Creating your own Prestige Classes for Fun and Nonprofit

by

Your Friendly Neighborhood Papermonk
of
The City of Doors Initiative
http://www.city-of-doors.com
irc: irc.neverwinterconnections.com #cityofdoors

So, you want to make your own Prestige Classes. Good for you! It's a fun and exciting hobby!

But, there are some things you should know first.

Caveats

1) If you're not familiar with doing custom content, you need to become familiar with it. A "Dummies Guide" is a bit beyond the scope of this, but almost anyone familiar with editing 2da files and the dialog.tlk and other such things will be okay here. What I'm getting at is that while I try to explain everything fairly simply, you do need to know the basics of how 2das, haks and the dialog.tlk file work to really get much out of it.

2) There are two ways to handle Prestige Classes. The Easy Way. And the Hard Way. This will cover the easy way, and the hard way to some extent. But to do it the "Hard Way" you will require the ability to script, and understand some of the minutia of the spells.2da and feats.2da.

3) Get this file: http://nwvault.ign.com/Files/other/data/1037585865411.shtml No excuses. I'm going to refer to it instead of just stealing all the words Eligio wrote. Anyone doing CC should have this, and anyone doing Prestige Classes (from here on out, I'll refer to them as *prcs*) really needs to print out some of it. And honestly, if someone writes me a question about this I will help them, but not if they've not looked here first. I'm a VERY busy Project Director, and I don't help those who won't help themselves. I simply don't have time.

4) If you're doing "MeteorMan PRC" that can cast Meteor Storm 1000 times a day, or something along these lines, don't ask me for help and if you do, pretend it's something else. I'm glad you can do it however you like, but I find some things just offensive and have no interest in taking any of my time to do something with it.

Starting the Job

First things first. Plan. You should get it all on paper first; figure it out just like you would design a prestige class for pen and paper.

I use this one when doing them for my pen and paper campaign, and it works fine for this.

Name:

Description:

HD:

Requirements

BAB:
Feats:

Skills:

Race:
Arcane Caster Level:

Class Skills

Skills Points at Each Level: __+ Int Modifier

Class Features

Level	Base Attack Bonus	Fort Save	Ref Save	Will Save	Special
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

I removed the “Spell Casting” area as, and this is important kids:
YOU CANNOT DO +1 TO SPELLCASTING LEVEL or give them a new spell table, etc. You can’t. End of story. Now, if you want to get creative in the scripting department and do something with items, etc… go for it. But as far as the prc construction goes, this isn’t an option. Deal with it.

What does this mean?

Well, this means that arcane caster prcs are pretty much out of luck. IF they want to go with a full force, up to 9th level spells, at least. But, unfortunately, it’s a bit worse than that. There is another thing to contend with. See, the prestige class isn’t gaining spell levels. And if it’s an arcane caster, it’s not gaining that power in exchange for the lack of wearing armor, etc. If they want to make use of their earlier spell casting levels, they really do have to keep to no or very light armor. This is a problem. Any class you make for straight casters, keep this in mind, and keep the power levels and what you are replacing in mind as well. To start with, I’d go up to at least a d6 hd. But these are design decisions. Our Daemothurge was originally an arcane caster with its own spelltable, but we modified it more towards the clerical end as the ability to cast won’t be impaired by heavy armor. We made a lot of their abilities spell like abilities, and I think it ended up pretty balanced.

PRCs are generally one or two things – they are either something very funky, a deviation from normal play (which really isn’t too possible with NWN) or an increase in specific abilities for a loss in general. The daemothurge is a good example – at first glance, it’s pretty potent. Then you realize almost all its power is directed at evil outsiders. They have a few spell abilities, but only a few of the 30-50 that are on their spell tables. So, we also gave them some enhanced abilities (fiendform and fiendish apotheosis) as well as a d6 hd. So, the character gets some minor marital protection, and some ability to really do some damage in fiend-form as well as some very potent resistances when they gain Fiendish Apotheosis.

Now, the question is, does all this equal the ability to cast all those spells?

Probably about even, really. You think about what an 18th level cleric or wizard can do, and then you think about it – it becomes apparent that’s its okay, maybe even underpowered.

Anyway, enough about design. Let’s get to the guts of this.

There are two ways of doing prcs, as mentioned earlier. The Easy Way, and the Hard Way.

The Easy Way is to take existing spells, feats, and special abilities and assign them to the prc. This means you just have to modify a few files, don’t have to do any scripting most the time and just in general – easier to do. The level of proficiency to do this is pretty simple. Hell, even I can do this sort fast and easily. I can’t script, so this is my option. Luckily, I work with a group with many talented scripters and can farm the work out.

The Hard Way is exponentially more complex. If you want to do new abilities, feats, spells, etc… you have to go the dirty route. This means scripting the new spells and such, then editing the spells.2da and feats.2da to accommodate these changes. Not too bad, but much harder for the novice than just picking and choosing some feats.

Okay, assuming you got past my diatribe, and planned it out…

So, you planned it out. Great!

This is real the work begins. First off, you need some programs. You need:

The newest version of *NWNExplorer* that can see the new bifs in the SoU expansion.

http://nwvault.ign.com/Files/other/data/1055545441750.shtml originally from Torlack, updated by DLA's Roboius

You need a *TLK Editor*
http://nwvault.ign.com/Files/other/data/1026158289672.shtml

and, optionally, *a 2da editor* (excel or notepad works fine) (for some VERY basic notes on how 2da's function, check this out: http://nwn.bioware.com/developers/Bioware_Aurora_2DA_Format.pdf)

Create a new directory. I tend to work from my desktop. Copy and paste the dialog.tlk from your NWN Directory there. You're going to need to edit this, so it's best if it's nice and convenient and not your original.

Open NWN Explorer. Go to File->Open Neverwinter Nights

Expand the *NWN SOU Data->Data\xp1.bif->Game Data*

Right click on Game Data and pick "export" then point it to your new directory. You don't need all these files, but it's easier for now. You can delete the ones you don't need later.

Now, what you need to do first is grab the classes.2da. Look at it, get used to it.. think about and try to figure out somewhat what each of those columns could do. You will remember this better the more you do on your own. You can open it in notepad.

You did that some? Good.

Now, grab these files: one of the cls_pres_xxxx.2da's, cls_skill_xxxx.2da's, cls_feat_xxxx.2da, maybe packages.2da, too.

Now, for the cls_xxxx you're going to be creating your own 2das. Best bet is to copy and paste a version of all these files in a new, subdirectory. Then rename them, and just change what you need in the files. For example... I tend to work in a system like this.

Desktop\Sou\Prcs\+ individual prc names as subdirectories+

Like I said, I'd spend some time just looking at these and trying to get a sense of what everything does. It's really not that hard if you have any experience with this.

But, let's go ahead and break them down.

PRC_PRES_XXXXX.2da			
ReqType	**What it is**	**ReqParam1**	**ReqParam2**
FEAT	Feat must be taken	# from feats.2da	****
FEATOR	This feat or the other Feat under **FEATOR**	# from feats.2da	****
RACE	Must be of this race, or others under **RACE**	# from racialtypes.2da	****
BAB	Base Attack Bonus	BAB #	****
VAR	Scripting Variable to turn on/off	script variable	0n/off (0 on)
ARCSPELL	If must be arcane spellcaster	Level of caster	****
SKILL	Must have this many ranks in this skill	# from skills.2da	ranks in skill

PRC_PRES_XXXXX.2da

Essentially, this file tells the game what the prequisites for the prc are. For example:

	LABEL	ReqType	ReqParam1	ReqParam2
0	Concentration	SKILL	1	5
1	Base_Attack	BAB	5	****
2	ScriptVar	VAR	X1_AllowFiendT	0
3	SpellFocusConj	FEAT	166	****
4	KnowledgeArcana	SKILL	28	5
5	KnowledgeReligion	SKILL	29	5

This is the prc_pres_fiendt.2da for the daemothurge. Let's explain what it all means. Take a look at that blue box, too.

Line

0 - This is a skill (ReqType: Skill) its position in the skills.2da is line 1 (ReqParam1) and you must have 5 ranks in it (ReqParam2)
1 - This is the base attack bonus, it's set at 5. For a fighter they would have this at level 5, a cleric or rogue at level 7, or a wizard at level 10.
2 - This tells is a script variable that someone could use to turn the prc off, or set up conditions when it would come on. In this case, it's called X1_AllowFiendT and 0 (ReqParam2) is saying "it's all good."
3 - This is a feat, it's telling it the character must have SpellFocus: Conjuration which is feat number 166 (ReqParam1) but doesn't need any more details (****) (ReqParam2)
4 - This is another skill. A custom one we're including, Knowledge: The Planes which would be skill number 28, and 5 ranks.
5 - Another skill. Skill #29, 5 ranks.

So, what's all this mean? Well, how you set up your prequisites is pretty important. It has everything to do with whom and when can take this prestige class. This is where knowing dnd mechanics pretty well helps.

Now, of course, NWN mechanics too. There are differences.

What this does is force a certain level. The Base Attack Bonus is first – this is a major way of handling it, as mentioned earlier – this directly relates to the level dependent on class. It's set at 5, which means at least level 7 for everyone but wizards and sorcerers.

Secondly, the skill Concentration. You must have 5 ranks. Well, if you are a class that has it as a class skill, you just have to have 2 levels to max it out. At any rate, you must use your precious skill points on it. Skill points are a "cost" of prestige classes, as are feats often times. Sometimes you have to take feats or skills that are otherwise unsavory to get to a prestige class. If it's a cross class skill, you'd need to be level 7 to max it out.

Spell Focus, Conjuration – now, this one is only available to classes that can cast – but that includes rangers and paladins...

Knowledge: Arcana – now, see, this is an example of "paying for it" most mods won't have any use for this, and those that do will only use it for dialogue checks. These skills aren't NEARLY as important for nwn as they would be in pen and paper. But, you're paying for it. So, you have to take it. Class Skill – level 2, cross class – level 7.

Knowledge: Religion – see “Knowledge: Arcana” above.

(I believe this can be used for normal classes as well... interesting things, one way or another)

CLS_SKILL_XXXXX.2da

Now, all classes need these. These tell the game what classes can get what, and what is a class skill and what is not. There are also “class only” skills like “use Magic Device” and “Perform” that you can’t even access unless it’s a class skill. (As a side note, restricted skills are being removed from DND 3.5)

I’m not going to post the entire file here, but what you need to know is this:

	Skill Label	Skill Index	Class Skill
0	Concentration	1	1

You have four columns; the first is just the line number, starting with 0. Second is the label, this is just to make it easier on readers... It could say “jello”, doesn’t matter. The *Skill Index* is actually what tells the game what skill it is, like the prc_pres_xxx file this points to its place in the skills.2da. Then *Class Skill* acts as an on/off switch for it being a class skill or not. 1 is on, 0 is off. Now, you only list here skills they have access to. Those “class only” skills are not listed, by any cross class skills otherwise are. A monk can never use “Use Magic Device” but could take ranks in some other cross class skill. If they can access these skills, you list them like any other. Otherwise, you leave them off the list entirely.

CLS_FEAT_XXXX.2da

This is actually rather refreshingly straight forward. Well, sorta.

See, there are four types of feats, really.
1) Passive
2) Activatable
3) Bonus
4) Blank Holder

Passive

These are feats that give benefits that aren’t activatable. Fire Resistance of 5, for instance. We can’t modify them, we can’t create new ones. If you want a class to get one of these as ability, you have to tell it. Also, some class abilities only work for those classes – when it’s one of these types of feats – just deal with it, you can’t do anything about it. Usually this is the case when the power is dependent on the number of levels in that class.

Activatable

These work the same for Bioware and us. This is a feat that works via the radial menu, something you activate. It, essentially, casts a spell. Now, not all “spells” are ‘spells’ but rather it fires a script attached to a spells.2da entry.

Bonus

We can’t currently implement these for custom prestige classes.

Blank Holder

This is our answer to passive feats. Basically, you set up a script to see if they have this feat and then have it do something. If we wanted to do something like fire resistance, this is how it’d be done and it’d show up as a spell effect. This has obvious drawbacks, but what can you do?

So, now we know what types of feats there are. How does this work with this file, and prcs.

Well, I'm going to tell you.

The file is constructed like this:

Line # FeatLabel FeatIndex List GrantedOnLevel OnMenu

FeatLabel	Name of the feat, taken from feats.2da
FeatIndex	Number of the feat, taken from feats.2da
List	0 = selectable, normally can take this.
	1 = uses regular or special feat slots (wizards/fighters)
	2=Selectable - uses SPECIAL slots only.
	3- not listed as selectable, must take it
GrantedOnLevel	What level the pc gains this feat, -1 means anytime
OnMenu	If this feat is "activatable" via radial menu

CLASSES.2da

This one is a real bugbear. It's big, it's complex, no good can come of it, etc....

But it's the big one. So, first, I'm going to list out a basic explanation of what does what from the "barebones" version of this tutorial. Bear with me, this is a LONG file.

One big thing here – and this is important. You must pad (put blank, ****'d out lines, between their lines and ours. At least 5)

Now, a lot of these are made pretty obvious in a few words. But, for completions sake – I'll talk about it here.

#	Label	Name	Plural	Lower
36	Name of PRC	Dialog.tlk Strref	Dialog.tlk	Dialog.tlk

Description	Icon	HitDie	AttackBonusTable
Dialog.tlk	icon file name	HitDie	name of bab 2da file

FeatsTable	SavingThrowTable	SkillsTable
CLS_FEAT_?	See Below	See Below

SkillPointBase	SpellGainTable	SpellKnownTable	PlayerClass	SpellCaster
Base # of skill pts	****	****	1	0

Str	Dex	Con	Wis	Int	Cha	PrimaryAbil
ideal stat	ideal stat	ideal stat	ideal stat	ideal stat	ideal stat	Primary Stat

AlignRestrict	AlignRstrctType	InvertRestrict	Constant	EffCRLvl01-20
See Below	See Below	See Below	CLASS_TYPE_(class name)	1-20

PreReqTable	MaxLevel	XPPenalty	Package
CLS_PRES_(class) - see 2da	Max Level of PRC	No XP Penalty for prcs	Packages.2da line #

# -	This one is not labeled "#" but this, like all 2da's, has a line number.
Label -	You can pretty much put what you like here unless Bioware opens up the game to read the next few entries instead of the dialog.tlk entries. No spaces.
Name -	This is a Strref, a dialog.tlk line number it's referring to. (e.g. 38230 (which could be Demondancer)
Plural -	Same as above. But plural. (e.g. 38231 (which could be Demondancers)
Lower -	Ditto, but in lowercase plural: demondancers
Description	This is a rather nice description of the prc in the dialog.tlk. For PRC's (and I think classes, too) this should discuss what the abilities are each level as well as a description of the class, its perquisites, etc.
Icon -	This is an icon graphic file for the class... like IR_CLERIC or somesuch... it's a small tga file. You can create your own, I imagine. Not sure. Also, don't know if you could use a spell icon instead.
HitDie -	Basic hit die (4, 6, 8, 10, 12, etc... per level)
AttackBonusTable	This is the name of the base attack bonus 2da file being used. See below for details.
FeatTable	This is what we've discussed before. Something along the lines of "CLS_FEAT_?"
SavingThrowTable	Pretty much like the Attack Bonus table, see below for details.
SkillsTable	See Below
SkillPointBase	Base number of skill points per level, 2, 4, 8, etc...
SpellGainTable	Neither of these are worth worrying about AT THIS TIME
SpellKnownTable	See above
PlayerClass	This needs to be 1, this tells the game it's a player class and not a monster race level
Spellcaster	Again, ignore it. Spellcasting classes cannot be made. Keep it at zero
Str, Dex, Con, Wis, etc..	These are the "ideal stats" you get by hitting recommended at character creation
PrimaryAbil	This is the most important stat for the class. Most classes have one. A rogues is Dex, a Wizard's Int, etc
AlignRestrict	These, along with the next two, are a bit tricky.. See below
AlignRestrictType	
InvertRestrict	
Constant	This is the scripting Constant you need to add, it should be something like CLASS_TYPE_(class name)
EFFCRLvl01 – 20	These are 20 columns. It's a weird way the game works, so some levels have a higher cr than others (which amounts to how much experience you gain per level) best to just go 1-20, unless the character is very weak at low levels or something – and even then, as a prc this shouldn't be a big issue.
PreReqTable	Works on normal classes, too! Basically, this refers to the CLS_PRES_(class).2da file
MaxLevel	Maximum level the prc can go to
XPPenalty	This tells the game not to xp penalty the prcs. So, this should be 0
Package	This refers to the packages.2da line for the class

Now, for all that “see below” business.

Attack Bonus Table

Attack Bonus Tables					
cls_atk_1.2da (Fighter)		cls_atk_2.2da (Rogue)		cls_atk_3.2da (Wizard)	
Level	Base Attack Bonus +	Level	Base Attack Bonus +	Level	Base Attack Bonus +
0	0	0	0	0	0
1	1	1	1	1	1
2	2	2	2	2	1
3	3	3	3	3	2
4	4	4	3	4	2
5	5	5	4	5	3
6	6	6	5	6	3
7	7	7	6	7	4
8	8	8	6	8	4
9	9	9	7	9	5
10	10	10	8	10	5
11	11	11	9	11	6
12	12	12	9	12	6
13	13	13	10	13	7
14	14	14	11	14	7
15	15	15	12	15	8
16	16	16	12	16	8
17	17	17	13	17	9
18	18	18	14	18	9
19	19	19	15	19	10
20	20	20	15	20	10
21	21	21	16	21	11
22	22	22	17	22	11
23	23	23	18	23	12
24	24	24	18	24	12
25	25	25	19	25	13
26	26	26	20	26	13
27	27	27	21	27	14
28	28	28	21	28	14
29	29	29	22	29	15
30	30	30	23	30	15
31	31	31	24	31	16
32	32	32	24	32	16
33	33	33	25	33	17
34	34	34	26	34	17
35	35	35	27	35	18
36	36	36	27	36	18
37	37	37	28	37	19
38	38	38	29	38	19
39	39	39	30	39	20
40	40	40	30	40	20

41	41	41	31	41	21
42	42	42	32	42	21
43	43	43	33	43	22
44	44	44	33	44	22
45	45	45	34	45	23
46	46	46	35	46	23
47	47	47	36	47	24
48	48	48	36	48	24
49	49	49	37	49	25
50	50	50	38	50	25
51	51	51	39	51	26
52	52	52	39	52	26
53	53	53	40	53	27
54	54	54	41	54	27
55	55	55	42	55	28
56	56	56	42	56	28
57	57	57	43	57	29
58	58	58	44	58	29
59	59	59	45	59	30

I'm only showing off three types, as these are the core choices. The others all look like one of these, so just point to one of them. You have fighters with the best attack bonuses – fighters, paladins, rangers and barbarians – then the next group is playing the middle: rogues, bards, monks, clerics and druids, and then finally – the weakest – wizards and sorcerers.

You can make new ones if you need to.

SavingThrowTable
This is like the base attack bonus table. There are a few main configurations.

High Fort	cls_savthr_fight
High Ref/Will	cls_savthr_bard
High Fort/Wis	cls_savthr_cler
High All	cls_savthr_monk
High Will	cls_savthr_sorc
High Ref	cls_savthr_rog

Best to just use these. You can make new ones if you need to.

SkillsTable
This has been discussed above with the Cls_skill_?.2da file. This goes there.

Alignment Restrictions
This is a real pain, involving hex code and other things as well. I'll turn it over to Shkuey who worked with GL to get this all figured out to explain it.

0x00 - No restrictions
0x01 - No Neutral
0x02 - No Lawful
0x03 - No Lawful, No Neutral
0x04 - No Chaotic
0x05 - No Chaotic, No Neutral
0x06 - No Chaotic, No Lawful

0x07 - No Chaotic, No Lawful, No Neutral
0x08 - No Good
0x09 - No Good, No Neutral
0x0A - No Good, No Lawful
0x0B - No Good, No Lawful, No Neutral
0x0C - No Good, No Chaotic
0x0D - No Good, No Chaotic, No Neutral
0x0E - No Good, No Chaotic, No Lawful
0x0F - No Good, No Chaotic, No Lawful, No Neutral
0x10 - No Evil
0x11 - No Evil, No Neutral
0x12 - No Evil, No Lawful
0x13 - No Evil, No Lawful, No Neutral
0x14 - No Evil, No Chaotic
0x15 - No Evil, No Chaotic, No Neutral
0x16 - No Evil, No Chaotic, No Lawful
0x17 - No Evil, No Chaotic, No Lawful, No Neutral
0x18 - No Evil, No Good
0x19 - No Evil, No Good, No Neutral
0x1A - No Evil, No Good, No Lawful
0x1B - No Evil, No Good, No Lawful, No Neutral
0x1C - No Evil, No Good, No Chaotic
0x1D - No Evil, No Good, No Chaotic, No Neutral
0x1E - No Evil, No Good, No Chaotic, No Lawful
0x1F - No Evil, No Good, No Chaotic, No Lawful, No Neutral

Align Restriction Type

0x0 - No Restriction (Can take class regardless of alignment and previous column)
0x1 - Restrict only on Law/Chaos spectrum (Can take class regardless of Good/Evil align, but must comply with Law/Chaos align)
0x2 - Restrict only on Good/Evil spectrum (Can take class regardless of Law/Chaos align, but must comply with Good/Evil align)
0x3 - Restrict on both Good/Evil and Law/Chaos (Must comply with all restrictions in previous column)

Couple of other things to point out. The Invert column does exactly that, any alignment that was allowed isn't, and any that wasn't is. And the No Neutral in the first sections refers to both sorts of neutrality, however, the restrict type can be used to force it to look at only one side of the alignment.

Some explanation for those who want to know:
The hex code represents a bit map, where each of the first 5 bits represent an alignment. 1 - Neutral, 2 - Lawful, 3 - Chaotic, 4 - Good, 5 - Evil, that's from right to left. If the bit is set (1), that alignment is disallowed if the restrict type allows it to be. The first bit, neutral, refers to both types of neutral.

PACKAGES.2da

Honesty, most the time, I'd not mess with it. But that's just me. It invites a ton more work, and I know most people who are downloading custom content and playing with it aren't using that "Recommend" button all that much.

Even explaining this file is more trouble than it's worth. In almost ALL cases, it's better just to point it to another line in the classes.2da file instead.

If you do want to play with it, by all means. It's not complex, but it's .... very detailed in some ways, and while not hard – it would take me a few more pages to adequately describe it and how to use it.

Special Abilities: Feats and Spells

A prc wouldn't be much of a prc without some interesting or new abilities.

This is where that whole, easy/hard thing comes in.

I'm not going to go into the hard part, really. It's a bit beyond this scope. For the hard, which involves NEW feats, spells and scripting – I'd refer to Eligio's guide for working with the feats.2da and spells.2da files, and you'd have to know how to script enough to get whatever you want to fire off via the spell. Then place the new feat in the cls_feat_*.2da and there you go. If it's activatable, it needs to have that pesky OnMenu bit to "On."

So, this is the thing... I'm not going to go into all that. If you can grab Eligio's guide and do most of it, you won't need me explaining it. So I won't.

Now, the easy bit.

Basically, go down the list of feats and pick which ones you want to use. But, keep in mind some things.

Some feats, mainly class ones, only work FOR that class. Like the blackguard's ability to Summon Fiends uses blackguard class levels –so things like that, don't be surprised if it doesn't work without some extra scripting.

But normal, "one you can take at level up" work fine, generally.

Currently, the whole "Bonus Feat" thing for wizards and fighters do not work for prestige classes.

Now, if you go this route – check out whatever number Bioware wants you to pad for these 2da's so everything is not messed up in the next expansion. Right now it's about 2000 for feats and 1500 for spells.

Summing it up

Okay, now that you know how it all works – it's pretty simple to do it.

Here's the big issues:

You must make dialog.tlk entires unless you can find suitable things already there or decide you don't need a description, etc. Dialog.tlk files CANNOT go into the haks. This is a problem when patches come out. Always direct people to back up their files before placing it in the NWN directory.

Everything else can go in a hak.

You will need to add a new constant for the class. Bioware suggests not putting the nwscript.nss in a hak file, so to create new CLASS_TYPE_ constants, use the const keyword in a central include file.

KNOWN ISSUES

The single biggest issue is the dependence on the dialog.tlk file. Now, it'd be helpful if the single word aspects of this could be handled in tlks that work like the appearance.2da, but the descriptions for spells, feats and the classes themselves remain an issue. The community at large is going to have to decide on how we want to handle these multiple tlk files being flung around. Sir PomPom has devised an interesting tool, and we hope to work with him to make it even easier. Still, this is something I'd like to see Bioware do something about.

Bug having more than one version of a character in your vault will generally cause the game to crash at character select, and other surrounding scenes.

Your basic option is to delete old versions when you save new ones.

Another option, though less happy, is to stick a copy of your modified classes.2da and prc_pre_?.2da set up so the classes are untakeable in your override folder.

Thanks

Nothing exists in a vacuum, and this doesn't either. I just played compiler and investigator with this, getting the basics of how these work down – a great deal of thanks go to Shkuey and Rabbithail (GL) for all their hard work into the very guts of it.

Thanks also go to CODI's industrious QA head RedRand Rabbithail again, who helped us nail down the problem with multiple characters causing bugs.

And of course, thanks go to Eligio for his incredible Custom Content guide which was a HUGE help during all of this.

Thanks in general go to the NWN Custom Content community, who continue to be an inspiration and I hope will make good use of this information. I hope to see legions of PRCS, I just hope we someday have a better option than tlk hacking.

And thanks also go out to the City of Doors Initiative itself, and all my team members there, and all the guys that just hang out in our chat room (irc.neverwinterconnections.com #cityofdoors) for the help they often render.