

## BG2Script

Version 1.11 1999-2000.  
Brent Knowles: [brent@bioware.com](mailto:brent@bioware.com).  
[knowledge.bizhosting.com](http://knowledge.bizhosting.com)

### Overview

BG2Script is intended as a supplement to assist in writing scripts for Baldur's Gate 2: Shadows of Amn(BG2). This program is freeware and may be distributed freely.

BG2Script highlights the different structures of the BG2 scripting in different colors to make it easier to view/edit the scripts.

### Writing Scripts

The best way to learn how to write scripts for BG2 is to look at examples of them. Included with BG2 are several player scripts, each of which highlights different aspects of the BG2 scripting language.

#### For Example:

```
IF
    See(NearestEnemyOf(Myself))
    HaveSpell(WIZARD_MAGIC_MISSILE)
THEN
    RESPONSE #100
    Spell(Myself,WIZARD_MAGIC_MISSILE)
END
```

Which follows the format

```
IF
    <TRIGGER>
THEN
    RESPONSE #100
    <ACTION>
END
```

Where **trigger** is a collection of triggers that must all be true for the **action** to happen.

There is a new type of trigger in BG2. It is the OR(x) trigger where **x** is the number of conditions that the or will check. If *any* of them are valid then the <trigger> will be valid.

#### For Example:

```
IF
    !Dead(Player2)
    OR(3)
    See(NearestEnemyOf(Myself))
    See(Player2)
    See([0.ANIMAL])
THEN
    RESPONSE #100
```

Attack(NearestEnemyOf(Myself))

END

This would check to see that Player 2 is alive and ONE of the following is true (you can see your nearest enemy, you can see player 2, and you can see an animal).

## Compiling Scripts

When you are satisfied with the script you've written you may compile it by clicking on the red checkmark. If the compile is successful BG2Script will tell you. If there are errors the footer will display the error message.

Doubleclicking on the footer displays a text file listing all the possible errors.

NOTE: Once you click compile the script is automatically saved.

### *Types of Errors*

1. Pre-compile errors: BGScript2 uses a primitive bracket matching system to make sure you have enough brackets. If you see this error it means that on the given line the brackets (square or round) do not match. Correct and recompile.
2. Compiler errors: read the error file (double-click on footer) and correct the problem. Recompile

## Adding Actions/Triggers/Objects to the Script

### *Actions*

The rightmost drop-down displays a list of Actions by default. Select an action from the list and enter the appropriate parameters.

### **For Example:**

Select

```
AttackReevaluate(O:Target*,I:ReevaluationPeriod*)
```

You would then filled in the Target and ReevaluationPeriod like so:

```
AttackReevaluate(NearestEnemyOf(Myself),15)
```

### *Triggers*

To select a trigger follow the steps above for actions with one difference. You have to select the trigger list. Go to **Reference|Trigger**. Now the drop-down will display the trigger list.

### *Objects*

As triggers, but select **Reference|Object**.

## Using the Editor

There are some special features of the editor you may want to use:

1. Auto-date: press Ctrl +D to insert the current date at the cursor position
2. Any text enclosed in \*<text>\* becomes a *Bookmark*. It will appear in the tab bar at the bottom of the BG2Script interface. Clicking on the *Bookmark* will jump you to the Bookmark position in the file.
3. Highlighting a section of scripting and pressing **F6** will automatically comment out the code (meaning the compiler will ignore the text. Comments are useful for disabling certain sections of code or for adding comment to explain what you attempted in the code). Pressing **F7** over highlighted text to uncomment code.

4. If you doubleclick an action or trigger a messagebox will appear telling you whether it is valid or not and what its parameters should be.

### **Adding Functions and Modifying Functions**

By pressing CTRL+A you can bring up a menu where you may choose either a function or a modification to (use the drop down list). For a function enter a line of text to describe the function and then click OK.

This will create an auto-generated framework displaying the basic function. Where it says F# [ ] you should enter a number for the function, which helps with debugging the scripts and recording modifications.

When adding a modification the first line of text on the form will be inserted at the cursor position. The second line appears at the end of the script and should detail your changes in greater detail.

### **Adding Spells to the Script**

The left-most drop-down is a list of mage spells. Simply select the spell you want the 'code' to implement the spell will be placed wherever your cursor is located.

The next drop-down is a list of priest/druid spells.

Clicking the open-book icon beside the spell drop-downs will open some information files about the spell. These determine what spells appear in the lists and what kind of spells they are (target myself or target enemies).

Mage/cleric.txt is a list of spells, grouped by level.

Mage/cleric.ini is a list of spells and their in-game reference tag. Also it contains information detailing what target is acceptable.

#### **For example:**

Hold Undead=P

The P means that the spell will use the **spellpoint** template (see templates, below). M (which is the default selection) will choose the **spell** template.

Stun means **StunHeld** template and MagAtt means **magicattack** template, S means **semagecleric** template (neither of which exist yet).

### **Reference**

As explained above the Reference menu can be used to assign particular lists to the script-helper drop-down. You can also select the more option which brings up a list of IDS files (data files used in BG2 to help with scripting).

You'll seldom need to use these lists with most of the common scripting commands. However you may need to use this as a reference if using a difficult command.

For example the CheckStat(O:Object\*,I:Value\*,I:StatNum\*Stats) command requires as the third parameter a Stat object.

Go to **Reference|More** and on the left-hand side of form select the *Stats* file. On the right hand side select the stat you want to check, let's say constitution or CON. Double-click on CON. You will see the CON stat added to your script at your cursor position.

A valid stat check would be:

```
CheckStat(Myself,17,CON)
```

This would check to see if my constitution is equal to 17, returning true if valid.

## Templates

There are two built-in spell templates you may modify. Go to **Templates|Edit Templates** and select *spell* from the list. A sample of code will appear. You may enter anything you want in here and press **Update** to update it.

From now on when a spell is autoselected from the drop-down it will follow the format presented in the template (see the spells section). Anything in the [\*Var\*] section will be replaced by the actual spell name, when the spell is selected from the spell resref.

You may define three templates (see spells) for other spell types not previously defined.

## Troubleshooting

### 1. My script won't compile

\* Make sure that the source script (the file you are working on in BG2Script) is saved in the /source directory of the bg2 scripting program

\* The script compiler must be in a directory without spaces in its name. I.e., c:\Program Files\ScriptCompiler is wrong because of the space between Program and Files. Its best to install the bg2 scripting program in c:\scriptcompile.

### 2. BG2Script won't run properly

\* BG2Script must be installed in directory c:\scriptw

## The Archive

Under properties you may hide/view an Archive view.

If you choose to view the archive view again(after you've previously hidden it) you must delete the sizer.ini file in your script directory, and then restart Script.

You may want to use the Archive if you create a group of scripts and want to use them as a reference when creating new scripts.

---

Thanks to Gintaras Pikelis of Baltic Solutions for his ColorMemo component

---

### Disclaimer

Neither Brent Knowles, Bioware, Interplay, Black Isle, are responsible in any way for any damage this program may cause.

BG2Script copyright 1999-2000 Brent Knowles.