TranslatorScript

Inhaltsverzeichnis

- 1 Benefits & Compatibility
- 2 Preparations
  - 2.1 Include TranslatorScript
  - 2.2 Setting Up Your Text
  - 2.3 File Encoding
- 3 Language File Configuration
  - 3.1 File Location
    - 3.1.1 Language File For Entire Folder
    - 3.1.2 Language File For Single *.mdl-File
  - 3.2 Syntax
  - 3.3 File Encoding
- 4 Example
- 5 Links

[Blockierte Grafik: http://ftp.train-fever.net/flaggen/de.png] Die deutsche Version gibt es hier

[Blockierte Grafik: http://ftp.train-fever.net/flaggen/gb.png] This litte script mod adds the possibility to translate (prepared) mods by just adding a language file.

[box]**Development aborted!**
With the possibilities of the new modding system (strings.lua) TranslatorScript is unnecessary. I will update this tutorial for the new system in the coming days.[/box]

1 Benefits & Compatibility

To translate the text of a model you normally have to provide a unique *.mdl-file for each language. Well, you think this is not a problem? Imagine you have to change a few numbers in your model. You have to do the changed in every file! Support is complex and time intensive.

With TranslatorScript you just have to provide one single *mdl-file for each model. The language data is outsourced to a language file. The user installs the language he wants beside your mod. TranslatorScript will do all the language integration into the mod. Only the language files themself must be made by hand.

Do you worry, that if TranslatorScript is not installed on your users installation, he can't use your mod? You don't need to. The include will just be skipped by Train Fever if TranslatorScript is not installed. Your mod will not be broken.

[box]If someone has not installed TranslatorScript this will **NOT** break your file![/box]

2 Preparations

2.1 Include TranslatorScript

Add the following code at the first line of your *.mdl-file:
Code

```
pcall(require, 'BR146_translate_script')
```

This line will be skipped if TranslatorScript is not installed.

2.2 Setting Up Your Text

Each text has to be surrounded with `_("YOUR TEXT")`. The important characters are the underscore and the brackets.
"YOUR TEXT" will be shown if

- no translation is installed
- the translation for this text is missing
- TranslatorScript is not installed

2.3 File Encoding

The *.mdl-file must be encoded as **UTF-8**.

3 Language File Configuration

3.1 File Location

[box]**A user only can have installed ONE language file at once! If no language file is installed, your original text is used.**[/box]
The path to the language file is generated as followed:
For a model in `res\models\model\<folder name>` the location of the language file will be `res\strings\models\<folder name>`

3.1.1 Language File For Entire Folder

For all * .mdl files in a shared folder, a single language file can be used. The name of the language file is then

`_TRANSLATION.lang`.

Example:
If the files are located in the path `res\models\model\railroad\BR146`, then he location of the language file is
`res\strings\models\railroad\BR146\_TRANSLATION.lang`.

3.1.2 Language File For Single *.mdl-File

For singe *.mdl-files - even in the same folder - can individual language files be used. The name is then formed from the file name of the *.mdl-file without the file extension (.mdl) and the suffix `.lang`.

Example:
If your model file is located at `res\models\model\railroad\yourModel.mdl`, the location of the language file is
`res\strings\models\railroad\yourModel.lang`.

3.2 Syntax

The language file consists of as many lines as in this example:
Code

```
return                                                                  {
["Orignaltext   mit   Leerzeichen"]   =   "the   translation   is   in   here",
["zweitens"]                  =                "second                 line",
...
}
```

3.3 File Encoding

The language file must be encoded as **UTF-8**.

4 Example

This example shows the skeleton of a *.mdl-file. The text to translate is "Signal" and "Signal von BR146". If only the mod is installed, Train Fever will show those two. If the language file (and TranslatorScript of course) is installed, "Segnale" will be shown instead of "Signal" (the same with the other text). If a text is missing, the original text from the mod is shown.

model file

Code

```
pcall(require,                                        'BR146_translate_script')


function                                                                  data()
return                                                                        {
[...]
metadata                                  =                                   {



            },
[...]
}
end
```

Alles anzeigen

language file

Code

```
return                                                                        {
                                                                  ["Signal"]
                  ["Signal        von        BR146"]        =        "Segnale        (BR14
}
```

5 Links

discussion thread
download