

String Modding

Information for people who are looking to create mods that affect Text.

- [Extracting Strings](#)
- [Creating String Mods](#)
- [String Customization](#)
- [String File Extensions](#)

Extracting Strings

See **Extracting Resources** before proceeding. You must have **gamerresources.resources** extracted.

A **string** is a sequence of characters (text) that appears literally.

Extracting Tools

BlangJsonGeneratorUI by PowerBall253 - [Download](#)

Modifies BLANG files and exports the changes as JSON.

BlangJsonGenerator by PowerBall253 - [Download](#)

Exports BLANG files into JSON, a format that can be edited with raw text editor, such as **Notepad++**.

BlangParser by proteh - [Download](#)

Directly modifies BLANG files with a simple GUI.

Currently, there are no released packages for BlangParser. You will either need to find a compiled version in the 2016+ Modding Discord server or compile it with a C# compiler.

Each BLANG file usually range from 1.5 MB to 2 MB of storage.

Finding BLANG files

- As of Update 6.66 - Rev 1.1, the highest-priority string files are located in **gamerresources_patch1\strings** from your resource output directory.
- Copy any language BLANG file of choice and paste it somewhere safe. This will be the string file that you will modify.

Extraction Instructions (BlangJsonGenerator)

It is **recommended** to create a folder containing both "**BlangJsonGenerator.exe**" and the unmodified **BLANG** file.

- Run "**BlangJsonGenerator.exe**".
- Select option **1**.
- Input the name of the **.blang** file. (ex: **english.blang**)
- Input the language name. (ex: **english**)
- Input the name of the output file. (ex: **english.json**)
- Close the executable.
- You may now edit each string value. See [Creating String Mods](#) for more information.

The output folder **must** end with **.json**

List of Resources

<u>Resource Name</u>	<u>Information</u>
english.blang	Contains strings for the English language
french.blang	Contains strings for the French language
german.blang	Contains strings for the German language
italian.blang	Contains strings for the Italian language
japanese.blang	Contains strings for the Japanese language
latin_spanish.blang	Contains strings for the Latin American Spanish language
polish.blang	Contains strings for the Polish language
portuguese.blang	Contains strings for the Portuguese language
russian.blang	Contains strings for the Russian language
simplified_chinese.blang	Contains strings for the Simplified Chinese language

spanish.blng	Contains strings for the European Spanish language
traditional_chinese.blng	Contains strings for the Traditional Chinese language

Creating String Mods

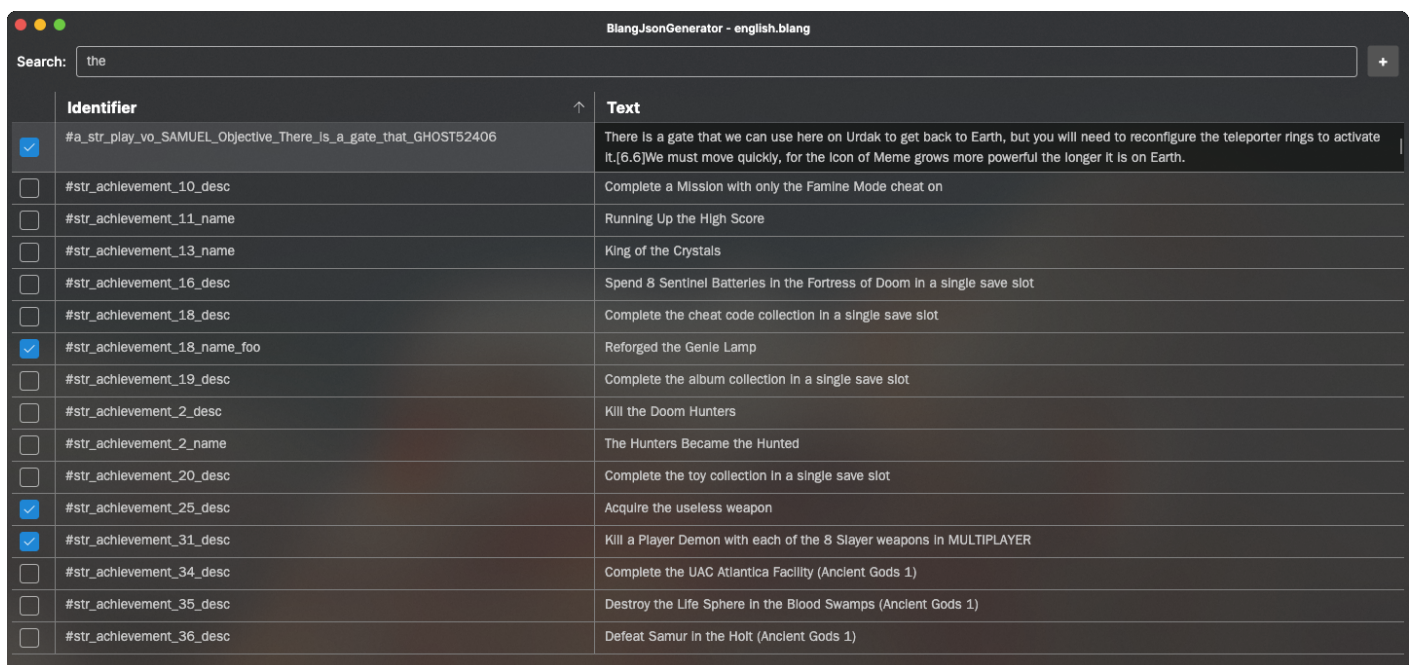
See [Extracting Strings](#) before proceeding.

Make sure to backup the unmodified **BLANG** file.

Strings can be added to **BLANG** files, but you must reference them with **decls** files.

String Editing (BlangJsonGeneratorUI)

- Extract the contents of **BlangJsonGenerator-*.zip**.
- Open **BlangJsonGenerator.exe**, then go to **File -> Open blang file...** to edit an existing BLANG file, or to **File -> Open blang from .resources file** to edit a BLANG file contained in a .resources file. You can also go to **File -> New...** to create a new BLANG JSON mod from scratch.
- You should now see something like this:



The screenshot shows the 'BlangJsonGenerator - english.blang' application window. At the top, there is a search bar with the text 'the' and a '+' button on the right. Below the search bar is a table with two columns: 'Identifier' and 'Text'. The table contains several rows of data, each with a checkbox in the left margin. The first row is selected, and its text is visible in the 'Text' column. The table is scrollable, and the bottom of the window shows a list of identifiers and their corresponding text.

	Identifier	Text
☒	#a_str_play_vo_SAMUEL_Objective_There_is_a_gate_that_GHOST52406	There is a gate that we can use here on Urdak to get back to Earth, but you will need to reconfigure the teleporter rings to activate it.[6.6]We must move quickly, for the Icon of Meme grows more powerful the longer it is on Earth.
☐	#str_achievement_10_desc	Complete a Mission with only the Famine Mode cheat on
☐	#str_achievement_11_name	Running Up the High Score
☐	#str_achievement_13_name	King of the Crystals
☐	#str_achievement_16_desc	Spend 8 Sentinel Batteries in the Fortress of Doom in a single save slot
☐	#str_achievement_18_desc	Complete the cheat code collection in a single save slot
☒	#str_achievement_18_name_foo	Reforged the Genie Lamp
☐	#str_achievement_19_desc	Complete the album collection in a single save slot
☐	#str_achievement_2_desc	Kill the Doom Hunters
☐	#str_achievement_2_name	The Hunters Became the Hunted
☐	#str_achievement_20_desc	Complete the toy collection in a single save slot
☒	#str_achievement_25_desc	Acquire the useless weapon
☒	#str_achievement_31_desc	Kill a Player Demon with each of the 8 Slayer weapons in MULTIPLAYER
☐	#str_achievement_34_desc	Complete the UAC Atlantica Facility (Ancient Gods 1)
☐	#str_achievement_35_desc	Destroy the Life Sphere in the Blood Swamps (Ancient Gods 1)
☐	#str_achievement_36_desc	Defeat Samur in the Holt (Ancient Gods 1)

- You may edit any string **Identifier** or **Text** you wish.
- To add a string, use the **+** button in the top-right corner.

- After modifying the strings go to **File** -> **Save...** -> Give it the same name as its original BLANG file, with the extension changed to JSON. (ex: **english.json**).
- Create the following folder directories in your **DOOMEternal\Mods** folder:

gameresources_patch1\EternalMod\strings

- Copy your **modified** JSON file into the strings folder.
- Run the **EternalModInjector** like normal.

String Editing (BlangJsonGenerator)

Any unused string can be **deleted**, so it can be **compatible** with other mods that affect other strings.

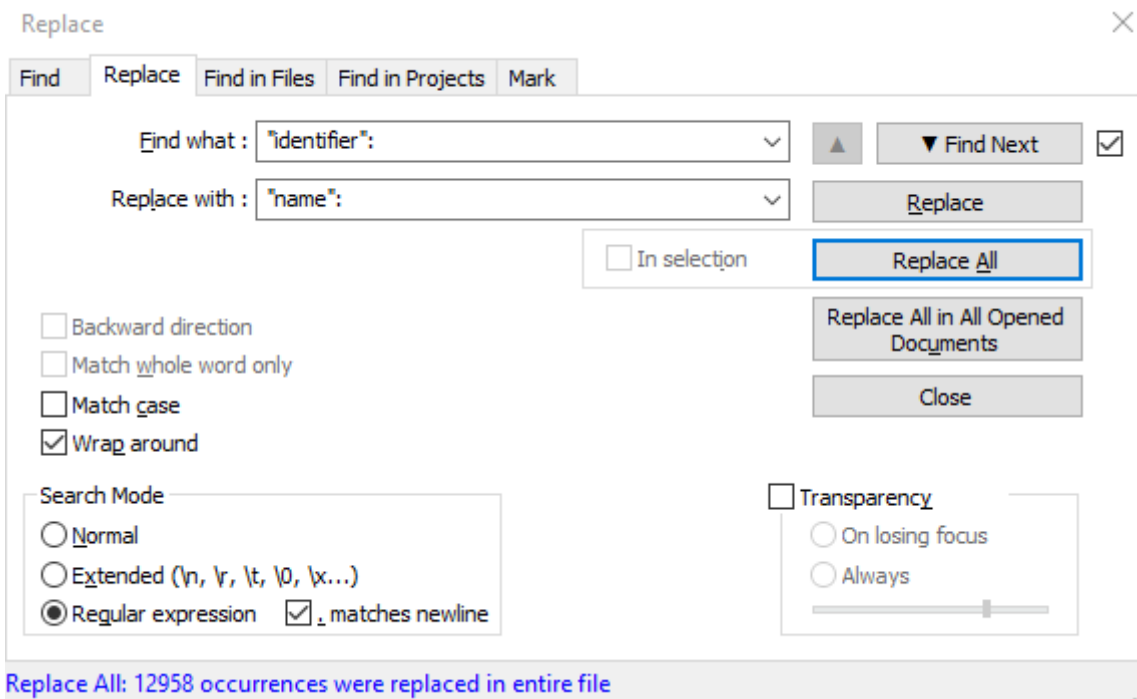
- Create a backup for the string JSON file.
- Open the string JSON file. (ex: **english.json**)
- The strings should be formatted like this:

```

1  {
2    "strings": [
3      {
4        "identifier": "#STR_SWF_ACCEPT",
5        "text": "Accept",
6        "modified": 0
7      },
8      {
9        "identifier": "#STR_SWF_CANCEL",
10       "text": "Cancel",
11       "modified": 0
12     },
13     {
14       "identifier": "#STR_SWF_CLOSE",
15       "text": "CLOSE",
16       "modified": 0
17     },

```

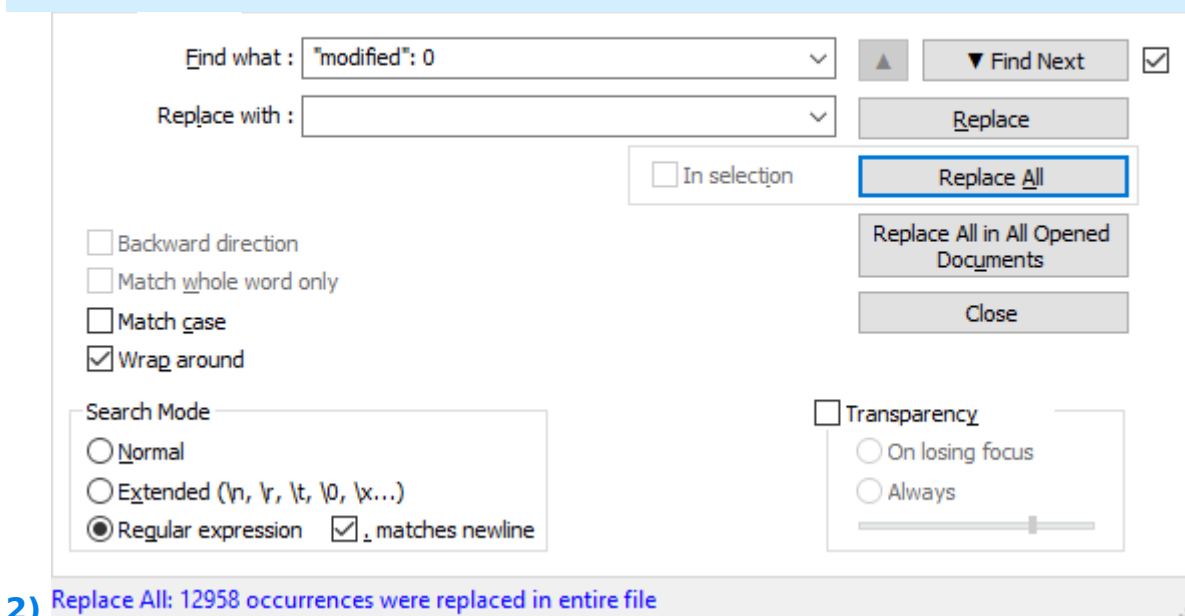
- If they are not, see the [Extracting Strings](#) page, under **Extraction Instructions: (BlangJsonGenerator)**
- Go to the Replace tab (CTRL + F) -> Replace -> Search Mode: Check **Regular expression**
- Input the following and select **Replace All** for each:



"identifier":

"name":

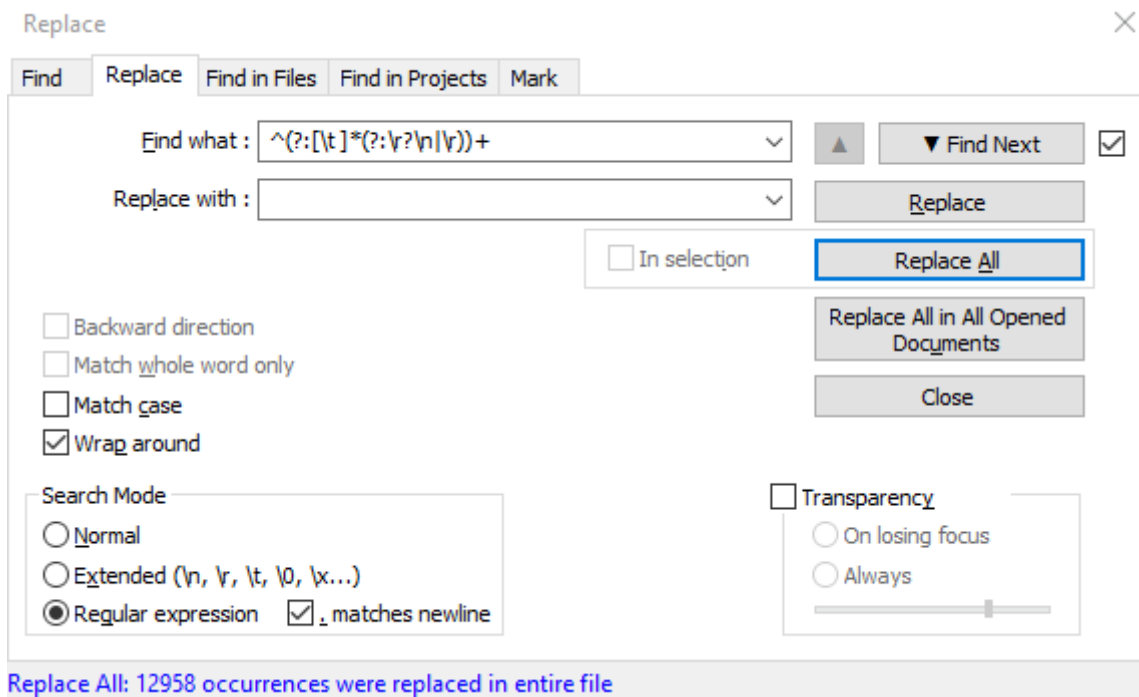
The "Replace with : " box is empty for screenshot 2.



2) Replace All: 12958 occurrences were replaced in entire file

"modified": 0

The "Replace with : " box is empty for screenshot 3.



- The results should be formatted like this:

```

1  {
2      "strings": [
3          {
4              "name": "#STR_SWF_ACCEPT",
5              "text": "Accept",
6          },
7          {
8              "name": "#STR_SWF_CANCEL",
9              "text": "Cancel",
10         },
11         {
12             "name": "#STR_SWF_CLOSE",
13             "text": "CLOSE",
14         },
15     ],
16 }

```

- The JSON file is now in a format that can be properly re-injected into DOOM Eternal. Save a backup of this JSON file as well, so you would not need to go through this process again (before the next DOOM Eternal update).
- Find the strings that you wish to modify. (CTRL + F)
- After modifying the strings of your choice, delete all other **unmodified** strings.
- Save your **modified** JSON file with the same name as it was exported. (ex: **english.json**)
- Create the following folder directories in your **DOOMEternal\Mods** folder:

gameresources_patch1\EternalMod\strings

- Copy your **modified** JSON file into the strings folder.
- Run the **EternalModInjector** like normal.

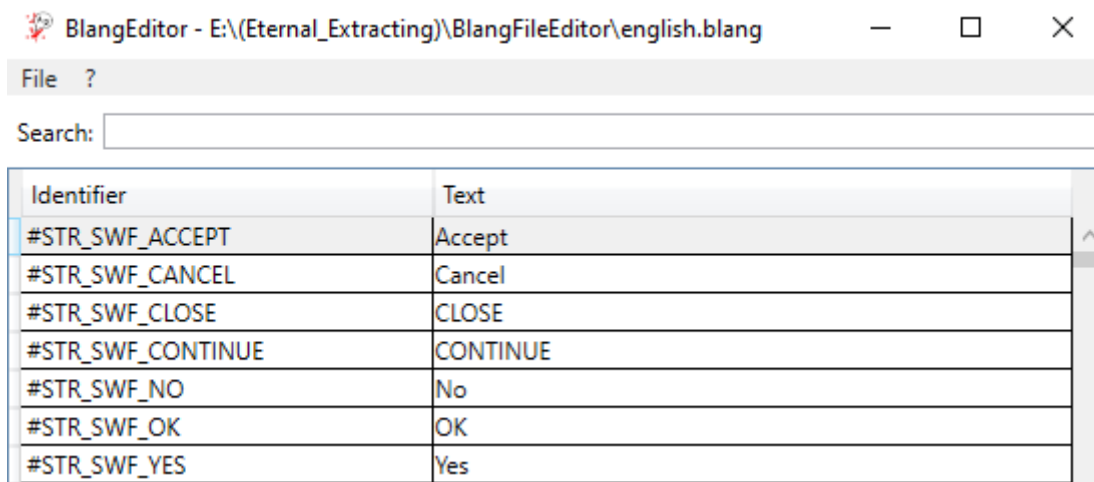
The commas "," at the end of each string name and text are unnecessary and can be deleted. However, they do not impact the syntax in any way.

Since JSON files can be edited with raw text editors,
you can execute scripts or **regular expressions** to edit strings more efficiently.

String Editing (BlangParser)

The entire **BLANG** file will be modified, which means it will be **incompatible** with other string mods.

- Extract the contents of **BlangEditorv#.#.zip**
- Open **BlangEditor.exe** you should be prompted with the following:



- You may edit any string **Identifier** or **Text** you wish.
- Upon highlighting a string line, you may select the **DELETE** key to remove it.
- To add a string, scroll down to the bottom of the list and add your own **Identifiers** and **Text** lines.
- After modifying the strings go to **File** -> **Save As...** -> Give it the same name as its original BLANG file.

(ex: **english.blang**)

- Create the following folder directories in your **DOOMEternal\Mods** folder:

gameresources_patch1\strings

- Copy your **modified** BLANG file into the strings folder.
- Run the **EternalModInjector** like normal.

It is not recommended to delete existing string lines because DOOM Eternal will fail to recognize them.

String Customization

Strings can be customized to help specific characters stand out.

String customization applies to both **BLANG** files and **idGuiEntity_Text** entities.

Available Colors

Start with ^ then a character from the list below, then end with ^7

Ex: "^5sample text^7" (without quotations) This will make the text **teal**.

^a - **Gold**
^b - **Light Blue**
^c - **Green Yellow**
^d - **Orange**
^z - **White**
^0 - **Black**
^1 - **Red**
^2 - **Green**
^3 - **Yellow**
^4 - **Blue**
^5 - **Teal**
^6 - **Purple**
^7 - **White**
^8 - **Gray**
^9 - **Black**

In general, helpful tips are colored **teal**, dangerous threats are colored **red**.

Spacing

\n - New Line
\t - Tab

Examples

^2Green Yellow^7 -> **Green Yellow**
^1Red^7 ^aGold^7 -> **Red Gold**
^1R^7^da^7^3i^7^2n^7^5b^7^bo^7^6w^7 -> **Rainbow**
This is a ^5sample^7 tip -> **This is a sample tip**

First line\nSecond line -> **First line**
Second line

Sample\tText -> **Sample** **Text**

Up to 9 separate color identifiers can be present in one string line. Any additional colorization will cause some to be ignored.

See Also

- **idGuiEntity_Text**

String File Extensions

These are the string file types you will be interacting with.

.blang

String files that **CANNOT** be read by a raw text editor.
BLANG files **CAN** be re-injected into DOOM Eternal.

Example:

english.blang

.json

Extracted string files that **CAN** be read by a raw text editor.
JSON files **CANNOT** be re-injected into DOOM Eternal.

Example:

english.json