

The Complete Guide To NPC Creation

Using WeíDU and the Infinity Engine

**Spawning, DIALOGUE, Romance, Quests,
Music, Voicing, and more!**

**By:
Kat 'theaces' Bella**

© 2007 [Kat "theacefes" Bella](#). Unauthorized use of this manual (selling, distribution without my permission) is prohibited.

Table of Contents

[Section 1: Introduction and Overview](#)

[Section 2: Making the NPC: Basics](#)

[2.1 Joining/Intro Dialogue](#)

[2.2 The CRE](#)

[2.3 Making the NPC appear in a location](#)

[2.4 Putting the NPC into the game \(Tp2s\)](#)

[Section 3: Making the NPC: Intermediate](#)

[3.1 Banter with another NPC](#)

[3.2 Banter with the PC](#)

[3.3 Interjections](#)

[3.4 Player-Initiated Dialogues](#)

[3.5 Tree of Life talk](#)

[Section 4: Making the NPC: Advanced](#)

[4.1 Timers](#)

[4.2 Romance/Friendship Tracks](#)

[4.3 Give your NPC a Quest](#)

[4.4 Making your NPC for ToB](#)

[4.5 Epilogues](#)

[Section 5: Multimedia](#)

[5.1 Portraits](#)

[5.2 Sound and Music](#)

[5.3 Adding Sounds/Voicing into your mod](#)

[5.4 Adding Music into your mod](#)

[Section 6: Troubleshooting](#)

[6.1 Things to Check For](#)

[Summary](#)

[Links](#)

Section 1- Introduction and Overview

Congrats on deciding to make your very own NPC mod with the Infinity Engine! This manual contains everything you need to know about making an NPC, putting him/her into the game, and even giving them dialogue, quests, a custom portrait, a romance, and more. Before we start, I think there are a few things we should go over, just to make sure you are ready to get into modding.

A few questions to ask yourself first. Can you:

- Use a computer and navigate through your operating system with confidence?
- Use a text editor?
- Download files/applications?
- Confirm that your Infinity Engine game install is patched and working properly?
- Work calmly without having childish fits when you make a mistake?
- Present yourself with tact on the Internet?
- Read and write English to a somewhat decent extent?

If you can say "yes" to all of these things, you are probably ready to begin modding. If not, maybe you should go back and rethink your decision to try and make a mod.

Now, most IE modders are a part of the Infinity Engine (IE) modding community. There are several sites that make up the community. You can find a list of these sites in the [Links](#) section. The residents of these sites are more than happy to answer any questions you might have about modding, but if you wish to participate in these sites and their discussion forums and get anything out of them, please remember:

- To be polite when posting. Rudeness is never a good way to get anything, and even if someone is rude to you, you 1) probably deserved it if you were rude first and 2) should ignore them, and take any advice they might have for you related to modding and put it to use. It's not always pretty but that's the internet for you.
- Don't type in ALL CAPS LIKE THIS because you seem like you are shouting. Also, l33t speak (lazy "code" typing) is looked down upon (you will seem like an idiot).

- Do your own research before posting a question on the Q and A forums.
- Post your question in only one forum on one site. Most of the folks in the community frequent multiple sites, and posting at all of them is disrespectful to the modders of the particular site you posted in first. Be patient, and if someone doesn't reply after a while, post another *polite* post to bump your topic to the top of the forum so others will notice it. If you still have not received an answer, try sending a private message or an email to an established modder in the community.

Okay, now since you are reading this manual, I'm going to assume that as a modder, you are taking responsibility for your idea and your work...meaning that you are willing to write and code this mod... but most importantly, you are willing to take on the bulk of the work. Some modders work in teams, which is great, but everyone has his/her own share of work that they are expected to complete. Modding is a hobby; if you are too lazy to do it yourself, don't expect someone else to do it for you.

With that aside, let's get started. First, you should know a few definitions and concepts.

- **Infinity Engine:** The engine that games such as Baldur's Gate I and II, Icewind Dale I and II, and Planescape: Torment run with. The engine is able to be modded to a great degree, however there are still some things that are hardcoded into the game, meaning we can't change them. Here is a list of such things.
- **Non-Player Character (NPC):** Any character that is not you- the main character (PC). Examples include characters like Imoen, Nalia, Jaheira, Minsc, and more.
- **WeiDU (Weimer Dialogue Utility):** The utility that compiles all the pieces of your mod, created by Wesley Weimer and maintained now by thebigg. Here is Weimer's [website](#).
- **Near Infinity:** A very powerful tool used to edit and view various types of files in the game.
- **Infinity Explorer:** A tool used to view all the files in an Infinity Engine game.
- **ShadowKeeper:** A tool used to create and edit characters.
- **Dialogue:** Dialogue files contain all of an NPC's conversations such as banter and interjections. They are created as D files and compiled into the game as DLG files.
- **Scripts:** Scripts contain code that tells the game what to do, whether it be spawning an NPC into an area, beginning a quest, starting combat, etc. Scripts are created as BAF files and compiled as BCS files.
- **Batch Files:** Files that launch a command for WeiDU to do something, such as decode compressed audio files.
- **Tp2:** A file that tells WeiDU what to install in your mod and where.

- **Global:** A variable in the game. Globals are one of the most important concepts in modding. We will expand on this.
- **[IESDP](#):** A compilation of researching of the Infinity Engine done by numerous people in the IE community. This will become a very important source of information.
- **Trigger:** Block of code that causes the character to initial dialogue

Tools:

Okay, so let's get the tools we need to mod.

- [WeiDU](#)
- [ShadowKeeper](#)
- [Near Infinity](#)
- A Text Editor (Notepad, ConTEXT, Crimson, etc, though I recommend something with highlighter so it's easier to spot mistakes)
- [Infinity Explorer](#) (optional but useful)

Okay, got those? Good. Let's get started for real.

Section 2- Making the NPC: Basics

This section will go through the very basic steps of getting an NPC into the game. I am going to use my NPC, John, as an example. I'm going to assume you have a background, personality, etc for your NPC so I'm not going to go over that. For the sake of this manual, we are going to assume you want your NPC to join the party. This section will discuss making an introductory/joining dialogue, making a CRE, getting the CRE to spawn in an area of your choice, and installing our basic mod.

Here we go!

2.1 Intro/Joining Dialogue

Prefixes: Having your own prefix that you use in all of your coding is very important if you want for your mod to avoid any conflict with other mods. An example of a prefix would be my use of "K#" in front of all of my mod files and variables. You can view a list of the already [registered prefixes](#) or register your own.

- 1) NPCs need to have a few different files to function properly in the game. This includes files that will hold all the NPC's dialogues if they are in the party, bantering, leaving the party, or just having their introductory talk with the PC. You can name these anything you want as long as it's 7 letter or less. You should use your prefix at the beginning of your filename. I'll use my first initial and # so that my intro dialogue is named K#John.
- 2) Open up your text editor and save the file as "all files" type and in the box where you name
- 3) Save the file, type "K#John.d" or whatever you decided to name your NPC.
- 4) At the top of the page, type "BEGIN K#John". Now, in coding, everything is based off of an "IF...THEN" process. So, let's go ahead and begin our introductory dialogue. Also, when you are coding, try using "//" followed by a comment to organize your code. WeiDU won't pick it up as code so you won't get any errors.

```
BEGIN K#John
```

```
IF ~NumTimesTalkedTo(0)~ THEN BEGIN Talk1
```

This means that IF the number of times the PC and John have spoken is 0, then this dialogue, which is called "Talk1" will occur.

```
BEGIN K#John
```

```
IF ~NumTimesTalkedTo(0)~ THEN BEGIN Talk1
```

```
SAY ~Hey there, I'm John and I'm a new NPC!~
```

The tildes "~" are like quotation marks. "SAY" is obviously what John is saying to the PC. Now comes the choices the PC has to answer John with. All of the choices must start with IF ~~ THEN REPLY. If there is a special condition for one of the choices, then it would be put into those ~~ in "IF ~~ THEN REPLY"

```
IF ~~ THEN REPLY ~I'm <CHARNAME>, how are you, John?~ GOTO
HiThere
```

```
IF ~~ THEN REPLY ~I'm sorry, but I don't have time to speak with
you
```

```
right now.~ GOTO EndTalk
```

```
END
```

<CHARNAME> is what's called a "dialogue token". If you put <CHARNAME> anywhere in the conversation, in the game the PC's name will appear. Neat, huh? GOTO HiThere/EndTalk is telling the game that there will be two blocks of dialogue (also known as states) called "HiThere" and "EndTalk". GOTO is connected the PC's choice with its respected block. END is ending the block.

```
IF ~~ THEN BEGIN HiThere
```

```
SAY ~I'm fine. Can I join your group?~
```

```
IF ~~ THEN REPLY ~Sure, why not?~ DO ~SetGlobal
("JohnJoined", "LOCALS", 1) JoinParty()~ EXIT
```

```
IF ~~ THEN REPLY ~I'm afraid I can't accept your offer right now.  
Perhaps later.~ EXIT  
END
```

IF ~~ THEN BEGIN HiThere is beginning the block for Hithere. A Global is a variable. Globals can either be labeled as GLOBAL or LOCALS. GLOBALS are variables that can be used by any script in the game. LOCALS are variables that can only be used by the script running it. "JohnJoined" is a global in this. 0 means that the global is not active, while 1 means that it is (for most globals). DO ~SetGlobal("JohnJoined","LOCALS",1)JoinParty()~ is telling the game to make John join the party. EXIT ends the conversation while END ends the block.

```
IF ~~ THEN BEGIN EndTalk  
SAY ~(sigh) Very well. Just come back if you change your mind.  
I'll be here.~  
IF ~~ THEN EXIT  
END
```

This is pretty simple. Usually, if I want to put an action on the NPC's part, I use () and ** for emphasis on a word, but it's not necessary.

Okay, so...that's the first part of the dialogue is done. Now we will create the block where the PC returns to John after rejecting him (in case the PC changed his/her mind).

```
IF ~NumTimesTalkedToGT(0)~ THEN BEGIN Talk2  
SAY ~You're back! Can I join now?~
```

```
IF ~~ THEN REPLY ~Sure, why not?~  
DO ~SetGlobal("JohnJoined", "LOCALS", 1) JoinParty()~ EXIT  
IF ~~ THEN REPLY ~On second thought, I don't think that's a good  
idea right now.~ EXIT  
END
```

NumTimesTalkedToGT(0) looks complicated but it's quite simple. Basically, GT means "greater than". So...if the number of times talked to is greater than 0, then the SecondMeeting will begin. Okay...now...we have to make John's parting dialogue, which is the conversation you have with him when you kick him out. Go ahead and right under the SecondMeeting block, make a space and put BEGIN K#JohnP

```
BEGIN K#JohnP
```

```
IF ~Global("JohnJoined", "LOCALS", 1)~ THEN BEGIN LeaveGroup  
SAY ~What? You don't need me?~  
IF ~~ THEN REPLY ~My mistake, please stay.~ DO ~JoinParty()~ EXIT  
IF ~~ THEN REPLY ~You suck. Bye.~ DO ~EscapeArea()~ EXIT  
END
```

EscapeArea means that if you kick John out, he will not come back. If you want for John to be available again, use this instead.

```
IF ~Global("JohnJoined","LOCALS",1)~ THEN BEGIN LeaveGroup
SAY ~What? You don't need me?~
IF ~~ THEN REPLY ~My mistake, please stay.~ DO ~JoinParty()~ EXIT
IF ~~ THEN REPLY ~You suck. Bye.~ EXIT
END
```

```
IF ~Global("JohnJoined","LOCALS",0)~ THEN BEGIN ComeBack
SAY ~You want me to rejoin?~
IF ~~ THEN REPLY ~Welcome back.~ DO ~SetGlobal
("JohnJoined","LOCALS",1)
JoinParty()~ EXIT
IF ~~ THEN REPLY ~Nope, you still suck.~ EXIT
END
```

So that's it! Here is the whole thing!

```
BEGIN K#John
```

```
IF ~NumTimesTalkedTo(0)~ THEN BEGIN Talk1
SAY ~Hey there, I'm John and I'm a new NPC!~
IF ~~ THEN REPLY ~I'm <CHARNAME>, how are you, John?~ GOTO
HiThere
IF ~~ THEN REPLY ~I'm sorry, but I don't have time to speak with
you
right now.~ GOTO EndTalk
```

END

IF ~~ THEN BEGIN HiThere

SAY ~I'm fine. Can I join your group?~

IF ~~ THEN REPLY ~Sure, why not?~ DO ~SetGlobal
("JohnJoined", "LOCALS", 1)

JoinParty()~ EXIT

IF ~~ THEN REPLY ~I'm afraid I can't accept your offer right now.
Perhaps later.~ EXIT

END

IF ~~ THEN BEGIN EndTalk

SAY ~(sigh) Very well. Just come back if you change your mind.
I'll be here.~

IF ~~ THEN EXIT

END

IF ~NumTimesTalkedToGT(0)~ THEN BEGIN Talk2

SAY ~You're back! Can I join now?~

IF ~~ THEN REPLY ~Sure, why not?~ DO

~SetGlobal("JohnJoined", "LOCALS", 1)

JoinParty()~ EXIT

IF ~~ THEN REPLY ~On second thought, I don't think that's a good
idea right now.~ EXIT

END

```
BEGIN K#JohnP
```

```
IF ~Global("JohnJoined","LOCALS",1)~ THEN BEGIN LeaveGroup
```

```
SAY ~What? You don't need me?~
```

```
IF ~~ THEN REPLY ~My mistake, please stay.~ DO ~JoinParty()~ EXIT
```

```
IF ~~ THEN REPLY ~You suck. Bye.~ DO ~EscapeArea()~ EXIT
```

```
END
```

OR

```
IF ~Global("JohnJoined","LOCALS",1)~ THEN BEGIN LeaveGroup
```

```
SAY ~What? You don't need me?~
```

```
IF ~~ THEN REPLY ~My mistake, please stay.~ DO ~JoinParty()~ EXIT
```

```
IF ~~ THEN REPLY ~You suck. Bye.~ EXIT
```

```
END
```

```
IF ~Global("JohnJoined","LOCALS",0)~ THEN BEGIN ComeBack
```

```
SAY ~You want me to rejoin?~
```

```
IF ~~ THEN REPLY ~Welcome back.~ DO ~SetGlobal  
("JohnJoined","LOCALS",1)
```

```
JoinParty()~ EXIT  
  
IF ~~ THEN REPLY ~Nope, you still suck.~ EXIT  
  
END
```

Okay, that's the end of the Dialogue. Mind you, this is a VERY simplified version of a dialogue.

2.2 The CRE

- 1) A CRE is a creature file. Every character/monster/chicken/etc has a CRE file. In order to make one, we must use NearInfinity. The easiest way to do this is to use a CHR (character file; you know, the ones you can import ex: FIGHTER, CLERIC, etc)
- 2) Go ahead and open ShadowKeeper and open up a character file. Edit the stats and weapons if you like. Then, when you exit, ShadowKeeper will ask you to name it a different file. So...I picked the Fighter chr. so I'll change the stats to whatever I like and rename it John.
- 3) Now open NearInfinity. Go to the tab on the side labeled CHR and click on John. Now go to FILE and click "CONVERT CHR to CRE". It should save as JOHN.CRE
- 4) Now go to GAME, then OPEN FILE. Then click BROWSE. Find JOHN.CRE and open it in a new window.
- 5) Okay, now we have our CRE file open. You will need to do a few things. First, scroll down to where it says FLAGS. Click on FLAGS and at the bottom there should be a bunch of options, one of which says EXPORTS ALLOWED and it is checked. Uncheck it.
- 6) Where it says GOLD, you can double click that and change it if you want.
- 7) ANIMATION ID: You can change the way your NPC's paperdoll looks, for example, my mod, Auren Aseph, is a human, but her paperdoll is an elf.
- 8) Colors: Where it says "major/minor/skin/hair/etc" color, make sure you keep them all below 68 or

69...otherwise the game will crash when you try to load your NPC.

9) Portraits: where it says large and small portrait, you can enter a different filename if you don't like the default portraits, just remember to use .bmp files (make your own portrait and stick it in the Portraits folder in your BG2 directory); the game won't accept jpegs.

10) Scroll down until you get to Override Script: type in K#John (or whatever your npc's filename is)

11) Where it says DEFAULT SCRIPT, scroll through the options and find NONE, then click "update value". If you want for your NPC to initiate the conversation by walking up to the PC, scroll and find INITDLG.BCS and click "update value"

12) For Enemy-Ally, find Neutral 128 and update value, so that your NPC will have a blue circle when you meet them.

13) GENDER: This is where you change the gender to male or female

14) RACE: This is where you choose what Race your NPC will be if you wish to change it.

15) ALIGNMENT: Changes your NPCs alignment (mind you, you can do all these little changes in SHADOWKEEPER before you make the CRE)

16) Script name: Also called the Death Variable or DV for short. This is kind of your NPCs ID. You will use this in banter and other conversations as well as scripts in the game. Double click it and type K#John.

17) Dialogue: Right click and choose "edit as string" and type K#John. This is the dialogue that we just made.

Wow...that was a lot, but it wasn't too hard, was it? That's it for the CRE. Just save it otherwise you'll have done all this work for nothing!

2.3 Making the NPC appear in a location

So, now we want for our NPC to show up at a certain place. Since the Copper Coronet is where many NPCs can be found, let's go ahead and make him appear there. Mind you that you can use CTRL +X in any location in the game and the game will tell you the exact coordinates of your cursor. You can use these coordinates to have your NPC spawn at.

1) Go ahead and open up a new text editor document and once again, save as "all files" type and save as "AR0406.baf" without the quotes of course...because AR means "area" and 0406 is the number of the Copper Coronet.

2) Go ahead and type this:

```
IF  
Global( "JOHNExists" , "AR0406" , 0 )  
THEN  
RESPONSE #100  
SetGlobal( "JOHNExists" , "AR0406" , 1 )  
CreateCreature( "JOHN" , [ 1483.1934 ] , 3 )  
END
```

JOHNExists is a global (variable) in the game now. The 1483.1934 are the x, y coordinates (the game screen is like a graph) and 3 is the direction the NPC is facing (it goes from 1 to 15).

Okay, so save the file and close it.

2.4 Putting the NPC into the game (Tp2s)

All right, we are almost done!

- 1) In your BG2 directory, make a new folder and call it John. Open the folder and inside it, make another folder and call it "backup".
- 2) Find WeiDU.exe (it's the one that looks like a black window). Copy and paste it in your BG2 directory and rename it: Setup-John.exe
- 3) Now open up yet another new text file and save it as Setup-John.tp2 This will be what tells WeiDU what to install.
- 4) Copy and move (or just move) your dialogue file, your CRE file, and your area file into the John folder.
- 5) Now, in your TP2 file, type this:

```
BACKUP ~John/backup~
```

```
AUTHOR ~theacefes (bohdi2185@msn.com)~
```

```
BEGIN ~John for BG2:SoA~
```

```
// This adds our CRE file
```

```
COPY ~John/JOHN.cre~ ~override/JOHN.cre~
```

// This allows for your NPC's name to be shown in the game during conversations and also on the character sheet

SAY NAME1 ~John~

SAY NAME2 ~John~

// This is the biography. In NearInfinity, go to that side tab and find CHR. Open that and find

JOHN.BIO You can create your own bio this way by simply deleting the default PC bio and

typing your own. Save this.

SAY BIO ~When you ask him about his past, JOHN says to shut up.~

// These are the different things that your NPC will say when they respond to your commands.

SAY MORALE ~Sticking around is only going to get us killed.~

SAY HAPPY ~This group is great~

SAY UNHAPPY_ANNNOYED ~You're starting to take the wrong path in morals.
~

SAY UNHAPPY_SERIOUS ~I'll leave if you don't stop!~

SAY UNHAPPY_BREAKING ~You suck, I'm outta here.~

SAY LEADER ~Yay I'm the leader!~

SAY TIRED ~I'm tired.~

SAY BORED ~I'm bored.~

SAY BATTLE_CRY1 ~TO BATTLE!~
SAY BATTLE_CRY2 ~Time to die!~
SAY BATTLE_CRY3 ~DIE NOW!~
SAY BATTLE_CRY4 ~DIE!~
SAY DAMAGE ~Ow!~
SAY DYING ~AH!~
SAY HURT ~Help me!~
SAY AREA_FOREST ~We're in a forest.~
SAY AREA_CITY ~We're in a city~
SAY AREA_DUNGEON ~We're in a dungeon.~
SAY AREA_DAY ~It's day.~
SAY AREA_NIGHT ~It's dark.~
SAY SELECT_COMMON1 ~Yeah?~
SAY SELECT_COMMON2 ~What do you need?~
SAY SELECT_COMMON3 ~Grunt.~
SAY SELECT_COMMON4 ~What?~
SAY SELECT_COMMON5 ~Yes?~
SAY SELECT_COMMON6 ~I'm listening.~
SAY SELECT_ACTION1 ~Uh huh.~
SAY SELECT_ACTION2 ~Sure.~
SAY SELECT_ACTION3 ~I'm on it.~
SAY SELECT_ACTION4 ~YAY!~
SAY SELECT_ACTION5 ~Not hard at all~
SAY SELECT_ACTION6 ~Yep~
SAY SELECT_ACTION7 ~Okay~

SAY SELECT_RARE1 ~I'm ready.~

SAY SELECT_RARE2 ~I got it.~

SAY CRITICAL_HIT ~Yeah!~

SAY CRITICAL_MISS ~Damn!~

SAY TARGET_IMMUNE ~No effect?~

SAY INVENTORY_FULL ~It's too heavy!~

// This is compiling John's dialogue

COMPILE ~John/K#John.d~

// This is adding our area file to the game

EXTEND_TOP ~AR0406.bcs~ ~John/AR0406.baf~

// Add this to the end (Appends .2DA files in the game so that it recognizes your NPC's files

APPEND ~pdialog.2da~

~K#John K#JohnP K#JohnJ K#JohnD K#Joh25P K#Joh25J K#Joh25D K#John25~

UNLESS ~K#John~

APPEND ~interdia.2da~

~K#John K#JohnB K#JohB25~

UNLESS ~K#John~

There you go. Go ahead and double click your new WeiDU install (Setup-John.exe) and run the install. If you run into trouble, check back and make sure you coded everything correctly. If you still get lost, check out the Troubleshooting section. Otherwise, let's move on!

Section 3- Making the NPC: Intermediate

Okay, so now you have a working NPC in the game. Time to expand on that. Now we're going to make your NPC banter with your PC, as well as other NPCs. We will also make your NPC interject when others are talking, making them more realistic, as well as give them a Tree of Life talk (the cheesy talk at the end of SoA). We will also explore player initiated talks.

Let's begin...again.

3.1 Banter with another NPC

Time to make your NPC a chatterbox. We are going to put to use the concept of globals and a new concept: CHAIN.

CHAIN is a style of dialogue writing that allows, in my opinion anyway, an easier way to code conversations between NPC and PC, NPC and NPC, or even NPC, NPC and PC, or multiple NPCs. CHAIN follows a structure which is as follows:

```
CHAIN
```

```
IF ~Conditions~ THEN "Name of file belonging to NPC starting
this dlg" "Name of Talk"
```

```
~Blah~
```

```
= "Banter file of other NPC" ~Blah~
```

```
EXIT
```

So, let's try a real talk then. Go ahead and open the text editor and make another D file called K#JohnB. Put BEGIN K#JohnB at the top. This will be John's banter file. We'll have John banter with Minsc. Now, there are different ways to decide when a banter will happen, or "fire". You can use timers, but you can also just leave it up to the games internal banter engine to fire it. We're going to do just that in this example.

```
CHAIN
```

```
IF ~InParty("K#John")
```

```
InParty("Minsc")
```

```
!ActuallyInCombat //This way they are not in combat
```

```
Global("K#JohnMinsc","GLOBAL",0) // We haven't had this banter before
```

```
THEN K#JohnB JohnMinsc1
```

```
~Minsc, how are you?~
```

```
DO ~SetGlobal(K#JohnMinsc","GLOBAL",1) // This makes sure that the banter can't happen again
```

```
== BMINSC ~Oh! Boo is a hamster!~
```

```
== K#JohnB ~Is he now?~
```

```
= ~That's cool~ // only 1 "=" is used when the same character is talking more than once.
```

```
== BMINSC ~Yes, and butt-kicking!~
```

```
EXIT
```

There you go. Go ahead and save that. In your Tp2, go down to where your Dialogue files are being installed and type this:

```
COMPILE ~John/K#JohnB.d~
```

3.2 Banter with the PC

Okay so now we want for our NPC to talk to the PC. Once again, we can use the game's banter engine for this as well as CHAIN.

```
CHAIN
```

```
IF ~InParty("K#John")
```

```
InParty(Player1)
```

```
!ActuallyInCombat()
```

```
Global("JohnPCTalk","GLOBAL",0)~
```

```
THEN K#JohnB JohnPC1
```

```
~Heya, <CHARNAME>, how goes it?~
```

```
DO ~( "JohnPCTalk" , "GLOBAL" , 1 ) ~
```

```
END
```

```
IF ~~ THEN REPLY ~I'm good, John, thanks for asking.~ GOTO  
JohnPC1-1
```

```
IF ~~ THEN REPLY ~Leave me alone.~ GOTO JohnPC1-2
```

```
// Now, we write the blocks of dialogue that these states lead  
to.
```

```
CHAIN
```

```
IF ~~ THEN K#JohnB JohnPC1-1
```

```
~No problem!~
```

```
EXIT
```

```
CHAIN
```

```
IF ~~ THEN K#JohnB JohnPC1-2
```

```
~Oh...okay.~
```

```
EXIT
```

See, now was that hard? Save your B file and reinstall the mod to make sure it installs properly. NOTE: You can see your dialogue in game by hitting CTRL + I multiple times.

In the future, a little trick that serves as an alternative to IF ~~ THEN REPLY is "++" without the quotations. For example you could do this:

```
++ ~Hi!~ + ATalk
```

That last "+" serves as the GOTO command. Please only use this if you are comfortable with

dialogue writing.

3.3 Interjections

Interjections are the little blurbs that NPCs have to say in response to someone else's dialogue.

So, open up the text editor again and create yet another D file. Call it K#JohnJ. (J means "joined") and put BEGIN K#JohnJ at the top. Now, here is where I recommend having Infinity Explorer. If you open up that program, (you can do this with Near Infinity as well) you'll see a Dialogs tab. Click that and you'll see a listing of every single dialog file in the game. Let's find Nalia's file where she joins the party (It's called NALIA). We'll have John make a comment about it. You'll notice that every line of dialogue has a PHRASE label on it. Let's have John interject after Nalia says "Is anyone willing to hear my plea...?"

For this example, we are going to use a command called INTERJECT_COPY_TRANS. You will need to put the name of the dialogue file followed by the phrase number and then you need to give your interjection a name.

```
INTERJECT_COPY_TRANS NALIA 45 johncommentnaliajoin
== K#JohnJ IF ~InParty("K#John") InMyArea("K#John")~ THEN
~Hey, look, a rich girl!..~
END
```

Save this and put it into your Tp2 file by typing:

```
COMPILE ~John/K#JohnJ.d~
```

Just save the Tp2 and reinstall to make sure it works correctly.

3.4 Player-Initiated Dialogues

You're doing well. Now we are going to let your PC start the talking for once. This is actually going to be a two part section. The first will discuss basic Player Initiated Dialogue, the second will cover the "flirt" style section, but that won't come along until our Romance section.

Now, open your J file and find a nice open space. You can go ahead and start right under your interjections!

In order to make John speak after the PC clicks on him for dialogue, we must use the following trigger.

```
IF ~IsGabber(Player1)~ THEN K#John PID1
```

You can put other conditions in there, too, such as romance variables, or a variable that a quest has been completed, etc.

As you can see, we will be using CHAIN for this. Here is an example of a simple PID.

```
CHAIN
```

```
IF ~IsGabber(Player1)~
```

```
THEN K#JohnJ PID1
```

```
~(John sees you looking at him strangely.)~
```

```
= ~What's up, <CHARNAME>?~
```

```
END
```

```
IF ~~ THEN REPLY ~Just seeing how you were.~ GOTO PID1-1
```

```
IF ~~ THEN REPLY ~You're a nice guy, John!~ GOTO PID1-2
```

```
CHAIN
```

```
IF ~~ THEN K#JohnJ PID1-1
```

```
~I'm fine! Thanks for asking!~
```

```
EXIT
```

```
CHAIN
```

```
IF ~~ THEN K#JohnJ PID1-2
```

```
~Why, thank you, <CHARNAME!~
```

```
EXIT
```

Notice that we do not set a global set to 1 in this. This is because we want for the player to be to ask John how he is whenever they want. You can also set conditions for player responses. For instance, I want for the player to be able to ask John something ONLY when they are in the Underdark.

```
IF ~Global("Chapter", "GLOBAL", 5)~ THEN REPLY ~Hi!~ GOTO PID1-3
```

From there you would have an appropriate block it would link to. Easy enough? I don't think I have to remind you to save your file.

3.5 Tree of Life talk

Okay, time for that extremely cheesy party at the end of SoA. Yep, right before you go to the Tree of Life to face Irenicus, and all the NPCs take their turns letting you know that they will stay with you until the end!

This is actually really easy! Open your J file, find a nice spot right at the bottom of your interjections, since technically, this is sort of an interjection (Your NPC is interjecting the narrator hehe)

```
INTERJECT PLAYER1 33 TreeofLifeJohn
```

```
== PLAYER1 IF ~IsValidForPartyDialog("K#John")~ THEN ~John:the really nice guy that has been at your side for 3 whole sections!~
```

```
END
```

```
++ ~John, don't do it. Don't risk your life for me!~ EXTERN K#JohnJ Option1
```

```
++ ~John, you're my buddy and I don't want anything to happen to you. Get out while you can.~ EXTERN K#JohnJ Option2
```

```
++ ~John, are you ready to rumble?!~ EXTERN K#JohnJ Option3
```

```
// And John replies...
```

```
APPEND K#JohnJ
```

```
IF ~~ THEN BEGIN Option1
```

```
SAY ~<CHARNAME>, I'm in!~ COPY_TRANS PLAYER1 33
```

```
END
```

```
IF ~~ THEN BEGIN Option2
```

```
SAY ~I hate sappy parts. Let's get this over with!~ COPY_TRANS  
PLAYER1 33
```

```
END
```

```
IF ~~ THEN BEGIN Option3
```

```
SAY ~Let's go!!!~ COPY_TRANS PLAYER1 33
```

```
END
```

```
END
```

See, that was easy! Of course, you can sub out the ++s for IF ~~ THEN REPLY but it's not necessary.

Save your file (duh) and let's get on to the good stuff!

Section 4- Making the NPC: Advanced

Woohoo! Time for some fun. This section is only for those who are either already competent with the stuff in sections 1-3 or have gone through 1-3 and want to try something new. We're going to cover Timers, making a Romance/Friendship, giving your NPC a Quest, as well as how to make your NPC ready for ToB and how to do an Epilogue. Note that in this section, it is highly recommended you use the [IESDP](#) to find new trigger conditions and actions for the game to perform, like starting combat or making the party rest. Now, if you're not at this level yet, or you feel like you need more time to work on the other sections, don't feel pressured to do this part. Otherwise, let's go!

4.1 Timers

Timers are great because you can use them to set off anything in your mod. We're going to use timers to set off banters. Timers can be set to go off in Game Days (time that passes in the game) or in Real Time (measured in seconds that you are actually playing the game). We're going to do both.

So, we want for John to have a conversation with the PC three game days after he joined the party. So, here is what we do.

Open your text editor and save it as K#John.baf. This is a script file. In it, type the following.

```
IF
    Global("JohnJoined", "LOCALS", 1)
Global("JohnPCTalk", "GLOBAL", 0)
THEN
RESPONSE #100 //happens 100 percents of the time
SetGlobal("JohnPCTalk", "GLOBAL", 1)
SetGlobalTimer("K#JohnTimer", "GLOBAL", THREE_DAYS)
END
```

If we want to use a Real Time timer, we replace SetGlobalTimer with RealSetGlobalTimer, and instead of THREE_DAYS, we put the amount of time in seconds, like...10 or something. Now for the part that will set the banter off.

```
IF
    GlobalTimerExpired("K#JohnTimer", "GLOBAL") // or RealGlobal
Global("JohnPCTalk", "GLOBAL", 1)
THEN
RESPONSE #100
SetGlobal("JohnPCTalk", "GLOBAL", 1)
Interact(Player1) //makes John talk to the PC
```

END

Save this. In your Tp2, you should make yourself a Scripts section if you haven't already (//Scripts). Make sure that this and any additional sections you add to the Tp2 are ABOVE the 2DA appending sections. Add this:

```
COMPILE ~John/K#John.baf~
```

Now, in your B file...let's set up the trigger so the timer will set the talk off properly.

CHAIN

```
IF ~InParty("K#John")
```

```
InParty(Player1)
```

```
!ActuallyInCombat()
```

```
Global("JohnPCTalk","GLOBAL",2)~
```

```
THEN K#JohnB JohnPC1
```

```
~Heya, <CHARNAME>, how goes it?~
```

```
DO ~( "JohnPCTalk","GLOBAL",3)~
```

```
END
```

```
IF ~~ THEN REPLY ~I'm good, John, thanks for asking.~ GOTO  
JohnPC1-1
```

```
IF ~~ THEN REPLY ~Leave me alone.~ GOTO JohnPC1-2
```

CHAIN

```
IF ~~ THEN K#JohnB JohnPC1-1
```

```
~No problem!~
```

```
EXIT
```

```
CHAIN
```

```
IF ~~ THEN K#JohnB JohnPC1-2
```

```
~Oh...okay.~
```

```
EXIT
```

Yay! Good job. After three games days from when John joined, he'll start talking to your PC. This of course can be expanded on. You can set the timer again in the banter, right after you set the talk global to 2, actually:

```
DO ~("JohnPCTalk", "GLOBAL", 3) SetGlobalTimer("K#JohnTimer", "GLOBAL", TWO_DAYS)~
```

Very nice.

4.2 Romance/Friendship Tracks

We all love to try and woo the NPCs in our party, or befriend them. It's actually quite simple, and involves making sure that certain globals are set during the talks.

Whether it be a romance or a friendship, they work the same way. You're going to be relying on a few globals for each. For a romance, you'll have the K#JohnRomanceActive global, which goes from 1 to 3: 1 being the flirty, getting to know you stage, 2 being a committed romance, and 3 meaning that the romance was terminated for one reason or another (the PC was mean, etc). The other global is the K#JohnTalkRomance. This will keep track of all your "Lovetalks" as they are called. For friendships, it's similar, except you will only need a K#JohnFriendshipActive 1 and 2...1 means you are friends with John and 2 means that you aren't, and a K#JohnTalkFriend global...which keeps track of all of your Friendship Talks.

Now, these globals can be set anywhere globals can be set, but you must be sure to stay consistent with them. For instance, if one PC choice leads to a block of dialogue in which the romanceactive variable gets set to 2, there should be another one that sets it to 3 to make sure that the PC has to act a certain way to win the NPC's affection. You can also set conditions for which the PC must meet requirements to romance the NPC, such as being a certain race, or having a certain charisma, etc.

There is no rule to writing romances other than to stay on top of your global setting. My recommendation would be to use timers to space out the talks so they aren't just fired at the mercy of the game's banter engine. You can make your very own file for you Lovetalks and Friendship Talks, but I prefer to keep them in my B file. Also, download and take a look at romance mods others have done. Find ones that haven't been translated so that the files are easy to look through.

Now for flirts. This is the second part of the PID section. This time, we are going to randomize some of our answers, so that John doesn't always say the same thing when you say something to him.

CHAIN

IF ~IsGabber(Player1~

THEN K#JohnJ JFlirt1

~(John notices you looking at him)~

= ~What's up, <CHARNAME> ?~

END

+~ RandomNum(2,1)~ + ~It is wonderful having you in the group, John.~
GOTO JFlirt1.1

+~ RandomNum(2,2)~ + ~It is wonderful having you in the group, John.~
GOTO JFlirt1.2

What the heck? Random what? "RandomNum" will mix up your replies so that the same thing doesn't always result when you ask John something. The first number is the number of total choices, in this case, 2. The second number represents the possibility of that reply being chosen. In this case, it should be 50/50, but 1 will always have a slightly higher chance than 2 of being picked. So we will see response JFlirt1.1 more often.

CHAIN

IF ~~ THEN K#JohnJ JFlirt1.1

~Why, thank you, <CHARNAME>. It is nice traveling with you as well.~

EXIT

CHAIN

IF ~~ THEN K#JohnJ JFlirt1.2

~Heh, thanks.~

EXIT

Nice job.

4.3 Give your NPC a Quest

Quests are fun. There is only so much advice I can give you here. Quests are a combination of dialogue, setting globals, and other scripting...depending on the complexity of the quest you want to do. Always have a global that keeps track of the quest, like a Global("K#JohnQuest","GLOBAL",0) which gets set to 1 under certain conditions, then 2, etc. You can add new NPC's (non joinable), new items, new areas, anything really.

4.4 Making your NPC for ToB

This part is simple really. You just want to make sure that all your files are named properly for the ToB portion of it (You **are** doing the ToB portion right?)

ToB uses all new dialogue and script files. You probably saw them in the Tp2 we made. They all have the number 25 in it. Your main script will be K#John25.baf. These are the dialogue files:

```
K#Joh25P K#Joh25J K#Joh25D K#John25 K#JohB25
```

However, your NPC's DV will always remain the same, as will globals that have been set in SoA, so don't worry about that. Just copy and compile the files (once you've gotten started on the ToB portion) as we've done with the other ones.

4.5 Epilogues

Aren't these why we play through ToB? Well, they are for me anyway.

Open up a text file and call it BlankEpilogue.2DA. Put the following text in it:

```
2DA V1.0
```

```
*PORTRAIT
```

	0	1
SWITCH	DEFAULT	DEFAULT
DEFAULT	71020	99999
DWARF	0	0
ELF	0	0

HALFELF	0	0
HALFLING	0	0
GNOME	0	0
HUMAN	0	0
HALFORC	0	0
MAGE	0	0
FIGHTER	0	0
CLERIC	0	0
THIEF	0	0
BARD	0	0
PALADIN	0	0
FIGHTER_MAGE	0	0
FIGHTER_CLERIC	0	0
FIGHTER_THIEF	0	0
FIGHTER_MAGE_THIEF	0	0
DRUID	0	0
RANGER	0	0
MAGE_THIEF	0	0
CLERIC_MAGE	0	0
CLERIC_THIEF	0	0
FIGHTER_DRUID	0	0
FIGHTER_MAGE_CLERIC	0	0
CLERIC_RANGER	0	0

Now create another BAF file and call it AR6200. Put the following in it:

IF

```
Global("StartEndBios", "AR6200", 1)
```

```
InParty( "K#John" )
Global( "K#JohnBio" , "GLOBAL" , 0 )
```

THEN

```
RESPONSE #100
    SetGlobal( "K#JohnBio" , "GLOBAL" , 1 )
    TextScreen( "K#JohnD" )
    SmallWait( 1 )
    Continue( )
```

END

This will set the game up to play John's epilogue. Now, in your Tp2, go ahead and put this under your Scripts section, or even where you put John's spawning area file.

```
EXTEND_TOP ~ar6200.bcs~ ~John/AR6200.baf~
```

You can make a new section if you like called Epilogues. Put the following:

```
COPY ~John/BlankEpilogue.2da~ ~override/K#JohnD.2da~
```

```
    REPLACE_TEXTUALLY ~PORTRAIT~ ~JohnL~ // The biggest portrait
file you have for the NPC
```

```
    REPLACE 99999 ~The Epilogue itself~
```

Don't worry about formatting the epilogue or anything. Just type straight through without using the Enter/Return button and the game will automatically format it for the epilogue.

Okay, save the Tp2! In the future, you may want to consider organizing your mod folder into subfolders to avoid clutter.

Good job! You now have a fairly developed NPC in the game. You can add on to these skills you've learned by writing more talks, creating a quest or two, or even making a romance for your NPC. It's up

to you. The next section focuses on multimedia, which is not required but it is still nice to have.

Section 5: Multimedia

Now that we have a nicely developed NPC in the game, time to add some accessories. Note that while these things aren't required, they are very nice to have.

5.1 Portraits

I can't draw well at all. But there are many who can! Sure, you can use one of the Bioware NPC portraits as a portrait for your NPC, but if your NPC is your very own custom one, you may prefer to have a custom portrait. There are many who will happily make you a beautiful portrait for your NPC for no cost at all.

Portraits must be in bitmap (.bmp) format, as this is the only format the game will take. You will need a Small(38x60px), Medium(110x170px), and if you are modding for ToB, a Large(210x330px) portrait.

Once you have those, open your Tp2 and type the following:

```
COPY ~John/johns.bmp~ ~override/johns.bmp~
```

```
~John/johnm.bmp~ ~override/johnm.bmp~
```

```
~John/john1.bmp~ ~override/john1.bmp~
```

Good. Now you have portraits in your mod. Next, Sound and Music.

5.2 Sound and Music

Adding voicing and music adds more feeling a realism to your NPC. Voice actors can be found all over the community, as there are many people who are happy to do it for free (this is a modding community: everything should be free). Same with music. You can either find a composer in the community who will either let you use their existing music for your mod or will make an entirely new one just for you. A few things about sounds.

- The game reads .acms and .wavs only
- Use .oggs to compress the .wavs to make download sizes smaller
- Make an ogg decoder to decompress the .oggs into .wavs so the game can play your files

This is not difficult to do. You are going to need a blank MUS file. Simply open up your text editor and make a file called K#Blank.MUS. Put the following in it.

Blank

1

A @TAG END

Good. Now save it. You should have your ogg sounds ready to go in an audio subfolder of your mod folder. Now, you need to use a specific naming system when naming oggs...otherwise you'll give yourself a whole lot of work to do. For instance, all of my oggs are named K#John**. The ** represent 2 character spaces where you can put letters, numbers, or symbols to label that particular sound file.

Now, open your text editor. Name it "johnaudio.bat". Inside, put this:

```
@echo off
```

```
@echo Installing audio...
```

```
cd override
```

```
oggdec K#John*.ogg
```

```
del K#John*.ogg
```

```
del oggdec.exe
```

```
@echo Finished installing audio.
```

Save this. Now, make another .bat file called "johnaudiouninstall". Put this in it:

```
@echo off
```

```
@echo Uninstalling audio...
```

```
cd override
```

```
del K#John*.wav
```

```
@echo Finished uninstalling audio.
```

Save that, too. Now, in your Tp2, put this someplace above the 2DA appending:

```
COPY ~John/audio/K#blank.mus~ ~music~
```

```
COPY_EXISTING ~songlist.2da~ ~override~
```

```
SET_2DA_ENTRY 0 2 3 ~K#BLANK.MUS~
```

```
ACTION_IF ("%WEIDU_OS%" STRING_COMPARE_CASE ~WIN32~ = 0) THEN  
BEGIN // if Windows
```

```
    COPY ~John/audio~ ~override~
```

```
    AT_NOW ~John/johnaudio.bat~
```

```
AT_INTERACTIVE_UNINSTALL ~John/johnaudiouninstall.bat~  
END ELSE BEGIN // if OS X or Linux  
    AT_NOW ~sh John/johnaudio.sh~  
    AT_UNINSTALL ~sh John/johnaudiouninstall.sh~  
END
```

This will make your mod sounds compatible with both Windows and OSX/Linux operating systems.

5.3 Adding Sounds/Voicing into your mod

To make any voice sound play in a mod, simply go to the line that you want voiced. At the end of the line, right after the punctuation, type the following:

```
[K#John01]
```

This is assuming you have a sound file called K#John01. Yours may be different. After you've installed the mod, the sound should play when that line is shown.

5.4 Adding Music into your mod

Adding music is a bit different. You must place it as an action in a dialogue (you know, DO ~blah~) or as a action in a script (THEN RESPONSE #**)

```
PlaySong(0) // Makes ingame music shut up
```

```
PlaySound("K#JohnM1") //plays your song
```

Now that we have pretty pictures and sounds in our mod, let's get on to troubleshooting when stuff *doesn't* work the way we want it to.

Section 6: Troubleshooting

Okay, you're installing your mod and you hit the I or Y to install it and all of a sudden, WeiDU explodes...erm...it says you have an error and the mod will not install.

6.1 Things to Check For

Did you forget a "~"? Typos of CHAIN anywhere (CHAN happens to me a lot)? Broken links in dialogues?

Normally, you'll get what's called a syntax error, which means that there is something typed wrong in the code. WeiDU will also give you a line number and some text that the error is near. Typically, the actual error is about 5 lines above from where the error is said to be. If you find the error, fix it, save the file, and reinstall.

If you are stuck and you've checked over everything, post your problem in a Q and A forum on any of the various IE modding sites (please, post it only once!).

Summary

Well done on making it this far. You'll find it helpful to check back through this guide for most questions you might have. Remember though that this is just a guide...not a mod making machine. You have to put the effort and creativity into your NPC mod to make it happen. Just remember that it's **your** mod, and while others can give their input on it, everything about your NPC is ultimately up to you.

Have fun modding!

Links

Here you can find various IE Modding related links.

Sites:

[The Gibberlings 3](#)

[Spellhold Studios](#)

[Pocket Plane Group](#)

[Chosen of Mystra](#)

[The Black Wyrms Lair](#)

[TeamBG](#)

[PPG Modlist](#)

[WeiDU](#)

[Bioware](#) (duh)

