

xp MsgServer

V1.3.0 – 2025-10-12 by Septirage

| | |
|---|----------|
| Guide for Admin | 2 |
| Requirements | 2 |
| Installation of xp_MsgServer : | 2 |
| Configuration of xp_MsgServer: | 2 |
| xp_MsgServer.ini : | 2 |
| Connection system: | 3 |
| OnConnection Script : | 3 |
| MsgServer NWNScript functions: | 3 |
| Connection Flow Example | 4 |
| Anticheat system: | 5 |
| AntiCheat LvlUp Error Formats : | 5 |
| AntiCheat CreateCharacter Error Formats : | 7 |

Guide for Admin

Requirements

To use the xp_MsgServer plugin you will need :

- [Nwnx4](#)
- xp_bugfix

Installation of xp_MsgServer :

Put xp_MsgServer.dll and xp_MsgServer.ini in your nwnx4 folders.

Add the nwnx_MsgServer.nss file in your module (or import it with xp_MsgServer.erf), this file contains all the function you can use.

Configuration of xp_MsgServer:

xp_MsgServer.ini :

- **loglevel** — will set the log verbosity
- **UseConnectionSystem** — set to true to use the whole connection system ability
 - **OnConnectionScript** — script called at the first step of connection.
Allowing you to start the whole connection securing process.
- **UseAnticheatCreation** — Enable the AnticheatSystem for CharacterCreation
 - **ScriptOnCreationError** — Script called on Creation Error.
 - **StopCreationFirstViolation** — If activated, will call the Script (or stop) at first (each) violations
 - **MinScaleX / MinScaleZ** — the minimal scale value allowed
 - **MaxScaleX / MaxScaleZ** — the maximal scale value allowed
- **UseAnticheatLvlUp** — Enable the AnticheatSystem for CharacterLevelUp
 - **StopLvlUpFirstViolation** — If activated, will call the script (or stop) at first (each) violation
 - **RangerCombatStyleFeats** — The list of the Rangers CombatStyle feats
 - **GrantedCondForEveryFeats** — Must come with changes on levelup panel. Some modification can allow you to use the GrantedConditions (GrantedOnLvl & GrantedPrereq) for every feat. If you do those changes, set this option to 1 in order to ensure correct checks.
 - **ScriptOnLvlUpError** — The name of the script called if an error is detected during levelup.

Connection system:

The connection system will trigger a script just after the end of hak download, before any other possible actions, allowing custom authentication and validation logic. See the provided connection_demo example files (scripts and GUI) and comments in xp_MsgServer.ini file for practical implementation examples.

OnConnection Script :

This Script has 5 parameters:

int **iUniquePlayerID** : The UniquePlayerID for this connection. Will be needed for specific MsgServer functions
string **sPlayerName** : The incoming Player account name (gamespy account, see GetPCPlayerName).
string **sIP** : The Ip of incoming player (see GetPCIPAddress)
string **sCDKey** : The public CDKey of the incoming player (see GetPCPublicCDKey)
int **iPrivileges** : Current Player/DM/Admin privileges of the incoming player (see XPMsgServer_GetIsXXX).

Return value : XPMSGSRV_HEIMDALL_RET_* constant, depending of the result wanted :

- XPMSGSRV_HEIMDALL_RET_WAIT: Do nothing and wait for further actions before authenticating the player.
- XPMSGSRV_HEIMDALL_RET_ALLOW : Allow player to enter the server
- XPMSGSRV_HEIMDALL_RET_KICK: Kick the player out of the server (all future messages from this connection will be discarded)

Note: the constants return values names are XPMSGSRV_HEIMDALL_RET_ for historical reasons, but they are used by the connection system regardless of your implementation.*

MsgServer NWNScript functions:

All those functions can be used by include nwnx_msgserver. This file also gives documentation for each of them but we will try to give you a quick overview here.

First, you have some “general Management” function. That will allow you activate or deactivate some plugin functionalities:

- XPMsgServer_SetAntiCheatLvlUpSystem: Will allow you to activate or deactivate the Anticheat system for the level up process.
- XPMsgServer_SetAntiCheatCreationSystem: Will allow you to activate or deactivate the Anticheat system for Character creation process.
- XPMsgServer_SetConnectionSystem: Will allow you to activate or deactivate the Connection system.

Another important function is **XPMsgServer_SetAuthorizedGUIScript**. This one will allow a script to be called by a non-authenticated player from a **UIObject_Misc_ExecuteServerScript** GUI callback.

This allowed script will be executed with 5 pre-inserted parameters, provided by MsgServer to help you.

Those parameters are the same as those described for OnConnection script, after those parameters, you can retrieve everything added by the UIObject_Misc_ExecuteServerScript call.

The script is called with OBJECT_SELF = Module.

The return value behavior is the same as in OnConnection script too.

Then, there are several GUI functions.

The vast majority will allow you to manage GUI for a player still not loaded into the module and so, without any related *object*.

All of those GUI functions will need the **iUniquePlayerID** instead of the classical **oPlayer** parameter so you will need to use them during the OnConnectionScript, Authorized scripts (or some related scripts).

Beyond this difference, you will recognize functions really close to the originals GUI ones.

Here is the list:

XPMsgServer_DisplayGuiScreen, XPMsgServer_CloseGUIScreen, XPMsgServer_SetGUIObjectHidden, XPMsgServer_SetGUIObjectDisable, XPMsgServer_SetGUIObjectText.

Finally, **XPMsgServer_SetAllowPlayer**, is for advanced usage and will be used to “Allow” a player. The main use of this function is to “Allow” a player if you are not directly in the OnConnection script or in an AuthorizedScript. (for example, after AssignCommand or DelayCommand call).

Connection Flow Example

Detailed Flow Example

Player connects



OnConnectionScript called

(receives: iUniquePlayerID, sPlayerName, sIP, sCDKey, iPrivileges)



| —→ Return **ALLOW**

| └→ Player can continue on the server normally



| —→ Return **KICK**

| └→ All further messages from this Player will be ignored



└→ **DisplayGUI** —→ Return **WAIT**

 └→ Player remains in pending state, custom processing continues



Player interacts with GUI



UIObject_Misc_ExecuteServerScript triggers **AuthorizedScript**

(receives same 5 parameters as OnConnectionScript + additional parameters)



| —→ **CloseGUI** —→ Return **ALLOW**

| └→ Player can continue on the server normally



| —→ **CloseGUI** —→ Return **KICK**

| └→ All further messages from this Player will be ignored



└→ Update GUI/etc —→ Return **WAIT**

 └→ Loop continues (go back to Player Interacts with UI)

Anticheat system:

The anticheat system validates character creation and level-up operations against game rules. When violations are detected, error strings are passed to the configured script for handling. Those scripts can choose to cancel or validate the Creation/Levelup. See the provided example scripts (sept_anticheat_creation.nss and sept_anticheat_lvup.nss) and xp_MsgServer.ini comments for configuration details.

AntiCheat LvlUp Error Formats :

Each error will be on the format “ #ErrorName:{ parameters }.

The text in bold represents values. (exemple **iClassID** represent the value of the chosen class ID).

List of errors :

#too_many_classes:{ **iClass0ID**, **iClass1ID**, **iClass2ID**, **iClass3ID**, **iClass4ID** } => the player has tried to get a fifth class.

#class_notvalid:{ **iClassID** } => the classID is not valid. This error is a final one. No other can be added after.

#class_notplayer:{ **iClassID** } => the chosen class is not a player class.

#class_level_too_high:{ **iClassID**, **iClassLvl**, **iMaxClassLvl** } => the player has tried to get **iClassLvl** for **iClassID** while the max lvl is **iMaxClassLvl**

#align_restrict:{Class: **iClassID**, GE: **iGoodEvil**, LC: **iLawChaos**} => the chosen **iClassID** is not allowed for the alignment

#class_prereq_error:{ **iClassID** } => the requirements are not met for **iClassID**.

#skill_restrict:{ **iSkill0ID** **iSkill1ID** } => one or more skill ID. Those skills are restricted and should not have been taken.

#skillrank_toohigh:{ **iSkill0ID**:**iNbSkill0Pts** **iSkill1ID**:**iNbSkill1Pts** ... } => one or more couple skill ID : skill points. Happens if the player has tried to put too many points in one (or several) skills

#toomany_skillPoint:{ **iSkillPtsAllowed** **iSkillPtsUsed** } => The player has used too many points

#rangercombatstyle_missing:{ } => The player has tried to level up a ranger (classID = 2) to level 7 and doesn't have any combat style feat.

#autofeatNotGranted:{ **iFeatOld** **iFeat1Id** } => one or more featID. Those feats should have been automatically granted and are not added.

#familiarType:{oldType:**iOldType** newType:**iNewType**} => the familiar type has changed

#CompanionType:{oldType:**iOldType** newType:**iNewType**} => the companion type has changed while neither feat 2002 nor 2168 have been taken.

#DomainChanged:{ } => One or more domains have changed

#DomainDontExist:{ **iDomainID** } => the DomainID doesn't exist

#tooManyFeats:{ **iNbNewFeats** : **iMaxNewFeatsAllowed** } => The player has taken too many feats.

#featSelectionError:{ } => Something is bad about feat selection (not right number of domainsFeat, bonus feat , ...)

#featPrereqNotValid:{ **iFeat0ID** **iFeat1ID** ... } => one or more FeatID. The feats don't meet the prerequisites.

#SchoolChanged:{} => *the Magic School has changed*

#SpellSchoolError:{ iSpellIID } => *The player has tried to take a spell forbidden by their magic school*

#SpellLevelError:{ iSpellIID:iLvLOfSpell:iLvUsed} => *The player has taken iSpellIID for iLvUsed while this spell is iLvLOfSpell*

#SpellError:{ iSpellIID } => *This spell doesn't exist or is removed*

#SpellClassProgressError:{} => *The player has tried to progress in a class they don't have*

#TooManySpellTakens:{} => *The player has taken too many new spells*

#SpellLvTooHigh:{} => *The player has taken a spell with a level above the current maximum level allowed.*

#TooManySpellTakens:{ iLv } => *The player has taken too many spells of iLv*

#BadSpellRemoved:{ iSpellIID } => *The player has tried to remove a spell they don't have*

#SpellRemovedTooHigh:{} => *The player has tried to remove a spell of a level above what is allowed*

#TooManySpellRemoved:{ iTOTALSpellRemoved } => *The player has tried to remove too many spells*

AntiCheat CreateCharacter Error Formats :

Each error will be on the format “ #ErrorName:{ parameters }.

The text in bold represents values. (exemple **iClassID** represent the value of the chosen class ID).

List of errors :

#BadRace:{ **iRaceRow**, **iSubRaceRow** } => *Something is not valid on the Race or Subrace*

#AbilityError:{ **iStr**, **iDex**, **iCon**, **iInt**, **iWis**, **iCha** } => *Some base abilities (before racial change) are below 8 or above 18*

#BadAmountAbilityPoints:{ **iStr**, **iDex**, **iCon**, **iInt**, **iWis**, **iCha**, **iAbilityPointsUsed** } => *The amount of ability point is != 32*

#AlignError:{ **iGoodEvil** , **iLawChaos** } => *Alignment doesn't match one of the default values.*

#ClassPrereqError:{ **iClassID** } => *Some of the pre-requirements of the chosen class are not met*

#SkillTooHigh:{ **iSkillOID** : **iNbSkillPoint** ... } => *one or more pair of skill ID – Nb of skill points. Too many skillpoints for those skillID*

#SkillNotAllowed:{ **iSkillOID** ... } => *one or more skill ID. Those skills are not allowed yet they have been taken.*

#TooManySkillPoint:{ **iSkillPtsUsed** : **iSkillPtsAllowed** } => *the player has tried to use too many skill points*

#RacialFeatNotFound:{ **iFeatID0** **iFeatID1** ... } => *one or more featID. Those racial feats should have been taken*

#AutoFeatNotFound:{ **iFeatID0** **iFeatID1** ... } => *one or more featID. Those auto-granted feats should have been here.*

#BackgroundReqError:{ **iBackgroundRow** } => *the selected Background doesn't meet the requirements*

#FeatNumber:{ **iFeatNb** } => *The player has tried to take more (or less) feats than expected.*

#FeatChoiceError:{} => *While the number of feat seems correct, the player has taken more regular/bonus or domains feats than legally expected.*

#DomainFeatNotFound:{ **iFeatOID** **iFeatIID** ... } => *one or more feat id. Those domains feats should be present.*

#TooManyBackgroundFeat{} => *The player selected more than 1 background feat.*

#AtLeastOneFeatNotValidated :{ **iFeatID** } => *At least this feat should not have been taken.*

#SpellSchoolError:{} => *The player has tried to take spell from a forbidden school*

#SpellLvlError:{} => *The player selected a spell of a lvIX for a lvIY slot*

#SpellUnknow:{} => *The player selected an unknown (or removed) spell*

#SpellNumberLvlError:{} => *The player didn't take the right amount of spell for a specific level*

#ScaleXandYDifferent:{ **fScaleX** , **fScaleY** } => *On default character creation, ScaleX and Y should be the same.*

#ScaleError:{ X: **fScaleX** ,Z: **fScaleZ** } => *The scales are out of bounds.*