



HOW TO PATCH CONTENTS OF A .CFG FILE

Prerequisites

- [How to apply modifications to vanilla .cfg files](#)

Introduction

Modifying the contents of a vanilla .cfg file is called .cfg patching. Patching allows doing the following:

1. Add a node.
2. Completely replace a root node.
3. Patch a child node.
4. Remove a node.

Let's inspect these operations in detail.

NOTE

While we'll be using the `.cfg_patch_*` mod application method in our examples, the .cfg patching mechanisms work seamlessly with any application method.

Adding a node

The process of adding a new node via a patch file is pretty straightforward.

Here's an example:

Test.cfg	Test.cfg_patch_mod	Parsing result
<pre>RootA: struct.begin Value1 = 1 Value2 = 2 struct.end</pre>	<pre>RootB: struct.begin ValueA = 5 ValueB = 6 struct.end</pre>	<pre>RootA: struct.begin Value1 = 1 Value2 = 2 struct.end RootB: struct.begin ValueA = 5 ValueB = 6 struct.end</pre>

Completely replacing a root node

By default, any node inside the .cfg_patch_* file will replace the entire node in the base .cfg file.

Here's an example:

Test.cfg	Test.cfg_patch_mod	Parsing result
<pre>RootA: struct.begin Value1 = 1 Value2 = 2 struct.end RootB: struct.begin ValueA = 3 ValueB = 4 ValueC = 5 struct.end</pre>	<pre>RootB: struct.begin ValueABC = ABC ValueD = D struct.end</pre>	<pre>RootA: struct.begin Value1 = 1 Value2 = 2 struct.end RootB: struct.begin ValueABC = ABC ValueD = D struct.end</pre>

NOTE

The RootB node and its child nodes have been completely replaced with the RootB node from the Test.cfg_patch_mod file. Nothing from the vanilla RootB node has been preserved.

Patching a node or its children – the {bpatch} keyword

The {bpatch} keyword can be used to modify, add, or replace child nodes of a given root node.

Here's an example:

Test.cfg	Test.cfg_patch_mod	Parsing result
<pre> RootA: struct.begin Value1 = 1 Value2 = 2 struct.end RootB: struct.begin ValueA = 3 ValueB = 4 ChildC: struct.begin ValueCCC = SomeValue struct.end struct.end </pre>	<pre> RootB: struct.begin {bpatch} ValueA = NEW_A ChildC: struct.begin {bpatch} ValueDDD = SomeAddedValue struct.end struct.end </pre>	<pre> RootA: struct.begin Value1 = 1 Value2 = 2 struct.end RootB: struct.begin ValueA = NEW_A ValueB = 4 ChildC: struct.begin ValueCCC = SomeValue ValueDDD = SomeAddedValue struct.end struct.end </pre>

NOTE

- The RootB → ValueA node has been replaced with the value from the Test.cfg_patch_mod file. The vanilla RootB → ValueB node has been preserved.
- The RootB → ChildC → ValueDDD node from the Test.cfg_patch_mod file has been added to the vanilla RootB → ChildC node. The vanilla RootB → ChildC → ValueCCC node has been preserved.

Removing a node – the removenode keyword

The removenode keyword can be used to remove the entire node from a given node. This keyword only works when also using the {bpatch} keyword.

Here's an example:

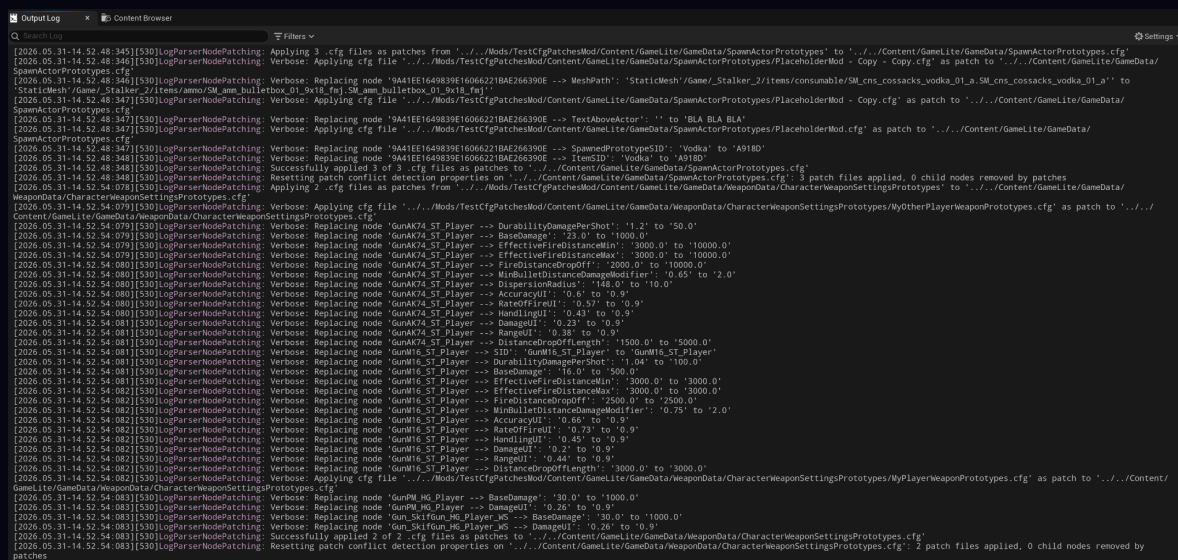
Test.cfg	Test.cfg_patch_mod	Parsing result
<pre> RootA: struct.begin [0] : struct.begin Name=SlotA Value=Knife struct.end [1] : struct.begin Name=SlotB Value=Pistol struct.end </pre>	<pre> RootA: struct.begin {bpatch} [2] : removenode struct.end </pre>	<pre> RootA: struct.begin [0] : struct.begin Name=SlotA Value=Knife struct.end [1] : struct.begin Name=SlotB Value=Pistol struct.end </pre>

```
[2] : struct.begin
      Name=SlotC
      Value=Bolt
struct.end
struct.end
```

Debugging .cfg patches

To improve the efficiency when testing your .cfg mods, you can use the LogParserNodePatching dedicated log category. By default, the patch application system logs only overview information: what .cfg patches are applied to what files; whether they have been applied successfully, etc. But by increasing the log verbosity to Verbose or higher, the system will print messages per each patched node.

Here's an output example:



```
Output Log | Content Browser | Filters | Settings
[2026.05.31.14.52.48.345][530]LogParserNodePatching: Applying 3 .cfg files as patches from './../Mods/TestCfgPatchesMod/Content/GameLite/GameData/SpawnActorPrototypes' to './../Content/GameLite/GameData/SpawnActorPrototypes.cfg'
[2026.05.31.14.52.48.346][530]LogParserNodePatching: Verbose: Applying cfg file './../Mods/TestCfgPatchesMod/Content/GameLite/GameData/SpawnActorPrototypes/PlaceholderMod - Copy - Copy.cfg' as patch to './../Content/GameLite/GameData/SpawnActorPrototypes.cfg'
[2026.05.31.14.52.48.346][530]LogParserNodePatching: Verbose: Replacing node '9A41EE1649839E16066221BAE266390E --> MeshPath: 'StaticMesh'/Game/_Stalker_2/items/consumable/SM_cns_cossacks_vodka_01_a.SM_cns_cossacks_vodka_01_a' to 'StaticMesh'/Game/_Stalker_2/items/amor/sm_bulletbox_01_bulletbox_01_bulletbox_01_bulletbox_01'
[2026.05.31.14.52.48.347][530]LogParserNodePatching: Verbose: Applying cfg file './../Mods/TestCfgPatchesMod/Content/GameLite/GameData/SpawnActorPrototypes/PlaceholderMod - Copy.cfg' as patch to './../Content/GameLite/GameData/SpawnActorPrototypes.cfg'
[2026.05.31.14.52.48.347][530]LogParserNodePatching: Verbose: Replacing node '9A41EE1649839E16066221BAE266390E --> TextAboveActor: '' to 'BLA BLA BLA'
[2026.05.31.14.52.48.347][530]LogParserNodePatching: Verbose: Applying cfg file './../Mods/TestCfgPatchesMod/Content/GameLite/GameData/SpawnActorPrototypes/PlaceholderMod.cfg' as patch to './../Content/GameLite/GameData/SpawnActorPrototypes.cfg'
[2026.05.31.14.52.48.347][530]LogParserNodePatching: Verbose: Replacing node '9A41EE1649839E16066221BAE266390E --> SpawnedPrototypesID: 'Vodka' to 'A918D'
[2026.05.31.14.52.48.348][530]LogParserNodePatching: Verbose: Replacing node '9A41EE1649839E16066221BAE266390E --> ItemID: 'Vodka' to 'A918D'
[2026.05.31.14.52.48.348][530]LogParserNodePatching: Successfully applied 3 of 3 .cfg files as patches to './../Content/GameLite/GameData/SpawnActorPrototypes.cfg'
[2026.05.31.14.52.48.348][530]LogParserNodePatching: Resetting patch conflict detection properties on './../Content/GameLite/GameData/SpawnActorPrototypes.cfg': 3 patch files applied, 0 child nodes removed by patches
[2026.05.31.14.52.54.078][530]LogParserNodePatching: Applying 2 .cfg files as patches from './../Mods/TestCfgPatchesMod/Content/GameLite/GameData/WeaponData/CharacterWeaponSettingsPrototypes' to './../Content/GameLite/GameData/WeaponData/CharacterWeaponSettingsPrototypes.cfg'
[2026.05.31.14.52.54.079][530]LogParserNodePatching: Verbose: Applying cfg file './../Mods/TestCfgPatchesMod/Content/GameLite/GameData/WeaponData/CharacterWeaponSettingsPrototypes/MyOtherPlayerWeaponPrototypes.cfg' as patch to './../Content/GameLite/GameData/WeaponData/CharacterWeaponSettingsPrototypes.cfg'
[2026.05.31.14.52.54.079][530]LogParserNodePatching: Verbose: Replacing node 'GunAK74_ST_Player --> DurabilityDamagePerShot: '1.2' to '50.0'
[2026.05.31.14.52.54.079][530]LogParserNodePatching: Verbose: Replacing node 'GunAK74_ST_Player --> BaseDamage: '23.0' to '1000.0'
[2026.05.31.14.52.54.079][530]LogParserNodePatching: Verbose: Replacing node 'GunAK74_ST_Player --> EffectiveFireDistanceMin: '3000.0' to '10000.0'
[2026.05.31.14.52.54.079][530]LogParserNodePatching: Verbose: Replacing node 'GunAK74_ST_Player --> EffectiveFireDistanceMax: '3000.0' to '10000.0'
[2026.05.31.14.52.54.080][530]LogParserNodePatching: Verbose: Replacing node 'GunAK74_ST_Player --> FireDistanceDropOff: '2000.0' to '10000.0'
[2026.05.31.14.52.54.080][530]LogParserNodePatching: Verbose: Replacing node 'GunAK74_ST_Player --> MinBulletDistanceDamageModifier: '0.65' to '2.0'
[2026.05.31.14.52.54.080][530]LogParserNodePatching: Verbose: Replacing node 'GunAK74_ST_Player --> DispersionRadius: '148.0' to '10.0'
[2026.05.31.14.52.54.080][530]LogParserNodePatching: Verbose: Replacing node 'GunAK74_ST_Player --> AccuracyUI: '0.6' to '0.9'
[2026.05.31.14.52.54.080][530]LogParserNodePatching: Verbose: Replacing node 'GunAK74_ST_Player --> RateOfFireUI: '0.57' to '0.9'
[2026.05.31.14.52.54.080][530]LogParserNodePatching: Verbose: Replacing node 'GunAK74_ST_Player --> HandlingUI: '0.45' to '0.9'
[2026.05.31.14.52.54.081][530]LogParserNodePatching: Verbose: Replacing node 'GunAK74_ST_Player --> DamageUI: '0.23' to '0.9'
[2026.05.31.14.52.54.081][530]LogParserNodePatching: Verbose: Replacing node 'GunAK74_ST_Player --> RangeUI: '0.38' to '0.9'
[2026.05.31.14.52.54.081][530]LogParserNodePatching: Verbose: Replacing node 'GunAK74_ST_Player --> DistanceDropOffLength: '1500.0' to '5000.0'
[2026.05.31.14.52.54.081][530]LogParserNodePatching: Verbose: Replacing node 'GunW16_ST_Player --> SID: 'GunW16_ST_Player' to 'GunW16_ST_Player'
[2026.05.31.14.52.54.081][530]LogParserNodePatching: Verbose: Replacing node 'GunW16_ST_Player --> DurabilityDamagePerShot: '1.04' to '100.0'
[2026.05.31.14.52.54.081][530]LogParserNodePatching: Verbose: Replacing node 'GunW16_ST_Player --> BaseDamage: '16.0' to '500.0'
[2026.05.31.14.52.54.081][530]LogParserNodePatching: Verbose: Replacing node 'GunW16_ST_Player --> EffectiveFireDistanceMin: '3000.0' to '3000.0'
[2026.05.31.14.52.54.081][530]LogParserNodePatching: Verbose: Replacing node 'GunW16_ST_Player --> EffectiveFireDistanceMax: '3000.0' to '3000.0'
[2026.05.31.14.52.54.082][530]LogParserNodePatching: Verbose: Replacing node 'GunW16_ST_Player --> FireDistanceDropOff: '2500.0' to '2500.0'
[2026.05.31.14.52.54.082][530]LogParserNodePatching: Verbose: Replacing node 'GunW16_ST_Player --> MinBulletDistanceDamageModifier: '0.75' to '2.0'
[2026.05.31.14.52.54.082][530]LogParserNodePatching: Verbose: Replacing node 'GunW16_ST_Player --> AccuracyUI: '0.66' to '0.9'
[2026.05.31.14.52.54.082][530]LogParserNodePatching: Verbose: Replacing node 'GunW16_ST_Player --> RateOfFireUI: '0.73' to '0.9'
[2026.05.31.14.52.54.082][530]LogParserNodePatching: Verbose: Replacing node 'GunW16_ST_Player --> HandlingUI: '0.45' to '0.9'
[2026.05.31.14.52.54.082][530]LogParserNodePatching: Verbose: Replacing node 'GunW16_ST_Player --> DamageUI: '0.2' to '0.9'
[2026.05.31.14.52.54.082][530]LogParserNodePatching: Verbose: Replacing node 'GunW16_ST_Player --> RangeUI: '0.44' to '0.9'
[2026.05.31.14.52.54.082][530]LogParserNodePatching: Verbose: Replacing node 'GunW16_ST_Player --> DistanceDropOffLength: '3000.0' to '3000.0'
[2026.05.31.14.52.54.082][530]LogParserNodePatching: Verbose: Applying cfg file './../Mods/TestCfgPatchesMod/Content/GameLite/GameData/WeaponData/CharacterWeaponSettingsPrototypes/MyPlayerWeaponPrototypes.cfg' as patch to './../Content/GameLite/GameData/WeaponData/CharacterWeaponSettingsPrototypes.cfg'
[2026.05.31.14.52.54.083][530]LogParserNodePatching: Verbose: Replacing node 'GunPM_HG_Player --> BaseDamage: '30.0' to '1000.0'
[2026.05.31.14.52.54.083][530]LogParserNodePatching: Verbose: Replacing node 'GunPM_HG_Player --> DamageUI: '0.26' to '0.9'
[2026.05.31.14.52.54.083][530]LogParserNodePatching: Verbose: Replacing node 'Gun_Skifun_HG_Player_KS --> BaseDamage: '30.0' to '1000.0'
[2026.05.31.14.52.54.083][530]LogParserNodePatching: Verbose: Replacing node 'Gun_Skifun_HG_Player_KS --> DamageUI: '0.26' to '0.9'
[2026.05.31.14.52.54.083][530]LogParserNodePatching: Successfully applied 2 of 2 .cfg files as patches to './../Content/GameLite/GameData/WeaponData/CharacterWeaponSettingsPrototypes.cfg'
[2026.05.31.14.52.54.083][530]LogParserNodePatching: Resetting patch conflict detection properties on './../Content/GameLite/GameData/WeaponData/CharacterWeaponSettingsPrototypes.cfg': 2 patch files applied, 0 child nodes removed by patches
```

NOTE

You can also use the -ProtoLoadingMode=2 command line argument to make sure that all .cfg files are loaded and patches are applied synchronously, which will improve the logs' readability.

Patch conflict detection

Because many different mods can patch the same .cfg files, patch conflict detection is active when patches are applied to a node and its children. While in Zone Kit, the aforementioned LogParserNodePatching category will print warning and error messages when it detects conflicts. Here are the cases covered by the system:

1. Readdition of a node by patch A, which has previously been removed by patch B.
2. Removal of a node by patch A, which (or children of which) has previously been modified by patch B.
3. Replacement of a node by patch A, which (or children of which) has previously been modified by patch B.