Table of Contents

If you use a ProjX application, you'll have different options to change the look and feel. The first one will be a custom Swing look and feel or custom CSS (HTML5 mode). The second one is extending default ProjX and change the style programatically. The third option is to implement your own theme.

A theme is not a replacement for Swing look and feel or CSS, because you can't change everything. A theme should be used to set default choice editors, checkbox editor, change image mappings, set toolbar button size, ...

It's very easy to create a new theme because there's only an interface with one method:

ITheme.java

```
public interface ITheme
{
   //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
   // Method definitions
   //~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

   /**
    * Applies the theme for the given application.
    *
    * @param pApplication the application
    */
   public void apply(Application pApplication);

}   // ITheme
```

Simply implement this interface and set the theme via application parameter.

ProjX supports two built-in themes. The implementations are ClassicTheme and ModernTheme. To switch the default theme, simply set the parameter:

Application.iconstyle to classic or modern

An example implementation could do the following:

```
public void apply(Application pApplication)
{
   UIImage.setImageMapping(UIImage.SAVE_SMALL,
"/com/sibvisions/demo/images/save.png");
   UIImage.setImageMapping(UIImage.SAVE_LARGE,
"/com/sibvisions/demo/images/save_l.png");

   UIImage.setImageMapping(UIImage.REDO_SMALL,
"/com/sibvisions/demo/images/redo.png");
   UIImage.setImageMapping(UIImage.REDO_LARGE,
"/com/sibvisions/demo/images/redo_l.png");

   NavigationTable.setDefaultButtonSize(21);
}
```

From:
https://doc.sibvisions.com/ - **Documentation**

Permanent link:
https://doc.sibvisions.com/applications/custom_themes

Last update: **2020/07/08 13:11**