

# LoreBox Tutorial

## Contents

How it works .....	1
Distributing Keywords .....	1
Using KID.....	2
Language files .....	2
HTML Tags.....	3
Adding images .....	3
Conditionally adding keywords .....	4

## How it works

The mod works by assigning keywords to items / spells. The only rule is, the keyword must start with “**LoreBox\_**”. The mod grabs any matching keywords and displays them in the UI when you’re inspecting the item or spell. The text that those keywords should display comes from language files which we’ll discuss how to make.

First step, create your keywords, and save the ESP file.

## Distributing Keywords

You can edit the items (weapon, armor, Misc items, etc.) and add the keywords there or in the case of Spells, add keywords to a MagicEffect in that spell. However, this causes conflicts with mods that also edit those items/MagicEffects so it’s not ideal. Better option is to use the [Keyword Item Distributor \(KID\)](#) mod, KID is a

magical mod that allows distributing keywords to stuff without editing them in CreationKit.

## Using KID

All you need is to add a create an .ini file and place it next to your ESP file, name it: **<your ESP>\_KID.ini**. Now open it up. To add keywords follow this format:

```
Keyword = MyKeywordID|Form Type|MyItemID
```

So you start with `Keyword =`, then type the editor ID of your keyword, then a pipe character `|`, then the type of Form you're distributing keywords to, so if you're adding keywords to a weapon you type `Weapon`, for MagicEffects you type `Magic Effect`, followed by another pipe, then the editor ID of the item(s) you want to receive said keyword. If you have multiple forms that need to get the keyword, you can comma-separate them. So here's an example of adding a keyword to the five Shrouded Armor (from Dark Brotherhood) pieces:

```
Keyword =
LoreBox_DBArmor|Armor|DBArmor, DBArmorBoots, DBArmorGloves, DBArmor
Helmet, DBArmorHelmetMaskLess
```

Notice that our keyword has “`LoreBox_`” as prefix.

So our keywords are set. Now we need to add translation files to set what text is actually displayed when an item/spell has a relevant keyword.

## Language files

To learn how to setup language files, please see the [MCM Localization tutorial](#) here.

The translation keys must be the exact keyword that we added to our item, including its “`LoreBox_`” prefix. In the example above where I added a keyword for DBArmor armors, I need to add this to my translation files:

```
$LoreBox_DBArmor      Text displayed for Shrouded Armor!
```

## HTML Tags

You can use basic HTML tags in the translation files to customize the text:

- <font size="20">: font size is 20
- <p align="center">: a paragraph with center-aligned text
- <font color="#ff0000">: change text color, in this case red (hex code: #ff0000).
- <font face="">: change the font
- <br>: break the text to go to next line

Here's an example HTML text:

```
This is a text, <u>this part will have underline</u> and next words will show <font color="#00ffff">in cyan color</font>!<br>Now we're on a new line.
```

If you're not familiar with how HTML is formatted, I recommend looking this up online, it's very easy to get into.

<i> and <b> tags are also supported (to make the text italic or bold respectively), but whether that works depends on the font used. The default Skyrim UI font doesn't actually have Bold or Italic variants in it. Via <font face=""> tag it's possible to change the font:

```
<font face='$SkyrimBooks'>This text will use the font used for in-game Books.</font>
```

You can find the font lists in **Data\Interface\fontconfig.txt**. The UI by default uses **\$EverywhereMediumFont** font, lucky for us this has a Bold style font:

```
<font face='$EverywhereBoldFont'>This text will now show as bold text.</font>
```

## Adding images

Images can be displayed via <img> tag, it works like this:

```

```

The path for the image starts from /data/interface folder, so in our example, we must place the image into “interface/MyImage.dds” directory inside our mod. The width and height attributes are mandatory, you must specify them in order for the image to show. LoreBox’s default skin has a width of 230 pixels, so `width="230"` makes the image the same length as the box. If it’s smaller, the image is center-aligned.

For better image quality, I’d recommend making the images twice or thrice the size of what you specify in the `<img>` tag.

All images are in .dds extension. To start, create your images in .png format, then download and install the [xMadHack](#), then right click on your images and select xMadHack Tools > Convert to DDS > Uncompressed. After conversion you can remove the png file, as it’s not needed anymore.

## Conditionally adding keywords

What if we want the LoreBox to appear only after certain condition is met, like finishing a quest or what have you. Powerofthree comes to our rescue again, with [Powerofthree's Papyrus Extender](#). The mod offers a function called `AddKeywordToForm()` which we can use to add a keyword to our desired Form using Papyrus.

First import the script in your own script:

```
Import PO3_SKSEFunctions
```

Use Properties to connect your script to game Forms, in this example I want to add a keyword to all Iron Swords:

```
Weapon Property IronSword Auto
```

```
Keyword Property LoreBox_MyIronSword Auto
```

Now call the function to attach our keyword:

```
AddKeywordToForm(IronSword, LoreBox_MyIronSword)
```