

API for the Alpha Pro COM object

You can use the features of Alpha Pro from your software written in Visual BASIC using the COM interface. The easiest way is to use the VBScript language built into the MS Internet Explorer by creating the HTML pages containing forms and VBScript routines.

The VBasic statement

```
Set AlphaProObj=CreateObject("AlphaPro.Document")
```

Means that the new AlphaPro "document" will be created and the program itself will be started if necessary. The newly created document is invisible to the end-user but is ready to accept commands from the COM interface.

Following are the methods available through COM interface:

long GetStatus() **player statue**

0 - the player is not active

1 - the player is busy executing the command

2 - the player is ready to accept a new command

void AssignValue(LPCTSTR Name, LPCTSTR Value)

assign the **Value** to the "environment variable" called **Name**

BSTR GetValue(LPCTSTR Name)

get the value of the "environment variable" **Name**

some "environment variable" have a predefined function

%PLAYERSTATUS	player status, like GetStatus()
%TIMECODE_LTC	LTC timecode if available, (-1) otherwise
%TIMECODE	timecode, relative to the start of the script
%TICK	time in 1/18th of second, from the Windows startup
%TIME	time of the day in seconds, from 00:00:00
%HOUR	hour of the day
%MINUTE	minute of the day
%SECOND	second of the day

The variables, which names begin with % (percent sign) are common for all of the objects of "AlphaPro.Document" type created by different applications running on the same computer. This makes it possible to transfer some data from one VBScript routine to another.

The **GetValue** function returns the empty string value "" if the variable with that name does not exist. The variable becomes existent if some value has been assigned to it by the **AssignValue** function. If the script opened in the AlphaPro.Document object (via "load" command) contains templates with the same name as the variable, all changes made in the variable will be reflected in the template. For the templates of "Runtime" type, the **AssignValue** function leads to immediate update of the area of the screen containing the template. The templates of "Render-time" type when changed, flag the entire page as Modified, so it will be re-rendered at playback.

long ExecCommand(LPCTSTR CommandLine)

execute the command-line, the syntax is the following

load filename.sc

load the script from the file specified

stop

stop the script, same as Esc

renderpages 5 10

render the pages from 5 to 10 inclusive

there is the third optional parameter, if it is set 1, like "**renderpages 5 10 1**" the execution will start in "Start Through Pause" mode

playpages 5 10

play the pages from 5 to 10 inclusive, rendering, if needed

pause

pause execution of the script

resume

resume execution

take 5

display the contents of the page specified as a static image

textout var_name style_name x y

display the text, contained in the variable **var_name** using the style, taken from the named style **style_name**

fadetext var_name style_name x y duration

display the text, contained in the variable **var_name** using the style, taken from the named style **style_name** using the fade in effect with the specified duration

trytext var_name style_name

prepare the characters of the style "**style_name**" contained in the text of the variable **var_name**

drawpicture bitmap_file_name x y width height

drawblurredpicture bitmap_file_name x y width height

display the picture from the file **bitmap_file_name** in the screen area defined by the coordinates **x y width height** . If the filename begins with "*" (asterisk), the name is the name of a memory-mapped file, with first four bytes defining the length of the file.

drawbox style_name x y width height

Display a rectangle filled with style **style_name**

faster

make scrolling faster

slower

make scrolling slower

drawavi avi_file x y repeat_count

play an animation

=====

Note on using the templates

The **Name** of the template defines the variable name the template is linked to.

Alignment, Vertical Alignment, Vertical Spacing define the formatting for output. The Style is defined by the **Style** field. The template text may contain CRLF characters. **Unlimited** alignment type makes it possible to use the template in the crawling line, make the line longer if the template does not fit.

Templates can be Run-Time or Render Time, depending on how they react to the changes in the variable linked with the template.

For the **Run-time** templates, any change to the variable is displayed directly on the screen, if the "template name"_visible variable contains anything but "0". Otherwise no visible changes occur. The "template name"_visible will be set to *n* automatically when the page number *n* is displayed and then to zero when it is overlaid by another page.

=====