Stream Labs Corporation

Alpha Pro 2.0

User Manual

V1.0

July 2004

Content

Document History

Date	Issue	Version	Author
July 6, 04	Created	1.0	

The latest information about the *Alpha Pro* software, and professional television symbol generating systems can be found at our Web-site: http://www.alpha-pro.com

The following denotation for the commands accessible from the menu is used in this manual: ***Video/Preferences/Player/Start Through Pause*** denotes the ***Start Through Pause*** option, working in the ***Player*** sub-window of the ***Preferences*** windows, called by the command of the same name in the ***Video*** menu.

Chapter 1 General Overview of Program Features

The program is used for creating **scripts** (sequences of titles output with various effects) and playing them on a device for superimposing computer graphics over television video signal. Hereinafter, such device is briefly called **videocard**.

Each script can be broken into an unlimited number of pages. Each such page can be assigned a video effect, which defines the method of the page appearance, movement and fading out from the video screen.

Video effects

The following types of video effects are supported:

- Vertical upward scrolling of an infinite page (drum) – *Roll*;
- Horizontal scrolling of an infinite page from right to left (creeping line) – *Crawl* ;
- Passage of a page of a size of the video screen from upwards or from right to left – *Reveal*;
- Static page appearing effect – *CrossFade*;
- Standard or customized *Wipe* effect executed with a mask (preset by the gradient file) with soft boundaries - *Alpha Gradient Wipe*;
- Animation played on a static page - *Animation*.

The following parameters for video effects can be preset:

- The effect rectangular area;
- The effect rate or duration;
- The duration of the pause for the page content display;
- Whether the effect is executed up to the end or not (*complete*);
- Whether the page is output over the graphics present on the screen, or substitutes it (*opaque*).

Text in a page

The text inside a page is broken into paragraphs. Each of them can be assigned individual and independent values of layout attributes. These attributes are as follows:

- Right and left indentations, specifying boundaries of the paragraph;
- Formatting mode (left/right/center/justify alignment).

Besides, it is possible to adjust the line spacing in a paragraph, which allows to pull the lines closer together or wider apart.

Frames

Several paragraphs can be placed inside a text frame located arbitrarily on the screen. The values of paragraph indentations in this case are measured from respective frame edges. When the frame text is added or edited, the frame vertical height changes automatically.

It is also possible to set a rectangular area with graphic filling (background), layout and size of which are defined by the boundaries of the *Box* type frame. Filling of the area can be performed with the same effects as for normal text characters.

Picture and *Animation* frames of type are provided for allocating graphics in a page. They allow to include images from *BMP* or *TGA* files, which can comprise an alpha-channel. An *Animation* frame can be assigned with a sequence of *TGA* files or an *AVI* file, if an appropriate *CODEC* is installed in the system. Such frames are used on pages with creeping line or drum effects.

An image from a *BMP* or *TGA* file (which may comprise and alpha channel) can be used as a page background (*Background*).

Text, graphics, animation, and background area frames are placed in layers located above the background layer. The order of frames along the *Z* axis is initially determined by the order of their creation. This order can later be modified arbitrarily.

The size of any frame (except an animation one) can be changed. Frames can be freely moved within a page and mutually superposed. Pixels outside text characters and background areas are transparent. The transparency of graphics and animation frames is defined with the color keying mechanism, or through the alpha-channel.

Attributes of text characters and background frames

Any fragment of the text, down to an individual character, or a background area frame can be assigned specific style attributes subdivided into following groups:

1. Internal area outline, layout, and filling;
2. Border area type, width, and filling;
3. Shadow area, or 3D side surface area, type, "depth", direction, and filling;
4. Character drawing quality and area boundaries smoothing filters.

Group 1:

- Font – a name of a font installed in the *Windows* environment (*Arial, Times New Roman*, etc.);
- Font style – normal, bold, italic, or bold italic;
- Character height and width expansion/condensation factor in pixels;
- Color filling of character internal area or background areas as a color gradient or texture;
- Image element transparency with respect to other graphics layers or video signals;
- Position variations for string characters – upward or downward shift with respect to the base line;
- Character spacing variation.

Group 2:

- Border area width in pixels;
- Border area style (solid, extrusion, or diffusion);
- Border area shape (round, square, rhombic, spherical, conic, or pyramidal);
- Special border style (double-sided or complementary);
- Border area color filling as a color gradient or texture.

Group 3:

- Shadow area, or 3D side surface area, "depth" in pixels;
- Area style (regular falling shadow – with "hard" or "soft" boundary, side surface with solid filling, with "displacement", or with filling separately accounting for height and directions);
- One of the eight directions of the area offset (accurate to 45°);
- Border area color filling as a color gradient or texture.

Group 4:

- Bitmap quality parameter;
- "Blurring" degree used to reduce jitter of thin horizontal lines.

The entered values of attributes can be saved as a style under some name, to be used later.

Defining a narrow background area at the bottom or the middle of a fragment or a text paragraph allows to produce an *Underline* or *Overstrike* effect.

A logo image can be inserted directly as a text character, by specifying the respective graphics file name. It is also possible to specify a vertical shift with respect to the string baseline, and a horizontal shift, which allows to superpose following string characters over the image. Moreover, it is possible to scale the image to definite vertical and horizontal sizes.

A text fragment with appropriate attributes of pages, characters, paragraphs, and frames can be copied to *Clipboard,* to be later inserted at any place of the current or any other script. A text stored in the *Clipboard* from another *Windows* application can also be pasted into the script.

Script replay and rendering

Script replay is launched by the ***Start*** command, and can be interrupted at any moment by the ***Stop*** command, which also clears the screen. Titles replay can also be paused (command ***Pause***). Effects can be executed in a step-by-step mode (command ***Next***), and the creeping line and drum effects execution can be accelerated or slowed down by the ***Faster*** and ***Slower*** commands, respectively.

The step-by-step title output mode can be controlled by a *GPI* interface.

To start a script replay without a delay, at an exactly specified moment, it is recommended to render images of graphics elements in advance, using the ***Render*** command. There also exists a mode allowing to output and edit titles simultaneously, which can be necessary for live broadcasting. This mode allows to output only some of the script pages, rather than the whole script.

The ***Render*** command only re-calculates graphics images of the script pages, which were modified. Processing results are automatically saved between the program (computer) work sessions. A dedicated font and file cache is provided for acceleration of script editing operations.

Editing process

The changes of the page text can be output to the screen through a videocard immediately (command ***Take***). If the ***Video/Display...*** option is active, the current page content is output to the videocard continuously, right in the course of input and editing.

The most realistic-looking output of the script units in edit windows of the program is yielded by the *Hi-Color* and *True-Color* operation modes of the *Windows* environment. A simplified output mode (***Draft***) can be installed for a fast, parallel to text editing, display in the program window. Sizes of graphic elements displayed in the window can also be reduced 2 or 4 times (***Zoom***), which can be handy for work with multiple scripts or large pages.

Chapter 2 The Structure of Program Scripts

2.1. Script Storing and Related Files

***SC*-files**

Scripts are saved by the program in files with the *.SC* extension. Such script file can be edited and takes very little of the disk space. If any graphics or animation files are used in the script, references to their file and path names are only stored in the file. Actually, although these graphics files are stored separately from the script, they make part of it. Saving the relative references allows to move scripts from one disk to another along with the related graphics files, saving the subdirectory structure.

Processing results

The ***Video/Render*** command processes a script prior to the output to the videocard, creating a series of binary files. These files are created in the *Cache* folder of the directory where the *Alpha Pro* program has been installed. These files are saved between the program (or the computer) work sessions, which saves the necessity of re-processing the script if it had not changed.

Tip. Files in the *Cache* folder are not necessary for the respective scripts. As they can take quite a lot of the disk space, it is possible to periodically delete **all files** in this folder, for example, when certain old scripts are not worked with anymore. For other scripts, if required, the ***Render*** command can be re-executed.

2.2. Splitting a Script into Pages

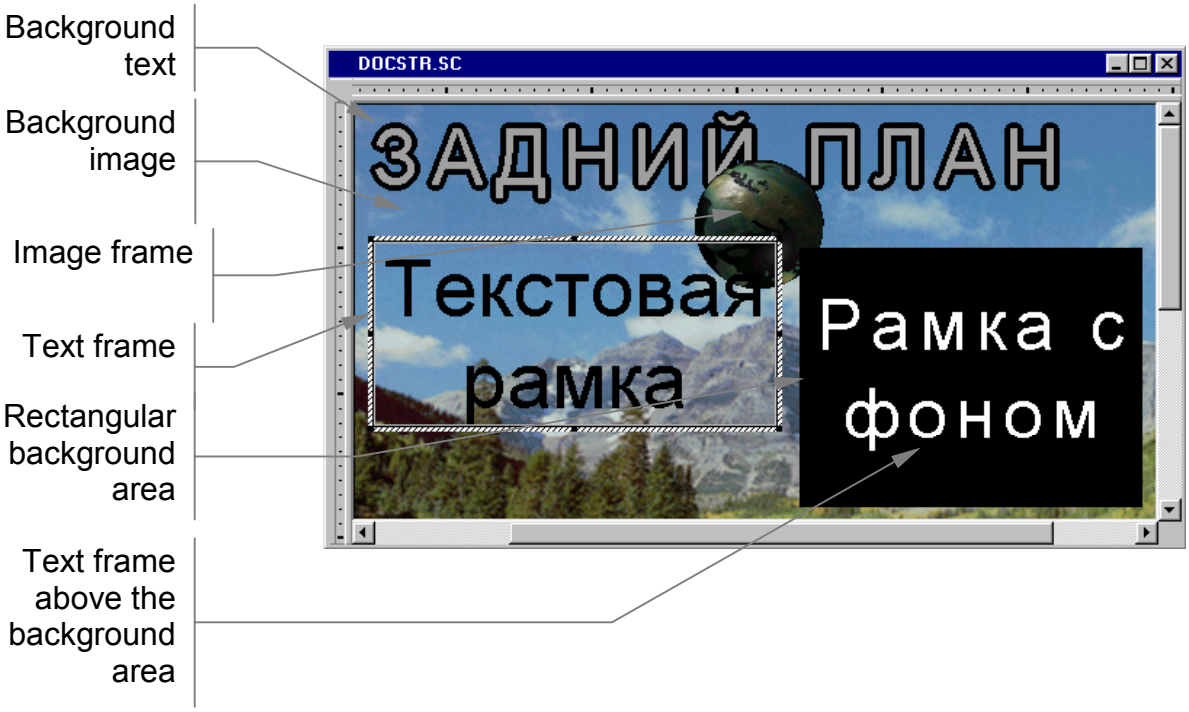
Setting output effects for each page

A script consists of one or several pages, which can have their own video effects (creeping line, vertical text scrolling, *CrossFade*, etc.). The number of pages in a script is unlimited, and it often turns out big enough, if complex effects are generated as combinations of simpler ones.

Maximum page size

Depending on the effect types, page dimensions can vary: for instance, if the creeping line effect is used, the maximum height of the page will not exceed the height of the video buffer, while its width can be unlimited. On the contrary, the use of the text vertical scrolling effect limits the maximum width by the size of the video buffer, while the height of page can be unlimited. Maximum page dimensions for other effects are the same, and restricted by the buffer size of the videocard used for titles output.

2.3. Page Multilayer Structure



Each new layer is a frame

Each frames and background text can be regarded as drawn at a separate transparent page (layer). Some of the pixels of this page become opaque, some – partially transparent, and some of them remain completely transparent. Then these sheets are superimposed and, if necessary, moved with respect to each other. As a result, taking into account the overall transparency of each point, the passing video signal can be seen behind this page.

The number of frames is unlimited, and the frame sequential number defines the order of its drawing. For example, after the background layer, first drawn is frame 1, then frame 2, etc. Finally, the frame with the largest number is drawn on top of all others. The order of frames can be changed arbitrarily in the course of editing.

Frames, along with text and background graphics can be used on pages with any effect, including the creeping line or vertical scrolling.

Background layer graphics and text

The *Background* is used to input the text information (with possible insertion of images (logos)). This text (and graphics) layer can be regarded as the farthest from the observer, everything else being allocated above it. The background layer is the most convenient for fast, real-time broadcasting of the text with the help of creeping line or vertical scrolling effects.

The background layer text is entered in a frame rigidly linked to the script page, where it can be moved only by changing the area of the effect use, or the paragraph horizontal indentation.

A graphics image from a *BMP* or *TGA* file with the alpha-channel can be put under this text. The name of the file is specified by the ***Page/Background ...*** command. The image is "rigidly" linked to the left upper corner of page. This method is convenient to apply when a specific unchangeable static picture from a file is to be used, rather than a series of different frames. Such file can be created in the program with the ***Page/Save As Bitmap ...*** command. The ***Page/Clear Background*** command allows to cancel the background graphics assignment.

2.4. Frame Types

The following types of frames can be used: *Frame* (text frame), *Box* (rectangular background area), *Picture* (image frame), and *Animation* (animation frame).

Text frame (*Frame*)

The text frame is used to create a text layer freely movable within the page (unlike the background layer). An image, background area, or other text can be put under the text of such frame.

Rectangular background area (*Box*)

A *Box* background area frame allows to create backdrop pieces for various graphics and text units of the page. Such frame can be used for text underlining, or creating rectangular edging with continuous, gradient, or textural filling, both opaque and semitransparent.

Image frame (*Picture*)

The *Picture* frame type is used for creating layers with images taken from graphics files. This type of frames is the most universal of all, since using graphics files, with or without alpha-channel, created by various graphics editors, it is possible to compose any collage and to place a text information in it.

Animation frame (*Animation*)

An *Animation* frame can be used for creating a layer containing an animated image from an *AVI* file or *TGA* file sequence. Frames of this type are used in pages with full-screen effects, such as creeping line, or scrolling. These frames allow to create animated icons. Such combined employment of several effects is most frequently used for displaying moving icons of weather, currency exchange rates, etc.

2.5. Text Information Structure

Paragraphs and their parameters

Information input in text frames and background areas is carried out in paragraphs. Each paragraph has a set of independent attributes: left and right indentation values with respect to the frame or page borders, text alignment methods, and the value of the line spacing adjustment with respect to that of the ***Leading*** paragraph.

Paragraph words splitting into lines and relative positioning

The word wrapping inside a paragraph during the text input and editing is carried out automatically. Words layout inside a line depends on the paragraph alignment method (left, right, centered or justified). Intervals between words can be adjusted through insertion of space and tab symbols. More precise allocation of words in lines and adjustment of line spacing can be achieved by dividing a paragraph into several.

Chapter 3 Work Starting and General Description of Program Interface

3.1. Work Starting

Software installation

The *Alpha Pro* software is installed by running the *SETUP.EXE* program under *Windows.* The software can be installed either with or without a video card present in the computer. This also applies to the operation of the *Alpha Pro* software itself; all you need is to install the hardware dongle to the *LPT1* parallel port of your computer.

Software Run

To launch the software, use the standard *Windows* routine. The program shortcut can be placed to the *Start-Up* folder for convenience.

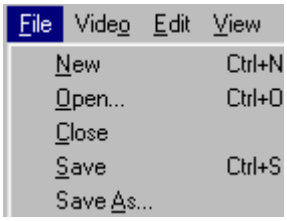
After the system is loaded, the passing-through video signal appears at the TV monitor connected to the video card output. Use the hardware setup window (***Video/Hardware Option***) to adjust the video card settings (the video input used, video signal format, etc.). To check the video card status, use the ***Grid*** button in the same window. If the video card works properly, a grid will appear at the video output upon pressing this button.

Program default state

If the command line does not provide any other options, the program will by default open a one-page script named *Script1*. Parameters of this page can be specified by pressing the ***Set As Default*** button in the ***Page Attributes*** bar. This page contains one empty paragraph, also with default parameters, specified in the ***Paragraph Attributes*** dialog box.

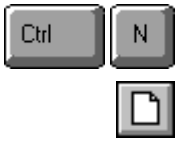
The program initial state also depends on the options set at the latest closure of the program. All options specified in ***Video/Preferences***, ***View/Draft***, ***Zoom***, ***Gridlines***, and ***Background*** setting windows are preserved between work sessions.

3.2. Script Creation, Storing and Loading



All operations with scripts, allowing to create a new script, load or save an existing one, are performed with the ***File*** menu, using the keyboard input or the ***Toolbar*** buttons.

Create a new script



Use the ***File/New*** command to create a new script. Before the script is first saved into a disk file, it has a name of the format: *Script1, Script2,* etc. It acquires the settings of the *Normal* style pre-defined in the program, and the page parameters are used as determined by the ***Set As Default*** button in the ***Page Attributes*** dialog box. This also applies to the single paragraph in this page: it is created with the default parameters defined by the ***Set As Default*** button in the ***Paragraph/Paragraph Attributes*** dialog box.

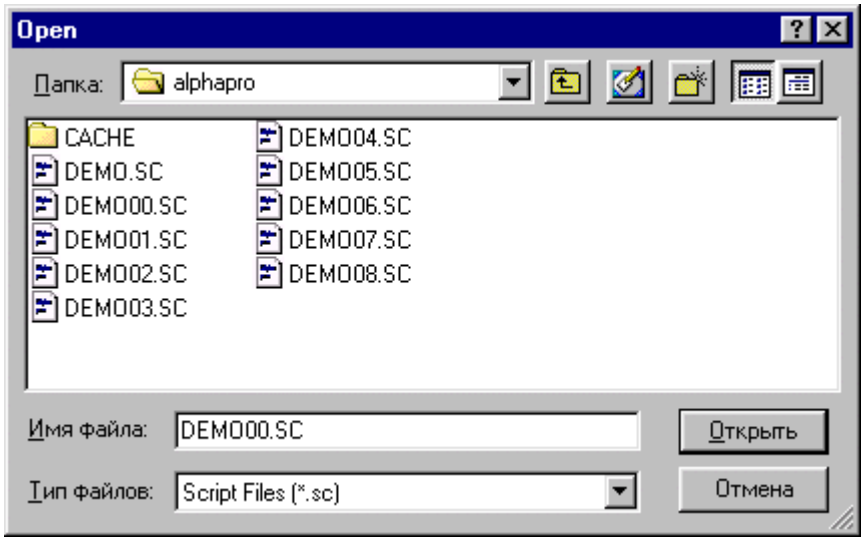
Save a script

Use the ***File/Save*** command to save the edited script in the file, from which it was loaded.

Save a copy of the script under a different name

Use the ***File/Save As*** command to save the edited script in text format under a new file name.

Load a previously created script



The ***File/Open*** command opens a dialog window allowing to input the file name of a previously saved script. To select the type of the file to be loaded, use the ***File Type*** field.

Load a *Windows*-encoding text file

Selection of the *All Files* value in this field allows to load a *Windows CP-1251* encoding text file. Characters of such text are ascribed the attributes of the program pre-defined *Normal* style. The text is then allocated to default-settings pages as determined by the ***Set As Default*** button in the ***Page Attributes*** dialog box. The End-of-Line (*Enter*) symbol is considered as a paragraph break. Paragraphs are also assigned default settings defined by the ***Set As Default*** button in the ***Paragraph/Paragraph Attributes*** dialog box. If the text contains an End-of-Page (*Form Feed*) symbol, a new page is created.

A *Windows*-encoding text file can also be inserted in the script by a *Clipboard* copy-and-paste operation (see **4.2 *Clipboard* Operations (Copying and Pasting)**). Such file can be opened in any *Windows* text editor (from *Notepad* to *Word*) providing the *Copy to Clipboard* (*Edit/Copy*) feature.

Close the script and its windows

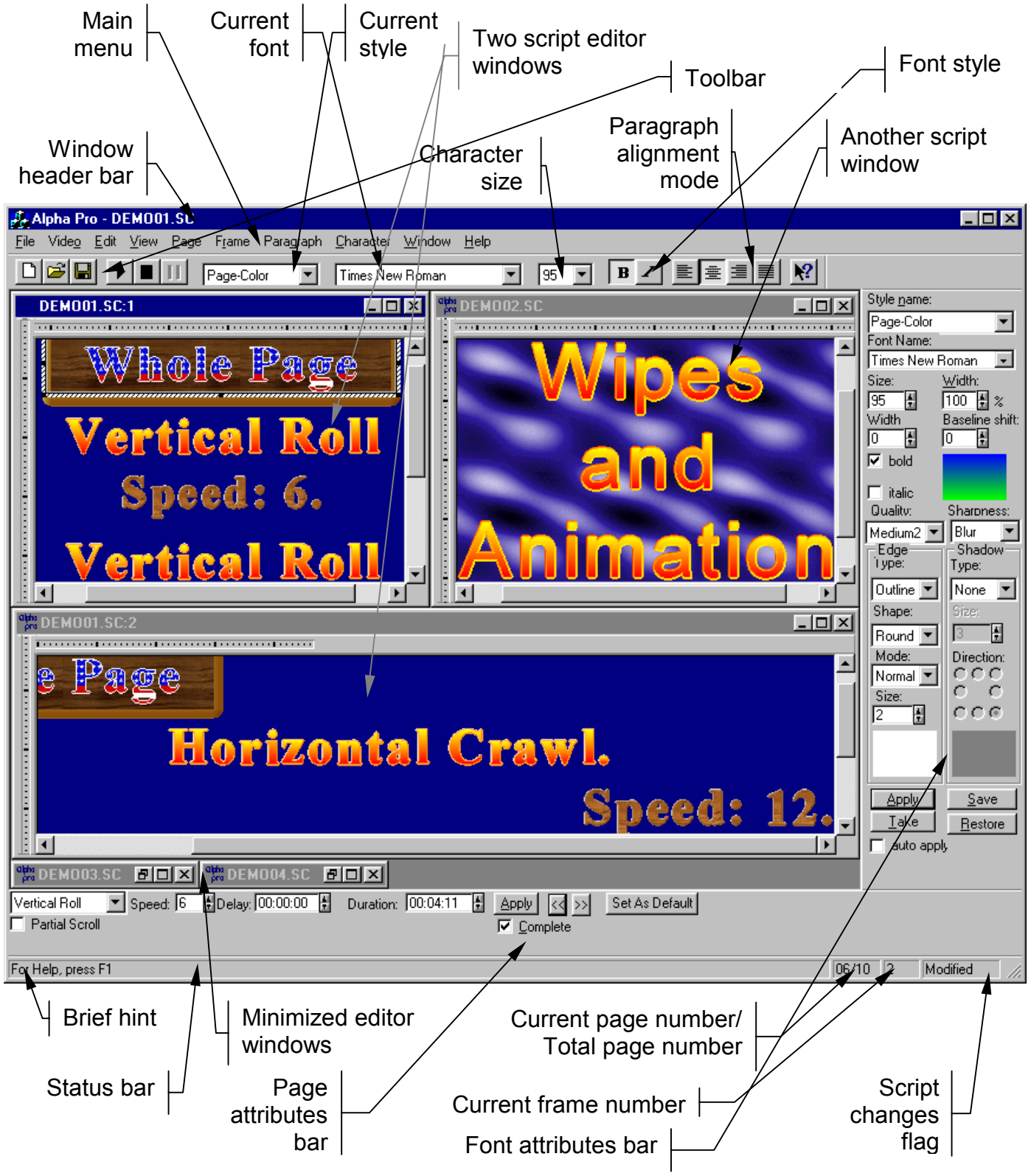
The ***File/Close*** command closes the current script, i.e. all windows used for its editing. However, if the script replay via the videocard is in progress at this moment, it does not stop. To stop the script replay, use the ***Video/Stop*** command.

If the script was changed after the latest disk saving, user's confirmation is asked for before the script is closed.

Script transfer issues

A script transfer to another computer, or change of folder names, does not cause any problem due to the fact that all references to graphic files stored in a script refer to their relative locations on the disk.

3.3. Main Program Window



Window header bar

The program header bar contains, after the text "*Alpha Pro*", the name of the file opened in the current window. If no file has been created, the script name of the type "*Script1*", "*Script2*", etc., is displayed.

Main menu

The main menu bar contains the following items:

File	file operations;
Video	titles editing and videocard output;
Edit	*Clipboard* operations;
View	information representation in the current window;
Page	script pages attributes and operations;
Frame	frame creation, attributes and order;
Paragraph	paragraph attributes;
Character	character attributes, style creation and logo insertion;
Window	program windows management;
Help	supplemental information from the Help file.

Program window workspace

The program window workspace is used to accommodate the titles output scripts to be edited. A script window can be minimized or restored. A window can be activated by a mouse left button click or using the *Ctrl+F6* button combination, which allows to look through all available windows consequently. To activate (and restore if minimized) a window, the ***Window/1,2,3,4...*** command, or a mouse left button double click can also be used.

Each script can be edited in several windows simultaneously; each of such windows can display a different part of the script. All changes made to the script are correctly and simultaneously displayed in all windows at once. Each of such windows can be assigned visualization attributes (***View/...***) independently.

Status bar

A status bar containing supplemental information is located at the bottom of the program window. It comprises:

- A brief one-line description of the used command;
- Number of the current page and the total number of pages in the script;
- Sequence number of the current frame (staring from 1);
- A flag indicating whether the script has been modified after its loading or latest disk saving.

To turn the status bar display on, activate the ***View/Status Bar*** option.

Toolbar

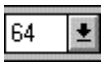
View/Toolbar.

To toggle the display of the toolbar usually located at the top of the program window, use the ***View/Toolbar*** option.

The toolbar comprises several drop-down list fields allowing to set the following attributes for the current text position or a highlighted text fragment:

Attributes/Style Name;
Character style (***Font Attributes/Style Name***);

Character font (***Font Attributes/Font Name***);



Character size (***Font Attributes/Size***).

The ***Toolbar*** buttons also allow to perform the following operations.

Create, load, or save scripts:


to open a new script to be edited in a new window – ***File/New*** command;


to load an existing script from a file into a new window – ***File/Open*** command;


to save a script in the current window to disk under the current name –***File/Save*** command;

Replay scripts:


to start the titles output according to the current window script – ***Video/Start*** command;


to stop the titles output – ***Video/Stop*** command;


to pause the titles output – ***Video/Pause*** command; after the button is pressed, the titles output stops, and the button "sticks"; when the button is pressed again, it "unsticks", and the titles output is resumed.

Font style buttons:


to set/discard the boldface style for the current text position, or all characters in a highlighted fragment – ***Character/Bold*** option;


to set/discard the italic style for the current text position, or all characters in a highlighted fragment – ***Character/Italic*** option;

Paragraph alignment buttons:


to set the left-edge alignment mode for the current paragraph or all paragraphs of a highlighted fragment – ***Paragraph/Left*** option;

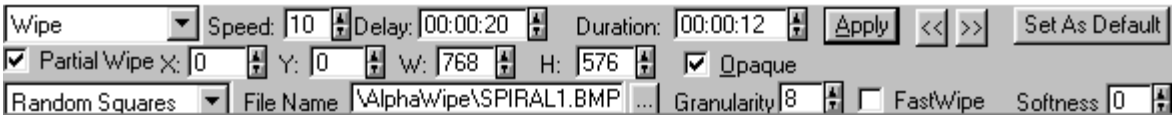

to set the center alignment mode for the current paragraph or all paragraphs of a highlighted fragment – ***Paragraph/Center*** option;


to set the right-edge alignment mode for the current paragraph or all paragraphs of a highlighted fragment – ***Paragraph/Right*** option;


to set the full alignment mode for the current paragraph or all paragraphs of a highlighted fragment – ***Paragraph/Justify*** option;

Hint button:


to call the description of specific commands or program element interfaces from the Help file. A special pointer with a question mark appears on the screen, to produce information found in the Help file on a clicked-upon item (not yet implemented).



Page attributes bar

The page attributes bar contains a drop-down list allowing to select a video effect for the active script fragment page(s). According to the selected effect, the choice of the fields present in this bar also varies.

See **Chapter 6 – Script Pages Editing** for more information on effects selection and settings.

To toggle the page attributes bar display, use the ***View/Page Attributes...*** option, or press *F7*.

Font attributes bar

The ***Font Attributes*** bar allows to customize current text style for input symbols or a highlighted text fragment. Besides, it can be used to edit and refresh the attributes of a specific style.

The window contains the following field groups:

Shape & Color	character internal area drawing style, color filling, and position;
Edge	character edge style, shape, size, and color filling;
Shadow	shadow, or 3D side surface style, direction, size, and color filling.

It also comprises additional fields setting the character drawing quality, edge smoothing degree, refreshing and restoring buttons for the selected style, as well as those for the result videocard display.

If no text fragment is selected, the bar reflects the style settings for the current (previous to the cursor) text character. If, however, some changes are made and accepted (by pressing the ***Apply*** button), the new settings are applied to new characters input in this position. If a fragment of the script text is highlighted, the bar displays the settings common for all the text in the fragment. If some of the attribute have different values within the fragment, corresponding fields of the bar display a special **indefinite** value.

See **4.6 Text Character Attributes** and below for more information on text character attributes.

To toggle the display of the font attributes bar, use the ***View/Font Attributes...*** option, or press *Shift+F7*.

Show/hide tool and status bars in program window

Large as the program window may be, if several editor windows are opened, the workspace area often proves insufficient. Temporary removal of ***Toolbar***, ***Page Attributes*** bar***, Font Attributes*** bar, or ***Status Bar*** not currently in use may help to enlarge the workplace area. To toggle the bars display, use the ***View*** menu items.

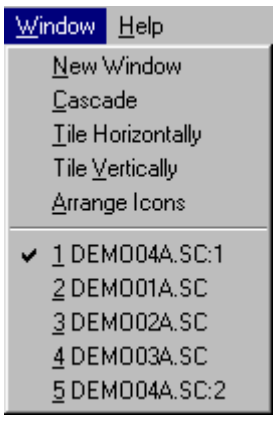
The ***Toolbar, Page Attributes*** bar, and ***Font Attributes*** bar can also be separated from the main program window and relocated at the display screen at will. To do so, double-click the mouse left button on the corresponding bar. The ***Toolbar*** can also be positioned horizontally or vertically. A mouse double-click on a separated bar returns it to its regular position.

Create new editor window(s)

A new editor window is created in the program window automatically when a script file is opened (with the ***File/Open*** command) or a new script is created (with the ***File/New*** command). The display settings for the window are used as specified in the ***View*** menu.

Create new window(s) for opened scripts

It is often convenient to get a parallel access to different parts, or different views, of the same script. The program provides for this need allowing to display a script simultaneously and correctly in multiple windows.



New windows (using the current script display settings) can be opened with the ***New Window*** command in the ***Window*** menu. If several editor windows are opened for one script, each window header displays after the file name, the window number– "*:1*" for the 1st window, "*:2*" for the 2nd one, etc.

Hint. Script editing in multiple windows allows to avoid unnecessary movements of the cursor between the pages; e.g., when a text fragment copying is concerned. Minimizing some of opened windows allows the program to re-draw the rest of them almost without slowing down. Re-opening the minimized windows can be used, e.g., to view the final version of the resulting page.

Activate/ deactivate editor window

There exist many ways to switch between editor windows. The most convenient method of window activating is to click the mouse left button when the pointer is over the window. If the window is minimized, double-click to restore and activate it.

All windows can be activated in consequence by pressing *Ctrl+F6*. Finally, the ***Window*** menu contains a list of all windows, which can also be used for window selection.

Minimize editor window

To minimize the temporarily unused editor windows, use the system menu and the standard *Windows* procedure.

Close editor window

To close the unused editor windows, use the window system menu, or press *Ctrl+F4*. Closing the last remaining editor window for a given script also closes the script file.

Types of windows arrangement

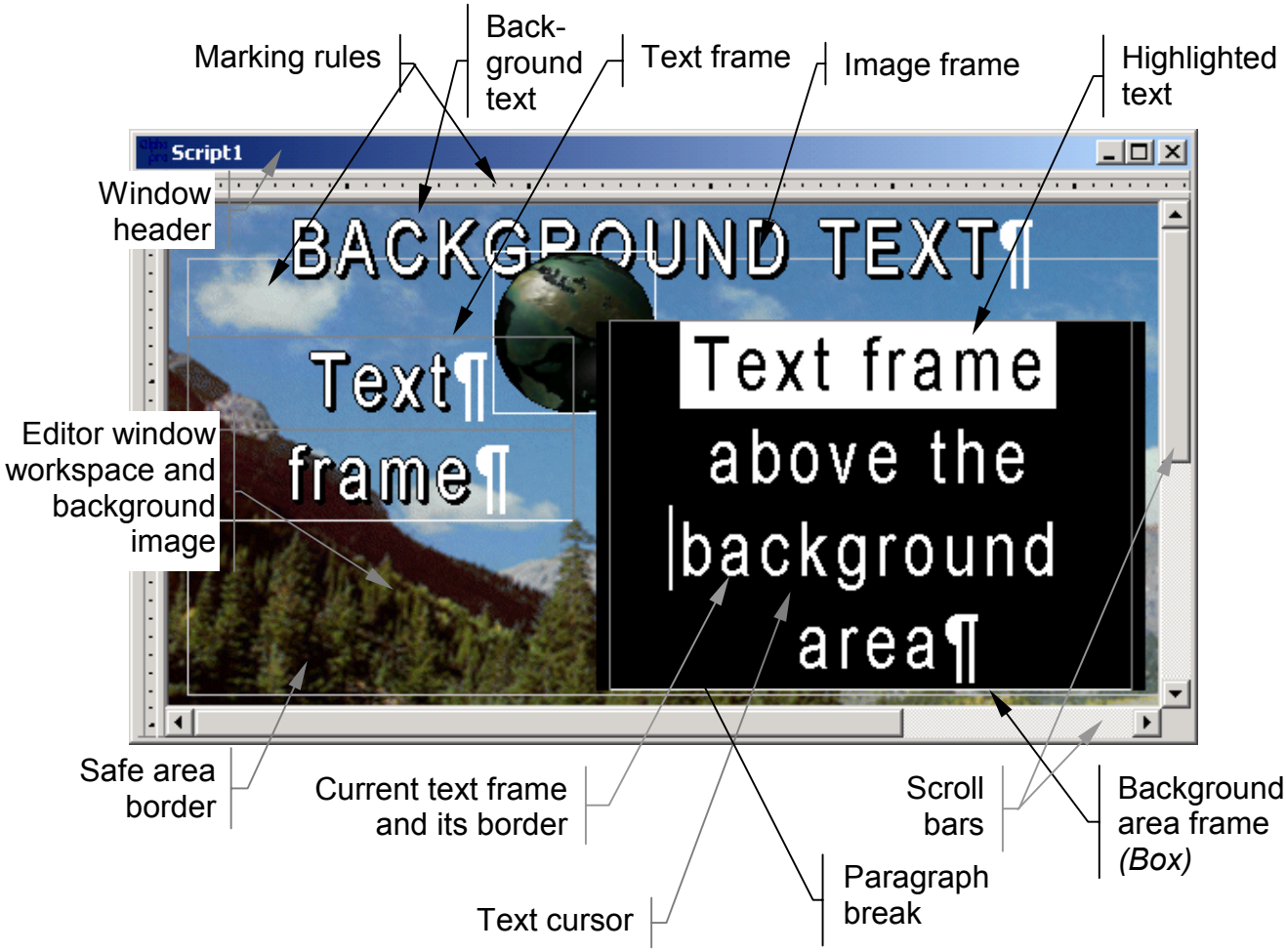
Multiple editor windows can be arranged in several different ways. The cascade arrangement (***Window/Cascade***) allows to switch between the windows easily using the mouse pointer. The tile arrangement (***Tile...***) allows to view the content of all opened windows simultaneously. The horizontal tile mode (***Window/Tile Horizontally***) is handier for pages with a creeping line effect, while vertical tile mode (***Window/Tile Vertically***) can be used for pages containing a vertically scrolled text. To accommodate whole script pages in the editor windows, re-scale their images using the ***View/Zoom*** option.

Minimized editor windows arrangement

Reducing of the program window size can put the minimized window icons beyond the workplace area, and out of user's sight. To get all

the icons back within the workplace area, use the ***Window/Arrange Icons*** command.

3.4. Script Editor Window



Editor window header bar

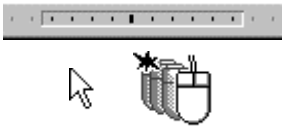
The window header bar displays the script file name or, if the file is not yet created, a name of the type "*Script1*", "*Script2*", etc. If multiple editor windows are opened for one script (with the ***Window/New Window*** command), the number of the window is specified after the file name – ":1" for the 1st window, ":2" for the 2nd one, etc. All this information constitutes the windows name.

Script scroll bars of editor window

If the text does not fully fit in the editor windows, standard scroll bar appear at the right-hand and bottom edges of the window, allowing to reach any part of the script.

Marking rules

Marking rules appearing at the top and left-hand edges of the window provide a convenient way of changing the application area of an effect if it is not supposed to be carried out all over the screen. In this case, the effect area boundaries also define the indentation values for background text (except the bottom margin).



Marking rules are provided with two pairs of sliders located at the edges of the “extruded” area of the bar and corresponding to the limits of the page effect area. When the mouse pointer is located over such slider, it turns into an arrow with a “support”. Moving the mouse with a pressed left button provides an easy way of changing the area borders. To specify the area more precisely, use the ***X, Y, W,*** and ***H*** fields of the ***Page Attributes*** bar.

Marking rules are provided with small ticks at every 10th pixel, and larger ticks at every 100th pixel.

Editor window workspace

All page graphic elements are located in the editor window workspace.

Color selection for "transparent" points

The workspace point color (the ***View/Background*** option) should be chosen so as to differ from the colors used in graphic elements. The selected color is further used for all workspace points transparent for the passing video signal, beyond the page elements.

Safe area

The safe area corresponds to the part of the TV screen visible in all types of monitors. Borders of this area are shown in the editor window as thin straight lines.

The area position and dimensions depend on the type and settings of the videocard used for titles output. These parameters are specified in the ***Safe Area*** attribute group of the ***Video/Preferences*** window. The upper left-hand corner point of the videocard buffer is ascribed the *(0,0)* coordinates.

Correct image re-drawing and recovery in editor window

Attention. In the course of editing the script current status in the window may be represented incorrectly. Although there is no dedicated command for the screen re-drawing, it can easily be achieved in a number of ways.

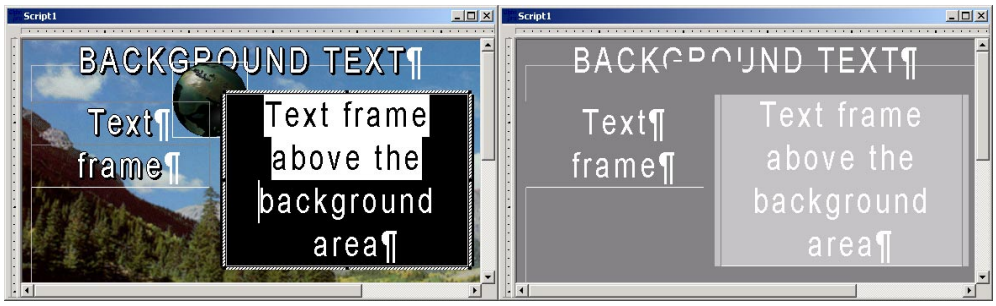
The simplest method is to minimize and restore the editor window. Alternatively, use the ***View/Draft*** option to change the script display mode in the window. Another way consists of switching first to another page of script, and then back to the initial page.

3.5. Types of Script Representation in Editor Window

An editor window always displays a single page of the script. The page is drawn exactly according to the sequence and allocation of frames, which allows to form a fairly precise impression of the page appearance. Frame position and dimensions can be changed at will; borders of the active frame are marked with eight rectangles, which can be moved with the mouse to change the frame height and width. Placing the mouse pointer at the frame border, but beyond the rectangles, displays a cross-arrow signifying that the frame can be moved across the page.

The so-called *Safe Area* is also displayed in the editor window to indicate the part of the page, which is certain to appear on a TV screen of any type.

Draft and Full



Usually, the first stages of text input do not require displaying all graphic elements in full detail. A simplified representation of these elements allowing to accelerate the process is available via the ***View/Draft*** option.

If this option is not selected, all script elements are fully displayed. In this mode, graphic elements are reproduced in the same way as at an actual titles output, except the *Anti-aliasing* feature and the display of frames and characters superposition taking into account the alpha channel signal. Alpha channel signal is taken into account by mixing the color of a semi-transparent point with black, in an appropriate proportion.

Zoom or actual size

If a script does not fit into an editor window, use the ***View/Zoom*** mode allowing to reduce the size of all output elements 2 or 4 times.

With/without grid lines and special characters

Page and frame borders, paragraph limits and indentations can be displayed in all modes. To make these borders, as well as the paragraph break (¶) symbols, visible, use the ***View/Gridlines*** option.

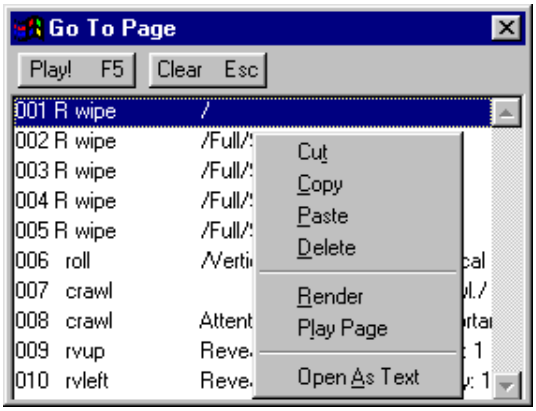
Multilayer page output in editor window

If an effect specified for a page is carried out in the environment produced in the video buffer by previous script events, the editor window displays residual elements of previous pages in addition to those of the current page. These elements are displayed without any frames, grid lines or special characters, and act as current page background items.

This background is output out of reach of the page effect area specified in the ***X, Y, W,*** and ***H*** fields of the ***Page Attributes*** bar. The ***Page Attributes/Opaque*** transparency parameter determines whether this background can be seen within the effect area.

3.6. *GoTo Page* Window

The ***GoTo Page*** window provides a full page list for the current script. To open this window, use the ***Page/Go To...*** command, or press *Ctrl+G*.



Each line of the list in the ***GoTo Page*** window corresponds to one page. Such a line contains the page number and all text information about the page; paragraphs are separated by slash (/) symbols. Besides, each line provides the name of the effect used in the page and a flag showing whether the page has been rendered (the *R* character).

The current page is always highlighted in blue in the ***GoTo Page*** window list.

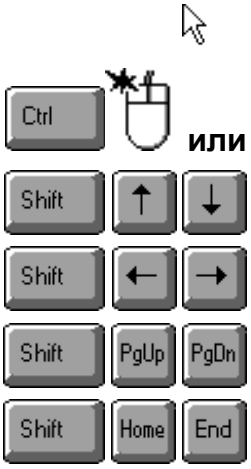
Page navigation

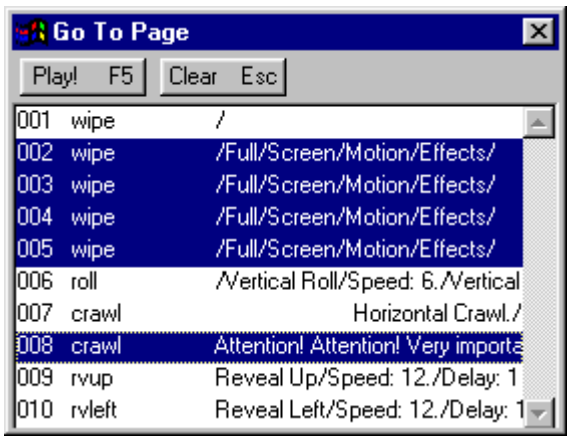
Initially, this window was meant to be used for easy navigation between the script pages (hence its name, ***GoTo Page***). To move to any page using the ***GoTo Page*** window, find the corresponding line in the list and click the mouse left button over it. Arrow buttons, page turning (pressing *Ctrl+PgUp* and *Ctrl+PgDn*), or direct moving to the list head or tail (pressing *Home* and *End*) can also be used for movements between the list lines.

Other page typesetting functions

Beside the navigation between the pages, this window can be used for a number of other operations with one or **several selected pages** (see below). These are *Clipboard* buffer operations (***Cut, Copy, Paste***), page removal (***Delete***), rendering (***Render***), and replay (***Play Page***). To open the submenu for these commands, click the mouse right button at any point of the ***GoTo Page*** window. The most frequently used commands (the selected script pages replay – ***Play!*** – and videocard buffer clearing – ***Clear***) are provided as dedicated buttons in the upper part of the window. See the documentation below for a more detailed description of these commands.

Selection of script pages





At least one page – the current script page – is always highlighted and selected in the ***GoTo Page*** window list.

To select a **set** of several pages in the ***GoTo Page*** window list, use the mouse left button, while holding the *Ctrl* button pressed. The selected pages can be located anywhere in the list, not necessarily in a continuous sequence. Selected pages are highlighted in dark-blue in the list.

A continuous sequence of pages can be conveniently selected using the arrow buttons, while holding the *Shift* button pressed. Each pressing of an arrow button adds one page to the head or the tail of the selection list, respectively. Pressing the *PgUp* and *PgDn* buttons, while holding the *Shift* button pressed, adds to the list several pages at once. Likewise, pressing the *Home* or *End* button adds all pages from the current one to the head or the tail of the list, respectively.

After a set of pages is selected, it can be altered using the mouse left button, while holding the *Ctrl* button pressed.

Another left mouse click over a selected page line with the pressed *Ctrl* button excludes this page from the selection.

3.7. Text Cursor and Current Position

The text cursor – a blinking vertical line – is used to indicate the current position (the point of character input) in the editor window. If the current position is beyond the text limits (is located at one of the frames), the cursor is not displayed, and the current frame is highlighted with a special rectangle.

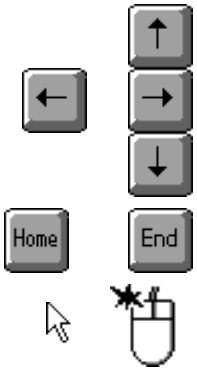
The cursor moves across the frames and text fragments of frames and pages according to their order; to transfer it to the next (or previous) frame or page, use the appropriate arrow button.

If the cursor points to a frame, it is outlined with a special rectangle with eight small squares allowing to adjust the frame dimensions and position.

Script movement within editor window

To move a script within the editor window **without changing the current position**, apply the mouse pointer to the horizontal and vertical scroll bars. Vertical movement can also be achieved by pressing *PgUp* and *PgDn*. Such vertical movement is limited within the page borders.

Current position movement within script page



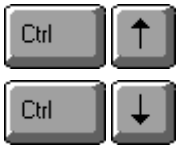
Use arrow *Home,* and *End* buttons to move the current position within the script. The latter two buttons can only be used within a text paragraph.

Horizontal arrow buttons are used to move the current position to the next or previous position. Vertical arrow buttons operate the same way if the current position points to a frame, and if the cursor is within a text paragraph, it is moved one line up or down. The *Home* and *End* buttons move the current position to the beginning or the end of the text paragraph line, respectively.

It is often more convenient to change the current position by setting the mouse pointer to the selected point and pressing the mouse left button. If the pointer is over a position in the text, the cursor moves to this position. If the pointer is over a frame, this frame is then selected as the current one.

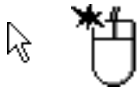
Attention. It should be noted that if the pointer is over an intersection point of several frames, the frame located above all others (closest to the viewer) is selected.

Current position movement between editor window pages



Arrow buttons can be used to move the current position to the previous or next script page, if the cursor is in the first or last position on the page, respectively. It is, however, handier to use the *Ctrl*+↑ and *Ctrl*+↓ button combinations, which move the current position to the beginning of the previous or next page.

Current position movement within *GoTo Page* window

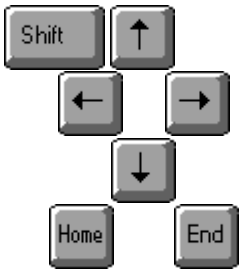


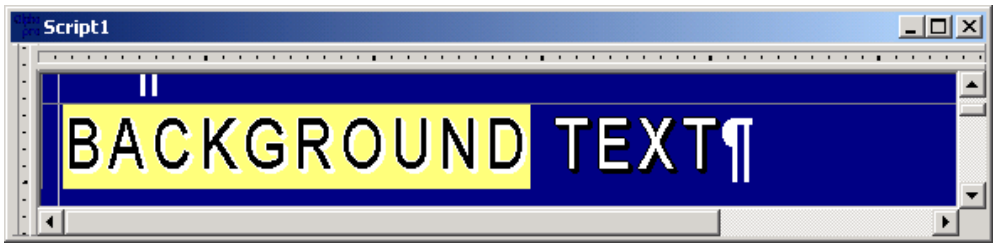
See **3.6 *GoTo Page* Window** above for more information on navigation between the script pages.

Current position movement with *Page Attributes* bar

The << and >> buttons of the ***Page Attributes*** bar can similarly be used to move the current position to the previous or next page.

3.8. Highlighting a Fragment of the Page Content





Moving the cursor within a script with a pressed *Shift* button allows to highlight a fragment of the script text, frame or page. All above rules for simple cursor movements using the buttons also apply to this operation.

Moving the mouse pointer with a pressed mouse left button also provides an easy and quick way to highlight a text fragment. However, this method can only be used to highlight the whole or a part of the text within a selected background area or text frame of one page.

After the text fragment is highlighted using the mouse, it can be edited using the keyboard.

One-word fragment

There exists a way for quick highlighting a one-word fragment. To do so, put the mouse pointer over the word and double-click the mouse left button.

Frame and page highlighting

To include a frame into a fragment, activate it and press *Shift*+→ or *Shift*+↓. Further moving the current position, while holding the *Shift* button pressed, allows to add other frames or whole pages to the fragment.

Such highlighting can be used for copy, past and delete operations offered in the ***Edit*** submenu. If a fragment spreads across a border between two pages, both of them are considered selected.

Highlighting all page content

To highlight all the text and all elements of a page, use the ***Edit/Select All*** command, or press *Ctrl*+*A*.

Display and editing of general attributes of text fragment elements

Attribute values for the current character and paragraph are displayed and edited using the respective dialog window fields. If a fragment of the script text is selected, the values represented in the corresponding attribute fields refer to the settings common for all elements of the fragment. This principle applies to script characters and paragraphs, but not to pages and frames.

If an attribute acquires several different values within the fragment, a special indefinite value is displayed in the corresponding field. In number fields and lists, it is represented by a blank value; in color filling fields, by an uncolored border; in option checkboxes, by grey squares; and in radio-button groups, by the state, in which none of the buttons is selected.

If such indefinite value is not changed in the course of the fragment editing, the corresponding attribute is not altered for the entire fragment. Selection of a new, definite value changes the corresponding attribute for all fragment elements.

If a fragment exceeds the paragraph borders

Attention. It should be noted that if a fragment limits cross (include) a paragraph break (the ¶ symbol), the border-crossing rule is applied: attributes for both adjacent paragraphs are displayed, edited and assigned simultaneously.

The border-crossing rule is applied in a somewhat different way when a text fragment is copied to the *Clipboard*. See **4.2 *Clipboard* Operation (Copying and Pasting)** for more information.

3.9. Parameters Setting and Editing

Indefinite parameters values

If and attribute acquires several different values within the fragment, a special indefinite value is displayed in the corresponding field. In number fields and lists, it is represented by a blank value; in color filling fields, by an uncolored border; in option checkboxes, by grey squares; and in radio-button groups, by the state, in which none of the buttons can be edited.

If such indefinite value is not changed in the course of the fragment editing, the corresponding attribute is not altered for the entire fragment. Selection of a new, definite value changes the corresponding attribute for all fragment elements.

Integer numbers input

Beside the option of the standard keyboard input of integer numbers, many dialog windows provide:

1) two software buttons with vertical arrows right of the input field, used to increase/decrease the field value by 1. Similar changes of the value can be achieved by pressing “*Gray* +” and “*Gray* –“, respectively. Standard “+” and “–“ buttons can be used to input the sign of numerical values.

2) a horizontal slide bar. To use it, double-click the mouse left button at the field name. A dialog box is displayed containing the parameter name, a numeric input field and a horizontal scale bar with a slider. The slider can be moved with the mouse, allowing a rapid change of the field parameter value within the entire allowed

range, while the corresponding value is displayed in the input field. The minimum possible value is indicated left to the field, the maximum value, right to it.

The resulting value is considered entered after the *Enter* button is pressed, or after a mouse click beyond the window. Press *Esc* to discard the input of the new parameter value.

Chapter 4 Editing of a Script Text

4.1. Input and Deleting

Text Information Input

Current attributes for inserted characters

Text entered from the keyboard is assigned the current attributes of the character at the previous cursor position. If the text is inserted at the very beginning of a paragraph, then at first, attributes of the next character are taken. When a new paragraph is created, certain attributes are assigned to the separator character **"¶"**, and used further when real input begins (see more on attributes assignment to empty paragraphs below, **Beginning a new paragraph**).

The simplest way to enter text characters is to use the keyboard. The program always inserts typed characters at the cursor position, moving it to the right together with all remaining text of the paragraph. At this movement, the text is transferred from line to line by words, while the height of the paragraph increases, moving the subsequent paragraphs downwards.

Using the *Windows Clipboard*

It is convenient to enter the text using the *Windows Clipboard.* To do this, open any text editor, for example *Notepad* or *Word*, to type the text there and then to copy it in *Clipboard*. In the program, paste it into the required position of a text frame or a background text. Inserted characters are assigned the current attributes, according to the text cursor position. Text division into paragraphs is not preserved.

From a *Windows*-encoded text file

It is also possible to load text files in *Windows CP-1251* encoding. Text characters of this kind are assigned default style attributes of the program (*Normal*). The text is placed onto pages with default attributes, as defined by pressing the ***Set As Default*** button of the ***Page Attributes*** toolbar. The end-of-line character *(Enter)* corresponds to paragraph breaks, also assigned the default attributes as set in ***Paragraph/Paragraph Attributes*** dialog box with ***Set As Default*** button. An occurrence of the end-of-page character (*Form Feed*) in the text creates a new page.

To load a *Windows*-encoded file, select the *All Files* option in ***File/Open/File Type*** field of the ***File/Open*** dialog box. It is recommended to save a script created with this method in a separate *.SC*-extension file.

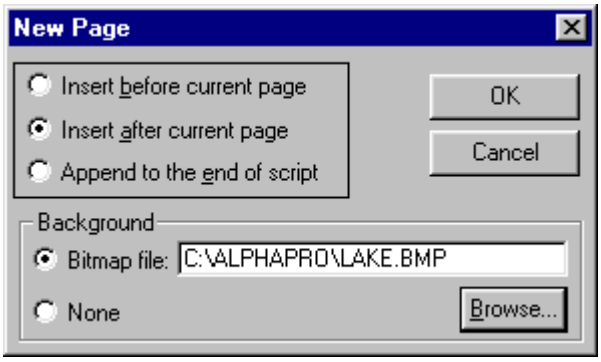
Beginning of Input

New script

To create a new script, use the ***File/New*** command, press *Ctrl+N,* or the respective ***ToolBar*** button. The created script initially receives a name of the format *ScriptN* and consists of a single page with an empty paragraph, with default parameters. The initially set current style for characters is the *Normal* style predetermined in the program.

New page

There are several possible ways to begin the input and design of a new script page. The first way is to call the ***New Page*** dialog box using the ***Page/New...*** command and to create one more page with default settings at the end of the script, after (or before) the current page. The current cursor position then moves to the beginning of the only empty paragraph of this page. This paragraph is assigned the same set of parameters as was used at the moment of the command execution. It is also true for the current attributes of characters.



The same dialog box can be used to specify a *BMP* or *TGA* format file for the page graphic background, similarly to procedure used in the ***Page/Background…*** command.

If the cursor is **inside a background text**, a new page can also be created by pressing *Ctrl+Enter* or using the ***Page/Insert Page Break*** command. In this case, the background text after (and including) the cursor is transferred on to the following, newly created page, while all background text preceding the cursor and all other frames will remain on the previous one. The parameters of pages and paragraphs broken this way are identical. The current style of characters is also preserved.

New frame

To create a new text frame, image frame, animation, or background area, use the ***New Frame***, ***New Picture***, ***New Animation***, or ***New Box*** commands in the ***Frame*** menu, respectively.

A text frame is created with one paragraph; its parameters and character style are determined by the paragraph parameters and characters style in use at the moment of frame creation.

New paragraph and new line

To begin the input of a new text paragraph, press *Enter*. The new paragraph acquires the parameters of the previous one, and the current character style is also preserved.

New lines in paragraphs are created automatically, as the new text is entered. Line offsetting is done by words. If it is necessary to fix words in specific lines of a paragraph, split the paragraph into several in proper places.

Remove Text

Text character or paragraph separator

To remove one text character, press *Del* with the cursor in the preceding, or *Backspace* in the next position.

Deletion of the separator character **"¶"** results in merging two paragraphs into one with the common text acquiring the attributes of the second paragraph.

Frame

To remove an entire frame, use the ***Frame/Delete...*** command or pressing *Del* with the current position pointing at the frame. A text frame can only be deleted by pressing *Del* if it is empty (does not contain any text). If a text frame is not empty, the characters inside it will be deleted first.

Page

Removal of an entire page is done by the ***Page/Delete*** command.

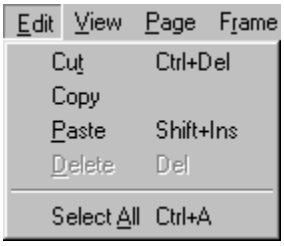
Script fragment

If the *Del* key is hit when a fragment is selected, all characters and all frames of this fragment (except for text frames) are deleted from the script. Text frames and pages completely belonging to a fragment remain, but become empty. If these frames are to be removed as well, it can be done as described above.

A fragment of the script can also be deleted by the ***Edit/Delete*** command. The ***Edit/Cut*** *command* (or pressing *Ctrl+Del*) deletes a fragment and copies it to the *Clipboard* (see below).

4.2. *Clipboard* Operations (copy and paste)

Copying into *Clipboard*



or 

Script editing often requires the use of recurring fragments or even whole pages. The *Clipboard* buffer of the *Windows* environment provides a convenient way for their transfer from one location to another. Commands of exchange with this buffer are invoked via the ***Edit*** menu or by pressing the mouse right button. Names of commands in the submenu or context menu depend on the current position in the editor window and the frame selection mode (see below).

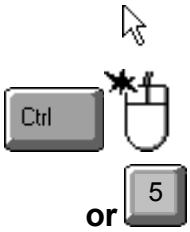
Script fragment

The most common way of copying is to select a fragment of the script and execute the ***Edit/Copy*** command (or by pressing *Ctrl+Ins*). In this case, the fragment actually copied may exceed what is selected in the program window. For example, if the fragment includes a frame border, the entire frame (with all the text within) is copied to the *Clipboard*. Similarly, if the fragment crosses a page border, all of the page is copied together with the next one. If the fragment begins in an area of the background text of the page and captures at least one frame (no matter which), all of the background text is placed in a text frame with coordinates (0,0) and width equal to 800 and is copied to *Clipboard* entirely, together with all other selected frames.

Entire page

To copy an entire page to the *Clipboard*, use the ***Page/Copy to Clipboard*** command. When entire pages are copied, they are captured together with their effects. When such *Clipboard* content is then pasted into a script, new pages are created. However, sometimes it is required to copy the content of an entire page and to transfer it to another one, leaving all effect parameters in the latter page intact, that is, without adding of a new page. In this case the content of the page should be selected with the ***Edit/Select All*** command (or by pressing *Ctrl+A*), and then copied with the standard ***Edit/Copy*** command. Note that in this case, all background text of the copied page is moved into a special text frame with coordinates (0,0) and width 800.

Selection and copying of frame groups



A group of one or more frames, not necessarily sequential, can be selected to copied entirely to the *Clipboard* in a standard way with the ***Edit/Copy Frames*** command or by pressing *Ctrl+Ins*. In particular, it is the only way to copy exactly one frame.

To select such a group, mark the respective frames by clicking the mouse left button on them, while holding the *Ctrl* key pressed. The single current frame can be selected with pressing *Gray 5*.

To discard the frame selection, press *Gray 5*, or, similarly to the selection routine, clicking the left mouse button on a frame to be de-selected, while holding the *Ctrl* key pressed.

Inserting of a Script Fragment from the *Clipboard*

To insert of any script fragment, page or group of frames previously copied to the *Clipboard*, use the ***Edit/Paste*** command (or press *Shift+Ins*). This command results in pasting of the fragment from the *Clipboard* into the current position. If this fragment only contains text, it can be inserted only if the current position specifies a character, rather than an entire frame. If the *Clipboard* contains frames they are pasted right after the last frame of the current page (the background text is pasted into the video in a special text frame), and if the *Clipboard* contains entire pages, they are pasted into the script before the current page, increasing the total pages number.

4.3. Splitting and Merging of Pages and Paragraphs

Splitting and merging of page text

To split a page in two, select the appropriate position in the background text and use the ***Page/Insert Page Break*** command (or press *Ctrl+Enter*). In this case the text preceding the cursor and all frames will stay on the old page, while the text following the cursor will become the background text of the new page. Attributes of the created page will be identical with those of the previous one. If the current position specifies a page frame, the page splitting function is unavailable.

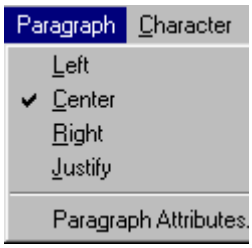
To merge two pages into one, use the *Clipboard* to copy and paste separately the background and frame text of one page to the other, then delete the superfluous page.

Splitting and merging of paragraphs

To split a paragraph in two, press *Enter* when the text cursor is in the position where a paragraph break is needed. Parameters of the two paragraphs will be identical to those of the old one. The current character style will also be preserved.

Deleting a "¶" separator character results in merging two paragraphs into one, with the text attributes of second of them becoming those of the resulting paragraph.

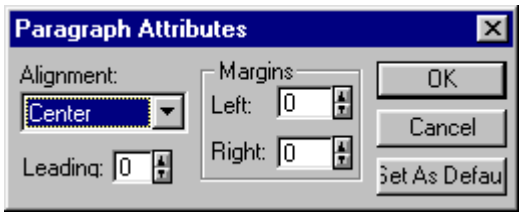
4.4. Setting of Paragraph Attributes



or 

If at least a part of the current paragraph(s) is contained in the script fragment, the paragraph attributes are defined by the items of the ***Paragraph*** menu. Alternatively, to assign these parameters, click the mouse right button when the current position (cursor) is located within the paragraph text. Some of the parameters can also be adjusted using the ***Toolbar*** buttons.

The *Paragraph Attributes* dialog box



The ***Paragraph/Paragraph Attributes...*** menu item allows to set the values of alignment parameters for the current paragraph, or paragraphs of the selected script text fragment through the ***Paragraph Attributes*** dialog box.

Types of paragraph text alignment

One of the following text alignment modes can be chosen for a paragraph:

 *Left* all text lines are aligned by the left edge;

 *Center* all text lines are aligned by the middle line between the edges;

 *Right* all text lines are aligned by the right edge;

 *Justify* all text lines of the paragraph, except the last one, fill in the space between the edges. The last line of the paragraph is aligned by the left edge.

Type of the paragraph alignment displaying and setting

Attributes of paragraph text alignment can be set through the ***Paragraph*** menu, ***Paragraph Attributes*** dialog box, or the ***Toolbar***. If paragraphs in a fragment do not have the same type of alignment, the corresponding field value is indefinite – that is, empty, when none of the items is checked in the ***Paragraph*** menu, and no button is pressed in the toolbar: all buttons are light-gray.

Adjustment of paragraph horizontal indention: *Left* and *Right*

Paragraph indention values are counted in pixels from respective frame or page borders, if the paragraph is at the background. To change the indention explicitly, use the ***Paragraph Attributes*** dialog box. Specifying the indention values in the ***Left*** and ***Right*** fields, one should be careful to avoid the negative paragraph width.

A 0 value in one of these fields corresponds to no additional indention.

Adjustment of paragraph line spacing: *Leading*

The ***Leading*** field in the ***Page Attributes*** dialog box allows to adjust the paragraph vertical line spacing in pixels. This value is added to the line height calculated automatically for all characters of the paragraph, and can be positive or negative. A positive value increases the spacing between paragraph lines. A negative value, on the contrary, reduces the spacing, but its absolute value must not exceed the initial (default) value of the line height. This field can have an indefinite (empty) value.

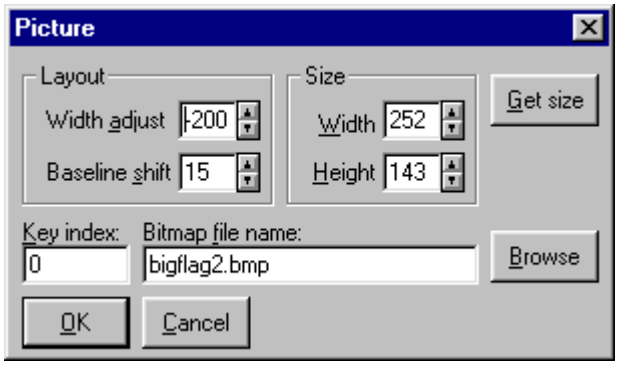
Setting and use of default paragraph attributes

The ***Set As Default*** button allows to save the current values of the ***Paragraph Attributes*** window fields as default paragraph attributes, which are later used for newly created scripts and text loaded from files.

4.5. Inserting of Logo Images into Text

The ***Character/Insert Picture*** command allows to insert a logo image directly as a text character, by entering the name of the corresponding graphics file. The logo image is then considered as a regular text character and can be moved within the paragraph together with other text. The only difference is that it is not assigned style attributes defined at the ***Font Attributes***.

The *Picture* dialog box



Insert Picture and ***Picture Attributes*** commands open the ***Picture*** dialog box for input of the name, layout parameters (***Layout***) and scaling (***Size***) of the graphic file used as a logo. The command is accessible if the text cursor is before the logo image.

Used types of graphics files and their transparency

The graphic file name is entered in the ***Bitmap file name*** field. ***Browse*** button is provided to simplify the search and selection of the logo file.

The following types of graphic files are supported:

- *Windows BMP 8, 24 bit* (without compression); `
- *Targa 24, 32 bit* (without compression or with *RLE* compression)
- The first frame of any *AVI* file can also be used as a logo image (provided an appropriate codec is available in the *Windows* environment).

The transparency for 256-color files is set by the key color index (the ***Key index*** field). If the key color index is not specified, the logo is considered opaque. *True-Color* file transparency can be set using the *Targa 32-bit* format with the alpha channel. *Windows BMP* and *Targa 24-bit* format files are considered completely opaque.

Logo image scaling: *Width* **and** *Height*

The initial image stored in a graphics file can be of a wrong pixel sizes. The program allows to correct it, by specifying the necessary sizes in the ***Width*** and ***Height*** fields. The ***Get size*** button is provided for restoring the initial dimensions, if necessary.

It is certainly better if the image has the required dimensions from the beginning; professional graphic editors, like *Adobe Photoshop*, usually provide better scaling quality. The program features allow to select the image size first, and then to insert it without any additional scaling.

Logo vertical arrangement: *Baseline Shift*

A logo can be moved vertically with respect to the base line, specifying the image downwards shift in pixels in the ***Baseline shift*** field. If a negative number is entered in the field, the logo, on the contrary, is moved upwards.

Adjustment of the line height for a paragraph with a logo

The height of a logo is considered (along with those of all other paragraph characters) when the paragraph line height is calculated. If the line height becomes too big after inserting a logo, it can be adjusted using the ***Leading*** field in the ***Paragraph Attributes*** dialog box.

Horizontal alignment of a logo with respect to line text: *Width adjust*

When inserting a logo, the following text can be made to overlay it or, on the contrary, to be spaced from it. The size of such horizontal displacement of the following text in pixels is set in the ***Width adjust*** field. A negative value allows to place the following characters over the logo. On the contrary, positive values make following characters to move further away to the right from the image.

4.6. Text Character Attributes

Text attributes for the current position or a fragment of the script can be assigned in several ways:

Using the *Font Attributes* toolbar

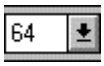
The ***Font Attributes*** style toolbar allows to set any attributes of characters directly changing the corresponding field values. Following the input of new values, do not forget to press ***Apply*** (or to select the ***Auto Apply*** option) to put the changes into effect. If the ***Apply*** button is not pressed, although the entered new values are displayed in the toolbar, the current text attributes remain unchanged, and the toolbar field values are restored immediately after the cursor moves.

Using the *Toolbar*

The ***Toolbar*** enables to set the following character parameters:

Bold typeface;

Italic typeface;



Font Size;

Font Name;

Style Name.

Setting character attributes through the toolbar does not require any additional confirmations, and is therefore faster than using the ***Font Attributes*** toolbar.

Using named styles

The use of the named styles allows to set all character attributes, selecting the style name on the ***Toolbar***, or in the ***Style name*** field of the ***Font Attribute*** toolbar. In the latter case, it pressing ***Apply*** is not sufficient; before doing this, it is still necessary to load (restore) the initial parameters of the chosen style by pressing ***Restore***, then the assignment can be made. For more details, see **4.7 Creation and Use of Named Styles**.

Setting of attributes during the keyboard input

If the cursor is inside the text, some current style attributes are always determined. Therefore it is not necessary to set character attributes during the typing, as they will be identical to the attributes of the previous character. To change the current style during the keyboard input, use any of the methods described above, and continue the input without changing the current position.

Setting of attributes for already entered text

If the text is already entered, its character attributes can be changed by the selection of the corresponding script fragment and the subsequent input of attributes in any of the ways described above. Thus, if any of the parameters had several different values in a fragment, all these values will stay constant if the corresponding field is not changed. This does not apply to the input using named styles. In this case, all parameters are redefined.

Copying of current character attributes

To transfer the current style attributes from one place of the script to another, select and copy to the *Clipboard* at least one character with required attributes, and insert it then into the required position. Then enter several characters starting from the position next to the inserted character and remove the first, superfluous character. Besides, the attributes can be saved in any named style (see below **4.7 Creation and Use of Named Styles**).

Editing of Character Typeface

The typeface only defines the outline of a text character, without considering its possible vertical and a horizontal displacement in line.

The character typeface is defined by the following parameters:

Font Name

Name of any vector font supported by the *Windows* environment.

This field can have an empty indefinite value.

The font name is additionally duplicated, and can be changed, on the ***Toolbar***.

It is important to note that the quality of the font strongly influences the character outline, especially when their height is small. Use license fonts, for example, those included into the *Windows* distribution kit.

Character height: *Size*

The font vertical size in pixels.

This field can have an empty indefinite value.

The value of the field is additionally duplicated, and can be changed, on the ***Toolbar***.

Typeface style: *Bold, Italic*

These buttons allow to set, or to discard the bold and italic typeface styles, respectively. The button act independently; their combinations correspond to *Normal, Bold, Italic,* and *Bold-Italic* typeface styles. If this field has an indefinite value, a gray square is displayed.

The buttons are also available in the ***Toolbar*** and the ***Character/Bold, Character/Italic*** menu items, where the corresponding boxes can be checked.

Width/height ratio: *Width*

This field defines the aspect ratio between text character width and height in percents. Note that the character vertical size remains unchanged, while their width varies. The parameter values can vary from 10% to 200%. This field can have an empty indefinite value.

200 % 100 % 50 % 10 %

Text underlining and background setting

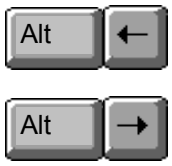
To underline a text or create a background for it, use *Box*-type background area frames. If the text is in the background layer, it can be outlined or underlined, by placing a frame above it. In a more convenient case, when the text is located within a *Frame*, it can have any position with respect to the underlining or background. Besides, selection of such frame and its underlining (outlining) frames as a group allows to move them within the page collectively.

The difference between the background and outlining frames is that the latter has a transparent internal area. See **5.2 Text Frame** and **5.3 Rectangular Background Area – *Box* Frame** for details on text frames and background area frames.

Character Alignment

Alignment of characters with respect to each other within a word or a line may sometimes have to be changed.

Intercharacter space: *Width adjust*



-30 -10 -40

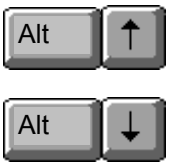
The value of the interval between characters of the text in pixels added to the default value (as set in the font). This interval can be increased or reduced for 2 adjacent characters on each side of the text cursor, or for the whole fragment. A negative value can be used to make characters touch or even overlay each other (manual *Kerning*). This method is used for a more compact positioning of characters in the line, taking into account the geometry of adjacent characters, which allows them to place them one under another.

The field can have an empty indefinite value.

Change of the intercharacter space for a text fragment: *Spacing*

Press *Alt*+→ and *Alt*+← to increase or decrease the intercharacter space for a whole fragment of the script text. In this case, spaces between characters of the whole fragment change simultaneously. Thus, the initial mutual adjustment of characters is preserved (words of the text look are extended or narrowed proportionally).

Vertical shift of characters with respect to the baseline:
Baseline shift



Vertical shift of the input characters with respect to the base line can be used to produce underscript (positive shift) or superscript (negative shift) characters.

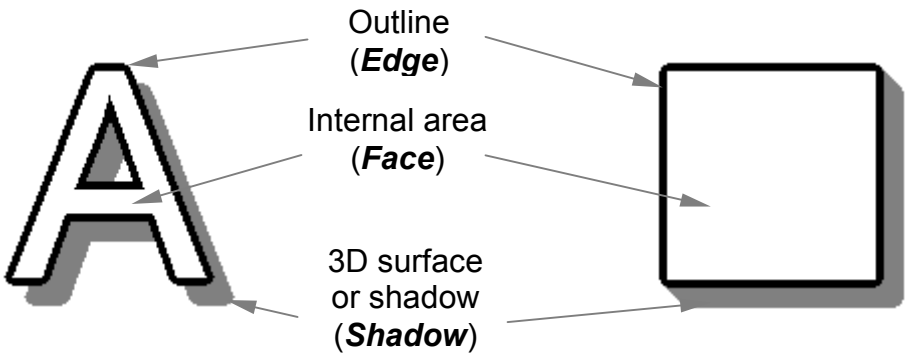
$x_1^2+y_1^2$

The shift size is measured in pixels, the field can have an empty indefinite value.

Changing of the vertical alignment with respect to the base line does not influence the line height (automatically calculated along with the paragraph line spacing) as far as shifted characters do not reach its top or bottom border. Further shift further increases the line height accordingly, but it can be adjusted, by changing the ***Leading*** parameter in the paragraph attributes (the ***Paragraph Attributes*** dialog box).

The line adjustment for a text fragment can be conveniently tuned by pressing *Alt*+↑ and *Alt*+↓.

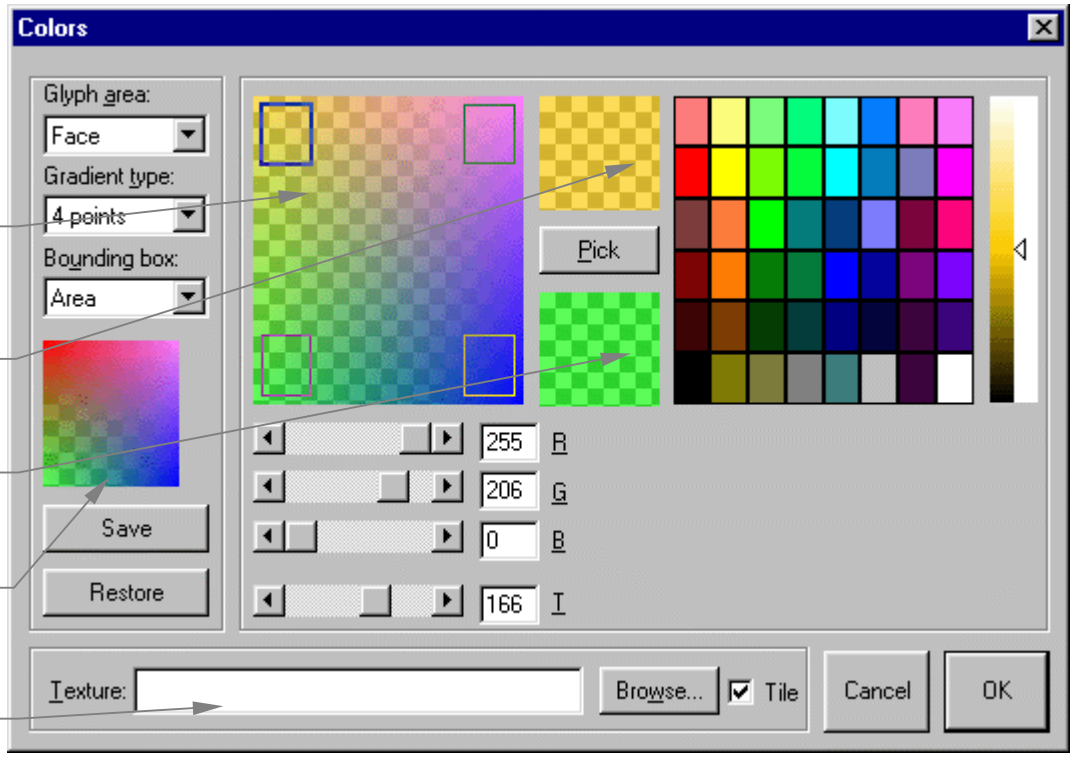
Character and *Box* Frame Design Elements



The design only defines the form of a text character contour – ***Face.*** The contour of a *Box* frame is rectangular. The contour can be "expanded" and the area of its expansion is the outline (***Edge***). Besides, a shadow (***Shadow***, *Offset*) or a side surface of some sort can be created by a contour translation in certain direction, which gives a character or a frame a three-dimensional look (***Shadow***, *3D*).

Color Filling of Drawing Elements

Filling color selection in the *Colors* dialog box



Many of the program dialog boxes allow to specify the character and *Box* frames filling color. To do so, always use the ***Colors*** dialog box.

RGB color definition

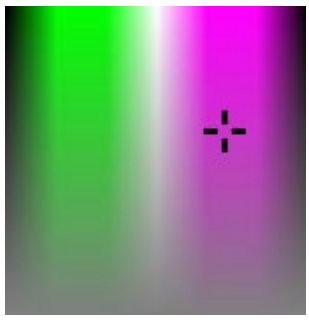
Use the ***R***, ***G*** and ***B***, fields, respectively, to set the red, green, and blue components of the color. The values of these parameters vary from 0 to 255. The black color corresponds to (0,0,0); and white, to (255,255,255).

Color transparency definition – *T*

The ***T*** field is used to define the degree of transparency. The value of this parameter can vary from 0 to 255 (full transparency corresponds to 0; complete opacity, to 255). If transparency is present, gray squares on white background appear in the square fields displaying the color.

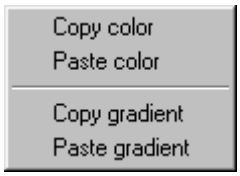
Use of the color palette

The ***Colors*** dialog box contains 48 basic colors in the upper left corner. To select a color, simply click the mouse left button on the corresponding color. Clicking the mouse right button on this palette transforms it to another one, containing all colors, and vice versa. A slider to the right of the palette is used for adjustment of the color brightness degree.



Saving of the current color

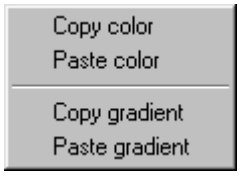
The current color field lays to the left and above the palette, and the saved color field is below it. To select this color, (together with the corresponding transparency) click the mouse left button on this field. Saving in this color field is provided by the ***Pick*** button.



The current color and transparency can also be saved or copied, by pressing the mouse right button and selecting ***Copy color*** or ***Paste color***, respectively.

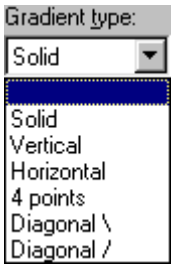
Saving of the current color filling

The color filling is specified in the center of the dialog box and it can be saved by pressing ***Save*** in the lower left corner. The saved filling is displayed a little higher. It is possible to read by pressing ***Restore***, which redefines the current filling.



The current filling can also be saved or copied, by pressing the mouse right button and selecting ***Copy gradient*** and ***Paste gradient***, respectively. A filling is saved as whole, along with all additional attributes described below.

Color filling gradient types



The ***Gradient type*** field used to define the type of the color filling gradient offers the following choice of gradient structure types:

	*Solid*	solid one-color filling
	*Vertical*	vertical gradient (two reference points at the top and bottom edges)
	*Horizontal*	horizontal gradient (two reference points at the right and left edges)
	*4 points*	based on 4 reference points in the corners
	*Diagonal *	based on two reference points at the upper left and lower right corners
	*Diagonal /*	based on two reference points at the lower left and upper right corners

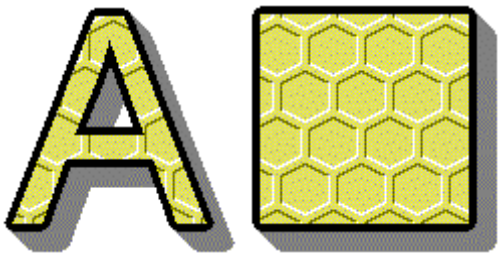
The ***Gradient type*** field can have an empty indefinite value.

Color filling reference points

Up to 4 reference points or areas are displayed as rectangular frames in the current color filling window. To change the color of such point, select the corresponding frame with the mouse left button, than adjust its color and transparency.

Textural filling

Files of the same formats as for frames and logo characters can be used to define a texture map in the ***Texture*** field:



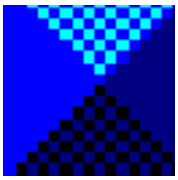
- *Windows BMP 8, 24 bit* (without compression);
- *Targa 24, 32 bit* (with or without a *RLE*-format compression).

The ***Browse*** button is provided for easier navigation and selection of graphics files as texture maps.

If a file name is indicated at the ***Texture*** field, the values of ***RGB*** fields for all reference points are ignored.

Area scaling or tile-filling with texture: *Tile*

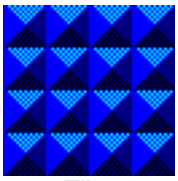
Maps scaling is only made if the ***Tile*** box **is not checked**. In this case, map is first resized according to the dimensions of the used area, and then the actual filling is performed.



No Tile

If the ***Tile*** box **is checked**, a mismatch between the area and the file sizes, results in a cutting from the right and bottom sides, or in a tile-filling by the texture (*Tile*).

The examples at the left show a 64 x 64 *Box* frame filled by the 16 x 16 pixels texture from the file *Treug.bmp* with scaling (above), and without it, on the tile principle (below).



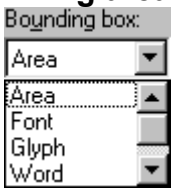
Tile

Semitransparent textural filling

A semitransparent texture can be set in two ways:

- if a *Windows BMP* or *Targa 24* format map file is used, i.e. a file without an alpha channel, the transparency is specified in the ***T*** fields, for each reference point separately;
- if a *Targa 32 bit* format file with an alpha channel is used, it sets transparency of points, and the ***T*** fields values are ignored.

Relative position of character drawing element and filling area



Size and layout of the filled area with respect to the character is an important factor of a color or textural filling of a text. The ***Bounding Box*** field is used to specify this layout. Whether the *Font* value is selected in this field or not is another essential factor for text characters. If the *Font* value is selected, the vertical filling area begins from upper border **of the font**, and ends at its lower border, irrespective of character heights. The *Font* value allows to align all filling areas vertically.

These considerations are very important for textural filling. For instance, a tile filling should be aligned vertically for consequent characters. When scaling texture is applied, the vertical scaling coefficient for the large and small characters must be the same.

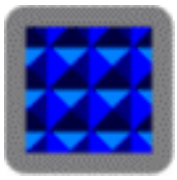
If the value selected in the ***Bounding Box*** field is different from *Font*, the filling area size for each character is individual and defined by the real size and layout **of the character**. Thus, for example, the area for capital letters is larger than for lowercase letters, and starts higher above. It can clearly be seen on the figure: e.g., the filling for characters "*y*" and "*p*" is different from that for other letters.

Relative position of the *Box* frame drawing element and filling area

The value of the ***Bounding Box*** field also matters with a similar color or textural filling of *Box* background areas. If the *Font* value is selected in this field, the filling area fully complies with the sizes and layout of the outline area (*Face*, *Edge* or *Shadow*). In this case, the *Face* area filling can begin not from the *Box* upper left corner, but after a certain space defined by the sizes of the border and the shadow.

Font

If the value selected in the ***Bounding Box*** field is different from *Font*, the filling area size and layout fully complies with the background area rectangle, together with its shadow and border. The upper left corner of the filling area then coincides with the *Box* upper left corner. This is frequently used when one texture file is used for the internal area, border area, and/or shadow area.



Glyph

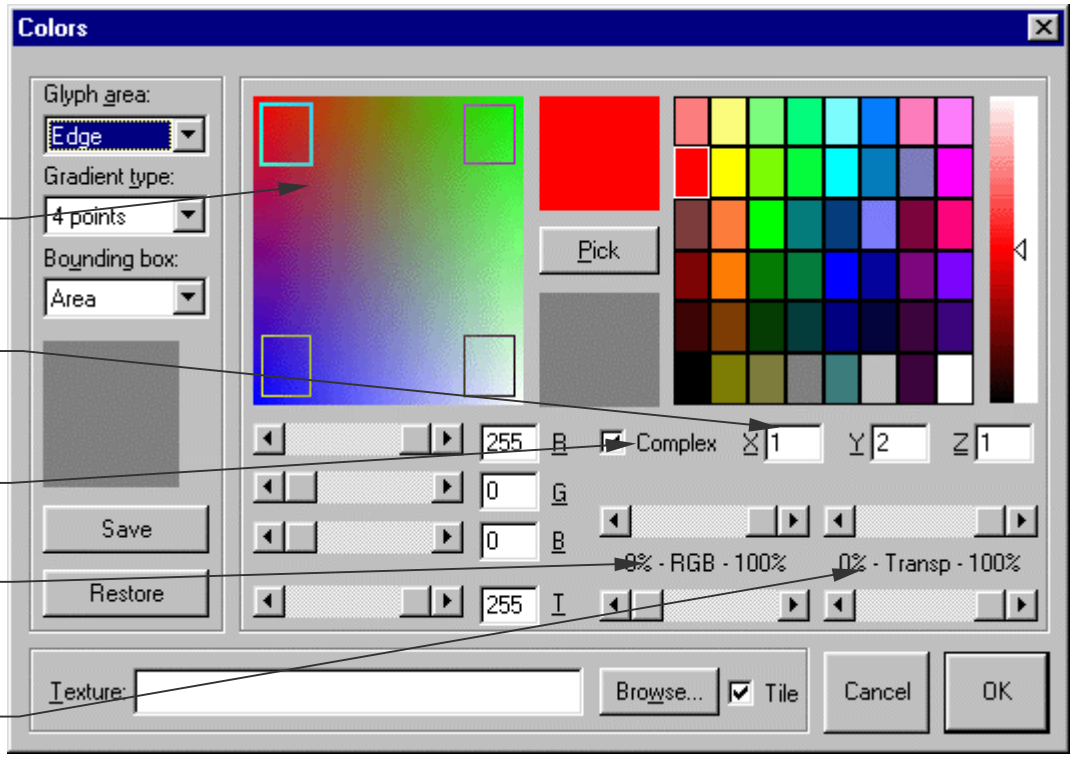
Drawing element selection for filling: *Glyph Area*

The two rectangles shown in the figures have a 64 x 64 pixel background and a 8 pixel wide border. The texture file *Treug.bmp* used for the *Face* area is 16 x 16 pixels large. The upper figure demonstrates that the texture starts at the upper left corner of the internal area, while in the lower figure it begins at the upper left corner of the *Box*.

Color filling parameters of the three character or background frame drawing elements (i.e., *Face, Edge,* and *Shadow*) can be edited without closing the ***Colors*** dialog box. The element type is selected from the ***Glyph Area*** list.

3D-Realistic Drawing Elements Color Filling

The *Colors* dialog box for 3D filling



The program allows to create special kinds of effects, where drawing element point colors are defined not only by the standard color or textural filling, but also by the type of element. It concerns the *Embossed* and *Glow* borders, as well as the *Extruded 3D-surface* side face. The light source coordinates (***X, Y, Z***) adjustment is only available for an *Embossed* 3D. Additional parameters of such effects are allocated in the lower right corner of the ***Colors*** dialog box, but they only appear when an appropriate type of the border or side surface is chosen.

3D color filling complex mode

Normally, the complex mode of filling is used (the ***Complex*** box is checked). In this mode, in addition to the standard color or textural filling as described above, the area point brightness and transparency is changed so as to achieve the necessary effect. When this box is checked, the effect color brightness (***RGB***) and transparency channel (***Transp***) coefficients can be adjusted. The figure on the right demonstrates these parameters for the *Embossed* border.

Customization of a *3D*-effect color: *RGB*

The drawing element point brightness depends on the effect type and light source position. The light source is considered white, therefore a blending with black in the defined proportion, i.e. depending on the position of the side surface point in respect to the main outline, takes place. Its brightness decreases for a defined value, and the ***RGB*** sliders determine how much the brightest and the darkest points of the surface are blacked out. This sets a range of this variation, by specifying the upper and lower brightness limits in percents. Thus, the upper slider allows to reduce the common brightness of the filling area points. The lower slider allows to move the lower limit of point brightness up, preventing them to become too dark.

This effect appears most strikingly when the lower slider is set at 0%, and the upper one, at 100%. If the slider values are identical, the effect does not show at all – the brightness just changes all over the area, according to their position. Decreasing the upper slider value produces an illusion that there light source becomes less bright. Contrariwise, with a raising of the lower slider value, the light becomes brighter (less contrasting). Decreasing of the difference between the sliders creates an illusion of a more scattered or distant light source.

Customization of a *3D*-effect transparency channel: *Transp*

The drawing element area point transparency can be changed according to the same principles as brightness. If both ***Transp*** sliders are set at 100%, the transparency preset in the main color or textural filling is preserved unchanged and does not depend on the point position with respect to the main outline. If other values of sliders are set, the point, which should be darker by the rules for colors, it is also more transparent, according to the position of the lower slider. Transparency of the brightest points is multiplied by the number set by the upper slider. These sliders can be used with the *Embossed* effect, for example, to create realistic transparent surfaces, when their transparency depends on the sight angle of the observer.

***3D*-effect light source direction:** *XYZ*

The ***X, Y, Z*** fields define the light source direction for the *Embossed* effect. Values of these fields (except ***Z***) can be positive or negative integers. The *X* axis direction is from left to right, *Y* axis is directed from downwards, and *Z* axis towards the observer. The relation between the values of these fields only matters; so the values (1,1,1) and (2,2,2) are completely equivalent. ***X*** and ***Y*** values define the direction towards the light source in the plane of titles, and the ***Z*** parameter sets the height of the source above the horizon of this plane. Thus, the light direction can be changed for an opposite one by changing signs of the ***X*** and ***Y*** field values.

3D filling with an unchecked *Complex* box

If the complex filling mode is not set (the ***Complex*** box is unchecked), the additional ***X, Y, Z*** fields are only available for editing (for the *Embossed* effect). Besides, only vertical color filling without texture can be used as a main filling (the *Vertical* value in the ***Gradient type*** field). Actually, the *Vertical* value only means here that there are two colors, with preset transparences, and the brightest points of the three-dimensional filling correspond to the values of the lower one, and the darkest points, to the values of the upper one. Other points of the area correspond to the intermediate colors and transparency values in the gradient. Using this color filling mode can make sense, for instance, if a non-white light source is to be represented. This can be done by using various tone gradations of this color for reference points of the main filling.

Internal Area Parameters – *Face*

Color filling

Only the color filling is specified (in the corresponding ***Colors*** field) for internal area of a character drawing or a frame. This field is usually located closest to the parameters defining the character outline. It is described in detail above (see **Color Filling of Drawing Elements**)

Outline Area Parameters – *Edge*

The border area parameters are located at the ***Font Attributes*** toolbar or in other similar windows, grouped under the name ***Edge***.

Border type: *Type*

The outline type is selected in the ***Type*** drop-down list, offering the selection of:

- *None*: no outline (in this case, all other fields of the ***Edge*** group are unavailable for editing);
- *Outline*: a border with a regular area filling;
- *Embossed*: a border area filling looking as if produced by extruding and illumination;
- *Glow*: a border filling with blurring with the distance from the internal area.

The field can have an empty indefinite value, which occurs when at least one of the below attributes has several different values within the text fragment. In this case, these attributes cannot be assigned uniformly for all the fragment, and should be changed separately for each character.

Border corner shape: *Shape*

The shape of the outline corners is set in the ***Shape*** drop-down list. For *Outline* and *Glow* outline types, the selection contains:

- *Round* – the round shape;
- *Square* - the square shape;
- *Rhomb* - the rhombic shape.

For the *Embossed* outline, the outline corner extrusion can have one of the following shapes:

- *Sphere* - spherical;
- *Concave* - concave;
- *Cone* - conic;
- *Pyramid* - pyramidal.

Border construction mode: *Mode*

The border construction mode is selected in the ***Mode*** drop-down list, which offers the selection of:

- *Normal* – the normal mode of border construction, when the method of its construction and color filling by any image does not influence the filling of the outline internal area;
- *Complete* – the border is created from the outside of the outline, and the construction method and color filling completely defines the color and textural filling of its internal area (the corresponding ***Colors*** field values are ignored);
- *Bidirectional* - the border is created from both the outside and the inside of the outline; without filling the internal area, which remains completely transparent (the corresponding ***Colors*** field values are ignored).

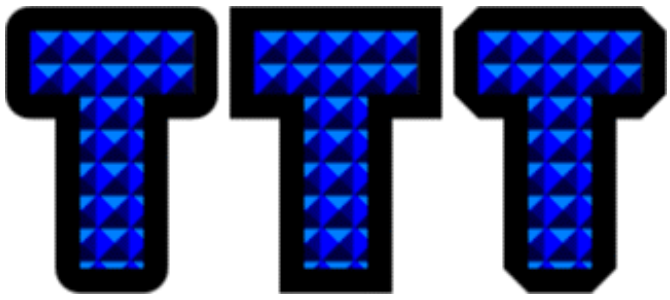
The *Complete* mode can be conveniently used with textures, as in this case the filling affects all common area – both the interior and the border.

Area size: *Size*

The ***Size*** field comprises the border area width in pixels. Border area for *Box* frames is built inwards, so it should not be too large to not overlap the frame by itself. For characters, on the contrary, the border is built outwards, starting from the outline.

Simple *Outline* border and its *Round*, *Square* and *Rhomb* shapes

An *Outline* border can be drawn in several ways, with round, square or rhombic corners. The border corner shape is set in the ***Shape*** field, which can have the *Round*, *Square* and *Rhomb* values, respectively.



Round *Square* *Rhombic*

Bidirectional and *Complete* modes of the *Outline* border construction

An *Outline* border is created in the *Bidirectional* mode both outside and inside of main outline, therefore it twice thicker. The interior area remains completely transparent (see the figure on the right).

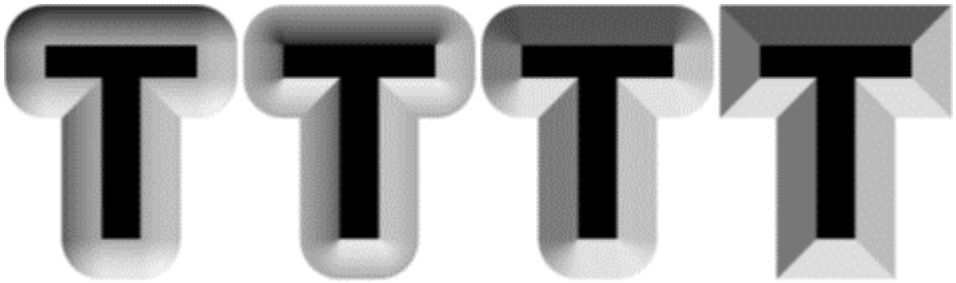
In the *Complete* mode the interior is filled too with filling of its border area.

Round *Square* *Rhombic*

Creation of boundary characters

Characters consisting only of an outline can be created by specifying an opaque border and a transparent internal area – ***T*** = *0*. Another way is to use the *Bidirectional* mode of border construction, but in this case, its width should be reduced twice.

***Embossed* border with extruding and its *Sphere, Concave, Cone* and *Pyramid* forms**



Sphere *Concave* *Cone* *Pyramid*

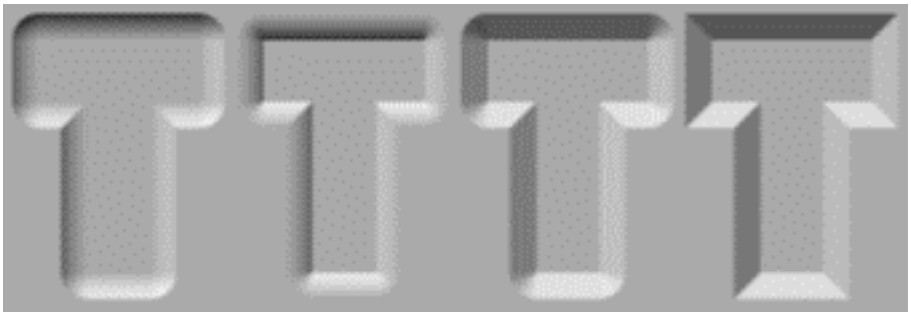
There are four methods to produce a border with an *Embossed* extruding, defining its form as spherical, concave, conic or pyramidal. The form of the extruding is set in the ***Shape*** field, which has the *Sphere, Concave, Cone* and *Pyramid* values, respectively.

Actually, the extruded border characters look similar to three-dimensional figures.

Parameters of border construction for the *Embossed* type extruding are described in detail above (see **3D-Realistic Drawing Elements Color Filling**). If this border type is used for various characters of the text on page, they all should use the same light source direction and contrast.

Characters with conic and pyramidal border forms can be considered, not "extruded" out of the plane towards the observer, but on the contrary, "cut" into the plane, and have, for example, a transparent internal area. The light direction in this case changes to opposite.

***The** Complete* **and** *Bidirectional* **modes of an** *Embossed* **border construction**



Sphere *Concave* *Cone* *Pyramid*

When an *Embossed* border is built in the *Complete* mode, the interior area is filled with the same filling, and simultaneously with the border area. In this case, the common brightness of all internal area points changes so as to create an illusion of the upper plane of the "displaced" character. This mode is most convenient when a “displacement” with a textural filling is to be produced.

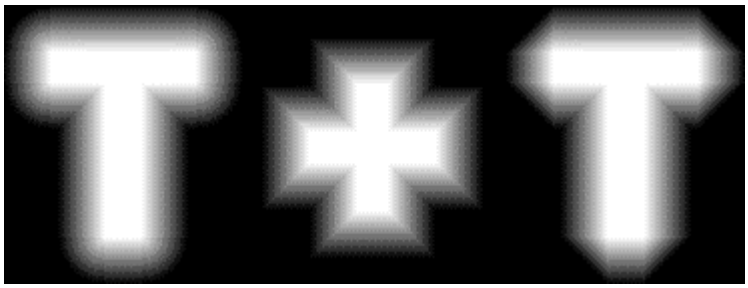
Sphere *Concave* *Cone* *Pyramid*

An *Embossed* border in the *Bidirectional* mode is built both outside and inside of the main outline, therefore the resulting border has a double thickness. In this case, the interior remains completely transparent. Since the border is also built inwards the outline, its sufficient thickness can result in a complete or partial overlap of the internal area by the border.

To construct the plane correctly, from which the displacement of the characters takes place, one can use a *Box* background area with the same filling style, and any, down to the minimum value of 1 pixel, border size. To produce a more realistic look, fillings may be diversified, using, for instance, a slightly rough texture of the same colors.

The *Glow* diffusion border and its *Round*, *Square*, and *Rhomb* forms

A *Glow* Border can be drawn in different styles, so that its corners would have rounded, squared or rhombic shapes. The border corner shape is set in the ***Shape*** field, which can have the *Round*, *Square,* and *Rhomb* values, respectively.



Round *Square* *Rhomb*

The *Glow* border filling in the complex mode

Normally (and by default) the complex mode of filling building is used – with the ***Complex*** box checked. In this the mode, in addition to a normal color or textural filling, area point brightness and transparency change according to their distance from the main outline. When this box is checked, the effect color brightness (***RGB***) and transparency channel (***Transp***) coefficients can be customized.

The ***RGB*** sliders set the degree of darkness (mixing with black) for the points of the border area closest to, and the most distant from, the outline. They set the range of this variation, specifying the maximum and minimum values in percents. Thus, the upper slider allows to change the brightness of the filling points closest to the outline, and the lower slider, that of the remotest points.

The ***Transp*** sliders set, similarly, the degree of the transparency for the border points closest to the outline and remotest from it. They set the range of this variation, specifying the maximum and minimum values in percents.

The effect is manifested most strikingly when the lower slider is set to 0%, and the upper one, to 100%. If the values of all sliders are identical, the effect is not shown at all; the transparency and brightness vary all over area, according to the position of the points.

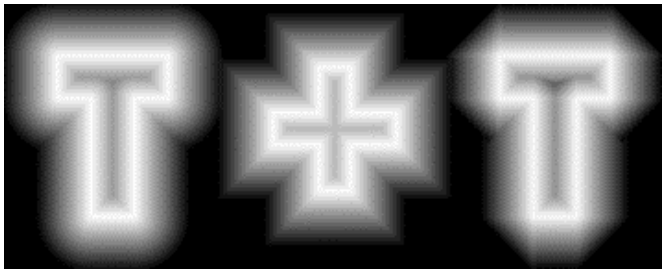
The *Glow* border filling with the *Complex* box unchecked

If the complex filling mode is not set (the ***Complex*** box is unchecked) only vertical color filling without texture can be used as the main one: the *Vertical* value is set in the ***Gradient type*** field. Actually, the *Vertical* value only means here that there are two colors, with preset transparences, and the brightest points of the

three-dimensional filling correspond to the values of the lower one, and the darkest points, to the values of the upper one. Other points of the area correspond to the intermediate colors and transparency values in the gradient. Using this color filling mode can make sense, for instance, if the distance of the area points from the main outline is supposed to change their color shade, and not just the brightness.

The *Bidirectional* and *Complete* modes of a *Glow* border construction

A *Glow* border in the *Bidirectional* mode is built both outside and inside of the main outline, therefore the resulting border has a double thickness. In this case, the interior remains completely transparent where it is not overlapped with a border (see the figure on the right).

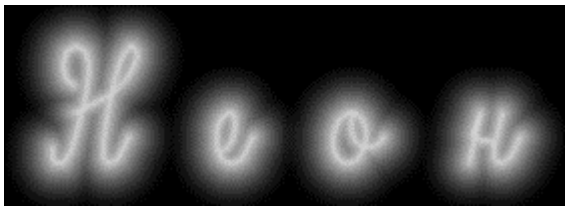


Round *Square* *Rhomb*

In the *Complete* mode, the interior is filled together with the border area, and with the same filling.

Creation of "neon" boundary characters

To create "neon" characters effect, a font is to be selected with the characters represented by outline only (for example, *Script*).



Font manufacturers also create special *Outline* type fonts, (e.g., *Antiqua Outline* and others). Characters of these fonts are represented by external and internal outlines, and not filled inside. They can be used to create "neon" characters in the same way as with the help of the *Bidirectional* mode, allowing to use normal fonts. Such special fonts allow to fill the character outline with other color than used for the *Glow* border.

To create the effect with the use of special fonts, a border is to be selected of a sufficient size and a round shape (the ***Shape*** field set to *Round*). The internal filling is made solid, with the same (or brighter) color as at the beginning of border gradient. Besides, a transparency gradient should be defined, for example, from ***T***=*200* to ***T***=*50*.

Parameters of a Shadow Area or a 3D Side Surface – *Shadow*

Shadow area parameters are located in the ***Font Attributes*** toolbar or other similar windows, and grouped under the name ***Shadow***.

Shadow type: *Type*

The type of a shadow or side surface of the character is set in the ***Type*** drop-down list, offering the following selection:

- *None*: no shadow (in this case, all other fields of the ***Shadow*** group are unavailable for editing);
- *Offset*: the ordinary method of filling the area of the falling plane shadow when a character and its shadow look as two separated plates;
- *Drop*: the ordinary, method of a solid filling of the area of a side surface. A "rising" character creates a filled surface, without taking into account the "height" of corresponding points;
- *3D*: the realistic method of filling the area of a side surface. A "rising" character creates a surface broken into sub-areas, according to their "height", and each of these sub-areas is filled individually;
- *Extruded*: a method of area filling of side surface using an "extruded" character, which creates a surface with a gradient filling in a preset direction, and according to the "height" of the points;
- *Soft*: a method of filling of an area of the falling plane shadow similar to the *Offset*, but the shadow on edges is made blurred.

The field can have an empty indefinite value, which occurs when at least one of the below attributes has several different values within the text fragment. In this case, these attributes cannot be assigned uniformly for all the fragment, and should be changed separately for each character.

Shadow direction: *Direction*

The direction of the shadow is set, with an accuracy of 45 degrees, in a group of 8 radio-buttons. Note that characters are drawn from left to right, and a very big size of a shadow directed to the left can make it overlay the previous character.

Shadow area shift value: *Size*

The ***Size*** field defines in pixels the shift of the shadow area in the direction defined by the ***Direction*** radio-buttons. The shadow or a character side surface is produced by a shift of the internal area and the border in this direction for the distance defined by the ***Size*** value.

The *Offset* falling shadow

A falling shadow is created by a repeated drawing of the internal and border areas with a certain shift. This area can be filled arbitrarily with continuous, gradient or textural fillings. A certain additional degree of transparency specified for the shadow area may add to the realistic effect.

The *Drop* side surface

A side surface of the *Drop* type appears when a character is raised and shifted in a certain direction creating a surface filled in a standard way, without considering the "height" of the points. To create a realistic effect, textural fillings can be used taking into account the selected direction.

The *3D* side surface

A *3D* side surface is produced similarly, as a character is raised and shifted in a specific direction creating a filled surface, but the "height" of the surface points is taken into account. For a realistic effect, gradient fillings can be used, allowing to place the lines with points of the same color along the chosen direction (specified by the ***Direction*** parameter). In this case, the outline interior is first filled fictitiously, according to the color or textural filling, then border points are multiplied in appropriate numbers in the specified direction creating the side surface. The two previous figures show the result of use of a vertical gradient from black to white with the *Drop* and *3D* types of side surface, respectively.

The same applies to the gradient filling with an alpha channel.

The *Extruded* side surface

An *Extruded* side surface is built as a character is raised and shifted in a specific direction creating a filled surface and taking into account the "height" of the surface points, but in the direction, perpendicular to that used in the *3D* surface. To produce a realistic effect, gradient fillings should be set. In this case, lines with points of the same color are placed in the planes perpendicular to the selected direction (specified by the ***Direction*** parameter). The same applies to the gradient filling with an alpha channel.

This type of side surface can be used to create an effect of "diffusing tails" for characters.

The *Extruded* side surface filling in the complex mode

Normally (and by default) the ***Complex*** box I checked, and the complex mode of filling construction is used. In this mode, in addition to normal color or textural filling, area point brightness and transparency change according to the point distance from the main outline. When this box is checked, the effect color brightness (***RGB***) and transparency channel (***Transp***) coefficients can be customized.

The ***RGB*** sliders set the degree of darkness (mixing with black) for the points of the border area closest to, and the most distant from, the outline. They set the range of this variation, specifying the maximum and minimum values in percents. Thus, the upper slider allows to change the brightness of the filling points closest to the outline, and the lower slider, that of the remotest points.

The ***Transp*** sliders set, similarly, the degree of the transparency for the border points closest to the outline and remotest from it. They set the range of this variation, specifying the maximum and minimum values in percents.

The effect is manifested most strikingly when the lower slider is set to 0%, and the upper one, to 100%. If the values of all sliders are identical, the effect is not shown at all; the transparency and brightness vary all over area, according to the position of the points.

The *Extruded* side surface fillings with the *Complex* box unchecked

If the complex filling mode is not set (the ***Complex*** box is unchecked) only vertical color filling without texture can be used as the main one: the *Vertical* value is set in the ***Gradient type*** field. Actually, the *Vertical* value only means here that there are two colors, with preset transparences, and the brightest points of the three-dimensional filling correspond to the values of the lower one, and the darkest points, to the values of the upper one. Other points of the area correspond to the intermediate colors and transparency values in the gradient. Using this color filling mode can make sense, for instance, if the distance of the area points from the main outline is supposed to change their color shade, not just the brightness.

Drawing Quality: *Quality*

When the drawing quality is to be improved

The video card buffer is usually rather small, about 768 x 576 dots. The duration of the string visible part about 52 microsecond gives the size of the TV screen monitor pixel of about 65 ns, which makes the jags on inclined and rounded lines clearly seen.

The program provides some built-in mechanisms of elimination of this defect. To choose one of them, use the ***Quality*** field in the ***Font Attributes*** toolbar. The *Extra* value corresponds to the maximum quality, when the effective size of a pixel is reduced to the value of about 8 ns, and the jags are certain to cease to be visible.

Beside the case of slanted lines, the quality improvement is required at drawing of small characters. The highest quality (*Extra)* can be necessary to render them well readable while preserving the correct shape.

Raw *Simple* *Medium1*

The *Raw* value specifies no smoothing. The *Simple* mode produces a higher-quality text character mapping due to the smoothing of jags on boundaries *(Anti-aliasing)*. The use of this mechanism may, however, result in certain "squareness" of the character rounded parts. To eliminate this defect, the drawing quality must be improved up to, for instance, the *Medium1* level.

Other values of this attribute are *Medium2*, *HQ* and *Extra*, corresponding to even higher-quality drawing, can be required for character of smaller size (the figure shows an example of 100-pixel character), or for the *Sharp* level of smoothing (the ***Sharpness*** parameter; the value in the figure – *Normal*).

This field can have an empty indefinite value, which means that characters of different drawing quality are present in the text fragment. In this case, the smoothing level cannot be changed for the whole paragraph, either.

How to avoid the work deceleration at the script preparation

Improvement of the drawing quality may multiply the time needed for a script rendering; therefore this resource should be used carefully. A user watching a TV program may not notice the difference between the characters drawn with different ***Quality*** attributes. Different script elements and text fragment can also have different levels of drawing quality defined experimentally.

Besides, the smoothing may not be used during script editing, for a quick results preview on the TV screen; it can be specified later, after the page layout is complete.

Smoothing of Transitions between Drawing Areas: *Sharpness*

Sharp *Normal* *Blur*

The value of this field determines the level of color "diffusion" and transparency (in all directions) on the boundaries of various areas of the character or frame outline. The *Sharp* value corresponds to no

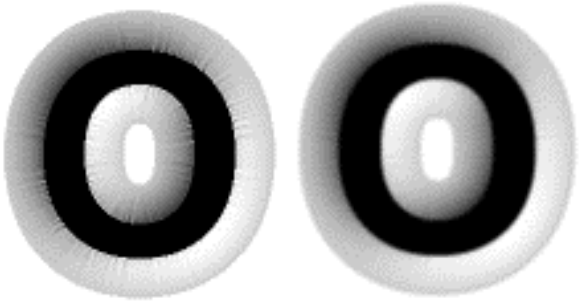
diffusion, while the *Normal* and *Blur* values set different levels of the diffusion. If the value of the ***Quality*** field is *Raw* or *Simple*, the *Normal* value is equivalent to *Sharp*.

The main purpose of this field is the use of the *flicker*-filter, which allows to partially suppress the flicking on boundaries for an output to an interlacing TV screen. If the video device supports an alpha channel, suppressing of the flicking also takes place on the graphics/passing video signal boundary.

This field can have an empty indefinite value, which means that characters of different levels of smoothing are present in the text fragment. In this case, the character drawing quality cannot be changed for the whole paragraph, either.

Smoothing of wrinkles in the *Embossed* and *Glow* border areas

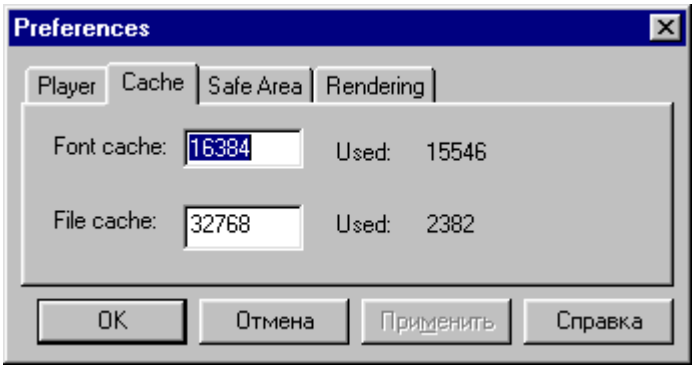
To smooth the wrinkles arising at the use of *Embossed* or *Glow* border types, the *Blur* value in the ***Sharpness*** field is used. The necessary smoothing level is defined experimentally and depends on the border area width. For example, for a normal text, the *Normal* value is sufficient.



Medium1, Sharp *HQ, Normal*

Setting of the Cache Memory Buffer Size for Character Drawing and Page Display: *Video/Preferences/Cache*

The setting of maximum capacity of the cache memory buffers is carried out in the ***Cache*** parameter group of the ***Video/Preferences*** dialog box; it is used by the program for fast redrawing of characters during editing (***Font cache)***, as well as for storing in memory of intermediate results of script processing (***File cache)***.



Dependence of editing speed on the size of the cache memory buffer

Sizes of these buffers are defined empirically, depending on the text of the script and on the RAM capacity. In the size is insufficient, certain time is required for page redrawing or change during editing. If the buffer size exceeds a certain value, these operations are performed practically instantly. The ***Used*** field helping to make a correct choice of the optimal size of the buffer displays how many bytes of the buffer are used in the moment.

Recommended sizes of cache memory buffers

For 64 *MB* RAM, the recommended value for the ***Font cache*** is 16,384 bytes, and for the ***File cache***, 32,768 bytes. If the computer has, for example, twice as much memory, the sizes of cache memory buffers can also be set 2 times larger. On the other hand, if less memory is available, the buffer sizes should be reduced accordingly, to allow the *Windows* environment operate normally.

4.7. Creation and Use of Named Styles

The use of named styles

It is often useful to preserve the most appropriate character and background area typefaces and fillings created in the process of work. The program allows to save the corresponding attributes under some user-defined style name. The attributes are stored along with the name in the same script separately from the text or the area background, and text characters (or background areas) are additionally associated with the name of their style. It means that at any moment the attributes of characters (or background) can already be different from the attributes of their style. To restore the conformity, when needed, select the text characters or the corresponding background area and press ***Restore*** and ***Apply*** in the ***Font Attributes*** toolbar.

Styles can be copied from one script to another, which allows to use separate scripts as libraries of styles.

Style creating and editing using the *Font Attributes* toolbar

To create a new style using the ***Font Attributes*** toolbar, enter its name in the ***Style Name*** field and specify other attributes, then press ***Save***. Existing styles can be used to set and edit text character attributes. If a style already exists in the script and had the attributes different from those saved, an addition confirmation of operation is requested before the style us updated.

To edit an already existing style, its attributes should be restored (by pressing ***Restore***) before any correction is applied to them. This is necessary since the ***Font Attributes*** toolbar reflects the current character attributes even if it belongs to the same style.

Initially, there always exists a style in the script predetermined by the program and named *Normal.* The attributes of this stile can further be edited and redefined.

Style setting using the *Toolbar*

The simplest way to set a style for the current position, a text fragment or the current background area is to choose its name in the corresponding window of the ***Toolbar***. This setting takes place automatically without any additional confirmations. It is also the most convenient way if a specific style was edited, and all text fragments referring to its name are to be changed respectively.

Style setting using the *Font Attributes* toolbar

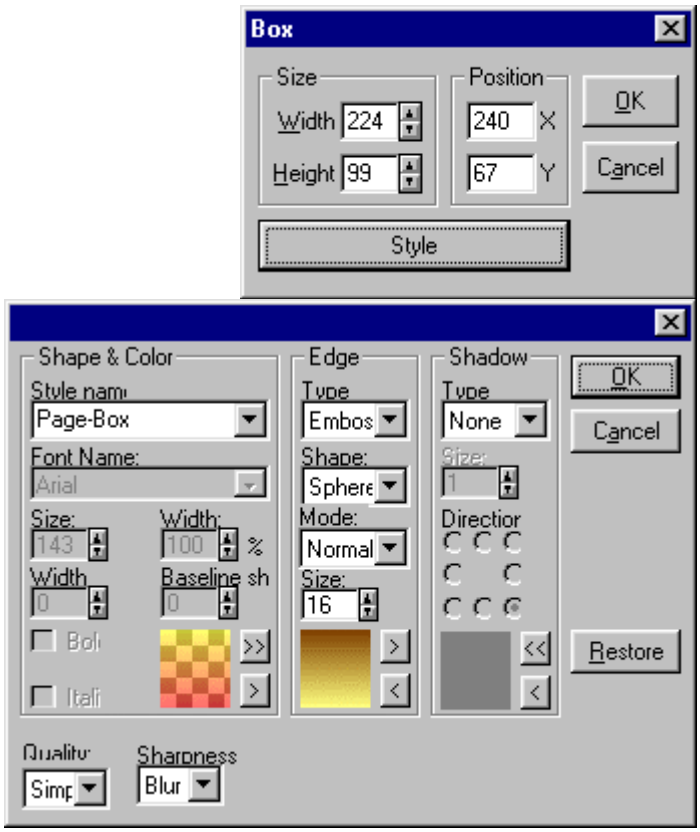
Alternatively, the ***Font Attributes*** toolbar can be used. It allows to redefine, for example, only the name of a style, without changing other character attributes. If all remnant style attributes are to be restored, use the ***Restore*** button located in the toolbar. To make the attributes valid, do not forget to press ***Apply***.

Copying of character attributes

The style attribute restoring button can be conveniently used, to transfer character attributes from one place of the script to another, without copying anything to the *Clipboard*. To do this, create a working style named, for example, *Temp*, use it to save all necessary attributes, and then restore them in the required place using the ***Restore*** button.

Assignment of styles to *Box* background frames

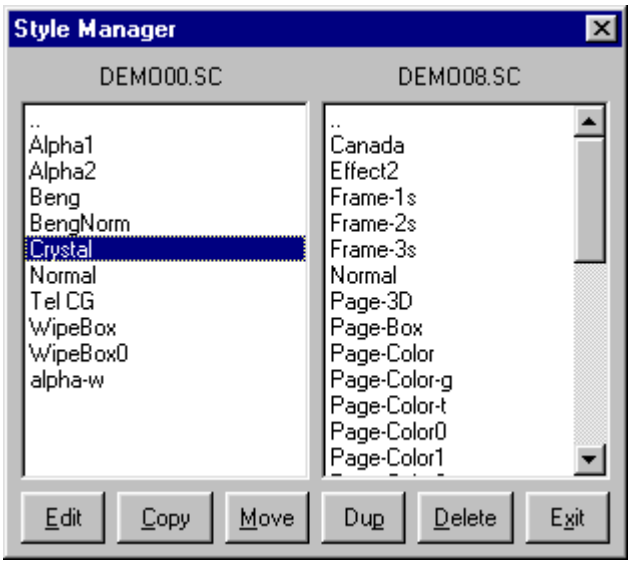
Named styles can also be used *Box* background area frames. To create and edit such styles use the same methods, as for characters via the ***Font Attributes*** toolbar or the ***Edit*** command of the ***Style Manager*** dialog box. Editing and assignment of attributes is performed in a dialog window, opened by pressing the ***Style*** at the frame parameter adjustment dialog box.



This window has the same set of parameters as the ***Font Attributes*** toolbar, except the font typeface attributes. To set a named style, select its name in the ***Style name*** field and press ***Restore*** and ***OK***. This window allows to edit parameters of any style, but it is impossible to save the performed changes using it.

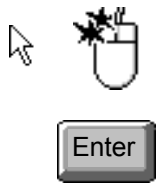
Style Operations Using the *Style Manager* Window

The ***Character/Styles*** subitem allows to manage the styles of the scripts currently opened in the program using the ***Style Manager*** dialog box.



Commands of this dialog box allow to perform various operations with the styles of the edited scripts. It is important to note that an automatic updating of attributes for all characters in the edited script referring to the current style is impossible; it has to be done separately: for example, using the ***Font Attributes*** toolbar, or the corresponding window of the ***Toolbar***.

Selection of scripts and styles by their names

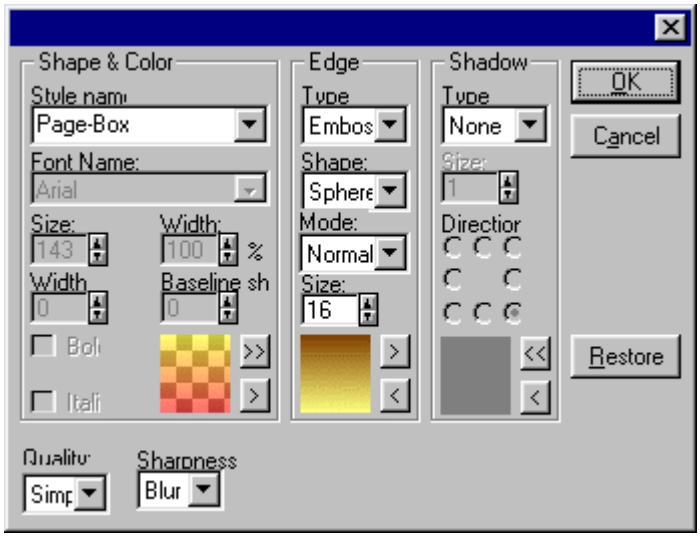


The dialog box comprises two panels allowing to select scripts to be edited and their styles. The "*Choose document...*" message is displayed until one of the scripts in the panel is selected. To select a script, position the cursor on its name using the mouse or the keyboard and press *Enter* (or double-click the mouse left button). The name of the script then appears above the panel, and styles contained in the script are listed in the panel. To return and select another script, select the "*..*" line.

To perform any operation with a style, put the cursor on the style name in one of panels and press the corresponding command button at the bottom of window.

Editing of style attributes using the *Style* window

The ***Edit*** button opens the ***Style*** dialog box, which allows to edit all attributes of the chosen style except its name. To style editing can also be entered by the mouse left button double-click on its name in the corresponding panel.



This window contains the same set of parameters as the ***Font Attributes*** toolbar, except that it cannot be used to change the style name.

Style copying from one script to another

The ***Copy*** button allows to copy a style from one edited script to another. To do this, open the corresponding scripts in the window panels. Then select the style to be copied in the panel corresponding to its script and press ***Copy***. If a style of such name already exists within the script where it is supposed to be copied, and has different attributes, it an additional confirmation request is issued for the replacement of the old style with the new one.

Style moving from one script to another

The ***Move*** button is used to move a style from one edited script to another, deleting it in the old script. To do this, open the corresponding scripts in the window panels. Then select the style to be moved in the panel corresponding to its script and press ***Move***. If a style of such name already exists within the script where it is supposed to be moved, and has different attributes, it an additional confirmation request is issued for the replacement of the old style with the new one.

Style duplication in the script under another name

The ***Dup*** button allows to create a new style at the selected script. Pressing of this button opens the ***Enter new style name*** dialog box, used to input the name of a newly created style possessing the same attributes as the selected one. To edit the new style, use the ***Edit*** button.

Style removing from the script

The ***Delete*** button is used to remove the selected style from the current script.

Style copying at the text fragment inserting from the *Clipboard*

When the text fragments are copied from one script to another using the *Clipboard,* corresponding styles are not copied. To transfer the styles from one script to another, use the corresponding copy function of the ***Style Manager*** dialog box.

Chapter 5 Script Page Multilayer Structure

5.1. Multilayer Structure Creating and Editing

Background layer - *Background*

A new page is always created with a single layer in it, the *Background* layer. This layer can accommodate a graphic image with text information and logo characters above it.

Animation pages do not use a background. Top and left margins of these pages specify the position of animation file frames.

Graphic image in the background layer

A graphic image from a *BMP* or *TGA* file with an alpha channel, with a name specified by the ***Page/Background…*** command, can be put in the background. This image is rigidly anchored to the upper left corner of the page, and this method is convenient to use when a certain picture from a file (without changes) is used, rather than a series of various frames. Such file can be created at the program, for example, using the ***Page/Save As Bitmap…*** command. To discard the image setting for the background, use the ***Page/Clear Background*** command.

Background text and full-screen effects

If an effect is executed in the entire screen area (the ***Partial*** box in the ***Page Attributes*** toolbar is unchecked), the background text is positioned from the left to the right and from the top downwards, beginning at the upper left corner of page. For all effects except the “creeping line”, the right border is simultaneously the edge of the page and of the videocard buffer. There is no vertical limit, but for all effects except the vertical scrolling, an image cut-off occurs at the bottom edge of the video buffer.

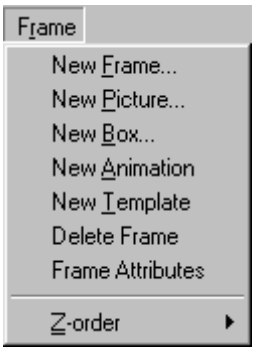
The “creeping line” effect has no horizontal restrictions: the page width is determined by its contents, both for background and frame text. The top edge of the background text will, however, be displayed in an editor window according to the level of the upper limit of line movement in the videocard buffer (***Y***). It should absolutely be considered at the arrangement of additional frames (they should lay at the same level) or creation of a background image – the meaningful part of a graphic file must have an equal top margin.

Movement of a text inside a background area can be achieved by inserting empty lines and changing the paragraph spacing. But it is more convenient to use text frames, since their arrangement in pages is not restricted in any way (see **5.2 Text Frame – *Frame***).

Background text and limited-area effects

If an effect is executed in a limited rectangular area (the ***Partial*** box in the ***Page Attributes*** toolbar is checked), the text position inside the background area is set by four parameters ***X***, ***Y***, ***W***, and ***H***, located in the ***Page Attributes*** toolbar. These parameters determine the effect scope. The left (***X***) and top (***Y***) offset values for the area upper left corner are measured in pixels from the beginning of the videocard buffer. The ***Y*** top margin defines the vertical shift of the text. The ***H*** height does not influence the arrangement of the background text on the page, but is used to define the lower edge of the effect scope. It is manifested, for example, for a page with vertical scrolling, when the text can be much longer than the effect area. Similarly, in the case of a creeping line in a limited area, the effect area is defined by the top margin and the height, and the text width is unlimited and can by far exceed the area size.

Creating of a new layer (frame)



An unlimited number of layers (frames) can be arranged on any page, except animation pages. Frames are created using the ***Frame*** menu, which contains a command for each frame type:

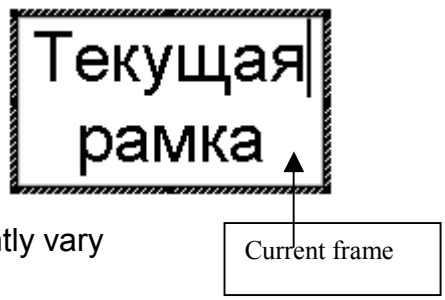
New Frame	for a text frame;
New Box	for a rectangular background area frame;
New Picture	for an image frame;
New Animation	for an animation frame.

The new frame is created immediately following the current one (or “above” the current layer). The ***Frame/Z-order…*** commands can be used for subsequent changes of a frame position in the hierarchy.

Current frame and its displaying in editor window

The frame (layer), which contains the text cursor, is considered current. If a position in the script is outside the background layer, it necessarily specifies a current frame. The text cursor symbol (in the ***Page Layout*** page mode) disappears if the current frame contains an image or a background area. Inside a text frame, the cursor is marked and specifies its position in the text. The number of the current frame on the page is displayed below, in the ***Status Bar***.

In a sequential mode of the script output in an editor window, the text cursor never disappears and always specifies a current frame, independently on its type. In a page mode the current frame is highlighted by a rectangle with eight squares on its edges. The type of the outline of this frame can slightly vary from one case to another.



Selection of the current frame

A current frame can be selected by moving the text cursor sequentially through the script using the keyboard buttons (see **3.6 Text Cursor and Current Position Display**). A faster and more convenient way is to click the mouse left button, when the pointer is located on the required frame. In the page output mode, when frames overlay each other, frame selection may present certain difficulties. In this case, if the mouse pointer is in a frame intersection region, the uppermost frame is selected. Therefore, it is important to construct the frame hierarchy of the page correctly, so that the frames would not screen the access to each other.

Editing the current frame parameters

The dialog box with the current frame parameters can be opened using the ***Frame*** menu. This menu contains several items corresponding to different frame types: ***Frame Attributes***, ***Box Attributes***, ***Picture Attributes***, or ***Animation Attributes***. The parameter editing window can also be opened by the mouse left button double-click when the pointer is located on the required frame. Another way to open the dialog box with the frame parameters is to click the mouse right button on the frame and to select the necessary subitem from the appearing ***Attributes…*** menu. In a special case of the text frame, the mouse pointer must be located on the frame outline; otherwise a text fragment is selected, or the ***Paragraph Attributes*** window appears, respectively.

Changing of frame position: *X, Y*

In the page output mode, the position of any frame can conveniently be changed using the mouse. To do this, put the mouse pointer on the frame outline, click the left button and, holding it pressed (without letting off), move it to a required place. After the left button click, the cursor turns into an arrowed cross. For an image or background area frame, there is no need to position the mouse pointer on the frame outline; it can be located anywhere inside the frame.

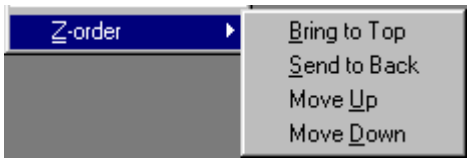
Another way to change a frame position with a higher precision is to use the frame parameter dialog box, which contains the ***X*** and ***Y*** fields. These fields specify the offset of the frame upper left corner in pixels with respect to the page beginning.

Frame resizing: *Width, Height*

To change the frame dimensions in pixels, use the ***Width*** and ***Height*** fields in the frame attribute dialog box, or use the mouse. The latter requires to position the mouse pointer allocates on an appropriate rectangle of the frame outline, to click the left button and, holding it pressed (without letting off), to move the mouse to resize the frame. After the left button click, the cursor turns into a double arrow with an appropriate direction.

The size of animation frames is defined by the size of the *AVI* file containing the frames (or image sequences); it is predefined, and cannot be changed.

Changing the current frame position in the hierarchy



Changing of a position of the current frame in hierarchy allows to set their correct order, i.e. to define the order of the frame drawing.

The position changing is performed with the ***Frame/Z-order…*** commands. After the execution of these, the moved frame remains current.

The dialog box with the commands changing the frame position in the hierarchy is opened by clicking the mouse right button on a frame. Then the properties menu appears, where the appropriate subitem should be selected.

To pull the frame down to the lowermost layer located immediately above the background, use the ***Send to Back*** command. The ***Bring to Top*** command, on the contrary, lifts the frame to the uppermost level, so it is be drawn last, above all other frames. ***Move Up*** and ***Move Down*** commands swap two adjacent frames, lifting or moving down the current frame, respectively.

Deleting the current frame

To delete the current frame, use the ***Frame/Delete Frame*** command. Alternatively, the *Del* key can be used, if the current frame contains an image or a background area. A text frame can only be deleted, if it is empty.

The dialog box containing the frame deleting command can be opened by clicking the mouse right button on a frame and selecting the ***Delete…*** subitem from the appearing properties menu.

Frame Group Selection

Selection of several frames into a group

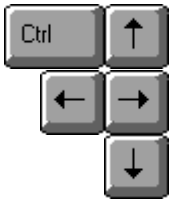
The program page mode allows to select one or several frames of a page as a group (these frames must not necessarily be in a sequence). To selection such group, press the *Ctrl* key and, holding it pressed, select the required frames by clicking the mouse left button on them. Text frames are to be clicked on the frame outlines, or on any character interior.

For a better selection accuracy, it is recommended to set the page displaying mode on the screen to the actual size: ***View/Zoom/100%***. If a group of intersected frames is selected, the order of their selection may sometimes be important, because otherwise, any of the already selected frames can overlap all others.

To select only one current frame, press *Gray 5*.

After a frame is selected in a group, its contour is outlined by thin lines with eight small squares on edges. Moving of the text cursor with the help of arrows keys is then unavailable, as these keys are used for other purposes.

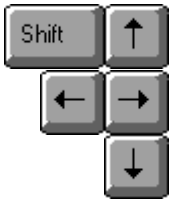
Moving of all frames of a group



An entire group can be moved within the page using the arrow keys. Each hit of a key moves all frames of the group in the corresponding direction by one pixel. If the *Ctrl* key is pressed and hold, each key hit moves the group by 16 pixels.

Moving is accompanied by the display of the new frame layout in the editor window in the page mode.

Alignment of all frames of a group



All frames of a group can be aligned to the boundaries of one of them. These commands are performed by the arrow keys with the *Shift* key pressed. In this case, the alignment is performed with respect to the last selected frame of the group. Thus, the “up” arrow draws all frames are drawn to the upper edge of the last frame of the group; the “down” arrow moves them to the to lower edge, the left and right arrows, to the left and right edges, respectively.

Another method of alignment can be used if several frames are selected simultaneously in a **text** fragment. **I**n this case, certain identical ***X*** and ***Y*** values can be set for all of them. For fields where new values have not been entered, they are assigned equal to the values of the first of the selected frames.

Copying a frame group to the *Clipboard*

A group of frames can be copied to *Clipboard* entirely by the methods described above in **4.2 *Clipboard* Operations (Copy and Paste)**. After a group of frames is inserted from the *Clipboard* into a script, they are positioned in succession, and their relative order is preserved. In particular, to copying one frame to the *Clipboard*, it must be selected into a group.

Discarding the frame group selection

To discard the frame group selection mode, press *Gray 5* or click the mouse left button on one of the frames, while holding the *Ctrl* key pressed – similarly to the selection procedure.

5.2. Text Frame – *Frame*

Text frame creation

The ***Frame/New Frame*** command creates a pre-determined text frame with one empty paragraph. The style of the paragraph, as well as its character style, is defined by the current values.

If several frames are created successively, they are not located in the same place, as the program shifts their initial point, preserving all other parameters.

Editing the text frame parameters

To edit the text frame parameters, use the ***Frame Attributes*** dialog box

X, ***Y*** and ***Width*** parameters (described above in **5.1 Multilayer Structure Creating and Editing**) correspond to the coordinates of the upper left corner and the border width in pixels. If several frames within a text fragment are selected simultaneously, these fields can have indefinite empty values. In this case, defined identical parameter values can be set for all these frames. If no new values have been entered for some of the fields, they are assigned according to those of the first of the selected frames.

The frame height is always defined automatically, based on the entered text size.

Clearing a text frame

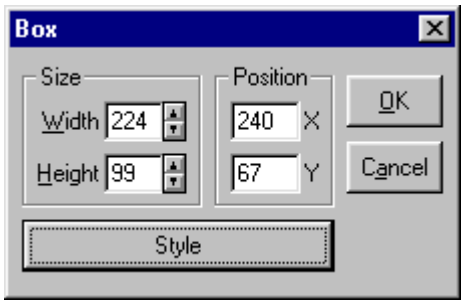
To clear the text information in a frame, select all of it in a fragment and press *Del.* In this case, the frame and its attributes are saved.

5.3. Rectangular Background Area – *Box* Frame

Box type background frame creation

The ***Frame/New Box*** command creates a frame with background area, opening the ***Box*** dialog box for this purpose.

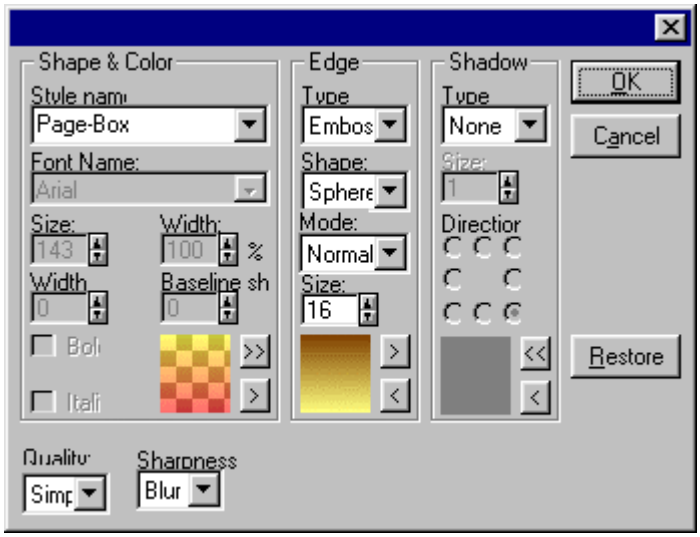
Editing the background area frame parameters



To edit the background area frame parameters, use the ***Box*** dialog box.

The ***X***, ***Y***, ***Width,*** and ***Height*** parameters (described above in **5.1 Multilayer Structure Creating and Editing**) correspond to the upper left corner coordinates, frame width and height in pixels.

Background area frame design style: *Style*



The background area has a rectangle contour, which can be designed by the same rules, as for normal text character. The ***Style*** button opens the dialog box used to define the design style of the background area. The set of parameters is almost identical to that of the ***Font Attributes*** toolbar, except those concerning the character typeface.

For details on background area frame drawing elements and filling methods see **4.6 Text Characters Attributes**.

Use of background area frames

Box frames can be conveniently used to design a scene background and text frame substrates.

Use graphics files for the frame internal area textural filling, allows to set an appropriate border for it. It should be noted, and, if necessary, used, that the graphic file image fills the frame areas according to the tile principle.

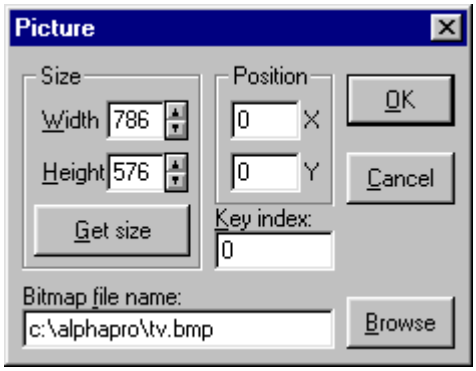
5.4. Image Frame – *Picture*

Image frame creation

The ***Frame/New Picture*** command creates an image frame, opening the ***Picture*** dialog box for this purpose.

Editing the image frame parameters

To edit the image frame parameters, use the ***Picture*** dialog box.



The ***X***, ***Y***, ***Width,*** and ***Height*** parameters (described above in **5.1 Multilayer Structure Creating and Editing**) correspond to the upper left corner coordinates, frame width and height in pixels.

Types of used graphics files and their transparency

The name of the graphics file is entered in the ***Bitmap file name*** field. To facilitate the search and selecting of a file for the frame, the ***Browse*** button is provided.

The following graphics file types are supported:

- *Windows BMP 8, 24 bit* (without compression); `
- *Targa 24, 32 bit* (with or without *RLE*-format compression).

The transparency for 256-color files is set by the key color index – the ***Key index*** field. If the key color index is not specified, the logo is considered opaque. *True Color* file transparency can be set using the *Targa 32 bit* format with an alpha channel. *Windows BMP* and *Targa 24 bit* files formats are considered completely opaque.

Map scaling

The initial image stored in a graphic file may not have the required dimensions in pixels. The program allows to correct this by setting the necessary values of ***Width*** and ***Height*** fields. If the initial sizes are to be restored, use the ***Get size*** button provided for this purpose.

It is generally better, however, to use an image of the required size: professional graphic editors (like, for example, the *Adobe Photoshop*), usually provide a better-quality rescaling. The program features allow first to select the image dimensions, and then to insert it permanently without any further scaling.

Suppressing the *flicker*-noise and image "jags"

No *flicker*-noise suppressing filters are provided for images. Therefore it is necessary to make an analogous filtering while preparing the image with other graphics packages (like, for example, the *Adobe Photoshop*), by "softening" it with a *Blur* filter. To suppress the jitter and "jags" at the edges of the image with a passing video signal, execute an appropriate filtering.

Use of image frames

Image frames can be conveniently used to design the scene background. Their number is unlimited, and any collage can easily be created directly in the program. In general, the use of image frames along with frames of other types provides a multilayer text-and-graphics editing, with a full alpha channel support.

Differences between image frames and logo characters

The main difference between the image frames and logo text characters is that frames can be intersected, freely moved within a page and with respect to each other, and their number is unlimited. They are not adhered to any position in the text and do not move along with it.

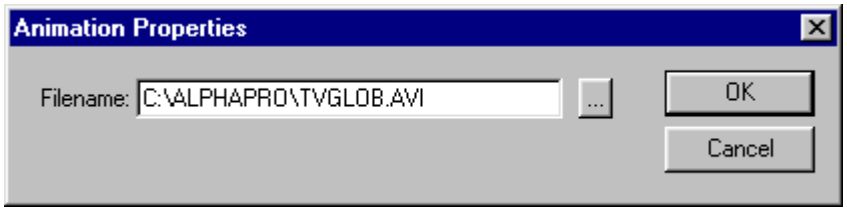
5.5. Animation frame – *Animation*

Animation frame creation

The ***Frame/New Animation*** command creates an animation frame, opening the ***Animation Properties*** dialog box for this purpose. An *Animation* type frame can be connected to a sequence of *TGA* files, or an arbitrary *AVI* file, provided the appropriate *CODEC* is installed in the system.

Frames of this type are used in pages with full screen effects such as "creeping line", or "drum", and provide a synchronous animation replay in one or several such frames in the "cycled" mode, during the text movement. In this case, animation frames should be located directly atop of the output text, which must have appropriate empty spaces under them.

Editing the animation frame parameters



To edit the animation frame parameters, use the ***Animation Properties*** dialog box.

Types of used files and their transparency

The name of the *AVI* file, or of the first file of a sequence, is entered in the ***Filename*** field. To facilitate the search and selecting of a file for the frame, the ***...*** button is provided.

The following file types are supported:

- All *AVI* files (provided the appropriate *CODEC* is installed);
- *Targa 24, 32 bit* sequences (with or without *RLE*-format compression).

The transparency of *AVI* files can be set at the file creation with a non-linear editing program using the special format, also preserving the alpha channel. Such programs normally allow to generate a sequence of in the *Targa 32 bit* format files with alpha channel instead of one *AVI* file. *Targa 24 bit* format files, as well as usual *AVI* files, are considered completely opaque.

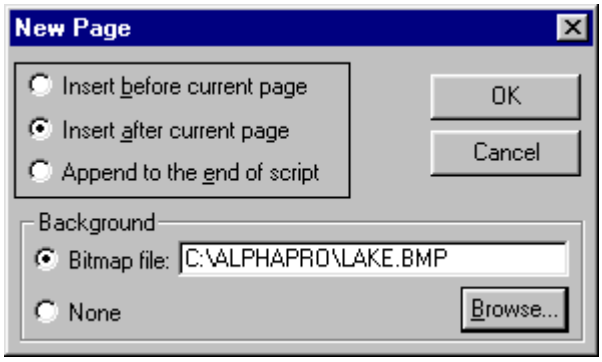
Chapter 6 Script Pages and Effects

6.1. Script Page Creation and Removal

There are several ways to create a new script page.

New page creation with the *Page/New...* command

The 1st way is to use the ***Page/New...*** command to open the ***New Page*** dialog box and create a new page with the default output effect. The page can be created in the end of the script, before or after the current page.



The current position moves to the beginning of the only (empty) paragraph of the newly created page. Attributes of this paragraph and the current character correspond to those in use at the moment of the command execution.

The ***New Page*** dialog box also allows to specify the name of a *BMP* or *TGA* format file to produce graphics for the page background, similarly to the use of the ***Page/Backround…***command.

New page creation by page splitting (*Page/Insert Page Break*)

The 2nd way can be used when the cursor is **inside a background text**. In this case, press *Ctrl+Enter* or use the ***Page/Insert Page Break*** command. The cursor and background text below it is transferred to the next, newly created page; the background text above the cursor and all frames remain at the previous page. Attributes of effects and text paragraphs, as well as the current character style in these split pages are preserved.

New page creation by copying to the *Clipboard* (*Page/Copy to Clipboard*) and pasting elsewhere in the script (*Edit/Paste*)

The 3rd way consists of copying of an entire existing page to the *Clipboard* and its subsequent pasting at another point of the script. To do so, use the ***Page/Copy To Clipboard*** command, which saves the page along with the corresponding effect attributes, and the ***Edit/Paste*** command (or *Shift+Ins*), which allows to paste the page from the *Clipboard* to any of the script currently in work just before the current page of this script.

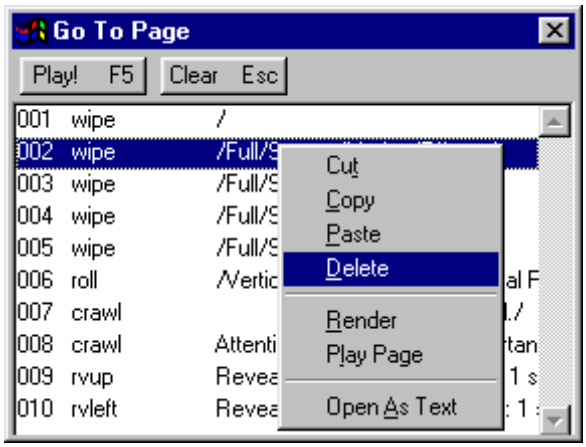
Hint. It may be helpful to create some dummy basic fragments of titling scripts, which can later be pasted into other scripts by page copying for final editing.

Script page removal (*Page/Delete*)

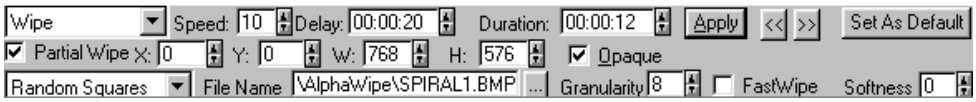
To remove a page from the script, use the ***Page/Delete*** command. The previous page then becomes current. If the removed page was the last in the script, the current position moves to the previous one.

Page copying and removal using the *Go To Page* **window**

Hint. *Clipboard* copying and removal operations described above can easily be performed on multiple pages using the ***Page/Go To…*** window. In this mode, necessary functions are called by a mouse right button click.



6.2. The *Page Attributes* Toolbar



Page effect attributes are conveniently edited using the ***Page Attributes*** toolbar. Use the ***View/Page Attributes...*** option or the *F7* keyboard button to toggle the toolbar display in the program window.

Setting of default attributes for new pages (*Set As Default*)

New pages created with the ***Page/New*** command acquire default attributes. To specify these, use the ***Set As Default*** button in the ***Page Attributes*** toolbar. These attributes are also used when a new script is created, or a text file loaded.

Setting of page effect attributes (*Apply*)

Attention! Page attributes are only changed after the ***Apply*** button is pressed in the ***Page Attributes*** toolbar. Failure to do so restores the page old attributes when the focus moves away from the toolbar.

6.3. General Settings of Page Effects

Page graphics substitution/ superposition over the video buffer old contents (*Opaque*)

The ***Opaque*** checkbox status is essential for all effects. If it is unchecked, the page elements are superimposed over the video buffer previous contents as the effect is executed. If the ***Opaque*** box is checked, the effect replaces the video buffer previous contents with the page graphic elements within the effect area. This also applies to the display of the effect area contents in editor window.

Effect completeness option (*Complete*)

The completeness option can be set for most effects using the ***Complete*** checkbox. If it is checked, the page disappears using the same methods and the same speed as were applied for is appearance, after the effect and the inner delay time (***Delay***) is over. In this case, the video buffer contents remain the same before and after the entire effect execution.

Effect inner delay (*Delay*)

An inner delay can be specified for most effects using the ***Delay*** field, to be executed after the effect is complete. If the ***Complete*** option is set, the delay takes place before the effect last stage – i.e., the page disappearance from the screen.

The delay time is set in *minutes: seconds: frames*.

How to specify an inter-effect delay?

The program does not provide special means to produce such a delay. It can, however, be construed by the use of a wipe effect with the ***Speed*** value exceeding 256, with a small scope and the ***Opaque*** option set off. The ***Delay*** value for this effect specifies the inter-effect delay time in *minutes: seconds: frames*.

Effect duration (*Duration*)

Effect execution time can be specified directly for all effects using the ***Duration*** field. The effect speed field (***Speed***) must be set a special empty value beforehand. Any value may be specified in the ***Duration*** field, not necessarily corresponding to an integer effect speed. The program takes care of an appropriate distribution of the effect time (including between the effect beginning and completion, if the ***Complete*** option is set); the effect speed continuously varies in the course of its execution, to fit the specified effect duration. These slight and repetitive speed changes can only be seen in movements. If, however, this is also unacceptable, change the ***Softness*** parameter only available for limited-area motion effects (with the ***Partial Scroll*** option set on).

Specifying effect duration using the effect speed (*Speed*)

Effect execution speed can be specified for all effects using the ***Speed*** parameter. The integer number specified in this field is used by the program to calculate, according to the effect type an parameters, the effect duration, which then appears in the ***Duration*** field using the minutes: seconds; frames format.

The ***Speed*** parameter specifies for motion effects the value in pixels of the page text displacement during one field change. In the case of screening (*Wipe*), appearance (*Fade In*) and disappearance (*Fade Out*) effects, the effect duration is measured in units corresponding to field changes in the passing video signal. The number of such units is calculated as 256/***Speed***. Animation speed cannot be changed: the output of each new animation frame begins at the moment of the field change. Hence, the effect duration depends on the number of frames in the animation file and the number of repetitions during the animation output, specified by the ***Repeat*** parameter.

The ***Complete*** option affects the effect total duration. If this option is set on, the effect acquires a final stage, and the time for its execution is added to the effect proper duration to be displayed in the ***Duration*** field.

The delay time specified by the ***Delay*** parameter is not taken into account in the ***Duration*** field. The total effect execution time is a sum of the values of these fields.

6.3.1. Page Effect Scope

An effect scope can usually be specified as a rectangular area for any effect. The disappearance effect (*Fade Out*) is the only exception, as it is always executed in the entire area of the video card buffer. The area of an animation effect is determined by the dimensions a position of the animation frame on the script page. For other types of effects, the scope can be specified explicitly. Unless it is done, the effect is considered a full-screen one and is executed in the entire video card screen (buffer).

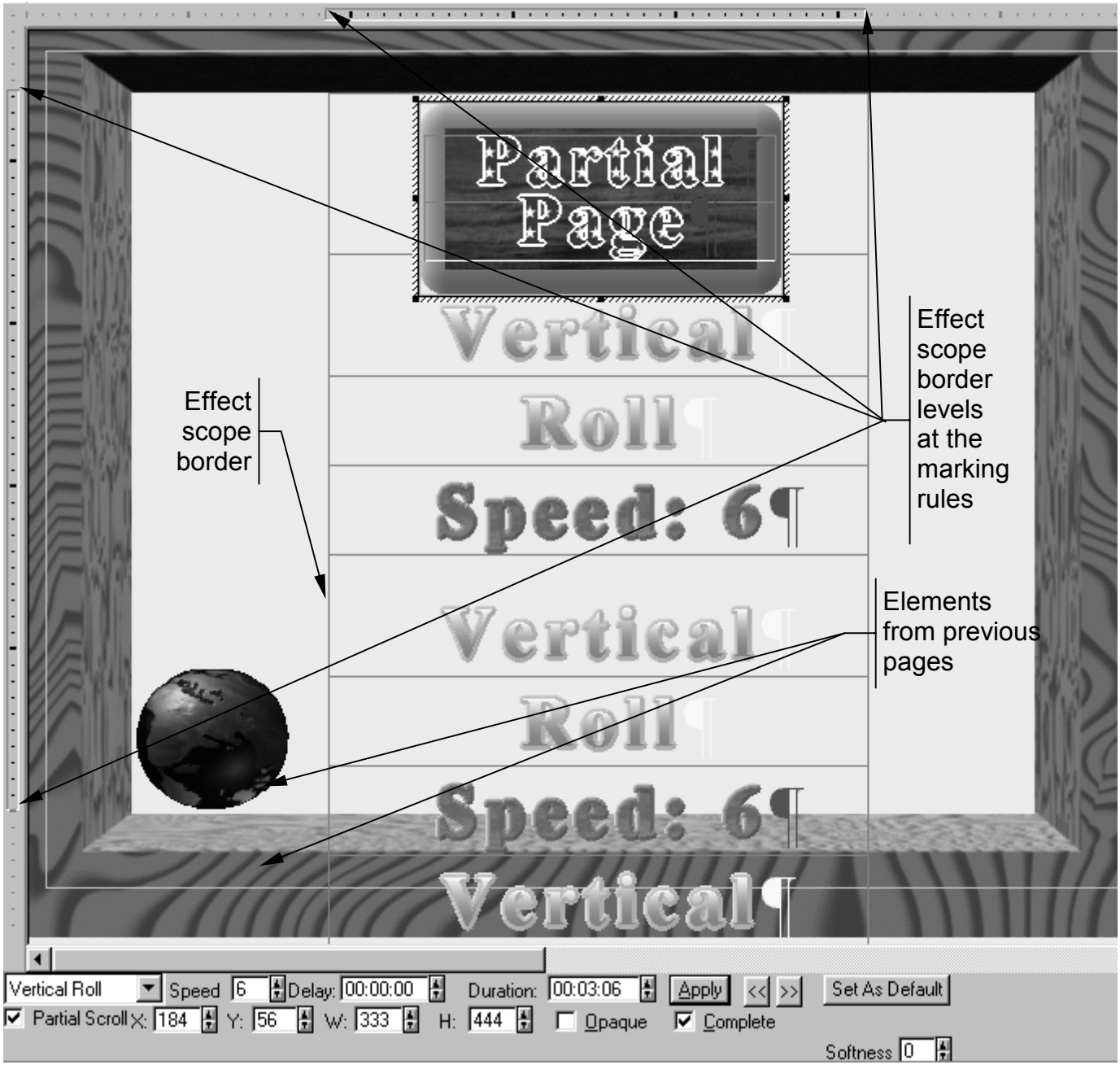
Full-screen effects and those executed in a limited rectangular area (*Partial…*)

If the ***Partial…*** checkbox is unchecked, the effect scope is supposed to include the entire video buffer. In this case, the background text position (except in the “creeping line” case) can be chosen without any additional restrictions. To position a “creeping line” (or several lines above each other) in the background, the top margin parameter (***Y***) must be specified, although the effect scope also includes include the entire video buffer.

Setting of effect scope limits and dimensions (*X,Y,W,H*)

If the ***Partial…*** checkbox is checked, the effect is considered to be executed within a rectangular area. Its border offsets are specified in the ***X, Y*** fields; width and height are determined by the ***W, H*** values, respectively. All these parameters are adjusted at the ***Page Attributes*** toolbar. The background text area in this case exactly coincides with the effect area, and the text in this area can only be moved using paragraph indentions and insertion of empty lines.

Hint. To avoid these restrictions of the effect area, it is more convenient to put the text in a *Frame* -type frame.



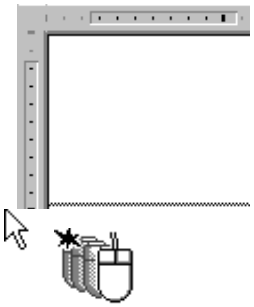
Editor window contents within and without the effect area

If the page output mode is set in the editor window, the video buffer status before the effect execution is displayed beyond the scope of the page effect.

Within the effect area in the editor window, the program displays the background text. If the “creeping line” or vertical scroll effect is used, the part of this text, which does not fit into the area, is also displayed in the editor window. It is located outside the effect area in a corresponding direction, atop of the video buffer previous contents.

The display of graphic elements taken from preceding pages in the editor window is affected by the status of the ***Opaque*** and ***Complete*** checkboxes. If the ***Opaque*** option is set on, the video buffer earlier contents is completely “wiped off” within the effect area; the elements of the current page only remain. If the ***Complete*** option is set on, the effect area is cleaned when subsequent pages are edited; otherwise it is filled.

Effect scope borders adjustment using the mouse



The above display principles provide a convenient way to adjust the effect scope borders in the page output mode using the marking rules. Position the mouse pointer on the edge of an “extruded” area of a rule, press the left button and, without letting it off, move the pointer. When the mouse pointer gets on the slider, it turns into an arrow with a support. While the button is pressed, a line crossing the editor window is additionally displayed and moved, which allows to see the effect border location with respect to other elements of the current and preceding pages.

How to specify effect scope borders close to characters and background areas

Attention! Border smoothing (as specified by the ***Sharpness*** parameter in the ***Font Attributes*** toolbar) of characters and *Box* background areas enlarge their dimensions by 1 pixel in all directions. It must be considered when specifying effect scope borders for the current and subsequent pages. The effect area should either include these additional pixels, or, to the contrary, not touch them, not to interfere with the smoothing. A similar smoothing takes place when pages are rendered for titles superposition using a mixer (the ***Video/Preferences/Rendering/Keying on mixer*** option).

This means that effect scope borders should not be set exactly at the borders of characters or *Box* areas: they should always be placed at least one pixel further to the outside.

Cutoff of page elements outside the effect scope during the rendering

Attention! Background text and graphics located outside the effect scope, as well as frame parts not included in the area, are cut off during the rendering, although they are fully displayed in the editor window. This should be carefully considered, although sometimes this fact may be taken advantage of (see **6.5 Combined Effects**).

In the special cases of the “creeping line” and vertical scroll effects, no cutoff is executed on the right and bottom effect scope borders, respectively.

Different presentations of elements of the current and preceding script pages

Hint. Complicated scripts sometimes present difficulties in distinguishing graphic elements of the current page from their predecessors from preceding pages, which are kept in the video buffer and are also displayed in the editor window. They can be told from each other by the fact that service marking line are only displayed for the current page elements.

6.4. Effects and Their Internal Parameters

Effect selection for a script page

Each page is assigned a proper effect, which defines the way it appears in the video buffer and vanishes from it. The effect type is specified using the drop-down list located in the upper left corner of the ***Page Attributes*** toolbar:

- *None* no effect (comment page);
- *Horizontal Crawl* “creeping line”;
- *Vertical Roll* vertical scrolling;
- *Reveal Up* upward ingress;
- *Reveal Left* leftward ingress;
- *Fade In* appearance against a background of the video buffer contents;
- *Fade Out* merging in the passing video signal;
- *Wipe* screening;
- *Animation* animation.

Effect types

The following effect types can be distinguished:

- empty effect – *None*;
- motion effects – *Horizontal Crawl, Vertical Roll, Reveal Up, Reveal Left*;

- appearance/disappearance effects – *Fade Out, Fade In*;
- screening effect – *Wipe*;
- animation effect – *Animation*.

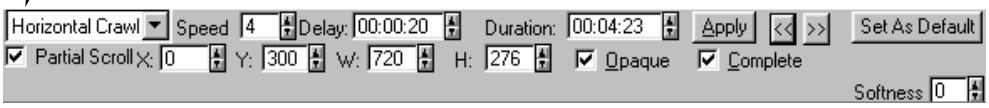
6.4.1. Empty Effect

A script can contain pages without any effects; the special -none- value is provided for this purpose. These pages do not affect the titles output; they can be used for comments, storage of dummies, or samples of character typeface styles. Comments can also be entered on pages making part of the titles output if they are located outside the effect scopes.

6.4.2. Page Motion Effects

Motion effects include:

- *Horizontal Crawl* "creeping line";
- *Vertical Roll* vertical scrolling;
- *Reveal Up* upward ingress;
- *Reveal Left* leftward ingress.



Additional parameters for limited-area motion

The following parameters can be specified for all motion effects:

- effect speed – ***Speed***, or duration – ***Duration***,
- delay time after the effect initial stage (ingress) – ***Delay***,
- completeness option – ***Complete***.

If the option of the effect execution in a limited rectangular area (***Partial Scroll***) is set, the following additional parameters can be specified:

- for the rectangular motion area, the upper left corner position (***X, Y***), width (***W***), and height (***H***);
- whether the page graphics is to be displayed atop of the video buffer old contents, or to expulse it in a specified direction (the ***Opaque*** option);
- степень сглаживания неравномерности движения – ***Softness*** the degree of motion irregularity smoothing (***Softness***).

Full-screen motion effects

All full-screen motion effects are executed in the entire video buffer area in the ***Opaque*** mode. Execution of such effect moves (and

expulses) all previous contents of the video buffer in a specified direction.

Hint. With some video cards, this may result in the program's inability to refresh the entire video buffer contents during the output of one frame.

When possible, try to use limited-area motion effects, specifying the minimally required motion area.

Execution of motion effects in the *Complete* mode

If the ***Complete*** box is unchecked, an effect consists of two stages – ingress and pause. After the motion is over, the page contents remain in the effect area and does not disappear (for "creeping line" and vertical scrolling effects, the rightmost or lowermost characters, respectively, and page frames remain on the screen).

If the ***Complete*** box is checked, the third stage is added: after the pause, an empty rectangle of the same size as the effect area "enters" the page and "expulses" its contents in the specified direction.

Motion irregularity smoothing (*Softness*)

If a motion effect is executed in a limited rectangular area, and its duration is specified manually, page motion may develop slight irregularities. The program provides a function to overcome this defect by calculating and creating graphic element images in the course of motion. Dots of these images are effectively located between the video buffer pattern dots. Number of these additional dots depends on the ***Softness*** parameter, which acquires integer values from 1 to 16. The 0 value means that no smoothing is carried out, and no "intermediate" dots are introduced.

Smoothing requires additional processor resources; therefore a minimal possible value should be assigned to the ***Softness*** parameter. Usually, the value 2 is sufficient.

Use of animation frames on pages with the "creeping line" or vertical text scrolling

Various frames can be used with the "creeping line" or vertical text scrolling. The text itself can be placed into a *Frame* type frame, and *Box* frames can be used to underline the text or to create a substrate.

Besides, up to **four** Animation frames can be positioned in the text empty spaces, between the words. The animation in the frames is then played back during the entire text output, in an auto-repetitive mode. The empty spaces are required, since when animation frames are displayed, they completely replace he corresponding part of the page graphical image with their contents, rather than be put atop of it.

Note that, if a non-zero ***Delay*** internal pause is specified for the page, the animation playback is stopped during this pause and after it (the last frame remains displayed).

"Creeping Line" (Horizontal Crawl)

The “creeping line” effect moves the page contents from right to left. Each paragraph of the background text is considered an individual line and is not formatted. It allows to output several lines located above each other simultaneously, which can be used, for instance, for a multilanguage translation.

The total length of the page image, which is moved at the effect execution, is defined by the maximum line length of the background text and the distance to the frame right edge.

Rendering and output of a full-screen "creeping line"

For a full-screen “creeping line”, the top margin value is specified in the ***Y*** field, defining the background text position. This margin also denotes the upper display limit of the page graphic elements: upon reaching this border they are truncated from above.

The lower border of the “creeping line” coincides with the video buffer lower edge. Hence, if the total height of the “creeping lines” and frames exceeds the video buffer size, they are truncated from below.

The above only concerns the contents of the page; in any case, the video buffer with the “creeping line” is moved as a whole.

Vertical Scrolling (*Vertical Roll*)

The vertical scrolling effect makes a page move upwards. The total height of the page image, which is moved at the effect execution, is defined by the height of the background text or the distance to the frame lower edges, whichever is larger.

Upward (*Reveal Up*) and Leftward (*Reveal Left*) Ingress

Upward (*Reveal Up*) and leftward (*Reveal Left*) ingress effects make a page move in the corresponding direction. The page image located in the effect area moves with a specified speed and stops for the period corresponding to the ***Delay*** value. Whether the motion in the same direction and with the same speed is afterwards resumed or not, depends on the status of the ***Complete*** checkbox.

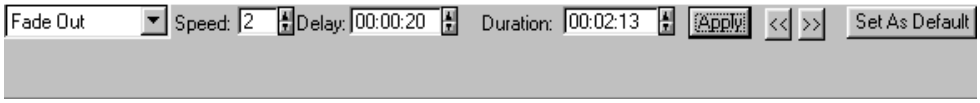
Ingress of an empty page van be used for “expulsion” of the entire video buffer contents (or some part thereof) in certain direction.

6.4.3. Appearance/Disappearance Effects

This type of effects include:

- *Fade Out* merging in the passing video signal;
- *Fade In* appearance.

Full-Screen Disappearance (*Fade Out*)



The disappearance effect deals with the entire video buffer contents. The principle of this effect is a gradual (with a specified speed) increase of the video buffer transparency with respect to the passing video signal.

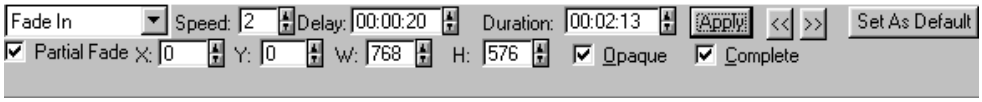
Attention! This effect can only be correctly executed if the videocard provides adequate hardware facilities. Otherwise, the buffer clearance is performed "instantly" (with the maximum available speed), while all delays produced by the effect duration and pause are preserved.

How to produce a disappearance in a limited area?

Hint. The program does not provide a disappearance effect for a limited rectangular area. To produce it, use the *Fade In* appearance effect (see below) with the ***Opaque*** mode set on. The corresponding page area must be empty.

.

Appearance (*Fade In*)



Full-screen appearance

The full-screen (with the ***Partial Fade*** checkbox unchecked) appearance effect causes a gradual reducing of the video buffer transparency with respect to the passing video signal, performed at the specified rate. If the output page contains completely transparent areas, they remain such before, in the course of, and after the effect execution.

Attention! Before the effect execution starts, the screen should be cleared of all graphics. Otherwise, prior to the effect execution, the video buffer becomes transparent for an instant, after which the appearance starts. It is done to copy the page image to the video buffer before the effect execution.

The *Fade In* effect can be combined with the *Fade Out* effect executed at the same rate after the ***Delay*** pause is over. To do so, check the ***Complete*** checkbox.

Attention! This effect can only be correctly executed if the videocard provides adequate hardware facilities. Otherwise, the output of the buffer contents is performed "instantly" (with the maximum available speed), while all delays produced by the effect duration and pause are preserved.

Limited-area appearance

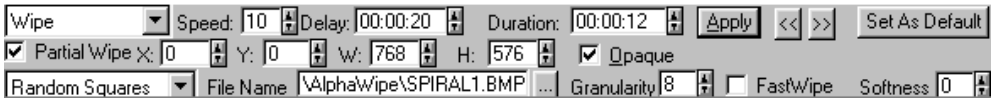
The appearance effect execution in a limited area (with the ***Partial Fade*** checkbox checked) involves a gradual replacement of the respective part of the video buffer with the page graphics, performed at a specified rate. If the ***Opaque*** option is set on, the effect area exactly corresponds to the page contents after the effect is over.

If the ***Opaque*** checkbox is unchecked, the page contents gradually show in the effect area **atop** of the graphics and the passing video signal, taking into account the transparency values.

The *Fade Out* effect can be combined with the *Fade In* effect executed at the same rate after the ***Delay*** pause is over. To do so, check the ***Complete*** checkbox. In this case, the video buffer (and screen) contents return to the initial state completely, after the effect execution is over.

Attention! An excessive effect area may cause the system to reboot. The maximum possible value of this area depends on the video card type and the computer capacity.

6.4.4. Screening (*Wipe*)



Screening effects are usually executed in a limited area (with the ***Partial Wipe*** option set on) with the borders defined by the ***X, Y, W,*** and ***H*** parameters in the ***Page Attributes*** toolbar. If the ***Partial Wipe*** checkbox is unchecked, the entire screen is considered the effect area.

The program offers a selection of standard screening methods (shutter types), to be specified using the drop-down list in the lower left part of the toolbar. An unlimited number of user-defined *Alpha Wipe* shutter types can also be used. The name of the

corresponding so-called “gradient” file is to be specified in the ***File Name*** field.

Substitution/ superposition screening mode (*Opaque*)

In the effect area, the page contents can be output in either substitution, or superposition mode. The substitution mode is entered when the ***Opaque*** checkbox is checked and supposes a removal of the video buffer old contents. In this case, the editor window is cleared within the effect area. In the superposition mode, new information is output atop of the older one, without removing it (as though the page is output in a new layer). The editor window looks within the effect area as if it contains an additional, remotest background plane remaining from the output of preceding pages.

Screening calculation (*FastWipe*)

If the time specified for the screening process execution is not sufficient (for example, with small ***Granularity*** values), the ***FastWipe*** option may be of certain help. It causes the screening shutter to be calculated beforehand. Since this operation requires additional RAM, it should not be excessively used.

This option is compulsory for “soft”, semitransparent-border shutters (with the ***Softness*** value greater than 0).

Semitransparent shutter border (*Softness*)

Any screening shutter can be output with “soft” semitransparent borders. To do so, check the ***Partial Wipe*** and ***FastWipe*** checkboxes, and specify a non-zero width of the semi-transparent border in the ***Softness*** field. If one of standard shutters is selected, set the ***Granularity*** parameter to *1*.

Standard Shutter Types

Элементы страницы внутри области действия эффекта выводится вместо (или поверх) содержимого видеобуфера одним из предопределенных способов, который задается в отдельном откидном списке:
Within the effect area, page elements are output instead (or atop) of the video buffer contents using one of the predefined methods, to be selected from a dedicated drop-down list:

- *<empty> - Cut* instantly and all at once;
- *Random Squares* random squares shutter;
- *Wipe Right* shutter moves from left to right;
- *Wipe Left* shutter moves from right to left;
- *Wipe Up* shutter moves up;
- *Wipe Down* shutter moves down;
- *Diagonal Wipe* shutter moves cornerwise;
- *Horizontal Jalousie* horizontally opening Venetian blinds;

- *Vertical Jalousie* vertically opening Venetian blinds;
- *Diagonal Jalousie* diagonally opening Venetian blinds;
- *Circles* widening circle.

Make sure that the ***File Name*** field specifying the "gradient" file name is empty!

Standard shutter granularity (*Granularity*)

The ***Granularity*** field determining the shutter simplification (roughening) degree specifies the shape of the shutter used at the screening. This specification may be required if the computer fails to perform the screening process in the specified time.

To "roughen" the shutter creation, increase the ***Granularity*** parameter value. The possibility of a timely update of the image depends on the size of the screening effect area and the shutter shape, and can only be determined empirically.

Instant output (empty value – *Cut*)

If an empty value is specified as a standard shutter type, the program performs an instant output of the page contents in the effect area – *Cut*.

Random squares shutter – *Random Squares*

A *Random Squares* shutter is output by randomly located squares. A higher granularity increases the size of these squares, which also decreases the minimum time of their output. Small squares, if rapidly output, make am impression of an explosion-like appearance of the new page in the video buffer. The *Random Squares* shutter can be used as a good substitute for the *Fade In* effect on video devices lacking the hardware facilities for appearance effects.

Up, down, rightwards, leftwards, and cornerwise moving shutters – *Wipe Right, Left, Up, Down; Diagonal Wipe*

Traditional shutters moving up, down, rightwards, leftwards, or cornerwise (*Wipe Right, Left, Up, Down,* or *Diagonal Wipe*) are produced by drawing of a corresponding rectangle or trapezium. A higher granularity increases the surface of such rectangle. The cornerwise shutter output is performed from the upper left corner towards the lower right one, at a 45° angle.

Attention! Horizontal shutters of big height should not be excessively used, as at their output, a characteristic prominence can be seen about the middle of the shutter. It is produced when the buffer contents cannot be completely updated in the time span of one field of the passing video signal.

Horizontal, vertical, and diagonal Venetian blinds shutters – *Horizontal, Vertical, Diagonal Jalousie*

Shutters output as horizontal, vertical, or diagonal Venetian blinds (or jalousies) is performed similar to traditional shutters of respective orientations. The effect area is divided into subareas; each of them widens in both directions, gradually revealing the video buffer new contents. A higher degree of granularity corresponds to a large number of these subareas. ***Granularity*** values close to *1* should not be used.

Diagonal Venetian blinds output output is performed from the upper left corner towards the lower right one, at a 45° angle.

Widening circle shutter – *Circle*

The widening circle shutter develops from the area centre towards its edges. Higher granularity yields a wider ring output at each step.

What if a standard screening method cannot be executed in time?

Hint. During each stage of a standard screening process, a portion of the current page contents is output. It may happen, however, that all updated information cannot be re-drawn in the time span of one field of the passing video signal. In this case, the shutter output takes more time than planned, and the effect execution looks slowed down. To let the update be performed in time, it is suggested to "roughen" the screening by raising the ***Granularity*** value. It decreases the number of screening stages and allows to perform transitions from one stage to another less often, i.e. not at each video signal field.

User-Defined *Alpha Wipe* shutters

The program allows to use any *Alpha Wipe* shutters defined in a so-called "gradient" file specified in the ***File Name*** field. The name of any *8-bit BMP* file may be used in this field. The program takes care of all necessary scaling, so that the image would fit the dimensions of the rectangular area of the screening effect.

In order to understand the principles of the screening process consider a file with a 256-grade gray palette, or *Grayscale.* Imagine that the area of the page graphic image containing the darkest (black) dots is the first to appear; appearing next are the slightly lighter dots, etc. The area containing purely white dots is the last to be output. Imagine also that the brightness of a dot denotes the depth of its position under a plane. The dot displaying process may then be compared to a lowering of the water level; at each stage water goes down to the next level of brightness. That is probably why such files used for *Alpha Wipe* shutters are called "gradient".

Semitransparent borders for *Alpha Wipe* **shutters**

Hint. Shutters look much better when output with "soft" semitransparent borders. To do so, check the ***Partial Wipe*** and ***FastWipe*** boxes, and specify a non-zero border width in pixels in

the ***Softness*** field. Note that the choice of the semitransparent area width should take into account the features of the shutter behaviour. A too large ***Softness*** value may change a shutter beyond recognition.

6.4.5. Animation

The use of animation in scripts allows to diversify titles output with any kind of graphics, including 3D effects, created using various software packages: *3D-Studio, 3D-MAX, Adobe Premiere, Adobe After Effects, Ulead Cool3D, etc.*

An animation effect is, in essence, an appearance/disappearance *Fade In/Fade Out* effect executed in a limited area and using a cycled playback of an animation file. That means that when appearance/disappearance effects are executed, the animation playback goes on, while the animation transparency with respect to the passing video signal or the video buffer preceding contents changes.

Hence, the ***Opaque*** and ***Complete*** options have the same meaning as above. Thus, if the ***Opaque*** box is checked, both static and animated graphics are displayed with a substitution of the video buffer old contents, and if it is unchecked, they are displayed atop. If the mode of the animation superposition is unimportant for the script, the ***Opaque*** option should better be set on, as this mode requires less computer resources.

The ***Fade*** parameter is equivalent to ***Speed***, except that when the value is 0, appearance effects are not executed at all.

A page can comprise, in addition to an animation, text and other (non-animation) frames. The animation then acts as a moving background, and all other elements are located in front of it.

Animation position on the screen and in the effect area

Attention! The ***X*** and ***Y*** fields determine the position of the animation upper left corner; the effect area is defined by the animation frame dimensions. Therefore, only the elements located in this area (in the editor window it is atop of the animation first frame output to the display screen) are output over the animation. Some measures can be taken to address these issues: for instance, it may be useful to create empty fields next to the frame borders beforehand. Frames cannot exceed the video buffer limits: this should also be taken into account in advance, so that corresponding fragments could be truncated when the animation is created.

Animation playback rate

Animation playback duration depends on the number of frames in it; a new frame is displayed every time the passing video signal fields change. That means that in 50 Hz TV signal output systems (*PAL, SECAM*), one second accommodates 50 animation frames, and in the NTSC system, 60 frames. A single playback can take more time if the size of transferred data is too large to let all frames to be redrawn in time.

Total duration of animation playback and number of repetitions (*Repeat*)

The number of animation repetitions is specified in the ***Repeat*** field. Total effect duration is then equal to the number of repetitions times the duration of a single animation playback.

This duration includes the appearance and disappearance stages. Therefore, if the ***Repeat*** value is insufficient, and the ***Fade*** speed is low, the effect may have not enough time for execution. To solve this problem, adjust either of these parameters.

Attention! If the number of repetitions exceeds 1, make sure that animation first and last frames "fit" each other but are different. Otherwise, a "hitch" can happen at the beginning of the repetition. Sometimes it happens because the generation program creates one frame more than required, when copying the first frame into the last one at the cycling.

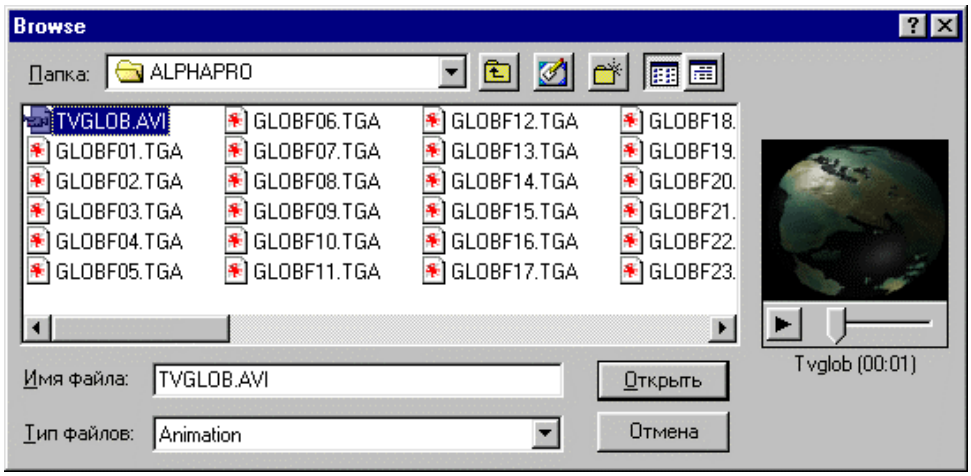
Supported types of animation files

The following file types are supported:

- *Video for Windows* (all *AVI*-files);
- *Targa 24* and *32 bit* (with or without *RLE* format compression) file sequences.

Animation file name input and file preview

The name of a graphic animation file is to be entered in the ***File Name*** field. To facilitate file search and selection, a ***...***button is provided opening the ***Browse*** dialog box.



A few seconds after the name of an *AVI* file is entered in the ***File Name*** field, a small preview window appears in the right-hand side of the dialog box. Press ▶ to play the animation back. If an *AVI* file

is selected, but no preview window appears, the file cannot be used as an animation source (e.g. because no appropriate *CODEC* is installed in the system).

Attention! Note that AVI files using YUV representation encodings cannot be used.

Transparency of animation files

Attention! The only way to adjust the transparency of an animation file is to use the alpha channel. Most *AVI* files have no alpha channel; therefore they are absolutely opaque. Some non-linear editing programs and 3D graphic editors (*Adobe Premiere, Adobe After Effects,* etc.) allow, however, to create 32-bit *AVI* files. The same programs usually allow creating 32-bit *TGA* file sequences as well, but it is generally much better to use a single file than a great number of separate frames.

A sequence of 24-bit *TGA* files provides an opaque animation. Transparency of a sequence of *Targa 32 bit* files is assigned according to the information from the alpha channel; therefore, its contents require a special attention. For instance, if all alpha channel values a zeroes, the animation is not seen on the screen at all.

6.5. Combined Effects

Combined effects as sequences of simple effects

Each of the effects described above is fairly simple; however, a mechanism of their multilayer consequential execution allows to create sufficiently complicated and interesting transitions.

Creation of combined effects with page copying

Creation of an effect sequence can conveniently be started with positioning all graphic elements eventually to appear in the same page. It is important to arrange all of them correctly to prevent subsequent re-arrangements to be required after the elements are copied to other, multiple pages.

After this “basic” page is laid out, it is useful to save it in a separate dummy script; then it is be copied as many times as required in the current script (using the ***Page/Copy To Clipboard*** and ***Edit/Paste*** commands). Then superfluous graphic elements are removed from each of resulting pages, and a proper effect is specified as a part of the sequence.

Use of the multilayer script representation in the editor window

The multilayer script pages representation in the editor window can be of a great help. It allows to control all stages of the effect sequence during the effect adjustment. It may be useful to change the transparency of certain scenes temporarily using the ***Opaque*** parameter. For a precise alignment of graphic elements and the effect area borders, it is recommended to use the actual view mode (***View/Zoom/100%***) for the editor window.

6.5.1. Examples of Combined Effects

Alternate flashing effect – *Flash*

The effect of alternate flashing of certain part of the image (*Flash*) is achieved by the use of a series of scenes with a *Random Square* shutter of low granularity. Scenes with and without the required element are alternating; the scene without the element is output in the substitution mode (with the ***Opaque*** option set on).

Script with a constantly present logo

The program allows to create a script with a logo (or any other information) constantly present on the screen, in parallel with the text output.

Begin the script with a scene, in which the logo appears. All subsequent effects are to be executed either in an area not including the logo location, or in the superposition mode (with the ***Opaque*** option set off). If a shutter (which is always executed in the superposition mode) is used, and something is to be output at the location of the logo, restore it by copying to the next page and assigning it some effect executed in an area including the logo.

The full-screen *Fade In* and *Fade Out* effects cannot, unfortunately, be used in such script: they execution would inevitably make the logo disappear for a while.

Line-by-line effects

Execution areas of some effect sequences can be positioned accordingly with the text lines layout on the page. To create such effects, first align the text in a *Frame* type frame and produce the required number of copies of the resulting page. If the text changes afterwards, new text has to be copied to all sequence pages along with the frame. This method ensures the conformity of text layout in different pages.

Text should not be placed to the background, as execution areas of the sequence effects are likely to be different, and the layout of different text parts can be hard to coordinate.

Examples of line-by-line effects

Various shutter combinations can be regarded as an example of line-by-line effects. Good horizontal and vertical panels can be produced if the motion is limited by the area of a single line. “Flying” lines can be created using upward or leftward motion, when the execution area for each next effect is exactly one (just produced) line smaller.

Character-by-character effects: "typewriter", "flying letters"

One-character effects can be used to imitate a typewriter, or to make a word be built letter by letter. Similarly to the case of line-by-line effects, output of each character requires a separate page to be created. Words should not be compressed using *Kerning*: it can hinder the division of the word into rectangular areas for the execution of character-by-character effects.

”Flying” characters can be produced using upward or leftward motion, when the execution area for each next effect is exactly one (just created) character smaller.

Execution area specification for single-line and single-character effects

To specify the effect area at each page, use corresponding sliders on the marking rules and thin marking lines between paragraph lines (visible if the ***View/Gridlines*** option is on). Effect areas should not be made too large, but defining their borders too closely to the boundaries of characters or background areas should also be avoided. This recommendation takes into account that smoothing of character and frame borders (the ***Sharpness*** parameter equal to *Normal* or *1*) use pixels adjacent to these boundaries. If such pixels are not included in the effect area, the smoothing is not effective.

Chapter 7 Script Output

7.1. Screen Output of Pages during Editing

Program two-screen operation mode

Script editing sometimes requires previewing a current page in its final state. The program is tailored to operate in a two-screen mode: computer display screen is then only used for script editing, while a dedicated TV monitor is provided for output. Contents of its screen stay unchanged until the script or some of its pages are output again.

Current page output using the *View/Take* command

To output the contents of the current page, use the ***View/Take*** command or press *F4*. The same command can also be executed by pressing the ***Take*** button in the ***Font Attributes*** toolbar.

This command outputs the entire contents of the current page to the TV monitor screen, looking exactly as at the actual real script playback. Some distinctive features are, however, to be named: 1) pages containing “creeping lines” are not output; 2) the entire current page contents are always output, regardless of the effect area position and dimensions; and 3) video buffer contents are not rendered by the moment of the output: a transparent background is always used.

Special features of vertically scrolling page preview on TV screen monitor during editing

When a page using the vertical scrolling effect is output to a TV monitor screen by the ***View/Take*** command, background text and corresponding frames are arranged in the normal order. Parts of the text (and frames) not included in the video buffer visible area are truncated.

Position of the background text being scrolled in a limited area corresponds to its position in the editor window (this also applies to all frames in the page). Parts of the text (and frames) not included in the video buffer are truncated.

Continuous display mode for page contents final state during editing (*View/Video Display*)

A continuous mode is provided for page contents display on the TV monitor screen, facilitating the final alignment of graphic elements with respect to each other. The output is performed in the same way as with the ***View/Take*** command, but is executed instantly, during pauses in editing. It is a more rapid and easy way to work, since there is no need to execute the ***Take*** command repetitively after each stage of editing.

Use the ***View/Video Display*** option to toggle the continuous display mode.

The main task of the continuous display mode is a real-time adjustment of the page graphic element parameters and their

respective layout. Any changes made in the editor window are reflected on the TV screen instantly. This applies to any text or frame displacements, changes of their features, and modifications of single character attributes.

Real-time adjustment of character parameters in a text fragment

To be able to adjust character attributes in real time, set the ***Auto Apply*** option in the ***Font Attributes*** toolbar on. Then all changes in character features in a selected text fragment are displayed both in the editor window and on the TV monitor screen without pressing the ***Apply*** button (if the ***View/Video Display*** option is activated).

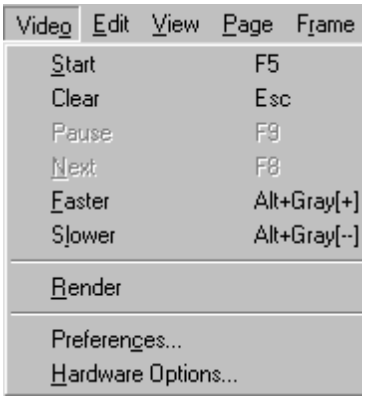
Continuous display mode requirements

Attention! Continuous display mode can only be activated in the page editing mode (***View/Page Layout***) with the actual size view (***View/Zoom/100%***). Continuous display is an option peculiar to a window, so changing the focus may pause or resume it.

The "viability" of the continuous display mode depends on the computer performance and available RAM resources to be allocated for cache buffer (***Video/Performance/Cache***).

7.2. Script Output Preparation – Rendering (*Video/Render*)

The ***Video*** menu deals with script rendering and playback. It includes both playback control commands (***Start***, ***Clear***, ***Pause***, ***Next,*** ***Faster,*** and ***Slower***) and the script rendering command (***Render***). Besides, the menu comprises the ***Preferences*** item used to modify program options and the ***Hardware Options*** submenu allowing to adjust video card settings.



A script must be rendered before it is played back; rendering creates binary images of the script pages, which are then transferred to the video buffer. Before it is done, the script is only connected to the video card via its video buffer size; if different video cards have equal buffers, the same script can be used with them.

Script rendering is executed by the ***Video/Render*** command. It may take a substantial time to be processed, and capture all of computer resources (especially RAM). So if another script is played back simultaneously, and its playback is not paused, image distortions and blinking may happen on the TV monitor screen.

If the script has not been rendered beforehand, rendering is executed automatically before the script is played back, after the ***Video/Start*** command is called. Script playback can start immediately after rendering is complete.

Rendering of edited pages and long-time storage of rendering results

A one-time rendering is sufficient for any script; rendering results are saved in a dedicated folder named *Cache*. Files stored in this folder are preserved at a program reload or computer reboot, so if later the script is loaded again without any further editing, it can be played back immediately. If, however, the script was edited (the *Modify* checkbox in the right-hand part of the status bar is checked), modified pages are only re-calculated. Removal of the *Cache* folder files (**only all folder files together can be deleted**) clears disk space but makes it necessary to render the corresponding script again before it can be played back.

Display of rendering progress in the *Status Bar*

Script rendering progress is reflected in the ***Status Bar***. An “extruded” rectangle appears in the left-hand part of the bar, next to the "*Rendering...*" message. The growing number of rendered pages is indicated by the widening black part of the progress bar.

Script rendering during live broadcasts

Hint. Script rendering features can be used in live broadcasting. To do so, open and render several scripts beforehand, corresponding to specific situations possible to arise during the broadcast. These scripts can further be called forth instantly, by focusing on their names in the program window and pressing the ▶ button in the ***Toolbar***; or by highlighting necessary pages in the ***Go To Page*** window and pressing ***Play!*** or *F5*.

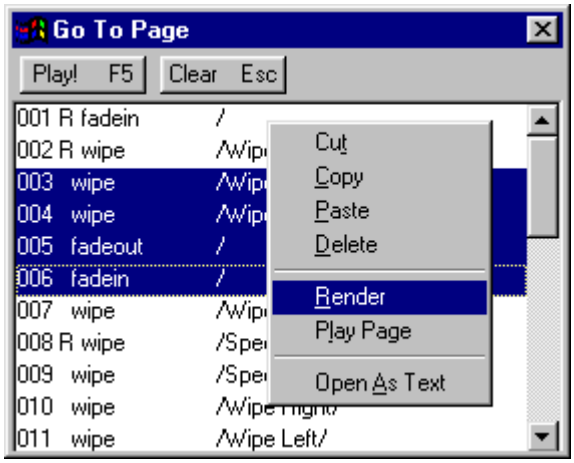
If some slight changes are to be made in the scripts, re-calculation of modified pages can be performed very quickly. Moreover, if the ***Start Through Pause*** option in the ***Video/Preferences/Player*** parameter group is not active, titles output from the ***Go To Page*** window and rendering of modified pages are performed **simultaneously**. This opportunity should, however, be used in moderation, so as not to disrupt the smoothness of titles output.

These features can be used for modifying information (for instance, people’s names, titles, phone numbers, etc.) during live broadcasts. In this case, opened scripts are used for templates. It is, however, more convenient to use the additional features of the extended version of the program to address these issues.

Use of *Windows* virtual memory for script rendering

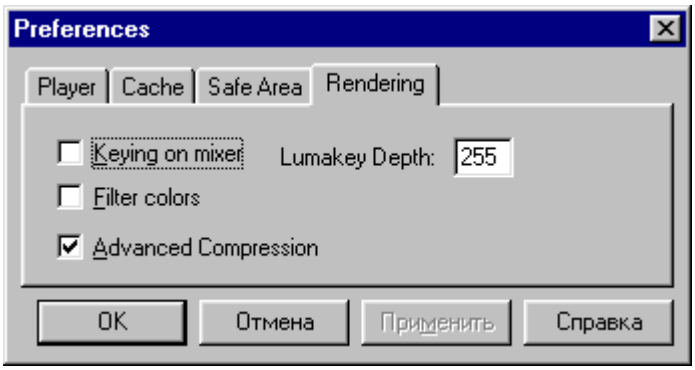
Attention! Use of virtual memory for program operation should be performed carefully. Page image rendering may require additional (at least virtual) memory resources; hence, *Windows* is to be able to create large *swap* files of unlimited size. Additional RAM resources would, however, provide a better solution, as the rendering speed can in this case be much higher.

7.2.1. Script Rendering in the *Go To Page* Window



The ***Page/Go To…***window allows seeing clearly, whether a page has already been rendered or not by the "*R*" (for "*Rendered*") character appearing in the list next to the page number if it is rendered. To render one or several selected pages, press the mouse right button with the pointer inside the window, and select ***Render*** in the appearing submenu. "*R*" characters appear in the list, next to the numbers of rendered pages, as the rendering process goes on.

7.2.2. Rendering Parameter Group (*Video/Preferences/Rendering*)



This group comprises options of the script graphic elements rendering for a subsequent video card output.

Parameters are saved between the program work sessions and affect all scripts opened later.

Graphics rendering for mixer superposition (*Keying on mixer*)

The ***Keying on mixer*** checkbox allows to activate the special rendering mode preparing graphics for mixing with the passing video signal using a mixer, which produces a computer graphics dots brightness key (*LumaKey).*

If this option is set on, all pixels at least partially transparent with respect to the passing video signal are proportionally mixed with the black color. Besides, all non-transparent areas are enlarged by black one pixel-wide framing. These methods provide a smooth brightness transition on the border between the computer graphics and the passing video signal, required for a high-quality mixer superposition.

Allowed range of graphics brightness for mixer superposition (*LumaKey Depth*)

A threshold value of computer graphics brightness must be specified for video mixer superposition in *LumaKey* mode (with the ***Keying on mixer*** option set active) in order to prevent the passing video signal appearance in non-transparent areas with minimum brightness of pixels.

The ***LumaKey Depth*** field specifies the variation range for computer graphics pixels brightness. All values within the ***LumaKey Depth*** from the absolute maximum (*255*, or pure white) are then allowed.

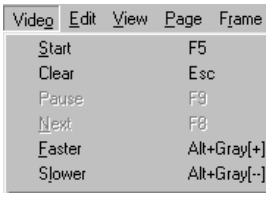
The value *255* means that the brightness of non-transparent pixels remains unchanged. The lower is the field value, the brighter is the page image on the screen. To compensate the rise of the computer graphics brightness, use corresponding mixer controls.

If the ***Keying on mixer*** checkbox is unchecked, the value in the ***LumaKey Depth*** field is ignored.

Additional compression of page binary images (*Advanced Compression*)

The ***Advanced Compression*** checkbox activates the compression mode for static (not moving) pages during their rendering. Pages using screening and appearance/disappearance effects belong to this category. This mode efficiently (up to 50%) saves RAM resources and disk space but requires higher processor performance. Therefore, if delays happen prior to the playback of static pages, this mode should be switched off.

7.3. Script Playback



Script playback controls can be found in the ***Video*** menu. These commands allow to:

- launch the playback: ***Start***,
- stop the playback and clear the screen: ***Clear***,
- pause/resume the script playback: ***Pause*** (***Resume***),
- output the next page: ***Next***,
- accelerate the output: ***Faster*** (),
- slow the output down: ***Slower*** (for motion effects).

Launching script playback (*Video/Start*)

Selecting the ***Start*** command or pressing *F5* launches the output of script titles to the video card, starting from the first page. Playback can also be started by pressing the ▶ button in the ***Toolbar***. To terminate the output, use the ***Video/Clear*** command or press *Esc*.

The video buffer is cleared before output starts. To launch a script playback without any delays, all of its graphical elements should be calculated beforehand (using the ***Video/Render*** command). Unless it is done, rendering is executed immediately prior to the playback.

Cache buffers clearing and script loading to RAM before playback

To make as much RAM available as possible, cache buffers used for characters and page images are cleared before the playback. Therefore, after a playback, page navigation in the editor window or character input may be somewhat slowed. In the course of subsequent work, however, the re-drawing rate returns back to normal level seen before the playback.

A script playback is only launched after all page images (if possible) are loaded to RAM. Note that the program always loads them to the Random Access Memory, rather than a virtual memory organized somewhere on the hard disk.

Time spent for this upload depends on the script volume and the rate of file reading from the disk. To avoid considering this (unknown) value, start the script playback after a pause provided by the ***Start Through Pause*** option in the ***Video/Preferences/Player*** parameter group.

Playback stoppage and screen clearing (*Video/Clear*)

The ***Clear*** command stops titles output and clears the video buffer immediately. This command can also be called by pressing *Esc*. Besides, the script playback can be stopped by pressing the ■ button in the ***Toolbar***.

Playback interruption and resumption (*Video/Pause* and *Resume*)

F9

Use of the ***Pause*** command or pressing *F9* interrupts titles output; a tick appears to the left of the *Resume* item in the ***Video*** menu. If then the ***Resume*** command is applied, the script playback is resumed immediately.

Alternatively, the  button in the ***Toolbar*** may be used to interrupt the script playback.

Script playback interruption would normally happen instantly; however, if certain effects are executed at the moment, the program may wait for the effect execution to be completed or to reach a specific stage (for instance, the ***Delay*** internal pause, or a completion of an animation cycle, etc.) before pausing the playback.

Use of the pause before and during script playback

Hint. Script playback interruption can be used with the "creeping line" or vertical scrolling effects, when a concept pause is required in certain places. The interruption command is called automatically before the script playback if the ***Start Through Pause*** option in the ***Video/Preferences/Player*** parameter group is set active. This interruption is used to synchronize the launch of titles output with other events, as well as to load the script to the memory. In this case, playback is actually launched by the ***Video/Resume*** command.

Output of the next script page (*Video/Next*)

F8

Using the ***Next*** command or pressing *F8* provokes an output of one following script page. This command is available if the playback is interrupted by the ***Pause*** command.

A page-by-page output mode can be used when the exact values of delay intervals between scenes are unknown to the staff, and the script output is carried out in the "live" mode, in parallel with the broadcasting or editing.

Change of output rate for motion effects (*Video/Faster*)

Alt +

(*Video/Slower*)

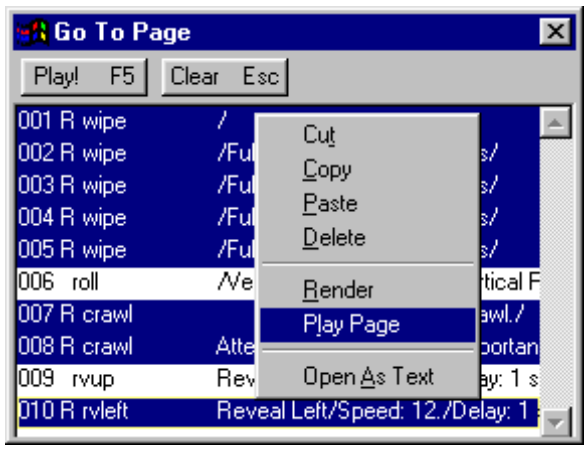
Alt -

Duration of all motion effects (*Horizontal Crawl, Vertical Roll, Reveal Left,* and *Reveal Up*) can be changed directly during their output; to fasten up or slow down the playback, use the ***Faster*** and ***Slower*** commands or the *Alt+Gray[+]* and *Alt+Gray[-]* keys, respectively. Such rate changes are usually rather slow and gradual, and speed values can cease to correspond to integer values of the ***Speed*** field in the ***Page Attributes*** toolbar. To keep the motion evenness and smoothness, set a non-zero value of the ***Softness*** field (*2* is usually sufficient).

Pages including motion affects are initially output at a rate corresponding to the required duration specified at the ***Page***

Attributes toolbar. This speed can further be raised or lowered manually, in order to fit into an unexpectedly truncated time span, or fill up a pause happening during a broadcast. Such unforeseen situations would often occur in live broadcasting.

7.3.1. Script Pages Playback Using the *Go To Page* window



The playback of a script, or a selection of its pages, can conveniently be performed using the ***Page/Go To…***window. This window lists all script pages, indicating the sequential number, effect type, text contents, and rendering status (if the page has been rendered, an *R* character is displayed after its number) for each of them. If a page, or a selection of (not necessarily sequential) pages is to be played back, it can be done by pressing ***Play!*** or *F5*. Alternatively, press the mouse right button while the pointer is inside the window and select the ***Play Page*** command in the appearing submenu. All ***Video/Preferences/Player*** options affect the page playback called from the ***Go To Page*** window except the ***Auto Repeat*** mode: in this case, playback is never looped, and each page is only output once.

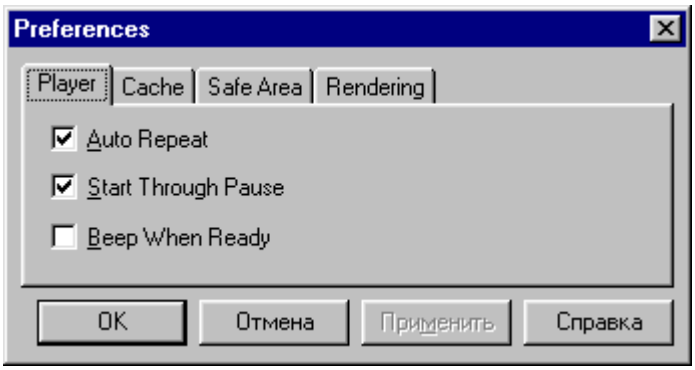
In addition to the playback command, this window also allows to use the video buffer clearing command – ***Clear*** or *Esc*.

Attention! It is important to bear in mind that if the script pages playback is launched using the ***Go To Page*** window, the video buffer is not automatically cleared before the output of the first page: its contents beyond the effect execution area are always preserved. And if the output is not performed in the substitution mode (the ***Opaque*** option is not set active), titles are output over the video buffer old contents in the effect area as well.

This program feature can be efficiently used in live broadcasts.

Script pages playback using the *Go To Page* window is executed over the video buffer old contents

7.3.2. Script Playback Settings (*Video/Preferences/Player*)



This group of options specifies the routine of the script output using the video device. The options are preserved between program sessions and affect the playback of all scripts opened by the program subsequently.

Script playback loop mode (*Auto Repeat*)

The ***Auto Repeat*** checkbox allows activating the “endless” script playback mode, outputting the first scene after the last one, etc. Such playback can only be interrupted manually, using the ***Video/Clear*** or ***Video/Pause*** command. If the playback of a script (or a selection of its pages) is launched using the ***Go To Page*** window, the status of the ***Auto Repeat*** option is ignored, and the script is only played back once.

Automatic interruption before the playback starts (*Start Through Pause*)

Checking the ***Start Through Pause*** box makes the program enter the pause mode after the script playback command is called, and the script is loaded to the memory. Pressing ***Video/Resume*** or *F9* causes an actual launch of the script playback in this case.

Setting of automatic interruption may be required for manual or GPI synchronization of the starting moment of titles output. In this mode, the script playback is always preceded by a maximal uploading of page graphic images to the **Random Access Memory** allowing performing the script playback smoothly, without calling the hard disk.

This pause can also be used for commutation of video devices and external equipment.

"Ready" beep (*Beep When Ready*)

If the ***Beep When Ready*** option is set active, an audible signal is issued when the script playback is paused (see above).

7.3.3. Why Can Scripts Be Played Back Unevenly?

Uneven, hopping script playback can usually be detected visually; in case of doubts it can be recorded on a videotape recorder and reviewed in a frame-by-frame or slow-motion mode.

Consider several of possible causes of uneven script playback, and ways to deal with them:

Hard disk calls or other program activity during script playback

If the titles output is accompanied by hard disk calls or active use of the processor by other programs, occasional pauses, “ruptures” or repetitions of graphic information blocks can happen.

Therefore, it is recommended not to launch (or, at least, not to use actively) other programs in parallel with the titles output. Firstly, such programs use RAM resources; secondly, working in multitasking mode reduces free time available for unpacking and output of information blocks to the video buffer.

Hard disk calls can be caused by a lack of RAM required to store images of script pages; then some of them are kept in virtual memory. To address this issue, consider building up the computer RAM. If the RAM cannot be increased, make sure that the ***Advanced Compression*** mode is used for page images (the corresponding checkbox in the ***Video/Preferences/Rendering*** parameter group is checked).

To save RAM resources, close unused applications, reduce the number of installed fonts, or cut down number of colors or resolution of the *Windows* desktop.

Too large area and/or speed of limited-area motion

Motion effects executed in limited rectangular areas can sometimes be performed intermittently. It usually happens due to a too large effect area, when the processor performance, or the processing speed of bus interface and card memory is not sufficient for transportation of such data volumes. In this case, if the limited-area motion speed is high, a distinct effect of output image “unsticking” can be observed close to the area edge, from which titles appear.

To solve this problem, try to reduce the motion area by cutting off its superfluous parts containing no information.

Too high speed of full-screen motion effects

If the speed of full-screen motion effects is too high, parts of information due to appear can constantly be observed at the edge of the screen, towards which the page moves – i.e., opposite to where information normally appears. Reducing the motion speed eliminates this effect. The maximally possible speed value depends on the information features and computer properties. For instance, enlarging height or width of the output page may require reducing the motion speed, and vice versa.

Odd speed values for full-screen horizontal motion

Full-screen horizontal motion effects are known to cause a uniform quiver of the image with the frame update frequency. This is caused by odd (or oddly even) values of the ***Speed*** parameter. Try to use even values for these speeds; if that is of no avail, use evenly even values. This limitation is a peculiar feature of specific video cards.

A fundamental way to deal with this issue is to replace full-screen effects with similar limited-area effects.

Insufficient operation speed of the computer for data unpacking

When additional page image compression is used at their output, much higher computer performance is required, since data unpacking is executed in real time. Therefore, if static pages (using screening and appearance/ disappearance effects) are output with a substantial delay, the ***Advanced Compression*** option in the ***Video/Preferences/Rendering*** parameter group should be set off. If setting off this option produces the desired effect, it means that the delay cause was determined correctly. It should be noted, however, that deactivation of this option may double the required memory size.

Too large area and/or speed of a screening or appearance effect

If the execution area of a screening (*Wipe*) or appearance (*Partial Fade In*) effect is too large, the effect execution may be completed substantially later than planned. This also may happen if the effect speed is too high.

To correct this situation, try to reduce the effect speed, raise the granularity, or set the ***FastWipe*** option on. If this is of no avail, the effect can be split in two or three smaller effects executed in sequence and covering the entire initially required area.

Too low shutter granularity in large effect area

If the granularity value (specified in the ***Granularity*** field at the ***Page Attributes*** toolbar) is too low, it can also cause delays or distortions of shutter output. If this is the case, try to raise the granularity value or curtail the effect execution area.

Uneven animation output

Animation effects require large amounts of various resources. Therefore, an animation output should not be accompanied by other actions.

Make sure that the superposition mode is set correctly (the ***Opaque*** box is unchecked): if this mode is not necessary it should not be used. Animations with large inter-frame differences should be avoided, since it is the maximum size of this difference, which is limited by the video card capacity.

Besides, animation should not be located in the upper part of the screen. If the video buffer sweep is faster than the output, a characteristic sawtooth prominence can appear; or the frame update can happen in an undulatory mode.

Use of *Windows* screensaver software facilities

Use of *Windows* screensaver software facilities may cause such system to become active during the titles output. Generally, use of additionally loaded *Windows* facilities should be avoided, since such software reduces the memory resources available to the titling software.

Use of timer for reckoning of intervals between video signal fields

If the program uses the system timer to reckon the time spans between the fields of the passing video signal, it toughens all restrictions applied to the titles output. The countdown must be performed using the interruptions generated by clock pulses of video signal frames. Contact the program authors for more related information.

Use of odd speed values for vertical motion

If an odd speed value is set for vertical motion, and the video card generates the signal using line interlacing (as it usually does), vertical resolution may be reduced by half. During the motion, only odd (or only even) text lines are output. To avoid this effect, use even values for vertical motion speed.

7.3.4. Script Playback Management Using *GPI* Interface

Effects controlled using the *GPI* interface can be executed during the titles output. In this mode, the script is played back by one page-scene, and the ***Start Through Pause*** option in the ***Video/Preferences/Player*** parameter group must be set active.

GPI interface control also allows a standard script playback. Pressing any of the control keys (*F9*, *F8,* or *Esc*) makes the program ignore further signals arriving from the interface.

The *use_gpi* parameter in the *PLAYER.INI* file

Switching the system to *GPI* interface-controlled operation requires, in addition to the manufacturing of an extension cable as described below, to edit the line

use_gpi=0

in the *PLAYER.INI* file with a text editor.

If *GPI* interface control is not required, a zero (default) value must follow the "=" sign. If the system is to be controlled with the ***Next*** and ***Pause*** commands, or the ***Next*** command only (see below), *0* is to be replaced with *1* or *2*, respectively.

Next and _Pause_ commands and distinctions of the start-stop mode (_use_gpi=1_) from simple start mode (_use_gpi=2_)

The ***Pause*** command, which interrupts any effect execution, to be resumed when the ***Next*** command is executed (or stopped entirely), is only available when *use_gpi=1*.

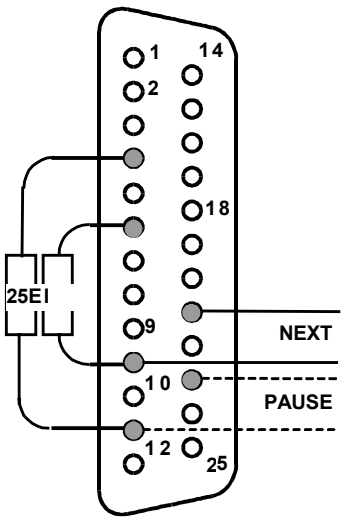
The ***Next*** command is multifunctional and is able to initialize several distinct operations: 1) to start the playback of the following page; 2) resume the final stage of an effect before the pause specified in the script is over; 3) resume the effect execution interrupted by the ***Pause*** command.

The ***Pause*** command can conveniently be used to interrupt long-time effects (e.g., “creeping line” or vertical text scrolling) if their output is to be paused for a while. All other tasks can be successfully performed by using the ***Next*** command alone.

Connection to the computer parallel port _LPT2_

Use of the interface requires a specific commutation of video device buttons and the contacts of the additional parallel port (*LPT2*) connector.

To call the ***Next*** command, ground the contact 10 – *nASKNLG* (i.e., connect it to one of the contacts numbered 18 to 25 – *VDC/GND*). To call the ***Pause*** command, ground the contact 12 – *CALL/PE*. Contacts 10 and 12 should also be connected to some of the contacts numbered 2 to 9 – *DATA x* – via ~25 kΩ resistors.



Keyboard emulation of the _Next_ command

If either of the *GPI* interface control modes is active, the ***Next*** command can be emulated using the *F8* key. Pressing this key then causes one next scene to be played back, after which the program enters the pause mode.

Chapter 8 Additional Issues

How to do without the change discard function (*Undo*)?

The program does not provide a possibility to discard the latest changes. It is therefore recommended to save scripts (using the ***File/Save*** or ***File/Save As*** commands) before any questionable changes are applied.

Use of special fonts for creating outlines, delimiters or icons

Hint. If a background area is to be created of a more complicated shape than a simple rectangle, use of symbol vector fonts can be suggested. These fonts include outlines of various shapes, with filled or empty internal areas.

Such background area can conveniently be moved if the symbol used for its creation is placed into a separate text frame. The area size and aspect ratio can be changed using the font-related ***Size*** and ***Width*** fields in the ***Font Attributes*** toolbar.

Special symbol fonts can also be used to insert special delimiters, icons, figures, etc., right into the text.

Program operation without a video card

Return to interruptions