Asset Management - Database, Packages, and Local Files

Loading Assets

Database

BeardLib can load Assets for your level in different ways. Currently the standard is to load assets from database, also known as DB Loading. BeardLib handles this process automatically when spawning a unit and there is next to no other step required. ble_assets_spawn.jpg

However you can also manually load assets with the **Load From Database** option. ble_assets_loaddb.png

Package

The old method was Package loading. When spawning a Unit, you had to load a package containing said Unit. The downside is that a package also contains lots of other units, taking up memory. In some cases like sound effects or voice lines you may still need to load some, just remember to pick the smallest available packages. ble_assets_package.jpg

You can load packages with the **Load With Packages** option. ble_assets_packs.png

Local Files / Extract

The third method is Local Files, or extract loading. This means the Unit's files are actually present inside your Map's assets folder. ble_assets_folder.png

It was used in combination with Package loading to reduce the memory usage of a heist. The downside is that since the files had to be extracted, the space the Map folder takes on your harddrive went up the more Units you load this way, resulting in longer download times. These days Local files are only really used for custom assets.

While extract loading has been removed from the editor entirely, and there isn't really a reason to have do it anymore, you can still include the files into your project with the following method:

Use the **Load From Database** option, but with the "With Options" toggle on.
ble_assets_extract.png

After picking your asset in the list you now get the option to include the files in your project.
ble_assets_extract_popup.jpg

Dependencies

Some Units are dependent on other assets, like Effects, and those sometimes don't get loaded properly. In this case you need to manually load them.

You can see the dependencies of a unit by extracting its .unit file using [Diesel Bundle Viewer](#), opening the file in a text editor and checking for a `<dependencies>` tag.
ble_assets_dependencies.png

Those assets need to be loaded for the unit to work, otherwise the game might crash.

Assets Manager

You can manage your loaded assets and packages from the Assets Manager in the Main-Tab. There you can see a list of every package and a list of every Asset currently loaded in your level.
ble_assets_manager.jpg

You have several options on the right side of the screen:

Remove And Unload Unused Assets Will unload all assets that are loaded but not used anywhere. For example a unit that isn't actually spawned in the level.
ble_assets_actions_unload.png

This sometimes also unloads important assets such as dependencies for units, to prevent this from happening you can tag assets as Used.
ble_assets_select_mark.png

They will show up in the list with a green background and will not be unloaded with the Unload Unused Assets button. ble_assets_used.png

Package Report Will let you select any package from the game. When selecting it, **all** units from the package will be spawned into your level.
ble_assets_actions_report.png

Scan Assets Directory Will scan through your project's assets folder and add all files to the add.xml. ble_assets_actions_scan.png

Clean Add Xml will clean up your level's add.xml by removing duplicate entries.
ble_assets_actions_clean.png

Unused Assets

Assets that are loaded but not used are marked in Yellow. While building your level those can stack up really fast, since just clicking on a unit in the spawn list will already load it. Before releasing your map you should make sure to unload all units that are not used and not needed to reduce memory usage.
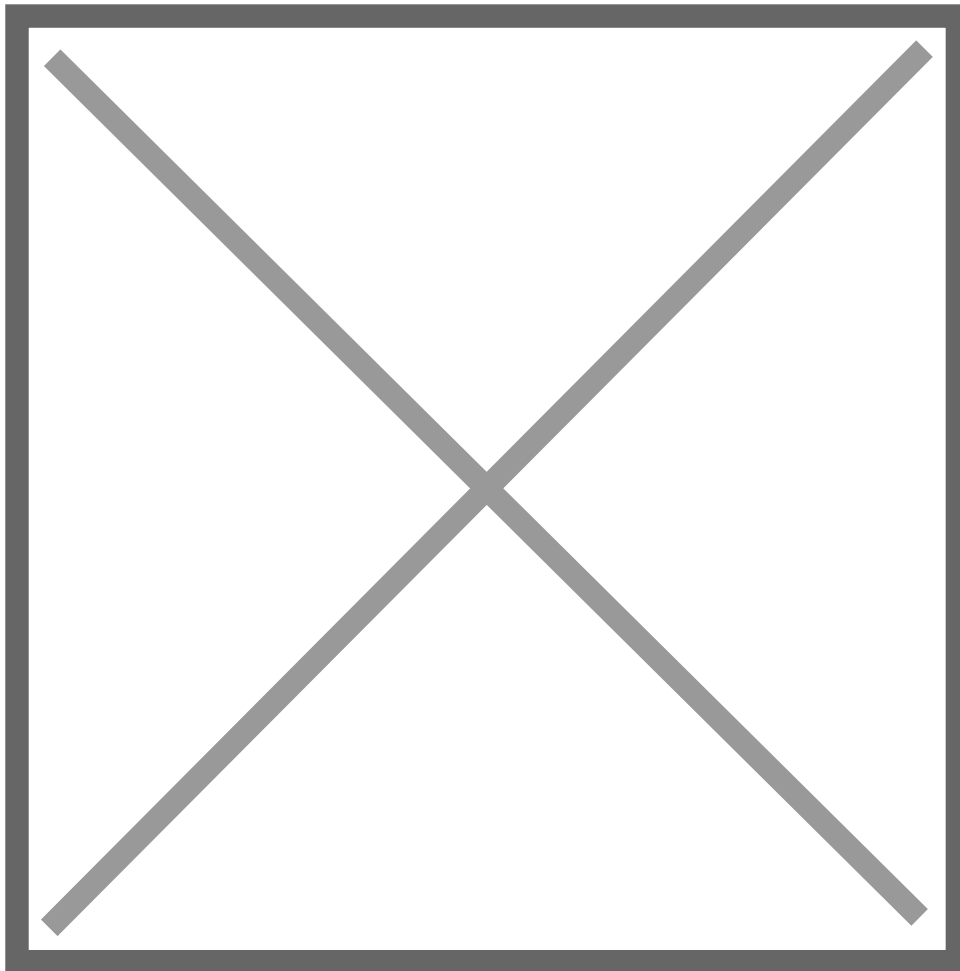ble_assets_unused.png

Unloaded Assets

Assets marked in Red are assets that are spawned in your level despite not being loaded.
ble_assets_unloaded2.png

Usually BeardLib automatically loads missing assets when opening your level and notifies you about them.

You can easily fix these with by either loading a package that's containing them, loading them directly from database ot just remove them from your level.
ble_asset_unload_action.png

add.xml

The add.xml tells BeardLib what assets are needed when loading into your level. It's unique for every level and can be found in `*Your Project*/levels/*your level*/add.xml`.
Assets loaded in-editor will be added here automatically, but you can always edit it manually using

a text editor.
ble_assets_addxml.png

Memory Usage

PAYDAY 2 has a theoretical limit of about **3.5GB** (**3500MB**), but usually you're gonna crash at around **3200**-**3300MB**. To keep your used memory as low as possible, you should load as few packages as possible and unload unused assets. Additionally assets that are only used inside your level, such as voicelines, interactions or hooks, should only be loaded while in your level. You can see how much memory your level is using with the -qa launch property. ble_qa_usage.png