

# Community Patch 1.72 Documentation

## Introduction

Welcome into a documentation about unofficial patch 1.72. This document expects you are aware of previous version 1.71 and its fixes and features. If you don't, you should read it about it on this [link](#).

First of all. Community Patch (CP) 1.72 is not just a patch, it is a **platform**. It's a whole system you can use to build modules and custom content for or around and do things that were not even thought before. It should be used with conjunction with NWNX+nwnx\_patch (for server) and NWCX+nwcx\_patch (for game client). But it will work without it too of course. Compatibility is still important, just as it was in previous patches, 1.72 still provides a 100% compatibility in terms that it won't break anything you might already built (*and if it does it is a bug*).

If you are a player, the greatest benefit of this patch lies in nwcx\_patch plugin fixing and improving major issues that are hardcoded in game engine (such as hellball crash). If you are a builder, then the question isn't why you should be using this patch, but why shouldn't you.

Let's get into the changes now...

## Fixes into previous Patch versions

Nothing is perfect and over time some issues with 1.70/1.71 has been found.

### Fixes:

- Fixed a bug in all cure wounds spells which didn't maximized the result on normal and easy difficulty.
- Fixed a bug that blocked OnUserDefined event to be triggered, when spell was cast at non-commandable creature.
- Added missing code for Wounding itemproperty handling (somehow didn't make it into 1.71 due to the overlook).
- Fixed name and description of the Eye of Gruumsh Blinding spittle ability (the first one, second had correct name).
- Fixed Shou Disciple's Dodge +2 and Dodge +3 feat name and description in character sheet/toolset item bonus feat selection.
- Duration of the paralyse from Planar binding spells was calculated incorrectly when spell was cast extended.

- Removed compiled version of the ttu01\_g15\_01 model as it could crash/bug with shiny water enabled under specific graphic cards due to the bug in external model compiler.
- Due to the overlook the promised fix for losing variables when crafting armor wasn't working.
- Fixed heads 50-63 textures (these heads aren't accessible normally that's why this issue wasn't found before).
- Fixed several highres head textures that were lowres in fact.
- Fixed levelling up Pale Master with less than 17/18 sorc/wiz levels taking epic spells (70\_mod\_levelup)
- New fix for Gedlee's Electric Loop 1 damage issue, CP fixed the issue with stunning the target when 1 damage was rolled and improved evasion, turns out that the CP solution created new opposite issue where the stun roll wasn't attempted if the 1 damage was rolled and target failed in the first save.
- Barbarian rage and Mighty rage abilities: fixed self stacking of the ac penalty + any possible custom effects applied via epic rages that might stack, also fixed incorrectly calculated AC penalty for Eye of the Gruumsh.
- Cure light wounds: due to the typo actually healed 3d8 hit points more.
- Dirge: fixed bug where the spell had no effect on entering enemies and only heartbeat worked
- Sunburst: added missing kill feedback in case spell killed vampire.
- Turn undead: re-enabled this ability to affect neutral targets in order to support scene in vanilla campaign where the cleric turning undead is not hostile towards the skeleton
- Undeath to death: added missing kill feedback.
- Fixed bug in spellsIsLightVulnerable function which always returned TRUE; this caused Sunbeam to always do 6d6 damage instead of 3d6 and allowed Sunburst to kill any undead not just vampires.
- Greater Sanctuary, Melf's Acid Arrow, Tenser's Transformation, Bigby's Clenched Fist and Bigby's Crushing Fist: fixed some of the declarations of caster level to account for spell overrides.
- Fixed bug where creature in darkness couldn't use gaze abilities even if she had ultravision/true seeing.
- Fixed bug where creature in darkness was considered immune to the gaze effects, even if she had ultravision/true seeing.
- Fixed bug in AI that prevented creatures to heal their allies.
- Fixed nonfunctional feature to cast dispel at AOE location - this should prevent issues when the dispel cast under self didn't dispelled the AOE spell.
- Fixed several issues in optional AI for polymorphed creatures where such creatures got stuck when trying to use certain abilities and only performed melee attacks.
- Fixed AI feature with heal/inflct spell substituting that could stuck creature due to the engine bug with TalentSpell.
- Ballista and "Ballista, arbalet" placeables set not to be static in 2da (could cause crashing and other issues on some graphic cards).

- Fixed an issue with multi-summoning feature, where summoning a creature close to the other player with his own summons count the other player as a master and could unsummon his own summon.
- Fixed a bug where under specific circumstances, an aoe spell like delayed fireball ignored target immunity to 9/8/7 level spells if this spell was acquired by some custom content like a polymorph or something.
- Fixed bug that prevented Mestil's acid arrow spell to apply the subsequent damage when the spell was cast by PM or any custom prestige class granting caster level on target with spell resistance.
- Re-enabled spellscripts to be forwards compatible again (when spellscript from CP was used by player without CP installed, spell were missing spellhook related features such as crafting support or firing user defined spellhook script)
- Fixed Sight of Gruumsh feat (Eye of Gruumsh) not providing the +2 bonus to saving throws.
- Removed incorrectly appearing Ability Bonus: in cloak descriptions.
- Fixed lowres version of female gnome head 1 texture not having colorable eyes and lips.
- Fixed custom spell resist calculation (used when dealing with AOE's) not counting spell penetration feats.
- Fixed targeting bug that allowed to cast any polymorph spell on any target no matter its 2da settings.

#### Revisited features:

- Removed custom main menu GUI and replaced with just 1.72 sticker under HotU expansion logo (as was done for NWN:EE version).
- Removed the feature to change all target-only-enemies spells into target-everyone happening on highest difficulty settings. *(Those who wish to have this can now code it themselves using new spell engine feature to dynamically change spell target type)*
- Removed ground targeting from the Blinding Spittle special ability (Eye of Gruumsh) as it caused misfires.
- Wounding property will be now functional even when the player is dying.
- Battletide and Dirge spells AI behavior changed from aura-like spells and moved to protection buffs as it was a bit weird to start with these spells and then use talents like fear aura + high AI was then able to cast battletide instantly which wasn't intended.
- Psionic inertial barrier: rescripted to use CP spell engine to fix the targeting issues that any polymorph spell can be cast on any object *(which in vanilla isn't problem as the script is written to only apply effects on OBJECT\_SELF, while 1.71 changed all these scripts to use GetSpellTargetObject() instead in order to provide custom content compatibility)*.
- Tenser's Transformation: removed the int/cha/wis undocumented bonuses that were added in 1.70 to fix losing spell slots *(which didn't even work anyway)*.
- Removed undocumented overriding crates models from NWNEhanced that were added in 1.70 *(it doesn't look like there is any fix in them, only reskin)*.

- Removed the "rotation key fix for base character models" from OTR which unfortunately caused loss of animations on cloaks for dwarfs.
- Changed code for spell immunity workaround that previously considered having a spell effect of polymorph: pixie spell for an immunity to enchantment spell school. Now the script will look into skin in order to provide custom content compatibility in case the pixie polymorph had modified skin properties or even when the pixie shape was replaced by another shape.
- Edit comments in the various scripts and includes has been improved so it's easier to see what was changed in 1.72 over 1.71.
- Delayed Blast Fireball and Glyph of Warding spells were given an exception from the new AOE behavior, thus they will affect spell mantle and won't be affected by the AOE related module switches.
- Battletide and Dirge AI behavior changed from aura-like spells and moved to protection buffs as it was a bit weird to start with these spells and then use talents like fear aura + high AI was then able to cast battletide instantly which wasn't intended.
- Continuous flame: items will be marked as stolen only in multiplayer now.
- Breach spells: new feedback will be printed also for caster if it is player.
- Ioun stones will no longer be dispellable by the standard means. This was implemented in early 1.70 and reverted before release due to the critique on this change. I know there are modules where they prefer ioun stones to be dispellable, but on the same hand there are modules which prefer not to. Since DnD rules are clear, I decided they will be not dispellable by default in 1.72. Either way, it's very easy to change this in scripts especially when 1.72 introduced single multiscript for all ioun stones 70\_s3\_iounstone.
- [NWNX\_Patch] Curse feature to ignore ability decrease immunity is no longer automatically enabled and is now dependant on a module switch.
- [NWNX\_Patch] Circle kick fix will work only for players now (often builder wants creature to switch targets and circle kick is easiest way to do that).
- disease.2da - unused column Type given back to make comparing 2das easier.
- The odd damage workaround in flame strike and firestorm spell scripts recoded to simpler solution.
- When nwn(c)x\_patch is enabled, stacked poison will show poison icon and green hp bar.
- Gruumsh blindsight 10 feat will overwrite the 5 feet variant in the character sheet feat list.
- SAVING\_THROW\_TYPE\_PARALYSE replaced in all scripts with direct value 20 to provide compatibility with vanilla nwscript.nss
- Recoded the double-respawn protection: instead of making sure death GUI won't appear twice (*which seemed not to work perfectly anyway*), the respawn will be canceled if player is not dead and not petrified either. (Note: this will only work when using nwnx or when builder merges content of the '70\_mod\_def\_resp' script)
- Changed the default level-up package for Eye of the Gruumsh and Shou Disciple classes to better matching one.
- Removed very high difficulty "balance change" in diffsettings.2da.

## HAK Version

- Removed unnecessary DOA base items files from patch172.hak.
- Added hidden/unused standard items: pouch (1x1 container), small box (2x1 container bag), repeating crossbow (exotic crossbow without standard crossbow drawbacks, requires nwn(c)x\_patch to work), old icon style cloaks (same name in the list, it's the lower one).
- Added one more hidden/unused creature: orc chieftain C *(wasn't added before because the model is missing animations - that turned out as an easy fix however, so there he is)*.
- Added quickened banshee visual effect into hak version - builders are recommended to add this vfx into their haks, as players without CP installed might think that banshee is bugged and is killing faster than it should, which isn't true, banshee kills quicker now as should and those with CP installed will see both death effect and wailing visual perfectly synchronized.
- New creature blueprints in hak recategorized, they now appear in brand [new category called Community Patch](#) for easier access.

## NWNX\_Patch and NWN CX\_Patch

Perhaps most exciting and useful fixes and features in 1.72 are done with nwnx\_patch and nwn cx\_patch plugin. To acquire all these fixes and features, you need to host your server via NWNX2.exe or run your game client via NWN CX Loader.exe (or patched nwmain.exe). If you are not familiar with this read more info about [NWNX](#) and [NWN CX](#). Once you get these packages running, you can use nwnx\_patch and/or nwn cx\_patch.

Since the amount of fixes and features is really overwhelming, it deserved a special readme which can be read [here](#). The rest of this readme will not mention anything that requires NWNX to avoid duplicating same informations.

*This content is unfortunately not available for NWN:EE version of the game.*

## Client Enhancements

There are notable client-side enhancements, such as graphical improvements, model corrections, tile corrections, tintable heads and description corrections.

### General enhancements:

- Hundred of tile related fixes, including [this](#).
- Added Rubies badger overhaul from [CCC June 2012](#).
- Added a small selection of colored GUI and feat icons from The Amethyst Dragon.
- New [metamagic icons](#) in radial menu/spellbook.
- Fixed epic spell scroll icons *(the fix is only partial, it's not an actual scroll but spell icon instead, but it's still better than "TEMP")*.

- Added ground model for shields.
- Ugly DOA cloak replaced with standard cloak (*this actually appeared only when placing cloak item from palette to area, but now it appears 100% matching with what you see at ground once you place it*).
- Improved [barkskin texture](#) changed to be closer to the vanilla in term of colours.
- Added desert male head reskin from [Yavel](#)
- Added [colorized elemental shapes](#) icons.
- Added [IMProvments](#) from \_six, package improving textures of imps and mephits, however only textures has been added not the models which means the VFX effects are not included.
- Colored emote icons from The Amethyst Dragon replaced by [Melliuc emoticons](#) from Melliuc
- Fixed human male right shoulder 14 and 21 models. (*The Barbarian*)

### PC Widget Tool improvements:

- Module Switches globalized. The switches setup will now be transferred into every single player module the player will play. Note, the module settings has always priority - nevertheless, player will be able to override even module settings if he wishes to.
- Added a new switches into PC Widget Tool:
  - switch to merge all items into all polymorph shapes
  - switch to merge casting abilities (cha,int,wis) into all polymorph shapes (to prevent losing spell slots from polymorphing)
  - switch to enable managing OC henchmen's inventory
  - switch to enable "weapon boost" spells on ammunition or throwing weapons
  - switch to enable hardcore evasion rules (light or no armor etc.)
  - and several switches depending on NWNX which are described in [NWNX\\_Patch readme](#)

### Notable description corrections:

- Battletide: area of effect medium not large.
- Darkfire: removed the mention about non-magical weapon as the spell doesn't take this into consideration.
- Several small corrections and improvements in Gruumsh/Shou class or feat descriptions.

And hundreds of individual tile fixes, full changelog at the end of this documentation.

### Fixes into official campaigns:

Community patch 1.72 also improves and corrects some known issues in campaigns.

### Community Patch exclusive fixes:

- Fixed SoU campaign bug in chapter 1 that prevented the conversation option to charm Hanniah work properly.
- Fixed HotU campaign bug in chapter 3 that didn't kill henchmen after surrendering their soul.
- Following items from SoU and HotU will now work in every module player will play including vanilla NWN campaign: Djinni Bottle, Ragged Cloak of the Worg, SoU starting class items.
- Player given items are no longer lost when OC henchman level up. *(But still be lost when moving between chapters! Also dependant on NWN(C)X plugin.)*
- Nasher items will grant additional benefits of the “full set” in every module player will play *(dependant on NWN(C)X plugin).*

#### Existing fixes from other authors:

- [Aribeth HotU Romance Fix](#)
- [Valen HotU Romance Fix](#)
- [HotU Smith Fix](#)

#### Various Creature fixes

- All creatures with F type of model, *except bodak and orc chief A*, allowed to select wing and tail *(comes with modified creature models)*. [Toolset showcase](#).
- Removed erroneous wing and tail from bodak and orc chief blueprints.
- Fixed kobold footpad and kobold thug's reflection settings *(Gruftlord)*.
- Corrected OnDamaged event script for many summoned creatures.
- Added flank immunity to elemental and ooze creatures *(using the bonus feat: defensive awareness 2, this also applies for druid in elemental shape forms)*.
- Fixed pauldron model rotation on the bone golem creature *(Pstemarie)*.
- Formians and hound archon added into list of creatures immune to petrification.
- Pseudodragon added into list of flying creatures.
- Bodak added into list of light vulnerable creatures.
- Gelatinous cube's paralysis will not longer be dispellable.
- Dexterity of dire panther polymorph shape raised by 4 points to make sense and to match its normal version *(previously dire panther had -2 con, when the values of normal panther has been swapped however, he had -4 dex)*.
- Rebalanced animal shape creature weapons. In vanilla NWN every shape has the same weapons and damage so both bear and badger has a 1d4 claws. This have been changed according to the standard creature counterparts. Table below shows the changes, first 3 columns are standard values, later columns are new values.

Shape	1.weapon	2.weapon	3.weapon	1.weapon	2.weapon	3.weapon
bear	claw 1d4	claw 1d4	bite 1d4	claw 1d8	claw 1d8	bite 2d8

panther	claw 1d4	claw 1d4	bite 1d4	claw 1d2	claw 1d2	bite 1d4
wolf	claw 1d4	claw 1d4	bite 1d4	nothing	nothing	bite 1d6
boar	claw 1d4	claw 1d4	bite 1d4	nothing	nothing	bite 1d8
badger	claw 1d4	claw 1d4	bite 1d4	claw 1d2	claw 1d2	bite 1d4
dire bear	claw 1d8	claw 1d8	bite 1d8	claw 2d4	claw 2d4	bite 2d8
dire panther	claw 1d8	claw 1d8	bite 1d8	claw 1d4	claw 1d4	bite 1d6
dire wolf	claw 1d8	claw 1d8	bite 1d8	nothing	nothing	bite 1d8
dire boar	claw 1d8	claw 1d8	bite 1d8	nothing	nothing	bite 1d8
dire badger	claw 1d8	claw 1d8	bite 1d8	claw 1d4	claw 1d4	bite 1d6

## Spell, Feat and Spell-abilities fixes and improvements

### Touch attack corrections

Spells requiring melee touch attack (*inflict spells, slay living, vampire touch, ghoul touch, harm, heal and cure spell when cast on undead*) will now skip the touch attack if the caster and target match. This is in order to support potions with such spells as it makes no sense to make touch attack after drinking potions (*while nobody would want to use harming potions on himself, CP allows to create a cursed potions which will trigger different spell such as inflict wounds instead of healing*).

### Line of sight correction

All spells with area of effect now checks "line of sight", that means they won't affect anyone behind doors or wall.

### AOE spells caster validity check

The AOE caster validity check improved, so it will no longer automatically destroy AOE created outside of the spellscript.

Also AOE caster validity check was added to these AOE spell that weren't using it: entangle, grease, web.

### Metamagic checks

Metamagic checks in all spells has been replaced with the binary AND (&) to support custom content such as metamagic stacking.



### **Fixed targeting a dying players**

Spells will now work on dying players, this will solve the weird behavior where player drops below 1 hp by some aoe spell like creeping doom and then he lay in the aoe not taking any damage at all.

### **Major spellscript rewrite**

Spells that normally doesn't allow saving throw rescripted to allow it via spellhook (ice storm, heal, harm, vampiric touch, acid arrow and few more). Spells that normally doesn't check spell resistance rescripted to allow add this check via spellhook (earthquake, crumble, greater thunderclap, creeping doom, dispels, breaches). This changes nothing by default, it only enables builder to add saving throw or spell resistance into these spells using new spell engine overrides.

### **NWNX support**

When NWNX + nwnx\_patch/nwncx\_patch is in use, the caster level override/modifier will now be able to work even against monks with high SR.

### **Dispel spells**

- will no longer affect dead targets (*basically only a visual change since it had no effect anyway*)
- dispel effect externalized into '70\_mod\_dispel' script, if builder would want to change the behavior of the dispel effect it can be done here without modifying spellscripts

### **Mind blank spells (+clarity)**

- added supernatural effect check as is in restoration spells - this allows to make a mind-affecting effect which won't be removed with these spell

### **Petrifying spells and special abilities**

- all spells and special abilities will execute custom script '70\_mod\_petrified' on a petrified targets, this script serves primarily for PW builders as a place to code persistency without need to touch the actual spell scripts

### **Restore spells**

- won't remove horse related effects anymore

### **Resurrection spells**

- both Raise dead and Resurrection will execute custom script '70\_mod\_resurrect', this script serves primarily for PW builders as a place to code persistency without need to touch the actual spell scripts

### **Weapon enhancement spells**

- spells are no longer dispellable (*this is just to prevent confusion - dispel could only remove hidden visual effect anyway, the actual benefits on weapon were always undispellable*)

### **Battletide**

- fixed a rare situation where the penalty from the aura wasn't properly removed after leaving the aura
- added special workaround to handle how AI casts this spell. (Default AI casts this spell on enemy and then gives benefits to caster, community patch changed every spell to give benefits to target to fix potions and enable 2da customization, but if someone didn't merged spells.2da properly AI could cast this on player and give aura to player.)

### **Bigby's Forceful Hand**

- stun vfx will no longer appear on target who has not been affected by this spell at all

### **Bless**

- visual effect duration in case of spell cast on bolts united to match the itemproperty duration

### **Color Spray**

- area of effect size prolonged to 11.0 to match the distance of the cone visual effect
- sightless creatures are now immune

### **Darkness**

- aoe effects are now undispellable (extraordinary)

### **Delayed Blast Fireball**

- when cast directly on single target make explosion immediately without making AOE object, this modification is especially useful for PWs as AOE's have negative impact on monster AI and lags
- added delay based on position from center of the AOE into effect and feedback applications

### **Entangle**

- added caster validity check (aoe will vanish with caster)

### **Find Traps**

- on hardcore difficulty will now reveal trap position for everyone in party including all associates and even players in different area

### **Firestorm**

- fixed bug that prevented the spell to affect placeables and doors.

### **Flame strike**

- spell will no longer damage allies under high difficulty settings

### **Freedom of movement**

- added supernatural effect check as is in restoration spells - this allows to make a movement decrease effect which won't be removed with this spell

### **Ghoul Touch**

- removed empower and maximize calculation from duration

### **Grease**

- added caster validity check (aoe will vanish with caster)

### **(Greater) Magic Weapon**

- Magic Weapon and Greater Magic Weapon will apply attack bonus instead of enhancement if ever targets a ranged weapon or gloves

### **Identify**

- added custom content support when the spell is modified to cast on item instead of caster, in such case the spell will identify targeted item without any check

### **Implosion**

- a small code improvement that allows to modify this spell to affect caster without need to code anything else to prevent the bug with automatic death without any roll

### **Mind fog**

- fixed a rare situation where the penalty from the fog wasn't properly removed after leaving the aura

### **Lesser Dispel Magic**

- added verbal component and appropriate chants (*in all sources DnD3.0/3.5/NWN2 this spell has verbal component*)

### **Power Word: Kill**

- immunity check didn't correctly passed caster into consideration

### **Protection against elements**

- duration unified with resist elements and energy buffer to 1 turn/level (*simply logical decision as these spells are implemented differently than in DnD*)

### **Protection against spells**

- spell target area changed to the large as per spell's description

### **Remove Fear**

- spell target area changed to the large as per spell's description

### **Searing Light**

- damage dice against undead targets raised from k6 to k8 to match spell description

### **Shield of Faith**

- removed the visual effect that was shared with protection against evil (*to prevent mistaking these two*)
- added "cessate" visual effect (*this will run at the end of the duration as a kind of feedback "you just lost some beneficial spell"*)

### **Silence**

- spell cast event is now signaled regardless of spell resistance result

### **Sleep**

- shape size wasn't correct (started with HUGE then continued with LARGE, which caused issues)

### **Tenser's Transformation**

- fixed losing the hp/attacks/fort when repolymorph happened
- fixed not losing temporary hps when polymorph was canceled before the actual spell expired

### **Tide of Battle (*unimplemented spell, however scrolls are available*)**

- added missing target type - which allowed to hurt anyone even in No-PvP area

### **Time Stop**

- added an immunity feature - builder can now make creatures immune to the time stop; from balance reasons, this immunity wasn't added to any vanilla creature though
- added freeze animations visual effect
- in server mode and if nwn(c)x\_patch is running, the spell will now show a timestop icon.

### **Wail of Banshee**

- number of targets pool was decreased even for targets that were skipped

### **Web**

- added caster validity check (aoe will vanish with caster)

### **All creature auras (blind, fear, menace, stun, unnatural, dragon fear)**

All effects from auras and similar abilities made supernatural as per DnD rules; this has basically only one effect, to prevent the same monster who stunned/feared/blinded you with his own aura to dispel that effect from you with Mordenkainen's Disjunction or beholder dispel.

### **Barbarian Rage**

- levels in Eye of Gruumsh class will stack with barbarian for purpose of determining greater rage bonuses

### **Bard Song and Curse Song**

- improved lvl 30 song attack bonus, damage, will, fort and reflex by +1 from balance reasons

### **Blinding Speed**

- no longer requires concentration and thus cannot be interrupted anymore
- effects made undispellable (extraordinary)

### **Chaos Slaad Spittle**

- removed ground targeting as it caused misfires.

### **Chaotic shield**

- effects made undispellable (extraordinary)

### **Choking powder**

- effects made undispellable (extraordinary)

### **Deafening Clang**

- the deafness effect from on-hit is now undispellable (extraordinary)

### **Dragon fear breath**

- due to the bug in new code, the promised fix for too long duration from 1.71 didn't work
- effects made supernatural (not dispellable)
- implemented missing effect scaling

### **Dragon breath paralyse**

- effects made supernatural (not dispellable)

### **Dragon breath sleep**

- effects made supernatural (not dispellable), added scaling into duration

### **Dragon breath slow**

- effects made supernatural (not dispellable), saving throw type changed to fortitude as per DnD rules

### **Gelatinous cube paralysis**

- effects made undispellable (extraordinary)

### **Harpy song**

- added deafness failure

### **Fear (Purple Dragon Knight version)**

- added missing effect and duration scaling feature into PDK's fear spell version

### **Thundering rage**

- when activating thundering rage with ranged weapons, the deafness on-hit property will be added onto ammunition instead (*as this property doesn't work on ranged weapons*)

### **Thunderstone**

- added sonic saving throw subtype, effects made undispellable (extraordinary)

### **Tymora's Smile**

- effects made supernatural (not dispellable)

### **Poison rules change**

- all poison special effects such as blindness or ability decrease now controls poison immunity, if poisoned character acquired poison immunity these effects will fail

### **Blue Whinnis poison**

- the sleep effect will be removable by neutralize poison spell now

### **Pit Fiend Ichor poison**

- the death will ignore death spells immunity only on hardcore and high difficult difficulty

## **Fixes to the 1.69 Horse System**

Let me start with saying that 1.69 Horse System is a complete mess. There are just so many issues and limitations that I even wonder this could get into a official patch...

The major problem here are limitations. Graphical lags are caused by the limitation of the game graphics engine. Issues like not being able to mount horse if you have effects of divine power spell are caused by limitation of the scripting etc. I tried to fix these issues as best I could, but in the end it is still far from perfect and will still cause similar issues if you ever try use 1.69 Horse System with custom content.

Note, that because horse system is a scripting system, the fixes and improvements from CP will work only in case module didn't modified any of these scripts: `x3_mod_def_enter`, `x3_x3_horse`, `x3_inc_horse`, `x3_s2_palmount`.

### List of fixes and improvements:

Removed 'null tail', existing characters with null tail should automatically lose it sooner or later provided there is no modification into horse scripts in module. Null tail is a special workaround to fix graphical lag when loading horse, unfortunately it is causing **massive graphical lags** just as horse itself when other characters meet character with null tail and from my tests it has really no impact on the animation lag that it is supposed to help with. The null tail could have been added to your players' characters even if your module doesn't have any horses.

Fixed major issues with relogging/reloading with mounted character. Relogging or reloading game while mounted could result into losing informations about mount and thus character logged back mounted but the game thought he isn't. Another issue is that relogging updated character height (space he occupies) - while this is actually right, since the update was happening only when relogging and thus character height was most of the time incorrect, community patch removes the mounts' height change.

Fixed possibility to mount horse that couldn't be reached (behind fence, pit etc.).

Fixed issue where player with effect of divine power spell couldn't mount horse. *Note, that the problem here is that the horse system is trying to make sure you cannot mount horse under knockdown effect but knockdown doesn't have constant and therefore shows as invalid effect type in scripting which the horse system checks for. Unfortunately this is not the only effect that has no constant, EffectModifyAttacks also returns invalid effect type which is reason why mounting wasn't possible. I could not fix this limitation, so I just simply added an exception to the code where I check for divine power spell id. This makes player to be able to mount horse with divine power but any custom content spell using EffectModifyAttacks will still cause original issue.*

Fixed nonfunctional paladin mount speed increase.

Fixed attack penalty stacking, and save-load issues with effects, all horse-related effects improved and reworked and are now custom content compatible; they are now created by unique object so they cannot be confused with different effects that player might have from custom content.

Fixed issue when the horse system changed player's appearance when (re)logging if player was polymorphed or when he had non-standard race or appearance.

And lastly, horse-related penalties won't be removed by cure spells such as restoration. *Coded inside the spellscripts, not in horse engine thus whether this is functional depends on whether you use spellscripts from community patch.*

## Character builds and gameplay concerned

1.72 contains only a very few changes/fixes affecting balance as these kind of changes are hot topic. Here's the list of them:

**Bard Song and Curse Song lvl 30 improvement** - Bard of lvl 30 gets +1 attack bonus, +1 damage bonus, +1 will, +1 fortitude and +1 reflex. This change was made in order to add more options when building a high level bard character. This was thoroughly tested and calculated and Bard 30+ builds are, even with this change, still weaker than builds with bard 20-26 only and multiclass with Red Dragon Disciple and combat classes.

**Animal companion damage rebalance** - this is described above in creature fixes/changes. Reasons behind are simple. In vanilla you have basically 2 choices when selecting animal companions. Either you pick the strongest one - bear, as he is beefiest of them and can tank/distract longer than others. Or you pick another animal usually from roleplay reasons or just because you like badgers. The changes in 1.72 are trying to add some additional thoughts and benefits for selecting animal companions and also bring the nwn animal companions closer to the DnD rules.

**Possessing animal companions** - this is feature from nwn\_cx\_patch which further improves animal companions. Possessing animal companions offers full control over their actions and you can also use them to scout areas or other actions for which different type of animals are better suited than others which further enhances the decisions behind selecting one.

Slightly increased reach of the attacks made with whip (should allow to attack distant placeables like in Indiana Jones).

Under hardcore and very high difficulty, reading scrolls won't be possible under blindness/darkness anymore, unless character has true seeing or ultravision effects to overcome the conditions.

Except these changes, what significantly impacts gameplay balance are AI fixes and improvements. Creatures and henchmen might behave bit differently than in vanilla due to these changes. This is inevitable drawback of making changes like this and in rare cases it can make encounter or boss much more difficult than before.

## General AI fixes and improvements

These fixes apply for all creatures including henchmen.



### List of improvements:

- fixed bug in Sneak Attack code that allowed henchman to attack his master
- fixed the AI engine so it can use any summoning feat (*which normally doesn't work because of bug in ActionUseTalentAtLocation which is why summoning feats are disabled by default*)
- fixed weird AI issue where creature with ranged weapon threatened in melee would often reequip her weapon
- fixed combat of huge creatures like dragons, who had much higher chance to magic than intended because their target was never "close enough"
- fixed an issue that could sent spellcasting creature to start melee attack before she runs out of spells
- no more spell/feat decrementing when talent doesn't pass filter checks - this was big problem with sorcerers
- creatures can now use taunt, provided they have a ranks in it, they are in close range to the target, they can actually succeed and the opponent has high AC
- creatures will try to find other enemy if their current enemy is petrified
- smarter cure behavior:
  - creatures will now be able to cure fear, mind-affecting effects, mobility impairing effects and petrify if they have right spell to do so
  - creatures will no longer try to cure level drain with lesser restoration
- better see invisible behavior:
  - creatures will try to use revealing talents if they lose track of their last enemy
  - creatures will try to use ultravision in darkness
  - creatures will be able to use also potions and non equippable items with revealing spells
  - creatures will use detect mode if they don't have any revealing spells or if those spell didn't revealed anyone
- creatures will use restoration - others and cure critical wounds - others spells properly
- improved healing of others, creatures won't use full heals if the target is not badly wounded and generally use lesser healing spells when dealing with less wounded creatures (*1.71 improved only self healing behavior*)
- enabled undead creatures to heal their undead allies with negative energy spells
- additional AI spell usage checks:
  - creatures won't cast haste if they got item with haste itemproperty
  - creatures won't cast stone bones on non-undead target
  - creatures won't use see invis, ultravision, true seeing talents if they got true seeing effect or item
  - creatures won't use any sort of invisibility talent if they are already invisible
  - creatures won't use sanctuary if they are under greater sanctuary effect
  - creatures won't use minor globe of invulnerability if they are already under effect of different/stronger version of this spell

- creatures won't use ghostly visage if they are already under effect of different version of this spell or ethereal visage spell
- creatures won't use any sort of spell mantle talent if they already have any spell mantle active
- creatures won't cast daze spell on targets with more than 5HD
- creatures won't try to substitute inflict spells when casting them on self/ally (in these cases it is on purpose)
- the haste and true seeing checks will now be also verified on items, creatures with these effects on item properties won't waste time casting these spells
- significantly improved AI for polymorphed creatures introduced in 1.71:
  - unwillingly polymorphed creatures will attempt to cancel polymorph if they are allowed to do that (shapechangers/shifters)
  - creatures polymorphed into chicken, penguin and cow will no longer try to fight and will flee from enemies
  - AI will now handle spellcasting while in polymorph with following rule: everything that's not an ordinary spell (UserType != 1) will be allowed to cast even polymorphed, if its ordinary spell and the innate level is 5 or higher, then the polymorph will be canceled and creature will cast this spell
  - however, creatures won't cancel polymorph earlier than after 3 rounds to avoid dumb situations
  - AI will now handle also polymorph self, tensor's transformation and shapechange polymorphs
  - AI will not use low CR polymorphs if the CR of the creature is way higher
  - AI will keep a track of the shifter form spell limits and won't use talent which has all uses depleted
- implemented "intelligence" check, intelligent creatures will avoid use spells on immune and such. Intelligence is determined via AI level (High/Very High) or racial type (player races, dragons, feys, outsiders, magical beasts, aberrations) or base intelligence score 10+ excluding vermins, oozes, animals and beasts (*note, many of these checks were there already, now will be used only by smart creatures*)
- readded check for using drown and level drain usage spells on naturally immune targets which was disabled in 1.70
- Tasha hideous laughter added into AI purpose mind-affecting spells list (*which will prevent casting this spell on immune targets if the creature is smart enough*)
- creatures will be able (with 50% chance) to cast silence spell at location instead of at target, granting no saving throw roll
- creatures will be able to count with the limit override in Power Word: Kill spell. That limit however must be enforced with spell id variable, ie. "131\_LIMIT\_OVERRIDE" either on creature or module
- rebalanced some of the spells and feats 2DA settings for a purpose of AI cast order
- and few small improvements in the AOE behavior to make it less problematic

## Henchmen specific AI fixes and improvements

These fixes apply only for henchmen, familiars and animal companions.

### List of improvements:

- fixed henchmen AI issue that could cancel ongoing spellcasting action
- fixed henchmen AI issue that could leave henchman unwilling to fight by his own
- fixed henchman healing in certain situations where it was broken before
- improved barbarian rage handling, creature using barbarian rage will immediately run into combat and won't stand flat-footed any longer
- henchman and other PC associates will be able to resurrect their master if they possess such talents
- improved intentional trap triggering by a henchman. It's still not perfect due to the engine limitations, but it's not more reliable than before.
- henchman will be now also able to handle trap that only his master can see
- added the "Cannot do" voice confirmation for a disarm trap henchman command when there is no trap around
- when explicitly told to deal with trap from radial command, the henchman conversation option to not help with traps will now be ignored

## High AI creatures improvements:

High AI is a new feature from 1.71. To make high AI monster use 70\_c2\_aihigh9 script in OnSpawn event or call SetAIlevel function with AI\_LEVEL\_HIGH parameter manually.

### List of improvements:

- high ai creatures will be able to summon epic dragon, epic mummy and fiendish servant
- high ai creatures will be able to use mass heal, healing circle, circle of doom and negative energy burst to heal their allies
- high ai creatures will be able to use Greater Restoration to heal self or their allies
- high ai creatures will use breach spells only when the enemy has spells which breach removes
- high ai creatures will try to substitute dispel spells for breach, if enemy has spell mantle/shadow shield
- high AI creatures and henchmen will be able to cast defensively with following principles:
  - chance to use defensive casting mode now depends on the chance to succeed in the check (at least 25%) and a the number of enemies threatening in melee
  - defensive casting won't be used if creature is in Expertise/Improved Expertise mode

## Other fixes

## 2DA based fixes:

- spells.2da:
  - fixed AI category for summon shadow spells (all 4 variants)
  - fixed AI category for true seeing (was TALENT\_CATEGORY\_BENEFICIAL\_CONDITIONAL\_AREAEFFECT while single target)
  - fixed Master reference for several shapeshifting subspells
  - removed ground targeting from the Chaos Slaad Spittle special ability as it caused misfires
- polymorph.2da: racial type of the super chicken polymorph form corrected to animal (was gnome)
- feat.2da: Blinding speed AI category changed to aura in order to force to use this feat as soon as possible and without checking close combat
- Baseitems.2da
  - creature skin set to have no "height" in order to workaround full inventory issue
  - ugly DOA cloak model replaced with standard cloak (*this actually appeared only when placing cloak item from palette to area, but now it appears 100% matching with what you see at ground once you place it*)

## Blueprint/model based fixes:

- scroll of Negative Energy Protection set identified
- sea surf small sound object: fixed the interval variation which was incorrectly set to 0.05 instead of 5 seconds
- sea surf large sound object: fixed the volume which was zero before
- fixed nonfunctional Rain Heavy and Rain Light sound objects
- x0\_plc\_wsd - fixed secret stone doors to be closed by default
- fixed OnSpawn script in multiple HotU summon blueprints (*this fixes issue where these summons weren't respecting multisummon module switch*)
- fixed several creature blueprints having erroneous wings and tails assigned

## Fixes and changes in various scripts:

- HotU associate conversation made custom content compatible (*if player had custom content familiar or animal companion he couldn't talk to him before*)
- x2\_s2\_dyearmor: Fixed losing undroppable/cursed flag after dyeing an item
- x2\_im\_nextpart: Fixed losing undroppable/cursed flag in HotU crafting
- x2\_im\_prevpart: Fixed losing undroppable/cursed flag in HotU crafting
- nw\_02\_onoff: improved placeable On/Off script, will attempt to activate/deactivate nearest sound object with same tag ([based on script from vault](#))
- nw\_ch\_ac1: enforced stealth mode when explicitly told to from associate conversation (which worked only in case that player ended conversation right after the spoken command)

- nw\_ch\_ac4: fixed bug that allowed to speak with associate in disable states such as petrify
- nw\_ch\_ac4: animal companions, familiars, summoned creatures and even dominate creatures will now use HotU associate conversation which allows to command cast spells
- nw\_ch\_action\_13 (default vanilla henchman "when join" script): made the script compatible with increased number of henchmen via custom content
- nw\_ch\_summon\_9 - henchmen will stay closer to the master as was intended
- x2\_im\_can\* scripts: the variables to disable item crafting can now be used also locally, ie. item with these variables won't be allowed to change
- X2\_mod\_def\_load - corrected comment above Use Poison switch
- x3\_mod\_def\_load - will now execute x2\_mod\_def\_load from compatibility reasons
- gargoyle/skeleton/zombie placeable scripts: added secondary protection to undisPELLable AOE's which will prevent AOE from being dispelled (destroyed) even in the module which doesn't use CP dispel scripts
- fixed few more scripts that could generate PC Skin item (*this was partially fixed in 1.70 which fixed creation of the skin from spells, however there were other scripts that could generate it*)

#### Fixes and changes in various include scripts:

- nw\_i0\_generic: generic creature ai detached into 70\_ai\_generic script to allow to change internal AI without need to recompile all scripts
- nw\_i0\_spells: MySavingThrow will ignore paralysis immunity if the module switch 72\_DISABLE\_PARALYZE\_MIND\_SPELL\_IMMUNITY is activated
- nw\_i0\_spells: DoSpellBreach feedback printed also for caster if its player
- nw\_i0\_spells: removed duplicated shadow shield on the breach spell list
- x0\_inc\_henai: henchman ai detached into 70\_ai\_henchman script to allow to change internal AI without need to recompile all scripts
- x0\_i0\_anims: function CheckIsCivilized() will now return TRUE for dwarfs as well
- x2\_inc\_itemprop: added new function IPGetIsThrownWeapon
- x2\_inc\_itemprop: function IPGetItemPropertyByID will now support all vanilla itemproperties (28 were missing, excluding those from 1.69 which were added in 1.70)
- x3\_inc\_horse: HorseGetIsMounted changed in a way it checks phenotype instead of integer on skin - this change should prevent creation of the PC Skin in a modules without horses

## New features in toolset

#### Various new features

- added blueprints for forgotten sound effects from patch 1.69, it is 8 sounds and you can find them in new category Community Patch at the top of the default palette

- added two new hidden visual effects in visualeffects.2da, VFX\_DUR\_QUESTION\_MARK and VFX\_DUR\_EXCLAMATION\_MARK, the constants will be available in script editor. *(Note: this will be visible only to players with this community patch version installed)*
- added feature to override maximum spell level to brew potion/craft wand. This can be overridden by the specific variable on module or player:
  - int X2\_CI\_CRAFTWAND\_MAXLEVEL value=1-9 - will override maximum spell level that can be crafted into wand
  - int X2\_CI\_BREWPOTION\_MAXLEVEL value=1-9 - will override maximum spell level that can be crafted into potion
- enabled cleric domains to grant spell of 0th level; a new column Level\_0 has been added into domains.2da file
- added new packages in packages.2da for Shou Disciple and Eye of the Gruumsh to be used with LevelUpHenchman function, the constants for these packages (plus PDK from 1.69) will be available in script editor.

## Further expanded itemproperties

Added some more options into what item property you can add to various items. This is further expanding all the possibilities provided by CP 1.71. Note that this doesn't change any pre-existing items, it only gives new possibilities in toolset for builders.

- Added more choices to throwing weapons: AC bonus, AC bonus vs. alignment/class/race, Cast spell, Damage immunity/reduction/resistance, Holy avenger, Immunity, Improved evasion, Improved saving throws, Immunity to specific spell/spell school/spells by level, True seeing, Freedom of movement.
- Throwing weapons will also offer Boomerang itemproperty, however this itemproperty will be functional only when using NWN(C)X.
- Cast spell itemproperty will be available also on healer kits and thieves tools.
- Attack, damage and enhancement penalty properties will be available on all equippable items *(yes it works)*.
- Freedom of movement will be available on mage staves.
- Limitation alignment/racial type will be available on scrolls.
- Bonus feat and bonus spells slots will be available on creature weapons.
- Allowed to use 0charges/use, 1-5 uses/day and unlimited uses/day on wands (previously only allowed on rods and mage staves).
- Added all standard feats as a bonus feats, starting on line 100. Some feats were excluded though: those that were already available and those that doesn't work as bonus feats or has undesired side-effects. This is specifically tailored to be easily used by scripting. To create a new bonus feat itemproperty with chosen feat via script, use `ItemPropertyBonusFeat(100+FEAT_* constant)`; just check if the property is valid - if not

it's either one of the feats that has the IP\_CONST\_BONUS\_FEAT\_\* counterpart or that doesn't work as bonus feat.

### Module switches:

Those not familiar with module switches should read [this tutorial](#). The documentation here will only list module switches added in 1.72 or new possibilities for older switches.

#### **Module Switch name:**

MODULE\_SWITCH\_CURSE\_IGNORE\_ABILITY\_DECREASE\_IMMUNITY

**Variable name:** 72\_CURSE\_IGNORE\_ABILITY\_DECREASE\_IMMUNITY

**Description:** This switch will modify the curse to bypass the ability decrease immunity such as from negative energy protection spell.

**Note:** Dependant on NWN(C)X\_Patch plugin.

**Module Switch name:** MODULE\_SWITCH\_DISABLE\_DAMAGE\_SHIELD\_STACKING

**Variable name:** 72\_DISABLE\_DAMAGE\_SHIELD\_STACKING

**Description:** This switch will allow to use only one damage shield spell at once: elemental shield, mestil's acid sheath, aura vs alignment, death armor, wounding whispers. When one of these spells is cast, any other such spell is dispelled.

**Module Switch name:** MODULE\_SWITCH\_DISABLE\_WEAPON\_BOOST\_STACKING

**Variable name:** 72\_DISABLE\_WEAPON\_BOOST\_STACKING

**Description:** This switch will allow to use only one weapon enhancement spell at once: magic weapon, greater magic weapon, bless weapon, flame weapon, holy sword, deafening clang, keen edge, darkfire and black staff. When one of these spells is cast on same item, the item gets stripped of all temporary itemproperties.

**Module Switch name:** MODULE\_SWITCH\_DISABLE\_AOE\_SPELLS\_STACKING

**Variable name:** 72\_DISABLE\_AOE\_SPELLS\_STACKING

**Description:** This switch controls how much persistent AOE spells of the same type can player cast in the same area. This will stop the cheesy tactics to stack dozen of blade barriers or acid fogs and lure monsters into it. Unlike weapon boosts and damage shields, this switch allows to set specific number of allowed AOE spells, so 1 = max 1, 2 = max 2 etc.

**Module Switch name:** MODULE\_SWITCH\_HARDCORE\_UNCANNY\_DODGE

**Variable name:** 72\_HARDCORE\_UNCANNY\_DODGE

**Description:** This switch will enable hardcore DnD rules for uncanny dodge 2 and sneak attack. Character with uncanny dodge II can no longer be flanked. This defense denies another rogue the ability to sneak attack the character by flanking her, unless the attacker has at least four

more rogue levels than the target does. Classes granting uncanny dodge stacks together for a purpose of this calculation.

**Note:** Dependant on NWN(C)X\_Patch plugin.

**Module Switch name:** MODULE\_SWITCH\_HARDCORE\_EVASION\_RULES

**Variable name:** 72\_HARDCORE\_EVASION\_RULES

**Description:** This switch will enable hardcore DnD rules for evasion and improved evasion. Evasion feats will only work in light or no armor. Also a character must not be helpless ie. under effects of stun, paralysis, petrify, sleep or timestop.

**Module Switch name:** MODULE\_SWITCH\_ENABLE\_FLYING\_TRAP\_IMMUNITY

**Variable name:** 72\_ENABLE\_FLYING\_TRAP\_IMMUNITY

**Description:** This switch will give flying creatures immunity to all ground traps. Specific traps can be set to ignore this immunity via "DISALLOW\_FLYING" int 1 variable on trigger.

**Note:** Dependant on NWN(C)X\_Patch plugin.

**Module Switch name:** MODULE\_SWITCH\_DISABLE\_MONK\_ABILITIES\_IN\_POLYMORPH

**Variable name:** 72\_DISABLE\_MONK\_IN\_POLYMORPH

**Description:** This switch will disable all monk abilities in polymorph. That is monk AC from high wisdom, monk AC from class levels, monk speed and monk unarmed attack progression.

**Note:** Dependant on NWN(C)X\_Patch plugin.

**Module Switch name:** MODULE\_SWITCH\_DISABLE\_PARALYZE\_MIND\_SPELL\_IMMUNITY

**Variable name:** 72\_DISABLE\_PARALYZE\_MIND\_SPELL\_IMMUNITY

**Description:** This switch will detach immunity to paralysis from the immunity to mind spells. This is useful if you want to nerf the classic mind immunity spells and give higher meaning to the freedom of movement, PM/RDD or items with paralysis immunity.

**Note:** Dependant on NWN(C)X\_Patch plugin.

**Module Switch name:**

MODULE\_SWITCH\_DISABLE\_SNEAK\_ATTACK\_CRITICAL\_IMMUNITY

**Variable name:** 72\_DISABLE\_SNEAK\_CRITICAL\_IMMUNITY

**Description:** This switch will detach immunity to sneak attacks from the immunity to critical hits. This will allow to make a creature immune to critical hits but not sneak attack. Applies also to the Death Attack.

**Note:** Dependant on NWN(C)X\_Patch plugin.

**Module Switch name:** MODULE\_SWITCH\_DISABLE\_TUMBLE\_AC\_BONUS

**Variable name:** 72\_DISABLE\_TUMBLE\_AC

**Description:** This switch will completely disable all AC bonuses from tumble.

**Note:** Dependant on NWN(C)X\_Patch plugin.

**Module Switch name:** MODULE\_SWITCH\_DEVASTATING\_CRITICAL\_ONCE\_PER\_TARGET



**Variable name:** 72\_DEVAST\_ONCE\_PER\_TARGET

**Description:** This switch will enforce a one roll only rule for devastating critical ability. This is a very efficient method of making this feat more balanced yet still allow to slain a monster with single blow as this was designed.

Value of 1 - works for everyone

Value of 2 - works only for players, monster's devastating isn't affected

**Note:** Dependant on NWN(C)X\_Patch plugin.

**Module Switch name:**

MODULE\_SWITCH\_POLYMORPH\_DISABLE\_POLYMORPH\_END\_CHECK

**Variable name:** 72\_POLYMORPH\_DISABLE\_POLYMORPH\_END\_CHECK

**Description:** This switch will disable "polymorph end" check which is performed every 6 seconds via pseudo heartbeat in order to clean all polymorph related effects such as ability bonuses, temporary hp etc. in case a module doesn't have properly merged module events with 1.72. Activating this switch will disable this check which is useful in multiplayer to make the polymorph new system more optimized. Make sure that you got OnEquip and OnUnEquip events merged properly (or you are using NWN(C)X) before disabling this!

**Module Switch name:** MODULE\_SWITCH\_POLYMORPH\_MERGE\_EVERYTHING

**Variable name:** 72\_POLYMORPH\_MERGE\_EVERYTHING

**Description:** This switch will allow to merge every items the character wears into every polymorph shape in game even Tenser's transformation. This automatically enable the "merge arms" switch.

**Note:** for unarmed shapes only defensive properties from weapon will merge.

**Module Switch name:** MODULE\_SWITCH\_POLYMORPH\_MERGE\_CASTING\_ABILITY

**Variable name:** 72\_POLYMORPH\_MERGE\_CASTING\_ABILITY

**Description:** This switch will merge intelligence, charisma and wisdom from all items no matter if the shape merges them or not. The reason for this is when you want to stop losing bonus spell slots from ability increases on items while polymorphed. In case the player is a monk, he will get AC decrease matching the increase in wisdom over what shape normally allows.

**Note:** use this switch only when module is not running NWNX. NWN(C)X\_Patch handles this automatically in a better way - slots which would be normally lost will only be consumed. Also note this won't fix, unlike NWNX, losing spell slots from bonus spell slot itemproperties.

**Module Switch name:** MODULE\_SWITCH\_ALLOW\_BOOST\_THROWING\_WEAPONS

**Variable name:** 72\_ALLOW\_BOOST\_THROWING

**Description:** By setting this to TRUE, weapon-boost spells like flame weapon will be able to target and boost throwing weapons just like any other weapon.

**Module Switch name:** MODULE\_SWITCH\_ALLOW\_BOOST\_AMMO

**Variable name:** 72\_ALLOW\_BOOST\_AMMO

**Description:** By setting this to TRUE, weapon-boost spells like flame weapon will be able to target and boost ammunition just like any other weapon.

**Note:** for unarmed shapes only defensive properties from weapon will merge.

### **Improvements into module switches from 1.71:**

Module switch to restrict musical instruments **71\_RESTRICT\_MUSICAL\_INSTRUMENTS** now allows value of 3 which combine both perform DC and bard song use restrictions. This switch will also be checked on an item and it will override module settings - thus it will be possible to enforce extra rules only for a specific musical instrument.

Module switch to stack ability bonuses in polymorph

**71\_POLYMORPH\_STACK\_ABILITY\_BONUSES** now also stacks bonuses to skills and saving throws.

Module switches **71\_POLYMORPH\_STACK\_ABILITY\_BONUSES**, **71\_POLYMORPH\_MERGE\_ARMS** and **71\_UNLIMITED\_SUMMONING** can now be used locally; ie. you can apply these rules only for certain PC, not just for everyone. To do this, use SetLocalInt on the player with corresponding module switch variable and value.

### **Item/Creature switches:**

Setting an integer variable called "**IMMUNITY\_DYING**" to **1** on a player character will allow him to ignore dying. As such he will be able to control his character even with hit points under 1.

This feature works only on players as NPCs die immediately on 0 hit points.

**Note:** Dependent on NWN(C)X\_Patch plugin.

Setting an integer variable called "**IMMUNITY\_FLANKING**" to **1** on a creature will guarantee this creature immunity to being flanked. That grants immunity to sneak attack caused by flanking and virtually +2AC as the flanking attackers won't get the +2AB bonus to hit.

**Note:** Dependent on NWN(C)X\_Patch plugin.

Setting an integer variable called "**IMMUNITY\_DEVAST**" to **1** on a creature will guarantee this creature absolute immunity to devastating critical ability.

**Note:** Dependent on NWN(C)X\_Patch plugin.

Setting an integer variable called "**IMMUNITY\_POLYMORPH**" to **1** on a creature will guarantee this creature absolute immunity to hostile polymorph abilities.

**Note:** This doesn't prevent the creature from using her own polymorphing talents.

**Note2:** There is no ability which could polymorph other creatures in vanilla, this is an extra feature for custom content made spells and abilities.

Setting an integer variable called "IMMUNITY\_TIMESTOP" to 1 on a creature will guarantee this creature absolute immunity to time stop.

**Note:** This works only with time stop spellscript from community patch.

Setting an integer variable called "72\_DISABLE\_ENCHANTMENT\_SPELLS" to 1 on an item disallows weapon or armor enhancement spells such as Flame weapon to work on this item.

### Custom scripts:

- 70\_ai\_generic - script that implements generic vanilla AI
- 70\_ai\_henchman - script that implements henchman vanilla AI
  - the purpose of these two scripts is to allow make changes into how AI functions without need to recompile all scripts that calls DetermineCombatRound function
- 70\_c2\_herbivore - custom OnSpawn script that makes animal to use [Enhanced Creature AI for Herbivores from LoCash](#)
- 70\_c2\_omnivore - custom OnSpawn script that makes animal to use [Enhanced Creature AI for Omnivores from LoCash](#)
- 70\_featfix: script that fixes darkvision (and low-light vision) feat to work when gained by Pale Master, Red Dragon Disciple or Shadowdancer or any custom content. Script works only when module is running NWN(C)X.
- 70\_s2\_epicrage: externalized function 'CheckAndApplyEpicRageFeats' from x2\_i0\_spells include into this script in order to make changes into this function independent on recompiling all scripts that calls it
- added four new multispell scripts, the point of this is to enable builder to modify all these spells in single script, note these aren't used by default, builder must install it via auto 2da merger
  - 70\_s0\_stoneskin (For stoneskin (and shadow variant), greater stoneskin and premonition)
  - 70\_s0\_elements (for endure elements, resist elements, protection against elements and energy buffer)
  - 70\_s0\_abilbuff (for all ability buff spells, including Harper and BG version and their greater versions, but excluding owl's insight)
  - 70\_s3\_iounstone (for all ioun stones)

### Custom module events:

Community patch adds several new custom module events.

### Scripted vanilla non-nwnx events:

- 70\_mod\_dispel - new script that implements the dispel effect called from spellscripts in order to make changes into how dispel functions independent on recompiling all scripts that calls it

- 70\_mod\_resurrect - new script which will fire after casting raise dead or resurrection spell
- 70\_mod\_petrified - new script which will fire after petrify effect application from any spell/ability
- 70\_mod\_polymorph - new script that handles polymorphing and related features such as item merging

#### **NWNX\_Patch related events:**

- 70\_mod\_leveldown (OnLevelDown)
- 70\_mod\_attacked (OnPhysicalAttacked for players)
- 70\_mod\_damaged (OnDamaged for players)
- 70\_mod\_effects (OnEffectApplied/OnEffectRemoved)
- 70\_mod\_hlight - OnPlayerHighlightObjects (aka press TAB)

#### **Community patch default module event scripts:**

This is a bit special. Some of these fixes were added in 1.71 already but because they were incorporated into vanilla HotU default module event scripts, these fixes worked basically only in official campaigns. They didn't work in custom singleplayer modules or persistent worlds by default. Module builders had to merge changes from these scripts into their own scripts to take effect of these fixes.

So this was revisited and in 1.72 this works differently. All these scripting changes were withdrawn from the vanilla scripts and coded into Community Patch default scripts (prefix 70\_mod\_def\_\*). These scripts will run automatically in every module if nwnx\_patch or nwncx\_patch is running. This means that these fixes and improvements will now work in every module you play. Similarly, on a PW - you no longer need to merge anything and all fixes below will work automatically without any additional work.

In case you don't want to use NWN(C)X\_Patch or you feel like running 2 scripts for one event is inefficient you should merge the code from these scripts into your own scripts for those events. To disable this feature in NWN(C)X\_Patch set the plugin switch DisableModuleEvents to 1. For more information read [plugin documentation](#).

Note, that from efficiency reasons, Community Patch uses and provides only a few default module event scripts. Only scripts existing in Community Patch are listed here, but the default module scripts run for all events, not just for these few. For more information read [plugin documentation](#).

- 70\_mod\_def\_aqu - craft duping protection + code for module switch allowing to keep items given to the vanilla henchman after his level-up
- 70\_mod\_def\_enter - code for darkvision/low-light vision fix
- 70\_mod\_def\_equ - code for new custom Wounding itemproperty + code for new polymorph engine + code for nasher set items extra bonuses

- 70\_mod\_def\_load - sends nwn(c)x\_patch plugin global module switch values, sets the flying creature trap immunity switch
- 70\_mod\_def\_lvup - this script controls few levelup exploits and delevels exploited player characters + code for darkvision/low-light vision fix
- 70\_mod\_def\_resp - upon respawn, player will be stripped of every effect, not just negative ones + outgoing AOE spells cast by the respawning PC in the area PC is in, will be dispelled + double-respawn protection code.
- 70\_mod\_def\_unaqu - fix for automatic unarmed attack when stack of throwing weapons gets destroyed + code for overfilled store automatic cleaning module switch + code for module switch allowing to keep items given to the vanilla henchman after his level-up
- 70\_mod\_def\_unequ - code for special workaround to solve skin issue with polymorph and relog + code for nasher set items extra bonuses

## Improvements into community patch “spell engine”

Before you start reading this, if you are not familiar with the spell engine from community patch you should read the [documentation](#) for it from 1.71. This will explain the basics which the information on this page counts with.

If you run server via NWNX or client via NWNCX the old limitation where the modified caster level was not reflected against dispels or monk spell resistance will be fixed! It should now work perfectly as expected.

The community patch spell engine was significantly improved to allow all kinds of stuff that would normally need to manually modify all spell scripts. Besides overriding or modifying caster level, metamagic and DC, in 1.72 you can modify almost everything: spell aoe range, duration type, actual spell duration, damage cap, damage type, saving throw type, limit (such as number of missiles for IGMS), target type and even add saving throws or spell resistance into spells that doesn't check it or remove it from spells that does as well.

Plus, this all can now be done:

- for a specific spell only
- for a specific item (with spells) only
- for a specific item and specific spell only
- for a specific creature only
- for a specific creature and specific spell only
- globally, for a specific spell but for everyone in module

This is even calculated together. So if you make a global/module change to the fireball to increase damage cap by +5, therefore making each fireball in module do up to 15d6 damage, if you apply the same modifier to the fireball only for player, it will count both modifiers together and will make fireball to deal up to 20d6 damage. This works only with modifiers, in case of

overrides, the priority is following: specific spell on creature/item override > any spell on creature/item override > global specific spell override.

And again, this all can be done externally without touching the spellscripts. To apply these changes all you have to do is to set a local variable on creature/item/module, either manually or from script.

## TODO

First of all, new variables to supplement already existing amplifiers from patch 1.71:

Setting an integer variable called "**X\_CASTER\_LEVEL\_OVERRIDE**" on item or caster, where X is a numeric spell ID constant will override caster level of this spell with the value you desire. This variable has higher priority than old variables from 1.71 so you can use this if you want to raise all creature special abilities to level 40 (which toolset doesn't allow normally) using `SPECIAL_ABILITY_CASTER_LEVEL_OVERRIDE = 40` and alter single spell to use a different value, perhaps `368_CASTER_LEVEL_OVERRIDE = 80`. In this example creature will cast all special abilities under caster level 40 and ice storm with caster level 80.

- new spell overriding variable, use int `X_DC_OVERRIDE` on item or caster to override DC of the spell with ID of X, example `87_CASTER_LEVEL_OVERRIDE int 60` = implosion with DC 60
- new spell overriding variable, use int `X_METAMAGIC_OVERRIDE` on item or caster to override metamagic of the spell with ID of X, example `463_CASTER_LEVEL_OVERRIDE int 7` = extended, empowered and maximized Bigby's Crushing Hand
- new spell overriding variable, use int `SPELL_METAMAGIC_MODIFIER` on caster to add metamagic on top of already existing metamagic (if any), example `SPELL_METAMAGIC_MODIFIER int 2` = every spell cast will be also extended to any of the used metamagic. This is only useful for spells, not special abilities and not items as those can't be already enhanced by metamagic so override is the only variable that's needed there.

- added new field into spell engine structure, `spell.Range`, this is used to replace radius size range in spells with area of effect

- added new field into spell engine structure, `spell.DurationType`, this is used to replace `RoundsToSeconds`, `TurnsToSeconds` and `HoursToSecond` calls with new function `DurationToSeconds`

- added new field into spell engine structure, `spell.DamageCap`, this is used to replace a damage cap in various damaging spells, spells affected: blade barrier, burning hands, call of lightning, chain lightning, cone of cold, delayed fireball, fireball, fire storm, flame strike, hammer of gods, horrid wilting, lightning bolt, searing light, sunbeam, vampire touch, bombardment, earthquake, firebrand, sunburst, mestil's acid breath, crumble, gedlees' electric loop, glyph of warding, horzikaul's boom, ice dagger, scintillating sphere, undeath to death

- added new field into spell engine structure, spell.Limit, this is used to replace cap/limit of the various spells, this can be number of missiles, or damage if it's not based on hitdice or max hitpoints affected etc, spells affected: epic hellball, epic greater ruin, power word: kill, isaac lesser missile storm, isaac greater missile storm, tide of battle

- new spell overriding variable, use int SPELL\_DAMAGE\_CAP\_OVERRIDE on caster to override spell damage cap (if any), example SPELL\_DAMAGE\_CAP\_OVERRIDE int 80 = every spell that has a damage cap will have this cap raised to the 80, thus for example Horrid Wilting 1d8 per level up to 80d8 damage.

- new spell overriding variable, use int SPELL\_DAMAGE\_CAP\_MODIFIER on caster to modify spell damage cap (if any), example SPELL\_DAMAGE\_CAP\_MODIFIER int 10 = every spell that has a damage cap will have this cap raised by 10, thus for example Horrid Wilting 1d8 per level up to 35d8 damage.

- new spell overriding variable, use int SPECIAL\_ABILITY\_DAMAGE\_CAP\_OVERRIDE on creature

- new spell overriding variable, use int SPECIAL\_ABILITY\_DAMAGE\_CAP\_MODIFIER on creature

- new spell overriding variable, use int ITEM\_DAMAGE\_CAP\_OVERRIDE on item

- new spell overriding variable, use int ITEM\_DAMAGE\_CAP\_MODIFIER on item

- new spell overriding variable, use float SPELL\_RANGE\_OVERRIDE on caster to override spell area of effect height, example SPELL\_RANGE\_OVERRIDE float 10.0 = sets all spells to have RADIUS\_SIZE\_COLOSSAL

- new spell overriding variable, use float SPELL\_RANGE\_MODIFIER on caster to modify spell area of effect height, example SPELL\_RANGE\_MODIFIER float 2.0 = all spells will have their area of effect heighten by 2.0

- new spell overriding variable, use float SPECIAL\_ABILITY\_RANGE\_OVERRIDE on caster

- new spell overriding variable, use float SPECIAL\_ABILITY\_RANGE\_MODIFIER on caster

- new spell overriding variable, use float ITEM\_RANGE\_OVERRIDE on item

- new spell overriding variable, use float ITEM\_RANGE\_MODIFIER on item

- new spell overriding variable, use int SPELL\_DURATION\_TYPE\_OVERRIDE on caster to override spell duration type, 4 possible values, 1 - seconds, 2 - rounds, 3 - turns, 4 - hours. Example: SPELL\_DURATION\_TYPE\_OVERRIDE int 2 = all spells will last rounds.

- new spell overriding variable, use int SPELL\_DURATION\_TYPE\_MODIFIER on caster to modify spell duration type. Example: SPELL\_DURATION\_TYPE\_MODIFIER int 1 = spells that lasted seconds will last rounds, spells that lasted rounds will last turns and so on.

- new spell overriding variable, use int SPECIAL\_ABILITY\_DURATION\_TYPE\_OVERRIDE on caster

- new spell overriding variable, use int SPECIAL\_ABILITY\_DURATION\_TYPE\_MODIFIER on caster

- new spell overriding variable, use int ITEM\_DURATION\_TYPE\_OVERRIDE on item

- new spell overriding variable, use int ITEM\_DURATION\_TYPE\_MODIFIER on item

- new spell overriding variable, use float SPELL\_DURATION\_OVERRIDE on caster to override spell exact duration, note this happens \*after\* extending the value you will use won't be doubled

anymore. Example: SPELL\_DURATION\_OVERRIDE float 18.0 = all spells will last only 3 rounds.

- new spell overriding variable, use float SPELL\_DURATION\_MODIFIER on caster to modify spell exact duration, note this happens *\*after\** extending the value you will use won't be doubled anymore. Example: SPELL\_DURATION\_MODIFIER float 6.0 = all spells will last one round more.

- new spell overriding variable, use float SPECIAL\_ABILITY\_DURATION\_OVERRIDE on caster

- new spell overriding variable, use float SPECIAL\_ABILITY\_DURATION\_MODIFIER on caster

- new spell overriding variable, use float ITEM\_DURATION\_OVERRIDE on item

- new spell overriding variable, use float ITEM\_DURATION\_MODIFIER on item

- new spell overriding variable, use int X\_DAMAGE\_CAP\_OVERRIDE on item or caster to override damage cap of the spell with ID of X, example 58\_CASTER\_LEVEL\_OVERRIDE int 40 = fireball doing 1k6/level up to 40k6

- new spell overriding variable, use float X\_RANGE\_OVERRIDE on item or caster to override area of effect height of the spell with ID of X, example 114\_RANGE\_OVERRIDE float 20.0 = Mass Heal will have height of 20.0 (twice as much as RADIUS\_SIZE\_COLOSSAL)

- new spell overriding variable, use int X\_DURATION\_TYPE\_OVERRIDE on item or caster to override duration type of the spell with ID of X, 4 possible values, 1 - seconds, 2 - rounds, 3 - turns, 4 - hours. Example: 88\_DURATION\_TYPE\_OVERRIDE int 2 = Improved Invisibility will last level/rounds instead of level/turns.

- new spell overriding variable, use float X\_DURATION\_OVERRIDE on item or caster to override exact duration of the spell with ID of X, note this happens *\*after\** extending the value you will use won't be doubled anymore. Example: 443\_DURATION\_OVERRIDE float 24.0 = Greater Sanctuary will last 4 rounds only.

- new spell overriding variable, use int X\_LIMIT\_OVERRIDE Y on item or caster to override various limit of the spell with ID of X, where X is an ID of the spell (448 IGMS, 640 Greater Ruin etc.) and Y is an actual limit value. Example 448\_LIMIT\_OVERRIDE 50 = Isaac's Greater Missile Storm doing up to 50 missiles (if caster has 50 caster level). Due to the speciality of this variable there is no ITEM\_ / SPECIAL\_ABILITY\_ / SPELL\_ variant only ID version since this can't be really used globally as each spell has a different limit, so each spell must be set standalone.

- new spell overriding variable, use object SPELL\_TARGET\_OVERRIDE on caster to override an actual target of the spell. This is designed mostly for DMs in mind (for example, DM uses DM tool feat on NPC to mark her as a target of the spells, then whatever spell he casts on himself will be cast on this creature), can be used for other purposes eg. sharing spells with familiars. Either way this variable expects a scripting system via spellhook.

## **New polymorph “engine”**

This is a scripted solution that handles all ailments of the polymorphing in NWN and adds new functionalities and possibilities for builders as well. Since this relies on scripts, most of its features will work only in modules with unmodified polymorph spell scripts, unmodified



equip/unequip module events, unmodified polymorph.2da or in modules tailored for this by builders.

This also means it won't automatically work in most modules unless you use NWN(C)X. Fortunately, the polymorph issues are rather a nuisance in server mode and you won't notice most of them when playing single player modules.

If you are a builder and you want to get this all fixed in your module, open script 70\_inc\_shifter and read the installation instructions inside. It isn't necessary to do, and the game will work just fine, but some of the promised fixes/improvements might not work without these steps.

- disabled casting spells in polymorph that player normally doesn't have access to
- fixed polymorph temp hitpoints stacking and replenish (dependant on polymorph.2da change)
- fixed losing all merged item properties when "repolymorph" happens
- fixed losing bonus spell slots from ability bonuses (but only up to the ability bonus that was merged into shape)
- fixed losing spell slots from both ability bonuses and itemproperties, instead the spells in slots will be "consumed" (dependant on NWNX\_Patch)
- fixed dying after unpolymorph, player should not die from unpolymorphing ever
- merged ability decreases now ignores immunity to ability decrease from shape or from other merged items
- merged skill bonuses and saving throw bonuses are now stacked as well, if ability bonuses are
- if weapon is allowed to merge but the shape is unarmed, itemproperties will be merged onto skin instead (to support staves and other weapons with defensive itemproperties)
- additionally, the new polymorph engine supports custom content abilities that polymorph others (such as Baleful Polymorph spell). This has been improved with extended immunity check and a possibility to cancel such polymorph for a shapechanger/shifter creatures or player characters.

And on top of that you gain the ability to modify the way item merging works and also dynamically change polymorph shape ID, number of temporary hit points and whether is shape locked or not. All without need to edit or recompile polymorphing spells.

Also recap of the previous community patch polymorph features which still apply:

- allowed to merge any custom non-weapon in left hand slot such as flags or musical instruments
- added optional feature to stack ability bonuses from multiple items together
- added optional feature to merge bracers (when items are allowed to merge)
- added benefits of being incorporeal to the spectre shape (50% concealment and the ability to walk through other creatures)
- fixed server crash when dying after unpolymorph

## Scripting changes and improvements

Community patch also adds many scripting-related features such as new constants, functions or includes. Some of the default functions' descriptions have been updated to remove incorrect statements.

Changes and fixes into nwscript.nss - definition file for vanilla functions and constants:

- changed maximum value in internal comment for ItemPropertyDecreaseSkill, now states 50 instead of 10
- added new ITEM\_VISUAL\_NONE constant for ItemPropertyVisualEffect function
- added nwscript constant for hidden visual effect VFX\_COM\_GIB, this effect will explode player body leaves nothing visible not even corpse and also locks camera
- added nwscript constants for two unlocked visual effects VFX\_DUR\_QUESTION\_MARK and VFX\_DUR\_EXCLAMATION\_MARK

New custom functions and constants can be found in community patch scripting libraries:

- 70\_inc\_main - contains function to return community patch version and few generic custom functions useful for building module
- 70\_inc\_nwnx - include file for custom functions that requires nwn(c)x\_patch
- 70\_inc\_spells - contains spell-engine and spell related functions
- 70\_inc\_switches - contains new module switches and creature variable constants
- 70\_inc\_itemprops - contains functions to work with itemproperties, new constants and functions to declare itemproperties from community patch such as boomerang
- 70\_inc\_dragons - contains few functions used in dragon abilities spell scripts
- 70\_inc\_ai - contains AI related custom functions
- 70\_inc\_shifter - "private" library for polymorph engine scripting system
- 70\_inc\_poison - "private" library for poison engine scripting system
- 70\_inc\_disease - "private" library for disease engine scripting system
- 70\_inc\_spellhook - "private" library for internal spellhook script/function

## Tileset fixes:

1.72 contains a huge number of tile fixes. Particularly, Beholder caves tiles were heavily cleaned and corrected, Rural/Castle (TNO) road in trees heavily corrected ([image](#)) and there were multiple vision issues in \*.set files fixed and other improvements. And also around hundred of singular tile fixes.

## Full changelog:

tdc01.set, tde01.set, tdr01.set, tsw01.set, tdt01.set, tdm01.set, tdc01.set - Big Door tiles visibility changed to fix the issue with revealing what's behind doors without opening them

**City Exterior:**

[tcn01\\_h02\\_02](#) - fixed one part of the wall that didn't fade properly

[tcn01\\_e04\\_02](#) - fixed polygon gaps in black coverage with tilefading enabled

[tcn01\\_z06\\_01](#) - whole tile was mispositioned by 1 pixel in both X and Y directions

**Crypt:**

[tdc01\\_a04\\_01](#) - fixed polygon gaps in black coverage with tilefading enabled

[tdc01\\_a13\\_01](#) - fixed polygon gaps in black coverage with tilefading enabled

[tdc01\\_f09\\_01](#) - fixed polygon gaps in black coverage with tilefading enabled

[tdc01\\_f11\\_01](#) - fixed polygon gaps in black coverage with tilefading enabled

[tdc01\\_f14\\_01](#) - fixed polygon gaps in black coverage with tilefading enabled

[tdc01\\_f16\\_01](#) - fixed polygon gaps in black coverage with tilefading enabled

[tdc01\\_f18\\_01](#) - fixed polygon gaps in black coverage with tilefading enabled

[tdc01\\_f19\\_01](#) - fixed polygon gaps in black coverage with tilefading enabled

[tdc01\\_f20\\_01](#) - fixed polygon gaps in black coverage with tilefading enabled

[tdc01\\_g09\\_01](#) - critical walkmesh fix (could trap characters close to wall)

[tdc01\\_g13\\_01](#) - fixed polygon gaps in black coverage with tilefading enabled

[tdc01\\_g14\\_01](#) - fixed polygon gaps in black coverage with tilefading enabled

[tdc01\\_g15\\_01](#) - fixed polygon gaps in black coverage with tilefading enabled

[tdc01\\_g17\\_01](#) - fixed polygon gaps in black coverage with tilefading enabled

[tdc01\\_g17\\_02](#) - fixed polygon gaps in black coverage with tilefading enabled

[tdc01\\_p13\\_01](#) - fixed polygon gaps + minor other fixes

**Dungeon:**

[tde01\\_a01\\_02](#) - fixed flickering of the lava behind the bars

[tde01\\_b03\\_01](#) - fixed polygon gaps in black coverage with tilefading enabled

[tde01\\_c13\\_01](#) - one mesh didn't faded properly

[tde01\\_o16\\_01](#) - pathnode changed to fix pathfinding issues

**Mines and caverns:**

[tdm01\\_c02\\_02](#) - fixed polygon gaps in the wall under water

[tdm01\\_d11\\_01](#) - fixed polygon gap in wall

[tdm01\\_d12\\_01](#) - fixed polygon gap in wall

[tdm01\\_h17\\_01](#) - fixed polygon gap in the bottom edge of one pillar

[tdm01\\_h18\\_01](#) - fixed polygon gaps in the bottom side of two pillars

[tdm01\\_o13\\_01](#) - fixed polygon gaps in stalagmites

[tdm01\\_s06\\_01](#) - fixed polygon gap

[tdm01\\_z01\\_01](#) - shiny water fixed

[tdm01\\_z05\\_01](#) - fixed polygon gap (*Zwerkules*)

[tdm01\\_z09\\_01](#) - small polygon gap fixed

[tdm01\\_z11\\_01](#) - small polygon gap fixed

**Ruins:**

[tdr01.set](#) - numerous visibility fixes in the interior group tiles

**Sewers:**

[tds01\\_edge.2da](#) - fixed missing door edge tile

[tds01\\_o13\\_01](#) - fixed path node rotation

[tds01\\_d06\\_01](#) - fixed visibility node, rotated by 90° (*allowed to see hidden rooms*)

#### **Sea caves:**

[tdt01\\_a02\\_03](#) - fixed polygon gap in between rock and wall

[tdt01\\_j02\\_02](#) - "up in the air" walkmesh fix

[tdt01\\_j11\\_01](#) - fixed polygon gaps in the wall

#### **Beholder caves:**

tib01\_a01\_01, a01\_02, a02\_01, a02\_02, a02\_03, a02\_04, a04\_01, a05\_01, a07\_01, a08\_01, a09\_01, a10\_01 - fixed many gaps and overlaps, fixed animloops

[tib01\\_a03\\_01](#) - emitter position corrected, fixed many gaps and overlaps

tib01\_a04\_01 - fixed gaps in the black tilefading coverage

tib01\_a08\_01 - fixed gaps in the black tilefading coverage

#### **Castle interior:**

[tic01\\_j15\\_01](#) - fixed small polygon gap in floor around wall pillar

#### **Drow interior:**

tid01\_a02\_01 - corrected few gaps between polygons

tid01\_b09\_01 - corrected few gaps between polygons

[tid01\\_b10\\_01](#) - fixed polygon gaps in floor

tid01\_b11\_01 - fixed polygon gaps in floor

tid01\_b12\_01 - corrected few gaps between polygons

tid01\_b13\_01 - fixed walkmesh issue that prevented to move into the doorway

tid01\_b13\_01 - visibility issue fixed

tid01\_b13\_01 - corrected two gaps between polygons

tid01\_b14\_01 - pathnode correction

tid01\_b17\_01 - pathnode correction

tid01\_d03\_01 - pathode correction

tid01\_d15\_01 - fixed polygon gap in wall

tid01\_f03\_01 - pathode correction

tid01\_h01\_08 - added missing doors which also fixed the visibility issue

tid01\_h02\_01 - added missing doors which also fixed the visibility issue

tid01\_h05\_01 - added missing doors which also fixed the visibility issue

tid01\_h08\_01 - added missing doors which also fixed the visibility issue

tid01\_h01\_08 - corrected two gaps between polygons

[tid01\\_h03\\_01](#) - fixed black coverage disappearing in certain camera angles, corrected several gaps between polygons

tid01\_k02\_01 - corrected several gaps between polygons

tid01\_k07\_01 - corrected few gaps between polygons

#### **Caste Interior 2:**

tni02.set - numerous visibility fixes in the round rooms tiles

#### **Rural/Castle:**

tno01.set - -270 rotation changed to 90 to avoid potential issues with custom content tools

tno01\_a51\_01 - fixed polygon gap in tree fence with tilefading enabled

tno01\_a26\_01 - fixed shadow issues (*Symphony*)

[tno\\_d31\\_01](#), [tno\\_d33\\_01](#), [tno\\_d34\\_01](#), [tno\\_d35\\_01](#), [tno\\_d36\\_01](#), [tno\\_d37\\_01](#), [tno01\\_z\\_2120](#) - tiles heavily cleaned and corrected: fixed flickering of the ground textures, fixed shades under the trees that were messed up completely, road texture position united with NWN default (z=1), in tiles [tno01\\_d31\\_01](#) and [tno01\\_d32\\_01](#), there was a polygon of grass missing, in tile [tno01\\_d36\\_01](#) few tree trunks had a black texture instead of bark one, fixed the tree trunks coverage texture -> united to black (was either none or fuzzy bark in many cases), fixed overlapping trunk with three wall in tiles [tno01\\_d35\\_01](#), [d36\\_01](#) and [d37\\_01](#)  
[tno01\\_n01\\_05](#) - missing tile variant added

#### **Frozen wastes:**

[tti\\_d02\\_01](#) - black coverage polygon didn't faded properly

#### **Desert:**

[ttf01\\_p12\\_01](#) - corrected two gaps between polygons, added black coverage to block view from below to above

[ttf01\\_z11\\_01](#) - fixed missing cliff edge tile in the corner of the area

#### **Forest:**

[ttf01.set](#) - fixed visibility node (was rotated off by 90°) for tiles [a03\\_01](#), [a03\\_02](#), [a03\\_03](#)

[ttf01\\_a02\\_03](#) (*CP tile*) - fixed gaps in the black tilefading coverage

[ttf01\\_a03\\_02](#) - corrected a little gap between polygons

[ttf01\\_h07\\_01](#) - fixed polygon gap in tree trunk with tilefading enabled

[ttf01\\_j09\\_02](#) - fixed polygon gap in black coverage with tilefading enabled

[ttf01\\_p12\\_01](#) - corrected two gaps between polygons, added black coverage to block view from below to above

[ttf01\\_z11\\_01](#) - fixed missing cliff edge tile in the corner of the area

#### **Rural:**

[ttr01\\_j17\\_01](#) - fixed polygon gaps in black coverage with tilefading enabled

[ttr01\\_l03\\_01](#) - fixed walkmesh around stairs, fixed several gaps between various polygons, added additional black coverage to door passage so there is no gap visible

[ttr01\\_l06\\_01](#) - fixed gaps between the wall and roof, added missing vertical texture for stairs, added additional black coverage to door passage so there is no gap visible

[ttr01\\_u04\\_01](#) - added missing black coverage to the upper arc, also filled the polygon gaps from bottom side of the arc which could be seen under specific camera angles

[ttr01\\_v04\\_01](#) - added missing black coverage to the upper arc, also filled the polygon gaps from bottom side of the arc which could be seen under specific camera angles

[ttr01\\_z17\\_01](#) - fixed polygon gap (*Zwerkules*)

#### **Rural winter:**

[tts01\\_d07\\_01](#) - tilefading corrected

[tts01\\_d09\\_01](#) - tilefading corrected

[tts01\\_h04\\_01](#) - removed flickering white line at the road crossing

[tts01\\_j17\\_01](#) - fixed gaps in the black tilefading coverage

[tts01\\_l03\\_01](#) - added additional black coverage to door passage so there is no gap visible

[tts01\\_l06\\_01](#) - fixed gaps between the wall and roof, added missing vertical texture for stairs, added additional black coverage to door passage so there is no gap visible

tts01\_s02\_01 - fixed polygon gap between stairs and floor, added black coverage to block view from below to above

tts01\_t01\_01 - vertical black coverage didn't faded properly

[tts01\\_z14\\_01](#) - fixed gaps in the black tilefading coverage, black coverage extended to cover edge too

**Underdark:**

ttu01\_f15\_01 - fixed polygon gap in black coverage with tilefading enabled

[ttu01\\_g08\\_01](#) - fixed polygon gaps in the wall on sides