



HOW TO APPLY MODIFICATIONS TO VANILLA .CFG FILES

Introduction

There are two distinct ways to apply modifications to vanilla .cfg files:

1. Using a .cfg_patch_* file.
2. Using a plain *.cfg file.

Let's inspect them in more detail.

Using a .cfg_patch_* file

In order to apply modifications to any non-prototype .cfg file, you need to create a corresponding .cfg_patch_*.

When loading such files, Zone Kit or the game will look for .cfg_patch_* files with the same filename in the same directory and apply them as patches.

Here's a table with references and examples:

Vanilla .cfg file to modify	Source .cfg files in Zone Kit	Packaged .cfg files in mod files
<code>Stalker2\Content\GameLite\GameData\Credits.cfg</code>	<code>Stalker2\Mods\MyCreditsMod\Content\GameLite\GameData\Credits.cfg_patch_MyCreditsMod</code> <code>Stalker2\Mods\MyOtherCreditsMod\Content\GameLite\GameData\Credits.cfg_patch_MyOtherCreditsMod</code>	<code>Stalker2\Content\GameLite\GameData\Credits.cfg_patch_MyCreditsMod</code> <code>Stalker2\Content\GameLite\GameData\Credits.cfg_patch_MyOtherCreditsMod</code>
<code>Stalker2\Content\GameLite\GameData\CoreVariables.cfg</code>	<code>Stalker2\Mods\GlobalOverhaul\Content\GameLite\GameData\CoreVariables.cfg_patch_GO</code>	<code>Stalker2\Content\GameLite\GameData\CoreVariables.cfg_patch_GO</code>
<code>Stalker2\Content\GameLite\GameData\AIGlobals.cfg</code>	<code>Stalker2\Mods\AIOverhaul\Content\GameLite\GameData\AIGlobals.cfg_patch_AIO</code> <code>Stalker2\Mods\AITweaks\Content\GameLite\GameData\AIGlobals.cfg_patch_AITweaks</code> <code>Stalker2\Mods\HardcoreAI\Content\GameLite\GameData\AIGlobals.cfg_patch_HardcoreAI</code>	<code>Stalker2\Content\GameLite\GameData\AIGlobals.cfg_patch_AIO</code> <code>Stalker2\Content\GameLite\GameData\AIGlobals.cfg_patch_AITweaks</code> <code>Stalker2\Content\GameLite\GameData\AIGlobals.cfg_patch_HardcoreAI</code>

NOTE

While the file extensions are unique, the file names, as well as the folder structure, correspond to the in-game ones.

Using a plain *.cfg file

In order to apply modifications to any prototype .cfg file, you need to create a .cfg file with a unique file name in the corresponding prototype subfolder.

When loading such files, Zone Kit or the game will look for all .cfg files inside the corresponding **prototype** folder and apply them as patches.

Here's a table with references and examples:

Vanilla .cfg file to modify	Source .cfg files in Zone Kit	Packaged .cfg files in mod files
<code>Stalker2\Content\GameLite\GameData\WeaponData\CharacterWeaponSettingsPrototypes.cfg</code>	<code>Stalker2\Mods\BuffedAK\Content\GameLite\GameData\WeaponData\CharacterWeaponSettingsPrototypes\BuffedAK.cfg</code>	<code>Stalker2\Content\GameLite\GameData\WeaponData\CharacterWeaponSettingsPrototypes\BuffedAK.cfg</code>
<code>Stalker2\Content\GameLite\GameData\AnomalyPrototypes.cfg</code>	<code>Stalker2\Mods\NerfedAnomaly\Content\GameLite\GameData\AnomalyPrototypes\MyAnomalyNerf.cfg</code>	<code>Stalker2\Content\GameLite\GameData\AnomalyPrototypes\MyAnomalyNerf.cfg</code>
<code>Stalker2\Content\GameLite\GameData\SpawnActorPrototypes.cfg</code>	<code>Stalker2\Mods\MyPlaceholderMod\Content\GameLite\GameData\SpawnActorPrototypes\WorldMap_WP\9A41EE1649839E16066221BAE266390E_MyPlaceholderMod.cfg</code>	<code>Stalker2\Content\GameLite\GameData\SpawnActorPrototypes\WorldMap_WP\9A41EE1649839E16066221BAE266390E_MyPlaceholderMod.cfg</code>

Config file cache

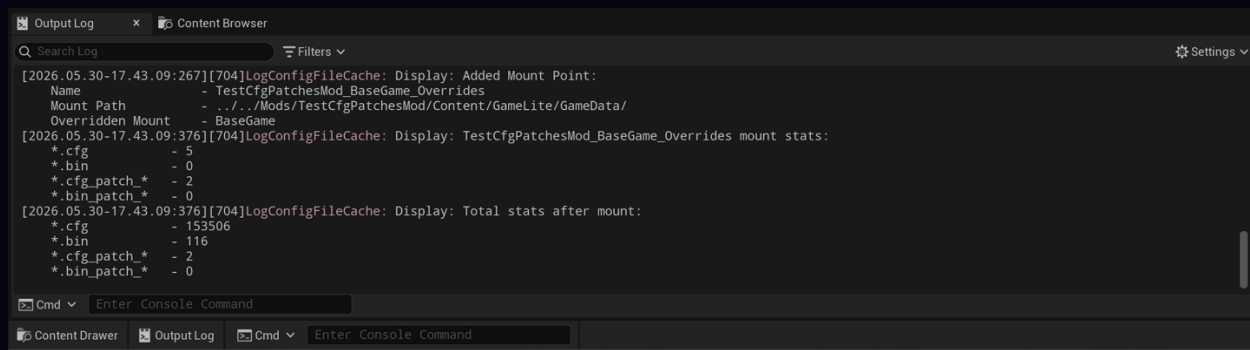
The ConfigFileCache is an entity that stores the following data:

- .cfg, .cfg_patch_*, .cfg.bin, .cfg.bin_patch_* file paths for fast access at runtime
- Mount points, which can contain the aforementioned files
- Metadata

The cache gets updated at certain events, e.g.:

- At the startup of Zone Kit or the game
- (De)selecting a mod in Zone Kit
- When some content pack – BaseGame, DLC, a mod – gets (de)activated
- Etc.

On each update, the cache prints relevant information to the Output window and/or the .log file using the LogConfigFileCache category.



In the context of creating and testing mods in Zone Kit, the cache is a useful debugging tool. If you require even more detailed information than the LogConfigFileCache category currently provides, you can also enable ConfigFileCache dumping by using the -DumpConfigFileCache command line argument. When using the argument on each update, the cache will dump all its contents into a ConfigFileCacheDump_ModEditor.txt file, located in the logs directory.

While testing your mods in Zone Kit, the cache ensures that any .cfg or .cfg_patch_* files from the currently selected mod content's directory are seamlessly applied as patches, enabling you to test your mods in PIE sessions.