Skin Architecture

Introduction

The MediaPortal graphical user interface is a modular solution that uses skins to separate the look and feel from the logic in the application code. MediaPortal application functionality is surfaced through user interface windows (GUIWindow) containing many graphical user interface controls (GUIControl). However, without a skin file, none of this functionality is visible to the user.

The user accesses the functionality surfaced from the application code through skins which are defined using XML. The application will load the appropriate XML skin file upon initialization (based on Window ID). For example, when MediaPortal starts, the first skin generally loaded will be the *myHome.xml* or *BasicHome.xml* skin file depending on which option the user has configured.

The use of XML simply provides a structured way to tell MediaPortal the layout, type of control and coordinates of every item on screen such as color of text, transparency if graphics etc. By changing the XML file within a skin the entire look and feel can be modified. Although the details are very different, the idea is the same as making a webpage however **XML files are CASE SENSITIVE**.

Skin Folders

Skin files are contained in the

```
skins
```

folder within the main MediaPortal application folder.

Within the main skins folder there are sub-folders for each skin (e.g. Default and DefaultWide are installed with MediaPortal) and those folders contain the individual skin XML files (e.g. *myHome.xml*).

The following sub-folders are used :

- \media - contains the graphics files used by the skin - usually .png format
- \sounds - contains the WAV sound files used by MediaPortal and plugins
- \themes - as of v 1.3.0 contains theme files - see Skin Themes
- \fonts - some skins install font files for use in the skin, although in some cases these are installed directly to the Windows fonts folder by the Skin Installer (e.g. MPEI)

Additional sub-folders may be added in some skins, usually to support optional layouts, translations of text used in the skin, etc.

Media

The media folder should have all the textures that the skin file uses (normally in PNG format for best quality).

Packed Graphic Files

MediaPortal 'packs' and caches png images from the media folder (not sub-folders).

The packed graphics files (are located within individual skin folders in the Cache folder e.g.:

- ...\MediaPortal\Cache\

Why are they needed?

Packed textures decrease the workload of the GPU with every single rendering pass.

How are they created?

Packed graphics are generated at runtime and are used to help keep the rendering portion of the application as efficient as possible. They are created only when no packed graphics already exist.

They can be identified by the filename prefix *packedgfx* and end with either .bxml (binary xml) or .png.

When the individual graphic files of a skin are modified, the packed graphic files and the related .bxml files must be deleted (i.e. recached) otherwise the changes will not be visible in MediaPortal.

What skin designers have to pay attention on when designing skins

1. **A 128MB GPU can only take 4 packed textures files.**
   a. **If your skin creates more, then it will not work with such gfx-cards!**
2. *texture sizes:* no other than power of two dimensions should be used
3. *texture sizes:* trim all unused space around the actual texture (if any)

See: Optimize your Skin for further information.

Sound

The sounds folder contain WAV files used by the MediaPortal application and plugins.

The following sounds are included in the default skins for navigational sounds. They can be adjusted in the configuration utility under General settings, Keys and Sounds.

- back.wav
- click.wav
- cursor.wav
- photo.wav

The following sounds are used by plugins:

- notify.wav (MyMail plugin)
- MyTetris.Block.wav (MyTetris plugin)
- MyTetris.Knock.wav (MyTetris plugin)
- MyTetris.Level.wav (MyTetris plugin)
- MyTetris.Line.wav (MyTetris plugin)

'Global' XML files

A number of xml files are used by MediaPortal 'globally' - i.e. they are displayed by the software under certain conditions and may be displayed in any or all windows: **Dialogs**, **Media Overlays**, **Topbar**

Dialogs

Dialogs are displayed when a user presses the More/Info key on their remote or the F9 key on their keyboard, or sometimes when a user selects an option from a context menu. Different Dialogs display depending on the window and current action. See List of Dialogs

Media Overlays

Media Overlays, sometimes called 'previews', are the small windows that overlay the currently playing video, TV or music on any window. The default overlays (videoOverlay.xml, musicOverlay.xml) can be turned on or off in any xml using the <allowoverlay> window element e.g.:

- <allowoverlay>yes</allowoverlay>.

A 'special' xml: videoOverlayTop.xml is also displayed whenever the default overlays are used and video is playing. The controls in this file display 'on top' of the video window.

Media overlays can also be defined uniquely in a skin xml file and then imported (see <import> files below) into any xml thus replacing the default overlays when <allowoverlay>no</allowoverlay> is set in the xml.

. **For further Information: See** Media Overlays

Topbar

The topbar displays at the top of most windows when the mouse hovers there, or when users select the up arrow key in most menus. It enables control of playback functions as well as exit MediaPortal, fullscreen video, home screen, etc.

There are two xml files which define the topbar:

- topbarhome.xml - defines the topbar in the MyHome window, and
- topbar.xml - defines it in all other windows.

Both topbars are fully 'skinnable'. Topbar.xml can automatically hidden or disabled in any xml using the window elements (before the <controls>): <autohidetopbar> (yes/no) and <disabletopbar> (yes/no) see Skin XML Structure for syntax.

See also Topbar

Control Types

Depending on the value of the control type tag in the control element above there will be additional tags specific to that type of control. A list of all available controls and a description on how to use them can be found on the Skin Controls page.

Example plugin with most controls implemented

If you want to see how the different controls look/work within MediaPortal download and install the plugin GuiControlsDemo, which can be found in the MP Repository.
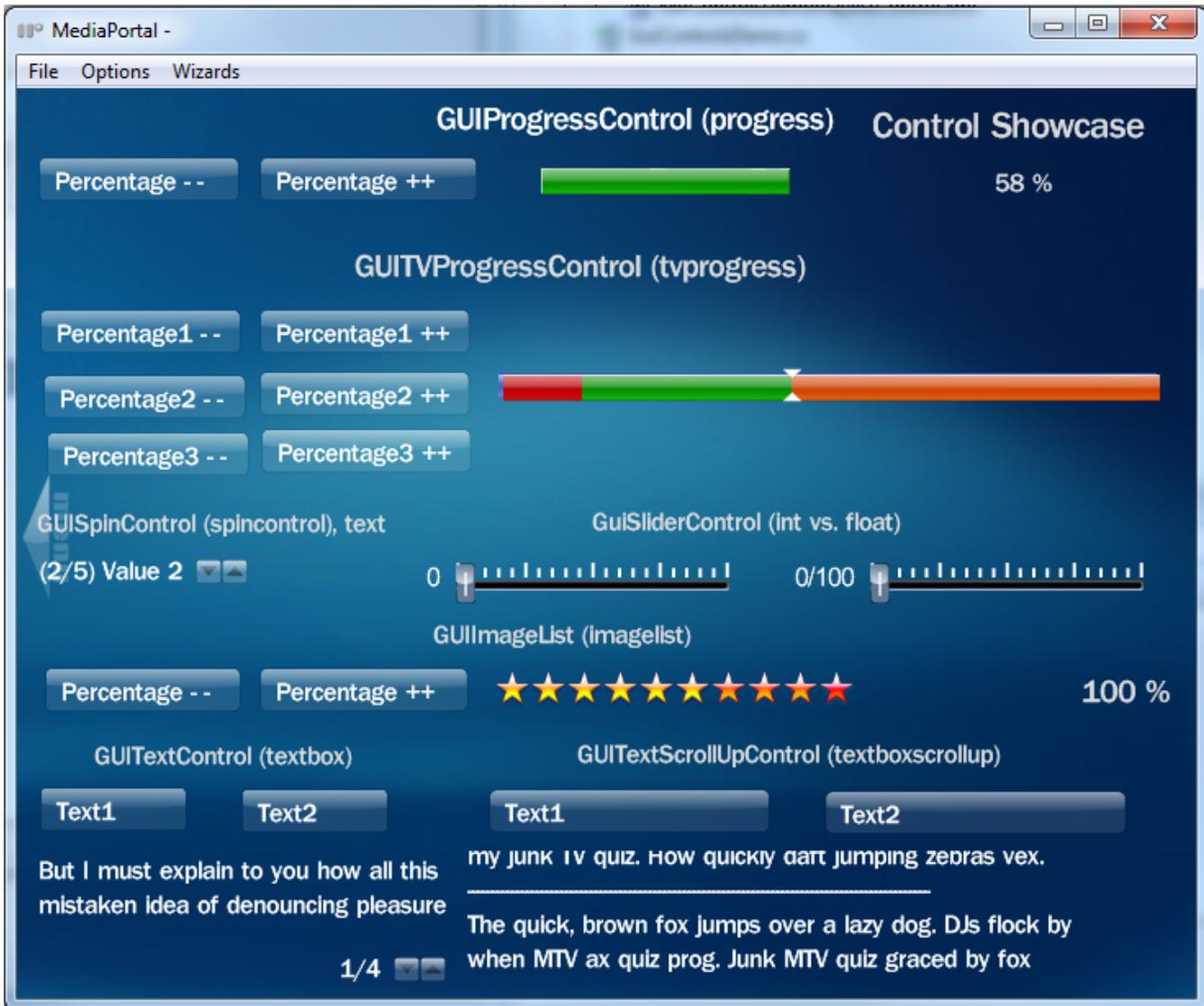
Detailled Description of GuiControlsDemo

Skin Related Changes

In virtually every MediaPortal release, changes for bug fixes, reworks and new skin features may require skin updates.

For information on the changes between versions:

- 1.0.0 to 1.0.1 Changes
- 1.0.1 to 1.1.0
- 1.1.0 to 1.2.0
- 1.2.0 to 1.3.0
- 1.3.0 to 1.4.0
- 1.4.0 to 1.5.0 Changes
- 1.6.0 to 1.7.0
- 1.8.0 to 1.9.0
- 1.9.0 to 1.10.0 Changes
- 1.10.0 to 1.11.0 Changes