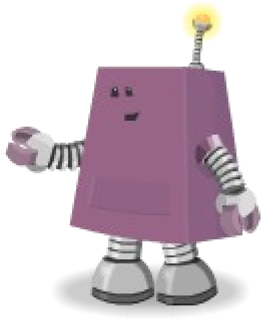
The HitchHiker's Guide



So What is it?

HitchHiker is a VB DotNet/ 3dsMax User Control and is designed to provide automatic thumbnailing of directories containing a variety of media types. This can be employed in a variety of situations as it is designed to provide a fast and simple method of building a fully functional asset browser or similar media browsing utility. HitchHiker only needs instruction of what directory to look in, and it will then build an interface that allows you see the folder's contents in a choice of different views. It is easy to configure the display layout according to presets accessible in the control, or it is fully customisable with a bit of scripting to look the way you would like. It's a DotNet equivalent to Rollout Creator. The advantage of using something like this is that the display logic is already built in, leaving the task of integrating it into 3dsMax a simple one, even for someone with very little scripting experience.

Usage

HitchHiker inherits the MaxUserControl class, a DotNet class that Autodesk added to 3dsMax as part of their DotNetSDK. It also uses the functionality of the ManagedServices.dll included in 3dsMax. This means that you can only use this user control in the 3dsMax environment.
Hitchhiker is free, all I would ask is that if you use it for something useful or cool, give a credit as to where it was from

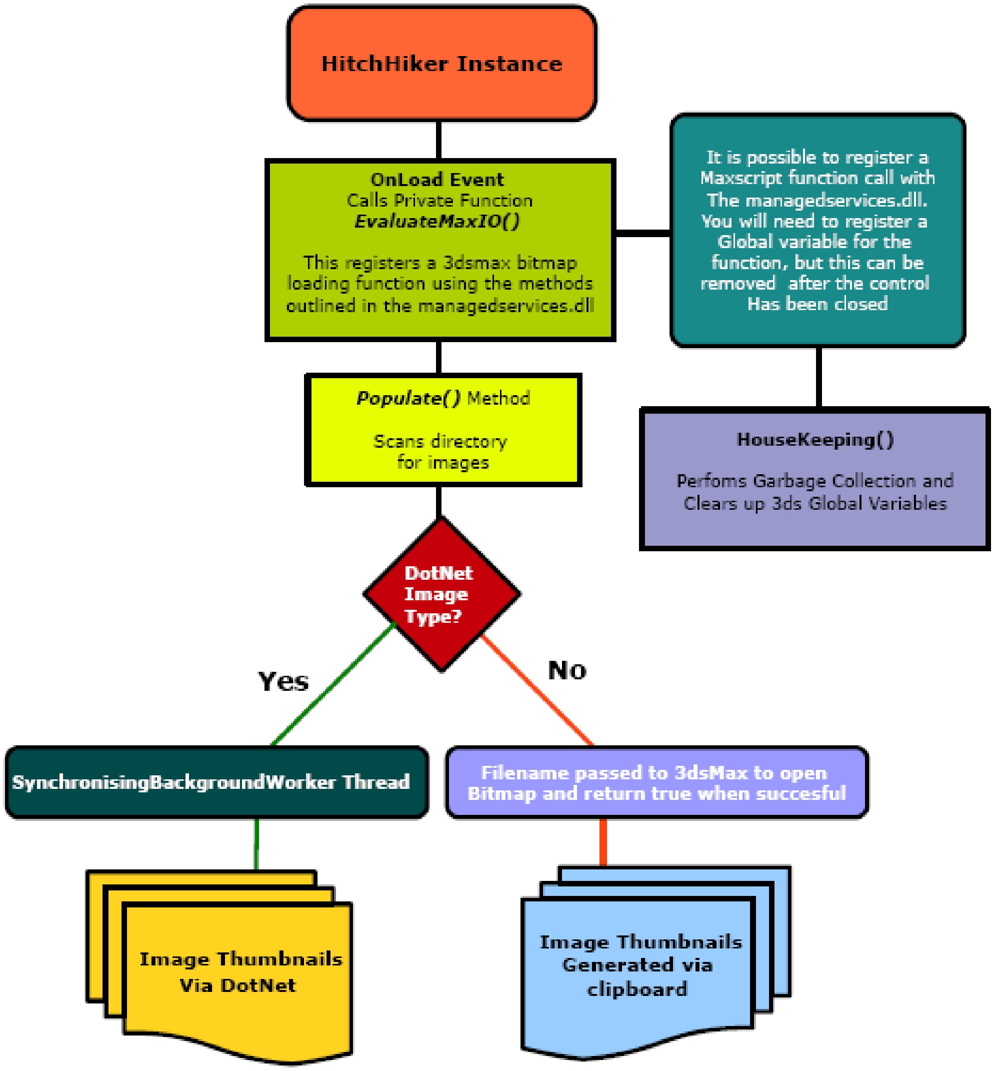
Supported Formats -

AVI	**BMP**	**JPG**
PNG	**PSD**	**TIF**
RPF	**TGA**	**WAV**

Third Party DotNet Image Libraries

If you are familiar with how windows uses various image file formats, you'll notice that there are several image types that windows will not open, but are useful to a 3D Artist. Most notably this is Targa, RPF and PSD. In order to open these file formats via DotNet, you would usually be tied to using a custom image library such as FreeImage or ImageMagick. This would add an extra deployment assemblies. However the biggest issue is when used on a 64 bit platform. I have found it difficult to find an x64 compatible image library that will let me integrate into Visual Studio and 3dsMax without a large configuration headache. With this in mind, HitchHiker uses a two step approach. Image formats that windows natively opens, like JPG, PNG and TIF are generated as part of a threaded GDI+ process within the control. These are subsequently fast to return, and keeps the UI thread of max separate. Non-windows formats are opened in 3dsmax and then passed back to the control as a bitmap where they are processed in a non threaded environment. The thumbnailing routine is still performed in the assembly rather than Max. You are simply using the 3dsMax Image readers to plug the gap in DotNet file types.
I would hope in the future, Autodesk would include a method for this in the DotNetSDK as it would significantly enhance custom control and UI development.
This is all made possible via the enhancements to the managedservices.dll. It struck me as odd that I was struggling with 3rd party image libraries, 3dsMax had the means to open these non-compatible formats. However it wasnt an option until 3dsMax 2010, when the MaxscriptSDK was introduced into the DotNetSDK. I have talked about this on my site before so you will hopefully be aware of some of the things you can do with it.

See the diagram for more details -


HitchHiker Instance
OnLoad Event
Calls Private Function
EvaluateMaxIO()
This registers a 3dsmax bitmap loading function using the methods outlined in the managedservices.dll
It is possible to register a Maxscript function call with The managedservices.dll. You will need to register a Global variable for the function, but this can be removed after the control Has been closed
Populate() Method
Scans directory for images
HouseKeeping()
Perfoms Garbage Collection and Clears up 3ds Global Variables
DotNet Image Type?
Yes
No
SynchronisingBackgroundWorker Thread
Filename passed to 3dsMax to open Bitmap and return true when succesful
Image Thumbnails Via DotNet
Image Thumbnails Generated via clipboard

How to Use HitchHiker

Hitchhiker is a DotNet UserControl and the easiest way to use it is on a rollout as a DotnetControl. The code for this is as follows -

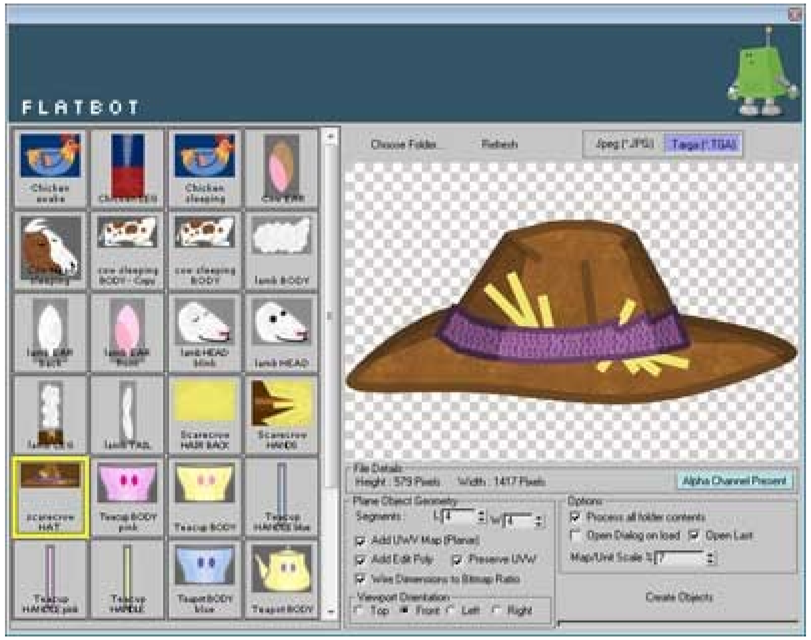
```
rollout HitchHikerRollout "" width:240 height:270
(
  dotNetControl HitchHiker "lonerobot.ui.character.HitchHiker" pos:[0,0] width:240 height:270

  on HitchHikerRollout open do
  (
          HitchHiker.filetype = (dotnetclass "lonerobot.ui.character.HitchHiker+filetypes").jpg
          HitchHiker.displaystyle = (dotnetclass "lonerobot.ui.character.HitchHiker+displaystyles").imagetext
          HitchHiker.populate @"C:\Dotnet\maxfiles"
  )

  on HitchHikerRollout close do HitchHiker.housekeeping()

  on HitchHiker thumbpicked sender args do
  (
          print args.FileName
          print args.Image
  )
)
createdialog HitchHikerRollout
```

It's actually quite straightforward. You call the ***Populate*** method with a path string. You then handle the clicked icon with the ThumbPicked event. This has a property called ***filename***, which is the currently selected file. (Should you wish to refer to this info outside the event you can get the filename from the ***hitchhiker.currentitem()*** property)

Methods

There are only a couple of actual public functions you can call on the control. You have already seen how you call ***populate(string)*** to set the control running. When this has been performed, you can call ***.getfiles()*** to get access to the file array in the same way you can in 3dsMax.

The other method available is ***CurrentImage(Multiplier as single)***
This returns an dotnet image object of the current file, useful when using HitchHiker with a picturebox control. Large images can be resized with interpolation to speed the loading of the return image, or to give a low-res preview. If you want the original image, call this function with a multiplier of **1.0**



File Details
Height : 579 Pixels Width : 1417 Pixels

Plane Object Geometry
Segments: L 4 W 4
[✓] Add UVW Map (Planar)
[✓] Add Edit Poly [✓] Preserve UVW
[✓] Wire Dimensions to Bitmap Ratio
Viewport Orientation: Top [● Front] Left Right

Options
[✓] Process all folder contents
[ ] Open Dialog on load [✓] Open Last
Map/Unit Scale % 7

There are a few properties to customise the layout of how HitchHiker displays the thumbnails. These are detailed next.

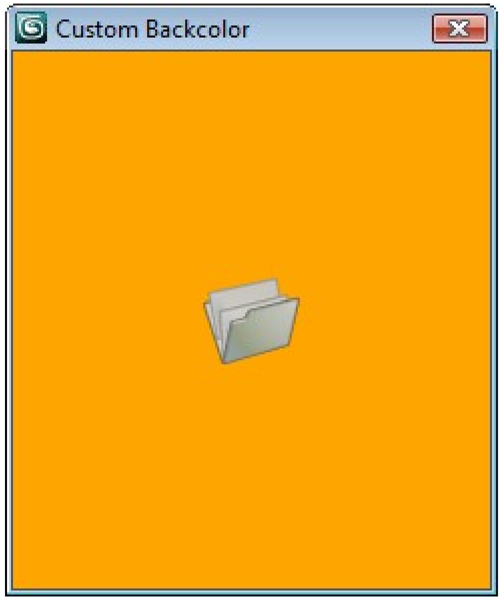
Control Properties

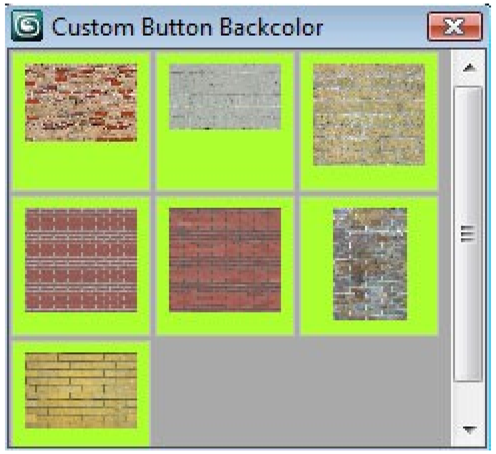
The following properties can be set on the control. For the examples, I am using the word HitchHiker to represent the instance of the control in your script, this is whether it is part of a 3dsMax rollout or a DotNet form class within 3dsMax.

.ButtonBackColor and .ThumbPanelBackColor

Property Type : System.Drawing.Color
MaxScript :
HitchHiker.ButtonBackColor = (dotnetclass "system.drawing.color").greenyellow
HitchHiker.ThumbPanelBackColor = (dotnetclass "system.drawing.color").orange

Changes the Button/Panel background colour independently to the rest of the control.





.CurrentItem
Property Type : String
MaxScript : HitchHiker.CurrentItem()

Returns the full path (including filename) of the current or last selected item.

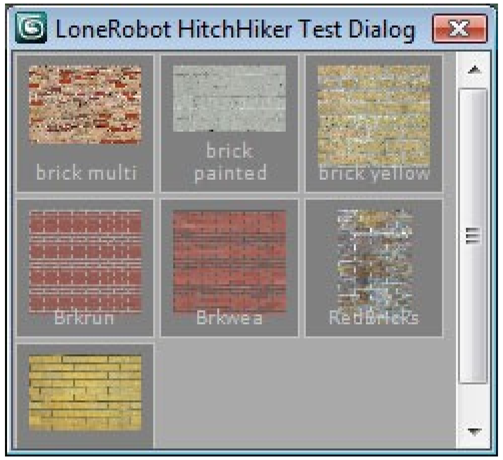
.CurrentMapFolder
Property Type : String
MaxScript : HitchHiker.CurrentMapFolder()

Returns only the path of the current or last selected item.

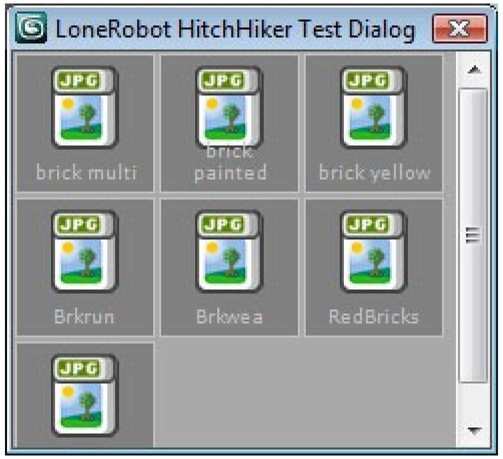
.DisplayStyle

Property Type : LoneRobot.UI.Character.HitchHiker.DisplayStyles (Enum)
MaxScript :

```
HitchHiker.DisplayStyle = (dotnetclass "lonerobot.ui.character.hitchhiker+displaystyles).images
HitchHiker.DisplayStyle = (dotnetclass "lonerobot.ui.character.hitchhiker+displaystyles).icons
HitchHiker.DisplayStyle = (dotnetclass "lonerobot.ui.character.hitchhiker+displaystyles).imageText
HitchHiker.DisplayStyle = (dotnetclass "lonerobot.ui.character.hitchhiker+displaystyles).icontext
HitchHiker.DisplayStyle = (dotnetclass "lonerobot.ui.character.hitchhiker+displaystyles).text
```



This is an Enum to set the style of the display. Note the use of the + sign when instantiating an Enum via Mascript. Enum will be highlighted in green for the rest of the document.



.FileType

Property Type : LoneRobot.UI.Character.HitchHiker.filetypes (Enum)
MaxScript :

```
HitchHiker.filetype = (dotnetclass "lonerobot.ui.character.hitchhiker+filetypes").tga
HitchHiker.filetype = (dotnetclass "lonerobot.ui.character.hitchhiker+filetypes").jpg
HitchHiker.filetype = (dotnetclass "lonerobot.ui.character.hitchhiker+filetypes").bmp
HitchHiker.filetype = (dotnetclass "lonerobot.ui.character.hitchhiker+filetypes").png
HitchHiker.filetype = (dotnetclass "lonerobot.ui.character.hitchhiker+filetypes").rpf
HitchHiker.filetype = (dotnetclass "lonerobot.ui.character.hitchhiker+filetypes").avi
HitchHiker.filetype = (dotnetclass "lonerobot.ui.character.hitchhiker+filetypes").wav
HitchHiker.filetype = (dotnetclass "lonerobot.ui.character.hitchhiker+filetypes").tif
```

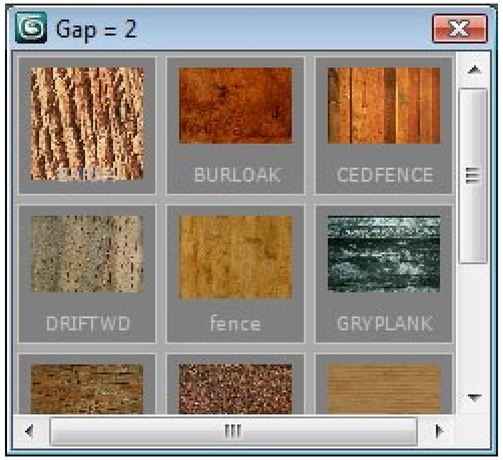
Enum to set the file type that HitchHiker searches for in the folder. If HitchHiker.RecursiveSearch is set to true, then Hitchhiker searches in all sub folders. If not, then it searches just the specified directory. When set to AVI, Hitchiker displays the frame halfway through the file, so that you avoid black frame thumb previews.

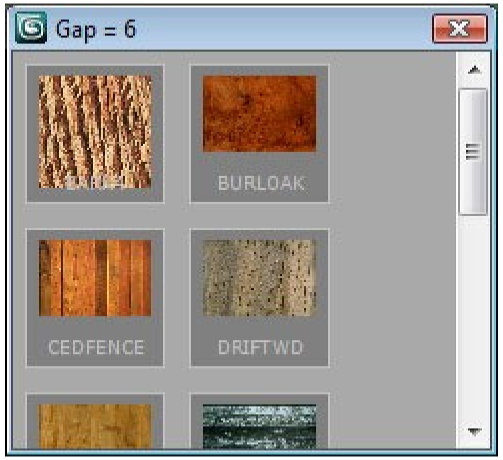
.Gap

Property Type : Integer
Maxscript : HitchHiker.Gap = 2

Value to set the spacing between the buttons on the HitchHiker panel. Property exposes the margin property of the embedded flowlayout panel.

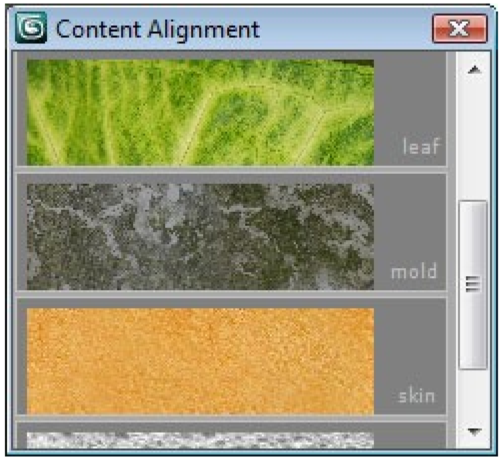
Default:0





.ImageAlign

Property Type : System.Drawing.ContentAlignment
Maxscript :
HitchHiker.ImageAlign = (dotnetclass “System.Drawing.ContentAlignment”).BottomCenter
HitchHiker.ImageAlign = (dotnetclass “System.Drawing.ContentAlignment”).BottomLeft
HitchHiker.ImageAlign =   (dotnetclass “System.Drawing.ContentAlignment”).BottomRight
HitchHiker.ImageAlign = (dotnetclass “System.Drawing.ContentAlignment”).MiddleCenter
HitchHiker.ImageAlign =   (dotnetclass “System.Drawing.ContentAlignment”).MiddleLeft
HitchHiker.ImageAlign =   (dotnetclass “System.Drawing.ContentAlignment”).MiddleRight
HitchHiker.ImageAlign = (dotnetclass “System.Drawing.ContentAlignment”).TopCenter
HitchHiker.ImageAlign = (dotnetclass “System.Drawing.ContentAlignment”).TopLeft
HitchHiker.ImageAlign = (dotnetclass “System.Drawing.ContentAlignment”).TopRight



Property for controlling the Image layout position on each button within the control.

.TextAlign

Property as above but for controlling the Text layout position on each button within the control.

.RecursiveSearch

Property Type : Boolean
Maxscript : `HitchHiker.RecursiveSearch = True`

Set to true to search subdirectories in the supplied directory. If set to False, only the main directory specifed is searched.
Default:False

.ReturnImageInterpMode and .ThumbInterpMode

Property Type : LoneRobot.UI.Character.HitchHiker.Interpolation (Enum)
Maxscript :

```
HitchHiker.ReturnImageInterpMode = (dotnetclass "LoneRobot.UI.Character.HitchHiker+Interpolation").Bilinear

HitchHiker.ThumbInterpMode = (dotnetclass "LoneRobot.UI.Character.HitchHiker+Interpolation").low
```

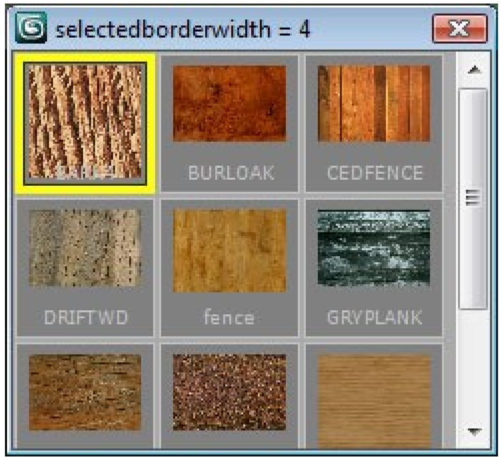
Other enum interpolation types -
.Bicubic
.Bilinear
.Def
.High
.HQBicubic
.HQBilinear
.Low
.Nearest

This property is identical to ThumbInterpMode. It decides what quality GDI bitmap is returned. Different settings for this can increase thumbnailing speed at the expense of quality. If using to make an image browser, you could experieemnt with setting the ThumbInterpMode to low and the ReturnImageInterpMode to bilinear. This way, the control would generate the thumbs as quickly as possible but allow the viewed image to be high quality.

.SelectedBorderWidth

Property Type : Integer
Maxscript : HitchHiker.**SelectedBorderWidth** = 2

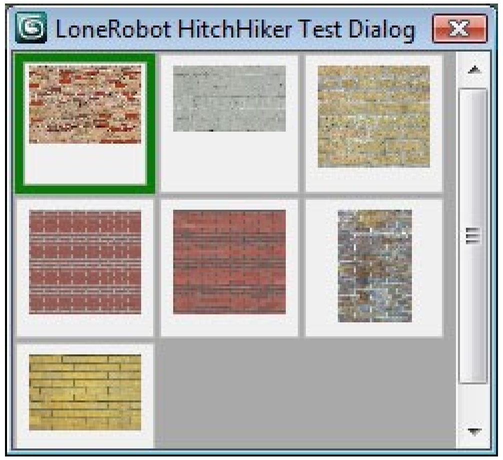
Value to set the width of the coloured outline around the active button on the HitchHiker panel.



.SelectedHighlight

Property Type : System.Drawing.Color
MaxScript : HitchHiker.SelectedHighlight = (dotnetclass “system.drawing.color”).yellow

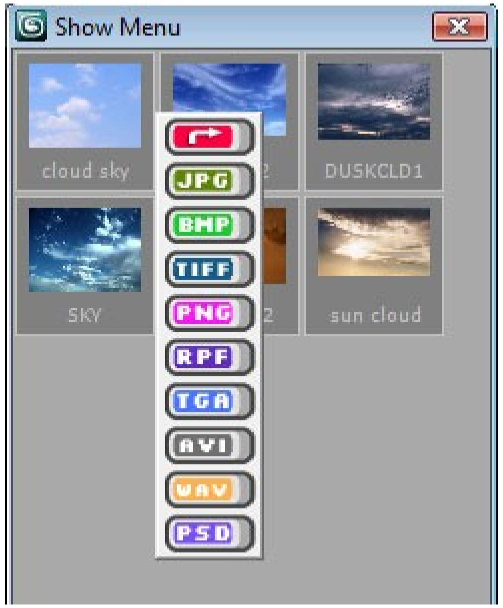
Value to set the colour of the border around the active button.



.ShowMenu

Property Type : Boolean
MaxScript : : HitchHiker.ShowMenu= true

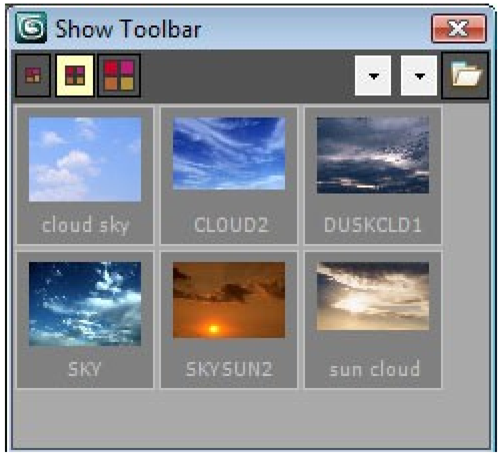
Enabled the display of the right-click menu to change the filetype and refresh the UI (The red icon at the top of the menu)



.ShowToolbar

Property Type : Boolean

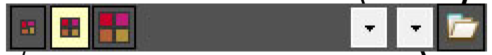
MaxScript : HitchHiker.ShowToolbar= true



The toolbar gives access to common operations within the control, and allows the user to change directory, and search and display parameters. It is small format so that when used in the Max command panel, it does not exceed the 160 pixel width for a control.

Choose a display mode from icons, icontext, images, imagetext and text

Change Directory



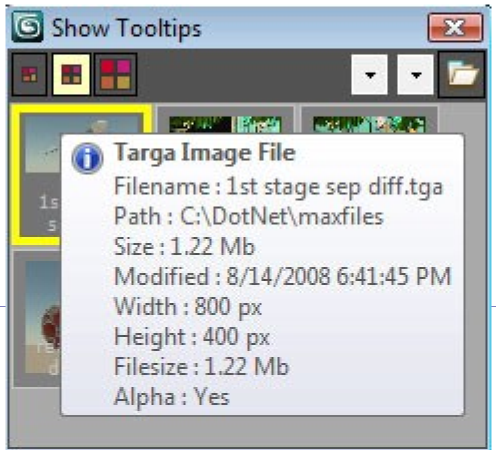
On the left, the three square buttons allow you to change the thumbsize to one of the thumbnail presets. Clicking the buttons in order will refresh the control with small, medium or large thumbnails. CTRL clicking the small and large thumbnail buttons will yield extra small and extra large thumbnails respectively. Thumbnails are not limited to these presets, as you can make them any size with the ***.thumbnails*** property.

Sort By Options

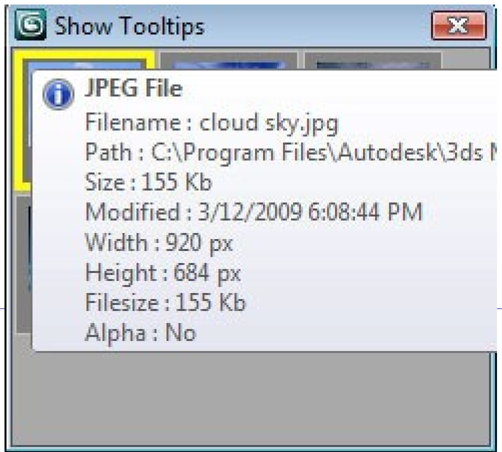
Here you can change the sorting options of the panel. See the ***.SortBy*** property for more details

HitchHiker Toolbar Options

.ShowTooltips



Targa Image File
Filename : 1st stage sep diff.tga
Path : C:\DotNet\maxfiles
Size : 1.22 Mb
Modified : 8/14/2008 6:41:45 PM
Width : 800 px
Height : 400 px
Filesize : 1.22 Mb
Alpha : Yes



JPEG File
Filename : cloud sky.jpg
Path : C:\Program Files\Autodesk\3ds M
Size : 155 Kb
Modified : 3/12/2009 6:08:44 PM
Width : 920 px
Height : 684 px
Filesize : 155 Kb
Alpha : No

Property Type : Boolean
MaxScript : HitchHiker.ShowTooltips= true

Displays detailed image information. Look, TGA previews in DotNet! :-)

This property calls a function that again uses managed services to gain image information from the MaxScript method GetBitmapInfo(). The information is then passed back to the dotnet assembly as a float, string, integer or boolean value according to the property.

.Sortby

Property Type : LoneRobot.UI.Character.HitchHiker.SortByOptions (Enum)

MaxScript :

```
HitchHiker.sortby = (dotnetclass "LoneRobot.UI.Character.HitchHiker+SortByOptions").FileName
HitchHiker.sortby = (dotnetclass "LoneRobot.UI.Character.HitchHiker+SortByOptions").LastWriteTime
HitchHiker.sortby = (dotnetclass "LoneRobot.UI.Character.HitchHiker+SortByOptions").FilesizeAscending
HitchHiker.sortby = (dotnetclass "LoneRobot.UI.Character.HitchHiker+SortByOptions").FilesizeDescending
```

Implements an Icomparer interface to provide custom sorting patterns for the resulting file array.

.TextImagerelation

Property Type : System.Windows.Forms.TextImageRelation

MaxScript : Hitchhiker.TextImagerelation = (dotnetclass “Windows.Forms.TextImageRelation”).ImageAboveText

Another method of text layout in the button controls. This property wraps the button property of the same name. For more control of the layout of the text and image use the ***.textalign*** and ***.imagealign*** properties

.Thumbnails

Property Type : LoneRobot.UI.Character.HitchHiker.ThumbNailSize (Enum)

MaxScript :

```
Hitchhiker.Thumbnails = (dotnetclass “LoneRobot.UI.Character.HitchHiker+ThumbNailSize”).Tiny
Hitchhiker.Thumbnails = (dotnetclass “LoneRobot.UI.Character.HitchHiker+ThumbNailSize”).Small
Hitchhiker.Thumbnails = (dotnetclass “LoneRobot.UI.Character.HitchHiker+ThumbNailSize”).Medium
Hitchhiker.Thumbnails = (dotnetclass “LoneRobot.UI.Character.HitchHiker+ThumbNailSize”).Large
Hitchhiker.Thumbnails = (dotnetclass “LoneRobot.UI.Character.HitchHiker+ThumbNailSize”).ExtraLarge
```

Sizes are as follows and are square.

Tiny = **24x24**
Small = **50x70**
Medium = **70x70**
Large = **90x90**
ExtraLarge = **150x150**

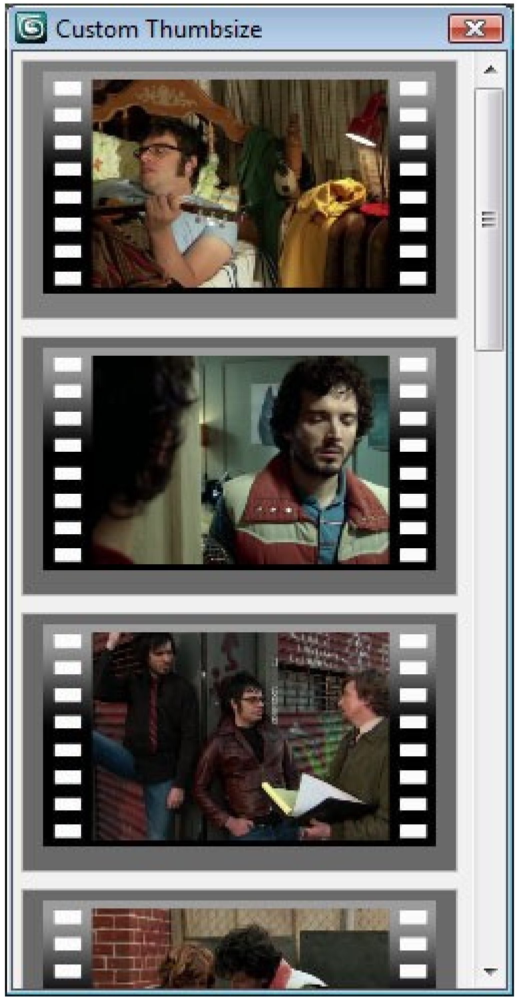
These are to use as quick presets and are also used by the thumbnail buttons on the toolbar. To specify a custom thumbnail size, use the ***.thumbsize*** property

.Thumbsize

Property Type : System.Drawing.Size

Maxscript : `HitchHiker.thumbsize = dotnetobject "system.drawing.size" 200 60`

Set a custom thumbnail size using this property. Useful if you want to show a larger thumbnail, for example when previewing video files. You may have noticed the filmstrip surround on the rollout. This was something I added to the video thumbnails so that it mimicked the way Vista displays thumbnails in explorer.



That's it! I hope you find this control useful.

If you have any questions about usage, or any bugs/suggestions, please mail me through the website contact page.