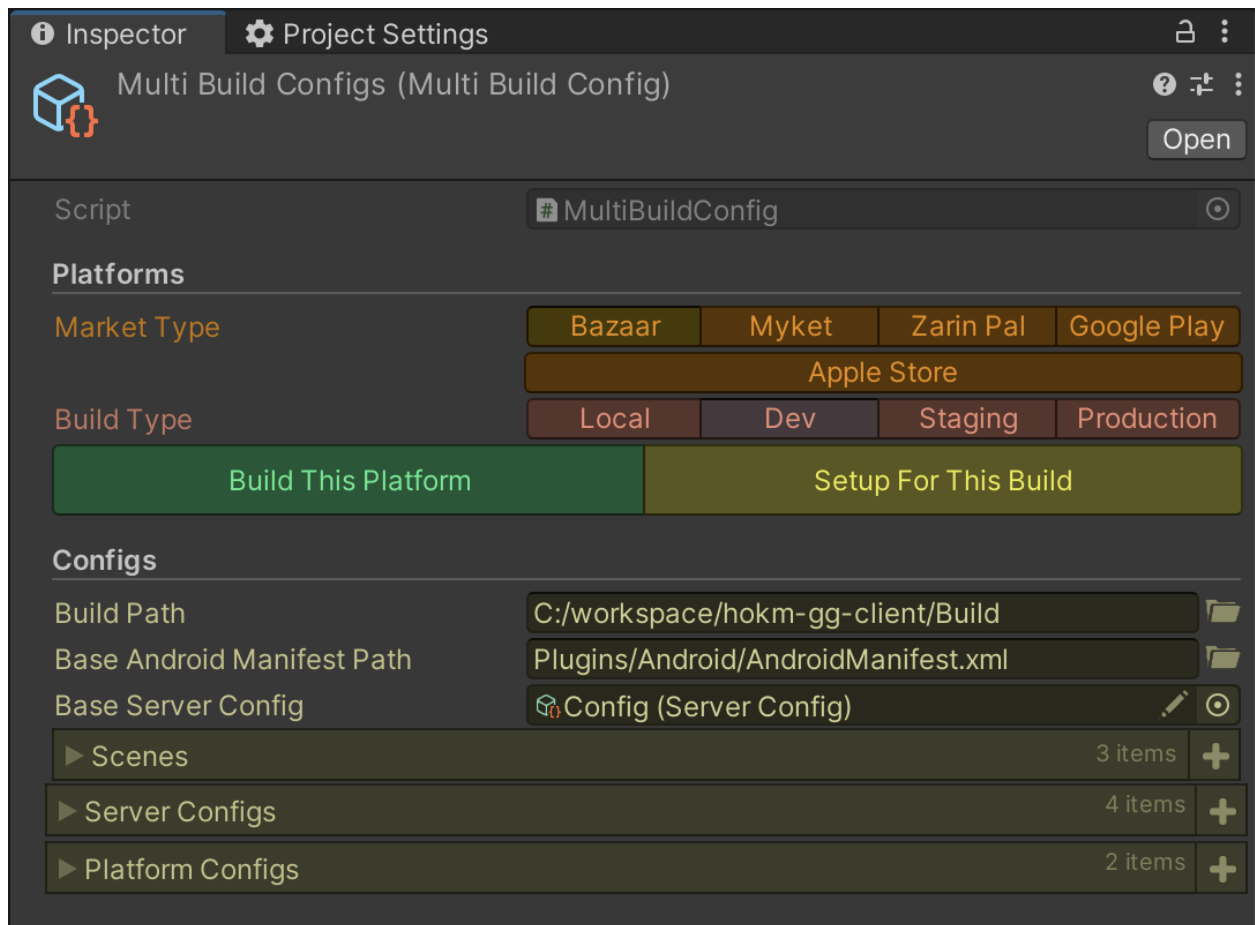


MultiBuild Tool

In HakemSho, a tool for easy building for multiple platforms is named “MultiBuild”.

This tool is designed specifically for HakemSho and can be transferred to another project with some minor configuration. Although, Most parts are general and can be used in every project.

In the next chapter, we will talk about all the parts and see if any of them need change for Other projects.



Parts of the MultiBuild Tool

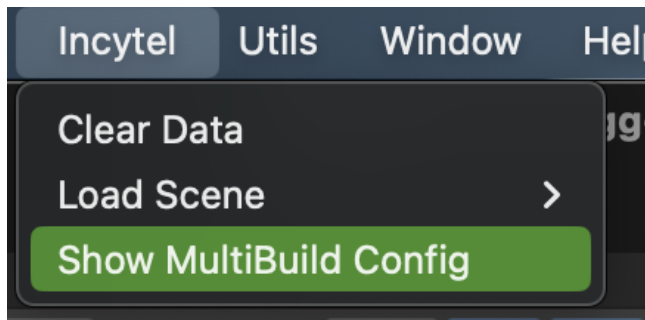
1. Access the Config

The config is a scriptable object located in the “Resources” directory.

If the config has not been created before, you should create it by right-clicking on the “Resources” folder -> Create/Incytel/Create Build Config

Or

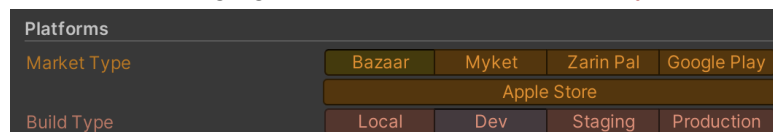
Just use the shortcut created in the tabs. Incytel/Show MultiBuild Config



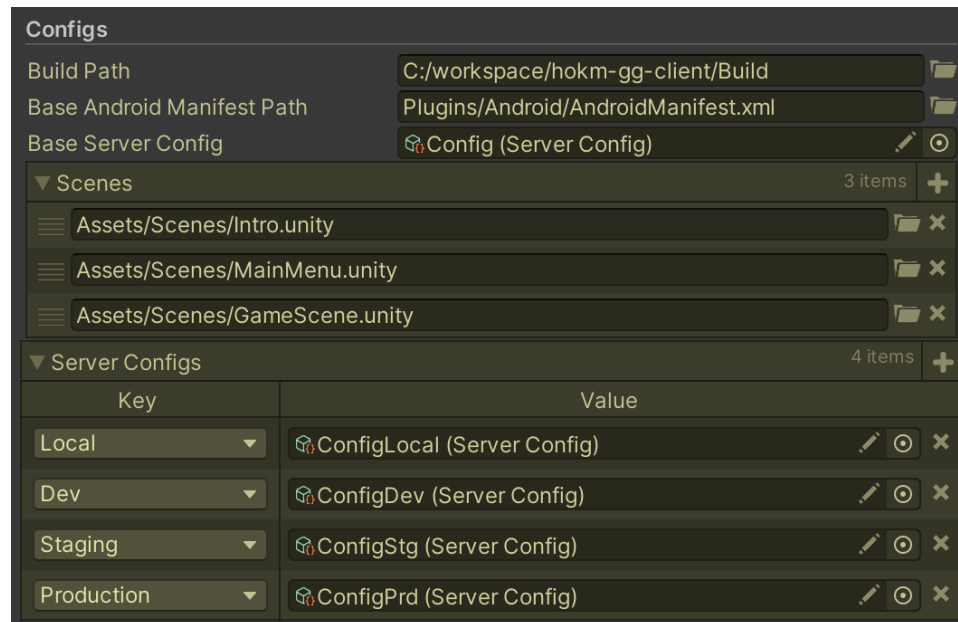
2. Configuration

Every store has its own configuration but there are common parts between all builds.

- a. At the top of the config, you can see the Market Type and the Build Type
 - i. In the market type section, you will choose the market to which the build will upload in there. These market types are set in the server too, so change it if yours are different.
 - ii. In the build type section, you should choose the server config of the build. (Local, Dev, Staging, and Production) **[specifically for the hakemSho]**

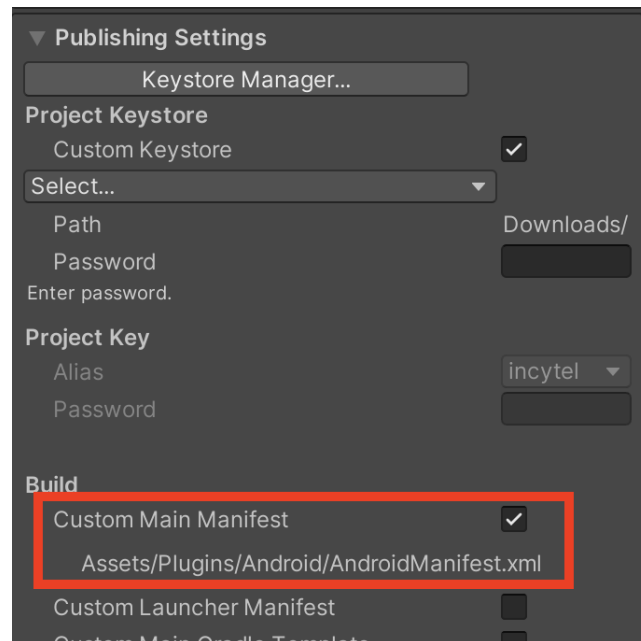


b. Configs section:



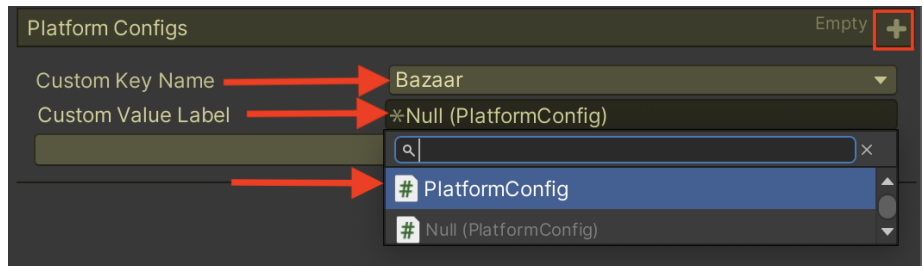
- i. Build Path: Choose the folder you want to place the build.
- ii. Base Android Manifest Path: Just drag and drop your main Android manifest to fill the path automatically. This part will be used for Myket build.

If you cannot find the android manifest, Check “Custom Main Manifest” in the project setting.



- iii. Base Server Config: As mentioned before, this server config is for HakemSho and must be changed in your project.[specifically for the hakemSho]
- iv. Scenes: Drag and drop all scenes you want to include in the build.

- v. Server Config: As mentioned before, this server config is for HakemSho and must be changed in your project.[specifically for the hakemSho]
- vi. Platform Configs: To add a new config
 1. Use the plus button at the right.
 2. Choose Market Type
 3. Click on PlatformConfig and create a new config



4. Click on add to add the config to the Dictionary

vii. Platform Config Details:

1. Everything is clear but I explain some parts:
 - a. Build with App Bundle will create aab instead of apk
 - b. Android Manifest path is the path of this build manifest, so you need to have multiple Android manifest templates for each build and drag and drop them into this field. This will replace the “Base Android Manifest Path” with the template automatically.
 - c. The store key is a key from markets to initialize their SDK.
 - d. Game Analytics should be in the project if you want to use the Game Analytics Config feature. If you don’t need it, remove the dependent code from the package.
 - e. Define Symbols To Add:

Imagine you want to use specific symbols in your code, By defining them in the script and choosing them in this section, the symbols automatically add/remove to/from the project.

⚠ The market type will add to the symbols automatically, so no need to define them in this section.

f. Include Folders:

In each build, some part of the codes should be compiled which should not exist in other builds. For example, if you want to build for Myket, some part of cafe bazaar SDK should not be in the build, and the same way for cafe bazaar. So you add the folders you want to include and these folders will add to the build and will exclude in other market's builds.

⚠ Remember to "Force Resolve" each time you change the Market type.

⚠ For now, These folders should be included in bazaar build:

Plugins/Android/Bazaar

Plugins/Vendor SDK/Bazaar/Poolakey/Scripts/Editor

And this should be included in myket build:

Plugins/Android/Myket

3. Main Functions

a. Setup For This Build:

Will set **everything** in the config based on the selected market and build type.

b. Build This Platform

Will build and put it in the chosen path. No need to build from the Build setting anymore.



4. Purchasing

The main class for purchasing is "Purchase Manager".

- Inject *IMultiBuildConfig* to this object.
- Use Buy() -> Consume() as you wish, But remember to call GetPendingPurchases() -> Consume() in the proper places (Awake of the shop and OnApplicationFocus of the shop).
- The myket or cafebazaar SDK will initialize based on the market type you choose on the config.