NWN(C)X Patch plugin documentation

Community Patch also offers a NWNX plugins, called nwnx_patch (server-edition) and nwncx_patch (client-edition) that adds additional fixes and features. These plugins are dependant on the NWNX and NWNCX packages and won't work without it. Both plugins works standalone and doesn't require installing the Community Patch itself. Since the NWNX behaves differently on various operational systems, there is both windows and unix version of the nwnx_patch plugin. The client version works only under windows and supporting unix client isn't in plan. It's recommended to use WINE for playing NWN on unix OSes.
Builders who want to use NWNX Patch plugin when hosting module with nwserver.exe needs the NWNX package.
Players who want to use NWNCX Patch plugin needs the NWNCX package. The client plugin works primarily only in single player and also in a LAN games, but certain features works even in multiplayer.

NWN(C)X Patch for Community Patch 1.71:

Functionalities:

- Enables to choose custom base class in initial character creation. [**C**,**M**]
- Increases the effect icons limit from 130 to 255. [**C**,**M**]
- Removes the limitation on weapon color from CopyItemAndModify function. [**C**,**S**]
- Fixes trident weapon focus bug. [**C**,**S**]
- Penalties to abilities from equipped items now properly ignores immunity to ability decrease. [**C**,**S**]
- Curse effect ignores immunity to ability decrease on a DnD Rules/Very High difficulty.[**C**,**S**] (note: this feature is not automatic anymore in 1.72 and needs to be enabled with module switch)
- Circle kick has been changed to swap the target back after the extra attack on a different one. [**C**,**S**] (note: this feature works for all creatures but 1.72 version changed it and allows this only to players)

Legend: **C** = client version feature, **S** = server version feature, **M** = works also in a multiplayer

All these features are automatically enabled and cannot be turned off. In windows version of this plugin at least, linux version works a bit differently and offers this functionality. Source and readme for linux version can be found here.

NWN(C)X Patch for Community Patch 1.72

This version is available only for windows at this moment.

It's possible to use nwn(c)x_patch externally without Community Patch 1.72 installed, however note, that some of the features like death attack paralysis duration are coded in the softcoded scripts and without it they won't be functional. The same apply for anything controlled by 2da. If you want those features you need to install 1.72 or grab those scripts and 2da files from it and put them into your module/haks.

Plugin compatibility:

Plugin was tested in all windows versions and should work fine under 64bit versions as well.

As for compatibility with other plugins, there seems to be no problems either. I was testing it with nwnx_cool, nwnx_events, nwnx_fixes, nwnx_funcs without problem. In case you would have problem visit the Plugin settings section. There is a way to disable certain features - you will want either weapon feats from nwnx_patch or nwnx_cool anyway because in this case one will overwrite the other.
Plugins now also contains the nwnx_connect functionality. It will work together but you should not use nwnx_connect together with nwnx_patch or you need to disable this feature with ini settings, otherwise it will double the client lag when connecting to the server.

Plugin functionalities:

Here comes list of plugin functionalities. Some of more complicated will be explained in detail lower in this readme.

Note: if not stated otherwise, all features are available both for client and server without any differences.

Engine bug fixes:

- Fixed client crash with hellball visual effect.
- Fixed crash exploit with special attacks.
- Fixed crash exploit with cursed (undroppable) items.
- Fixed server crash when player with more than 255 memorised spell slots logs in.
- Fixed server crash when player gains or loses 255 spell slots at the same time.
- Disabled the double equip or so called throwing weapons exploit.
- Fixed the action cancel issues when joining/removing party and possibly also when changing PvP status.

Game-mechanic bug fixes:

- Fixed bug with deleveling RDD that didn't removed bonus abilities properly.
- Fixed bug with Defensive Stance mode getting cancelled when casting a spell/using potion and many other cases.
- Fixed the instant healer's kit usage bug *(when healer's kit was used on a healthy target, next usage was instant)*.
- Similar issue with instant taunt and pickpocket has been fixed as well.
- Fixed Holy avenger enhancement bonus bug *(when holy avenger was added on a weapon with enhancement bonus greater than 5, after removal the greater enhancement bonus got removed instead of the +5 from HA).*
- Fixed bug when Improved Sneak Attack didn't work for assassin's without at least 1d6 rogue or BG sneak attack.
- Assassin's death attack paralysis is now triggered only once per surprise *(was originally triggered for each attack in flurry)*.
- Assassin's death attack paralysis duration is now round/assassin level instead of (character level*1d6)rounds. *(note the duration is now also subject to the duration scaling)*
- Fixed bug in Deflect Arrow ability that determined the defender's weapon relative to the attacker creature size *(thus when there was kobold shooting, if defender had a scimitar in hand this ability didn't worked before etc.).*
- Fixed Ki Critical offhand bug *(for weapon in offhand, the weapon of choice was incorrectly retrieved from main hand weapon).*
- Fixed losing spell slots and spell uses after polymorph - instead, spells in the slots will be "consumed". Also, merged bonus spell slots won't be consumed at all *(note this is dependant on new scripting engine for polymorph from 1.72 which marks which item was merged and which wasn't, without it this feature won't be 100% functional.).*
- Fixed losing spell uses from spontaneous casters in several cases where this shouldn't happen (unequipping same item twice, losing charisma buff etc.)
- Holy Aura and Unholy Aura damage shield now properly affects only evil/good creatures.
- Fixed OnHitCastSpell itemproperty overriding the spell concentration check which led into casted spells being incorrectly uninterruptable (issue details).
- Complete rewrite of the effect caster level initialization:
    - fixed caster level of spell effects cast from an item (which was set to last normal spell caster level or -1 if none such spell was cast by the creature)
    - properly takes into account negative levels
    - takes into account Pale Master levels if the appropriate module switch is active
    - takes into account any custom prestige class levels if the conditions are met (ie. arcane class with prestige class with ArcSpellLvlMod or divine class with prestige class with DivSpellLvlMod)
    - takes into account community patch caster level override and modifier features
- Fixed combat informations update after level down.
- Fixed monk speed interaction with haste. (issue details)
- Fixed buying items from store with full inventory issue *(such item was destroyed from*

store if not set as infinite).

- Fixed bug in unarmed damage calculation where medium sized Monk11+/Shou Disciple5 had unarmed damage 2d20 instead of 1d20.
- Fixed devastating critical hit not triggering (greater) cleave properly.
- Fixed quivering palm not respecting immunity to critical hits from deathless mastery feat.
- Disabled levelling up in polymorph to prevent exploits.
- Attacks of Opportunity won't be triggered to target casting a spell, performing taunt, pickpocket, lock/trap actions and when holding a magical staff.

Various improvements and new features:

- Allowed to put Counterspell ability into quickslot. [Singleplayer feature only, works also in multiplayer]
- Hak and tlk preloading when connecting into multiplayer server to allow create new character with custom class/feat/spell etc. This is the same functionality as in nwnx_connect, so you should not be using nwnx_connect together with nwnx_patch.
- Spellcasting softcoded and externalized into 2das, it is now possible to create custom spellcasters or modify vanilla ones.
- Prestige class spellcasting progression support for spontaneous casters which means that Pale Master (or custom prestige spell enhancing class) levels increases spell level for sorcerer/bard properly. Note, this doesn't bring the spell selection at pale master levels, but next time player will level up as bard or sorcerer, he will be able to select spells from levels gained by PM (or custom prestige spell enhancing class).
- Allowed to select custom race/subrace when creating new character.
- Allowed possession of the animal companions.
- Disabled automatic canceling of combat modes when moving, they will stay on until player manually disables them.
- Enabled spontaneous casting for other classes than cleric.
- Enabled unlimited casting.
- Enabled to create fully functional custom ranged weapons.
- Added new feature to modify/add weapon feats. This is done by the new custom 2da called weaponfeats.2da, the structure is self-exemplary. This allows to add weapon feats for custom content weapons or even make a ranged weapon to be weapon of choice for Weapon Master.
- Added new feature to modify whether is weapon finessable or not. To do this, set an integer variable "finesse" with value 1 (allowed) or 0 (disallowed). Note: this does not take into consideration creature/weapon size (intent)!
- Added new feature to modify whether is weapon monk weapon or not. To do this, set an integer variable "ubab" with value 1 (allowed) or 0 (disallowed).
- Enabled using items in polymorph, this is controlled with a new column UseItems in polymorph.2da.
- Added possibility to cast spells in polymorph, use function NWNXPatch_SetCanCastInPolymorph.
- Added a new feature to create a favored enemy feat for custom races. This is done by

the new column FavoredEnemyFeat in racialtypes.2da.

- Added a new feature to modify favored enemy bonus. This is done by the integer variable “FE_MODIFIER” on a creature (can be both positive and negative).
- The smite evil and smite good damage calculation made custom content compatible: it will automatically stack any custom class levels with paladin/blackguard if such class have smite evil/good in its feat list.
- Added new feature to modify default item's Item Level Restriction value. To override the required level use an integer variable "CILR" int new_value.
- Enabled and implemented Boomerang itemproperty. Throwing weapon with this property will return to hands of wielder = never decreases quantity.
- Enabled detaching sneak attack immunity from the immunity to critical hits. This feature must be enabled via module switch (see x2_inc_switches).
- Enabled detaching immunity to paralysis from the immunity to mind spells. This feature must be enabled via module switch (see x2_inc_switches).
- Allowed to disable monk abilities in polymorph. This feature must be enabled via module switch (see x2_inc_switches).
- Allowed to override or even disable tumble AC bonus. This bonus can be either completely disabled via module switch (see x2_inc_switches) or the actual value can be overridden by an integer variable "TumbleAC" on the fly.
- Allowed to override creature spell resistance. Use an integer variable “GetSpellResistance” to override the default SR value.
- Allowed to modify and override saving throws. You can use following integer variables to change specific saving throws. Note, this apply for base saving throw values, that is before engine counts effect bonuses in.
    - “ReflexSaveOverride” - overrides base reflex saving throws
    - “FortitudeSaveOverride” - overrides base reflex saving throws
    - “WillSaveOverride” - overrides base will saving throws
    - “ReflexSaveModifier” - adds given value into base reflex saving throws
    - “FortitudeSaveModifier” - adds given value into base reflex saving throws
    - “WillSaveModifier” - adds given value into base will saving throws
- Enabled to set next attack of opportunity to be sneak (or death) attack.
- Enabled to modify number of attacks of opportunity per round. Use an integer variable “NUM_AOO” on creature to allow her perform more than 1 AoO in same round.
- Enabled to modify number of main hand attacks per round. Use an integer variable “NUM_ONHAND” on creature to modify number of attacks (limit unknown).
- Enabled to modify number of off-hand attacks per round. Use an integer variable “NUM_OFFHAND” on creature to modify number of attacks (limit unknown).
- Enabled to modify number of extra attacks per round. Use an integer variable “NUM_EXTRA” on creature to modify number of attacks (limit unknown).
- Enabled to increase number of epic dodges per round. Use an integer variable “NUM_ED” on creature to set how many times she can use epic dodge (this value is unlimited).
- Enabled to increase number of “deflect arrows” per round. Use an integer variable

“NUM_DEFLECT” on creature to set how many times she can use deflect arrow (this value is unlimited).

- Allowed to create a custom special attack. To do it, make an useable feat with no SPELLID. Then alter 70_s2_specattk and code your special attack from scratch.
- Allowed to override weapon’s critical threat. Use an integer variable “CriticalThreatOverride” to override default value.
- Allowed to override weapon's critical multiplier with custom values. Use an integer variable “CriticalMultiplierOverride” to override default value.
- Allowed to override immunity result. Using function SetImmunityOverride/RemoveImmunityOverride you can (temporarily) change whether if creature immune to specific immunity or not.
- Added possibility to permanently modify creature ability scores on the fly. Use function NWNXPatch_SetAbilityScore.
- Added possibility to modify natural AC, armor AC or shield AC of creatures. Use function NWNXPatch_SetBaseAC.
- Added possibility to turn on/off tile animloops, use function NWNXPatch_SetTileAnimLoop.
- Added possibility to play a movie for specific player.
- Added possibility to highlight specific object for specific player.
- Added possibility to completely disable object highlighting (aka TAB feature) from specific player.
- Added possibility to create and modify any engine effect such as EffectIcon, EffectTaunt etc. see function NWNXPatch_SetEffectTrueType.
- Added possibility to create custom effect using function NWNXPatch_SetEffectTrueType and new event 70_mod_effects
- Added possibility to loop (and possibly) remove internal effects, see function NWNXPatch_GetFirstEffect and NWNXPatch_GetNextEffect.
- Added custom effect NWNXPatch_EffectAttackIncreaseUncapped allowing to increase AB over +20cap.
- Added custom effect NWNXPatch_EffectModifyBAB allowing to modify an actual pre-epic BAB bonus to specific value *(useful to code PnP Divine Power etc.)*.
- Allowed to make creature immune to flanking. Use an integer variable “IMMUNITY_FLANKING ” = 1 to grant creature immunity to being flanked.
- And much more...

NWScript improvements:

- Improved the OnClientLeave module event: the GetExitingObject() will now return fully valid player object, ie. you can get location and any other info there.
- Improved function SetCustomToken to be able to set system tokens 0-9, very useful for sending player texts with <CUSTOM0-9>.
- Enabled to read GetLastDamager(), GetTotalDamageDealt() and GetDamageDealtByType() functions inside OnPhysicalAttacked event.
- Also enabled to read all the attack data informations in OnPhysicalAttacked event such

as whether was attack hit or miss, critical hit, sneak etc. use function NWNXPatch_GetLastAttackCombatData.

- ApplyEffectToObject function now allows to use DURATION_TYPE_EQUIPPED (3) and DURATION_TYPE_INNATE (4) constants. Applying effect with these duration types allows to apply it on dead targets and makes the effect completely permanent (not removed by death), completely undispellable and also invisible to the vanilla GetFirst/NextEffect functions. Use function NWNXPatch_GetFirst/NextEffect to access such effect.
- VersusAlignmentEffect and VersusRacialType functions enabled to work with EffectDamageShield *(previously these functions had no effect on this effect type)*.
- Increased limit in SetItemCharges function to 100.

New module events and existing module events improvements:

- Enabled OnPhysicallyAttacked event for players. Runs script “70_mod_attacked” if exists.
- Enabled OnDamaged event for players. Runs script “70_mod_damaged” if exists.
- Added new module event OnPlayerLevelDown which will run script “70_mod_leveldown”. Script runs twice, before the delevel happens and after.
- Added new module event which will run script “70_mod_effects” when the engine applies or removes any custom effect (truetype 96+). Allows to code what such custom effect does.
- Plugin will also run a “duplicate” event for each module event except heartbeat. Main purpose for these events is to enforce some of the scripted community patch features in every module without need to merge module event scripts. You can reuse this feature to similar goal - to make something to work in every module you (or those who you send the script to) will play. Also, the plugin allows to bypass running original event script, to do it, set local variable “BYPASS_EVENT” int 1 on module object. Clarification: while it is possible to cancel running original event script, it is just the script that is blocked, not the actual action that caused the event. So it is not possible to disallow player to unequip items etc.

OnAcquireItem	- 70_mod_def_aqu
OnActivateItem	- 70_mod_def_act
OnClientEnter	- 70_mod_def_enter
OnClientLeave	- 70_mod_def_leave
OnCutsceneAbort	- 70_mod_def_abort
OnHeartbeat	- not running extra script
OnModuleLoad	- 70_mod_def_load
OnPlayerChat	- 70_mod_def_chat
OnPlayerDeath	- 70_mod_def_death
OnPlayerDying	- 70_mod_def_dying
OnPlayerEquipItem	- 70_mod_def_equ
OnPlayerLevelUp	- 70_mod_def_lvup
OnPlayerRespawn	- 70_mod_def_resp

    - OnPlayerRest - 70_mod_def_rest
    - OnPlayerUnEquipItem - 70_mod_def_unequ
    - OnUnAcquireItem - 70_mod_def_unaqu
    - OnUserDefined - 70_mod_def_user

Some of the event scripts are not distributed within community patch or package with this plugin. That's because there was no need to use them within community patch and it wouldn't be efficient to provide an empty script doing nothing. If you want to use an event which uses script not provided, simply create new script and save it under specified name.

Softcoded feats, skills and abilities:

- Softcoded the healer's kits - builder is now able to edit and modify this item power entirely within NWScript, the new script name is "70_s3_healkit"
- Softcoded the devastating critical ability into "70_s2_devattk" script.
- Softcoded the animal companion summoning into "nw_s2_animalcom" script.
- Softcoded the familiar summoning into "nw_s2_familiar" script.
- Softcoded the Assassin death attack OnHit paralyse ability into "70_s2_dthattk" script.
- Softcoded the defensive stance feat into "70_s2_ddstance" script.
- Softcoded the attacks of opportunity into "70_s2_aoo" script.
- Softcoded the pickpocket skill into "70_s2_pickpocket" script.
- Softcoded the taunt skill into "70_s2_taunt" script.
- Softcoded learning spell scrolls into "70_s2_learnscroll" script.
- Softcoded all special attacks: knockdown, disarm, smite evil, smite good, sap, stunning fist, quivering palm, called shot, ki damage (partially). All these special attacks will fire script "70_s2_specattk". This script also fires for cleave, greater cleave, circle kick, whirlwind, improved whirlwind, attack of opportunity, parry riposte attack and also custom special attacks.

NWNX_Patch also adds new NWScript functions:

- NWNXPatch_ActionUseSpecialAttack
- NWNXPatch_AddKnownSpell
- NWNXPatch_BootPCWithMessage
- NWNXPatch_BroadcastAttackOfOpportunity
- NWNXPatch_DumpEffects
- NWNXPatch_EffectAttackIncreaseUncapped
- NWNXPatch_EffectModifyBAB
- NWNXPatch_ForceApplyEffectToObject
- NWNXPatch_GetAreaAmbientColor
- NWNXPatch_GetAreaDiffuseColor
- NWNXPatch_GetAreaChanceOfLightning
- NWNXPatch_GetAreaListenModifier
- NWNXPatch_GetAreaShadowsEnabled
- NWNXPatch_GetAreaSpotModifier

- NWNXPatch_GetAreaWind
- NWNXPatch_GetBaseAC
- NWNXPatch_GetBodyBag
- NWNXPatch_GetConversation
- NWNXPatch_GetDomain
- NWNXPatch_GetEffectInteger
- NWNXPatch_GetEffectRemainingDuration
- NWNXPatch_GetEffectTrueType
- NWNXPatch_GetEncounterFrom
- NWNXPatch_GetEncumbranceState
- NWNXPatch_GetEquippmentWeight
- NWNXPatch_GetFactionId
- NWNXPatch_GetFirstEffect
- NWNXPatch_GetFlanked
- NWNXPatch_GetIsFeatGranted
- NWNXPatch_GetIsFlatfooted
- NWNXPatch_GetKnownSpell
- NWNXPatch_GetKnowsSpell
- NWNXPatch_GetLastAttackCombatData
- NWNXPatch_GetMaxSpellSlots
- NWNXPatch_GetMemorisedSpellSlot
- NWNXPatch_GetMovementRateFactor
- NWNXPatch_GetNextEffect
- NWNXPatch_GetNoRestFlag
- NWNXPatch_GetNthArea
- NWNXPatch_GetNumAreas
- NWNXPatch_GetObjectById
- NWNXPatch_GetPCFileName
- NWNXPatch_GetPlayerLanguage
- NWNXPatch_GetPvPSettings
- NWNXPatch_GetSoundsetId
- NWNXPatch_GetSpellSchoolSpecialization
- NWNXPatch_GetStartingPackage
- NWNXPatch_GetSurfaceMaterial
- NWNXPatch_GetTileAnimLoop
- NWNXPatch_GetTotalDamageDealtByType
- NWNXPatch_GetWeaponCriticalMultiplier
- NWNXPatch_GetWeaponCriticalThreat
- NWNXPatch_HighlightObject
- NWNXPatch_CheckItemFitsInventory
- NWNXPatch_IsInUse
- NWNXPatch_JumpToLimbo
- NWNXPatch_PlayMovie

- NWNXPatch_PossessCreature
- NWNXPatch_RemoveKnownSpell
- NWNXPatch_SetAbilityScore
- NWNXPatch_SetAge
- NWNXPatch_SetAreaAmbientColor
- NWNXPatch_SetAreaDayTime
- NWNXPatch_SetAreaDiffuseColor
- NWNXPatch_SetAreaChanceOfLightning
- NWNXPatch_SetAreaListenModifier
- NWNXPatch_SetAreaShadowsEnabled
- NWNXPatch_SetAreaSpotModifier
- NWNXPatch_SetAreaWind
- NWNXPatch_SetBaseAC
- NWNXPatch_SetAttackCoupeDeGrace
- NWNXPatch_SetAttackCriticalThreatRoll
- NWNXPatch_SetAttackKillingBlow
- NWNXPatch_SetAttackRoll
- NWNXPatch_SetAttackSneak
- NWNXPatch_SetBaseItemType
- NWNXPatch_SetBaseSavingThrowFortitude
- NWNXPatch_SetBaseSavingThrowReflex
- NWNXPatch_SetBaseSavingThrowWill
- NWNXPatch_SetBodyBag
- NWNXPatch_SetCanCastInPolymorph
- NWNXPatch_SetConversation
- NWNXPatch_SetCreatureSize
- NWNXPatch_SetCurrentHitPoints
- NWNXPatch_SetDisableObjectHighlight
- NWNXPatch_SetDomain
- NWNXPatch_SetEffectCasterLevel
- NWNXPatch_SetEffectCreator
- NWNXPatch_SetEffectInteger
- NWNXPatch_SetEffectSpellId
- NWNXPatch_SetEffectTrueType
- NWNXPatch_SetFactionId
- NWNXPatch_SetGender
- NWNXPatch_SetGold
- NWNXPatch_SetMemorisedSpellSlot
- NWNXPatch_SetMovementRate
- NWNXPatch_SetMovementRateFactor
- NWNXPatch_SetNoRestFlag
- NWNXPatch_SetPvPSettings
- NWNXPatch_SetRacialType

- NWNXPatch_SetSoundsetId
- NWNXPatch_SetSpellSchoolSpecialization
- NWNXPatch_SetStartingPackage
- NWNXPatch_SetTag
- NWNXPatch_SetTileAnimLoop
- NWNXPatch_SetTotalDamageDealtByType
- NWNXPatch_SetTrapCreator
- NWNXPatch_SetTrapFlagged
- NWNXPatch_SetXP
- NWNXPatch_TakeItemFromCreature
- NWNXPatch_UnhighlightObject
- NWNXPatch_VerifyClientRunningNWNCX
- GetFavoredEnemyModifier
- GetNumberOfAOOs
- GetWeaponIsFinessable
- GetWeaponIsMonkWeapon
- RemoveImmunityOverride
- SetFavoredEnemyModifier
- SetImmunityOverride
- SetItemLevelRestriction
- SetNumberOfAOOs
- SetTumbleACBonus
- SetWeaponCriticalMultiplier
- SetWeaponCriticalThreat
- SetWeaponIsFinessable
- SetWeaponIsMonkWeapon

Details about these functions can be found in 70_inc_nwnx.nss include file.

Plugin settings:

NWNX_Patch differs from other plugins in one crucial point. It has both client and server version and both versions offer same functionalities. This concept revealed the flaws in the standard way of customizing plugins via NWNX.ini file. Client side version of NWNX doesn’t have this file at all and even if it had, customizing plugin features on client side part of the plugin doesn’t really fit there anyway. It should be the module builder, who should decide if he wants certain feature disabled in his module, not player. From this reason the several toggleable features are using module switches or variables on item/creature/object.

Available module switches:

- 72_HARDCORE_UNCANNY_DODGE int 1 will to enforce PnP Uncanny Dodge 2 feat behavior.
- 72_CURSE_IGNORE_ABILITY_DECREASE_IMMUNITY int 1 will enable the curse effect feature from 1.71 version of this plugin.

- 72_ENABLE_FLYING_TRAP_IMMUNITY int 1 will enable the flying creatures immunity to ground traps (won't trigger them). If needed, specific trap can be made to ignore this rule using variable “DISALLOW_FLYING” int 1.
- 72_DISABLE_MONK_IN_POLYMORPH int 1 will disable monk abilities in polymorph.
- 72_DISABLE_PARALYZE_MIND_SPELL_IMMUNITY int 1 will detach paralysis from mind spells immunity.
- 72_DISABLE_SNEAK_CRITICAL_IMMUNITY int 1 will detach sneak attack from critical hits immunity.
- 72_DISABLE_TUMBLE_AC int 1 will completely remove the AC bonus from tumble skill.

All these switches can be found in the 1.72 version of the x2_inc_switches include.

Additional nwnplayer.ini settings:
[Server Options]

- Disallow New Characters = 0/1 - if 1, server will not accept new characters (purpose is to enable premade characters only with usage of base server vault)
- Num Tiles Explored = 1-10 - will change how many tiles are revealed in front of character. Allowed values 1-10, default value is 8, recommended value for higher immersion is 4.
- Disallow Animal Companion Possessing = 0/1 - if 1, players will not be able to possess animal companions

These are all optional/toggleable features. The rest of the features are mainly bugfixes, externalizations or improvements of existing stuff without drawback and such there shouldn’t be a reasons to turn them off.

However for the testing purposes and the compatibility with other plugin there is possibility to disable other features via nwnplayer.ini.

To do this add this code into your nwnplayer.ini:

```
[Community Patch]
DebugLevel=0 ; 0: almost no debug messages (default value), 1: prints debug messages for using NWNXPatch_ functions, 2: prints debug messages for most hooks too, 3: prints debug messages for all hooks including those running extremely often
Disable Sticky Combat Modes=0
DisableSetLocalStringHook=0
DisableConnect=0
DisableLoadModuleFinishHook=0
DisableWeaponHooks=0
DisableUncannyDodge=0
DisableGetTotalACSkillModHook=0
DisableGetUseMonkAbilitiesHook=0
DisableAttackOfOpportunity=0
```

```
DisableResolveAttackHook=0
DisableAddEquipItemActionsHook=0
DisableSummonAnimalCompanionHook=0
DisableSummonFamiliarHook=0
DisablePossessing=0
DisableUseItemHook=0
DisableEffectImmunityHook=0
DisableCustomEffects=0
DisableResolveDamageShieldsHook=0
DisableGetRelativeWeaponSizeHook=0
DisableHolyAvengerHook=0
DisableToggleModeHook=0
DisableSRHook=0
DisableApplyEffectOnObject=0
DisableEffects=0
DisableGetTotalEffectBonus=0
DisableGetBaseAttackBonus=0
DisablePlayerLeave=0
DisableModuleEvents=0
DisableValidateServerCharacter=0
DisableEffectCasterLevel=0
DisableAdjustSpellUses=0
DisableComputeFeatBonuses=0
DisableReadItemsFromGff=0
DisableOnHitCastSpellHook=0
DisableSellItem=0
DisableLevelDownHook=0
DisableLevelUpHook=0
DisableResolveAmmunitionHook=0
DisableSetActivityHook=0
DisableDefensiveStance=0
DisableStartBarterHook=0
DisableGetMinEquipLevelHook=0
DisableClearAllPartyInvalidActionsHook=0
DisableFireTrapHook=0
DisableFavoredEnemyHook=0
DisableAddPickPocketAction=0
DisableAddTauntActions=0
DisableLearnScroll=0
DisableSavingThrows=0
DisableStartCombatRound=0
DisableAttackOfOpportunity=0
DisableApplyOnHitCastSpell=0
```

```
DisableInitializeNumberOfAttacks=0
DisableSpecialAttacks=0
DisableCustomSpellcasters=0
DisableSpellSlotsInPolymorphHook=0
```

Change to 1 to disable specific feature.

Extra informations on some features:

Custom ranged weapons details:
NWN will now respect custom ranged weapons defined in baseitems.2da. Ranged weapon must have column “RangedWeapon” set to value referring to the baseitem of ammo or new weapon itself.

NWN now recognizes two types of ranged weapons. Weapons that will refer in baseitems.2da into different base item type will be considered ammo based ranged weapons while weapons that will refer to itself will be considered to be throwing weapons.

Ammo based ranged weapons will respect itemproperty unlimited ammo while throwing weapons will respect boomerang itemproperty.

Besides this, every custom ranged weapon will fire on full attack progression (ie. if you make a copy of crossbow, such crossbow will not be limited to 3 attacks per round). Rapid Shot combat mode will work with every custom ranged weapon as well. Additionally, I hardcoded a special quirk of repeating crossbow on baseitem 30 and that’s disabling all attacks in flurry in which new round of ammunition is equipped.

Special note to custom ammunition: It is possible to create a custom bolt with for example max stack size of 5, ammunition that will be equipped into off-hand etc, but creating custom ammunition models remains hardcoded.

Using items in polymorph:
This feature is controlled by the new column UseItems in polymorph.2da, which has 3 possible values:

- -1: no items neither potions
- 0: default behavior (potions only)
- 1: can use any unequippable useable items

Community patch 1.72beta has polymorph.2da with already preset values for each polymorph so players/builders doesn't need to set anything. Basically every humanoid and monstrous humanoid shape with exception of Tenser's Transformation and Shapechange shapes can use all items. The rest can still use only potions.

Effect caster level initialization detailed info:
The effect caster level is much complicated than it seems to be. On this page I wrote what it does normally and how it behaves now. To explain some of the terms inside that document. Delayed effect is an effect declared in a function that is delayed by DelayCommand or AssignCommand from spellscript. This is for example a case of Bigby's Clenched Fist spell where the stun effect is declared this way.

Note: This will not change the caster level of spells! This feature only apply to effects' internal calculation which is not accessible in NWScript and that has basically only one usage and that's dispel check. The actual effect duration or strength is calculated in spellscript. Community patch already offers altering the caster level and allows adding PM levels into calculation.

Creating new character with custom race
Is it possible to receive custom playable race in character creation via the subrace field - player must input the string that matches custom race "Label" column in racialtypes.2da.

This allows player to play Wemic and Brownie in any module with CEP. Simply write Wemic or Brownie into subrace field and you will get correct ability modifiers, racial feats and character model.

Note for builders: only dynamic appearances are allowed, client crashes if you allow player to select race that doesn't have such appearance.

Tips for making your custom class compatible with existing feats
Patch plugin adds automatic support for custom classes. Any custom class which has smite evil feat in its feat list will be count in smite evil damage calculation, likewise with smite good and also sneak attack 1d6 and uncanny dodge II rule (if enabled).

However, you might not want your custom class to grant these feats, this is especially problem with creating custom "sneaky" class as the sneak attack 1d6 will not stack with rogue so you want to use a custom feat.

Solution is actually very simple. Plugin doesn't check which level you get these feats, so you can set your class to get smite evil/good/sneak attack 1d6 at level player cannot get such as 60.

Custom spellcasters readme

NWN(C)X_Patch plugin allows to create a true custom spellcasters. No workarounds, everything works within standard spellbook GUI. Everything is 2da controlled. All you have to do is to set correct values into several 2das. That is all.

Important: this whole feature requires players to run NWNCX+nwncx_patch to work correctly. If you are running server, you should make sure players will use NWNCX to join your server if you want to provide your players custom spellcasters.

base features:

- cleric-type of casting (automatically knows all spells of given level)
- wizard-type of casting (knows only spells from lvl up)
- sorcerer-type of casting (spontaneous)
- fully working level up/character creation for custom spellcasters
- prestige class spellcasting progression support for spontaneous casters
- fully controlled by 2DA

additional features:

- removed hardcoded restriction on paladin and ranger spellcasting - both classes can now cast from lvl 1 if builder changes cls_spgn_rang/pal.2da appropriately
- hardcoded intelligence, charisma and wisdom for spellcasting purposes changed to class primary ability
- possibility to create a spellcaster using strength, dexterity or constitution
- possibility to create any type of spellcaster with domains (note: you need to give such class cleric package or make new one, otherwise recommended button in domain select will give player twice Air domain)
- custom spellcasters can learn spells from scrolls (note: multiclass of two or three such classes is not supported atm - need to figure how to select which class to use)
- complete rewrite of the familiar and animal companion selection - will now dynamically appear at the level the class gets the summon familiar/animal companion feat **and regardless of casting type**. This works with vanilla classes and custom classes, if you change ranger to obtain animal at lvl 1, the selection will automatically appear, if you change summon familiar for summon animal companion for sorcerer, he get animal companion selection instead of familiar.
- softcoded familiar and animal companion summoning - already existing CPP feature - this is needed to calculate level or rather resref of the summoned creature as normally the level is calculated from sorc+wiz and druid+ranger levels
- fixed vanilla bug in spellbook where spontaneous caster at the 3rd class position had no spell icon for cantrips if such class had only cantrips (which was true for bard)
- fixed racial ability modifier not being properly added when determining some of the spell related informations
- possibility to modify number of spells learned on levelup for wizard-type classes

- possibility to make set specific spells to be cast spontaneously for non-spontaneous spellcasters (ie. feature of cleric cure spells)
- possibility to make unlimited spellcasters

missing features:

- not possible to enable both animal and familiar selection
- not possible to create class which would select spells only at certain levels, the spell selection either shows from lvl 1 and every other level too (however, setting 0 in cls_splr_* or cls_spgn_* should allow to do this in a way...)
- not possible to modify number of domains selected
- selecting domains and spell school only available at lvl 1
- selecting domains will prevent to pick up familiar/animal companion or learn spells
- selecting spell school will automatically pop up the spell selection window, even if that class doesn't learn spells

still problem:

- spell school can be selected for custom class, but it doesn't work at all **(TODO: verify if still true)**

in plan:

- enable class-restricted familiars, animal companions and domains
- 2da based ArcSpellLvlMod and DivSpellLvlMod allowing to specify exact levels where spellcasting will be enhanced

Features explanation and instructions

Spontaneous casting

This feature is controlled by already existing column SpontaneouslyCast in spells.2da, where value is now ID of class that can cast the spell spontaneously. 10 = Wizard, 3 = Druid, 2 = Cleric, etc. Value of 1 will for compatibility reasons works as Cleric, obviously. Bard and Sorcerer are not able to use this feature as they already cast spontaneously. It is not possible to select more than one class but if you would want to make a spontaneous spell for both cleric and druid for example you can always create a two versions of that spell, one that will be in druid spellbook and the other one that will be in cleric spellbook only. This works correctly only if player runs NWNCX + nwncx_patch.

Unlimited spellcasting

You can make an entire spell level unlimited by setting the respective spell level value in cls_spgn_*.2da to -1. Note, this works only for a class level you set it, so if you want to enable sorcerer to cast all cantrips unlimited from level 1 you need to set column SpellLevel0 to -1 on all lines in cls_spgn_sorc.2da.

2DA settings instructions

classes.2da

- custom spellcaster must have column SpellCaster equal to 1 (obviously)
- spellcasting possibilities are controlled by the new column CastType, the value is a bitlflag which has these flags:
  - 1 - Restrict Spellbook (Show Only Known/Show all)
  - 2 - Spontaneous/Prepared 1/0
  - 4 - unused
  - 8 - Select domains
  - 16 - Select spell school
  - 32 - unused
  - 64 - Can learn scrolls
  - 128 - Arcane/Divine
- don’t forget to fill SpellGainTable and also SpellKnownTable for spontaneous casters
- to change number learned spells for wizard-like spellcasters, setup new collumn SpellLearnTable with a new 2da that follows cls_splr_wiz format (this 2da is included in 1.72 builders resources)

Spells_level.2da (sample download)

this is special new 2da that controls which spell shows in custom spellcaster class spellbook
basically, if you add a custom spellcaster class, you need to add a new column in this 2da with name = row_id of the class
so in my case as I allowed to cast Assassin which ID is 30 I added new column "30" without quotes, then just add values for each spell you want this class to know, values are the same as for classes.2da columns

Notes:
Last update v 1.32

- wizard cantrips are checking intelligence score without racial bonus, which means that half-orc with intelligence score 8 will learn all cantrips at lvl 1, and if he get his intelligence raised to 10 (by RDD for example) he will be able to memorize and cast them
- same with dwarf sorcerers/bard
- number of spells to learn at 1st wizard lvl is 3 + int modifier, if positive
- client internal intelligence and wisdom positions are reversed compared to server
- max number of classes and domains is 254
- max number of spells and feats is 65 534
- radial menu has a bug where the class id has only one digit which is a problem with custom casters but I fixed it

* specialized spellcaster support for spontaneous type of spellcasting
* selecting spell school works with spontaneous spellcasters as well - they get one extra spell known and spell per day for each spell level