

Overview

CryENGINE uses LUA for its scripting language.

The Entity System can attach a script proxy to any entity. This appear in the form of a table that can include data and functions. The AI System has behaviors and GoalPipes written in scripts. Additionally, several game system such as the Actor, Item System, Vehicle System and the GameRules rely on scripting to extend their functionality.

Some of the advantages of using scripts are the following:

- Fast iteration - Scripts can be reloaded within the engine.
- Run-time performance - Careful usage of the available resources can result into scripts running nearly as fast as compiled code.
- Easy to use troubleshooting - An embedded LUA debugger can be invoked at any time.

Most of the engine systems expose ScriptBind functions that can be called with scripts.

Various systems in CryENGINE expose ScriptBind functions which allow Lua scripts to call existing code written in C++. The existing ScriptBind functions are listed in the reference section.

How to run scripts

Lua files

Scripts are usually stored in the `\Game\Scripts` directory.

The scripts will have to be invoked by calling the LoadScript function from within C++ code. More information can be found [TOPIC Integration Between Scripts and C++](#)

Another workflow is creating a script entity as explained in the [Structure of a Script Entity](#) topic.

Console

Script instructions can be executed within the in-game console. This can be done by appending the # character before the instructions.

This functionality is limited to Sandbox or running the launcher in Dev Mode (using the `-DEVMODE` command-line argument).

Reloading scripts during run-time

In Sandbox it's always possible to reload entities within the user interface. When selecting a Script Entity, this can be done by clicking on the Reload Script button found in the Rollup Bar.

It's also possible to use the following ScriptBind function to reload scripts.

- `Script.ReloadScript(filename)`
- `Script.ReloadScripts()`

Hint: it's always possible to invoke these functions from the console. The following syntax needs to be used.

```
#Script.ReloadScript("Scripts\\EntityCommon.lua")
```