# Table of Contents

The Image Library is used to conveniently access pictures in an application. Pictures are usually stored in the class path of an application and loaded using the complete resource path. For example:

```
Toolkit.class.getResourceAsStream("/jvx/rad/genui/images/32x32/about.png");
```

The Image Library allows access via the complete resource path as well as via predefined and/or user-defined picture names. The following example loads the picture about.png from our previous example via the resource path and the predefined picture name:

```
//possible
UIImage.getImage("/jvx/rad/genui/images/32x32/about.png");

//recommended
UIImage.getImage(UIImage.ABOUT_LARGE);
```

The Image Library already contains important application-pictures/icons that are commonly used, such as:
SAVE_SMALL, SAVE_LARGE,
COPY_SMALL, COPY_LARGE,
PASTE_SMALL, PASTE_LARGE,
SEARCH_SMALL, SEARCH_LARGE,
etc.

A small (16×16 px) and a large (24×24 px) version exists for each predefined picture. The small version is used for menus and buttons, whereas the large version is used for notifications, details or toolbars.

Please use the class UIImage for a complete list of predefined pictures.

Using the Image Library and picture names, it is very easy to replace single pictures at a central location for the entire application. It is not necessary to search for a picture in each work-screen and to change the resource path. Changing the picture definition is sufficient.

**Example**

Our Application only uses the Image Library for the display of pictures/icons. We use predefined pictures and additionally define our own pictures that are not contained in the Image Library. We also change some of the predefined pictures as they do not suit our taste.

We define the changes in our application´s constructor:

```
//We don't like the predefined add/remove images
UIImage.setImageMapping(UIImage.ADD_SMALL,
"/com/sibvisions/apps/packung/images/16x16/new.png");
UIImage.setImageMapping(UIImage.REMOVE_SMALL,
"/com/sibvisions/apps/packung/images/16x16/delete.png");

//We don't need the large images
UIImage.setImageMapping(UIImage.ADD_LARGE, null);
UIImage.setImageMapping(UIImage.REMOVE_LARGE, null);
```

```
//We have special images which are not predefined
UIImage.setImageMapping(PackungApplication.IMAGE_NOBODY,
                        "/com/sibvisions/apps/packung/images/nobody.png");
```

On one of our workscreens, we now use the Image Library for our buttons' pictures:

```
UIButton butOK = new UIButton();
butOK.setImage(UIImage.getImage(UIImage.OK_SMALL));
```

It is also possible to use the picture names instead of the resource path information. This is the case when no IImage instance is required, e.g.:

```
/** Yes/ No Choice editor. */
new UIChoiceCellEditor(new Object [] {"Y", "N"},
                       new String [] {UIImage.CHECK_YES_SMALL,
                                      UIImage.CHECK_SMALL},
                                      UIImage.CHECK_SMALL);
```

Compare this to the access without the Image Library:

```
/** Yes/ No Choice editor. */
new UIChoiceCellEditor(new Object [] {"Y", "N"},
                       new String []
{"/jvx/rad/genui/images/16x16/check_yes.png",
"/jvx/rad/genui/images/16x16/check.png"},
"/jvx/rad/genui/images/16x16/check.png");
```

If we now decide to change the application´s pictures, we do not need to search the entire application. It is sufficient to change the definitions:

```
UIImage.setImageMapping(UIImage.CHECK_YES_SMALL,
"/com/sibvisions/apps/packung/images/16x16/packung_check_yes_small.png");
UIImage.setImageMapping(UIImage.CHECK_SMALL,
"/com/sibvisions/apps/packung/images/16x16/packung_check_small.png");
```

## Note

The Image Library saves pictures that have already been used, which prevents loading the same picture multiple times. This not only saves memory, but also positively affects performance.