Adding and Customizing UI Icons in Bforartists

Preface

There are two icon types in Bforartists, the big ones in the tool shelf (introduced with Blender 2.8), and the small ones found throughout the rest of the UI.

This tutorial will cover the small icons. These are the ones that can be found, for example, in the toolbar at the top and in the text menus. At the moment there is a missing icon in the View menu of the 3D view. This "Remove from Local View" command will be the example used in this tutorial.

IMPORTANT! This tutorial assumes that you use Inkscape 1 or higher.
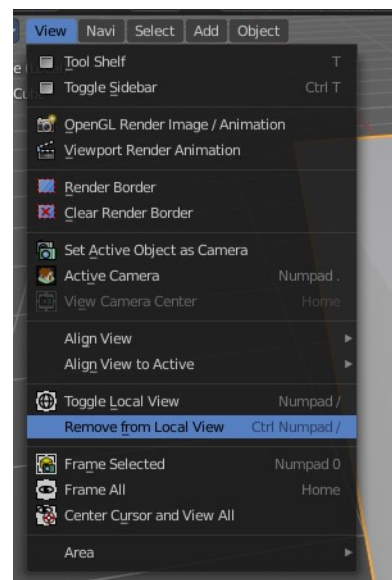
Setup

Before beginning the tutorial, please ensure the following have been done:
✔ The repository has been downloaded (https://bit.ly/2rVvNd5)
✔ Python 2.7x and 3.xx are installed (https://www.python.org/downloads/)
✔ Inkscape,newest version (https://inkscape.org/)
✔ Blender is installed (https://www.blender.org/download)

In case you work at Windows, both, Blender and Inkscape needs to be added to the Windows PATH variable. At the time of writing this tutorial, the Inkscape installer was not able to do this for you. So you might add it by hand. Adding Python should happen automatically.

Adding System variables

Here's an example where you add the path to the Blender exe in case it is missing. Windows System control,

Advanced system settings -> System properties.
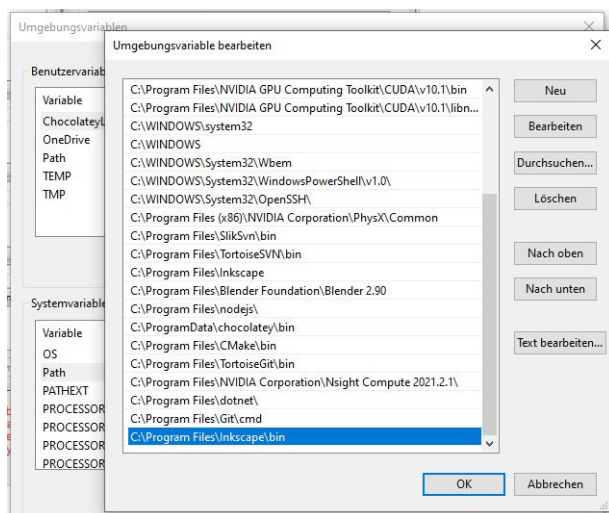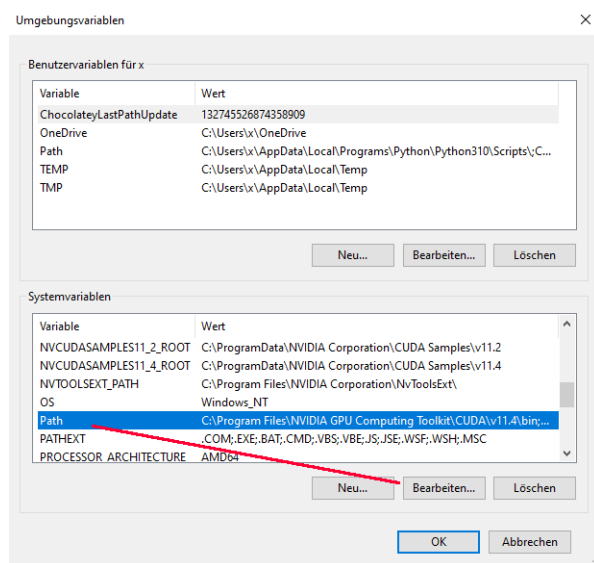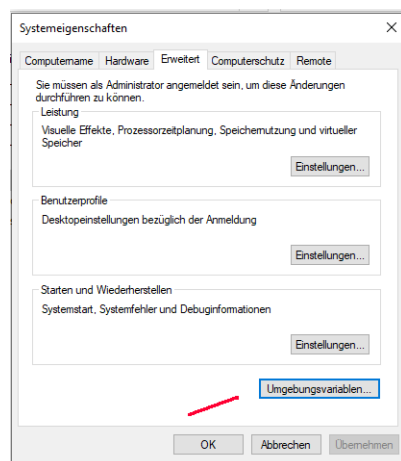
Sorry for the german shots.

Inkscape document origin

The icon sheets are made with an older version of Inkscape. And this older version has the origin at the lower left corner. The default for Inkscape and higher is to have the document origin at the upper left corner though.

And then the icon creation does not fit anymore.

So open up the Inkscape settings, and set the document origin to be down left by unticking the origin in the upper left corner
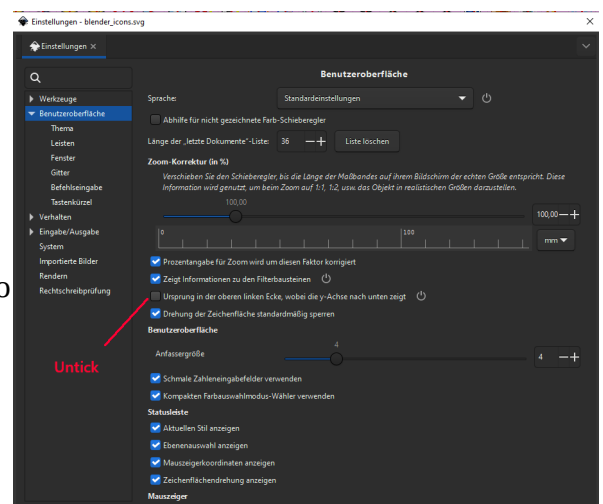
Process Overview

The process for both adding and customizing icons in Bforartists is as follows:

1) Add an icon to the *blender_icons.svg* **file** in the *release\datafiles* **folder**.

2) Add the name of the icon to the *source\blender\editors\include\UI_icons.h* **file**. This file is found in the same **folder** that holds *blender_icons.svg*. Use an empty icon slot from the `#ifndef DEF_ICON_BLANK_SKIP` list, and make a note of which slot is used.

3) Run *blender_icons_update.py* from the console. For the example here, the repository is located on drive "H" in folder *bforartists*. Adjust your paths to your needs.
   ```
   h:
   cd bforartists\bforartists\release\datafiles
   python blender_icons_update.py blender_icons.svg
   ```

4) Compile. This makes the icon available to the Python code.

5) Add the icon to the operator in the Python file by changing a single line of code. This tutorial focuses on the *view3d.localview_remove_from* **button** in the view menu in this tutorial:

   Change
   ```
   layout.operator("view3d.localview_remove_from")
   ```
   to
   ```
   layout.operator("view3d.localview_remove_from", icon = "VIEW_REMOVE_LOCAL")
   ```

Creating Icon Art

The first step is creating the icon art itself. What follows is just a suggestion; as every artist has their favored tools, any bitmap or vector graphics software can be used to create the art. **Gimp 2.10*** is used in this example to create the icon as a bitmap graphic, which will then be converted to the .svg format with **Inkscape****.

*GIMP is open source raster graphics creation and editing software. GIMP can be downloaded here: https://www.gimp.org/downloads/.

**Inkscape is open source vector graphics creation and editing software. Inkscape can be found here: https://inkscape.org. A vector graphics program capable of working with .svg files is required to edit the icon sheet in the Bforartists repository, as the icons are all stored in a single large .svg file. Other programs can be used to edit the .svg icon sheet, but this tutorial will stick to Inkscape because it is freely available.

The new icon artwork can be created at any size. It will be resized later. Here the icon is being created at a roomy 512 x 512 pixels in GIMP. "Grid" is toggled on and configured to 16 x 16 fields,with each block 32 pixels in width and height. Not coincidentally, the final icon must be 16 x 16 px, so this canvas setup is very convenient. The grid also allows the use of the Snap tool.

Make sure that there are no gradients in the icon! For example, the Circle Select tool creates antialias gradients on its border when you fill it. These gradients make conversion more difficult in the next step. Good practice is to change the color mode to "Indexed" with the maximum number of colors set to equal the numbers of colors used in the icon art, then switch it back to RGB. This removes gradients.
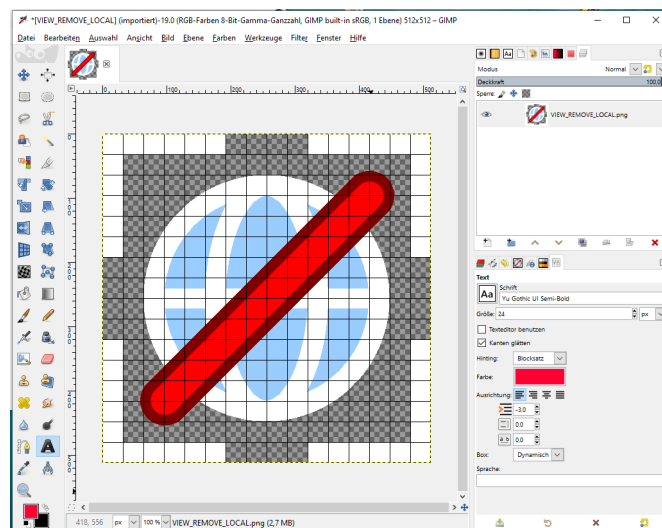
As for the artwork, it can be anything the user can imagine! Remove is a red cross in most icons, so this example adds such a symbol to the aforementioned "Toggle Local View" icon.

The new icon is saved in GIMP and titled VIEW_REMOVE_LOCAL.png. There are strict conventions concerning icon names. Icons are defined by CAPITAL LETTERS in the code.

Convert the Icon Art to .svg

The new icon art must be converted to .svg so it can be added to the icon sheet. Most vector graphic software can export files into that format. Inkscape will be used to perform the conversion in this example.

Open the new icon png image file in Inkscape. Use the defaults for the import. We don't want any corners optimized or anything like that. We need it as it came out of Gimp. Pixel perfect!
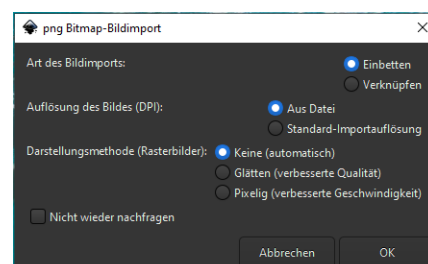
In Inkscape go to Path > Trace Bitmap
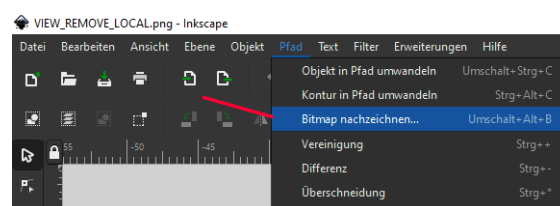
The dialog opens at the right.

Switch to multicolor.
Set the mode to Colors.
Set up the scans to the number of colors plus minimum 1. The reason is the transparent background versus white colors in the icon itself. This transparent background turns white in the conversion process. And we don't want it to union with white areas in the image.
Untick the details. They can ruin the scan. We want the icon to arrive as it was shaped in Gimp. Pixel perfect.

When done click at Apply. This creates a vector graphics layer.
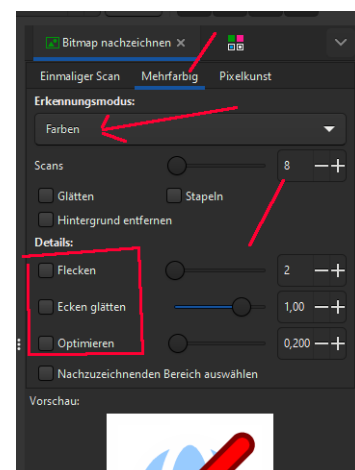
Delete the Bitmap layer from the file. This can be done in the XML editor.

Rename the vector graphics layer to match the icon file name created in the previous section. With close to 1,400 icons on the sheet so far, clear nomenclature is a vital part of keeping track of everything

Resize the canvas down to 16 by 16 pixels in case it is not in this size already.
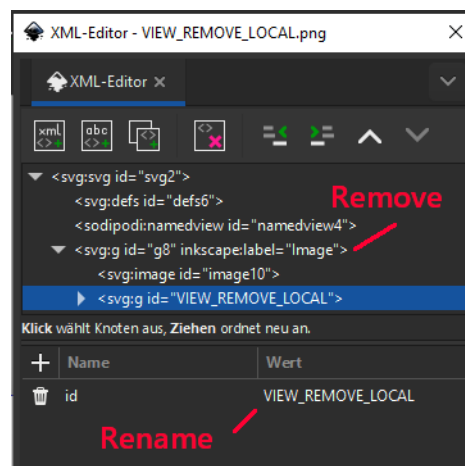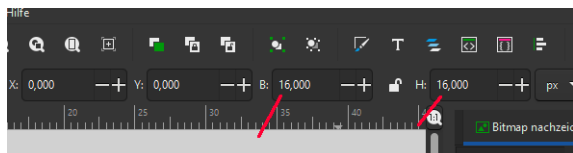
Save the Inkscape file with the icon name. Use SAVE AS! Save will overwrite your png file ...


VIEW_REMOVE_LOCAL.svg

Add the icon to the Icon sheet

The icon sheet is a .svg file that contains all the small icons used by the Bforartists UI. Before a new icon can show up as a part of the UI, it must be included in the master icon sheet. The next step is adding the new icon to this master document.

Find and open the file "blender_icons.svg". It can be found in the folder release\datafiles in the repository.

At the time of this writing, the file contains over 1700 icons.
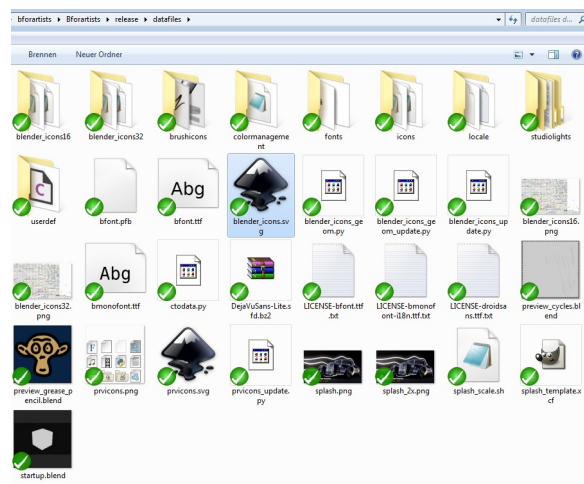
The icons themselves are in a group on Layer 1. Click on one of the icons to open the icon group. The group needs to be opened to ensure the new icon is added to the right part of the sheet.
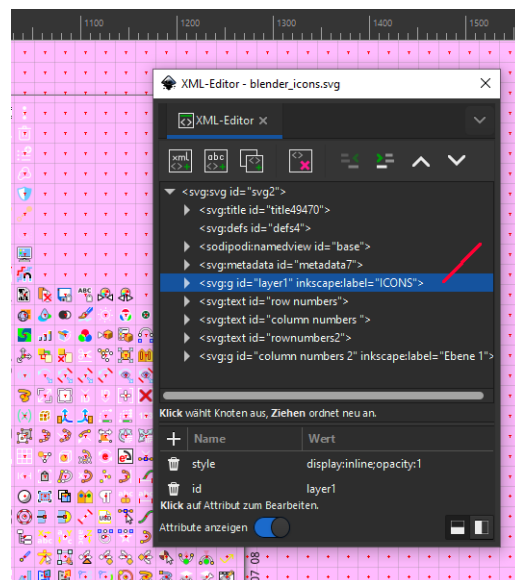
Before progressing any further, another file needs to be opened. Open the repository and go to source\ blender\editors\include. Open the UI_Icons.h file found there. This is a list of the icons that Bforartists uses to define the UI art. The order of this list goes from the lower left to the upper right of the image.
This list has been divided by separation lines to make navigation easier.

Look for `#ifndef DEF_ICON_BLANK_SKIP` labels.

Icons defined thusly are free, and their space is available to use. In this case DEF_ICON(extend_0011) and DEF_ICON(extend_0012) will be utilized. There are two gaps at the end of row seven on the icon sheet corresponding with the two `DEF_ICON_BLANK_SKIP` items on the list.

Activate snapping. Best without Object frame option. And place the icon in the sheet at the position where you want to have it.
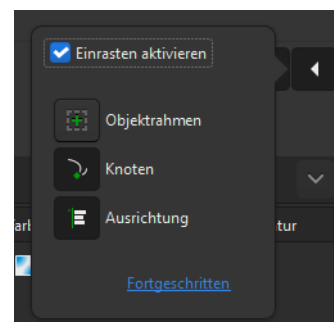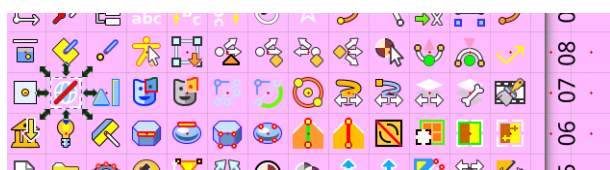
The new icon is pasted into on of these blank spaces. This task is made easier by the forthought that was utilized in resizing the canvas down to 16 x 16 units. The other icons in the blender_icons.svg are also 16 units tall.

Save the icon sheet file.

Dropping the icon into the iconsheet was just half the show. Now the UI_icons.h file needs to be updated. Be aware of the order. The new icon was dropped at the position of DEF_ICON(extend_0011). Replace the extend_0011 with the name of our icon. And move it out of the define group DEF_ICON_BLANK_SKIP. This icon should not be skipped anymore.

Save the .h file.
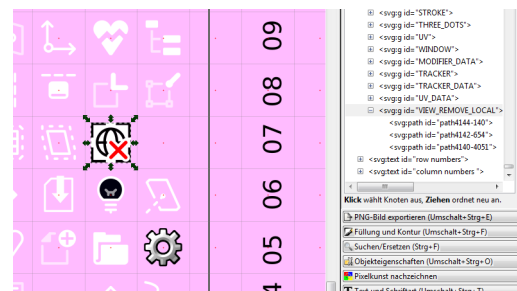
```
457     DEF_ICON(LIGHT_POINT)
458     DEF_ICON(LIGHT_SPOT)
459     DEF_ICON(LIGHT_SUN)
460     DEF_ICON(LIGHTPROBE_CUBEMAP)
461     DEF_ICON(LIGHTPROBE_GRID)
462     DEF_ICON(LIGHTPROBE_PLANAR)
463     DEF_ICON(VIEW_REMOVE_LOCAL)
464
465  ⊟#ifndef DEF_ICON_BLANK_SKIP
466
467        DEF_ICON(extend_0012)
468
469     #endif
470
471     /* ---------------------- Row 8 ------------------------*/
472
```

Note that the icon is not defined by the name in the svg file. The name in the svg file does not play any role. The icon is defined by the order and name in the UI_Icons.h file!

Add the new icon to the source\blender\editors\datafiles\ Cmakelist.txt

There is another location where the icon must be added. There is a Cmakelist.txt file that holds a list of all icons.

Go to source\blender\editors\datafiles\Cmakelist.txt, and add the name of the new icon to the list of icons in the **set(ICON_NAMES** class.

Beware, in this file the letters must be lower case!

Take care with the order and the position of the new icon in the list. It must match the UI_icons.h file.

```
1 CMakeLists.txt
479     light_spot
480     light_sun
481     lightprobe_cubemap
482     lightprobe_grid
483     lightprobe_planar
484     view_remove_local
485     shape
486
487     mask_slice_fill
488     mask_slice_new
489     simplify_sample
490     simplify_adaptive
491     toggle_close
492     separate_gp_strokes
493     separate_gp_points
494     separate_gp_layer
495     bone_layer
496     clean_channels_frames
497
498     # ---------------------- row 8 -----------------------*/
499
```

Create the dat files for the icons

The code does not deal directly with the .svg file. It needs the icons in dat format so that it can be compiled into the Bforartists exe. These .dat files can be found in the blender_icons16 and blender_icons32 folders in the release\datafiles directory.

To create those dat files we need to run the blender_icons_update.py script, which is included in the release\ datafiles folder as well.
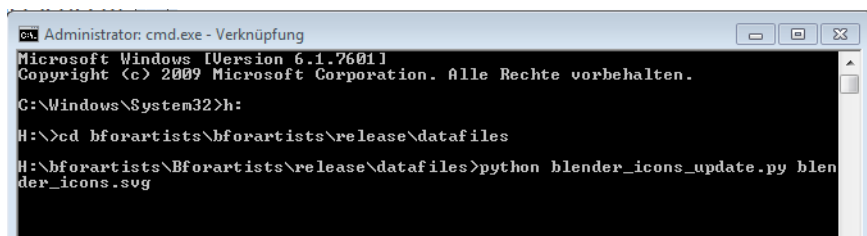
What this script does is to export the whole .svg file to a .png file. Then it loads this png file into Blender, slices it into the single icons, and exports them as .dat files. The pieces of this intermediate step can be found in the datafiles folder (the two .png files with the icons).

This is done in the command line. On Windows, open cmd.exe. In the command line navigate to the release/datafiles folder. In the example here, the repository is on drive H in the folder bforartists. Therefore, the following is typed into the cmd window:

h:
cd bforartists\bforartists\release\datafiles

Then we execute the script typing:

python blender_icons_update.py blender_icons.svg

Hit Enter. This starts Blender in the background, and it converts the icon files to the dat files in the blender_icons16 and blender_icons32 folder.

Wait a few moments so that the process can finish, and there will be a new icon in the blender_icons16 and blender_icons32 folder: the newly created icon.
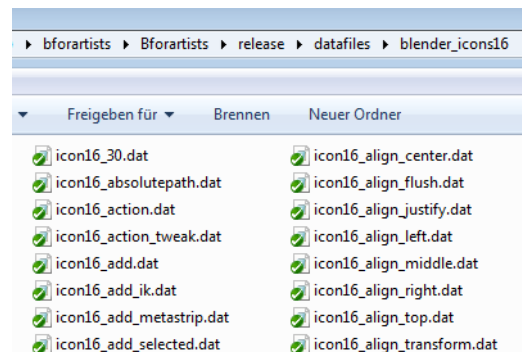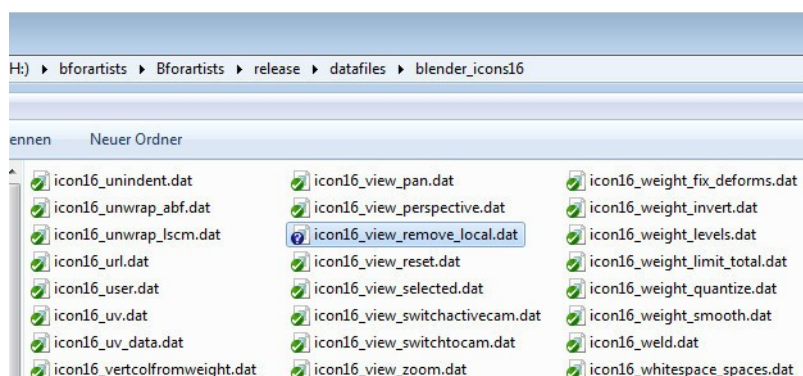
Note that existing icons are overwritten as well. So long as they have not changed, this overwrite will not alter the UI in any way. Also note that the number and naming in the blender_icons16 and 32 folders needs to fit exactly to the number and naming in the UI_Icons.h file. One dat file too many or one dat file missing leads to a compile error.

This error can happen when an incorrect icon name is added. If the script returns an error, correct the name and run blender_icons_update.py again. The icon with the wrong name will still be in the blender_icons16 and 32 folders. The script does not remove icons, it just add them.

The program is now ready to be compiled and checked.

Add the new icon into the menu

Bforartists must be compiled before continuing any further. This will make the icon available for the Python UI code.

The new icon must now be associatted with the menu item. Open Bforartists. Activate the developer tools in the preferences. Python Tooltips and the Developer Extras are necessary for this task.
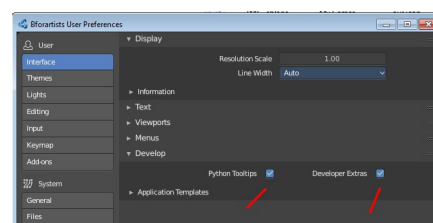

The menu is Python codeWhen a menu element is right-clicked, an "Edit Source" entry appears.

Switch to Scripting layout. Right click on the menu in the header where the new icon is needed, and click "Edit Source". The UI script that contains this menu will now load, and it will jump to the first location where this menu is defined.
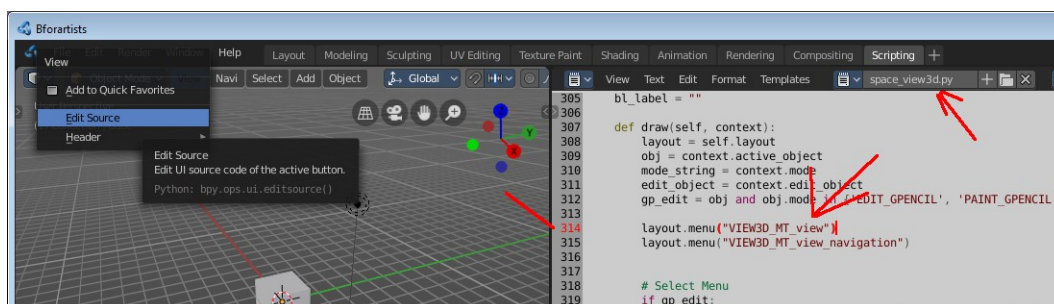
This is not the menu class itself, but here the menu gets referenced. The menu class itself must now be found1. This can be done in an external editor like VS Code, or in Bforartists directly. The important part here was to find the name of the menu quickly so that we can find the button in the code.

Note that it is the UI file in the compiled exe that loads here, not the one in the repository. The compiled file should be moved into the repository when done.
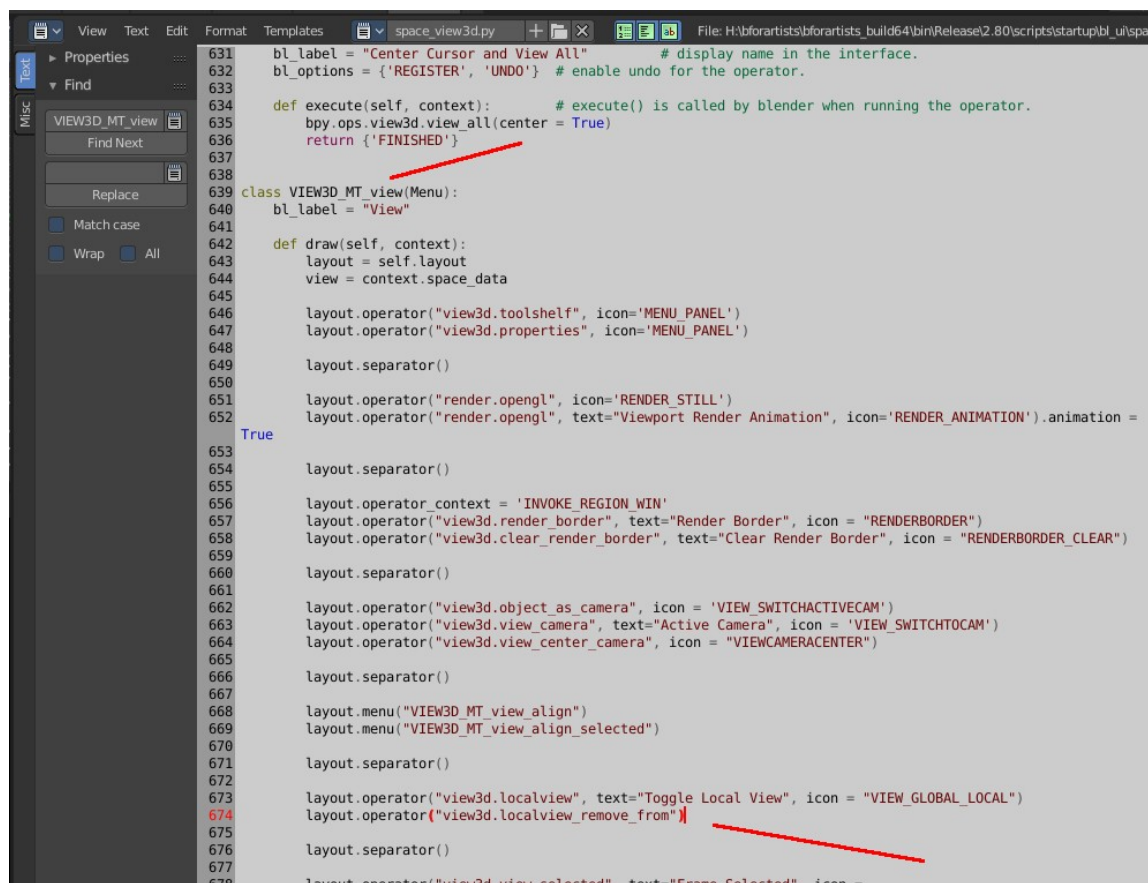
And here is the layout operator with the remove from local view button.

Above, in the localview operator, the term is seen with an icon and text. Now to add the icon part to the operator. The button label is fine, so there is no need to change it.

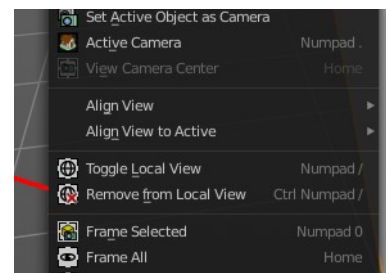Change

layout.operator("view3d.localview_remove_from")

to

*layout.operator("view3d.localview_remove_from", icon = "VIEW_REMOVE_LOCAL")*

Save the file. Close Bforartists. Reopen Bforartists. The new icon is now implemented.

What is left to do is to move the modified Python file into the repository. Here it was the space_view3d.py file that was have modified.
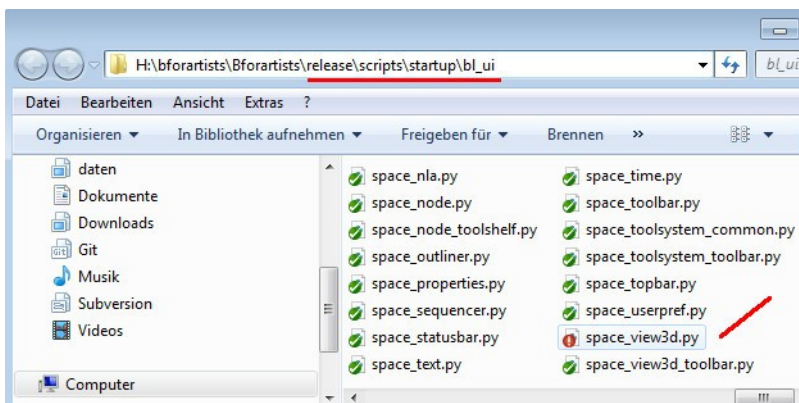
Most of the Python UI scripts are in a subfoder in scripts\startup\bl_ui .

For the binary that's *2.80\scripts\startup\bl_ui*
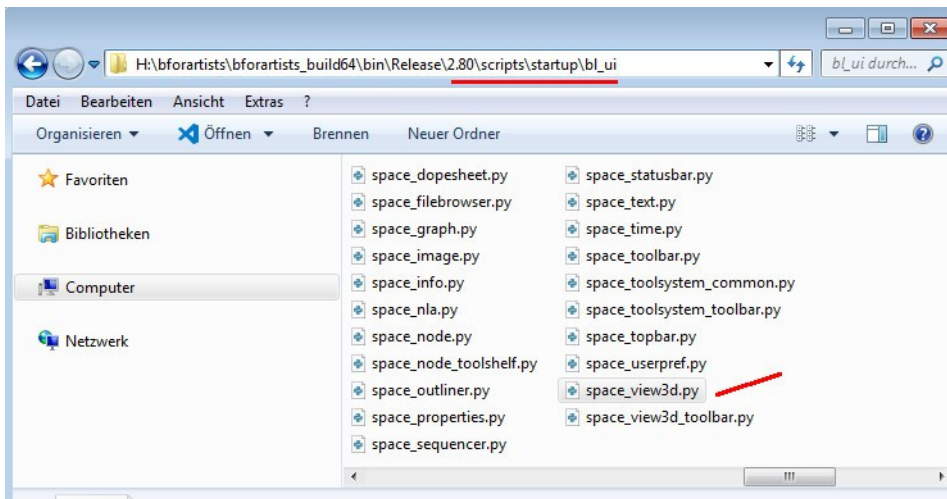
In the repository this can be found in *release\scripts\startup\bl_ui*

Copy it over, and commit the changes.