Skinning with ViStart, the ***Windows Start Menu***

The purpose of this technical document is to explain how resources work within ***start menu*** and how you can customize those resources to be able to create your own unique themes for ViStart. This guide assumes you are using the default resources packaged within ViStart.

Basic Skining - Resources

The first thing you should do is navigate to your program files directory *(for 64bit users, ensure you navigate to the x86 version of your program files directory).* From there you should open the “ViStart” folder and you should see a “Resources” folder. In this folder there are many [Portable Network Graphic files](#) (.PNG).

You will be able to see these files when they are used on the ***start menu***. Using a digital graphics editor such as Photoshop you can edit these files to your liking. Don’t be afraid to resize these images, ViStart will try to automatically accommodate for different sizes.

Frames

You may notice that some of the files here have the same or similar image repeated a few times inside the one file. For example bottombuttons_arrow.png has 3 frames inside the 1 image. I did this to save having many files related to the same graphic. Each frame represents a different state of the arrow. Normal, Over, Pressed in that order vertically descending. The same philosophy applies to all other buttons in the ***start menu***.

The great mystery of the hidden graphics

You may have noticed that when you look at the Resources folder there are still some graphics you can’t find that appear on the ***start menu***. That’s because there are some resources that are actually embedded inside the ViStart executable. Don’t worry though, that doesn’t mean you can’t replace them! ViStart will check to see if the following files are available as a file in the Resources folder first, in the event it cannot find that resource it will default to the embedded resource.

Notice: As of build 4010, there will be no support for large icons. Instead large icons are converted to standard 32x32 icons so you don’t need to worry about making 48 icon versions of your resources

freq_rollover_32.bmp

Frequent programs rollover

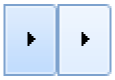
search_icon.bmp

The search icon used next to the "show all results" text.

rollover.bmp

The rollover image used on the program menu

jumplist_button.bmp



The end button that appears at the far right of an item on the frequent programs menu.

tree_seperator.bmp

The thin gray line that appears next to the search results category. like “Programs”.

jumplist_arrow.bmp

The arrow that appears on an item on the frequent programs menu if a jump list is available for it

startmenu_mask.bmp, When present it blurs the black area defined in the bitmap of the startmenu.png

Final Notice, In order to maintain backwards compatibility with older skins, ViStart will check if the skin has a “startmenu_expanded.png” file in it’s resources and in the event it does not the skin is viewed as a “ViStart 7” skin. This means certain features and graphics will be unavailable and unusable on your skin.

Layout.xml - The “LEGO” of ViStart

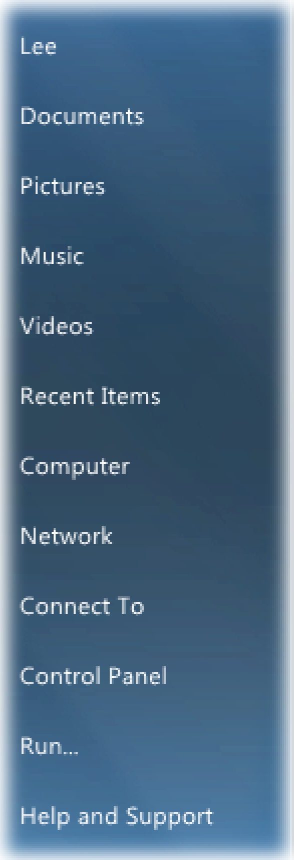
In order to understand how skins in ViStart ***start menu*** work you must at least have a basic understanding of XML. If you are not already familiar with XML I suggest you take a quick look at the XML section at w3schools.

*It’s important to note that if your skin does not provide a layout.xml file then ViStart will run in “ViStart 6” compatibility mode and that means elements will be positioned in a way in which they look like Windows **Vista start menu**.*

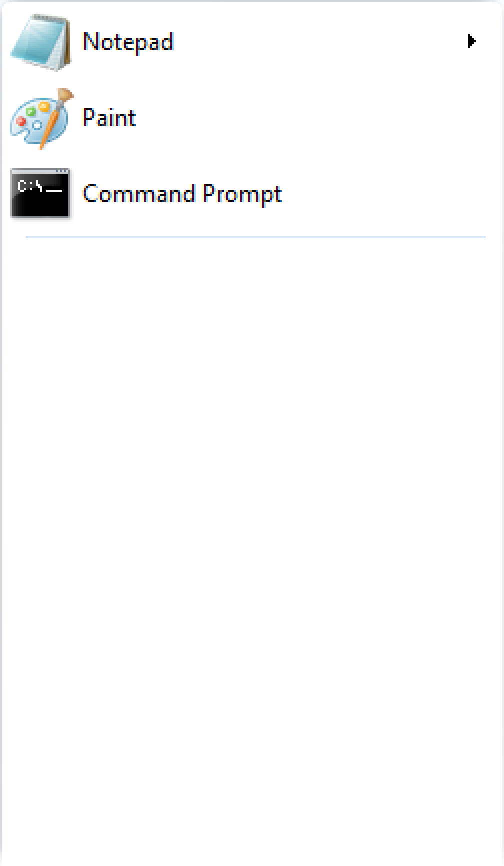
ViStart resource data is stored in the ViStart directory which is in the program files folder. The XML file we will be concerned with is the layout.xml file. This file essentially controls every aspect and every element in the start menu. ViStart may seem like 1 complete sophisticated start menu, but infact it’s made up from many smaller building blocks which are in themselves mini programs. For example, the `programmenu` where you see programs being listed. This is a component or an “Element” of ViStart. Let’s call these mini programs. “ViElements”. Let’s look at all the these ViElements in the ***start menu***.

rolloverplaceholder

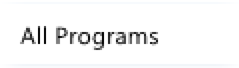
shutdown_button, shutdown_text



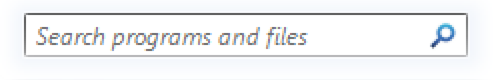
groupoptions



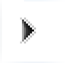
frequentprogramsmenu



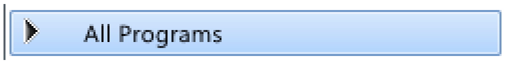
allprograms_text



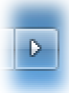
searchbox



allprograms_arrow

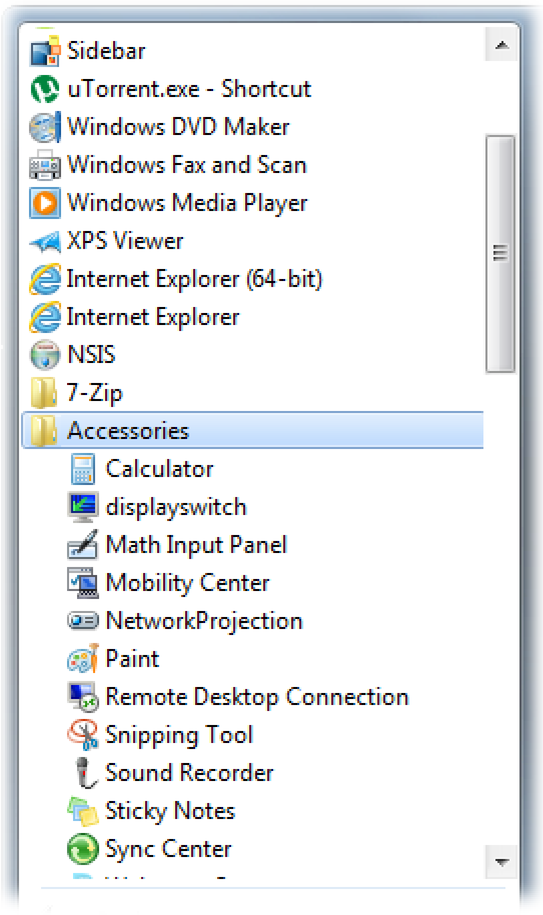


allprograms_rollover

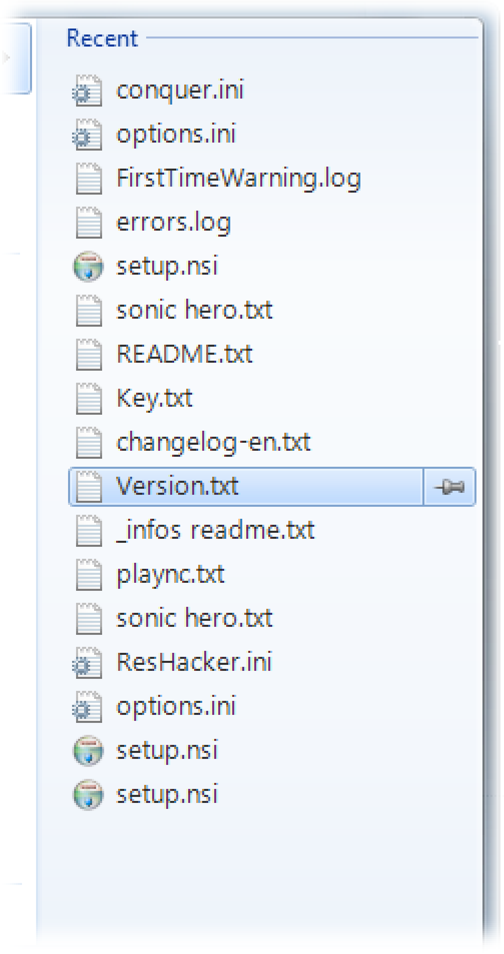


arrow_button

logoff_button



programmenu



jumplist_viewer

There are also some elements you can't see in the default windows start menu skin for ViStart but can still be used for your skin. At the time of writing, these are:

logoff_button - Originally part of the ViStart 6 series for when the original start menu had a separate log off button. *By default this button is not visible.*

Many of these ViElements share similar customizable attributes such as weather or not you want to make them visible. The ViElement XML type is used to describe a broad range of object types within ViStart so not every possible attribute may be applicable to every element. *For example, an image button such as the start orb (the button that summons the start menu) has no text so it will ignore any fonts assigned to it.*

ViElements

```
<ViElement
	width, integer.
	height, integer
	backcolour, webcolour
	font, string(ViFont id reference)
	visible, boolean
/>
```

The `width` attribute suggests a width for the ViElement to use. If you want the **`programmenu`** to be bigger you could adjust the **`width`** attribute accordingly. So you can probably guess what the `height` would do. `width` and `height` attributes are supported by the following elements
`programmenu`,
`jumplist_viewer`,
`searchbox` and
`frequentprogramsmenu`

The `backcolour` attribute allows you to set a back colour for the element. Many elements however are naturally transparent in nature. The only elements that support the `backcolour` attribute are
`programmenu`,
`searchbox` and
`frequentprogramsmenu`.

The `font` attribute enables you to specify a ViFont ID. ViFonts are elements you must define also within the layout XML file. ViFonts

The `visible` attribute enabled you to specify whether or not you want to hide or show various elements inside ViStart.

ViFonts

```
<vifont
	id, string
	face, string
	size, integer
	colour, webcolour
/>
```

ViFonts allow for much more customization than ordinary fonts and therefore should be thought of as an extended Font supporting more customization features like colour and size. There's no limit to the number of ViFonts you make and you have to name them yourself so you can reference them later when specifying the fonts for use with the ViElements.

The `id` attribute is the identifier for the ViFont. You should choose an ID that’s easy to remember because it acts as a reference for this ViFont. So for all elements that support a “font” attribute, they will require a ViFont ID if they are to be customized.

The `face` attribute is the actual Windows Font name, for example “Tahoma”. Be careful not to choose a font name that might not exist on the user’s system. If you do use a unique font you should include it with the skin installer. ViStart will use “Segeo UI” or “Tahoma” (subject to availability) if the specified font is unavailable on the local system.

The `size` attribute specifies the size of the text that will use this ViFont.

The `colour` attribute allows you to pick a fore colour for the text that will use this ViFont.

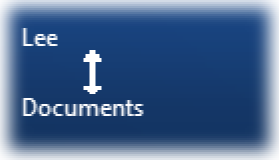
Additional Customization

The problem with ViElements is that they only define certain generic properties of elements within the start menu. However there are a number of elements which have additional customization available to them.

```
<viorb,
      fullheight_superbar, boolean
/>
```

`fullheight_superbar` dictates whether or not the start orb in ViStart will consider the full height of the taskbar. I suggest you play with this one depending on your needs.

```
<groupoptions,
      separator, integer
      limit, integer
/>
```



`separator` enables you to set the spacer between each option vertically.

`limit` allows you to set a limit on the number of options can be shown on the `groupmenu`. Failure to do this could result in options defined in settings.xml overflowing onto other parts of the your ***start menu*** skin.

```
<searchbox
      focuscolour, webcolour
      forecolour, webcolour
/>
```

`focuscolour` is the colour of the text inside the searchbox when the user has keyboard focus. *Keyboard focus is represented by a blinking | character.*

`forecolour` is the usual colour of the text inside the searchbox when the user's keyboard isn't focused on the search textbox

Distribution Guide

Many of the people who use the internet don't know how to extract files or they do but they can't be bothered in many cases. I recommend you make an installer that does the following *(this isn't mandatory but it makes your life easier, because there will be less complaints)*

I am in the process of making a program that will automatically wrap your skin so you can distribute it. I will update this part of the guide when it's ready.

```
Check Executable Exists "%programfiles%\ViStart"
If Not Exists
    Notice:: "ViStart isn't installed. Please go to http://lee-soft.com/vistart"
    Abort

Check Executable Version is equal or greater than the version you tested your skin with.
If Version is less than
   Notice:: "ViStart is out of date. Please go to http://lee-soft.com/vistart"
   Abort

Check if ViStart.exe is running as a process
If True
   Notice:: "File access error. Please make sure ViStart is closed"
   Loop Until Closed

Warn/Prompt user "%skin_name% is about to be installed and made the default skin for ViStart. Do you wish to continue?"

Delete ALL files in %programfiles%\ViStart\Resources
Check if any files exist in %programfiles%\ViStart\Resources\

If Exists
   Notice "File access error. Unable to delete a critical resource file."
   Add your installer to the registry to "Start" once at startup.
   Prompt User Restart
```

Abort

If your skin will overwrite the settings.xml file you should prompt the user that their ***start menu*** settings need to be changed.

Extract your skin
Execute the “ViStart.exe”