DirectSmile Image Personalization API

(COM-Interface)

DirectSmile GmbH
Torstraße 164
10115 Berlin

Tel.   +49 (0)30 627 77 0
Fax   +49 (0)30 622 85 65
www.directsmile.de

General

Name
Filename: DirectSmile_ImageAPI.dll
ClassID: {89A834F2-4A18-485D-ABCD-309FB1F3A100}
Location: c:\windows\System32\DirectSmile
Version: 1.0.0.19
Type: ActiveX/COM

Description
The DirectSmile ImageAPI is an ActiveX dll, which offers a simple communication to the DirectSmile Generator or Picserver to render images.

The dll uses UDP to communicate one to one against the DirectSmile Generator or Picserver.
It holds an COM Interface to provide a simple access by other programs.
The communication between the dll and the Generator or Picserver is with intent very simple. It returns an image or an error message if the rendering failed.
It also returns an error message if the Generator or Picserver is in process with another request. For example, if the Picserver gets two different requests it starts the rendering in the moment the first requests arrives and returns the message "BUSY with another job" to the second caller. Actually this is not an error, it's just a process message. The Generator or Picserver has no queue to store requests, and the request must be send again until the Generator returns an Image.

Synchronous and asynchronous calls

By default the call is synchronal, but you can change the behavior to an asynchronous call. In that case you have implement the events ImageReady and Error. The Error event will be fired if an error occurs in the image rendering process and the ImageReady, if a new image is done.

Methods

Create

Sends a rendering request to a DirectSmile Generator or Picserver.

Syntax

Create()

Return Values

If	Create returns
Create successful	Boolean expression, true

CallBack

Internal method will be called asynchronously by the DirectSmile Generator or Picserver.

Syntax

CallBack(String**)**

The **CallBack** function syntax has these arguments:

Part	Description
String	Required. **String** expression, containing the status message, created by the DirectSmile Generator or Picserver.

Properties

Compression

Gets or sets the JPG compression of the rendered image as percentage.

Syntax

*Object.***Compression** [=Integer]

The **Compression** property syntax has these parts:

Part	Description
Integer	Required. Integer expression, defining the compression.

ErrorMsg

Gets an error message thrown by the DirectSmile Generator or Picserver.

Syntax

*Object.***ErrorMsg** [=String]

The **ErrorMsg** property syntax has these parts:

Part	Description
String	Read only. String expression. Is either empty or contains a human readable error message.

ExecutionMode

Gets or sets the behavior how the DirectSmile Generator or Picserver should asynchronal or synchronal respond the request.

Syntax

*Object.***ExecutionMode** [=DSMExecutionModeE]

The **ExecutionMode** property syntax has these parts:

Part	Description
DSMExecutionModeE	DSMExecutionModeE expression, refer Enums

ImagePixelWidth

Gets or sets the width in pixel of an image to be rendered by the DirectSmile Generator or Picserver.

Syntax

Object.**ImagePixelWidth** [=Long]

The **ImagePixelWidth** property syntax has these parts:

Part	Description
Long	Long expression, width of the image to render in pixel

JPGFileName

Gets or sets the filename of an image to be rendered by the DirectSmile Generator or Picserver.

Syntax

Object.**JPGFileName** [=String]

The **JPGFileName** property syntax has these parts:

Part	Description
String	String expression, full path and filename of the rendered image or the image that should be rendered.

Key

No longer supported.

NeededSeconds

Gets the time in seconds it took the DirectSmile Generator or Picserver to create the requested image.

Syntax

*Object.***NeededSeconds**[=Single]

The **NeededSeconds** property syntax has these parts:

Part	Description
Single	Read only, Single expression, time the image generation took.

SetName

Gets or sets the name of the DirectSmile Set which is used image generation by the DirectSmile Generator or Picserver.

Syntax

*Object.***SetName**[=String]

The **SetName** property syntax has these parts:

Part	Description
String	String expression, name of the used DirectSmile Set.

TextInPicture

Gets or sets the text that is rendered on the image by the DirectSmile Generator or Picserver.

Syntax

*Object.***TextInPiture**[=String]

The **TextInPiture** property syntax has these parts:

Part	Description
String	String expression, text to render.

TimeOutInSeconds

Gets or sets the maximum time in seconds the generation process can take until it will be canceld.

Syntax

*Object.***TimeOutInSeconds**[=Long]

The **TimeOutInSeconds** property syntax has these parts:

Part	Description
Long	Long expression, maximum time in seconds for an image to get processed.

WatermarkType

Gets or sets the watermark which is placed on the rendered image. The default is the DirectSmile watermark. Watermarked images can have in maximum 600 pixel (horizontal or vertical). If you turn of the Watermark , your system will charge you 0.2 Smilys for images <= 600 Pixels and 1 Smily for Images > 600 pixel

Syntax

*Object.***WatermarkType**[=DSMWaterMarkTypeE]

The **WatermarkType** property syntax has these parts:

Part	Description
DSMWaterMarkTypeE	DSMWaterMarkTypeE expression, refer Enums.

Events

_Error

Fires if an error occurred while rendering the image.

Syntax

Private Sub Obj_Error(sError As String)

The **_Error** event syntax has these parts:

Part	Description
sError	String expression, containing a human readable error message.

_ ImageReady

Fires if the generation of an image has finished.

Syntax

Private Sub Obj_ImageReady(FileName As String)

The **_ImageReady** event syntax has these parts:

Part	Description
FileName	String expression, containing the path and filename of the rendered image.

Enumerations

DSMExecutionModeE

Constant	Value	Description
dxcm_Hold_Till_Image_Is_Finished	0	**Create**() method will be call synchronous and the program will wait until the requested image is created.
dxcm_Release_And_Fire_Events	1	**Create**() method will be call asynchronous. Either the **_ImageReady** or **_Error** event will be fired.

DSMWatermarkTypeE

Constant	Value	Description
dwmt_DirectSmile	1	Default**,** place a DirectSmile watermark on the image.
dwmt_Generic	2	Place a generic watermark on the image..
dwmt_None	0	Do not place a watermark on the image. (Smilys will be charged)

Samples

The Create() method in VB 6

Description

The sample creates a new instance of the DirectSmile_ImageAPI, sets the necessary properties and calls the Create method. By default the access is synchronal, that's why the program stops until the Create method releases. If the ErrorMsg property is not empty we show the error message, else we load the image, by passing the filename to the LoadPicture function.

Setup

Open a new Project. Open references and add the reference to the DirectSmile_ImageAPI. Place a Command button and an image control on to the form. Doubleclick on the Command1 button and paste the following code in to the Click event function.

```
Private Sub Command1_Click()
     Dim p As New DirectSmile_ImageAPI.DSMPicture
     p.JPGFileName = "c:\test.jpg"
     p.SetName = "\sets\dsm_balloons\dsm_balloons.dset"
     p.TextInPicture = "test"
     p.Create
     If p.ErrorMsg <> "" Then
          MsgBox p.ErrorMsg
     Else
          Set Me.Image1.Picture = LoadPicture(p.JPGFileName)
     End If
     Set p = Nothing
End Sub
```

The Create() method in VB.NET

Description
The sample creates a new instance of the DirectSmile_ImageAPI, sets the necessary properties and calls the Create method. By default the access is synchronal, that's why the program stops until the Create method releases. If the ErrorMsg property is not empty we show the error message, else we load the image, by passing the filename to the LoadPicture function.

Setup
Open a new Project. Add a COM reference to the DirectSmile_ImageAPI. Place a Command button and an image control on to the form. Doubleclick on the Button1 and paste the following code in to the Click event function.

```vbnet
Private Sub Button1_Click(ByVal sender As System.Object, _
                          ByVal e As System.EventArgs) Handles Button1.Click
        Dim p As DirectSmile_ImageAPI.DSMPicture
        Dim ErrMsg As String
        Dim MyImage As Bitmap

        p = New DirectSmile_ImageAPI.DSMPicture
        p.JPGFileName = "c:\test.jpg"
        p.SetName = "\sets\dsm_balloons\dsm_ballons.dset"
        p.TextInPicture = "test"
        p.Create()
        If p.ErrorMsg <> "" Then
                MsgBox(p.ErrorMsg, MsgBoxStyle.Exclamation + MsgBoxStyle.OKOnly, _
                "DirectSmile ImageAPI")
        Else
                PictureBox1.SizeMode = PictureBoxSizeMode.CenterImage
                MyImage = New Bitmap(p.JPGFileName)
                PictureBox1.Image = CType(MyImage, Image)
        End If
        p = Nothing
End Sub
```