

Tutorial: Creating an item with WeiDU

This tutorial shows how to create a very simple mod that adds an item to a game. This mod creates a sword +5.

What you need:

1. An installed copy of Baldur's Gate 2.
2. The NearInfinity program.
3. A text editor. You can use Notepad, but I recommend Notepad++ with Weidu syntax highlighters.
4. The latest version of the WeiDU executable.

Preparing the system

- Create a new game installation to do your modding and rename the main folder to `Baldur's Gate II - My modding`. This will prevent you from messing up the installation you use for playing. Enable the debug mode in this installation.
- Download the latest version of NearInfinity to the main game folder of this modding installation.
- Download the latest version of `weidu.exe` to the main game folder of this modding installation. Rename `weidu.exe` to `setup-MagicSword.exe`. This is the program that will take care of the internal details of installing the mod.
- In the main game folder create a folder named `MagicSword: <game_path>\Baldur's Gate II - My modding\MagicSword`. Inside this folder create another folder called **English**: `<game_path>\Baldur's Gate II - My modding\MagicSword\English`.

Creating the sword file

Infinity Engine games store their items in item files. The easiest way to create a new item is to make a copy of a similar item that is already present in the game and modify it. In this tutorial you'll use the file for swords +1 as the base item to create a sword +5:

1. Open Near Infinity.
2. The panel on the left displays a list of folders. Open the folder named ITM.
3. A very long list of files is displayed. Each of these files corresponds to a game item. Click `SW1H05.itm`. This is the file for long swords +1.
4. On the program menu, click `File` → `Export` and export this file to the `MagicSword` folder (that you previously created in your modding installation) with the name `MagicSword.itm`.
5. Click `Game` → `Open file` and open the file you just created, `MagicSword.itm`.
6. In the Abilities panel (bottom right of the screen) make sure `Melee (1)` is selected and click `View/Edit`.
7. Select the `Edit` tab.
8. Set `Bonus to hit` and `Damage bonus` to 5.
9. Close this window, close the main item window, click `Save changes`, `Overwrite`, `OK`.

Preparing the text for the sword

Normal items in the IE games have a name and a description. Magic items, however have two names and two descriptions: The name and description displayed before they are identified, and the name and description displayed after they are identified. Each of these (unidentified name, identified name, unidentified description and identified description) are pieces of text,

also called strings, that you need to create. These strings are not stored in the item file, instead they are stored in a special text file called language file.

Each string in a language file is identified by a number called string identifier. The syntax of each line in a language file is:

```
string_identifier = ~string~
```

This means that each line contains a string identifier followed by an equals sign (=) and a string enclosed in ~ characters. The syntax of string identifiers is:

```
@#
```

I. e., they are an @ character followed by a number.

For example, the line that stores the unidentified name of a magic sword, looks like this:

```
@1 = ~Long Sword~
```

As stated previously, you need four strings for the sword item: unidentified name, identified name, unidentified description, and identified description. Long strings, like the description strings, are broken into several lines:

```
@1 = ~Long Sword~
```

```
@2 = ~Sword +5~
```

```
@3 = ~These swords are usually referred to as doubled-edged swords, war swords, or military swords. In many cases, the long sword has a single-edged blade. There is no single version of the long sword; the design and length vary from culture to culture, and may vary within the same culture depending on the era. Among the most common characteristics of all long swords is their length, which ranges from 35 inches to 47 inches. In the latter case, the blade is known to take up 40 inches of the total length. Most long swords have a double-edged blade and a sharp point at the tip. Despite the tip, the long sword is designed for slashing, not thrusting.
```

```
STATISTICS:
```

```
Damage: 1D8
```

```
Damage type: slashing
```

```
Weight: 4
```

```
Speed Factor: 5
```

```
Proficiency Type: Long Sword
```

```
Type: 1-handed
```

```
Requires: 6 Strength
```

```
Not Usable By:
```

```
Druid
```

```
Cleric
```

```
Mage~
```

```
@4 = ~Sword +5
```

Swords +5 are some of the most powerful swords you can find in the realms.

```
STATISTICS:
```

THAC0: +5 bonus
Damage: 1D8 +5
Damage type: Slashing
Weight: 2
Speed Factor: 2
Proficiency Type: Long Sword
Type: 1-handed
Requires: 11 Strength
Not usable by:
Druid
Mage
Beast Master~

In this mod you'll use a language file called `english.tra`.

Steps:

1. Create a new text file in your text editor and copy to that file these four strings with their string identifiers.
2. Save this file as `english.tra` in `<game_path>\Baldur's Gate II - My modding\MagicSword\English`.

Main mod file

The core of a mod is a tp2 file. A tp2 file is a little program that tells the computer what to do with the other files in the mod, in this case the files `english.tra` and `MagicSword.itm` that you created in the previous sections. tp2 files are text files that contain a list of commands for the computer. These lists of commands are called programming code, or simply code. Code is written in a programming language called WeiDU. WeiDU is the programming language used in Infinity Engine mods. The WeiDU language is quite complex. This tutorial doesn't explain how to use it, it just shows a little example of what it can do.

Below are the contents of the tp2 file that you'll use for your mod. The lines that start with two forward slashes are not actual commands, they are comments that explain what the actual commands do. For this tutorial I've divided the tp2 file in three parts. Each part starts with a comment like this: `//Part 1: Header`. Part 1 is a header that contains general setup for the mod. Part 2 assigns the string identifiers you created in `english.tra` to the sword. Part 3 puts the sword in the game, giving it to an otyugh (a type of monster):

```
//Part 1: Header  
BACKUP ~MagicSword/backup~  
AUTHOR ~myemail~  
LANGUAGE ~English~ ~MagicSword/english~ ~MagicSword/english/english.tra~  
BEGIN ~MagicSword~  
  
//Part 2: Add the sword to the game)
```

```
COPY ~MagicSword/MagicSword.itm~ ~override/MagicSword.itm~  
  
SAY NAME1 @1  
  
SAY NAME2 @2  
  
SAY UNIDENTIFIED_DESC @3  
  
SAY DESC @4
```

//Part 3: Give the sword to the otyugh in Irenicus' dungeon

```
COPY_EXISTING ~IOTYUGH.cre~ ~override/IOTYUGH.cre~  
  
ADD_CRE_ITEM ~MagicSword~ #0 #0 #0 ~NONE~ ~inv11~
```

Steps:

1. Create a new file with this programming code.
2. Save it as `MagicSword.tp2` in `<game_path>\Baldur's Gate II - My modding\MagicSword.`

Testing

To test that your mod works follow these steps:

1. Run `setup-MagicSword.exe` (the file you created at the beginning).
2. Start a new game in your modding installation. Use the console to move your character to the otyugh room in Irenicus' dungeon. Kill the otyugh and... Voila! There is your magic sword +5.

Packaging

If you want to distribute your mod, make a zip archive that contains the file `setup-MagicSword.exe` and the `MagicSword` folder.