

Index

Important Note.....	2
Detailed Guide (recommended).....	3
What is IEEE PCVI?.....	3
Key Features.....	4
Supported Soundslots.....	4
How to use IEEE PCVI.....	5
1. Download.....	5
2. Unzip.....	5
3. Verify Contents.....	6
4. Renaming Character Voice Sound Files.....	6
4.1 Base Name.....	6
4.2 Postfix.....	7
5. Moving and Organizing Sound Files.....	7
5.1 Copy Files.....	7
5.2 Automatic Distribution.....	8
6. Building the Mod.....	10
6.1 Generate WeiDU Code and Edit Voice Prompts.....	10
7. Finalizing the Mod.....	12
8. Testing your Mod.....	13
8.1 <i>Installing Your Mod</i>	13
8.2 TLK Repair (Optional).....	13
8.3 <i>When to Apply TLK Repair</i>	14

Important Note

It is recommended to use the [online guide/information](#) which should be located on the [repository wiki](#). This document should only be used in an emergency when there is no internet connection or the wiki for the official repository is missing for some reason.

This document may not be maintained or may be maintained very rarely!!

Detailed Guide (recommended)

What is IEEE PCVI?

IEEE PCVI, short for "*Infinity Engine Enhanced Edition Player-Character-Voice-Installer*", is a system designed to install pre-recorded sound files into Infinity Engine games. However, it excludes *Planescape Torment Enhanced Edition* as its main character already has a unique voice. The games which are supported, are:

- *BGEE* (Baldur's Gate Enhanced Edition)
- *SoD* (Siege of Dragonspear)
- *IWDEE* (Icewind Dale Enhanced Edition)
- *BG2EE* (Baldur's Gate 2 Enhanced Edition)



Figure1: Sound customization menus (BGEE2, BGEE, SoD, IWDEE)

Key Features

- Installs pre-made sound files for player character voices. It is capable of installing WAV and [OGG sound files](#). This renames OGG file extensions to WAV file extensions.
- Optionally scans "*dialog.tlk*" with the installed soundset and automatically repairs any incompatible "*strref*" entries, ensuring seamless integration with the selected voice pack.
- Serves as a template for creating custom player character voice sets.
- Requires minimal coding knowledge

Supported Soundslots

IEEE PCVI mainly targets soundslots used by the original Infinity Engine games in their selectable voice sets within the character customization menu.

It uses the [unofficial soundset convention](#) for advanced sound slots, ensuring compatibility with sound files developed under this standard.

If you are interested installing more voice lines for the player character, an alternative system is the [soundset installation tool](#).

The supported sound slots as a list for the individual games are marked with Postfix and their meaning can be found in the “*guides*” folder or online:

Postfixes	
b	l_
c	i
d	j
a	k
8	0_
9	1_
g_	2_
h_	3_
l	w
m	z
e	1
n	2
o	3
p	4
q	5
r	6
f	7
g	s
h	t
0	u
k_	v

Online-Links to postfix tables:

BG:EE (Baldur's Gate:Enhanced Edition)

SoD (Siege of Dragonspear)

BG2:EE (Baldur's Gate 2:Enhanced Edition)

IWD:EE (Icewind Dale:Enhanced Edition)

How to use IEEE PCVI

1. Download

- Get the latest [release version of IEEE PCVI](#). (See example in **Figure2**)
- You'll find it as a file named **IEEE-Player-Character-Voice-Installer--<version>.zip**.

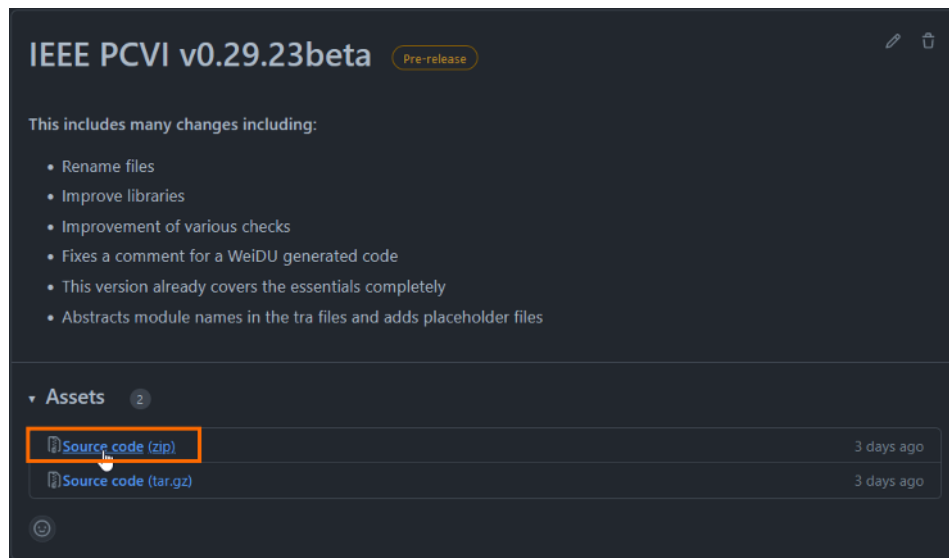


Figure2

2. Unzip

- Extract the contents of the zip file.
- Inside, you'll find a folder with the same name as the zip file (as an example, see **Figure3**).

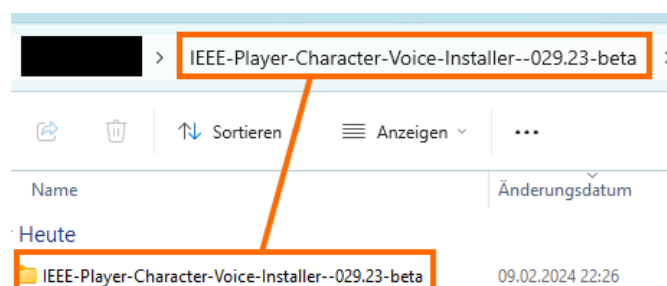


Figure3

3. Verify Contents

- Open the extracted folder. It should at least contain the files shown in **Figure 4**. These are the most important ones as IEEE PCVI would not function properly without them. If there are additional files and folders not shown in **Figure 4**, they were most likely added later and should not affect functionality. It should be safe to delete these added files and folders.

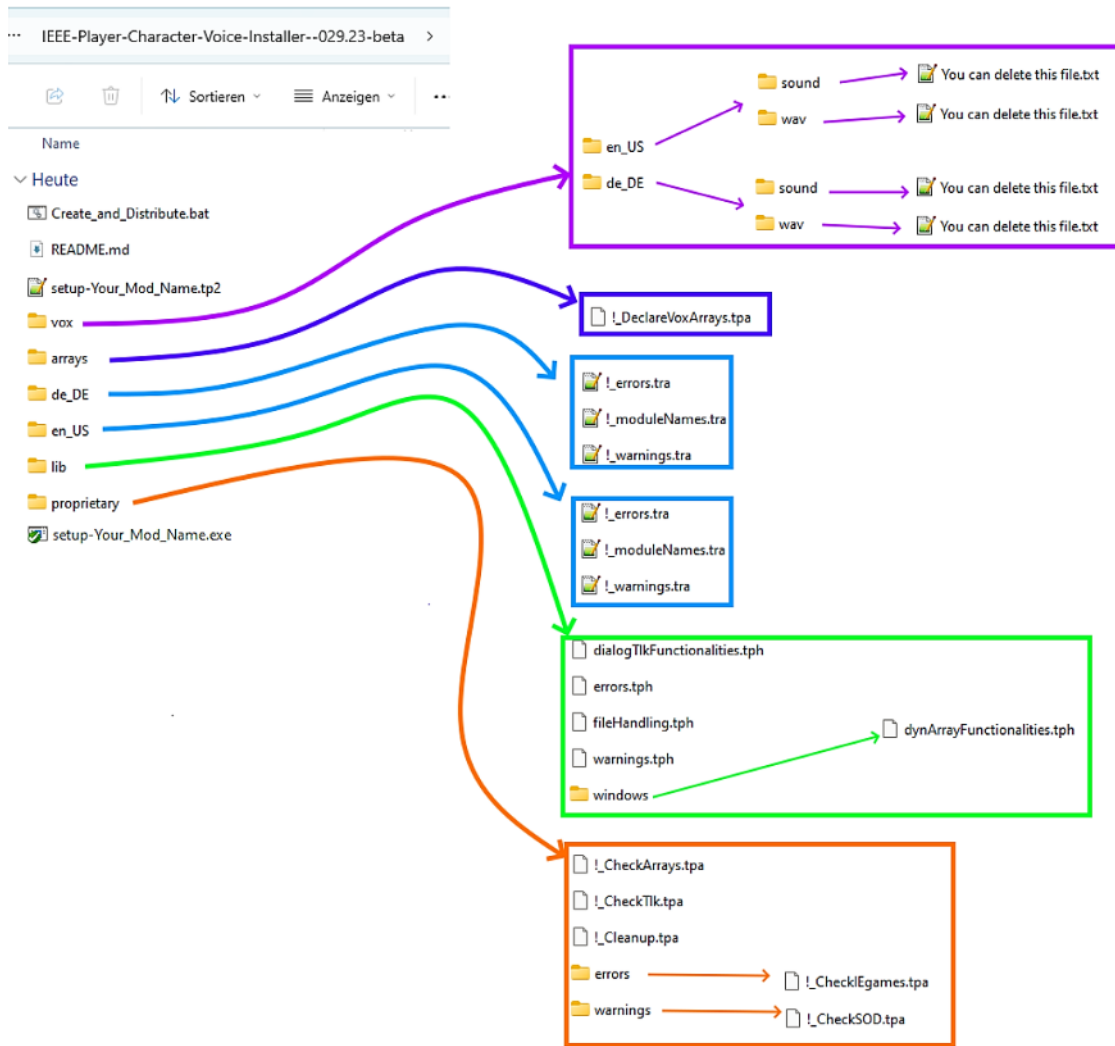


Figure4

4. Renaming Character Voice Sound Files

To customize voice sound files, you'll need to rename them using a base name and a postfix.

4.1 Base Name

- Choose a unique base name (maximum 6 characters) for each selectable voice in the game's sound customization menu.
- Example: **!_DEMO** for the custom character voice.

4.2 Postfix

- Add a postfix to the base name to define specific sound files.
- Example: **!_DEMO**b**** for the sound played when selecting the custom character as party leader.

Naming Rules

Base names cannot be subsets of other base names. Here follows an example:

Basename 1	Basename 2	Condition
!_DEMO	!_D	Not Allowed , !_D is a subset of !_DEMO
!_DEMO	!_DU	Allowed !_DU is not a subset of !_DEMO

The reason for this restriction is to prevent [duplication](#) in the game's sound customization menu.

5. Moving and Organizing Sound Files

5.1 Copy Files

- Once you've renamed your sound files, move them to the appropriate language folder within the **...\vox** directory.
- Example: German files go to **...\vox\de_DE**, English files go to **...\vox\en_US** (see **Figure5** for a visual example).

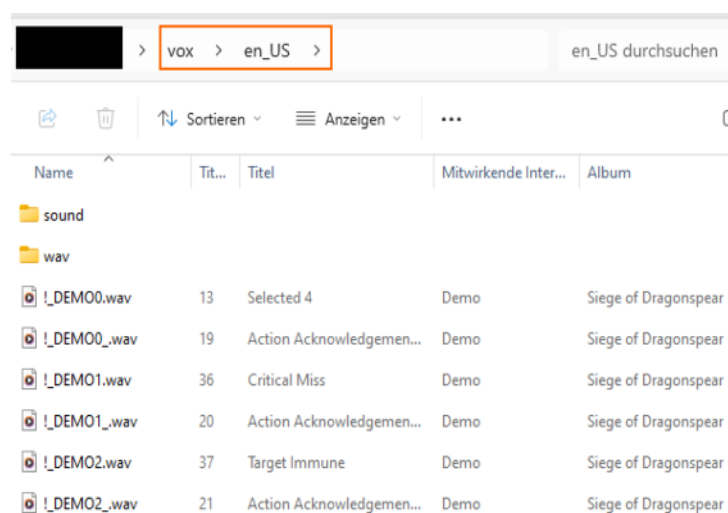


Figure5

5.2 Automatic Distribution

The script handles distributing files to **sound** and **wav** subfolders within each language folder.

- **sound:** Files with official postfixes.
- **wav:** Files with unofficial postfixes.
- **Code:** The script also generates new WeiDU code files for further integration.

Before proceeding, you need to edit the **Create_and_Distribute.bat** file:

1. Use a code editor like [Notepad++](#) to open the file.
2. Find the section labeled **INIT Soundset File Names** (see **Figure6** for an example).

```

9      REM -----
10     REM INIT Soundset File Names
11     REM -----
12     Set soundsetName[0]=!_DEMO

```

Figure6

3. Replace **!_DEMO** with your desired base name.
4. For multiple base names, add additional lines following the format **Set soundsetName[x]=<base name>**, where x is the index number (1 for the second name, etc.). Refer to **Figure7** for an example of how the configuration should look with two base names.

```

REM -----
REM INIT Soundset File Names
REM -----
Set soundsetName[0]=!_DEMO
Set soundsetName[1]=!_DEMB

```

Figure7

5. Find the section labeled **INIT Languages** (see **Figure8** for an example).

```

REM -----
REM INIT Languages
REM -----
REM (en_US = English, de_DE = German)
Set languages[0]=en_US

```

Figure8

6. Replace **en_US** with your desired language code. For example, use **de_DE** for German. If your custom voices support multiple languages, add new lines below the first one following the same format **Set languages[x]=<language code>** where x is the index number starting from 1 for the second language.

For example(see **Figure9**), to add German, you would include the line **Set languages[1]=de_DE**.

```
REM -----
REM INIT Languages
REM -----
REM (en_US = English, de_DE = German)
Set languages[0]=en_US
Set languages[1]=de_DE
```

Figure9

7. Refer to the **README.md** file or [online resources](#) for a list of supported languages.
8. Find the section labeled **INIT Soundset Custom Names** (see **Figure10** for an example).

```
REM -----
REM INIT Soundset Custom Names
REM -----
Set en_US[0]=!_Demo(Complete):Female Robot
```

Figure10

9. This section explains how to define the names displayed for your custom voices in the character customization menu (see Figure 11 for an example with two voices).

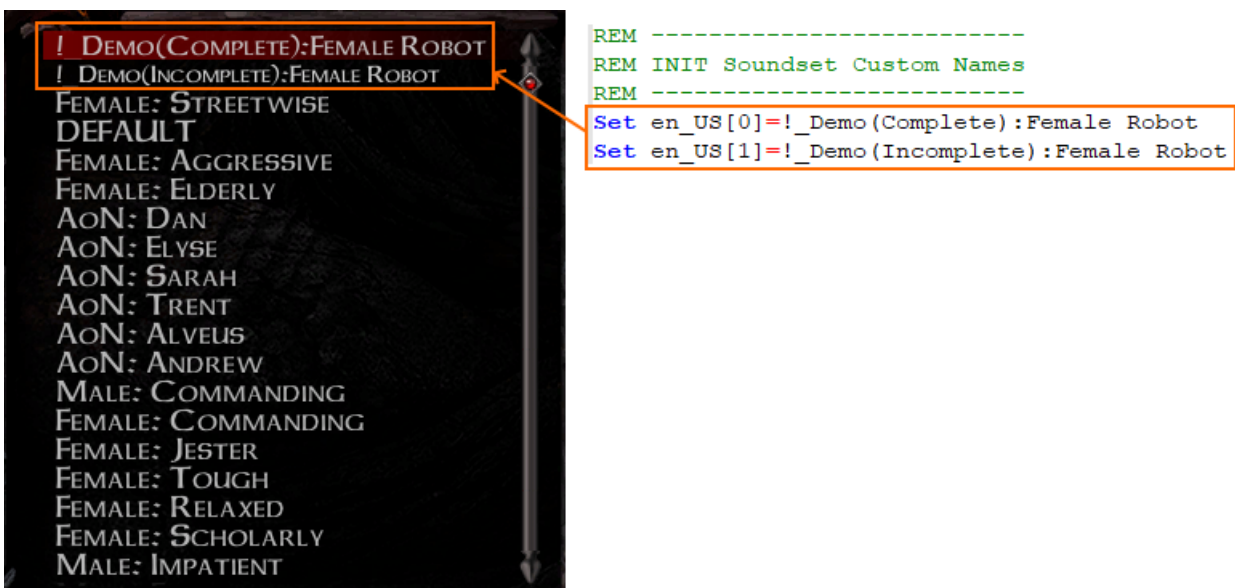


Figure11

10. Use the following format to set custom voice names for each language:

Set <language code>[x]=<custom voice name>

- **<language code>**: Replace with the appropriate code (e.g., en_US for English, de_DE for German).
- **x**: The index value representing the voice number in the menu (0 for the first, 1 for the second, etc.).

- **<custom voice name>**: The desired name for the voice in that language.

11. Refer to **Figure12** for an example of how to configure names for two custom voices in both English and German.

```
REM -----
REM INIT Soundset Custom Names
REM -----
Set en_US[0]=!_Demo(Complete):Female Robot
Set en_US[1]=!_Demo(Incomplete):Female Robot
Set de_DE[0]=!_Demo(Vollständig):Weiblich Roboter
Set de_DE[1]=!_Demo(Unvollständig):Weiblich Roboter
```

Figure12

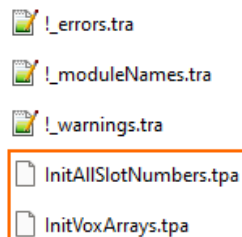
- The index value x starts at 0 and increments by 1 for each voice within a language.
- Ensure the number of voice names matches the number of custom voices you have created.

12. Save changes to **Create_and_Distribute.bat**.

6. Building the Mod

6.1 Generate WeiDU Code and Edit Voice Prompts

1. Double-click **Create_and_Distribute.bat** to process sound files and generate WeiDU code.
2. Depending on your script configuration, you'll find two .tpa files(see **Figure13**) in the relevant language folder (e.g., **de_DE** or **en_US**).

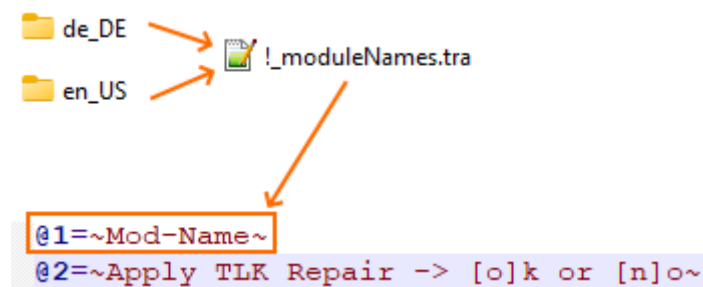
**Figure13**

3. Open **InitVoxArrays.tpa** with a code editor like Notepad++. (**Figure14** shows an example of its content.)

```
// ---INITIALIZE DYN ARRAY TEXTS
OUTER_SPRINT ~Texts0~ ~TODO~ /*"!_DEMB0.wav" Action Acknowledgement 4 for SoD/BGEE/IWDEE or else BGEE2 Select Rare 1
OUTER_SPRINT ~Texts1~ ~TODO~ /*"!_DEMBb.wav" Becoming Leader
OUTER_SPRINT ~Texts2~ ~TODO~ /*"!_DEMBf.wav" Selected 1
OUTER_SPRINT ~Texts3~ ~TODO~ /*"!_DEMO0.wav" Selected 4
OUTER_SPRINT ~Texts4~ ~TODO~ /*"!_DEMO0.wav" Action Acknowledgement 4 for SoD/BGEE/IWDEE or else BGEE2 Select Rare 1
OUTER_SPRINT ~Texts5~ ~TODO~ /*"!_DEMO1.wav" Critical Miss
```

Figure14

4. For each line containing `~TODO~`, enter the text you want your custom voice to speak. For example, replace `OUTER_SPRINT ~Texts0~ ~Yes, I will do as you say!~` with your desired phrase.
5. Look for comments starting with `//...` to understand what kind of text is expected for each line. These comments often mention the sound file name (including base name and postfix) and provide context (e.g., "`Selected 1`").
6. If you have sound files for more than one language, repeat steps 3-5 for each generated `InitAllSlotNumbers.tpa` file in the respective language folder.
7. Refer to the provided links ([BGEE](#), [SoD](#), [IWDEE](#), [BG2EE](#)) or the postfix tables in the `guide` folder for more information about sound files. In addition, the following rule in [this link](#) is important to understand which slot depends on which.
8. Save changes to `!_moduleNames.tra`.
9. In the same folders, change `@1=~Mod-Name~` in `!_moduleNames.tra` to your chosen mod name (e.g., `@1=~My_Voice_Collection~`). **Figure15** shows the path and the files content.

**Figure15**

10. Save changes to `!_moduleNames.tra`.
11. Open `setup-Your_Mod_Name.tp2` in a text editor.
12. Edit the "`Information`" section with your desired mod details.

```

// -----
// Information
// -----
BACKUP ~weidu external/Your Mod Name/backup~ // %MOD_FOLDER% doesn't work here
AUTHOR ~Incrementitis (incrementitis@gmail.com)~ B
VERSION ~v0.31.23~ C
AUTO_EVAL_STRINGS// This will prevent the need of EVAL for string variables

```

Figure16

- A: This is the mod name. A backup is saved here.
- B: Your name and contact details come here
- C: The version number of your mod

13. If offering only one language, delete the unused "**Languages**" area. Keep all the ones you need if offering multiple languages. See Figure17 for language areas(e.g., delete A or B or keep them both).

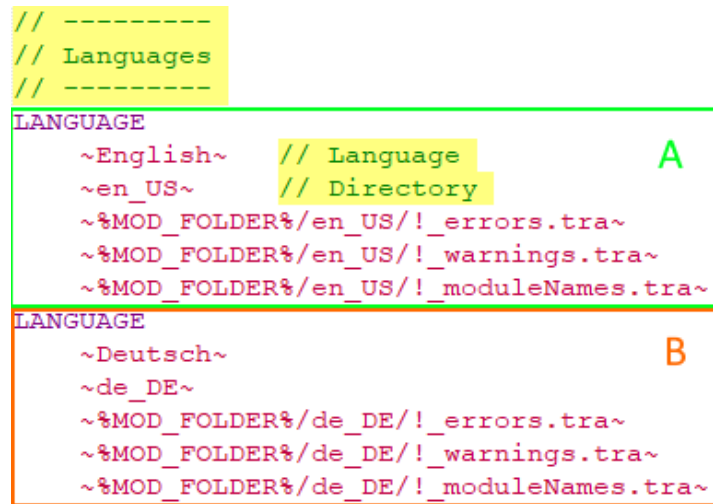
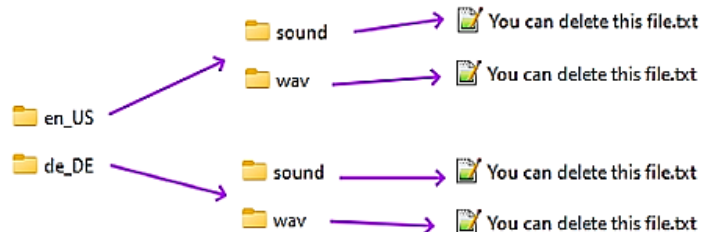


Figure17

14. Save changes to **setup-Your_Mod_Name.tp2**.

7. Finalizing the Mod

1. Rename **setup-Your_Mod_Name.tp2** to your chosen mod name
2. (e.g. **setup-My_Custom_Voices.tp2**).
3. Rename the **IEEE-Player-Character-Voice-Installer-<version>** folder to your desired mod name.
4. Rename **setup-Your_Mod_Name.exe** to a fitting name
5. (e.g., **setup-My_Custom_Voices.exe**).
6. Delete the following files:
 - **Create_and_Distribute.bat**
 - **You can delete this file.txt** in:
 - ...\vox\en_US\wav
 - ...\vox\en_US\sound
 - ...\vox\de_DE\wav
 - ...\vox\de_DE\sound
- **README.md**
- You can delete every file in the **guide** folder and replace it with your own guide(s)



8. Testing your Mod

8.1 Installing Your Mod

- Move your entire mod folder (example in **Figure18**) into your main game directory.
- Locate the file named **setup-Your_Mod_Name.exe** inside your copied mod folder (e.g., renamed in **Figure18** to **setup-Demo_SoD.exe**). Copy this file directly into your main game directory and then run it. This will initiate the mod installation.

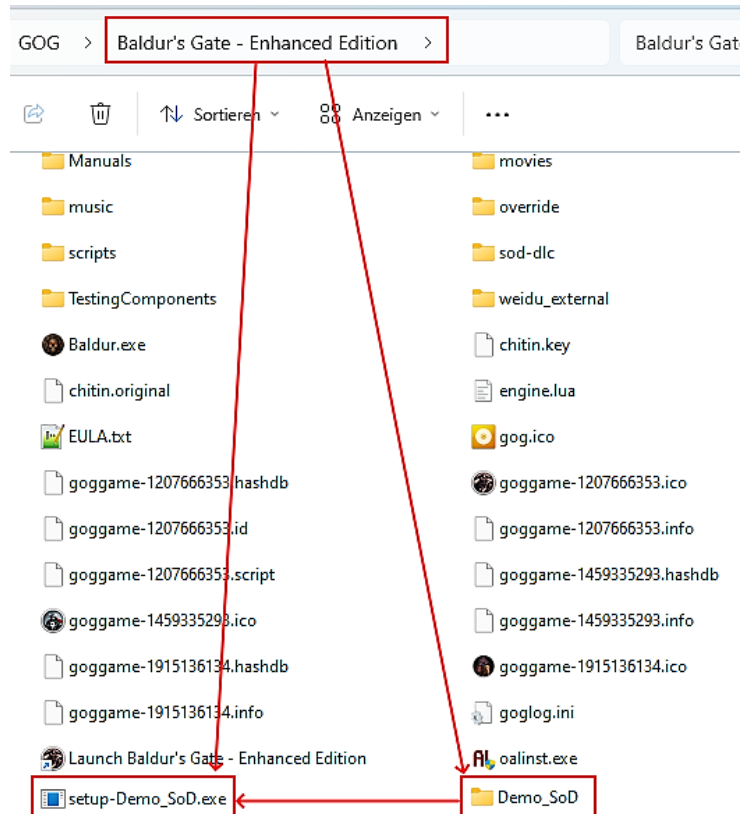


Figure18

8.2 TLK Repair (Optional)

This step is only required if you've manually edited text in **InitVoxArrays.tpa** outside of this guide (e.g., correcting typos). Skip it for your first installation (see example in see **Figure19**).

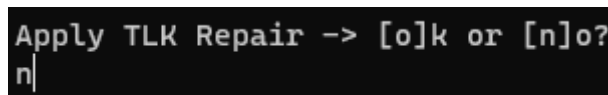


Figure19: Press „n“ for not to repair

8.3 When to Apply TLK Repair

- You finished your mod and distributed it (e.g., on GitHub). You later realize text needs fixing in `InitVoxArrays.tpa` due to accidental changes.

Applying TLK Repair will ONLY repair relevant text in your game's `dialog.tlk` file that matches your mod's sound file names. This prevents accidental changes and keeps `dialog.tlk` lean.

Note: Repairing a large `dialog.tlk` can take time. Expect longer times for bigger mods. (E.g., on my machine, repairing a MegaMod-sized `dialog.tlk` with nearly 700,000 entries took 3 minutes 40 seconds. 700,000 entries is seven times the size of *BG2EE* `dialog.tlk` entries..)

My System Specs (for reference):

WeiDU: Version 24900

Operating System: Microsoft Windows 11 Home

System Manufacturer: Acer

Processor: 12th Gen Intel(R) Core(TM) i7-12700F

RAM: 16 GB

Congratulations! You've reached the end of this guide.