

Projects

Creating, modifying and using Projects within UBuild.

- [Advanced Projects](#)
- [Supported Platforms](#)
- [Build Presets](#)
- [Advanced Build Presets](#)

Advanced Projects

Below are the more advanced configurations for UBuild Project.

Compression

UBuild can compress your builds to a single zip file. There are three options of compression:

- **Don't Compress** - This build will not be compressed.
- **Compress** - This build will be compressed.
- **Compress & Clean** - This build will be compressed, after which all files which have been compressed will be deleted. (*leaving only the compressed file*)

GitHub

UBuild can Fetch & Clone from Git, to achieve this follow the steps below:

1. Within the Project configuration file, update the following:
 - **ProjectFolder**: This is your Git repository's URL. (ie: the URL you would usually clone from)
 - **GitUser**: This is the user you would like to authenticate as.
 - **GitToken**: This is your legacy Git token that will be used during authentication.
 - **GitBranch**: This should be the branch you wish to use.
2. If all these details are correct UBuild can now clone the specified repository to the "/Git/" folder once you queue a build, after it has cloned, by default, it will now fetch before building to ensure the repository is up to date.
3. Optionally, within your selected preset, you can set "ForcePullFreshProject" as "true" to ensure a fresh repository is created for each build. (*this can be found within your Project configuration file*)

Steam

UBuild can integrate directly with Steam to publish your build to Depots & branches. This system uses SteamPipe and a Steam developer account is required to use this feature.

It is recommended that you create a separate, UBuild-specific Steam account **without mobile 2-factor authentication** as you'll need to provide the UBuild with a valid Steam user with access to upload builds for your Steam App via SteamWorks. This information is only stored within your local Project configuration and is not shared.

To upload your builds to Steam:

1. Within your Project configuration file, update the following:
 - **SteamAppID**: This is your Steam AppID.
 - **SteamUsername**: This is the Username of the user you would like to login with.
 - **SteamPassword**: This is the password you would like to use.
2. For each of your **SupportedPlatforms** you can specify which **SteamDepotID** to publish the build on.
3. Within your **BuildPreset** you can also specify which **SteamBranchName** to publish your build on.
(Note: This uses SteamWorks "SetLive" which will automatically publish the changes to the branch available, this isn't allowed for your default branch. ([read more](#)))

FTP

UBuild can upload completed builds, neatly to an FTP site of your choice.

UBuild will also mention the path to this build in email notifications. (if enabled)

To configure FTP follow the steps below:

1. Within the Project configuration file, update the following:
 - **FTPHost**: This is the remote site you would like to upload to.
 - **FTPUser**: This is the user with access to the above remote FTP site.
 - **FTPPassword**: This is the password associated with your FTP user.
2. Ensure that your selected Preset has "SendToFTP" as "true". (*this can be found within your Project configuration file*)

Email

UBuild has support for email notifications of build success & failure.

In its most basic form, this will allow you to receive remote notification of complete builds, but if expanded upon, allows you the option of further automation after the build process has been completed such as Jenkins jobs, forwarding to QA teams, triggering Power Automate scripts etc.

Commercial License holders can skip step 1. If you wish, as a Commercial License holder you can use the UBuild Emailing Service.

To set up email notifications, please follow the steps below:

1. Within the Project configuration file, update the following:
 - **SMTPServer:** This is your SMTP email server address (usually something like "mail.mydomain.com")
 - **SMTPPort:** This is the port at which your SMTP server is listening. (usually 587)
 - **EmailUser:** This is the email address to login with.
 - **EmailPassword:** This is the password associated with your EmailUser.
2. Within the Project configuration file, update the following:
 - **EmailTo:** This is the address to send emails to.
3. Ensure that your selected Preset has "SendToEmail" as "true". (*this can be found within your Project configuration file*)

Supported Platforms

Supported Platforms allow you to specify which platform you're building to, this is usually an operating system but, can also be a console.

Editing Supported Platforms

By default, you should have a "Windows" Supported Platform within your Project. This is configured for Unity currently but we can change this by modifying the values below:

- **Name:** This is how the Platform should be displayed within UBuild. (*this is also used to separate builds within the Presets build folder.*)
- **PlayerCommand:** This is the command passed to either Unity Or Unreal to specify which platform to build this for. For Unity, this is referred to as a Build Player (see Build Arguments [here](#)). For Unreal, this is referred to as a Build Platform, for Windows this is "Win64" (*Unreal forums got nuked so gl find sources for this* [link](#))
- **MethodToExecute:** This is exclusively for Unity. This is the method to execute before building. (*optional*)
- **ExecutableExtension:** This is the executable extension for this platform. (ex: *.exe/.so/.app/etc*)

Build Presets

Build Preset allows you to easily save pre-configured builds for later, this saves you from having to re-set up settings each time you would usually switch platforms within the engine. This can be used for particular configurations of builds within the same platform (ex: Steam/Epic/Non-DRM/etc)

Editing Build Presets

You can edit build presets from within the Project configuration file, you can also "on-the-fly" edit some preset settings above the "ADD TO QUEUE" button.

Remember that any changes made to the build preset within the UI are temporary and will not affect the saved build preset within the Project configuration file but, will affect any builds added to the queue while those changes are present.

You can edit Build Preset by updating these values within the Project configuration file:

- **Name:** This is how your Build Preset should be displayed within UBuild. *(this also is where to store builds within the build folder, containing folders for each Supported Platform)*
- **BuildFolder:** This is the folder in which builds using this preset will be placed.

Advanced Build Presets

Below are some advanced methods of configuring Build Presets. These settings can be found within the Project configuration file.

Pre-build

These are pre-build, Build Preset settings:

- **MethodToExecute:** Exclusively for Unity. This will execute this method before building.
- **AdditionalBuildArgs:** These are additional arguments to pass into the builder. (*either the UnityEditor or Unreal Automation Tool*) (*this is an array of strings*)

Post-build

These are post-build, Build Preset settings:

- **SteamBranchName:** This is the name of the branch you would like to upload your build to.
- **IncludeFilesFromFolder:** This is a directory which UBuild will copy all files from to include in the final build.
- **ExcludeFilesFromBuild:** These are files that will be deleted from the final build.
- **ExcludeFoldersFromBuild:** These are files that will be deleted from the final build.
- **DeleteBuildExceptZip:** This will determine whether UBuild will clear the final build folder of all files except the compressed file. (*this only applies if CompressToZip is true*)
- **CompressToZip:** This will determine whether UBuild will compress the final build to a zip archive.
- **SendToEmail:** This will determine whether UBuild will send email notifications for this build.
- **SendToFTP:** This will determine whether UBuild will send this completed build to the FTP site.
- **ForcePullFreshProject:** This will determine whether UBuild will force a fresh clone from Git before building. (requires Git configuration, see more [here](#))