Simlogical and the Dead End Nights

3 users are in chat:.

Search

Home **Forum** Help Search Calendar Login Register

Simlogical and the Dead End Nights » Forum » The Sims 3 » TS3 Tutorials and Reference » Gameplay Tuts and Refs (YibToilet: Nona Mena) »
How to Create and Load Custom Moodlets with a Pure Script Mod

« previous next »

Pages: [**1**] 2 3

PRINT

This site gives you cookies (no, not cooties...)

FIND STUFF
Click here to look for downloads, tutorials or tools by our easy keyword search.

Author **Topic: How to Create and Load Custom Moodlets with a Pure Script Mod**

0 Members and 1 Guest are viewing this topic.

☐ **Nona Mena**
Moderator
🟩🟩🟩🟩

Posts: 768
Gender:

How to Create and Load Custom Moodlets with a Pure Script Mod
« **on:** 20 June 2012, 15:07 »

This tutorial assumes you know the basics of creating a pure script mod, and that you have at least some understanding of C#. All of the information you will find here is actually available elsewhere. I have simply put it all together in one post.

Pre-requisite Tutorials
If you have never made a script mod for the Sims 3, I highly recommend you take a look at the following tutorials before beginning with this one.

- Pure Scripting Modding Tutorial at MTS
- Creating a Game Compatible Visual Studio Project
- Localized Coding

To complete this tutorial, you will need:
Microsoft Visual C# Express 2010 (or 2012) - This is the free version. You can use the pay version, too, of course.
S3PE
Peter's Unprotect program - This is a small console application Peter compiled using twallan's code to make it easy to unprotect the core libraries.
Twallan's STBL program, available at his wiki (used to convert a text file to STBL, which I find easier when working with a lot of strings).
STBL Duplicator - You can use this tool to make strings for all languages.

Also highly recommend (free):
ILSpy - To decompile the core .dlls of the game
Notepad++ - Because it makes reading XML files easier.

There are several parts to this tutorial:
Part One: Creating the Code
Part Two: Creating Your Buffs XML
Part Three: Buff Elements and Localization
Part Four: Adding Moodlets and Buff Origins in the Script
Part Five: Finishing Up
Additional Credits
Bonus: Useful Overload Methods

Plan Ahead
To load a custom moodlet into the game, you need *all* of the following:
1. Custom script which parses an XML file to load the moodlet data, and contains a GUID for each moodlet.
2. An XML file containing moodlet data.
3. Buff information: Buff Name, Buff Description, Strength, Axis Effected (happy, uncomfortable, etc.), Buff Length, Buff GUID, Custom class and assembly name, Buff Moodlet Icon
4. Localization: You need a STBL file which contains the strings for your buffs. Note: Your buffs must be fully localized for all languages (even if you use Enlgish for all languages) or else they will not show properly in non-English games.

A word on terminology in this Tutorial:
In the Sims, the words buff and moodlet are interchangeable. I have also used the terms interchangeably in this tutorial. Whenever I refer to a buff, it means moodlet, and vice versa.

Now, Let's get started. Go to: Part One: Creating the Code

« Last Edit: 09 January 2013, 23:01 by Nona Mena » Logged

Visit My Blog | My Mod Index | My profile at MTS | Fix Your Premium Content

☐ **Nona Mena**
Moderator
🟩🟩🟩🟩

Re: How to Create and Load Custom Moodlets with a Pure Script Mod
« **Reply #1 on:** 20 June 2012, 16:55 »

Part One: Creating the Code

Before you begin

If you haven't already done so, you'll need to create a game compatible Visual Studio project. Please do not forget to add *[asssembly: Tunable]* and *using Sims3.SimIFace;* to your AssemblyInfo.cs

Posts: 768
Gender:

IMPORTANT: When you add the Sims 3 libraries to your references, skip the Sims3GameplaySystems.dll. Instead, run it through Peter's/Twallan's unprotect program (Download program) and reference the unprotected library. You can run all of the libraries through the unprotect program, but I recommend that you unprotect only what you need.

1. Loading Custom Moodlets

Step 1: The Script
Loading custom moodlets into the game is a lot easier since the Store Premium Content Items. EA has a method to load store moodlets which we can basically hijack and slightly alter to load our own moodlets. You can find it in Sims3.Gameplay.ActorSystems.BuffManager, CreateBuffTable().

I recommend creating a new class in your VS project, and naming it BuffBooter (you can name it whatever you want. I think I stole this name from Twallan/Buzzler. Just make sure you change the name in the Instantiator class as well). Paste this in:

Code: [Select]

```
public void LoadBuffData()
{
    this.AddBuffs(null);
    Sims3.UI.UIManager.NewHotInstallStoreBuffData += new Sims3.UI.UIManager.NewHotInstall
}

public void AddBuffs(ResourceKey[] resourceKeys)
{
    ResourceKey key = new ResourceKey(ResourceUtils.HashString64("Your_Buffs_XML_Name_Her
    XmlDbData data = XmlDbData.ReadData(key, false);
    if (data != null)
    {
        BuffManager.ParseBuffData(data, true);
    }
}
```

(source) Example BuffBooter Class

Notice where it says: **Your_Buffs_XML_Name_Here**. That's where you'll put the name of your Buffs XML, which will just be a normal XML file (doesn't need to be a BUFF file). You can name your Buffs XML whatever you like. The store Buffs XML is called buffs_store. We'll talk more about the Buffs XML later.

Step 2: The Instantiator
Now, to get the game to actually read this code and load your custom moodlets, you'll need an instantiator. This is thoroughly explained in the Pure Scripting Modding tutorial at MTS. *You can find an example Instantiator class at the bottom of this post*. Add this line to the static constructor of your instantiator class:

Code: [Select]

```
LoadSaveManager.ObjectGroupsPreLoad += OnPreLoad;
```

And this callback method:

Quote

> private static void OnPreLoad()
> {
>   (new BuffBooter()).LoadBuffData();
> }

It should look like this in your code:

Code: [Select]

```
public static class Instantiator
    {
        [Tunable]
        internal static bool kInstantiator = false;

        static Instantiator()
        {
            LoadSaveManager.ObjectGroupsPreLoad += OnPreLoad;
        }
        public static void OnPreLoad()
        {
            (new BuffBooter()).LoadBuffData();
        }
    }
```

(source)

Don't forget that you're going to need an XML file with your kInstantiator tunable in it so the game loads your mod. Again, this is explained in the Pure Scripting Modding tutorial at MTS. *You can find an example Instantiator XML at the bottom of this post.*

Finally, make sure that you add *[asssembly: Tunable]* and *using Sims3.SimIFace;* to your AssemblyInfo.cs. This is also explained in the Pure Scripting Modding tutorial at MTS.

2. Creating Guids for your Buffs

This is the last part of the coding involved with loading the custom moodlets. You'll need to create a guid for each of your moodlets. Take a look at my TastyTreatBuff:

Code: [Select]

```
internal class BuffTastyTreat : Buff
    {
        private const ulong kNonaTastyTreatBuffGuid = 0x9628DB30E3C25B4B;
```

```
        public static ulong StaticGuid
        {
           get
           {
                              return 0x9628DB30E3C25B4B;
           }
        }
        public BuffTastyTreat(Buff.BuffData data) : base(data)
        {
        }
     }
```

(Source: Buzzler)

0x9628DB30E3C25B4B is the FNV64 Hash of NonaTastyTreatBuff. The only important part about this Guid is that it is unique. You can call the constant whatever you want, and the hash can be a hash of anything. It just needs to be unique.

Now that we've got all of the coding part done, we can move on to the Buffs XML.

Recap:

1. Use the code the game uses to load store object buffs.
2. Make sure that the LoadBuffData() method is called by calling the BuffBooter in your Instantiator.
3. Create a class and Guid for each of your buffs.

Documents you might find helpful:
Example BuffBooter Class
Example Instantiator Class
Example Buff Class

Instantiator XML file to import into S3PE

Next: Part Two: Creating your Buffs XML

« Last Edit: 09 January 2013, 23:09 by Nona Mena » Logged

Visit My Blog | My Mod Index | My profile at MTS | Fix Your Premium Content

---

☐ **Nona Mena**
Moderator
🟩🟩🟩🟩

Posts: 768
Gender:

Re: How to Create and Load Custom Moodlets with a Pure Script Mod
« **Reply #2 on:** 21 June 2012, 21:54 »

Part Two: Creating your Buffs XML

Now we can talk about the Buffs XML. Your Buffs XML has to follow a very specific format! In fact, I know this from first hand experience, after I omitted a line and my buffs wouldn't load. It was a source of much frustration. So make sure you don't accidentally delete anything.

You can start by taking a look at this sample Buffs XML, which is from my Breastfeeding mod.

Download: buffs_NonaMena_BreastfeedBaby.zip

1. The XML declaration and the root element

If you've installed NotePad++, this is where it will be useful. Open the XML file and let's take a look at what's inside.

Code: [Select]

```
<?xml version="1.0"?>
<buffs_NonaMena_BreastfeedBaby>
...
</buffs_NonaMena_BreastfeedBaby>
```

These lines at the beginning and end of the file are important. The first line is the XML declaration (<?xml version="1.0"?>), and you must not change it. The second line is called the root element (<buffs_NonaMena_BreastfeedBaby> in this case). Your XML must have a root element. You can name your root element any way you like, it doesn't matter at all. But it must be present. Make sure you define the end of the root element by closing the tag as the last line of the XML.

2. The BuffList

I seriously hope you downloaded and opened up the XML I uploaded, because I will now refer to it a lot. The first part of your BuffList is a sort of header template. It contains all the possible fields for the buffs. You won't need to fill out all of these fields, just the important ones. The template starts with the first <BuffList> tag and ends with the first </BuffList>. After that you must add each moodlet as its own separate BuffList.

If you follow the format of the example BuffList I've uploaded, you should have no problems with your BuffList. Alternatively, if you have a copy of a **ccmerged.package**, you can take a look at the BUFF xml included. Let's just go through each tag briefly.

Code: [Select]

```
<BuffList>
  <BuffName>PeacefulBabyBuff</BuffName>
  <BuffDescription>BreastFeedBaby_PeacefulBaby_BuffDescription</BuffDescription>
  <FirstPersonDescription>BreastFeedBaby_PeacefulBaby_BuffFirstPerson</FirstPersonDescrip
  <Category>Physical</Category>
  <AxisEffected>Happy</AxisEffected>
  <EffectValue>20</EffectValue>
  <TimeoutLength>120</TimeoutLength>
  <SKU>BaseGame</SKU>
  <Stackable>False</Stackable>
  <Hex>NonaPeacefulBabyBuff=0xBD673252B81C7857</Hex>
```

```
    <CustomClassName>NonaMena.BreastFeedBaby.Buffs.BuffPeacefulBaby, Nona_BreastfeedBaby</C
    <ThumbFilename>moodlet_NonaPeacefulBabyBuff</ThumbFilename>
    <Version>1.0</Version>
  </BuffList>
```

Some of the tags are self explanatory, so I'll leave those be. If you have a question about them, feel free to ask in the comments.

Tag	Description
<BuffDescription>	This will be part of your Key for Localization
<FirstPersonDescription>	Don't know the point of this. You could probably leave it out as it's not in the Store BUFF file.
<Category>	Valid categories include: Physical, Mental and Social
<AxisEffected>	Valid Axis includes: Happy, Stressed, Uncomfortable, None
<TimeoutLength>	Length of buff, in sim minutes
<SKU>	Valid: Store, BaseGame, EP1, EP2, EP3, EP4, EP 5, EP 6
<Stackable>	Whether it's possible to have more than one instance of the buff on a single sim
<Hex>	Must include the GUID you have in your script.
<CustomClassName>	Namspace and class of the moodlet, Assembly name
<ThumbFilename>	Name of the IMAG resource for the moodlet icon

Now that you know what each tag does, you're ready to make your own Buffs XML. Feel free to take the XML I've provided and edit it to fit your own buffs.

The TGI of your Buffs XML should be: 0x0333406C-0x00000000-FNV64Hash_of_the_filename_in_your_Script

To Recap:
1. Your XML **must have a root element**. You can call it whatever you like. It does not need to be unique.
2. Each buff must start and end with <BuffList></BuffList>
3. The value in the Hex tag should match the GUID in your script, for each buff.

Next: Part Three: Buff Elements and Localization

« *Last Edit: 22 June 2012, 14:54 by NonaMena* »

Logged

Visit My Blog | My Mod Index | My profile at MTS | Fix Your Premium Content

---

Nona Mena

Moderator

Posts: 768

Gender:

Re: How to Create and Load Custom Moodlets with a Pure Script Mod
« **Reply #3 on:** 22 June 2012, 13:31 »

Part Three: Buff Elements and Localization

This next part of the tutorial will focus on the different elements you need to make a working moodlet, and how to localize your moodlet.

1. Making a working moodlet: Required Buff Elements

Some of the information provided here will overlap with Part Two, but that's ok. If you don't get all the piece of your moodlet together, you won't actually have a moodlet! So it's important to make sure that you don't miss anything.

Every moodlet requires:

1. A Buff Name
2. A Buff Description
3. The Axis Effected
4. Buff Strength (+20, -50 etc.)
5. Timeout length (how long the buff lasts, in sim minutes)
6. Moodlet Icon
7. Optional: Custom Origin (for example, "From Nursing) ** This is done entirely via scripting and localization

If you're anything like me, then coming up with a good moodlet icon is probably the hardest aspect of this part.

Buff Name & Buff Description: How you name and describe your buff is up to you, but I do think it's better if they're more "sims-like" and cute. Try to keep both fairly short, so it's not difficult to read in game.

Buff Strength: Personally, I value custom moodlets that are not entirely overpowered.

Moodlet Icon: This one has some technical aspect you should be aware of. **The moodlet icon must not be larger than 54x54 pixels, and it must be in .png format.** Your moodlet icon should also have a transparent background, and it is better if it is not too detailed, since it is so small.
Note: The TGI for the moodlet icon is IMAG 0x2F7D0004-0x00000000-FNV64Hash_ofThumbFilename_IntheBuffsXML

Custom Origin: If you want your Buff to have a custom "From blablabla," you'll have to take care of that in the script, and in the localization. It's super easy though (Thanks to velocitygrass). I will cover custom origins in Part Four: Adding Moodlets and Buff Origins in the Script.

2. Localization

Localization is absolutely unavoidable if you want to make custom buffs. If you don't have any STBLs, your buffs will be completely empty.

In this part of the tutorial, the following tools and mods will be very helpful:

- Twallan's STBL program, available at his wiki (used to convert a text file to STBL, which I find easier when working with a lot of strings).
- STBL Duplicator - You can use this tool to make strings for all languages.
- Untranslated Key - Game Mod by Twallan which will show you what Key the game is expecting, instead of showing blank text.
- Stubble/String collector mod - Useful for trying to find already existing game keys.

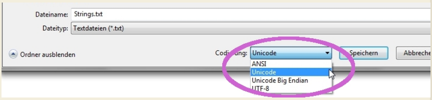
Also: S3PE has an excellent STBL editor. I have used this many times for adding new entries to an STBL, or for making an STBL that only needs a few lines. However, when I'm working with a lot more information, I find it much easier to create a text file and run that through twallan's STBL program.

Before we begin working on Localization, download these example strings:

NonaEdibleFood_Strings_EN.zip

You can use this document as a template for your own strings, or you can create a new text document. To create a new text document:

Open NotePad or NotePad++ to start a new text document. In NotePad, Use File > Save As... to save your document. Choose a name for your file, but don't save yet. Next to the save button, you'll find an Encoding menu. Click on that and choose **Unicode** (See picture below). Then Save.



Dateiname: Strings.txt

Dateityp: Textdateien (*.txt)

Codi ung: Unicode

Now you can begin working on your strings. Copy and paste the following into your text document:

Code: [Select]

```
<?xml version="1.0" ?>
<TEXT>



</TEXT>
```

You will put all of your strings inbetween the <TEXT></TEXT> tags. You can leave blank space between the strings, which will make it easy to separate the information by moodlet.

Each string that you need in the game must have two lines:

Code: [Select]

```
<KEY> </KEY>
<STR> </STR>
```

The STR field is where you put the actual text you want to be display in game. The KEY is where you put the location that the game needs to find the string. For moodlets, the key will always be (unless it's a BuffOrigin, then it will have a slightly different key, more on that later):

Quote

> Gameplay/Excel/buffs/BuffList:**YourBuffInformation**

The bolded part is unique to your moodlet, and you will need to use whatever you put in your Buff XML. Let's take a look at one of my own sets of strings for a single moodlet:

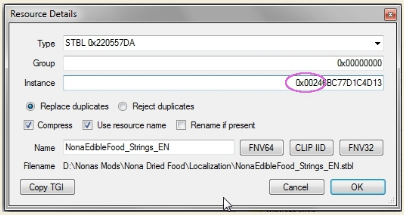
Quote

> <KEY>Gameplay/Excel/buffs/BuffList:TastyTreatBuff</KEY>
> <STR>Tasty Treat</STR>
> <KEY>Gameplay/Excel/buffs/BuffList:NonaDriedFood_TastyTreat_BuffDescription</KEY>
> <STR>Nothing hits the spot like a scrumptious treat.</STR>
> <KEY>Gameplay/Excel/buffs/BuffList:NonaDriedFood_TastyTreat_BuffFirstPerson</KEY>
> <STR>NonaTastyTreatFirstPerson</STR>
> *<KEY>Gameplay/Excel/buffs/BuffOrigins:98C5408FC6EFFFD9</KEY>*
> *<STR>(From Eating Mouthwatering Treat)</STR>*

This is what you'll need for each of your moodlets. If you're not going to use a custom Origin, you can skip the italicized part. Notice that the Key for the custom origin is a bit different from the other keys. I will go a little more in depth with custom Origins soon, but the **98C5408FC6EFFFD9** is the FNV64 Hash of "FromEatingMouthwateringTreat", with the "0x" from the beginning omitted. If you want the origin to look like the ones in game, you'll need to put your text in parentheses and start the text with "From."

Make one of these for each of your moodlets. Don't forget, in the <KEY></KEY> fields you must use the same BuffName etc, that you put in your BuffList in the Buffs XML. Once you've finished, save the document and then drag and drop it on top of Twallan's STBL.exe. This will create a new .stbl file, which you can import into S3PE.

To import your STBL into S3PE, simply drag and drop the .stbl file into S3PE. The Type should be: STBL 0x220557DA and enter 0 for the Group. The Instance ID does not actually matter, but the first two digits of the Instance ID must match the language code of your region. **For English, this is 00**. You can find the rest of the codes at the NRaas Wiki.



Resource Details

Type: STBL 0x220557DA

Group: 0x00000000

Instance: 0x00246BC77D1C4D13

● Replace duplicates ○ Reject duplicates

☑ Compress ☑ Use resource name ☐ Rename if present

Name: NonaEdibleFood_Strings_EN

Filename: D:\Nonas Mods\Nona Dried Food\Localization\NonaEdibleFood_Strings_EN.stbl

In the above picture, I've used the FNV64 Hash of "NonaEdibleFood_Strings_EN" for the Instance ID, and then changed the first two number to 00.

If you've gotten this far, you're almost done with your custom moodlets! The next section will focus on how to add the moodlets to sims, which must be done in your script (i.e. your interaction, or perhaps even an object, like the culinary career reward).

Next: Part Four: Adding Moodlets in the Script and Buff Origins

« *Last Edit: 26 June 2012, 17:19 by NonaMena* »

Logged

Visit My Blog | My Mod Index | My profile at MTS | Fix Your Premium Content

---

Nona Mena

Moderator

Posts: 768

Gender:

Re: How to Create and Load Custom Moodlets with a Pure Script Mod

« **Reply #4 on:** 22 June 2012, 14:52 »

Part Four: Adding Moodlets in the Script and Buff Origins

You're almost done with your custom moodlets! In fact, this section will be so short, you'll probably wonder why it's a section at all. I'm kind of wondering that too.

1. Adding Moodlets in the Script

In order to get your custom moodlet to actually appear when a sim does something, you're going to need a script. This tutorial isn't about how to make a script mod, or how to code in C#, so I won't tell you how to do that. In fact, if you've made a few mods already, you probably already know how to add a moodlet to a sim. But since this tutorial would be incomplete if I didn't cover this part, I have to cover it!

In EA code, a buff is added to a sim like this (for example):

Code: [Select]

```
actor.BuffManager.AddElement(BuffNames.Defeat, Origin.FromBeingSubduedByWasher);
```

If you want to add a buff to the target, instead of the actor, you use target, instead of actor.

For our custom buffs, we'll use the following overload method:

Code: [Select]

```
AddElement(ulong guid, Origin origin)
```

So, for example, to add the Peaceful Mama buff:

Quote

> this.Actor.BuffManager.AddElement(**0x3E8A488119A1FA27**,
> (Origin)ResourceUtils.HashString64("FromNursingChild"));

The bolded text is the Guid I assigned to the buff in the script (and also in the Buff XML).

2. Adding the Custom Buff Origin

Now take a look at this part of the line of code from above:
(Origin)ResourceUtils.HashString64("FromNursingChild")

This is how we get our custom origin into the game. Simple isn't it? Here's the localization for that Origin, from my Breastfeeding mod:

Quote

> <KEY>Gameplay/Excel/buffs/BuffOrigins:**64A3A132190947C9**</KEY>
> <STR>(From Nursing Child)</STR>

The hex number is the FNV64 Hash of FromNursingChild: 0x64A3A132190947C9, just without the 0x in front. (Source: Velocitygrass)

« *Last Edit: 24 June 2012, 10:00 by NonaMena* » Logged

Visit My Blog | My Mod Index | My profile at MTS | Fix Your Premium Content

---

☐ **Nona Mena**
Moderator

Posts: 768
Gender:

Re: How to Create and Load Custom Moodlets with a Pure Script Mod
« **Reply #5 on:** 22 June 2012, 15:49 »

Part Five: Finishing Up

The last few steps are upon us.

Once you're done testing and tweaking your mod and buffs, you can get ready for release. If you want players in other regions to be able to use your mod in English, you need to copy your English strings into STBLs for the other languages. This would be a lot of work, but luckily, there is a nifty STBL Duplicator program that will do this for you.

Using the STBL Duplicator

1. Simply download the STBL Duplicator (linked in Part Three) from MTS and install it.
2. The first time you run the program, it will ask you what language to use as default.
3. Choose your Default language and then drag your finished package file over the **TS3.STBLDuplicator.exe**

You have now successfully fully localized your mod.

Keep in mind that if you want to update your English strings and use the STBL duplicator, the STBL duplicator will replace all of the strings for the other languages. Keep copies of any translations you receive, so you can simply use the Replace function in S3PE to replace the strings with updated translations (don't forget to include any new fields). This is why having people translate the text document is so convenient.

Additional Credits

« *Last Edit: 23 June 2012, 10:16 by NonaMena* » Logged

Visit My Blog | My Mod Index | My profile at MTS | Fix Your Premium Content

---

☐ **Nona Mena**
Moderator

Posts: 768
Gender:

Re: How to Create and Load Custom Moodlets with a Pure Script Mod
« **Reply #6 on:** 22 June 2012, 15:56 »

Additional Credits

Like I said at the beginning of this tutorial, I did not come up with, discover or otherwise figure out any of this stuff on my own! I have merely put all of the information out there together into one super thread. This tutorial would not exist without:

Buzzler, Twallan, Velocitygrass and Peter Jones - All of them have helped me in some manner when it comes to this tutorial.

Cmo and Inge, because they are the ones who encouraged me to write this in the first place, and have been watching as I have painstakingly prepared this thread.

Sources:
Custom Buffs thread at MTS, started by ani
Custom buffs and origins thread at MTS, started by kjmarket

« *Last Edit: 23 June 2012, 10:16 by NonaMena* » Logged

Visit My Blog | My Mod Index | My profile at MTS | Fix Your Premium Content

---

☐ **Nona Mena**
Moderator

Posts: 768
Gender:

Re: How to Create and Load Custom Moodlets with a Pure Script Mod
« **Reply #7 on:** 22 June 2012, 15:56 »

Bonus: Useful Overload Methods for Buffs

Here are some useful overload methods that you can add to your Buff class. If you poke around in Sims3.Gameplay.ActorSystems.Buff, you might find some more useful stuff 🙂

Many thanks to Cherry for suggesting that I add these to the tutorial.

Code: [Select]

```
public virtual void OnAddition(BuffManager bm, BuffInstance bi, bool travelReaddition)
```

Code: [Select]

```
public virtual void OnRemoval(BuffManager bm, BuffInstance bi)
```

Hint: Take a look at Sims3.Gameplay.ActorSystems.BuffAdrenalineRush to see how you can use OnAddition(BuffManager bm, BuffInstance bi, bool travelReaddition) and OnRemoval(BuffManager bm, BuffInstance bi) to add another level of complexity to your moodlets.

Code: [Select]

```
public virtual void OnTimeout(BuffManager bm, BuffInstance bi, Buff.OnTimeoutReasons reas
```

Code: [Select]

```
public virtual bool OnReAddition(BuffManager bm, BuffInstance existingBuff, BuffInstance
```

« *Last Edit: 24 June 2012, 10:06 by NonaMena* » Logged

Visit My Blog | My Mod Index | My profile at MTS | Fix Your Premium Content

---

☐ **Nona Mena**
Moderator
🟩🟩🟩🟩

Posts: 768
Gender: ♀

Re: How to Create and Load Custom Moodlets with a Pure Script Mod
« **Reply #8 on:** 22 June 2012, 21:47 »

Coming Soon: Creating different types of buffs with the buffs XML.

« *Last Edit: 27 June 2012, 09:08 by NonaMena* » Logged

Visit My Blog | My Mod Index | My profile at MTS | Fix Your Premium Content

---

☐ **Nona Mena**
Moderator
🟩🟩🟩🟩

Posts: 768
Gender: ♀

Re: How to Create and Load Custom Moodlets with a Pure Script Mod
« **Reply #9 on:** 24 June 2012, 08:33 »

Feel free to ask any questions or point out mistakes, inaccuracies, typos, etc. I also like suggestions for improvement. (thanks, Cherry!)

« *Last Edit: 27 June 2012, 09:08 by NonaMena* » Logged

Visit My Blog | My Mod Index | My profile at MTS | Fix Your Premium Content

---

☐ **Nona Mena**
Moderator
🟩🟩🟩🟩

Posts: 768
Gender: ♀

Re: How to Create and Load Custom Moodlets with a Pure Script Mod
« **Reply #10 on:** 27 June 2012, 09:07 »

reserved

Logged

Visit My Blog | My Mod Index | My profile at MTS | Fix Your Premium Content

---

☐ **cmo**
Normal
🟨
Posts: 241

Re: How to Create and Load Custom Moodlets with a Pure Script Mod
« **Reply #11 on:** 08 August 2012, 19:13 »

Thanks a lot for this tutorial, Nona. I couldn't have made my Smoking Mod without it.
🙂

Logged

☐ **gumbyscout**
Normal
🟨

Posts: 7
Gender: ♂

Re: How to Create and Load Custom Moodlets with a Pure Script Mod
« **Reply #12 on:** 19 August 2012, 18:22 »

I went through your tutorial, and I still can't seem to get the moodlets/buffs to show up in the game. I've even gone through your breast feeding mod to see if you did anything totally different, and I still can't find the problem. I haven't added an interaction that applies the moddlet to a sim yet(I, however, did for a previous mod that wouldn't load the moodlets either. The object interaction menu would show up, but wouldn't do anything.).

Here's my .package file you can help me find the issue:

[attachment deleted by admin]

Logged

☐ **Nona Mena**
Moderator
🟩🟩🟩🟩

Posts: 768
Gender: ♀

Re: How to Create and Load Custom Moodlets with a Pure Script Mod
« **Reply #13 on:** 20 August 2012, 12:18 »

Quote from: gumbyscout on 19 August 2012, 18:22

> I went through your tutorial, and I still can't seem to get the moodlets/buffs to show up in the game. I've even gone through your breast feeding mod to see if you did anything totally different, and I still can't find the problem. I haven't added an interaction that applies the moddlet to a sim yet(I, however, did for a previous mod that wouldn't load the moodlets either. The object interaction menu would show up, but wouldn't do anything.).
>
> Here's my .package file you can help me find the issue:

Thanks for uploading the package. I looked at the Buffs XML and that was fine, so I went through your .dll and found the problem. The main culprit is that you didn't add *[assembly: Tunable]* attribute to your AssembyInfo. This is covered in the Pure Scripting Modding Tutorial at MTS, which is why I didn't add it to this particular tutorial. But perhaps I should add a reminder.

You can add the *[assembly: Tunable]* attribute to the AssemblyInfo.cs. You will need to add *using Sims3.SimIFace;* as well. Alternatively, you could add it to your Instantiator class, after the usings. But I recommend just getting into the habit of always adding *using Sims3.SimIFace;* and *[assembly: Tunable]* to the AssemblyInfo.

After you do that go back and clean up your instantiator class so that it is looks like the example Instantiator class I linked.

That should get your moodlets loading. Good luck! Don't forget to come back and share your results.

« *Last Edit: 20 August 2012, 12:22 by Nona Mena* »

Logged

Visit My Blog | My Mod Index | My profile at MTS | Fix Your Premium Content

---

☐ **gumbyscout**

Normal

Posts: 7

Gender: ♂

Re: How to Create and Load Custom Moodlets with a Pure Script Mod

« **Reply #14 on:** 24 August 2012, 00:51 »

I hate to bother you again, but after going through your tutorial the third time from scratch, I was finally able to get the moodlet from the tutorial to show up in the game. However, I haven't been able to reproduce it with different code. I've been pulling my hair out trying to figure out what small thing caused the test to work, but is causing the other ones not to work. Also, how does the logger class work in your breastfeeding baby mod? Having some sort of debugging thing would be of great help, and I haven't seen anyone discuss it.

Here's the files, the first one is the test file that works, the second is the main project that I'm working on, where the one defined moodlet doesn't show up. Thank you for your patience and assistance.

[attachment deleted by admin]

Logged

Pages: [**1**] 2 3

PRINT

« previous next »

Simlogical and the Dead End Nights » Forum » The Sims 3 » TS3 Tutorials and Reference » Gameplay Tuts and Refs (YibToilet: Nona Mena) » How to Create and Load Custom Moodlets with a Pure Script Mod

Jump to: ===> Gameplay Tuts and Refs go

SMF 2.0.5 | SMF © 2013, Simple Machines

TinyPortal © 2005-2012

XHTML RSS WAP2