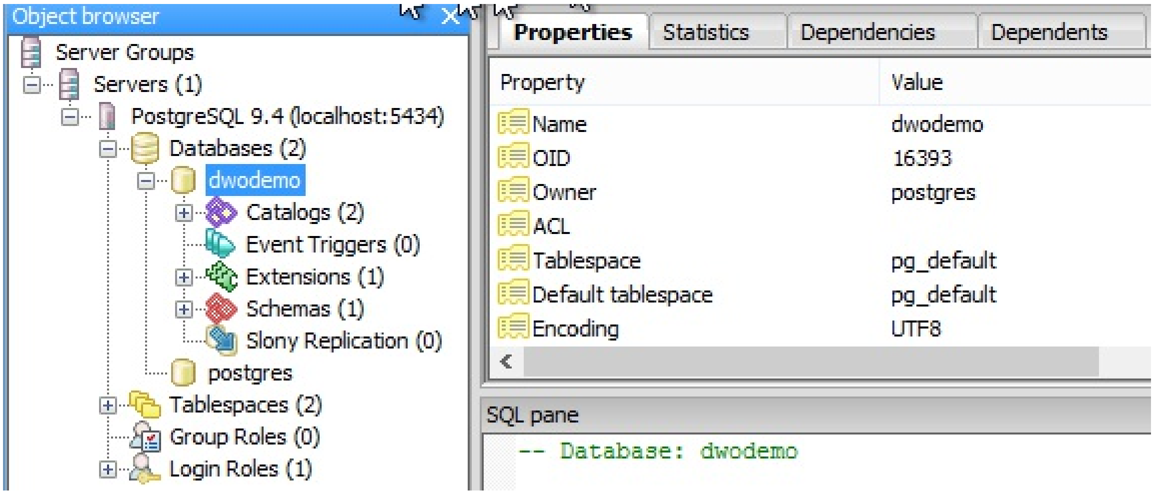


3. Right click on the database and select 'New Database...'



4. Name it `dwodemo` and hit ok. All the other default settings

Once done, you should see the database created after double clicking it in the list:



Property	Value
Name	dwodemo
OID	16393
Owner	postgres
ACL	
Tablespace	pg_default
Default tablespace	pg_default
Encoding	UTF8

```
-- Database: dwodemo
```

Basic NPM Settings

Darwino requires NodeJS and NPM to compile the new generation of JavaScript applications.

The Darwino JavaScript assets are distributed through a private NPM server serving all the packages scoped by @darwino.

To be able to consume them, you should instruct you local NPM to use that server for @darwino. This is done by running the following command from the command line (it assumes that NodeJS and NPM are installed): `npm adduser --registry https://npm.darwino.com --scope @darwino`

It will ask you for your darwino user/password and your email. Once done, you'll be able to consume all the darwino packages as described at: https://npm.darwino.com

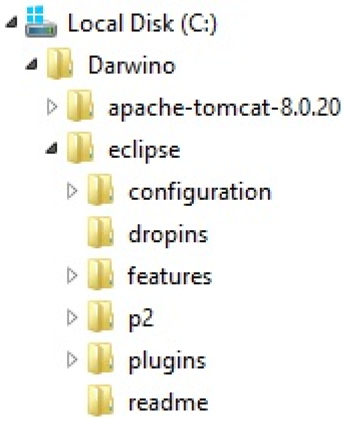
Installing the Eclipse IDE

Although the use of an IDE, and in particular Eclipse, is not a requirement, it certainly eases the development of Darwino applications. In particular, it also helps validate that all the requirements are met (installed software, configuration files...)

In order to configure web servers and get the best web development experience, please download and install the latest version of '**Eclipse IDE for Java EE Developers**' (NOT Eclipse IDE for Java Developers).
http://www.eclipse.org/downloads/

Darwino run best with the latest Eclipse version, currently Neon-1. Although it has been tested with earlier versions like Mars-1, the maven integration definitively works best with Neon.

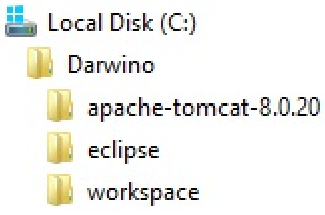
Installing Eclipse is achieved by unzipping the archive in its target directory. The following directory structure shows eclipse unzipped into the c:\Darwino directory:



```
Local Disk (C:)
  Darwino
    apache-tomcat-8.0.20
    eclipse
      configuration
      dropins
      features
      p2
      plugins
      readme
```

Launching Eclipse for the first time

When Eclipse is launched for the first time, it prompts for a workspace directory. The default value is fine, but well, you could also make this directory a peer of the eclipse one, to get the whole environment contained within `c:\Darwino` :



```
Local Disk (C:)
  Darwino
    apache-tomcat-8.0.20
    eclipse
    workspace
```

Installing the Darwino Studio add-on

The first module to install is the Darwino Studio. Bellow are the instructions:

1. Select Help > Install New Software.

2. In the "Work With:" field, enter the following URL, depending on the edition you selected:

   - Enterprise Edition: https://eclipse.darwino.com/darwino/studio/enterprise/
   - Community edition: https://eclipse.darwino.com/darwino/studio/community/

   Note that you can also install the nightly builds by using the URLs bellow. The nightly builds provide access to the latest version, under development

   - Enterprise Edition: https://eclipse.darwino.com/darwino/studio/enterprise-nightly/
   - Community edition: https://eclipse.darwino.com/darwino/studio/community-nightly/

3. When prompted, enter your Darwino user and password, as provided during the registration process.

4. In the Available Software dialog, select the Darwino Studio checkbox.

5. In the next window, you'll see a list of the tools to be downloaded. Click Next.

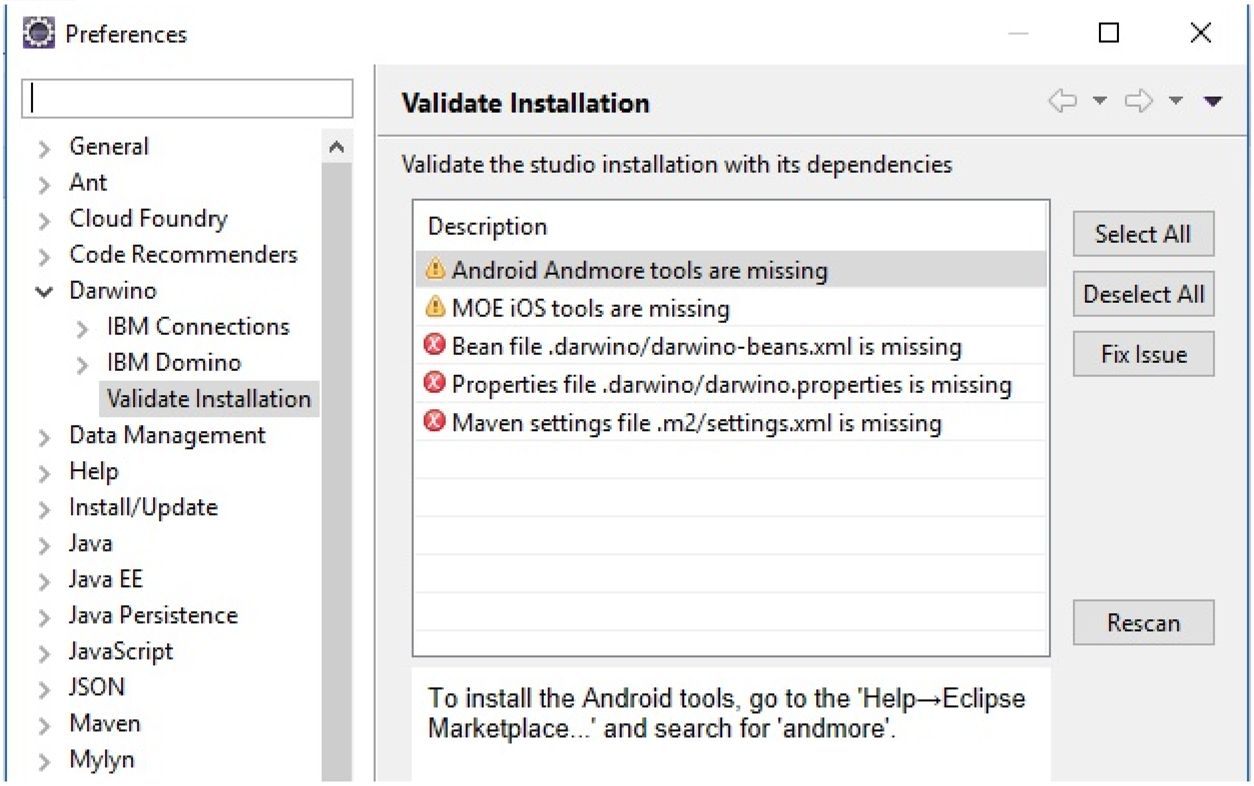
6. Read and accept the license agreements, then click Finish.

7. If you get a security warning saying that the authenticity or validity of the software can't be established, click OK.

8. When the installation completes, restart Eclipse.

Validating the configuration

The Darwino studio, once installed, features an installation validation module available under `Window->Preferences / Darwino`:



Validate Installation

Validate the studio installation with its dependencies

Description
⚠ Android Andmore tools are missing
⚠ MOE iOS tools are missing
☒ Bean file .darwino/darwino-beans.xml is missing
☒ Properties file .darwino/darwino.properties is missing
☒ Maven settings file .m2/settings.xml is missing

To install the Android tools, go to the 'Help→Eclipse Marketplace...' and search for 'andmore'.

This module checks your environment and gives you guidance on the next actions:

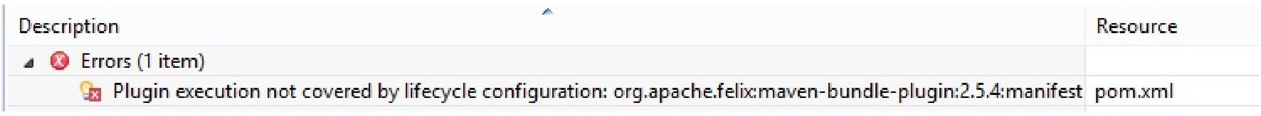
- Having Maven settings.xml properly configured
- Having the Darwino configuration files at the right place
- Making sure that the Eclipse dependencies are installed
- ...

Manually Configuring Eclipse

The configuration validator exposed above should give you all the necessary information but some details are exposed bellow.

Darwino requires some extra plug-ins bellow to be installed on top of the version of Eclipse. These plug-ins can generally be installed from the Eclipse Marketplace or by selecting `Help->Install New Software...`, then use the provided update site URL

- If Android is a desired target platform:
    - Andmore (successor to Google's ADT), Select `Help->Eclipse->Marketplace...` and search for `andmore`. It will be called something like "Andmore: Development Tools for Android"
    - Note: during the process for installing m2e adapters (as with Tycho below), the m2e adapter for Android may prompt you to install the earlier Android Development Toolkit. If so, and you already have Andmore installed, deselect those plugins during the m2e setup.
- If iOS is a desired target platform:
    - Multi-OS Engine. The current update site is http://dl.bintray.com/multi-os-engine/eclipse
- If IBM Domino is a desired target platform
    - The Darwino project wizard as well as the sample applications use Tycho to target OSGi platforms (Eclipse, IBM Domino...). As a result, it requires the Tycho m2e connector to be installed. The easiest way to get it installed is to wait from Eclipse to display a build problem (see bellow), right click on the issue and choose quick fix. Eclipse will automatically propose to download and install the connector.



Description	Resource
Errors (1 item)	
Plugin execution not covered by lifecycle configuration: org.apache.felix:maven-bundle-plugin:2.5.4:manifest	pom.xml

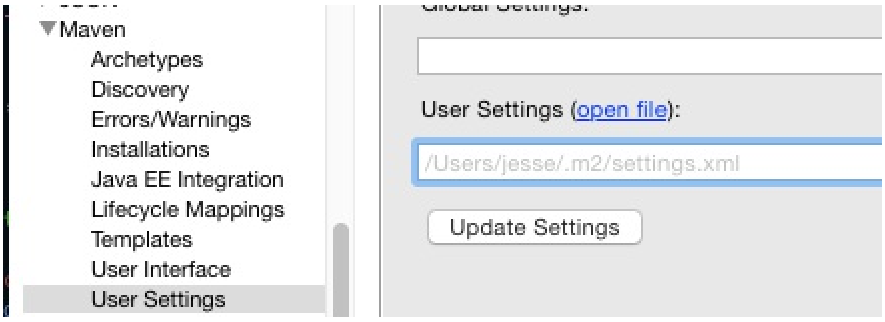
Eclipse Add-ons Information

- Darwino Studio
    - Update site URL: https://eclipse.darwino.com/darwino/studio/enterprise/ https://eclipse.darwino.com/darwino/studio/community/ https://eclipse.darwino.com/darwino/studio/enterprise-nightly/ https://eclipse.darwino.com/darwino/studio/community-nightly/
- Andmore (Android Tools)
    - Install from the Eclipse Marketplace
- Multi-OS Engine
    - Instructions: https://multi-os-engine.org/start/
    - Update site URL: http://dl.bintray.com/multi-os-engine/eclipse

Basic Maven Settings

Darwino projects require Maven to be properly configured to point to the Darwino repository. This is best done by modifying Maven's settings file, which is a file called `settings.xml` located in the `.m2` directory in your home folder.

If you're using Eclipse and this file already exists, you can open it for editing by going to the Maven → User Settings pane in Eclipse's Preferences and clicking "Open File":



User Settings (open file):

/Users/jesse/.m2/settings.xml

If the file does not yet exist, this link will be absent. In that case, create a new file in a text editor and, when saving, browse to your home directory (e.g. `C:\Users\yourname` on Windows, `~/` on a mac), create a folder named ".m2" (with the leading dot), and save the file as "settings.xml":

If it is a new file, use the following content or merge it with the existing file:

```
<?xml version="1.0"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaL
    <profiles>
        <profile>
            <id>darwino-repository-profile</id>
            <properties>
                <!-- Enter the filesystem path to the Android SDKs folder here -->
                <android.sdk.path></android.sdk.path>
            </properties>
            <repositories>
                <repository>
                    <id>darwino-repository</id>
                    <name>Darwino Platform</name>
                    <url>https://maven.darwino.com/darwino-enterprise-edition</url>
                </repository>
                <repository>
                    <id>openntf-repository</id>
                    <name>OpenNTF</name>
                    <url>https://artifactory.openntf.org/openntf</url>
                </repository>
            </repositories>
        </profile>

        <profile>
                <id>bintray-moe</id>
                <repositories>
                        <repository>
                                <snapshots>
                                        <enabled>false</enabled>
                                </snapshots>
                                <id>bintray-repository</id>
                                <name>bintray</name>
                                <url>https://dl.bintray.com/multi-os-engine/maven-dev</url>
                        </repository>
                </repositories>
                <pluginRepositories>
                        <pluginRepository>
                                <snapshots>
```

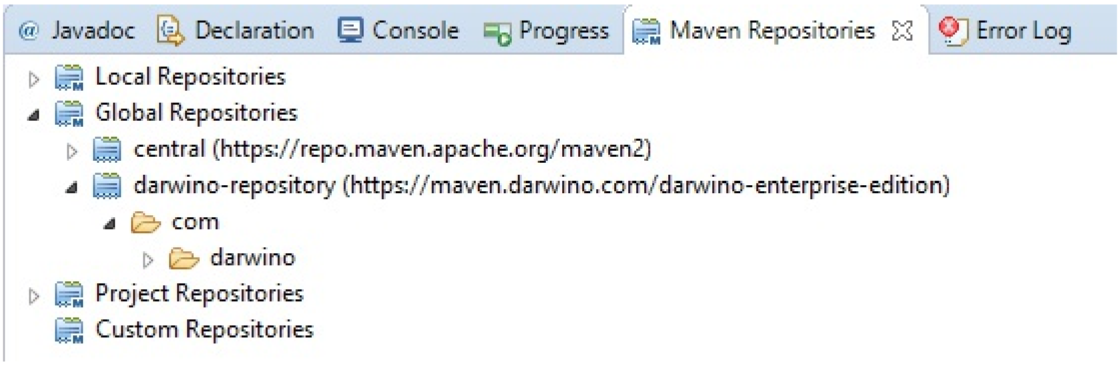
```
                                        <enabled>false</enabled>
                                </snapshots>
                                <id>bintray-repository</id>
                                <name>bintray-plugins</name>
                                <url>https://dl.bintray.com/multi-os-engine/maven-dev</url>
                        </pluginRepository>
                </pluginRepositories>
        </profile>
    </profiles>
    <activeProfiles>
        <activeProfile>darwino-repository-profile</activeProfile>
        <activeProfile>bintray-moe</activeProfile>
    </activeProfiles>
    <servers>
        <server>
            <id>darwino-repository</id>
            <username>[username]</username>
            <password>[password]</password>
        </server>
    </servers>
</settings>
```

This example file can be downloaded here.

Of course, you have to set your own user and password and, depending on the edition you are willing to use, set the repository URL to:

- Enterprise Edition https://maven.darwino.com/darwino-enterprise-edition
- Community Edition https://maven.darwino.com/darwino-community-edition These maven repositories contain both the production builds and the nightly ones (aka xxxx-SNAPSHOTS)

To validate that your maven settings are correct, display the Maven Repositories view in eclipse (`Window->Show View->Other...`). Hit the refresh button to make the Darwino repository appear. Finally, right click on this repository and select `Rebuild Index` ,. As a result, you should see the Darwino assets like in:



Local Repositories
Global Repositories
central (https://repo.maven.apache.org/maven2)
darwino-repository (https://maven.darwino.com/darwino-enterprise-edition)
com
darwino
Project Repositories
Custom Repositories

Installing the Android Libraries

The Darwino sample projects, as well as the studio wizards, assume that these libraries are available.

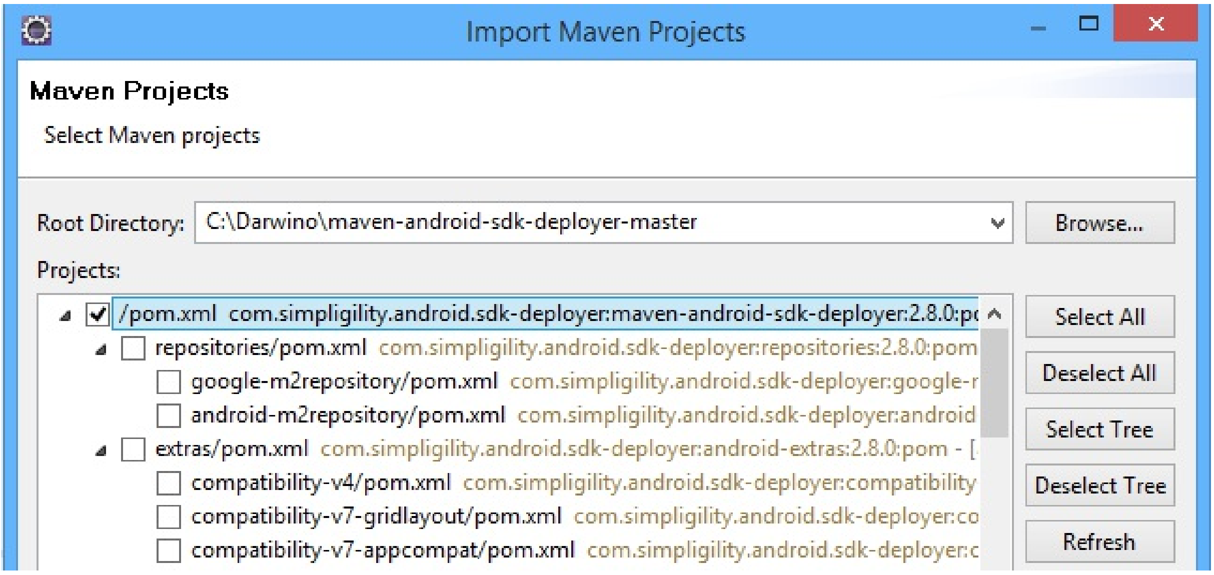
The Android libraries are currently not available from the maven central repository, so each developer has to install them in his local repository. Fortunately, there are some available scripts and instructions provided here.

Tip: there are two ways to install these libraries:

1. Using maven and the command line, as documented in the Github repository. This requires maven to be properly installed. See: https://maven.apache.org/install.html. We suggest to only install the platforms that you

need using `mvn install -P xxx` , where xxx is the desired platform release.

2. Using the Eclipse built-in maven install. To do this, make sure that `ANDROID_HOME` is defined before you launch Eclipse. Then, follow the instruction above to download the installer. In Eclipse, import the main project (and only this one) in your workspace:



Import Maven Projects

Maven Projects

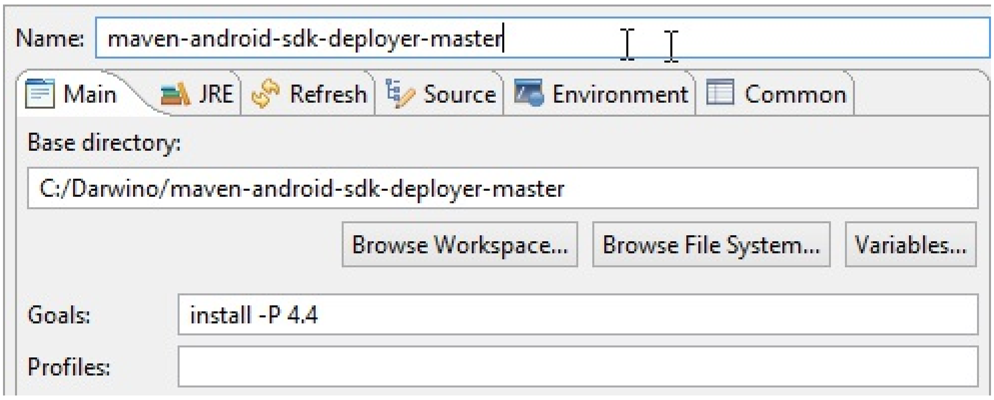
Select Maven projects

Root Directory: C:\Darwino\maven-android-sdk-deployer-master

Projects:

```
◢ ☑ /pom.xml  com.simpligility.android.sdk-deployer:maven-android-sdk-deployer:2.8.0:pc
  ◢ □ repositories/pom.xml  com.simpligility.android.sdk-deployer:repositories:2.8.0:pom
      □ google-m2repository/pom.xml  com.simpligility.android.sdk-deployer:google-r
      □ android-m2repository/pom.xml  com.simpligility.android.sdk-deployer:android
  ◢ □ extras/pom.xml  com.simpligility.android.sdk-deployer:android-extras:2.8.0:pom - [
      □ compatibility-v4/pom.xml  com.simpligility.android.sdk-deployer:compatibility
      □ compatibility-v7-gridlayout/pom.xml  com.simpligility.android.sdk-deployer:co
      □ compatibility-v7-appcompat/pom.xml  com.simpligility.android.sdk-deployer:c
```

Finally, right click on the project, select "Run As...Maven Build..." and manually enter the install command with the desired platform:



Name: maven-android-sdk-deployer-master

Base directory: C:/Darwino/maven-android-sdk-deployer-master

Goals: install -P 4.4

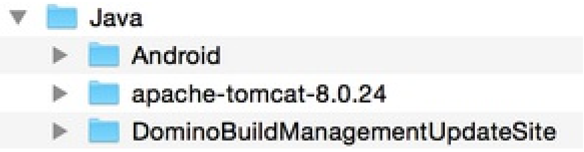
Profiles:

Once done, you can remove this maven project from your workspace, unless you want to install more platforms going forward.

IBM Domino Specific Settings

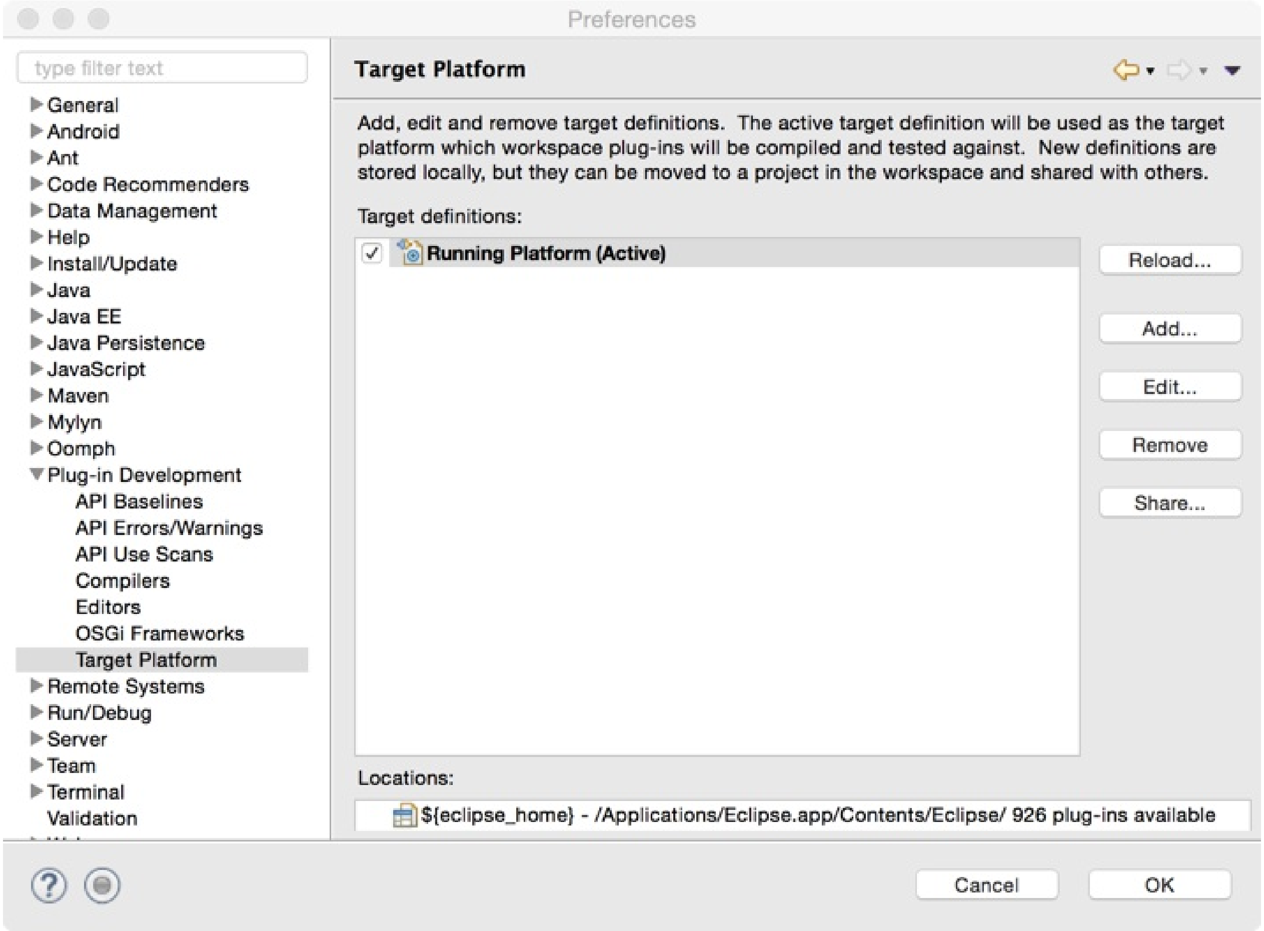
If you will be working on Domino-related projects, it is necessary to download and install and reference the IBM Domino Update Site for Build Management from openntf.org. It contains the OSGi plugins required by Domino.

The Update Site can be downloaded here. Expand the download zip file and place the contents in the Java install directory.



- Java
  - Android
  - apache-tomcat-8.0.24
  - DominoBuildManagementUpdateSite

In Eclipse Preferences, add the Update Site to the target platform. To do this, under 'Plug-in Development\Target Platform' select the Running Platform and choose 'Edit...'. In the resulting dialog, choose 'Add...', then 'Directory', then point to the Update Site directory and click 'Finish'.



Target Platform

Add, edit and remove target definitions. The active target definition will be used as the target platform which workspace plug-ins will be compiled and tested against. New definitions are stored locally, but they can be moved to a project in the workspace and shared with others.

Target definitions:

☑ Running Platform (Active)

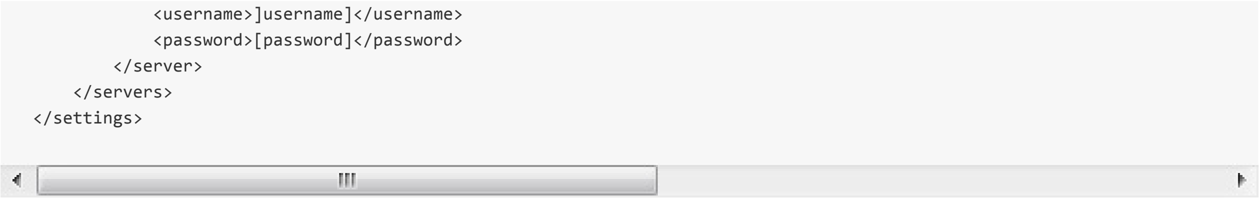
Locations:

${eclipse_home} - /Applications/Eclipse.app/Contents/Eclipse/ 926 plug-ins available

The important information to add to your Maven settings is the location of the Darwino repository, which contains the core Darwino components as well as needed third-party dependencies.

Here is an example of a complete settings.xml containing the remote repository using the generic repo accessor username. If your settings.xml file didn't exist previously, you can use this as the file's contents. If you will be working on Domino-related projects, change the file path specified in the notes-platform property to your Java directory. If you will NOT be working on Domino-related projects, you can delete the notes-platform property altogether.

```
<?xml version="1.0"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLo
    <profiles>
        <profile>
            <id>darwino-repository-profile</id>
            <properties>
                <notes-platform>
                    file:///*path to Java installation*/DominoBuildManagementUpdateSite/UpdateSite
                </notes-platform>
            </properties>
            <repositories>
                <repository>
                    <id>darwino-repository</id>
                    <name>Darwino Platform</name>
                    <url>https://maven.darwino.com/darwino-enterprise-edition</url>
                </repository>
            </repositories>
        </profile>
    </profiles>
    <activeProfiles>
        <activeProfile>darwino-repository-profile</activeProfile>
    </activeProfiles>
    <servers>
        <server>
            <id>darwino-repository</id>
```



```
            <username>]username]</username>
            <password>[password]</password>
        </server>
    </servers>
</settings>
```

To merge this into an existing settings.xml, the pertinent components are the `<profile>...</profile>` block, including the named profile in the `activeProfiles`, specifying the credentials used when accessing the repository, and adding the notes-platform property (if you will be working on Domino-related projects).

Getting Started

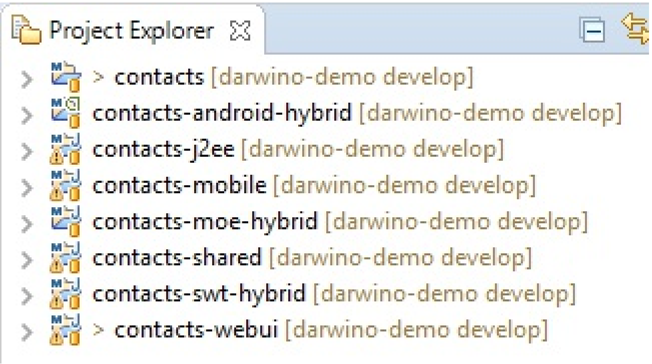
The Contacts sample is a simple project showing the capabilities of the Darwino platform:

- Sample view/form based application
- Runs as a web app as well as mobile hybrid on mobile devices
- Leverages the UI code generator for the UI

Installing the Contacts application in Eclipse

The Contacts application is provided by the darwino-demo Github repository as a set of maven enabled projects. To import them in Eclipse once the Git repository has been cloned, select `Import...->Existing Maven Projects` and choose `darwino-demo/contacts`.

The result should be a set of projects in your workspace:



Project Explorer

> contacts [darwino-demo develop]
contacts-android-hybrid [darwino-demo develop]
contacts-j2ee [darwino-demo develop]
contacts-mobile [darwino-demo develop]
contacts-moe-hybrid [darwino-demo develop]
contacts-shared [darwino-demo develop]
contacts-swt-hybrid [darwino-demo develop]
> contacts-webui [darwino-demo develop]

Note: if this is the first project you import, you might be prompted by Eclipse to install some M2E connectors. If this happens, just install the connectors and reboot Eclipse.

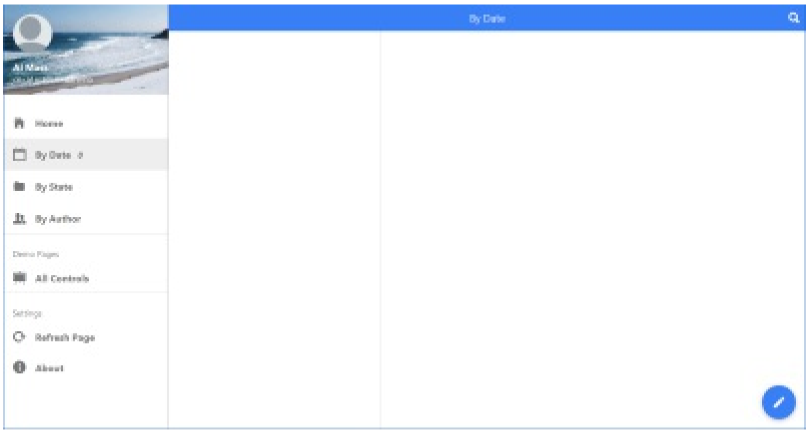
Running the Web Application

There are two ways to run the contact application:

- By starting TOMCAT using the Maven plugin. This is the simplest solution to start with.
- By installing the application within a full TOMCAT

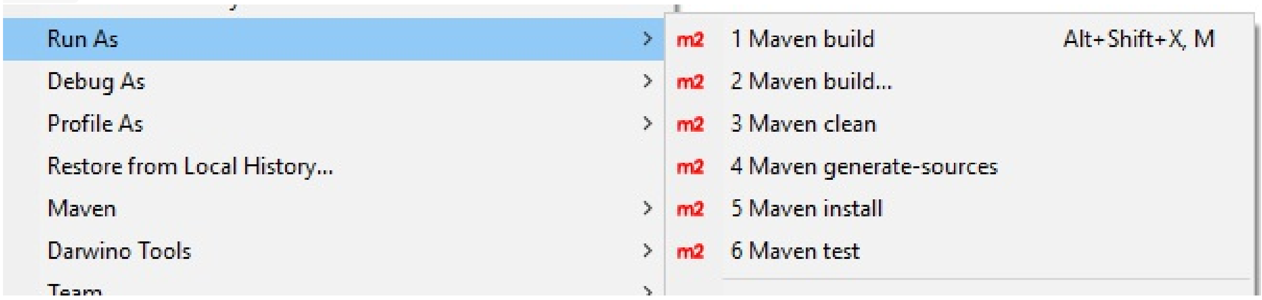
The URL to the application is: http://localhost:8080/contacts/mobile/index.html

The application rendering, with no data, should look like this:

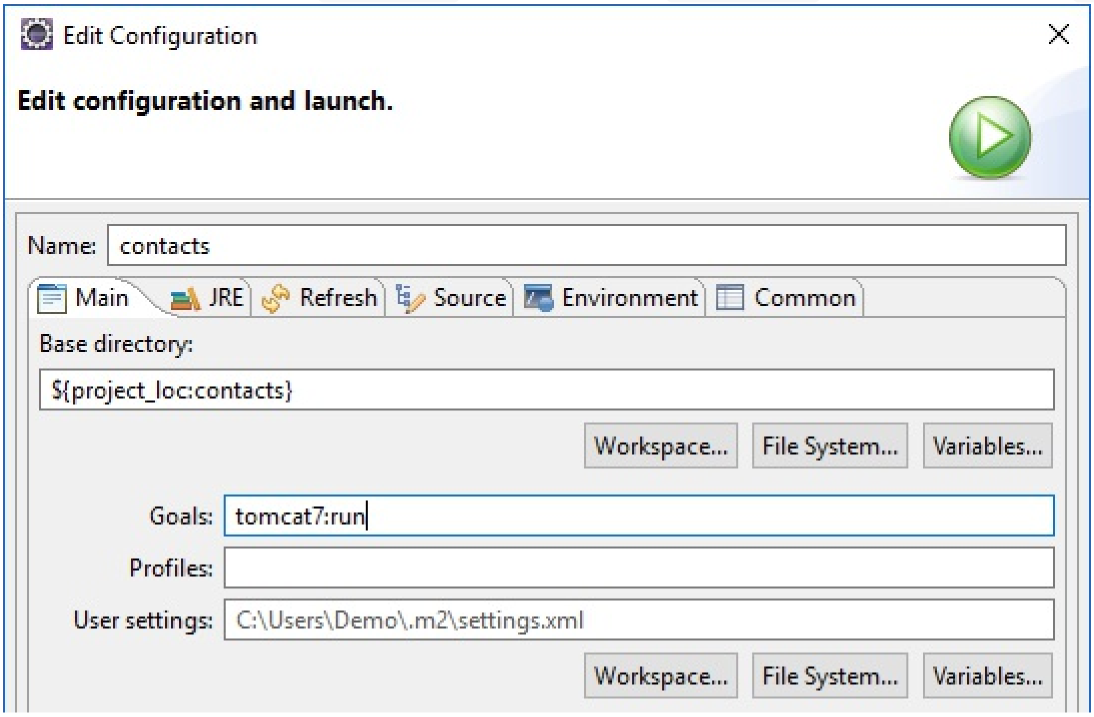


Running the application through Maven

You first have to build the project using Maven. For this, right click on the `contacts` root project and select `Maven install`:



Once done, right click on the `contacts-j2ee` project and select `Maven build...`. In the goal field, enter `tomcat7:run`:


Edit Configuration

Edit configuration and launch.

Name: contacts

Base directory: ${project_loc:contacts}

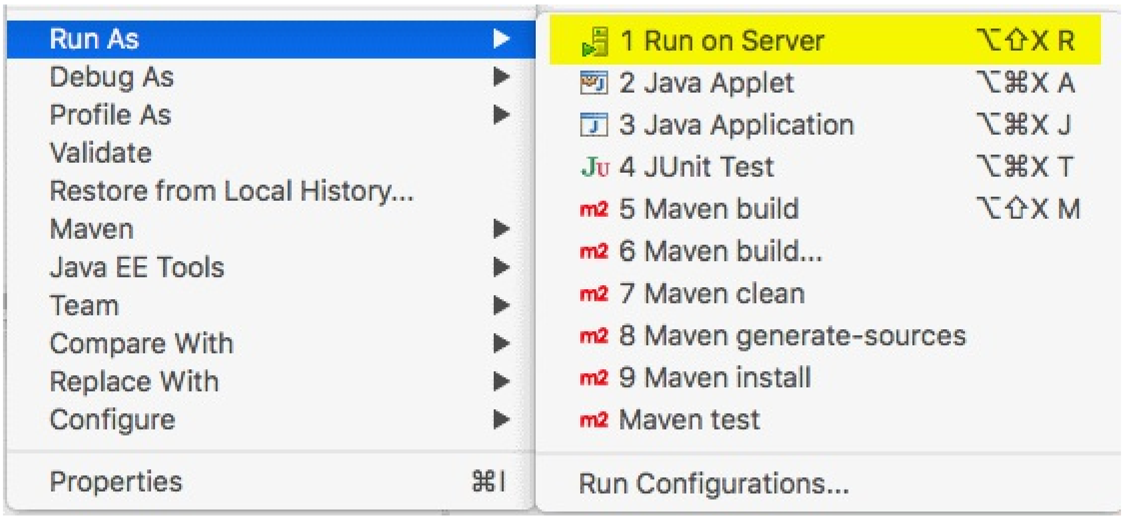
Goals: tomcat7:run

Profiles:

User settings: C:\Users\Demo\.m2\settings.xml

If this is the first time TOMCAT is run this way, then all the dependencies will be downloaded by Maven, which will take a few minutes.

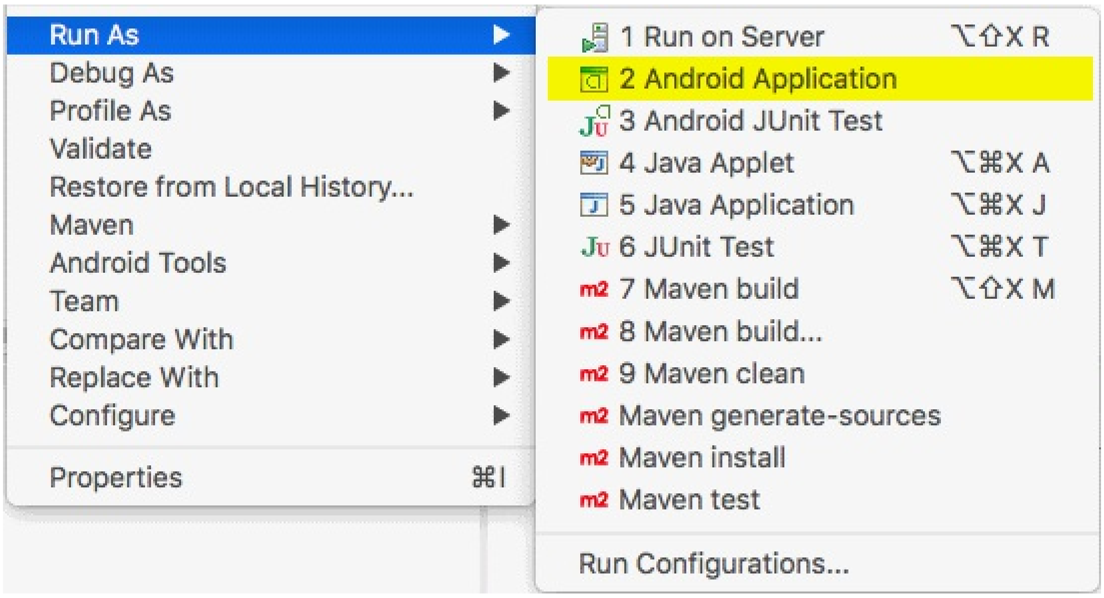
Running the application using Eclipse WTP



This will launch the app in Tomcat and open up the default landing page for the application.

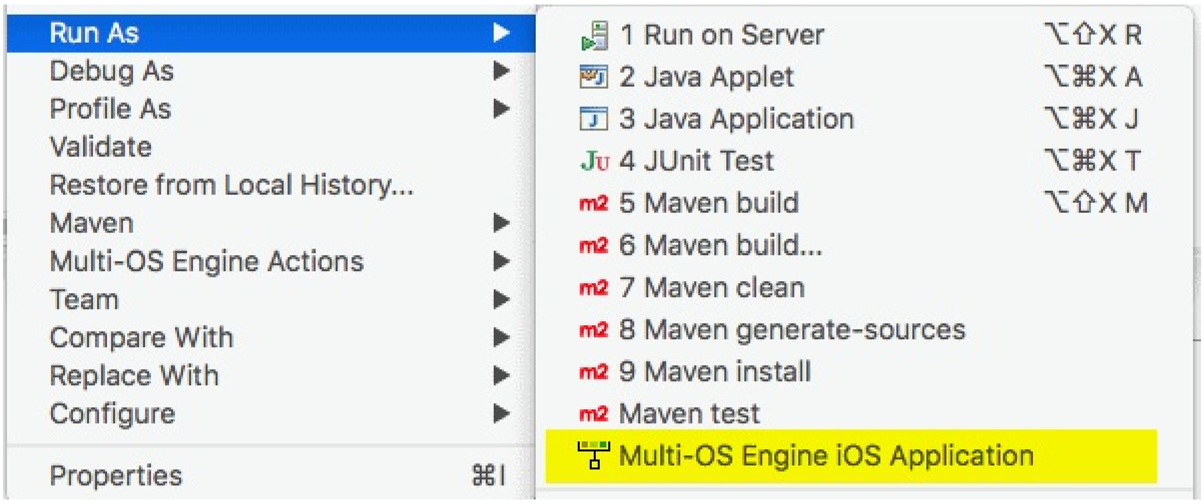
Android Applications

The Android application is `contacts-android-hybrid` . To run it, right-click the project and choose Run As → Android Application:



iOS Applications

If you are running on a Mac, you can run the iOS application, which is `contacts-moe-native` . To do so, right-click on the project and choose one of the Run As → iOS Simulator App options:



Adding data from IBM Notes/Domino

If you're an IBM Domino developer, then the Contacts application is a great example for setting up the replication with an existing NSF database

This assumes that you have some IBM Notes/Domino knowledge and both the Darwino runtime, as well as the Sync Admin database are properly installed on an IBM Domino server. It also assumes that the Darwino configuration files (beans ,properties) are setup, and finally the database JDBC driver in added to the Domino JVM. Be careful, as the Domino server can be executed as a system user and thus the user.home directory can be

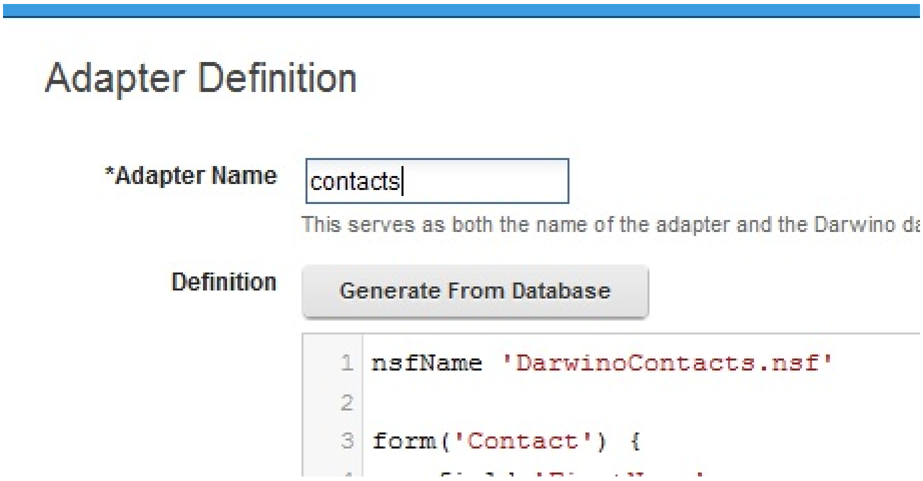
`C:\Windows\System32\config\systemprofile` .

As a first, create a new NSF based on the template 'DarwinoContacts.ntf. This template has an XPages UI that you can reach using a URL like: http://localhost/DarwinoContacts.nsf

Then, from the header bar of the XPages application, click the 'Admin' option and generate contacts documents by hitting 'Create 500 fake contacts'.

Once done, you can setup the synchronization between the Domino database and the Darwino one. For this, launch the synchronization admin UI, running on your Domino server: http://localhost/DarwinoSync.nsf

First, create an adapter: name it `contacts` to match the Darwino database name. Generate the definition by clicking 'Generate From Database' and selecting DarwinoContacts.nsf.



Adapter Definition

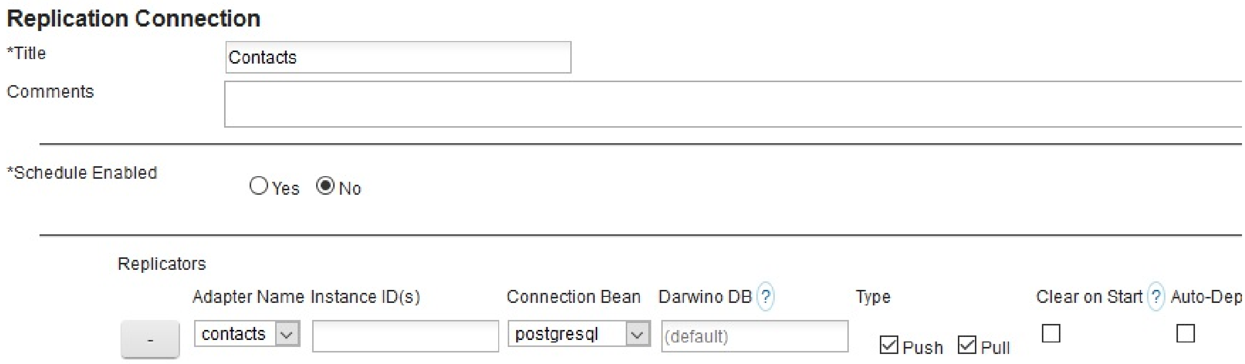
*Adapter Name: contacts

This serves as both the name of the adapter and the Darwino da

Definition

```
nsfName 'DarwinoContacts.nsf'

form('Contact') {
```

Once done, create a schedule service for this adapter and the proper Darwino connection bean:



Replication Connection

*Title: Contacts

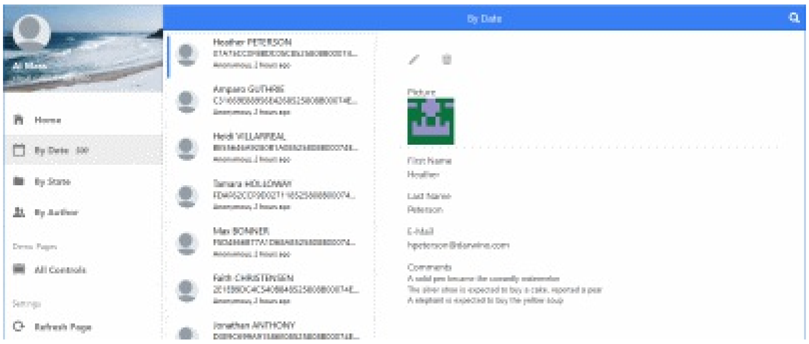
Comments:

*Schedule Enabled: ○ Yes ● No

Replicators

	Adapter Name	Instance ID(s)	Connection Bean	Darwino DB ?	Type	Clear on Start ?	Auto-Dep
-	contacts		postgresql	(default)	☑ Push ☑ Pull	□	□

You can schedule the replication, or execute it once from the main replication service view. After the replication is run, then the Darwino application should show the data:



Picture

First Name
Heather

Last Name
Peterson

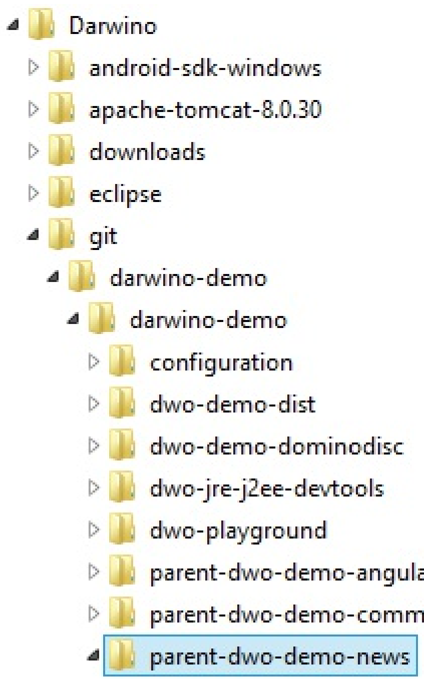
E-Mail
hpeterson@darwino.com

Comments

The News applications are an example of a syncing news-reader app with randomly-generated article content. It contains a J2EE application with two UI examples and which can act as a remote server for its mobile applications. It also includes pairs of examples for both Android and iOS: native UIs and hybrid (HTML-based) UIs.

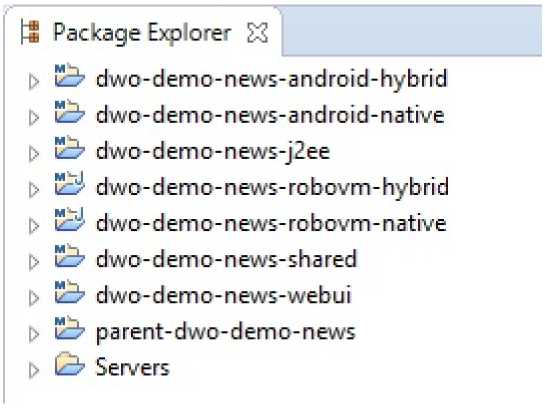
Installing the news application in Eclipse

The new application is provided by the darwino-demo Github repository as a set of maven enabled projects. To import them in Eclipse once the Git repository has been cloned, select `Import...->Existing Maven Projects` and choose `parent-dwo-demo-news` .



```
Darwino
  android-sdk-windows
  apache-tomcat-8.0.30
  downloads
  eclipse
  git
    darwino-demo
      darwino-demo
        configuration
        dwo-demo-dist
        dwo-demo-dominodisc
        dwo-jre-j2ee-devtools
        dwo-playground
        parent-dwo-demo-angular
        parent-dwo-demo-commons
        parent-dwo-demo-news
```

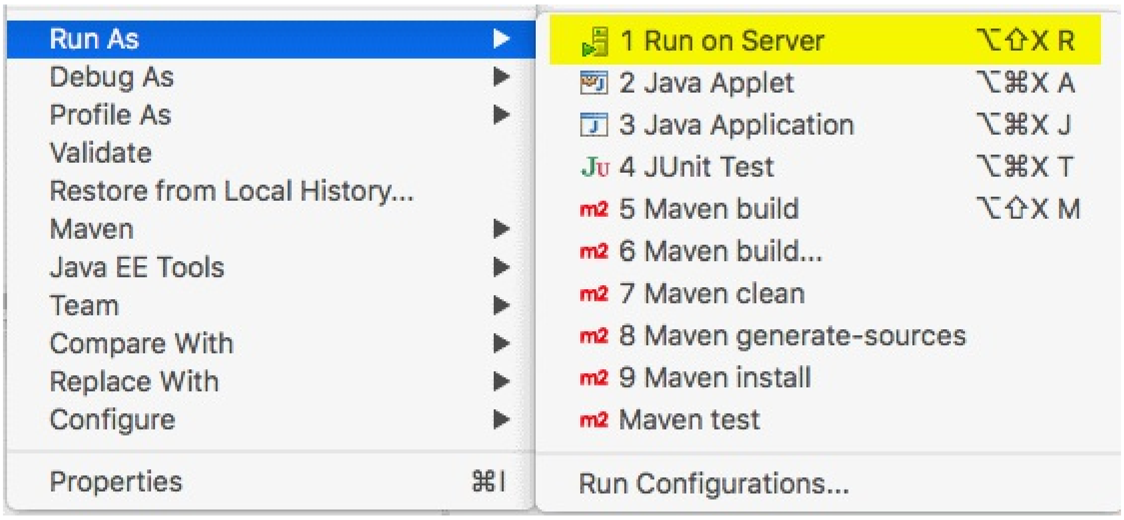
The result should be a set of project in your workspace:



```
dwo-demo-news-android-hybrid
dwo-demo-news-android-native
dwo-demo-news-j2ee
dwo-demo-news-robovm-hybrid
dwo-demo-news-robovm-native
dwo-demo-news-shared
dwo-demo-news-webui
parent-dwo-demo-news
Servers
```

J2EE Application

The J2EE application is named `dwo-demo-news-j2ee` . To run it using a configured Tomcat server inside Eclipse, right-click on the project and choose Run As → Run On Server:



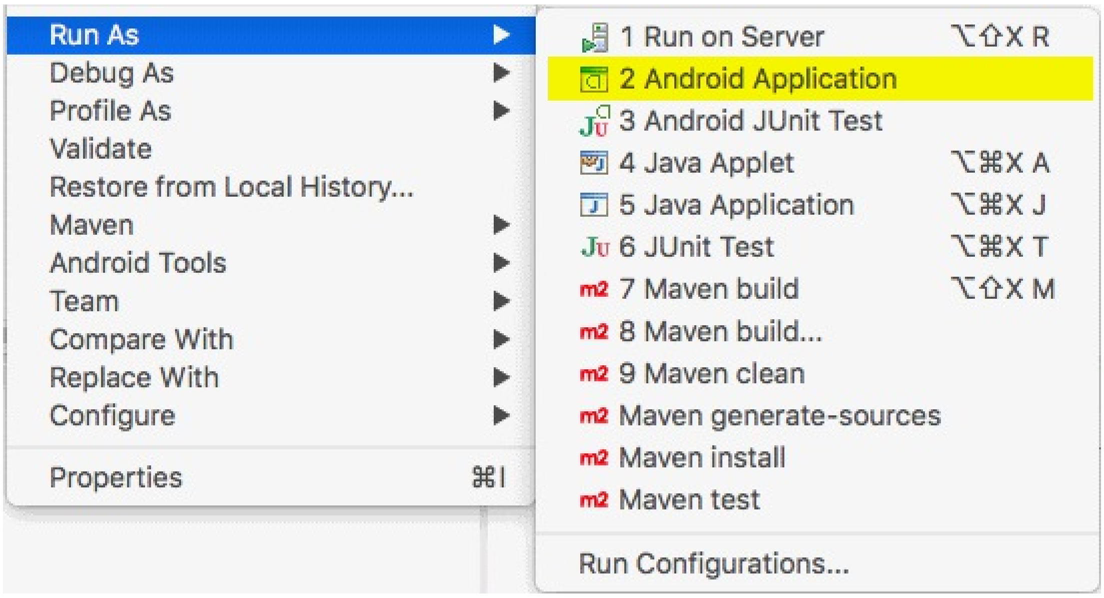
This will launch the app in Tomcat and open up the default landing page for the application. In addition to the default JSF-based UI, there is also an Angular-based UI, which is shared with the hybrid mobile applications.

This application can serve as a central server for replication with the mobile demo apps.

It connects to PostgreSQL for its data storage. By default, it uses the port and credentials specified on the [[Preparing the Development Environment]] page. If needed, this can be modified by changing either the `src/main/resources/darwino_default.properties` file in the `dwo-demo-commons` project (which affects the other demos as well) or by creating a copy of this file named `src/main/resources/darwino.properties` in the `dwo-demo-news-j2ee` project itself.

Android Applications

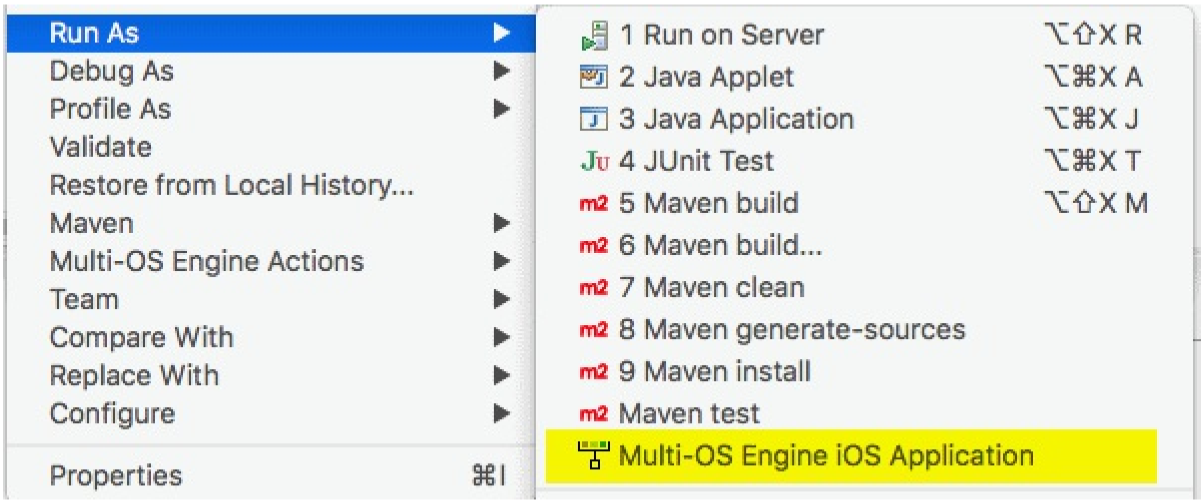
The two Android applications are `dwo-demo-news-android-hybrid` and `dwo-demo-news-android-native`. To run either of these apps, right-click the project and choose Run As → Android Application:



iOS Applications

If you are running on a Mac, you can run the iOS applications, which are `dwo-demo-news-moe-native` and `dwo-demo-news-`

`moe-hybrid`. To do so, right-click on the project and choose one of the Run As → iOS Simulator App options:

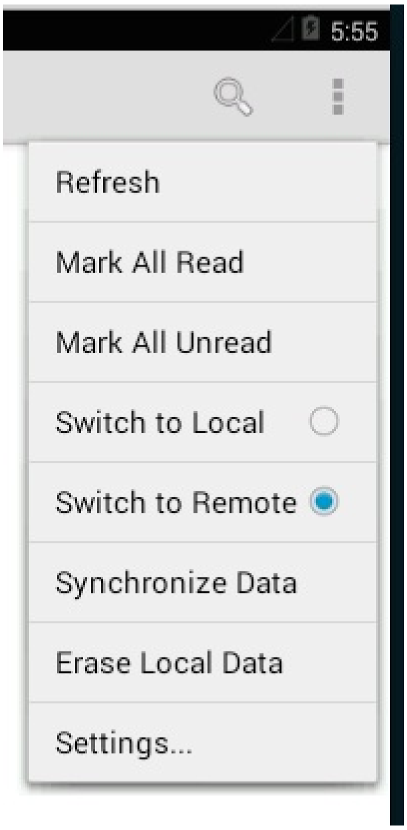


Connecting to the J2EE App from Mobile

To connect to a running instance of the J2EE application from either the Android or iOS applications, modify the connection settings in the `src/main/resources/predefined_connections_default.json` file in the `dwo-demo-commons-mobile` project, replacing `{0}` with the base URL of your Tomcat server in the first settings block. For example, where the Tomcat server is "10.0.1.8" and running on port 8081:

```
{
    "enabled": true,
    "local": false,
    "name": "Local Server - http://10.0.1.8:8081",
    "url": "http://10.0.1.8:8081",
    "userId": "amass",
    "password": "floflo",
    "dn": "cn=al mass,o=triloggroup",
    "cn": "Al Mass"
}
```

These settings can also be modified from within the mobile applications when running by going to the settings for the app, which is available by tapping "Settings" in the bottom-right corner of the iOS native app and in the top-right drop-down menu otherwise:

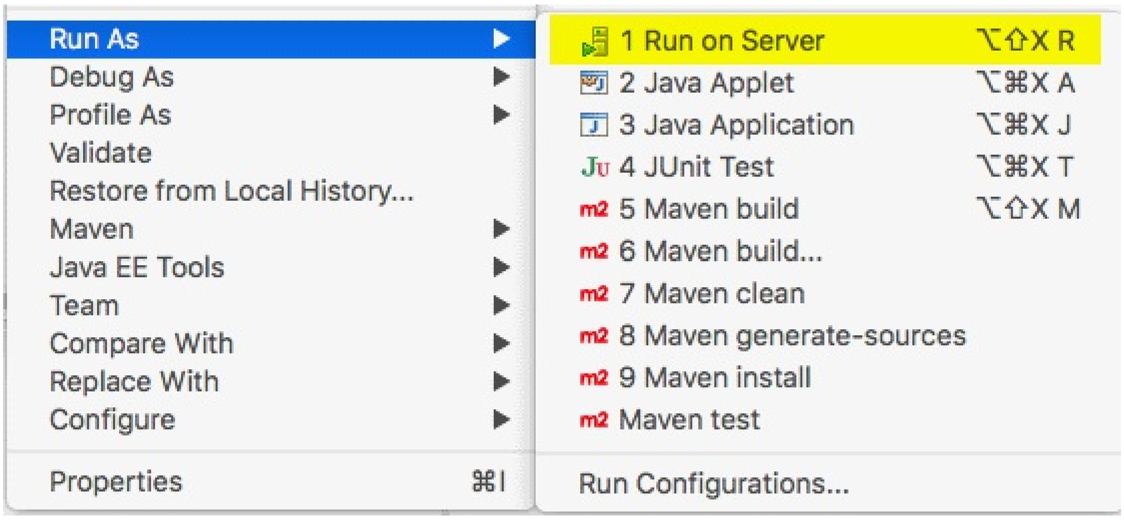


5:55
Refresh
Mark All Read
Mark All Unread
Switch to Local ○
Switch to Remote ●
Synchronize Data
Erase Local Data
Settings...

The Domino Discussion applications are an example of replicating with a standard Domino NSF-based discussion database.

J2EE Application

The J2EE application is named `dwo-demo-dominodisc-j2ee`. To run it using a configured Tomcat server inside Eclipse, right-click on the project and choose Run As → Run On Server:



This will launch the app in Tomcat and open up the default landing page for the application.

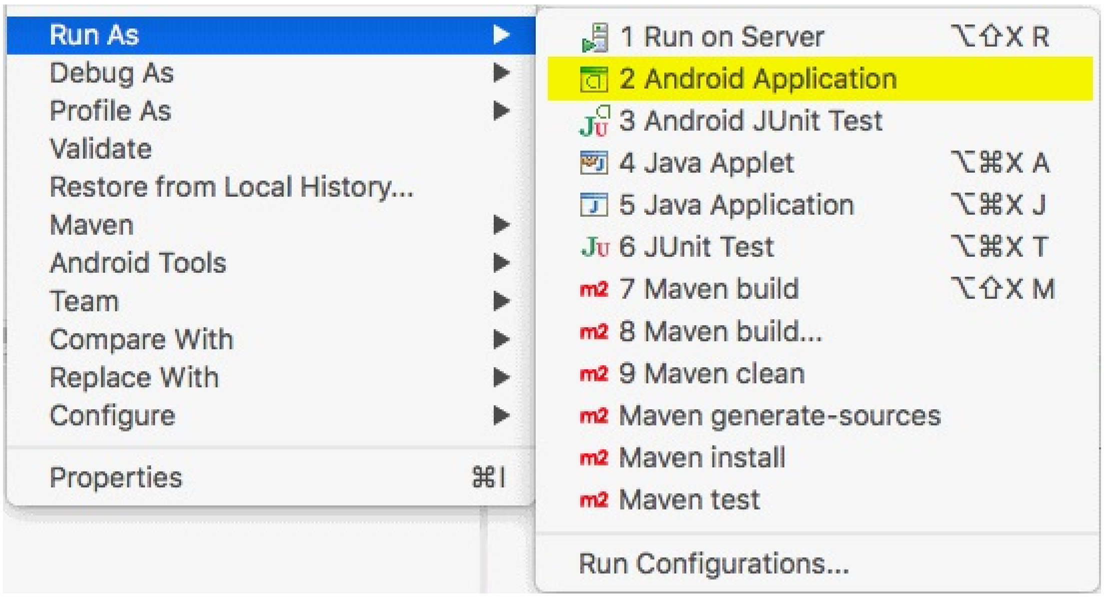
This application can serve as a central server for replication with the mobile demo apps.

To configure its connection to a Domino server running the Darwino sync service, which is required for proper usage, modify the `src/main/webapp/WEB-INF/web.xml` file in the `dwo-demo-dominodisc-j2ee` project.

It connects to PostgreSQL for its local data storage. By default, it uses the port and credentials specified on the [[Preparing the Development Environment]] page. If needed, this can be modified by changing either the `src/main/resources/darwino_default.properties` file in the `dwo-demo-commons` project (which affects the other demos as well) or by creating a copy of this file named `src/main/resources/darwino.properties` in the `dwo-demo-dominodisc-j2ee` project itself.

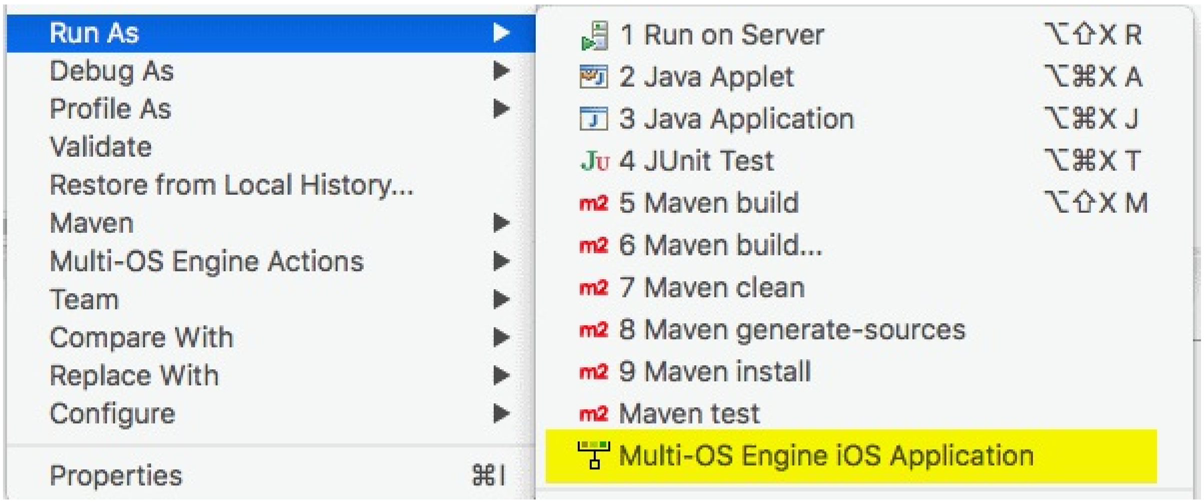
Android Application

The Android application is `dwo-demo-dominodisc-android-hybrid`. To run this app, right-click the project and choose Run As → Android Application:



iOS Application

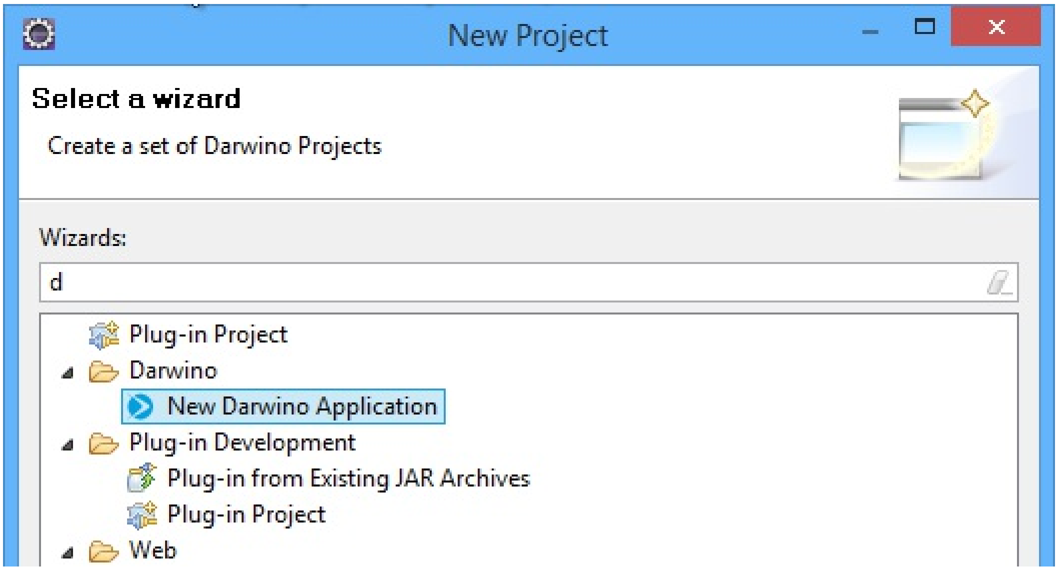
If you are running on a Mac, you can run the iOS application, which is `dwo-demo-dominodisc-moe-hybrid`. To do so, right-click on the project and choose one of the Run As → iOS Simulator App options:



Although Darwino does not require the use of any IDE to work, the Darwino studio is the fastest way to get started with a new project.

Using the Darwino Application Wizard

From the Eclipse main menu. select File...New...Project... and the search for Darwino:



New Project

Select a wizard

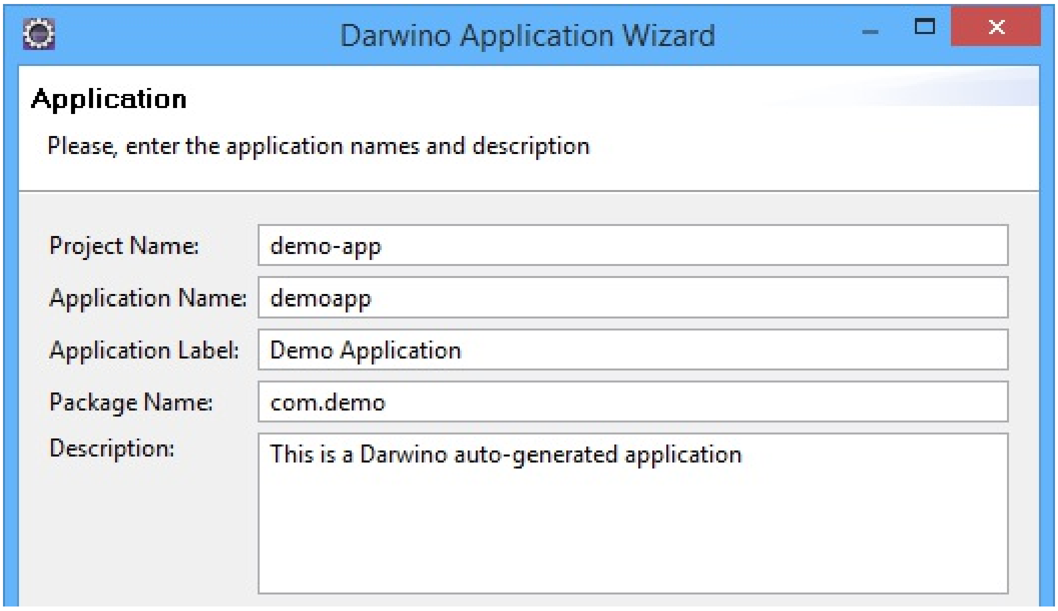
Create a set of Darwino Projects

Wizards:

d

Select 'New Darwino Application' hit Next.

For your first project, you can live the default values as is



Darwino Application Wizard

Application

Please, enter the application names and description

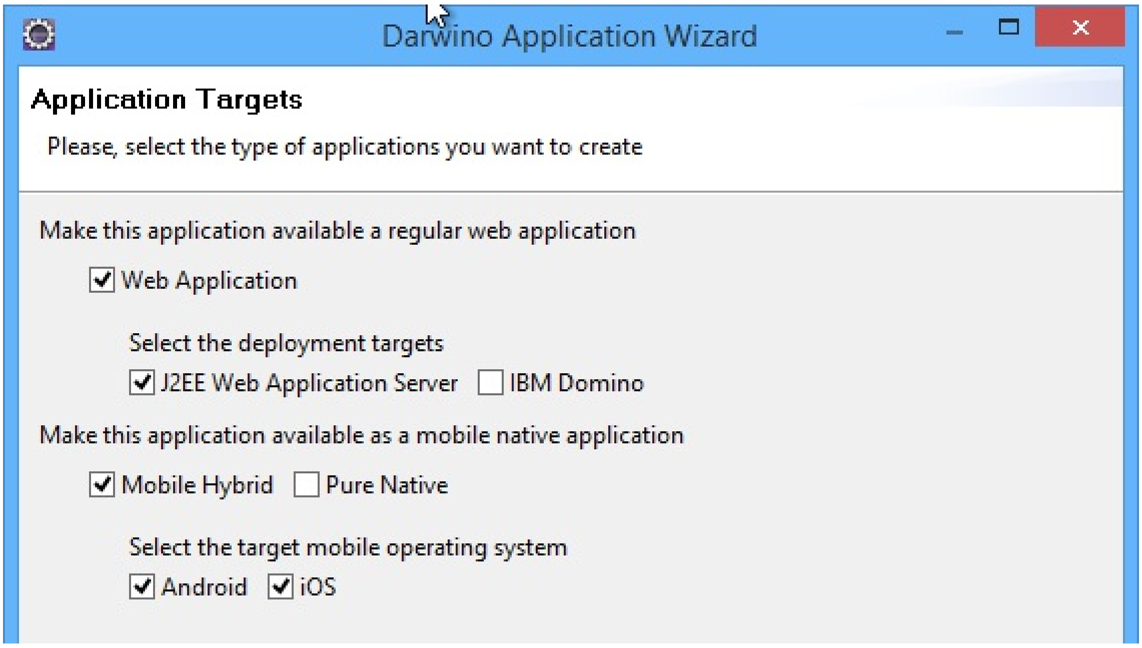
Project Name: demo-app

Application Name: demoapp

Application Label: Demo Application

Package Name: com.demo

Description: This is a Darwino auto-generated application



Application Targets

Please, select the type of applications you want to create

Make this application available a regular web application

☑ Web Application

Select the deployment targets

☑ J2EE Web Application Server □ IBM Domino

Make this application available as a mobile native application

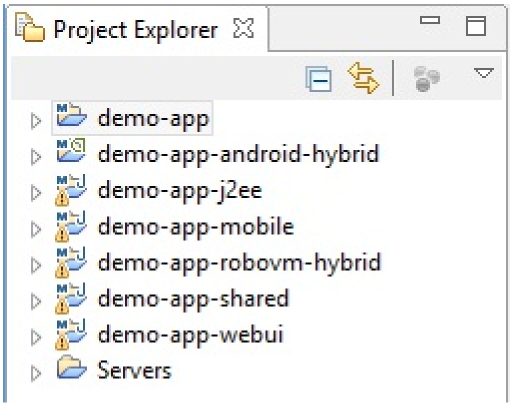
☑ Mobile Hybrid □ Pure Native

Select the target mobile operating system

☑ Android ☑ iOS

Note: the first generation might take several minutes, as Eclipse is downloading all the necessary dependencies from the maven repositories. Subsequent runs of the wizard will be a lot faster :-)

This will create a Darwino Application that can run on a J2EE server, an Android mobile device and an iOS mobile device. To support all these platforms while sharing code, the wizard generated several projects in the Eclipse workspace:



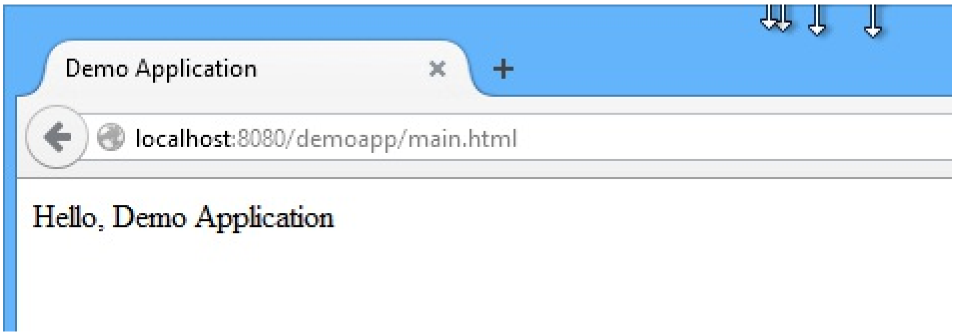
- demo-app
- demo-app-android-hybrid
- demo-app-j2ee
- demo-app-mobile
- demo-app-robovm-hybrid
- demo-app-shared
- demo-app-webui
- Servers

Running the Web Application

If you're using a TOMCAT server, then you should add it in Eclipse from the 'Servers' view. Right click on the server, select 'Add and Remove...'. Select the new application, add it to the list of configured applications and hit 'Finish'.

Then, right click on the server, select 'Publish'. Finally right click another time on the server and hit 'Start'.

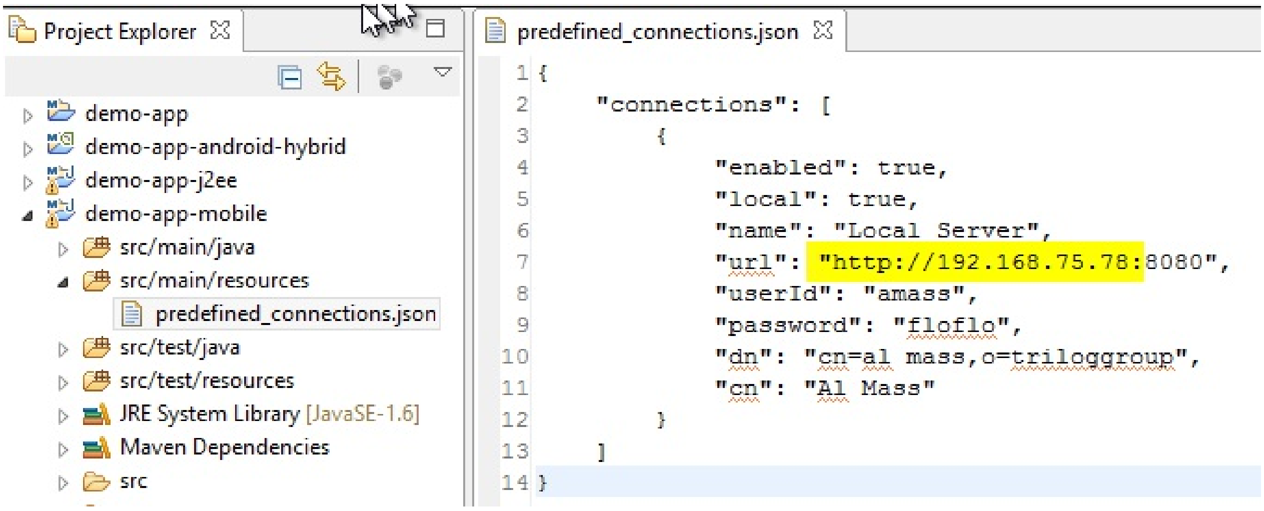
You should be able to hit the application from your browser by using a URL like: http://localhost:8080/demoapp/main.html



Hello, Demo Application

Running the Mobile Applications

Before running the mobile applications, you have to configure the server URL that the applications are using. This is not required, but heavily recommended to avoid every single user to have to configure the mobile application. To do this, open the predefined_connections.json file and ensure that the server address is available from the device or emulator running the application:



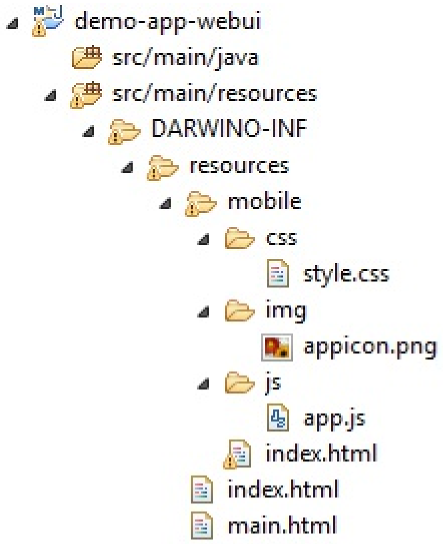
```
{
    "connections": [
        {
            "enabled": true,
            "local": true,
            "name": "Local Server",
            "url": "http://192.168.75.78:8080",
            "userId": "amass",
            "password": "floflo",
            "dn": "cn=al mass,o=triloggroup",
            "cn": "Al Mass"
        }
    ]
}
```

Then run the app by right clicking the application and select 'Run As... Android Application' or 'Run As...iOS xxx' depending on your platform.

Extending the Generated Application

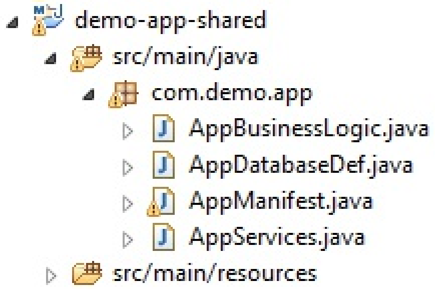
There are different places where the application can be extended

- Developing the UI A simple AngularJS based UI is located in the webui project:



```
demo-app-webui
    src/main/java
    src/main/resources
        DARWINO-INF
            resources
                mobile
                    css
                        style.css
                    img
                        appicon.png
                    js
                        app.js
                    index.html
                index.html
                main.html
```

- Adding business logic Business logic can be added to the shared project



```
demo-app-shared
    src/main/java
        com.demo.app
            AppBusinessLogic.java
            AppDatabaseDef.java
            AppManifest.java
            AppServices.java
    src/main/resources
```

And of course, everything can be customized!

Installation

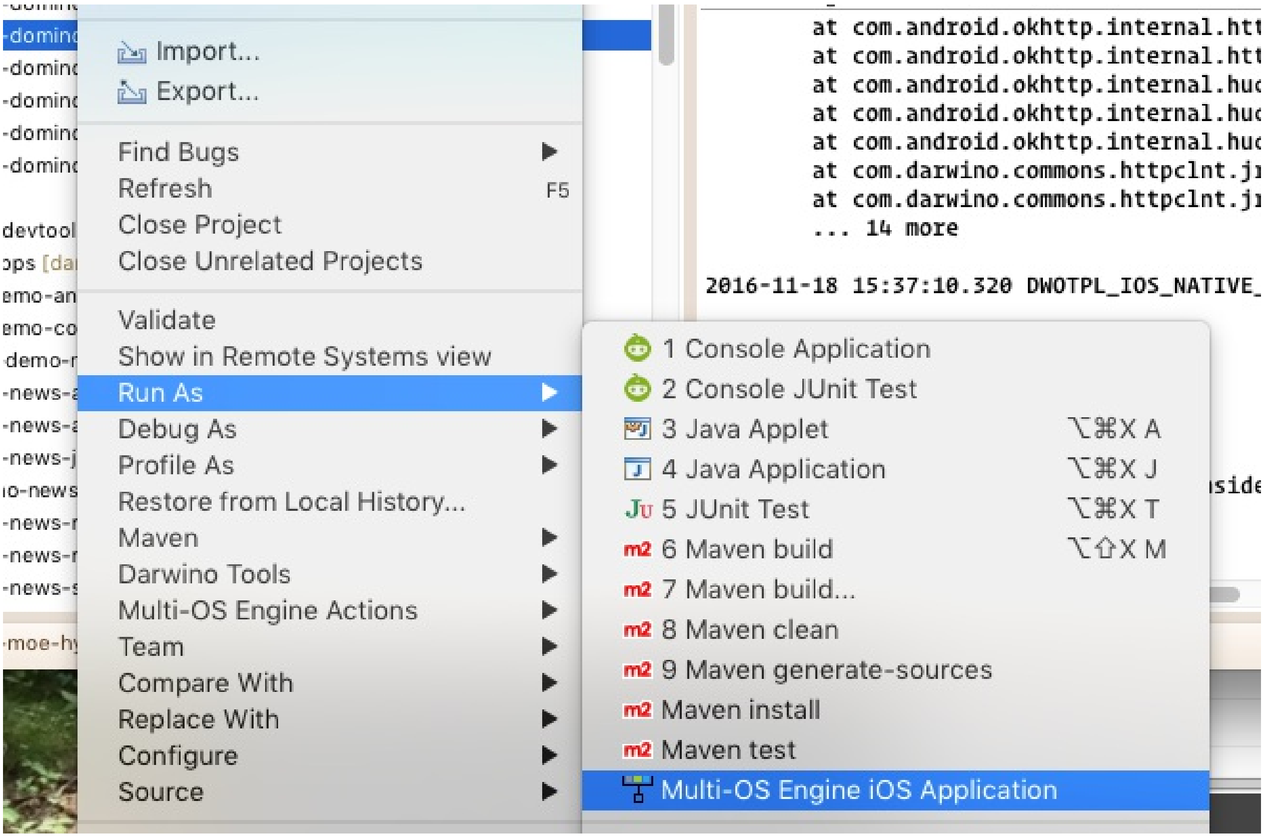
Building the iOS demo application requires a Mac with a valid iOS developer profile registered via Xcode's Accounts preference panel. To see if your computer has any active profiles, run:

```
security find-identity -v -p codesigning
```

That should list any available signing profiles, and valid ones will contain "iPhone Developer" or "iOS Development".

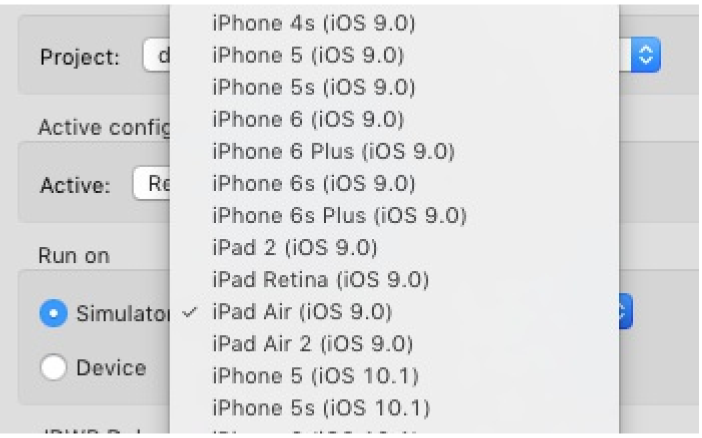
Running in the Simulator

To run iOS applications in the simulator, right-click the iOS project in Eclipse and choose Run As → Multi-OS Engine iOS Application:



```
        at com.android.okhttp.internal.htt
        at com.android.okhttp.internal.htt
        at com.android.okhttp.internal.huc
        at com.android.okhttp.internal.huc
        at com.android.okhttp.internal.huc
        at com.darwino.commons.httpclnt.jı
        at com.darwino.commons.httpclnt.jı
        ... 14 more

2016-11-18 15:37:10.320 DWOTPL_IOS_NATIVE_
```

To change the device or OS version used in the simulator, choose "Run Configurations" and select a different target device from the list:



Running on an iOS Device

Running an application on a physical iOS device can be done by adjusting the Run Configuration as above, but instead choosing a physical device from the list.