NPC Modding Guide

For Baldur's Gate: Enhanced Edition

Introduction

Modding infinity-engine games is not as obvious as it could be. This tutorial is an attempt to explain the basics of IE modding. At the end of this tutorial you will hopefully have added your own NPC to the game.

This tutorial will accomplish following things:
- Create an NPC Creature
- Add a small dialog to the NPC
- Add a small script to the NPC
- Spawn the NPC in an existing area
- Package everything with a simple installer

The tutorial NPC can be downloaded here.

Tools

In this tutorial, we'll be using several tools:

Baldur's Gate Enhanced Edition

The obvious. We'll be using the game to create as well as test our NPC.

Enabling Cheats
You may want to enable cheats for easier developing & testing.
1. Open **"baldur.ini"** file with a text editor (notepad++) located in your **"Documents/Baldur's Gate - Enhanced Edition"** folder.
2. Add following line below "INSERT INTO options ROWS ("
   'Program Options','Debug Mode', '1',
3. In-game, you can press *ctrl* + *space* to open the console

WeiDU - The Weimer Dialog Utilities

http://www.weidu.org/main.html#weidu
WeiDU started out as a tool for writing dialogs, but today it's much more than that. This tool allows us to package our mod and maintain optimal compatibility with other mods (and other editions of BG).

DLTCEP - DragonLance TC Editor

http://sourceforge.net/projects/gemrb/files/Utilities/
This is an unofficial game editor for IE games. While there are other options as well, I'll be using this one.

Configuring DLTCEP

Before we can use DLTCEP, we'll have to configure the editor to use BGEE settings. You can add configurations for multiple games (BG2, BGEE, IWD, …) but we'll just stick to BGEE.

After running DLTCEP, the editor will warn you to setup first. Click the **Setup** button in the bottom-left corner. A new window will open where you can change your configuration options.

- Give your setup a name and change the game type to **BG[2]EE**.
- Make sure the global preference **read-only** is disabled. Otherwise you won't be able to save files.
- You'll need to locate the **chitin.key**. This file is located in your BGEE installation directory and allows DLTCEP to locate all files that are used in BGEE.
- You'll need to locate IEEP effect descriptions. The appropriate file is **BG2Effects.dat**, and is located in your DLTCEP installation directory.
- Press **Save & Back**.
- DLTCEP will now reload all BGEE data.
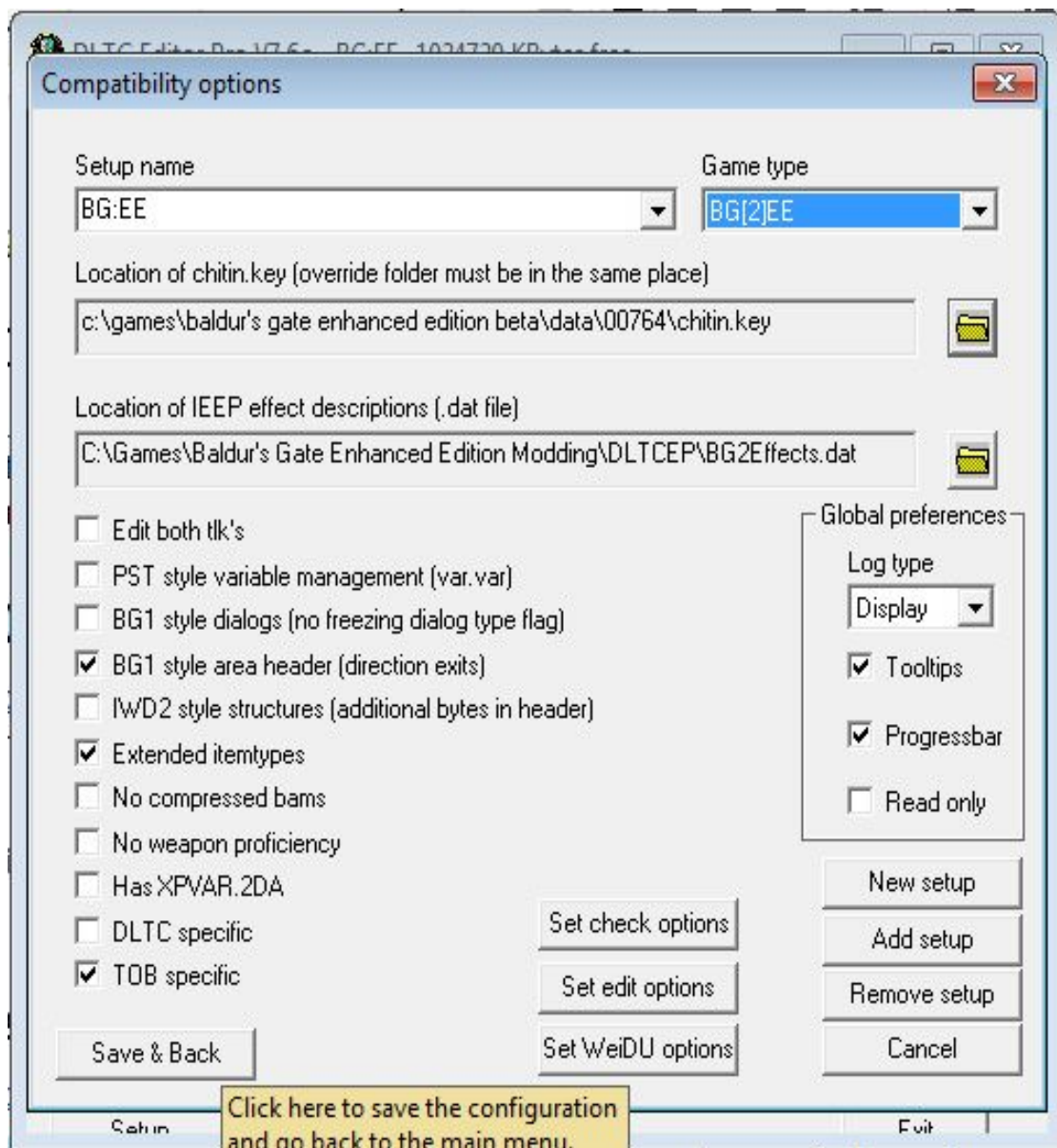
Notepad++

We'll need a text editor for creating dialogs & scripts. Any text editor (like notepad) will do, but I'll be using notepad++ which supports syntax highlighting. This allows us to easily identify basic scripting errors.

Syntax Highlighting

If you've installed notepad++, you'll have to install the syntax highlighters for WeiDU files.
https://docs.google.com/file/d/0B6Y0ez-SKHL1NHJ5M05nLVVkX1E/edit?usp=sharing
Unpack the 4 files somewhere.

Open Notepad++. Go to *Language* > *Define Your Language* > *Import* and select *"WeiDU D.xml"*. You've now imported syntax highlighting for dialog files. Continue importing all 4 xml files in the archive.

Things you need to know

Before we start, there are a few things you need to know about modding IE games. The Infinity Engine was developed in the late nineties and as such comes with several limitations.

File name limit

File names can only be at most 8 characters long. That is, the creature file for an NPC named "Thea Branden" cannot be named *"TheaBranden.cre"* (11 characters long). *"THEA.cre"* would be valid though.

Name Collisions

Name collisions with other files will cause the newer file to completely overwrite the old file. To ensure your mod will not accidentally overwrite existing files and remain compatible with other mods, it's recommended to use a two-letter prefix for all your mods. For example, I'm using the *"E)"* prefix for all my mods. My "Thea" NPC file will be named *"E)THEA.cre"*.
Note that with the 8 character limit, that leaves only 6 characters for naming your files.

You may reserve your own prefix here: http://forums.blackwyrmlair.net/index.php?showtopic=113

File Organization

To keep things structured, we'll need some way to organize our files.

Create NPC Creature

The first thing we'll do is create our creature file. A creature file ends with the *.cre* extension and stores all creature properties such as experience, strength, spells, etc.

Export CHR File

The easiest way to create a new creature is by exporting a new character from Baldur's Gate.


Start a new game and work your way through the character generation of Baldur's Gate. Enter a placeholder name & biography, we'll be changing this later.

If you have cheats enabled, you may set your NPC's abilities to maximum (all 18's) by pressing *ctrl +*

8. This allows you to set abilities without having to reroll a hundred times.
When you're ready, accept & start the game.

In the game itself you may use cheats to "finalize" your character.
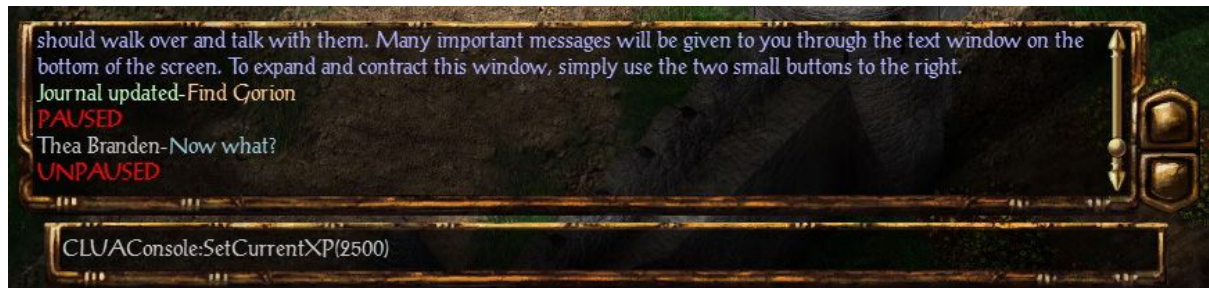
You can open the console by pressing *ctrl + space*. If your player doesn't start at level 1, it's easier to add experience now and level-up so Baldur's gate will automatically calculate your new abilities (THAC0, spells, …)

When you're ready, go to your character sheet and press *Export*. Don't worry about forgetting something, changes can always be made later.

Your exported character file can be found in *"Documents/Baldur's Gate - Enhanced Edition/Characters"*.

Create CRE file in DLTCEP

The next step is to create a **.cre** file in DLTCEP.
Open DLTCEP. Go to *Edit* > *Creatures* and click **Load External Creature**.

In the file dialog window, make sure you have selected files of type *"Character Files (.chr)"*.
Locate your **CHR** file that you have previously exported.
Congratulations, you've now imported your character.
Don't worry if your character doesn't have a name or biography.

Required Changes

Before we continue, you **must** ensure that you modify following fields:
- General Properties:
  ○ Class Flags: *0x0000 Normal* (this disables the "export" button on the creature)
  ○ Reaction: *0x80 neutral* (this makes the character behave neutral instead of like a "PC")
- Icons & Scripts:
  ○ Override Script: *E)Thea* (this identifies our script file)
  ○ Default Script: *NONE*
  ○ Dialog: *E)Thea* (this identifies our dialog files)
  ○ Scripting name (Death variable): *E)Thea* (this is a unique identifier for our NPC and allows scripts & dialogs to reference our NPC)

Note that our dialogs & scripts can have any name. It's probably a good idea to use the same name for the override script, dialogs and death variable.

Optional Changes

You can now continue editing your character as you please. I'll be adding some gold to my creature so when Thea joins the party, we gain some gold.

Something you may want to change are the portrait file references.
- Icons & Scripts
  ○ Small portrait: *E)THEAS* - the small portrait file
  ○ Large portrait: *E)THEAM* - the medium portrait file

Don't worry if your portrait won't be displayed.

When you're ready, press **Save Creature As** and browse to your **TheaNPC** directory. I'm saving the file as **E)THEA.cre**. Remember: files can only be up to 8 characters long.

Portraits

Portraits for BG:EE are 24bit bitmaps (.BMP). The size of BGEE portraits differ from other IE games.
Refer to Appendix IV to maintain compatibility with older versions of Baldur's Gate.

Medium Portrait

Size: 169x266

Create Dialogs

Our NPC needs to have a minimum amount of dialog in order to be able to join and leave our party. Of course banter and other dialogs will give personality to our NPC, but for now we'll be sticking with the basics.

The Infinity Engine uses **.DLG** dialog files. These are binary files and very hard to edit manually. Don't worry though, there's also a **.D** format. This is a simple text file that can be converted (compiled) into one or more **.DLG** file(s) using WeiDU.

The WeiDU readme has an introduction section on dialogs here.

The Basics

Dialogs in Baldur's Gate are simple finite state machines. You can think of finite state machines as a type of flow-charts. A dialog is composed of multiple states that are connected by transitions. Transitions can be taken by selecting a specific dialog option in-game.

When the player initiates a dialog with a creature in the game, the game will scan through all dialog states until it encounters a dialog state with a non-empty trigger that evaluates to true. If a dialog state is found, this is the starting point of the dialog. If no dialog state is found, the game will display a "Thea - has nothing to say to you" message.

An NPC has 3 different dialogs attached to it:
1. **E)THEA.d** - Initial dialog
2. **E)THEAP.d** - Post-party dialog (this runs after kicking our NPC from the party)
3. **E)THEAJ.d** - Join-party dialog (this runs while the NPC is in the party)

Again, note that the name of our dialogs can be at most 8 characters long. You're free to use any name you want, but it's a good idea to use your NPC's name, suffixed by **P** and **J**.

Let's get started and create a new dialog file.
Open Notepad++ and create a new file. Save it as **E)THEA.d** in the **"TheaNPC/Dialog"** directory.
This single **.D** file will contain all 3 dialogs (initial, post-party and join-party).

The Initial Dialog

```
BEGIN ~E)THEA~ // this identifies our initial dialog file
```

Any text following two slashes (//) is a comment and will be ignored by WeiDU. Comments can be useful to keep things organized. A comment only spans a single line.

Any text between **/*** and ***/** is a block-comment and will be ignored by WeiDU. Block comments can span multiple lines but must be closed.

Any text within tildes (~) is a string, or a piece of text. The sequence **~~** is simply an empty string.

BEGIN is a keyword that tells WeiDU that we want to begin a new dialog. The name of the dialog is identified by the string **~E)THEA~** that follows **BEGIN**.

Note that as you type, certain words will light up in a certain color. This helps early detection of errors (for example forgetting to close a string with a tilde). Dialogs & Scripts are case sensitive.

```
// New Dialog State (named "Introduction")
IF ~NumTimesTalkedTo(0)~ THEN
BEGIN Introduction
    // Dialog contents go in here
END // End Introduction
```

This is a single dialog state.
The triggers of this dialog state are put between the **IF** and **THEN** keywords. This basically means: if the player initiates a conversation with Thea, and the number of times Thea has talked to you is 0, this dialog state will activate.

BEGIN is a keyword that tells WeiDU that we want to begin a new dialog state. The name of the dialog state is called **Introduction**.

END marks the end of the dialog state. After this keyword we may start another dialog state.

```
    SAY ~Hello to you, my <LADYLORD>! I'm Thea Brandon, may I know your name?~
```

This is what Thea says to you. A **SAY** statement is *mandatory!*
Certain tokens like **<LADYLORD>** are replaced by the game with the appropriate word (lord if the player is male, lady if female). You can find a list of tokens [here](#).

After the **SAY** statement comes a list of transitions. Transitions are dialog options that the player can take in order to go to a different dialog state (and invoke a response from Thea).

```
// First transition
    IF ~ReputationLT([PC], 6)~ THEN
        // This is your first reply
        REPLY ~Certainly, I'm <CHARNAME>. What are you doing here?~
        GOTO IntroductionEvil
```

The triggers of a transition are put between *IF* and *THEN* keywords. This means that certain dialog options will only be available to the player if certain conditions are met. In this case, this dialog option is only visible if the reputation of the player is less than 6.

REPLY is what the player will say when this transition is taken.
GOTO identifies the dialog state to go to when this dialog option is chosen.

Our complete dialog state will look like this:

```
IF ~NumTimesTalkedTo(0)~ THEN
BEGIN Introduction
    SAY ~Hello to you, good <LADYLORD>! I'm Thea Brandon, may I know your name?~

    IF ~ReputationLT([PC], 6)~ THEN
        REPLY ~Certainly, I'm <CHARNAME>. What are you doing here?~
        GOTO IntroductionEvil

    IF ~ReputationGT([PC], 5)~ THEN
        REPLY ~Certainly, I'm <CHARNAME>. What are you doing here?~
        GOTO IntroductionGood

    IF ~~ THEN
        REPLY ~Bugger off.~
        EXIT

END
```

Let's recapitulate what's happening in this dialog state:

The dialog state is only accessible if the number of times **Thea** has talked to you is 0 (i.e. the first time you talk to her).

There are 3 transitions (dialog options) that the player can take.
1. He can say his name, and go to the next dialog state (IntroductionEvil). This dialog option only appears if his reputation is below 6.
2. He can say his name, and go to the next dialog state (IntroductionGood). This dialog option only appears if his reputation is above 5.
3. He can tell Thea to bugger off and end the conversation.

We need to add a new dialog state (IntroductionEvil) for when the player has chosen the first dialog option in the Introduction.

```
IF ~~ THEN
BEGIN IntroductionEvil
    SAY ~So *you* are <CHARNAME>? You will fall by my hand!~

    IF ~~ THEN
```

```
            DO ~Enemy()~
            EXIT

END
```

This dialog state has *no* triggers, since there's an empty string (~~) between **IF** and **THEN**.
It can *never* be accessed when the player *initiates* a conversation with Thea.
However, the **GOTO** statement in <u>Introduction</u> can still go to <u>IntroductionEvil</u>.

Note that **SAY** is mandatory, but **REPLY** is not.
In this case, Thea will say something to the player and then attack him.

A transition has 0 or more features:
- **REPLY ~text~** - The player replies this dialog line.
- **JOURNAL ~text~** - Update your journal.
- **SOLVED_JOURNAL ~text~** - Update your journal (solved section)
- **UNSOLVED_JOURNAL ~text~** - Update your journal (unsolved section)
- **DO ~actions~** - These scipt actions are executed.

A transition must have exactly 1 flow feature:
- **GOTO state** - The dialog advances to the specified state
- **EXTERN file state** - The dialog advances to the specified state in a different file
- **EXIT** - the dialog ends

We need to add another dialog state (IntroductionGood) for when the player has chosen the second dialog option in the <u>Introduction</u>.

```
IF ~~ THEN
BEGIN IntroductionGood
    SAY ~<CHARNAME>? I must join your party!~

    IF ~~ THEN
        REPLY ~I'd be happy to have you in my party.~
        DO ~SetGlobal("KickedOut", "LOCALS", 0)
           JoinParty()~
        EXIT

    IF ~~ THEN
        REPLY ~No, thanks.~
        EXIT

END
```

If the first transition is taken, Thea will join our party.

DO ~JoinParty()~ will force Thea to join the party.

DO ~SetGlobal("KickedOut", "LOCALS", 0)~ will set a variable.

Variables are stored in savegames and keep track of your progress.
This action requires 3 parameters:

1. A name. A variable can be given any name. In our case *"KickedOut"* will keep track of Thea's status in the party.
2. A scope. This defines where the variable can be referenced.
   a. *LOCALS* - this references the active creature (Thea). The variable can only be used in Thea's dialogs.
   b. *GLOBAL* - this is a global variable and is accessible anywhere (also in dialogs of other creatures like Minsc).
   c. *ARcode* - this references a certain area. For example: *AR1200* references the docks area in Baldur's Gate.
3. A numeric value. In our case, 0 will mean that Thea is in the party and 1 means Thea was kicked out.

The Post Party Dialog

This dialog runs when **Thea** has been kicked out of the party. It should handle everything related to being kicked out & rejoining.

```
BEGIN ~E)THEAP~
```

This identifies the post-party dialog.

```
IF ~Global("KickedOut", "LOCALS", 0)~ THEN
BEGIN Kickout
    SAY ~Oh, you don't need me no more?~

    IF ~~ THEN
        REPLY ~No, I made a mistake. Please stay.~
        DO ~JoinParty()~
        EXIT

    IF ~~ THEN
        REPLY ~Yes, it's time to part ways.~
        GOTO KickoutFinal

END
```

There are several ways to handle being kicked out of the party. I personally prefer the BG2 way: the NPC walks up to you and asks you to reconsider. The player gets 2 dialog options: one in which the NPC rejoins, one in which the NPC really leaves.

```
IF ~~ THEN
BEGIN KickoutFinal
    SAY ~Very well. I'll travel back to the Friendly Arms Inn.~
```

```
    IF ~~ THEN
        DO ~SetGlobal("KickedOut", "LOCALS", 1)
            EscapeAreaMove("AR2300", 3326, 2528, 14)~
        EXIT

END
```

In this dialog state, the NPC will tell the player where he can be found, before leaving the area.

DO ~EscapeAreaMove("AR2300", 3326, 2528, 14)~ will make the NPC exit the current area and reappear in the specified area. This action requires 4 parameters:
1. Name of the area to reappear. In our case, this is **AR2300**, the Friendly Arms Inn.
2. X coordinate of the position.
3. Y coordinate of the position.
4. Facing angle of the NPC. This is 0 for south, 4 for west, 8 for north and 12 for east. In our case, **14** will be south-south-east.

There are several ways of finding coordinates on a map. If you have cheats enabled, you can run Baldur's Gate and press **ctrl + x** to view the coordinates of your mouse cursor.

```
IF ~Global("KickedOut", "LOCALS", 1)~ THEN
BEGIN Rejoin
    SAY ~You want me back in the party?~

    IF ~~ THEN
        REPLY ~Yes, I need your services.~
        DO ~SetGlobal("KickedOut", "LOCALS", 0)
            JoinParty()~
        EXIT
```

```
    IF ~~ THEN
        REPLY ~I don't think so.~
        EXIT
END
```

Finally, we'll need a dialog state to allow the NPC to rejoin our party. This should be straight-forward by now.

The Join Party Dialog

Since this part of the dialog isn't strictly needed and only contains interjections & banter, we'll leave this empty. See Appendix II for more information.

The complete dialog file

For reference, here's the complete dialog file for Thea Brandon.

```
/*
    Thea Brandon Initial Dialog
*/
BEGIN ~E)THEA~

IF ~NumTimesTalkedTo(0)~ THEN
BEGIN Introduction
    SAY ~Hello to you, good <LADYLORD>! I'm Thea Brandon, may I know your name?~

    IF ~ReputationLT([PC], 6)~ THEN
        REPLY ~Certainly, I'm <CHARNAME>. What are you doing here?~
        GOTO IntroductionEvil

    IF ~ReputationGT([PC], 5)~ THEN
        REPLY ~Certainly, I'm <CHARNAME>. What are you doing here?~
        GOTO IntroductionGood

    IF ~~ THEN
        REPLY ~Bugger off.~
        EXIT

END

IF ~~ THEN
BEGIN IntroductionGood
    SAY ~<CHARNAME>? I must join your party!~

    IF ~~ THEN
        REPLY ~I'd be happy to have you in my party.~
        DO ~SetGlobal("KickedOut", "LOCALS", 0)
            JoinParty()~
        EXIT

    IF ~~ THEN
```

```
            REPLY ~No, thanks.~
            EXIT

END

IF ~~ THEN
BEGIN IntroductionEvil
      SAY ~So *you* are <CHARNAME>? You will fall by my hand!~

      IF ~~ THEN
          DO ~Enemy()~
          EXIT

END

IF ~NumTimesTalkedToGT(0) ReputationGT([PC], 5)~ THEN
BEGIN IntroductionLater
      SAY ~You're back. Are you ready to apologise?~

      IF ~~ THEN
          REPLY ~Yes, I'm sorry. Do you want to join my party?~
          DO ~SetGlobal("KickedOut", "LOCALS", 0)
              JoinParty()~
          EXIT

      IF ~~ THEN
          REPLY ~Bugger off.~
          EXIT

END

/*
    Thea Brandon Post Party Dialog
*/
BEGIN ~E)THEAP~

IF ~Global("KickedOut", "LOCALS", 0)~ THEN
BEGIN Kickout
      SAY ~Oh, you don't need me no more?~

      IF ~~ THEN
          REPLY ~No, I made a mistake. Please stay.~
          DO ~JoinParty()~
          EXIT

      IF ~~ THEN
          REPLY ~Yes, it's time to part ways.~
          GOTO KickoutFinal

END

IF ~~ THEN
BEGIN KickoutFinal
      SAY ~Very well. I'll travel back to the Friendly Arms Inn.~

      IF ~~ THEN
          DO ~SetGlobal("KickedOut", "LOCALS", 1)
```

```
            EscapeAreaMove("AR2300", 3326, 2528, 14)~
        EXIT

END

IF ~Global("KickedOut", "LOCALS", 1)~ THEN
BEGIN Rejoin
    SAY ~You want me back in the party?~

    IF ~~ THEN
        REPLY ~Yes, I need your services.~
        DO ~SetGlobal("KickedOut", "LOCALS", 0)
            JoinParty()~
        EXIT

    IF ~~ THEN
        REPLY ~I don't think so.~
        EXIT
END
```

Scripts

Scripts are very similar to dialogs. They both share many aspects like triggers & actions.
Like dialog files, script files (.**BCS**) are binary files. Also like dialog files, there's an alternative
text-based format (.**BAF**) that can be compiled into .**BCS** files using WeiDU.

The Basics

Scripts are composed of individual script blocks. The game regularly scans through all active script
files, searching for a script block that has a trigger that evaluates to true. As soon as the game finds
such a block, that block is executed and the process ends, until the game scans through the script
again.

Thea Override Script

In Thea's override script, we'll force Thea to talk to you as soon as she sees you for the first time.
Open Notepad++ and create a new file. Save it as *E)THEA.baf* in the *"TheaNPC/Script"* directory.

```
IF
    See([PC])
    NumTimesTalkedTo(0)
THEN
    RESPONSE #100
        Dialogue([PC])
END
```

Like dialog files, script files have triggers between *IF* and *THEN* that determine when this script block
will execute.This script block will be executed when Thea sees the PC and has never talked to him
before, and force her to talk to the PC.

RESPONSE #100 is a weighted response. When a script block has multiple response blocks, only one of the blocks will be randomly executed.

```
IF
    See([PC])
    NumTimesTalkedTo(0)
THEN
    RESPONSE #50
        Dialogue([PC])
    RESPONSE #50
        DestroySelf()
END
```

In this example, there are 2 response blocks, each with 50% chance of executing. Note that the response weighing is relative.
RESPONSE #1 and **RESPONSE #2** are just as valid as **RESPONSE #33** and **RESPONSE #67** and both symbolize a 33% vs 67% execution chance.

Multiple triggers

As you might have noticed, scripts & dialogs may have multiple triggers. A script block with multiple triggers will only execute if ALL conditions are met.

If you want a script / dialog to run when at least one of the conditions is met, you can use the **OR** trigger.

```
IF
    See([PC])
    OR(2)
        NumTimesTalkedTo(0)
        NumTimesTalkedTo(3)
    InParty("MINSC")
THEN
    RESPONSE #100
        Dialogue([PC])
END
```

This script block will run only if:
1. The creature can see the PC *and*
2. Minsc is in your party *and*
3. The creature has talked 0 *or* 3 times to you.

OR receives 1 parameter, which defines how many of the following triggers are optional. In this case, the next 2 triggers are optional.

Spawn script

Up to this point, we still haven't added **Thea** to the game. The easiest way to add our NPC is to spawn her with a script.

Most areas in Baldur's Gate have an area script with the same name as the area. For example, **AR2300.are** (Friendly Arms Inn) has a complementing script file **AR2300.bcs**.

```
IF
    Global("E)SpawnThea", "GLOBAL", 0)
THEN
    RESPONSE #100
        SetGlobal("E)SpawnThea", "GLOBAL", 1)
        CreateCreature("E)THEA", [3326.2528], 14)
        Continue()
END
```

We first change the global **E)SpawnThea** to ensure that Thea is only spawned once.

CreateCreature receives 3 parameters:
1. **E)THEA** - the name of the .cre file
2. **[3326.2528]** - the coordinates of the creature. [x dot y]
3. **14** - the facing angle

Continue allows the game to continue scanning the script for other blocks that may await execution.

Packaging our mod

With all necessary files ready, it's time to package our mod. In this section we'll be providing installation instructions for WeiDU so you can install the mod simply by running a setup executable.

TheaNPC	23/10/2013 19:54	File folder	
Setup-TheaNPC.exe	29/09/2013 18:50	Application	928 KB
Setup-TheaNPC.tp2	23/10/2013 18:56	TP2 File	2 KB

At this point, the **TheaNPC** directory should contain all the files we have created so far.

● Download the latest version of WeiDU to the top-level directory and rename it to **Setup-TheaNPC.exe**.
● Create a new text file with notepad++ and save it as **Setup-TheaNPC.tp2**.

Both files can have any name you like, as long as they are both named the same.
It is a common convention to use **Setup-<something>** as a name.

WeiDU TP2 Instructions

The **TP2** file provides installation instructions that can be read by WeiDU to automate installation. WeiDU will take care of the nasty details for you.

```
BACKUP ~TheaNPC/Backup~
AUTHOR ~Alkalurops (vanbeurden.bart@gmail.com)~
VERSION ~v1.0~
```

BACKUP - A backup directory must be provided so WeiDU can log changes, allowing the mod to be uninstalled without harming Baldur's Gate.
AUTHOR - If an installation error occurs, your name & email will be displayed to the user so he can contact you.
VERSION - This is optional.

```
LANGUAGE
    ~American English~
    ~American~
    ~TheaNPC/Lang/American/Setup.tra~
```

This directive allows a user to select a language, allowing your mod to be available in multiple translations. At least 1 language must be specified.

LANGUAGE receives 3 strings:
1. ***~American English~*** - this is the name of the language as it is presented to the user.
2. ***~American~*** - this is the name of the language directory
3. ***~TheaNPC/Lang/American/Setup.tra~*** - This is the path to the directory containing the setup translation file. This file contains the english translation of ***Thea***. We'll be creating this file soon.

```
BEGIN ~TheaNPC Tutorial for BGEE~
```

BEGIN marks the beginning of a subcomponent. A mod may have many subcomponents that may be installed separately by the user. For a simple NPC mod, we'll just have 1 subcomponent.

```
COPY ~TheaNPC/E)THEA.cre~ ~override/E)THEA.cre~
```

COPY tells WeiDU to copy a single file or directory to a destination.
1. ***~TheaNPC/E)THEA.cre~*** - this is the path to the file to copy
2. ***~override/E)THEA.cre~*** - this is the destination path.
   Most of the time this will be in the ***override*** directory.

```
COPY ~TheaNPC/E)THEA.cre~ ~override/E)THEA.cre~
    SAY BIO @100
    SAY NAME1 @101
    SAY NAME2 @101
```

```
    SAY MORALE @200
    SAY LEADER @210
    SAY TIRED @220
    SAY BORED @221
    SAY BATTLE_CRY1 @230
    SAY BATTLE_CRY2 @231
    SAY BATTLE_CRY3 @232
    SAY BATTLE_CRY4 @233
    SAY BATTLE_CRY5 @234
```

After copying a file, we can make some last-minute modifications. This is called a **patch**. In our case, we'll be giving our creature a name, biography, and some sound set lines. These modifications are entirely optional, but do note that not giving your NPC a name or sound set will obviously make her nameless and mute.

SAY offset ~string~ - This directive assigns a string of text to the creature file.
1. **offset** - can be a number or an identifier. In our case, we use **NAME1**, **BIO** or **BATTLE_CRY1** to reference the name, biography and battle cries of our NPC.
2. **~string~** - this is the text to be displayed.
   For example: **SAY NAME1 ~Thea Branden~** will rename our creature to **Thea Branden**.

You might have noticed that we're not using strings here, but we're using **@100** or **@221**.
This is a reference to a translation entry. See the section on translation files below.

```
// Portrait Files
COPY ~TheaNPC/Portrait/EE/E)THEAS.bmp~ ~override/E)THEAS.bmp~
COPY ~TheaNPC/Portrait/EE/E)THEAM.bmp~ ~override/E)THEAM.bmp~

// Audio Files
COPY ~TheaNPC/Lang/%LANGUAGE%/Audio~ ~override~
```

Copy some more files to the override folder. Note that we're not copying our audio files individually but simply copy the entire **Audio** directory to the override folder.

%LANGUAGE% is automatically replaced with the language that the user selects, e.g. **~American~**

```
// Dialogs
COMPILE ~TheaNPC/Dialog/E)THEA.d~
```

The **COMPILE** instruction tells WeiDU to compile a dialog (**.D**) file into **.DLG** files.
In our case, **E)THEA.d** has 2 dialogs: **E)THEA** and **E)THEAP**.
E)THEA.d will be compiled into **E)THEA.dlg** and **E)THEAP.dlg**.

```
APPEND ~PDIALOG.2da~
~E)THEA E)THEAP E)THEAJ E)THEAD~
UNLESS ~E)THEA~
```

PDIALOG.2da is a [2DA](#) file where various dialogs are assigned to an NPC.

APPEND will insert a new row into the file.
Each row represents the dialog files for an NPC.
1. **E)THEA** identifies the NPC. This corresponds with the value that you have entered in the dialog field of the creature in DLTCEP.
2. **E)THEAP** identifies the post-party dialog
3. **E)THEAJ** identifies the join-party dialog
4. **E)THEAD** identifies the dream sequence (bg2 only)

UNLESS will make sure the row is only inserted if there's no other **E)THEA** row yet.

```
APPEND ~INTERDIA.2da~
~E)THEA E)THEAB~
UNLESS ~E)THEA~
```

Similar to PDIALOG.2da, INTERDIA.2da contains the dialog reference for banter dialogs.
Even though we haven't created any banter dialogs yet, there's no harm in adding this entry.

```
// scripts
COMPILE ~TheaNPC/Script/E)THEA.baf~
EXTEND_TOP ~AR2300.bcs~ ~TheaNPC/Script/AR2300.baf~
```

Like dialog files, script files have to be compiled.
COMPILE ~E)THEA.baf~ will compile that script into **E)THEA.bcs**.

EXTEND_TOP will take an existing script and add additional code to it.
In this case, we add some code (to spawn Thea) to the already existing area script for the Friendly Arms Inn.

Translation Files

Translation files are very simple files that map references (like **@100**) to strings.
This makes it easy for your mod to be translated to a different language while retaining the basic structure of your TP2 & dialog files.

```
@100    = ~When asked about her past, Thea Branden has little to say. She claims
she's a clean slate.~
@101    = ~Thea Branden~
@102    = ~Thea Branden~
```

```
@200      = ~The battle turns against us!~ [E)DNOMF1]
```

A TRA file is nothing more than a list of reference translation strings.
The basic syntax is simple: *@reference = ~string~*.
Whenever your TP2 file references *@reference*, it will be replaced by the referenced string.

In addition to a string, you may associate an audio file.
For example: *@200* references *~The battle turns against us!~*. Whenever your NPC says this combat line, the game will look for an audio file named *E)DNOMF1.ogg* or *E)DNOMF1.wav*.

2DA Files

2DA stands for 2-dimensional arrays.
This file format represents a table (or 2d array) with rows represented by lines and columns represented by spaces.

For example:

```
PDIALOG.2DA
MINSC     MINSCP     MINSCJ     MINSCD
E)THEA    E)THEAP    E)THEAJ    E)THEAD
...
```

Each line represents a row, in this case all dialog files for Minsc, Thea, and other NPC's.

The complete TP2 file

```
BACKUP ~TheaNPC/Backup~
AUTHOR ~Alkalurops~
VERSION ~v1.0~

LANGUAGE
    ~American English~
    ~American~
    ~TheaNPC/Lang/American/Setup.tra~

BEGIN ~TheaNPC Tutorial for BGEE~

COPY ~TheaNPC/E)THEA.cre~ ~override/E)THEA.cre~
    SAY BIO @100
    SAY NAME1 @101
    SAY NAME2 @101
    SAY MORALE @200
    SAY LEADER @210
    SAY TIRED @220
    SAY BORED @221
    SAY BATTLE_CRY1 @230
    SAY BATTLE_CRY2 @231
    SAY BATTLE_CRY3 @232
    SAY BATTLE_CRY4 @233
    SAY BATTLE_CRY5 @234
```

```
    SAY DAMAGE @240
    SAY DYING @241
    SAY HURT @242
    SAY SELECT_COMMON1 @260
    SAY SELECT_COMMON2 @261
    SAY SELECT_COMMON3 @262
    SAY SELECT_COMMON4 @263
    SAY SELECT_COMMON5 @264
    SAY SELECT_COMMON6 @265
    SAY SELECT_ACTION1 @270
    SAY SELECT_ACTION2 @271
    SAY SELECT_ACTION3 @272
    SAY SELECT_ACTION4 @273
    SAY SELECT_ACTION5 @274
    SAY SELECT_ACTION6 @275
    SAY SELECT_ACTION7 @276
    SAY SELECT_RARE1 @280
    SAY SELECT_RARE2 @281
    SAY REACT_TO_DIE_GENERAL @290

// Portrait Files
COPY ~TheaNPC/Portrait/EE/E)THEAS.bmp~ ~override/E)THEAS.bmp~
COPY ~TheaNPC/Portrait/EE/E)THEAM.bmp~ ~override/E)THEAM.bmp~

// Audio Files
COPY ~TheaNPC/Lang/%LANGUAGE%/Audio~ ~override~

// Dialogs
COMPILE ~TheaNPC/Dialog/E)THEA.d~

APPEND ~PDIALOG.2da~
~E)THEA E)THEAP E)THEAJ E)THEAD~
UNLESS ~E)THEA~

APPEND ~INTERDIA.2da~
~E)THEA E)THEAB~
UNLESS ~E)THEA~

// scripts
COMPILE ~TheaNPC/Script/E)THEA.baf~
EXTEND_TOP ~AR2300.bcs~ ~TheaNPC/Script/AR2300.baf~
```

The complete TRA file

TheaNPC/Lang/American/Setup.tra

```
// Thea Branden NPC Tutorial

@100    = ~When asked about her past, Thea Branden has little to say. She claims
she's a clean slate.~
@101    = ~Thea Branden~
@102    = ~Thea Branden~

@200    = ~The battle turns against us!~ [E)DNOMF1]
```

```
@210      = ~I will do my best to keep us under harm's way.~ [E)DNOBL1]

@220      = ~If I have to take another step my feet will fall off.~ [E)DNOTR1]
@221      = ~I think I can feel mold growing on my feet.~ [E)DNOBO2]

@230      = ~Your time draws near!~ [E)DNOBC1]
@231      = ~Face me if you dare!~ [E)DNOBC2]
@232      = ~May the Gods have mercy on you!~ [E)DNOBC3]
@233      = ~Surrender!~ [E)DNOBC4]
@234      = ~I'll take care of this one!~ [E)DNOBC5]

@240      = ~Ah!~ [E)DNOBH3]
@241      = ~Ah! Aaaah!~ [E)DNODY2]
@242      = ~I need healing!~ [E)DNOHU1]

@260      = ~Something amiss?~ [E)DNOSC1]
@261      = ~I hear you.~ [E)DNOSC2]
@262      = ~Do you need me?~ [E)DNOSC3]
@263      = ~Name your task.~ [E)DNOSC4]
@264      = ~Hmm?~ [E)DNOSC5]
@265      = ~Can I help?~ [E)DNOSC6]
@266      = ~Yes?~ [E)DNOSC7]

@270      = ~Immediately!~ [E)DNOSA1]
@271      = ~Here?~ [E)DNOSA2]
@272      = ~I'll take care of it.~ [E)DNOSA3]
@273      = ~All right.~ [E)DNOSA4]
@274      = ~Well and good.~ [E)DNOSA5]
@275      = ~I'll do my best.~ [E)DNOSA6]
@276      = ~With all haste.~ [E)DNOSA7]

@280      = ~A subtlety isn't your best quality.~ [E)DNOSR1]
@281      = ~If I write a story about our adventures, I'm leaving out this part.~
[E)DNOSR2]
@282      = ~Look, I'm listening, all right.~ [E)DNOSR3]
@283      = ~You are making me mad on so many levels right now.~ [E)DNOSR4]

@290      = ~May your final journey be a peaceful one.~ [E)DNORD2]
```

Conclusion

That's it!

You should now be able to install your mod by copying everything to **Baldur's Gate/Data/00764/** and running **Setup-TheaNPC.exe**

Appendix I: Helpful Links

1. IESDP - Infinity Engine Structures Description Project
   a. File Formats
   b. Effect Codes (lists all effects that can be added to items, …)

Appendix II: Banter