

**ALGE DisplayStudio**

**Manual**

**ALGE**  
**TIMING**

## Table of Content

|        |  |    |
|--------|--|----|
| 1      | General .....  | 3  |
| 2      | Getting started.....                                 | 3  |
| 2.1    | Main Window .....                                    | 3  |
| 2.2    | Tree view for project navigation.....                | 4  |
| 3      | Lists.....   | 5  |
| 3.1    | List Builder .....                                   | 5  |
| 3.1.1  | Text panel .....                                     | 6  |
| 3.1.2  | Animation.....                                       | 6  |
| 3.1.3  | List compiling .....                                 | 7  |
| 4      | Animations and wipes .....                           | 7  |
| 4.1    | Adding animation/wipe to the project.....            | 8  |
| 4.2    | Animation/wipe editing.....                          | 8  |
| 5      | Backgrounds .....                                    | 8  |
| 6      | Fonts .....  | 9  |
| 6.1    | Adding fonts to the project.....                     | 9  |
| 6.2    | Font Editing .....                                   | 9  |
| 7      | Project settings.....                                | 10 |
| 7.1    | Display loading .....                                | 11 |
| 8      | List scheduler .....                                 | 12 |
| 9      | Remote control.....                                  | 13 |
| 10     | Display parameters.....                              | 14 |
| 11     | Active Panels.....                                   | 15 |
| 12     | Script file.....                                     | 17 |
| 13     | Auto Upload.....                                     | 18 |
| 14     | Script file example .....                            | 20 |
| 15     | Add new language.....                                | 22 |
| 16     | Excel to Display .....                               | 24 |
| 16.1   | Variables for Timing Devices and Game-Consoles ..... | 25 |
| 16.1.1 | Variables for Timers.....                            | 25 |
| 16.1.2 | Variables for Game-Consoles D-CKx .....              | 25 |
| 17     | Memory Map configuration .....                       | 27 |
| 18     | Tutorial.....  | 28 |

**Copyright by:**

ALGE-TIMING GmbH & Co  
Rotkreuzstrasse 39  
A-6890 Lustenau  
Telefon: +43 5577-85966  
Fax: +43 5577-85966-4  
e-Mail: [office@alge-timing.com](mailto:office@alge-timing.com)  
Internet: <http://www.alge-timing.com>  
Version: 070530

**Änderungen im Sinne des technischen Fortschritts vorbehalten!**

## 1 General


With the DisplayStudio Software package you can program your Display with Text and Graphic information. You can also upload Competitor lists for later usage together with any of our Timing-Devices. You can use existing multimedia, or you can create your own new multimedia data for display. With the preview function you can check your display content before uploading to the display.

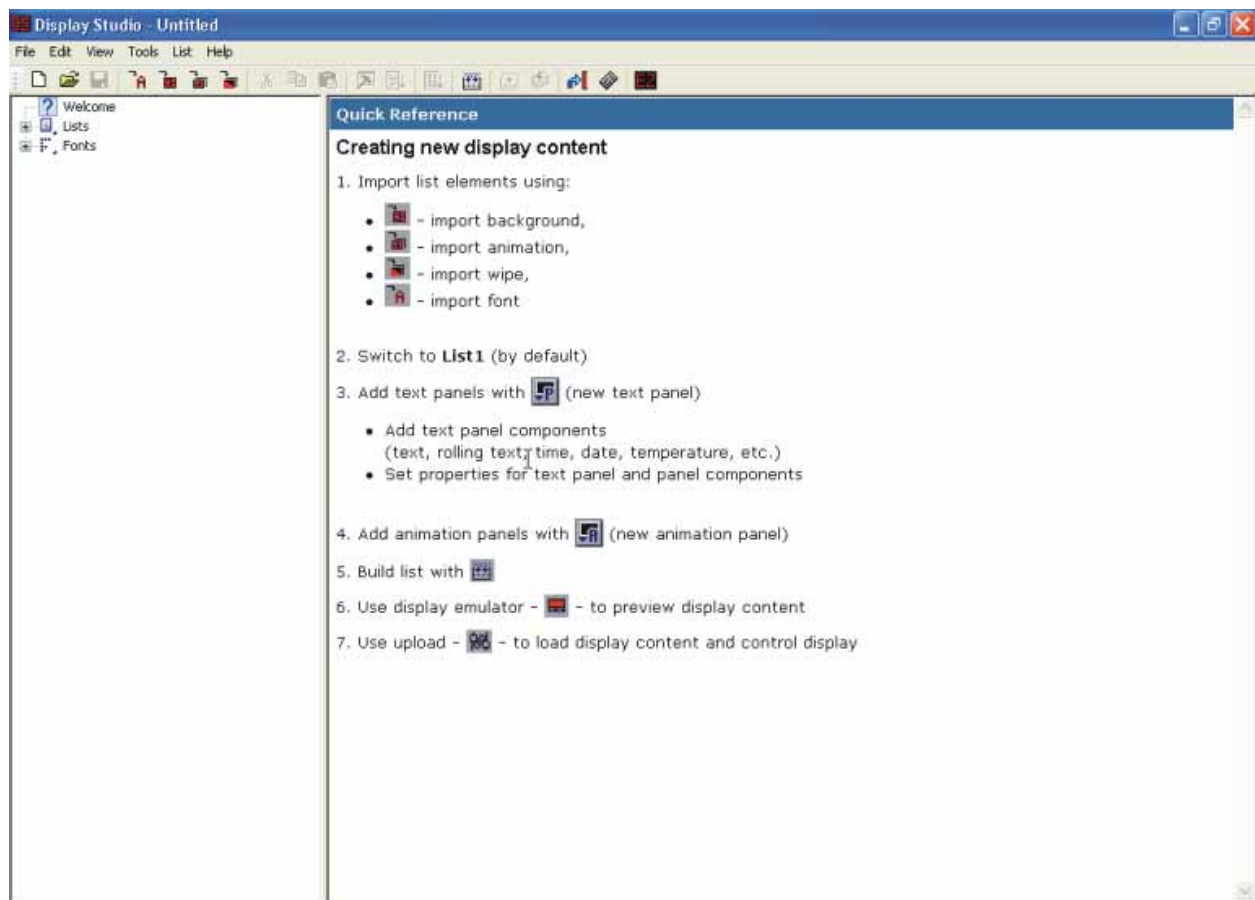
## 2 Getting started

### IMPORTANT!

Before you can connect to your Display you have to refer to the Manual of your Display to see how you can adjust the IP-Configuration that the Display is communicating with your Computer.

### 2.1 Main Window

By default, after program launching a blank project is open. Project encapsulates the whole variety of data that could be displayed on LED displays. Project file format is with extension .dds. User can load previously saved project using **File/Open**, or via  toolbar button. Picture below shows application main window just after launching.

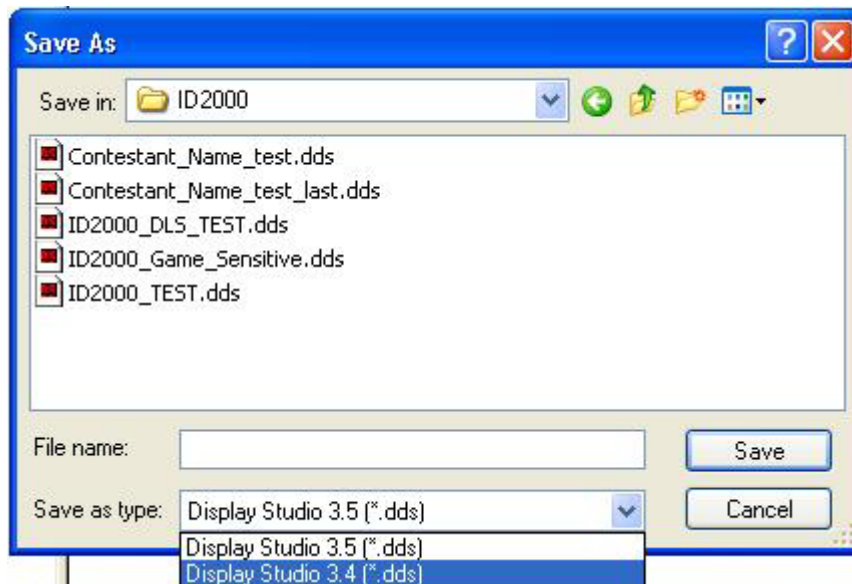


Left panel is a tree view that enables navigation through project. Right-hand side panel of the main window displays details for current tree view selection.

## 2.2 Tree view for project navigation

As was mentioned earlier, tree view on the left-hand side of the main window is intended for navigation through and displaying of the project contents. Project data categorization could be described as:

- **Lists** – project lists. **List** is a procedure by which contents (textual and animations) are displayed. List completely defines instructions for display processor and this code is automatically generated from list. By the way, user can enter this code directly.
- **Animations** – all loaded animations in project.
- **Wipes** – all wipes currently assigned to project. A purpose of a wipe is an effective animated transition from previous image into new image. When we define a panel with a starting wipe, which means that the last snapshot from previous panel, whether it was animation or textual panel, will “amalgamate” into the first snapshot of the panel. A series of snapshots of this «animated» transition is defined using the rule that white pixels mark pixels from previous image that will be shown and black from new image. In the other words, wipe, as we see it as a standalone animation, is a wipe from white to black.
- **Fonts** – all fonts assigned (added) to the project
- **Backgrounds** – all images assigned to the project that will be used as a background.
- If display is previously loaded, you can download project from the display. On menu bar press **File/Open from Display**. Application will download display’s lists and create project with that lists.
- If you want to use dds file, created under Display Studio 3.5 or higher number version, with Display Studio 3.4 or lower version number, then from menu choose **File/Save as...** , dialog will appear as shown on picture below.
- Then save project file as Display Studio 3.4 type.



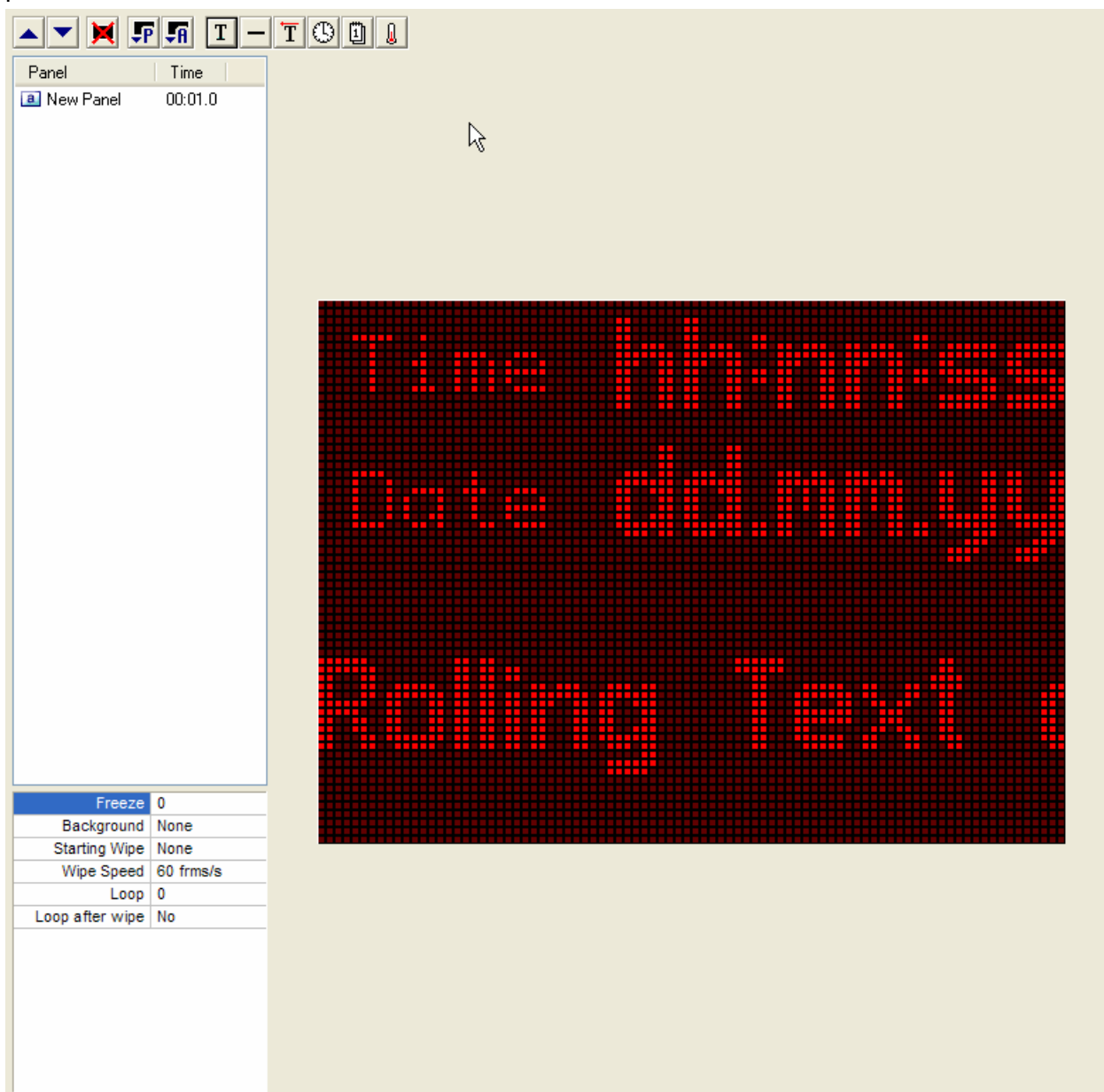
### 3 Lists

New project contains one blank list, **List1**. New lists are added using menu option **File/Insert/List**. There are two modes for working with lists: **List Builder** and **List Source**. Modes can be switched by corresponding toolbar buttons or by menu options: **List/Builder** or **List/Source**.

#### 3.1 List Builder

List builder is used for display multimedia contents defining. List consists of panels: textual or animated. **New text panel** and **new animation panel** buttons are used for adding panels.

**Move panel up** and **Move panel down** buttons can be used to change sequence of panels.



### 3.1.1 Text panel

Panel properties are shown and can be edited in the box under the list of panels (bottom-left of the picture above). Text panel properties are:

- **Freeze** – Duration of the panel appearance on the display ( \* 100ms), i.e. Freeze = 10 means that panel is going to be displayed 1 sec.
- **Background** – defines a name of the background that panel appears with, if it's set to **None**, panel appears without background.
- **Starting Wipe** – Define starting wipe of the panel. There will appear the wipes that are included in the project. Beside those wipes, there also can appear 'Default wipes'. Those are the wipes that are pre-loaded into the display. Number of preloaded wipes is defined with one of **display parameters (embedded wipe count parameter)**.
- **Wipe Speed** – refers to wipe speed (0 – stands for the slowest, 10 – fastest).
- **Loop** define how many times the panel will be displayed
- **Loop after wipe** – Wipe effect is included in Loop if **Loop after wipe** is checked, otherwise wipe effect is only at transition form previous panel to current.

User defines new values in the fields on the right-hand side of the property box, by selecting from combo box, and then pressing ENTER.

User adds components to text panel via command buttons placed on the top of the **List builder View**. After adding a panel component or after selecting a component from the panel view (right side of the list builder window), property box displays the selected component properties. Properties **Left** and **Top** can be changed by component dragging and moving in the panel view.

Setting of the **Font** property is enabled only to fonts that are assigned (added) to project. **Zoom** represents text zooming and takes integer values between 1 and 4. This, along with other properties will be explained shortly while describing various panel components. The panel components are:


- **Text** – Static text.
- **Horizontal line –Ypos** refers to vertical coordinate of the line placement.
- **Rolling text –Left Start** property is a horizontal coordinate of the initial left text edge (just before start scrolling). If the flag **AutoSize** is set to **Yes** properties **LeftStart** is determined automatically so the text would scroll the whole display area. **Speed** property defines scrolling speed in fps (frames per second). **Direction** determines rolling direction, can be **Right to Left** and **Left to Right**. Direction is important for a script type different than Latin (Right to Left), like Arabic (Left to Right).
- **Time** – current time. Use the **Format** property to set form of time displaying. **Date** – current date, **Format** defines form of its displaying.
- **Temperature** – current temperature (**C** – Celsius, **F** – Fahrenheit).

### 3.1.2 Animation

**New animation panel** button is used for adding animation. Animation properties are:

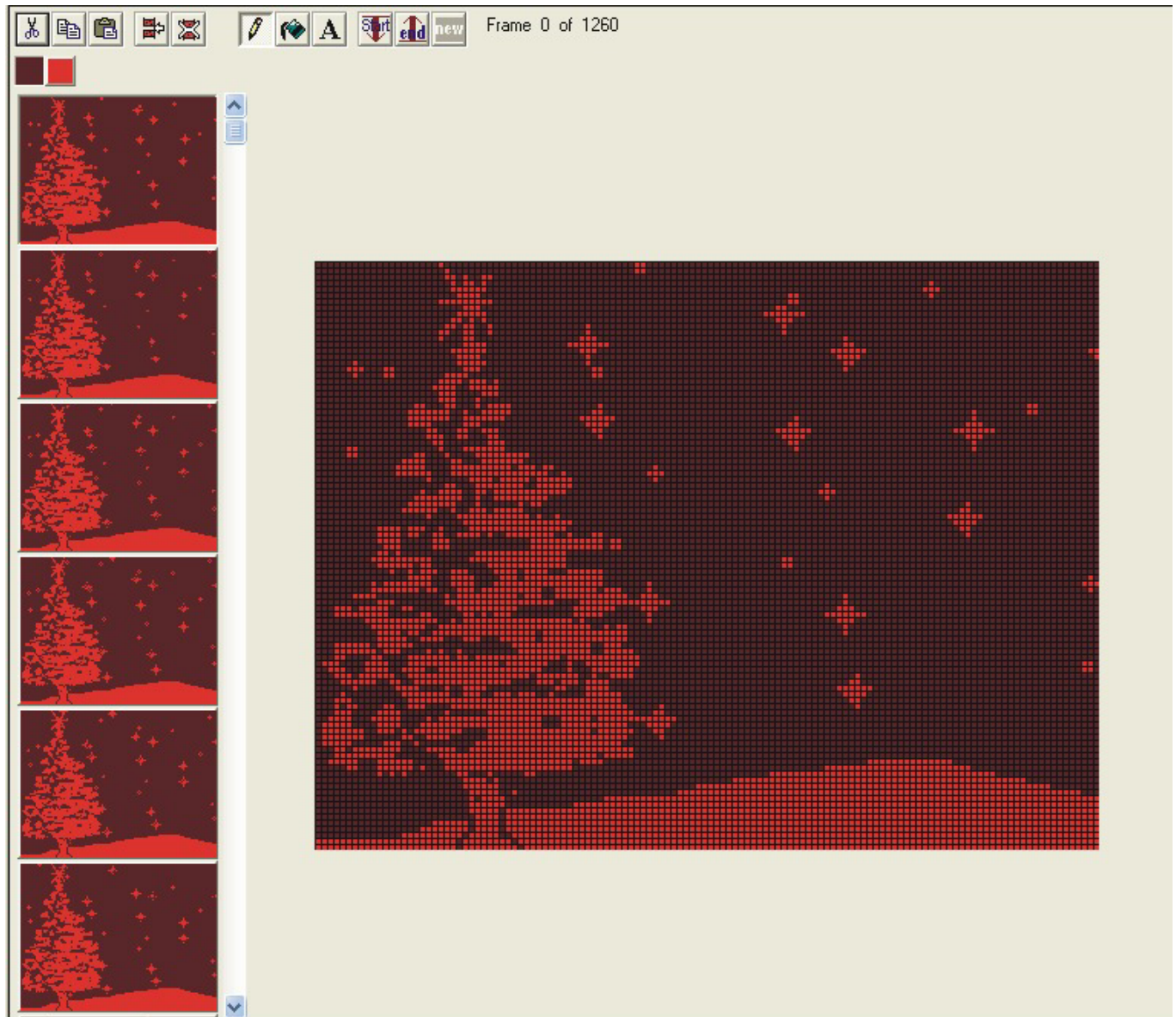
- **Animation** – the animation name.
- **Speed** – playing speed in fps (frames per second)
- **Freeze** – duration of last frame in tenths of a second
- **Starting Wipe** – Define starting wipe of the panel, if it is set to **None**, panel appears with no starting wipe.
- **Wipe Speed** – refers to wipe speed (0 – stands for the slowest, 10 – fastest).
- **Loop after wipe** – Loop choose animation after wipe.

### 3.1.3 List compiling

List definitions from the list builder should be translated to a sequence of microinstructions before loading to display. A click on the **Build** button  from the toolbar or menu options **List/Build** invokes such action.

After that, generated code can be displayed by changing mode to **List Source**

## 4 Animations and wipes




#### 4.1 Adding animation/wipe to the project


User can use existing animation by importing it from .avi or .abm file by options **File/Import/Animation** or **File/Import/Wipe**. There is always conversion from color to monochromatic performed when the animation is imported, so we can use color animations also.

User can also add new animations (**File/Insert/Animation** and **File/Insert/Wipe** menu options).

New animations or wipes can be created from existing by selection of a part from them. We

mark frame of an existing application as a starting frame (select frame and click **start**  toolbar button) and also mark one as an ending frame (by selecting it and clicking on **End**

 toolbar button). When we do like it's described, sequence of frames from starting to

ending will be marked as selected. A click on the **New**  toolbar button, creates a new animation/wipe made of selected frames. In the case of selecting just one frame and creating new, a background image would be created.

#### 4.2 Animation/wipe editing

Animations and wipes are located under items **Animations** or **Wipes** in the tree view on the left side of the main window. When we select an animation or a wipe (background also), *Animation editor* displays animation contents, frame by frame, on the right side of the main window.

We select the frame whose contents want to zoom or edit and frame is displayed on the right side of the *Animation editor*, as on the picture above.

The toolbar on the top of this window enables easy switching between different modes (i.e. we can draw in point-by-point mode or we can fill an area with one move).

### 5 Backgrounds

Background image is a static image that is present at the background of current contents displayed on the display. We can either use existing (option **File\Import\Background**) or create new (menu option **File\Insert\Background**).

As it was described at the animation and wipes section, we can select a random frame and create a background from it.

Editing is equivalent to animations/wipes editing.

## 6 Fonts

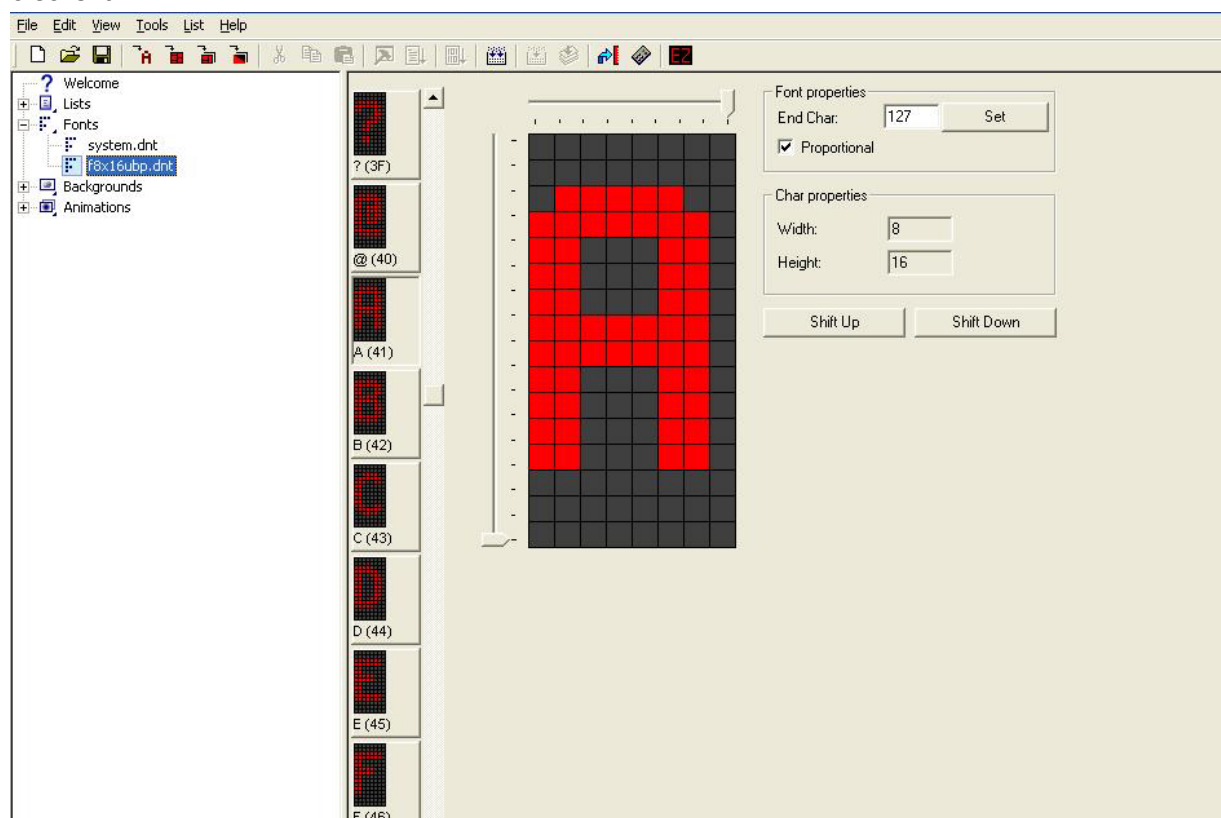
### 6.1 Adding fonts to the project

User can import fonts from DNT file (menu option **File/Import/Font/DNT**) or create new (menu option **File/Insert/Font**).

Also user can import windows true type font (menu option **File/Import/Font/Windows font**), but after importing this kind of font You must do some **Font editing** (see next section) to correct font dimension.

### 6.2 Font Editing

Fonts are located under **Fonts** node in the tree view on the left side of the main window. Selecting of an item that represents font opens **font editor** at the right side of a main window. Font editor is made of 2 parts: list for character selection on the left side, and editing area on the right side. In editing area left mouse button sets a pixel while right mouse button clears it.



Using **End Char** text box, user can customize a number of characters in the font. Enter in text box and click on **Set** button set this value. The **Proportional** check box is used to specify whether fonts are proportional (each character has it's own width according to their shape) or not (all characters has the same width).

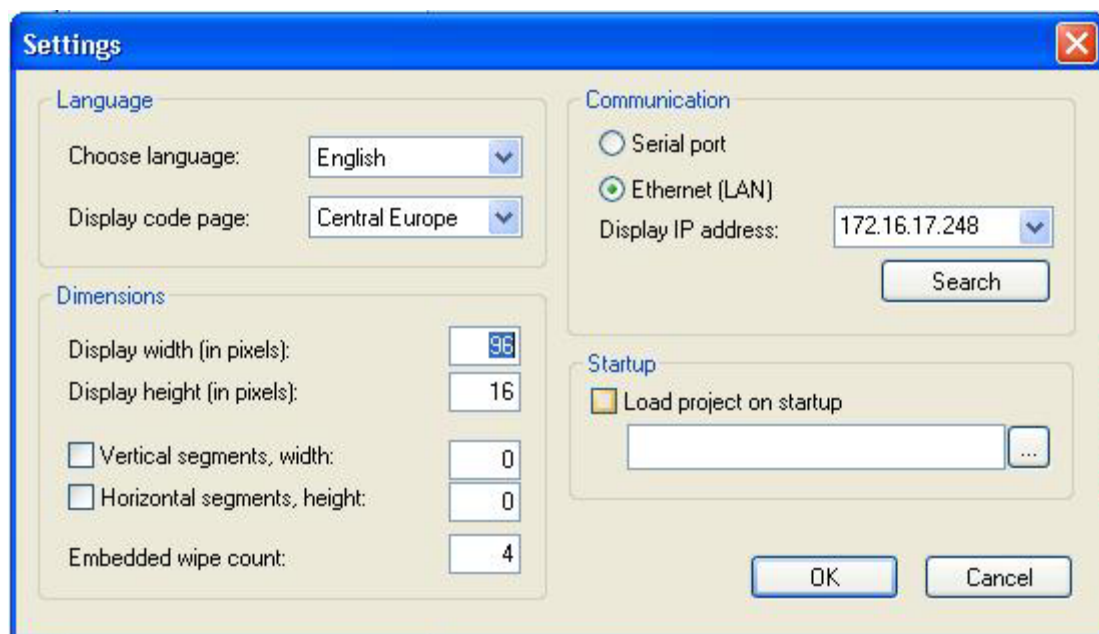
**Shift Up** and **Shift Down** buttons are for moving selected font object one place up or down, respecting that first font object have number 20 and the last have number defined with **End Char** property. Numbers beside font object are in hexadecimal format.

The fonts height and width are specified with sliders above and left of display of selected character.

## 7 Project settings

For establishing communication with the display, basic project parameters must be set. Project settings are launched by option **Tools/Settings**. Here you can set some options. Options are:

- **Choose language** – from drop down menu choose appropriate language. Upon installing 2 languages are available: English, German. Adding new language is described in **Adding new language and welcome (quick reference) screen** section of this document. Upon changing the language, Display Studio must be restarted to accept the changes.
- **Display Code Page** – from drop down menu choose appropriate code page. Useful for different keyboard layout. Example: If You use German keyboard switch **Display Code Page** to Western.
- **Display width, height (in pixels)**
- **Vertical segments, width** – if display is divided in vertical segments
- **Horizontal segments, height** – if display is divided in horizontal segments, for multi-line displays
- **Embedded wipe count** – number of embedded wipes in the display.
- **Serial Port** - communication via com port
- **Ethernet (LAN)** – communication via Ethernet (LAN)
- **Display IP address** - can be choose from combo box after pressing search button.
- **Load project on startup** – for choosing project which will be loaded upon starting Display Studio



## 7.1 Display loading

Loading content into display is launched by option **Tools/Upload** or by corresponding toolbar button (**Upload**). Loading is performed using **Upload** dialog controls (presented under).



- **Load display** – starts loading to display.
- **Cancel loading** – stops loading.
- **Clear Flash disk** – Clears display flash disk (containing animations, wipes and backgrounds) and loads all media content. If unchecked, only new or changed media will be loaded
- **Verify uploaded content and generate log file** – upon uploading the display, if this option is checked, uploaded content (in display memory) is compared with corresponding list from Display Studio. After comparing Log files are generated and placed in the Display Studio folder.

## 8 List scheduler

You can define time/date of list activation using list scheduler option (**Tools/Scheduler**).

| Lists        | Date       | Time     | Method |
|--------------|------------|----------|--------|
| Commercial   | 12.03.2005 | 15:23:00 | Insert |
| TestLines    | 12.03.2005 | 15:25:00 | Insert |
| TimeDateTemp | 12.03.2005 | 15:30:00 | Insert |

- Procedure for scheduling lists is:
- Enter list activation date in field Date as dd.mm.yyyy. If Everyday button is activated, list will be activated/inserted every day at given time.
- Enter list activation time in field Time as hh:mm:ss
- Pick a list in List name field
- Choose list activation method – Insert selected list (list content is displayed once, after that previously displayed list continues) or Set selected list as active (list content loops until some other list is set active)

Press **Add** button. List activation info appears in list below. Existing list schedules can be removed using **Remove** button.

## 9 Remote control

Remote control is used for controlling display emulator or real LED display. It has following functions:



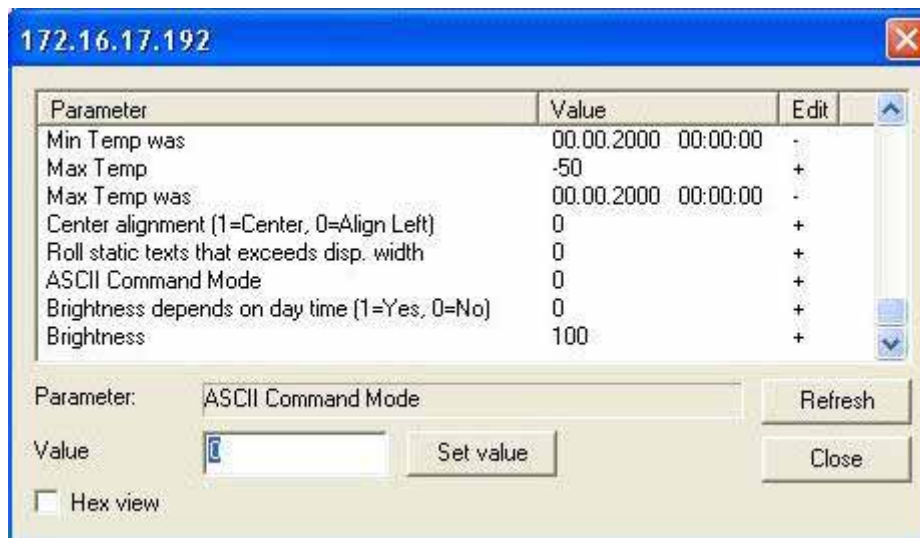
- **EMU** - switches to emulator mode, display content is shown in emulator window (with black background)
- **DISP** - switches to display mode
- **Play** - starts playing
- **Stop** – stops playing and turns display off
- **Pause** - Pauses playing
- **Frame by frame** – toggles frame-by-frame playing mode. When this mode is on, **Play** command shows next frame.
- **Skip** – skips to next panel (doesn't work in emulator mode)
- **Reset** – simulates display reset.
- **Show lists** – extends dialog to show available lists
- **Insert selected list** – starts playing selected list, when selected list completes, playing continues with active list
- **Set active list** – sets selected list as active
- **Set time** – adjusts display time to PC time (doesn't work in emulator mode)

Remote control is activated with **Tools/Remote control** or  button on taskbar.

## 10 Display parameters

(Only for advanced users)

User can see or even edit some display parameters. The list of parameters appears by meny option **Tools/Parameters**. Some of the parameters relates to display work (like Working Hours or Number of Resets or Time of the last reset), some relates to hardware display configuration (Height, Width, Flash disc size,...) some to specific software features (like 'CR force LF', 'Special start number' ...) etc. Number of parameters and their values depend on concrete display installation.



## 11 Active Panels

Each List is contains '**panels**'. Panels can be textual or animation and can be **fixed** or **controlled**. Animation panels can be only fixed, defined in 'Display studio'. Textual panels can be fixed (defined in 'Display studio') or can be 'controlled' or so called '**active panels**'. Active panels are placed in the lists as textual panels that contain only static text \$Rxx or \$Cxx. xx are 2 digits – panel number and \$R and \$C are commands that refers on RAM or Code panels. When display, interpreting the panels, arrive on text \$Rxx or \$Cxx, it will display the content defined by the values of corresponding active panel's attributes. So, at that moment will not be displayed \$R(C)xx on the display but will be displayed text that is in \$R(C)xx Text attribute, with the effect that is in \$R(C)xx Effect attribute, with font that is in \$R(C)xx Font attribute etc.

Each list can consisted of a desired number of panels repeated in desired order. For example, in list4 you can have 2 animation panels, 4 fixed textual panels and 3 controlled panels. The number and order of panels in the list can be defined only in 'Display studio' software.

Active panels have following **attributes**:

|                      |  |
|----------------------|--|
| <b>Active</b>        | 1 digit, '0' panel is inactive or '1' - panel is active  |
| <b>Wipe</b>          | 1 digit, starting wipe number, '0' – no starting wipe, '1' – '4' one of four embedded wipes.   |
| <b>Wipe speed</b>    | 1 digit, from '0' (slowest) to '9' (fastest)   |
| <b>Effect</b>        | 1 digit, appearance, can be:<br>'0' – static text<br>'1' – flashing text<br>'2' – rolling text, from left to right<br>'3' – rolling text, from right to left |
| <b>Rolling speed</b> | has effect only if effect is 'rolling'. 1 digit, from '0' (slowest) to '9' (fastest)   |
| <b>Font</b>          | Font number in order how they are loaded from display studio, starting from '0'.   |
| <b>Zoom</b>          | 1 digit, '0'-'3'   |
| <b>Freeze</b>        | Two digits, from '00' to '99' Has effect only if effect is 'static'. Freeze time in 1/10sec units,   |
| <b>Loop</b>          | How many times the panel will be displayed   |
| <b>Text</b>          | Panel text, string that will be displayed  |

Actually, active panels are data structure into the display whose attributes define their appearance. Controlling active panels means changing their attributes (like font, starting wipe, freez time,...) using one of few available methods (like Serial ASCII commands, Easy Messenger, Keyboard, Auto Upload using script file, over the Web using Web Ticker technology etc.). Because of variety of available methods for changing their attributes, active panels are the most common and the easiest technology for interfacing the display with external application software, external hardware, Web or LAN data, keyboards and other devices or software. Some of those methods will be explained later.

The difference between RAM and Code panels is that panels in RAM memory can be programmed very fast as many times as you want. The only disadvantage is that it loses the data after each power off or reset.

Panels in CPU FLASH memory, once programmed remain in the display as long as new content is programmed on the same locations, but disadvantage is that programming is slower that might cause display flickering or slowing down while programming is in progress.

Another disadvantage is that this memory can be programmed 100.000 times. It means that data that are frequently changing (for example race running time) should be stored in RAM panels.

Active panels are used for display auto upload via script (txt) file.

## 12 Script file

Keyword '\$Panel' is for panel saved in flash memory (after reset is valid) while keyword '\$RAMPanel' is for panel in RAM.

after key word '\$Panel' or '\$RAMPanel' state panel's attributes. Next 6 rows is for panel's text lines.

at text line beginning You can define line effect with command \$E(rolling, blinking or static) and font with \$F.

if \$E command is missing, text line is static and centred

if \$F command is missing, font for that text line is system 7x6 (letters are 7x5 + 1 spacing)

Lines starting with // are comment lines and they are excluded from analysis

\$Panel Panel Number Active Wipe Wipe Speed Rolling Speed Blinking Period Freeze Loop

[\$Ex,][\$Fx,] TextLine1

[\$Ex,][\$Fx,] TextLine 2

[\$Ex,][\$Fx,] TextLine 3

[\$Ex,][\$Fx,] TextLine 4

[\$Ex,][\$Fx,] TextLine 5

[\$Ex,][\$Fx,] TextLine 6

PanelNumber = 2 digits from 00 to 99

Active = 0 (inactive) or 1 (active)

Wipe = transition from previous panel to current, from 0 to 9 only for embedded wipes

WipeSpeed = from 0 (slowest) to 9 (fastest)

RollingSpeed = from 0 (slowest) to 9 (fastest), declared for one rolling panel but applied for all rolling panels in display

BlinkingPeriod = 2 digits from 00 to 99, declared in 1/10 sec. (0 to 9.9sec), declared for one blinking panel but applied for all blinking panels in display

Freeze = 2 digits, from 00 to 99, declared in 1/10 sec. (0 to 9.9sec). Counting starts after last rolling

Loop = define iterations count for display line, 3 digits, from 000 to 999 (000 and 001 represent 1)

\$Ex, Effect, x can be: S – static centred B – blinking centred R - rolling, L – static left aligned or b - blinking left aligned

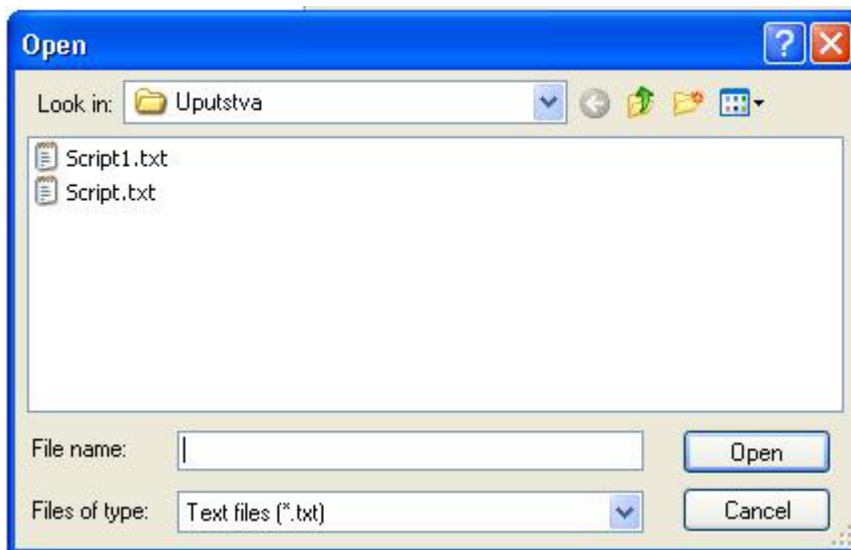
\$Fx = Font, x from 0 to FC-1, where FC is **FontCount** - number of fonts included in display studio current project (from which script file is loaded) and loaded in the display.

x=0 is for first font in project file, x=1 for second, x=FC-1 for FC font in project file.

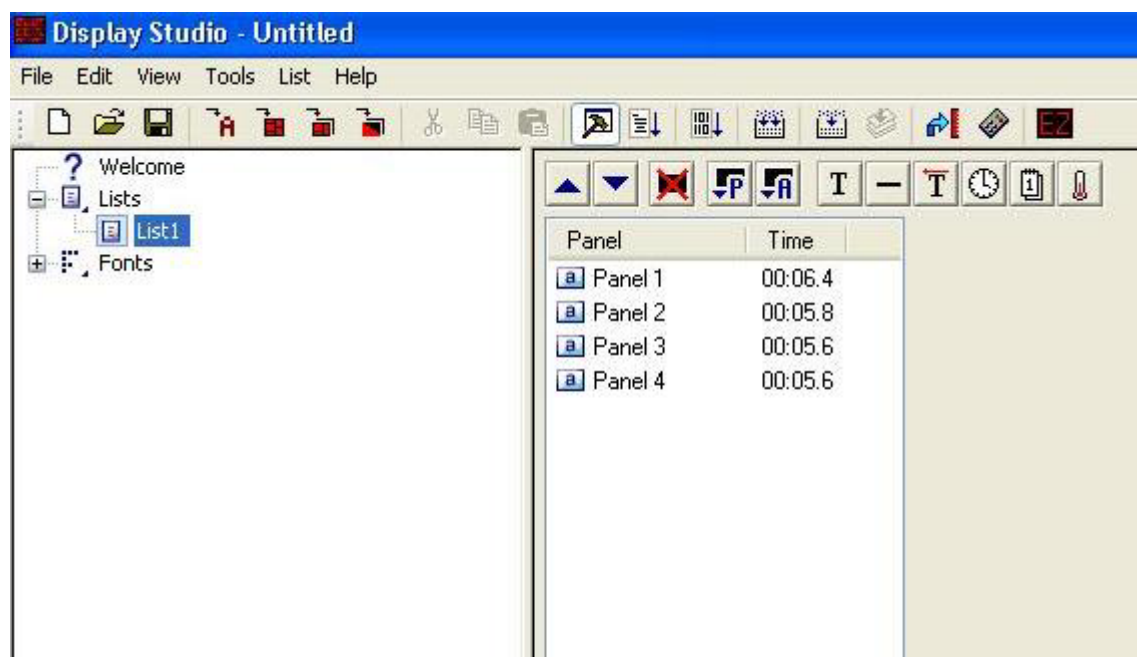
## 13 Auto Upload

Auto upload via Script file is very useful when You have a lot of displays on same or different places that must be controlled from one place. Using script file for Displays auto upload give excellent platform for end-users. End users can use software that edit script file automatically.

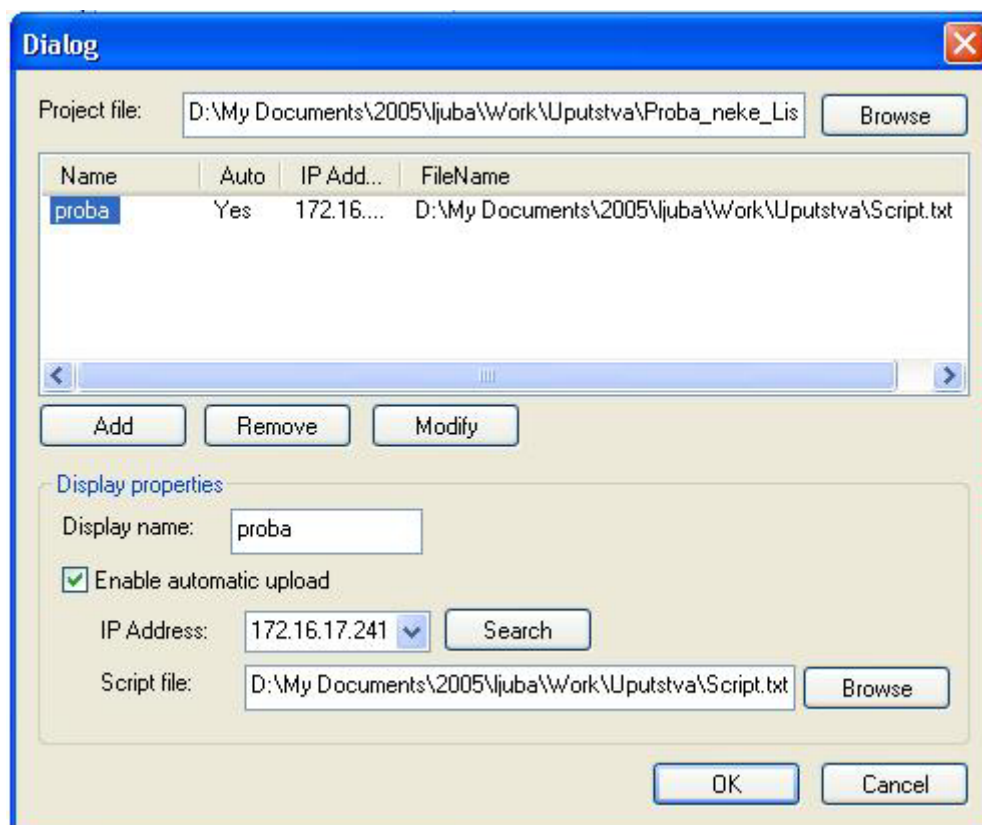
**Auto Upload** option is active if at least one list is created in project (dds) file. First create project and adjust display dimension (**Tools/Parameters** to see dimensions and then **Tools/Settings**). Then import as many fonts as You will use in script (txt) file (see Script file example). Point to list, then from **Tools/ Test script file**.



Choose script file. If script file is correct, project adds from script file panels(see picture)



Compile list, then upload in display. Save list.From **Tools/ Auto upload**.



For project file choose previously saved project file.

Press **Search** button and from drop down list choose ip address of the display. Press **Browse** button and choose script (.txt) file from which project is created. Check **Enable automatic upload**, in **Display name** enter name of the display and press **Add** button. In upper window will appear name of the display, auto upload status, ip address and script file full path. Press **OK**. Changes in script file will have effect on the display only if Display Studio is active (can be minimized on taskbar). Script file content can be changed from any text editor or other application. You can use as many displays as You want with **Auto Upload** feature, with same or different script file. Each of displays hat is used with Auto Upload feature must be uploaded as described in this (Auto Upload) section.

## 14 Script file example

```
// Beginning of script file
//keyword '$Panel' is for panel saved in flash memory (after reset is valid) while keyword
'$RAMPanel' is for panel in RAM and it is not valid after reset.
//after key word '$Panel' or '$RAMPanel' state panel's attributes. Next 6 rows is for panel's
text lines.
//at text line beginning You can define line effect with command $E(rolling, blinking or static)
and font with $F.
//if $E command is missing, text line is static and centred
//if $F command is missing, font for that text line is system 7x6 (letters are 7x5 + 1 spacing)
//Lines starting with // are comment lines and they are excluded from analysis
//-----
//$Panel PanelNumber Active Wipe Wipe Speed Rolling Speed Blinking Period Freeze Loop
//[$Ex,][$Fx,] TextLine1
//[$Ex,][$Fx,] TextLine 2
//[$Ex,][$Fx,] TextLine 3
//[$Ex,][$Fx,] TextLine 4
//[$Ex,][$Fx,] TextLine 5
//[$Ex,][$Fx,] TextLine 6

//PanelNumber = 2 digits from 00 to 99
//Active = 0 (inactive) ili 1 (active)
//Wipe = transition from previous panel to current, from 0 to 9
//WipeSpeed = from 0 (slowest) do 9 (fastest)
//RollingSpeed = from 0 (slowest) to 9 (fastest), declared for one rolling panel but applied for
all rolling panels in display
//BlinkngPeriod = 2 digits from 00 to 99, declared in 1/10 sec. (0 to 9.9sec), declared for one
blinking panel but applied for all blinking panels in display
//Freeze = 2 digits, from 00 to 99, declared in 1/10 sec. (0 to 9.9sec). Counting starts after
last rolling
//Loop = define itration count for display line, 3 digits, from 000 to 999 (000 and 001
represent 1)
//$Ex, Effect, x can be: S – static centred B – blinking centred R - rolling, L – static left
alligned or b - blinking left aligned
//$Fx = Font, x from 0 to FC-1, where FC is FontCount - number of fonts included in display
studio current project (from which script file is loaded)
x=0 is for first font in project file, x=1 for second, x=FC-1 for FC font in project file.
```

```
//-----  
-----  
//Example for IP 30 display with 6 text lines  
///-----  
-----
```

```
$Panel 01 1 1 9 10 20 000  
$ER $F0 Welcome to Atlas Bank  
*****  
$ER $F0 Interest for CSD savings  
$F0 at sight 9.2%  
$F0 time deposit:  
$ER $F0 1 year 10.2 %, 2 years 11.4%, 3 years 12.7%
```

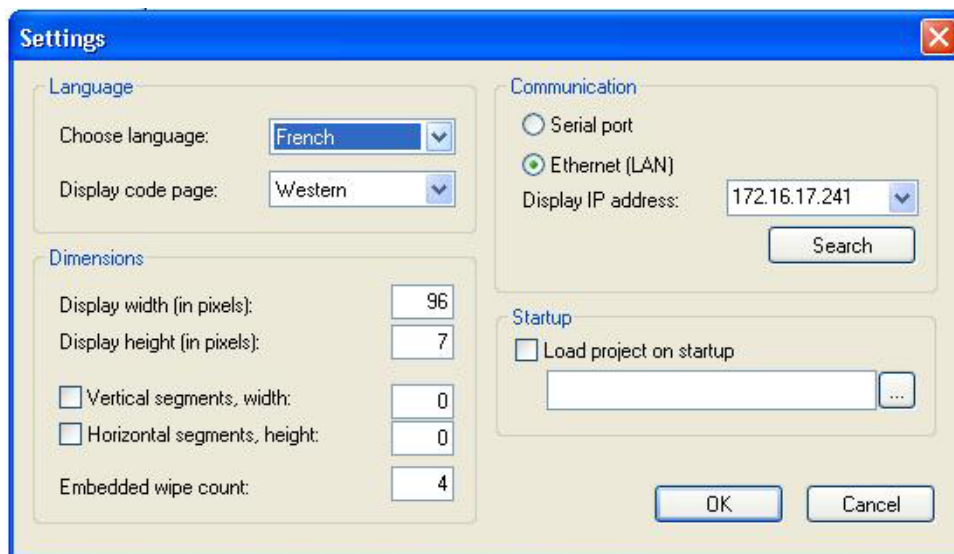
```
$Panel 02 1 1 9 10 20 000  
$ER $F1 Welcome to Atlas Bank  
*****  
$ER $F1 Interest for EUR savings  
$F1 at sight 2.2%  
$F2 time deposit:  
$ER $F2 1 year 3.2 %, 2 years 4.4%, 3 years 4.7%
```

```
$Panel 03 1 1 9 10 20 000  
$ER $F3 Welcome to Atlas Bank  
*****  
$F3 Loan for  
$F3 estate  
$F3 Interest 8.2%  
// End of script file
```

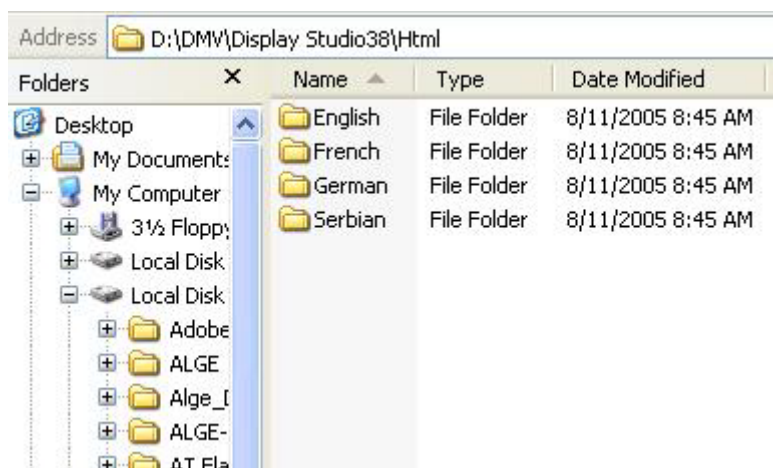
## 15 Add new language

**(Only for advanced users)**

Upon installing Display Studio 2 languages are available for usage: English, German. In Display Studio folder there are 2 files that directly describe available languages: **Ing\_english.ini** for English and **Ing\_german.ini** for German language. Adding new ini file in this form (**Ing\_language.ini**) in Display Studio folder is enough for new language. Example: valid **Ing\_french.ini** file in Display Studio folder result new language (French) in **Tools/Settings** dialog (see picture below). Form of ini file is **Ing\_LANGUAGE.ini**, **Ing\_** is descriptor that specify ini as language file, while **LANGUAGE** is language name that appear **Tools/Settings** dialog.




Adding new welcome (quick reference) screen is necessary when new language is added. **Html** subfolder is placed in Display Studio folder. Inside **Html** folder each of installed languages have folder (see picture below).













If new language is French, we added **Ing\_french.ini** in Display Studio folder, and we add folder **French** in **Html** folder. Generally, for adding new **Language name** and **language** you need:

**Ing\_LANGUAGENAME.ini** in Display Studio folder, **LANGUAGENAME** folder in **Html** subfolder.

Each of language folder have welcome.htm page (picture below) that appear upon starting Display Studio. Easiest way to create language folder is to create folder with name of the language (eg. for French create French folder in Html folder) and then copy content of English folder in that folder. After you translate the welcome.htm page from English to appropriate language your new language is available.

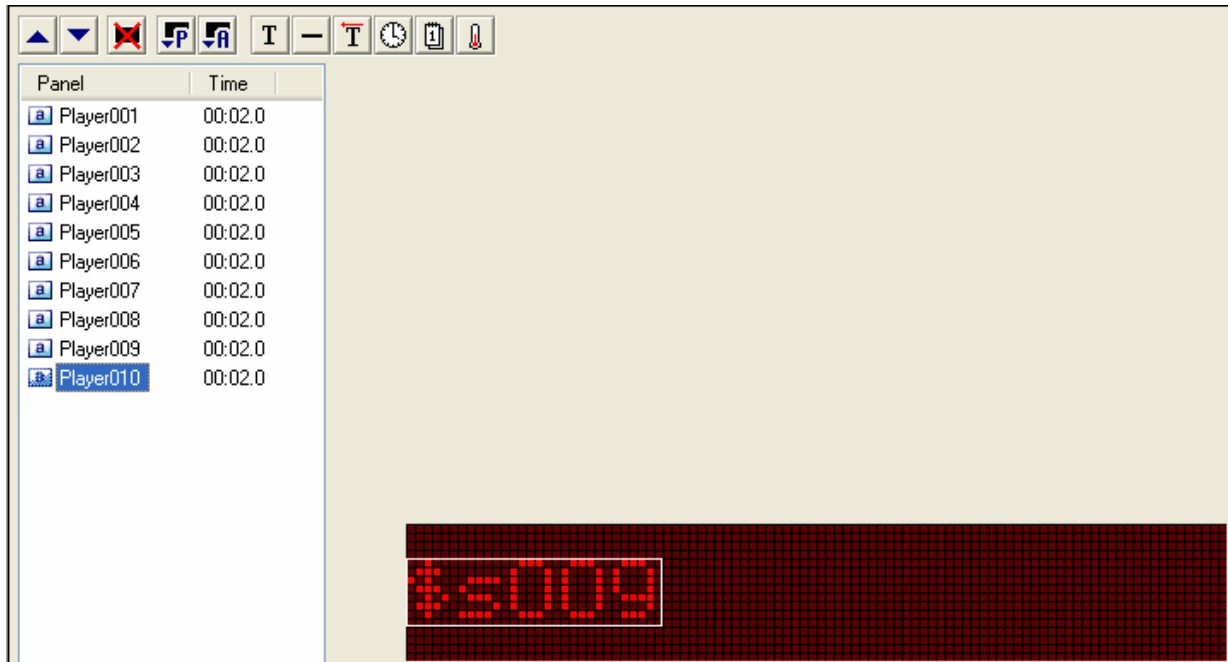
Address  D:\DMV\Display Studio38\Html\French

| Folders             | Name   | Size | Type          | Date Modified      |
|---------------------|--|------|---------------|--------------------|
| vents               |  build.gif    | 1 KB | GIF Image     | 11/29/2001 9:55 AM |
| ster                |  emulator.gif | 1 KB | GIF Image     | 11/29/2001 9:56 AM |
| copy (A:)           |  impanim.gif  | 1 KB | GIF Image     | 11/29/2001 9:26 AM |
| Disk (C:)           |  impbmp.gif   | 1 KB | GIF Image     | 11/29/2001 9:25 AM |
| Disk (D:)           |  impfont.gif  | 1 KB | GIF Image     | 11/29/2001 9:24 AM |
| lobe                |  impwipe.gif  | 1 KB | GIF Image     | 11/29/2001 9:26 AM |
| .GE                 |  newanim.gif  | 1 KB | GIF Image     | 11/29/2001 9:39 AM |
| ge_Display_Studio   |  newpanel.gif | 1 KB | GIF Image     | 11/29/2001 9:38 AM |
| .GE-TIMING          |  upload.gif   | 1 KB | GIF Image     | 11/29/2001 9:58 AM |
| T Flash & USBDriver |  welcome.htm  | 4 KB | HTML Docum... | 3/26/2005 8:47 PM  |
| ackup2004           |  |      |               |                    |
| ackup2005           |  |      |               |                    |

## 16 Excel to Display

(Only for advanced users)

First You must create list with panels containing **static text** strings \$sxxx, where xxx is player number.



Then build project and upload it into display.

From **Tools** choose **Excel to display**, dialog appear.

Press **Open**. From **Open dialog** choose appropriate xls file, ranking order and player name appear in dialog.

Press **Send** to load player names into display.

## 16.1 Variables for Timing Devices and Game-Consoles

You can create an active list which is showing Timing-Data and Names directly from your ALGE Timing-Device.

### 16.1.1 Variables for Timers

Received by RS232 or black-yellow banana-sockets.

| \$ Variable   | Description   |
|---------------|---|
| \$D195        | Name of the Bib last received from names entered from scoreboard software |
| \$D196        | Current Time received from Timing-Device, automatic format, hh:mm:ss.th   |
| \$D197        | Current Rank received from Timing-Device                                  |
| \$D198        | Current Bib received from Timing-Device                                   |
| \$D199        | Name of current Bib from Excel-Startlist                                  |
| \$D200        | Nation of current Bib from Excel-Startlist                                |
| \$D201        | Club of current Bib from Excel-Startlist                                  |
| \$s000        | Name of Player with Bib 001-999   |
| \$sA00        | Nationality of Player with Bib A00-A99=000-099; B00-B99=100-199;...       |
| \$sa00        | Club of Player with Bib a00-a99=000-099; b00-b99=100-199;...              |
| \$D000-\$D194 | Byte from Datapacket  |

### 16.1.2 Variables for Game-Consoles D-CKx

Anzeigelisten die folgende Variablen enthalten zeigen die jeweiligen Informationen direkt auf der Anzeige an.

| \$ Variable | Description                                     |
|-------------|---|
| \$B00       | Score home – 3 numerals, clear leading zeros    |
| \$B01       | Score guest – 3 numerals, clear leading zeros   |
| \$B02       | Time minutes                                    |
| \$B03       | Time seconds                                    |
| \$B04       | Time 1/10 seconds                               |
| \$B05       | Remaining TimeOut Time                          |
| \$B06       | Home team name                                  |
| \$B07       | Guest team name                                 |
| \$B08       | Guest team name, left alignment                 |
| \$B09       | Number of fouls for home team                   |
| \$B10       | Number of fouls for guest team                  |
| \$B11       | Last player who has committed the foul          |
| \$B12       | Foul number for that player                     |
| \$B13*      | 1 point percentage for home team                |
| \$B14*      | 2 points percentage for home team               |
| \$B15*      | 3 points percentage for home team               |
| \$B16*      | 2 and 3 points (field) percentage for home team |
| \$B17*      | Number of fouls for home team                   |
| \$B18*      | 1 point percentage for guest team               |
| \$B19*      | 2 points percentage for guest team              |

|        |  |
|--------|--|
| \$B20* | 3 points percentage for guest team   |
| \$B21* | 2 and 3 points (field) percentage for guest team   |
| \$B22* | Number of fouls for guest team   |
| \$B23* | Name of the team who has exchanged player(s)   |
| \$B24  | Penalty1 home in format PP mm:ss (PP – player number, mm - minutes, ss – seconds)  |
| \$B25  | Penalty2 home in format PP mm:ss (PP – player number, mm - minutes, ss – seconds)  |
| \$B26  | Penalty1 guest in format PP mm:ss (PP – player number, mm - minutes, ss – seconds)                                       |
| \$B27  | Penalty2 guest in format PP mm:ss (PP – player number, mm - minutes, ss – seconds)                                       |
| \$B28  | Misconduct Penalty home in format MPP mm:ss (M – sign for misconduct<br>PP - player number, mm - minutes, ss – seconds)  |
| \$B29  | Misconduct Penalty guest in format MPP mm:ss (M – sign for misconduct<br>PP - player number, mm - minutes, ss – seconds) |
| \$B30  | ScoreHome for 1. time period   |
| \$B31  | ScoreGuest for 1. time period  |
| \$B32  | ScoreHome for 2. time period   |
| \$B33  | ScoreGuest for 2. time period  |
| \$B34  | ScoreHome for 3. time period   |
| \$B35  | ScoreGuest for 3. time period  |
| \$B36  | ScoreHome for 4. time period   |
| \$B37  | ScoreGuest for 4. time period  |
| \$B38  | Sign for poss home or set home(volleyball)   |
| \$B39  | Sign for poss guest or set home(volleyball)  |
| \$B40  | Game Period (has value E for extra time period)  |
| \$B41  | Number of timeouts for home team   |
| \$B42  | Number of timeouts for guest team  |
| \$B43  | Shot Time  |
| \$B44  | Shot Time or TimeOut (for shoving shot time and timeout time in same place)  |
| \$B45  | Penalty home which is start first.<br>(when only one penalty per team is shown) (format: PP mm:ss)                       |
| \$B46  | Penalty guest which is start first.<br>(when only one penalty per team is shown) (format: PP mm:ss)                      |
| \$B47  | Minutes of misconduct penalty home when only minutes for this penalty is shown)<br>(format: Mmm)                         |
| \$B48  | Minutes of misconduct penalty guest when only minutes for this penalty is shown)<br>(format: Mmm)                        |
| \$B49  | Score Home - two digit score (format xx)   |
| \$B50  | Score Guest - two digit score (format xx)  |
|        |  |
| \$Hxx* | Home player name, xx is player number, can be 00-99  |
| \$H9:* | Home coach name  |
| \$H9;* | Guest coach name   |
| \$H9<* | 1st arbiter  |
| \$H9=* | 2nd arbiter  |
| \$H9>* | official delegate  |
| \$H9?* | Number of spectators   |
| \$Gxx* | Guest player name, xx is player number, can be 00-99   |
| \$Pxx* | Player name leaded with player number. Depending on xx is:<br>0-4 5 Home players currently in the field                  |

|       |  |
|-------|--|
|       | 5-9 5 Guest players currently in the field<br>10-14 Home players who were in the field<br>15-19 Guest players who were in the field<br>20-24 Home players who were and who are currently in the field<br>25-29 Guest players who were and who are currently in the field<br>30-34 Players of the team that exchanged player(s) currently in the game<br>40-44 Players of the team that exchanged player(s) were in the game<br>50-54 Players of the team that exchanged player(s) were and are in the game |
| \$hxx | Home player name, typed from command console, xx can be 0-11   |
| \$gxx | Guest player name, typed from command console ,xx can be 0-11  |
| \$ixx | Home player number, typed from command console, xx can be 0-11   |
| \$jxx | Guest player number, typed from command console, xx can be 0-11  |
| \$Mxx | Messages typed from command console, xx is message number, can be 00-19  |

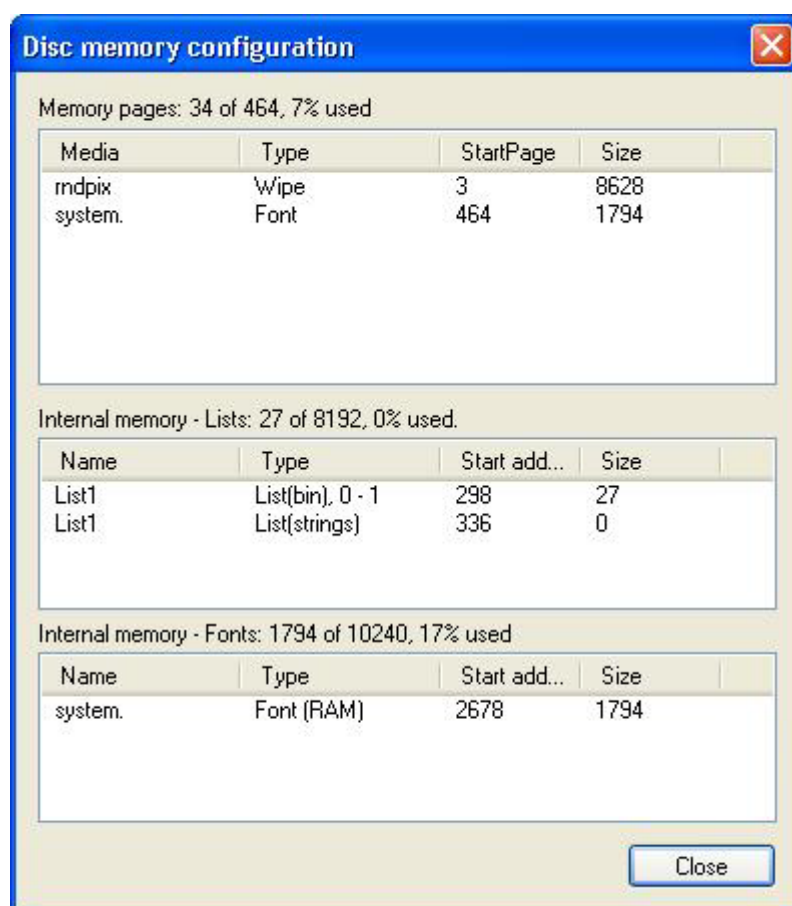
\* Those variables are entered from Scoreboard software or Basketball statistic software.

## 17 Memory Map configuration

(Only for advanced users)

Animations, wipes and backgrounds, on the display side, are physically located at display flash disk, or as we call it display memory. This feature is used to define and manage storing in display memory. By creating or compiling of lists, memory reallocation is automatically performed. User can be informed about real address in memory display that a particular frame is stored at. User can also see the amount of consumed memory.

Memory map appears by menu option **View/Memory map**. There is a list of all lists, animations, wipes, backgrounds and fonts with exact location in the memory where they are loaded and interval of addresses they are using.



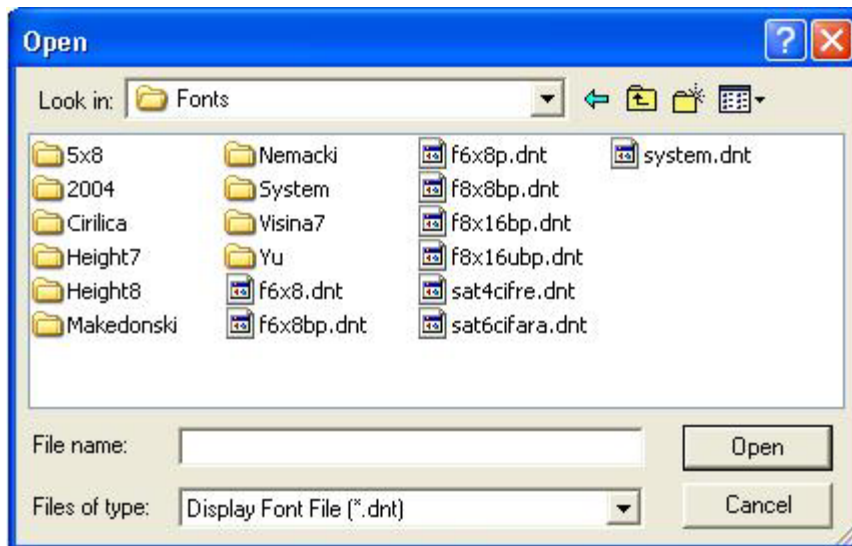
## 18 Tutorial

As a short demonstration of all described above, we will create display contents with 2 textual panels with «rolling» text, which will be divided by a wipe effect. After that, an already created animation (we have it in .avi file) will also be displayed.

So, lets go on with work. Here are project requirements:

1. Client wanted firm logo at the background and rolling text at the bottom of the display.
2. Client wanted a commercial rolling on the whole display area (maximum possible font size).
3. Client has prepared animation he wanted to be displayed.

In order to work with text we need to add fonts to project. Select menu option - **File/Import/Font** and chose font file (.dnt).



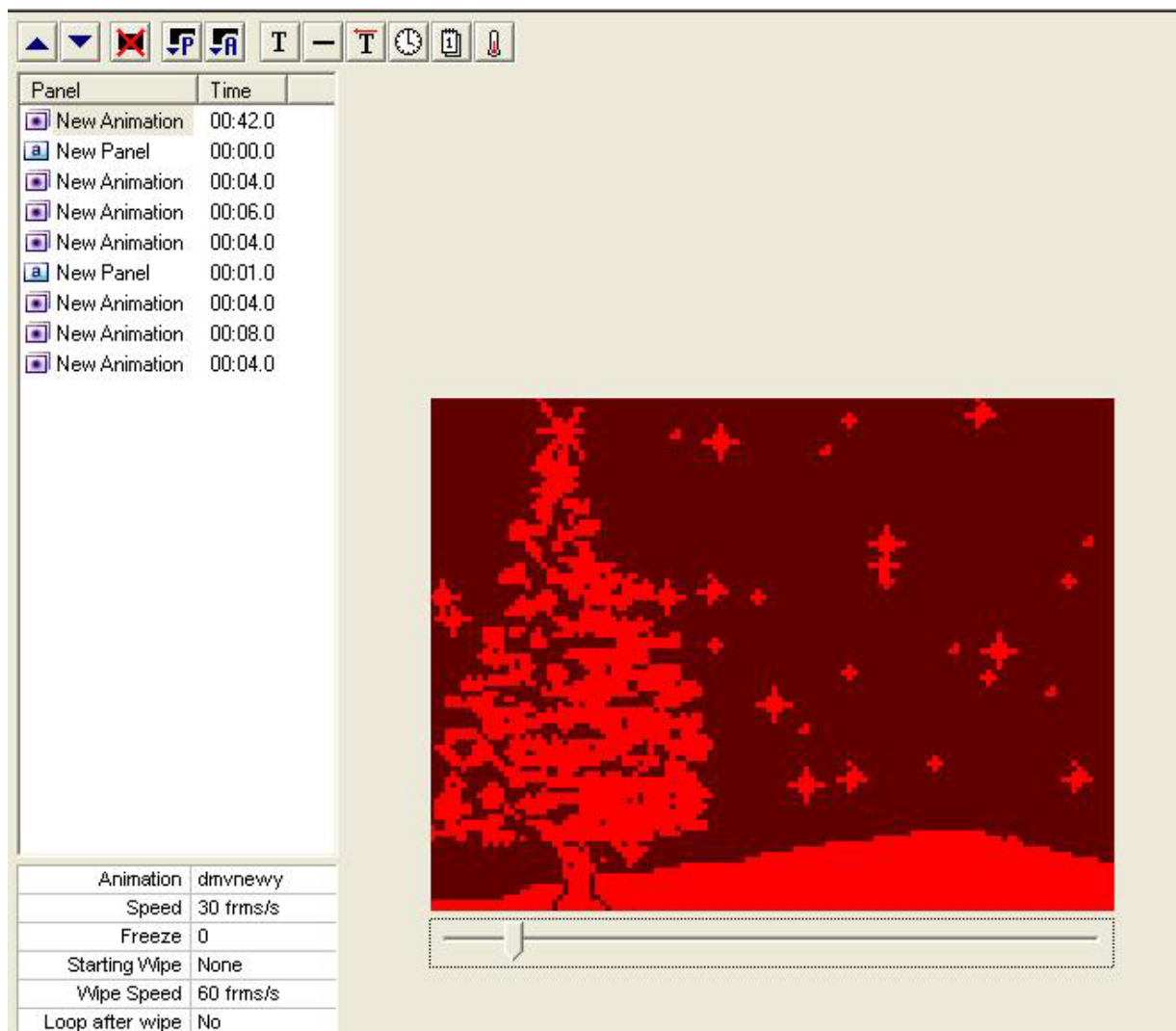
You can import Windows font, also (option **File/Import/Windows Font**).

Now, lets import first client's firm logo bitmap. Go to **File/Import/Background** and dialog similar to one about fonts, will appear. Select wanted .bmp file and press **Open**. As a result we'll get a picture under **Backgrounds** section in the left side tree view.

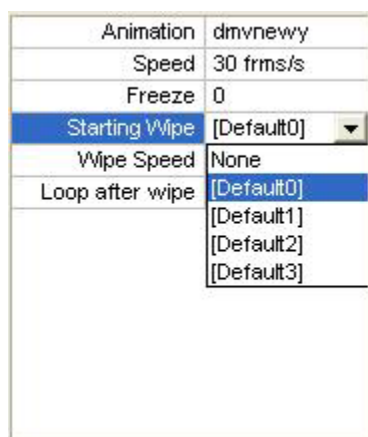
Open list, from the **Lists** section in the tree view on the left, which we want to edit. Right panel displays toolbar palettes and list editor.

Use toolbar to add textual panel (4th button from left). Set **Background** property to **Yes**, and specify background name. To add rolling text we need to click on the 5th button from the right. Set corresponding properties as it presented on the picture.

This made work about adding of first panel finished. Second panel is even easier than this. It does not require background. The only difference between the jobs, is that second text has to be displayed with large fonts. This can be accomplish easy (option **Zoom**, picture under)



Animation which we have in .avi file, we add to project in the same way as other contents, by **File\Import\Animation**. Then, go to list editor, click on the toolbar button labeled «A» (5th by left) and animation is added to the end of the list.



To add wipe between first and second panel, select **File\Import\Wipes**, and select one we want from **File/Open** dialog.

Then go back to list editor and set **Starting Wipe** property (when we set this to second panel it means that 1st and 2nd will be separated with this wipe effect) in second panel.

Compile it (**List/Build**) and we can start with loading to display. Before loading, you can preview display content using **remote control**.





