# TownWizard Android Application Build Instructions

This document provides information on how to build and test TownWizard android container and partner applications.

## Basic Procedure

Assuming all prerequisites (described below) are met, building the applications is a matter of going through the following steps:

* Pull the release branch from the git repository
* Add image files for the desired partner to the **build** directory
* Optionally edit **build/partners.ini** file in order to specify partner application and package names
* Run **./build.php** <partner\_id> command
* Test in an Android emulator or physical phone (optionally)
* Push the changes to the **partners.ini** file and added graphics to the git repository

## Prerequisites

The following conditions must be met in order for the build to work.

* Android SDK installed (as of now it should support level 8 (android 2.2) API)
  + Check: adb version
* Eclipse with Android plugin installed (optional but highly desired)
* Ant installed
  + Check: ant –version
* Git installed
  + Check: git –version
  + git config --global –edit (this command should show the git global configuration file with the user name and email in it. If it doesn’t show anything create the new config with git config command)
* The TownWizard key store located at:  
  **~/.android/TownWizard**

## Pulling the Source Code

When first time, clone the repository with these commands

cd <some\_projects\_directory>

git clone –b release <https://github.com/TownWizard/androidV2.0.git>

cd <project\_directory>

Also, for the first time only, it is necessary to tell the build the location of android tools in the system. This is done by creating a **local.properties** file under both town-container-android and facebook projects. This is done by executing the following commands:

cd <project\_directory>/town-container-android

android update project --path .

cd ../facebook-…/facebook

android update project --path .

For all other cases, just pull the latest code from the repository as:

cd <project\_directory>

git status (make sure it says, that there is nothing to commit, the working directory is clean, and you’re on the release branch)

git pull

## Updating Partners Information (optional)

Under the source code tree, there is a **build/partners/partners.json** file which contains all partners’ information in a JSON format. Here, you can find a partner id, name, and package name (which is usually the same as partner url backwards, like com.destin).

If no partner exists for the name you’re looking for, this file can be updated by running a PHP script provided as:

cd <project\_directory>/town-container-android/build

./update\_partners.php

Once this is done, the updated file should be pushed to the git repository along with other changes.

## Editing Partner Configuration File (partners.ini)

If necessary, that is when building a partner app, and the partner information in the **partners.json** file is different from desired, it is possible to override this information by placing an entry into the **build/partners/partners.ini** file. The format is show in the example below (not that name and package are optional entries, and the value in the brackets is the partner id):

[1]

name = Destin

package = com.destin

[20]

name = Pcola

## Adding Graphics

For each partner, there should exist a directory with the name of the partner id under **partners/** directory (like partners/1, partners/24 etc). In here, you can mimic the layout of the Android **res** directory and place the images here.

For example, for partner 24 there could be two files located as:

**build/partners/24/drawable-mdpi/icon.png  
build/partners/24/drawable-mdpi/splash.png**

During the build these images will be will be copied under the original Android **res** directory and will replace the original icon.png and splash.png.

## Building the App (both container and partner)

This should be as simple as running

cd <project\_directory>/town-container-android/

./build.php <partner\_id>

If the build is successful, it will create a new APK file under **town-container-android/deploy** directory. This file is digitally signed and ready to be published.

In order to build a container app, supply 0 as partner\_id, and the it will create an APK file named **TownWizard.apk** in the build directory.

IMPORTANT: if the build fails for some reason, before taking any action, clean up the project by running

ant clean

## Installing and Testing

This section assumes that Android emulator is configured and is running, or a physical Android phone is connected to the system.

To make sure that Android SDK knows about available devices, execute:

adb devices

This should show available devices.

To install a partner of a container app on the device, run:

cd <project\_directory>/town-container-android/deploy

adb install –r <apk\_file\_name>

The –r switch is optional and means “replace existing application”.

Note, that it is also possible to install the application during the build step by using –install switch (again, assuming the emulator or phone is available) as:

./build.php <partner\_id> -install

This can be useful when doing repeated installations for testing.

## Pushing Updated Project to the Repository

This is an important step and should not be skipped. The git commands to do it can vary, but most of the times this is what should be done:

**git status**  
 Observe the changes and make sure everything is looking good.

**git add** <resource>  
This command will place updated/deleted/new files into so called git stage area. It should be executed several times until all the updated resources are not added to the “ready for commit” section. For example, the change of command could look like (assuming you’re at the porject’s root):

*git add townwizard-container-android/build/partners/partners.ini  
git add townwizard-container-android/build/partners/partners.json  
git add townwizard-container-android/build/partners/1*

Since shortcuts are available, most of the times you would probably need just one command in order to add all the changes:

*git add townwizard-container-android/*

**git commit** –m “Added ‘Pensacola’ configuration and images”This will commit the changes.

**git status**Verify that the changes have been committed and the working directory is clean.

**git push**Naturally, pushes your local changes to the remote repository.

And this is all I have to say about building it.

## Appendix:

## Building with a Different Key store File for Old Partners

Before the build, modify **townwizard-container-android/ant.properties** file. Comment out the lines

key.store=${user.home}/.android/TownWizard  
key.alias=townwizard

and replace with the different values. After the build, undo the changes, and do not commit the changes to the repository.