

Overview

The following topics address the Cry WAF Cross Platform Build System. Based on the WAF Build System; credits go to Thomas Nagy, it is a powerful tool within the CRYENGINE.

Its flexibility allows you to quickly jump between various build pipelines while ensuring you only build what you really need to build.

Extensions such as automatic project generation or a simple GUI allowing you to tweak your projects requirements, ensure a fast turn over time and a more productive team.

In This Topic

- [Getting Started with WAF](#)
- [WAF File Structure](#)
 - [WAF Branch Spec \(waf_branch_spec.py\)](#)
 - [WAF Project File \(projects.json\)](#)
 - [WAF Spec File \(*.json\)](#)
 - [WAF Default User Settings \(default_settings.json\)](#)
 - [WAF Code Folder](#)
 - [WAF Module File \(wscript\)](#)
 - [WAF File List \(*.waf_files\)](#)
- [WAF Tutorials](#)
 - [WAF Adding a Project](#)
 - [WAF Adding a Spec](#)
 - [WAF Adding a Module](#)
 - [WAF Linking Modules](#)
- [WAF Interaction](#)
 - [WAF Interface](#)
 - [WAF Command-Line Interface](#)
 - [WAF GUI](#)
 - [WAF Build Options](#)
 - [WAF Commands](#)
- [WAF Extensions](#)
 - [WAF with IncrediBuild](#)
 - [WAF with Visual Studio](#)
 - [WAF with QT](#)
 - [WAF with Recode](#)
- [WAF References](#)
 - [WAF Supported Platforms](#)
 - [WAF Supported Compilers](#)
 - [WAF Supported Project Configurations](#)
- [WAF FAQ](#)