UnicView AD v1.6.0
User Guide
1.0

Proculus Technologies Limited

12/20/2017

The contents of this document are subject to change without notice. All technical information in this document is for reference purposes only. System configurations and specifications in this document supersede all previous information received by the purchaser.

Proculus Technologies Limited makes no representations that this document is complete, accurate or error-free and assumes no responsibility and will not be liable for any errors, omissions, damage or loss that might result from any use of this document, even if the information in the document is followed properly.

This document is not part of any sales contract between Proculus Technologies Limited and a purchaser. This document shall in no way govern or modify any Terms and Conditions of Sale, which Terms and Conditions of Sale shall govern all conflicting information between the two documents.

For Research Use Only. Not for use in diagnostic procedures.

PATENT PENDING

Contents

1 Introduction

This section contains important information on how to read this document.

1.1 Document Overview

This document provides a general overview of UnicView AD features and utilities, instructions on how to use it, and descriptions of all functionalities. It assumes the user has basic prior knowledge about microcontroller or computer programming and binary and hexadecimal numeric representations.

If you are a new user of UnicView AD, we recommend reading this document sequentially, from start to end. However, this document's sections are structured to be used as a reference guide, where you can look for specific information on each subject quickly.

LCM Family

This document applies only to the **V Family** of Proculus LCMs.

1.2 Conventions Used in this Document

This section presents the textual conventions and notations used in this document. Knowing these conventions will make it easier to read this document.

1.2.1 Notations

Notation	Description	Examples
(button+…), ([button/button])	Keyboard or Mouse button or button combination. Buttons inside square brackets represent multiple options.	(CTRL+A) = *Control + A* (ALT+LMB) = *Alt + Left Mouse Button* (ESC) = *Escape* (SHIFT+[UP/DOWN]) = *(SHIFT+UP) or (SHIFT+DOWN)*
0x0000	Number in hexadecimal notation. The number of digit pairs indicates the number of bytes considered.	0x0064, 0x01
0b00000000	Number in binary notation.	0b01001011, 0b11110000
(0,0), [0,0]	In value range context – represents a range of values. Rounded brackets symbols indicate the value is not included in the range, and square brackets indicate the value is included in the range.	(0,3) = *Range from 1 to 2.* [0,3) = *Range from 0 to 2.* (0,3] = *Range from 1 to 3.* [0,3] = *Range from 0 to 3.*
(0,0)	In coordinates context – represents a pair of values, usually coordinates.	(23, 172) = *Coordinates on the LCM screen.*
"X - Section Title"	Link to a section of this document.	Section "1.2 - Conventions Used in this Document".
<u>Section Title</u>	Link to a section of this document.	Insert a <u>Basic Touch</u>.
<FieldName>FixedText	Represents variable and fixed text fields.	<FileName>_AD.bin = *could be Controls_AD.bin, Images_AD.bin, System_AD.bin, etc.*
VPC(0x####), *VP(0x####)	Refers to the value stored on a VP. Can be used in conjunction with C Language-style syntax.	VPC(0x0000) == 1 = *Read as "The content of VP 0x0000 equals 0".* *VP(x0030) + 3 = *Read as "The content of VP 0x0030 plus 3".* *VP(0x0111) = 0xFFFF = *Read as "The content of VP 0x0111 receives (becomes) the value 0x0111".*

1.2.2 Information, Caution and Warning Statements

This document may contain Information, Caution and Warning statements.

ℹ	Info	This is an Information statement. It draws attention to certain key aspects about the current topic.
⚠	Caution	This is a Caution statement. It describes a situation that could potentially damage your software, equipment or cause data loss.
⚠	Warning	This is a Warning statement. It describes a situation that could potentially cause harm or injury to you.

The information in Caution and Warning statements is provided for your protection. Read each Caution and Warning statement carefully.

2 Software Overview

UnicView AD (Advanced Design) is a GUI Designer (Graphical User Interface Designer) used to create the configuration project for AD LCMs. It provides an intuitive and easy-to-use environment to create Human-Machine Graphical Interfaces very quickly. It provides a great set of configurable pre-made functions (keyboard, slider, numeric displays, etc.), shrinking GUI development time as the developer no longer needs to worry about low-level graphical management. Instead, they will just drag and drop the Interface Objects, configure their parameters, and the LCM will communicate in a simple 5-command protocol with the external controller.

AD LCMs are applicable on most systems that need a GUI, as AD firmware is very simple to use and design for. The set of tools it provides can handle 90% of real-world scenarios. Only when the application requires greater low-level control a more complex solution is needed (UnicView RD).

2.1 Minimum Requirements

- 1.5 GHz Processor
- 1 GB RAM
- Windows 7® or above, 64 bits
- Microsoft® .NET Framework 4.0
- Microsoft® Visual C++ Redistributable x64 2013
- Available Internet connection for licensing (Serial Key)

2.2 Installation and Licensing

UnicView AD installer is provided as a RAR package.

⚠	Caution	Fully extract the RAR archive (UnicView_AD_x.x.x.rar) before running setup.exe.

⚠	Caution	Uninstall any previously installed UnicView AD version before installing a new one.

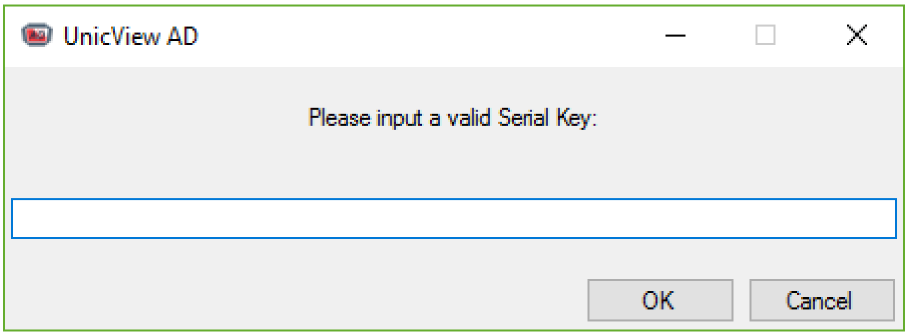
Extract the **UnicView_AD_x.x.x.rar** archive, and run **setup.exe**. Follow the instructions shown on the screen. After the installation is complete, run UnicView AD. A splash screen is shown while UnicView AD is initializing:

Initializing UnicView AD...

ℹ	Info	The installer may ask for Administrator rights before finishing the installation process.

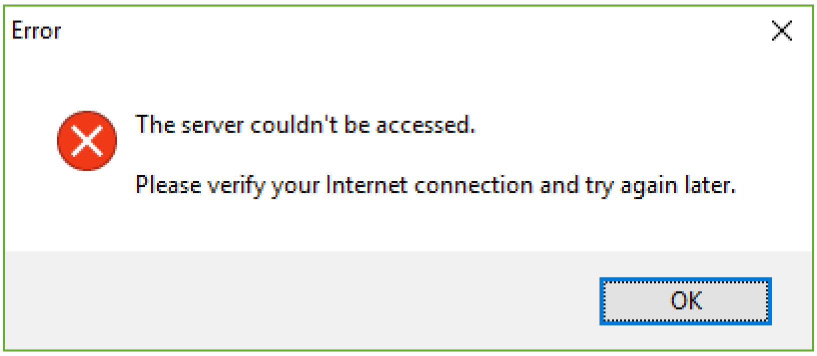
ℹ Info During the first application run, UnicView AD may ask for Administrator rights.

Once the application starts, it will ask for a Serial Key, to complete the registering process:



Please input a valid Serial Key:

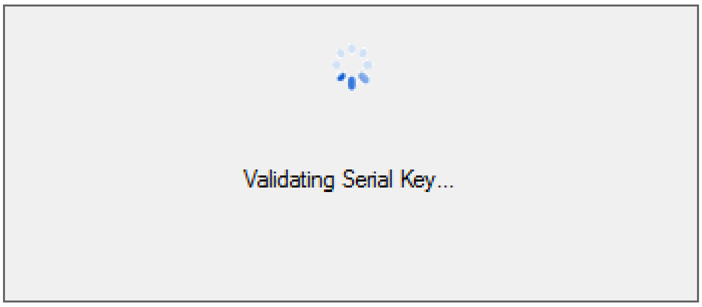
This process requires an available Internet connection. If a working connection is not detected (or our servers can’t be accessed by any reason), the following dialog will be shown:



The server couldn't be accessed.

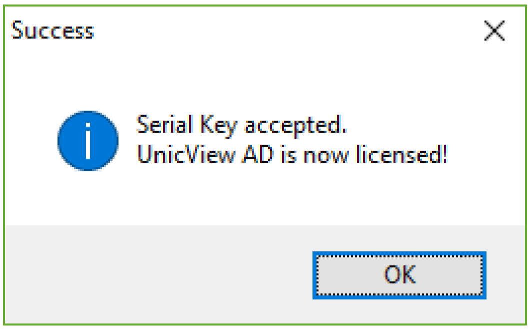
Please verify your Internet connection and try again later.

After inputting your Serial Key, it will be validated by our servers.

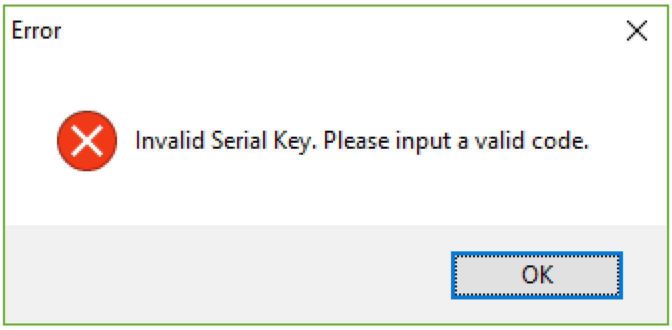


Validating Serial Key...

The result of the validation will be displayed either on a **Success** or **Error** dialog:



Serial Key accepted.
UnicView AD is now licensed!



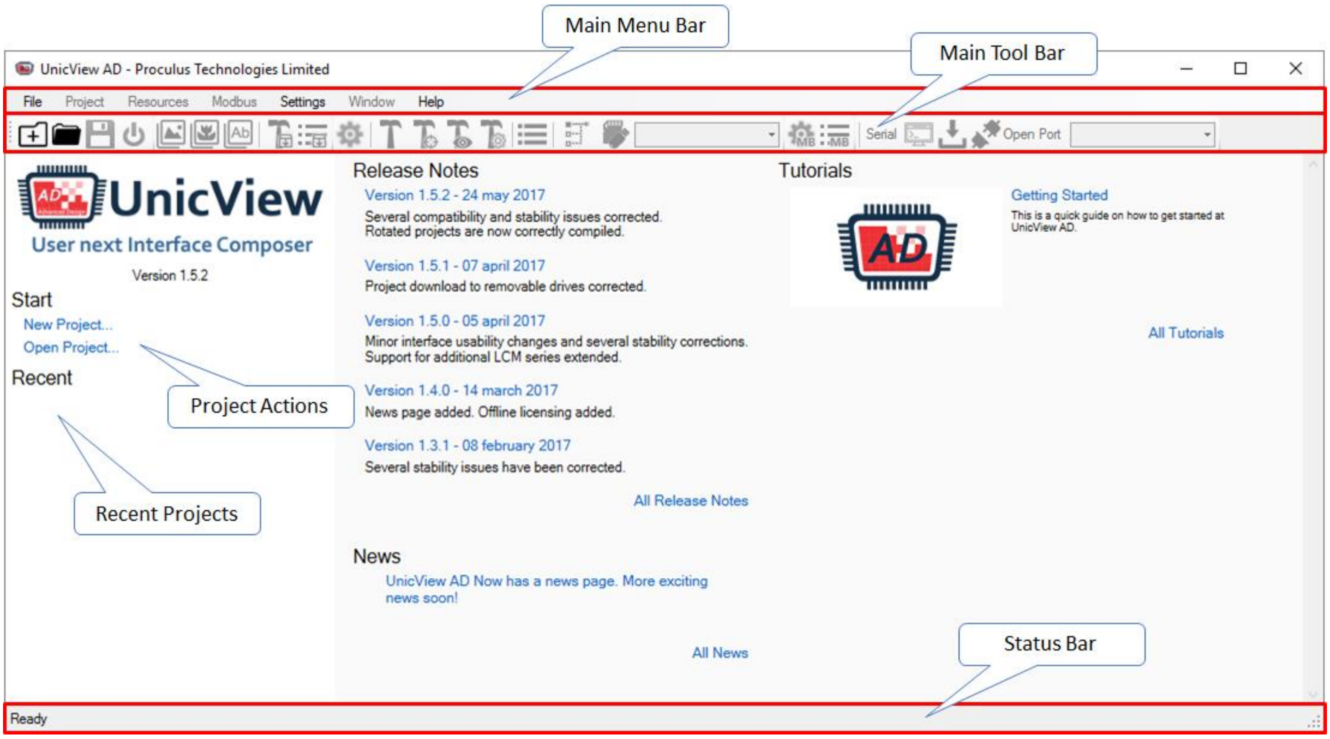
Invalid Serial Key. Please input a valid code.

2.3 Non-personal Information Collection

In the event of an application crash or otherwise exceptional occurrence, UnicView AD gathers non-personal data about the error, and automatically sends it to Proculus Technologies databases. This information is only for bug tracking and correction, and no personal information is retrieved.

2.4 Main Window

This is the main working area of UnicView AD, shown when the application starts. At this point, there are no projects open, as seen on the **Status Bar**, and only application-wide menus are enabled. The **News** panel is visible, providing the latest news (updated every time the application is opened) and project actions.



Release Notes

Version 1.5.2 - 24 may 2017
Several compatibility and stability issues corrected.
Rotated projects are now correctly compiled.

Version 1.5.1 - 07 april 2017
Project download to removable drives corrected.

Version 1.5.0 - 05 april 2017
Minor interface usability changes and several stability corrections.
Support for additional LCM series extended.

Version 1.4.0 - 14 march 2017
News page added. Offline licensing added.

Version 1.3.1 - 08 february 2017
Several stability issues have been corrected.

All Release Notes

Tutorials

Getting Started
This is a quick guide on how to get started at UnicView AD.

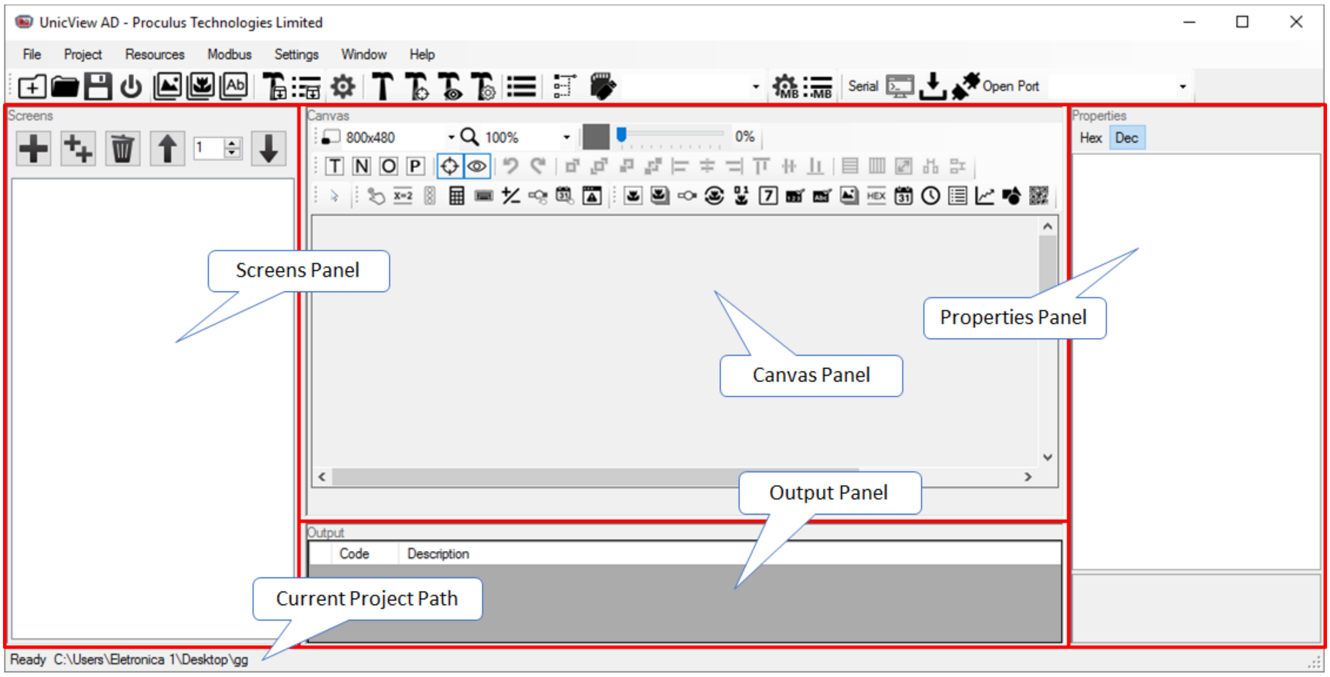
All Tutorials

News

UnicView AD Now has a news page. More exciting news soon!

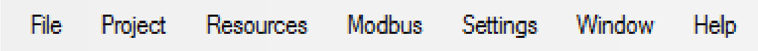
All News

When a project is opened, all menus and **panels** will be visible and enabled. The **Status Bar** will display the current project's path.



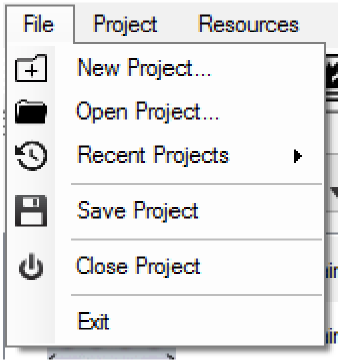
2.4.1 Main menu bar

The Main menu bar is located at the top of the **Main** window.



2.4.1.1 File menu

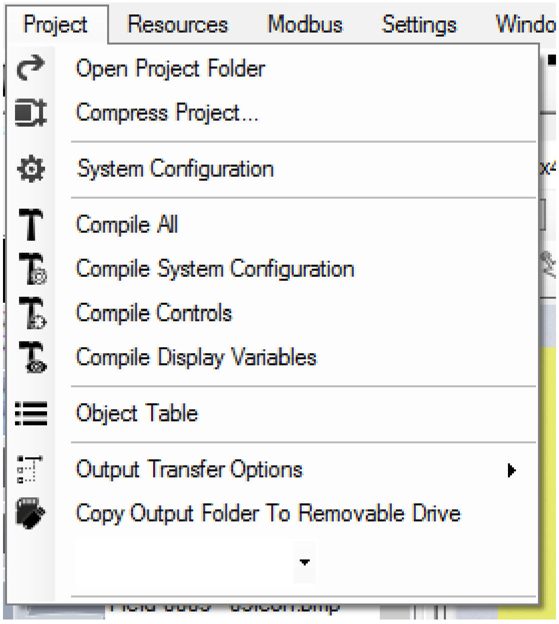
Project opening, saving and closing functions.



Item	Description
New Project...	Creates a new project
Open Project...	Opens an existing project
Recent Projects	Opens an existing project, from a recently opened projects list
Save Project	Saves the current project (overwriting)
Close Project	Closes the current project
Exit	Closes the current project and exits the application

2.4.1.2 Project menu

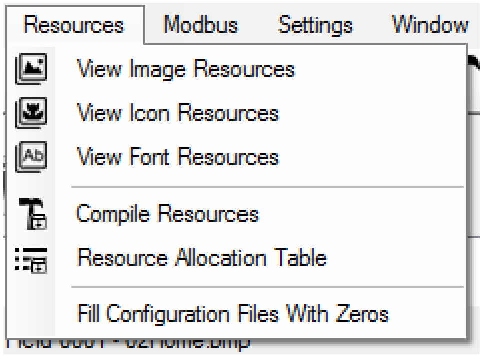
General project options and functions.



Item	Description
Open Project Folder	Opens the current project folder
Compress Project…	Packs the current project files into a compressed file
System Configuration	Opens the System Configuration window
Compile All	Compiles all files
Compile System Configuration	Compiles the System Configuration file (currently disabled)
Compile Controls	Compiles the Control Configuration file (currently disabled)
Compile Display Variables	Compiles the Display Variable Configuration file (currently disabled)
Object Table	Opens the Object Table window
Output Transfer Options	Selects which files should be transferred to the LCM
Copy Output Folder To Removable Drive	Copies the contents of the Output folder to the removable drive, be it an SD Card, Flash Drive or USB LCM, according to **Output Transfer Options** (above). The drive is selected on the Dropdown List below this button

2.4.1.3 Resources menu

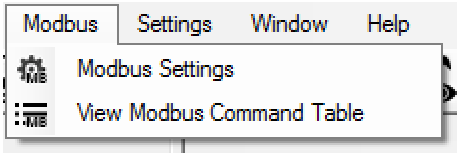
Resource management functions for the current project.



Item	Description
View Image Resources	Opens the Image Resources window
View Icon Resources	Opens the Icon Resources window
View Font Resources	Opens the Font Resources window
Compile Resources	Compiles all the resource files (currently disabled)
Resource Allocation Table	Opens the Resource Allocation Table window
Fill Configuration Files With Zeroes	Sets the filling method of configuration files (not available on V family)

2.4.1.4 Modbus menu

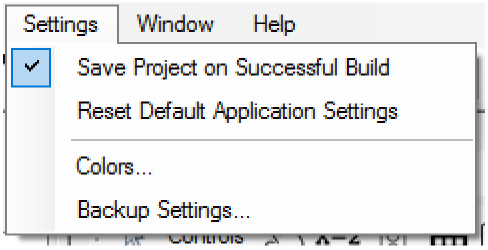
Modbus communication protocol settings.



Item	Description
Modbus Settings	Opens the Modbus configuration window
View Modbus Command Table	Opens the Modbus Command Table

2.4.1.5 Settings menu

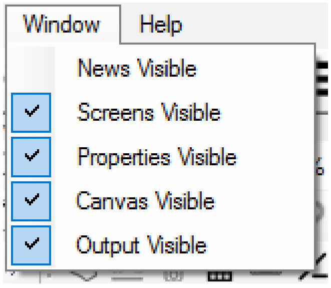
Application-wide settings.


Settings
Window
Help
✓ Save Project on Successful Build
Reset Default Application Settings
Colors...
Backup Settings...

Item	Description
Save Project on Successful Build	Selects if the project is saved after a successful build
Reset Default Application Settings	Resets all application-wide settings to their default values
Colors…	Color options for the application
Backup Settings…	Automatic project backup settings

2.4.1.6 Window menu

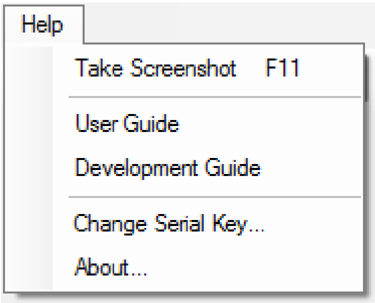
Window and panel visibility options.



Item	Description
News Visible	Sets the visibility of the News Panel
Screens Visible	Sets the visibility of the Screens panel
Properties Visible	Sets the visibility of the Properties panel
Canvas Visible	Sets the visibility of the Canvas panel
Output Visible	Sets the visibility of the Output panel

2.4.1.7 Help menu

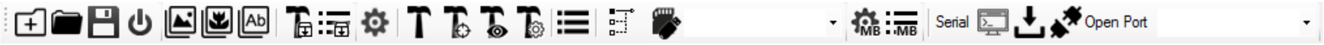
Comprises documentation files and additional software information.



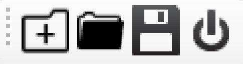
Item	Description
Take Screenshot	Takes a screenshot of the application
User Guide	Opens the User Guide document (this document)
Development Guide	Opens the Development Guide document, containing information about the AD firmware
Change Serial Key…	Opens a dialog to change the current serial key registered
About…	Opens version and contact information about UnicView AD

2.4.2 Main tool bar

The Main tool bar is located right below the Main menu bar. It contains most of the tools presented on the tools menu, as well as the condensed serial port controls.

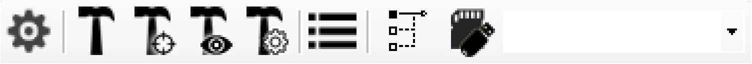


2.4.2.1 File bar



Item	Description
⊞	Creates a new project
📁	Opens an existing project
💾	Saves the current project
⏻	Closes the current project

2.4.2.2 Project bar



Item	Description
⚙	Opens the System Configuration window
🔨	Compiles all files
🔨⊕	Compiles the Control Configuration file
🔨👁	Compiles the Display Variable Configuration file
🔨⚙	Compiles the System Configuration file
☰	Opens the Object Table window
⋮	Selects which files should be transferred to the LCM
💾	Copies the contents of the Output folder to the removable drive, be it an SD Card, Flash Drive or USB LCM, according to **Output Transfer Options** (above). The drive is selected on the Dropdown List below this button

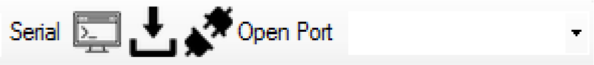
2.4.2.3 Modbus bar

Item	Description
⚙MB	Opens the Modbus configuration window
☰MB	Opens the Modbus Command Table

2.4.2.4 Resources bar

Item	Description
🖼	Opens the Image Resources window
🖼	Opens the Icon Resources window
Ab	Opens the Font Resources window
🔨	Compiles all the resource files
☰	Opens the Resource Allocation Table window

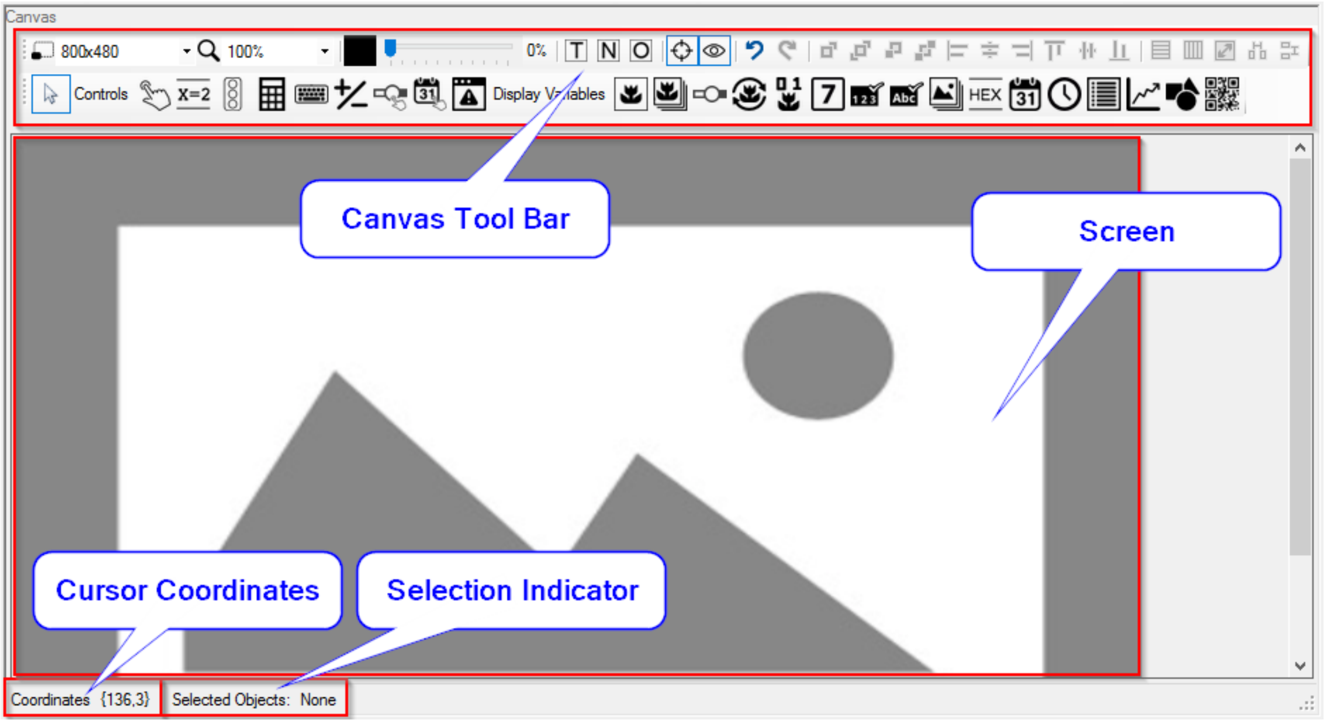
2.4.2.5 Serial Port bar



Item	Description
>_	Opens the Serial Terminal
⤓	Sends the contents of the Output folder to the LCM via Serial Communication, according to **Output Transfer Options**. **Currently not supported**
Open Port Close Port	Opens or closes the Serial Port. The Port is selected on the Dropdown List beside this button

2.5 Canvas Panel

The Canvas panel is the Screen composition region. It represents what each Screen on the LCM will be like. This section details each area of the Canvas panel.

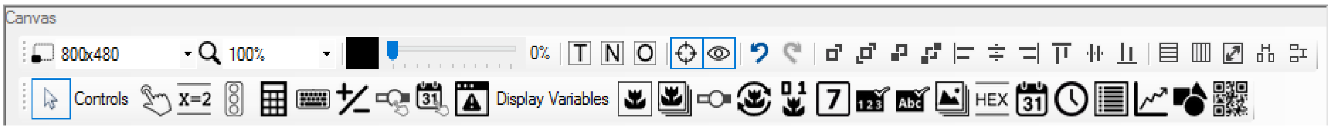


2.5.1 Screen area

This is the area that shows a preview of each LCM Screen. Here you'll layout the Controls and Display Variables and see the Background Image. Right clicking **(RMB)** on the Screen area will open a context menu.

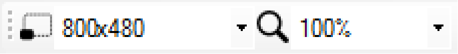
2.5.2 Canvas tool bar

The Canvas tool bar contains the necessary tools to make Screen Composition (the creation of each Screen on the LCM).



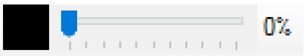
2.5.2.1 Screen view bar

The Screen view bar controls the Canvas size and zoom.



2.5.2.2 Screen effects bar

The Screen effects bar contains background image graphic effect controls.

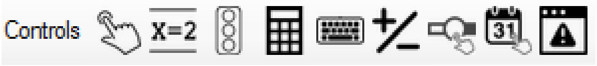


2.5.2.3 Cursor bar

Contains the Cursor button, used to select and move Interface Objects.

2.5.2.4 Controls bar

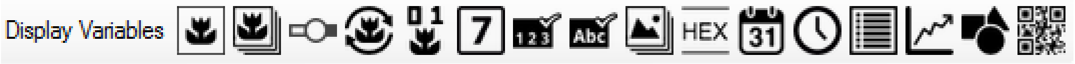
Contains all the Controls. If the selected firmware doesn't support a specific Control, its button will be disabled.



Detailed information about Controls can be found in section "4 - Interface Objects".

2.5.2.5 Display Variables bar

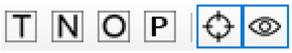
Contains all the Display Variables. If the selected firmware doesn't support a specific Display Variable, its button will be disabled.



Detailed information about Display Variables can be found in section "4 - Interface Objects".

2.5.2.6 Object Visibility bar

Controls the visibility options for Interface Objects on the Canvas.

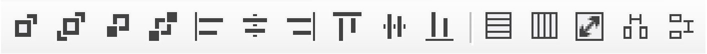


2.5.2.7 Operations bar

Contains buttons related to actions the user can perform on the Canvas panel. Currently, it contains the Undo and Redo actions.

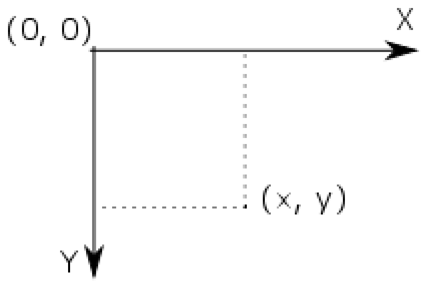
2.5.2.8 Object Layout bar

Contains all functions related to Interface Object positioning, ordering and sizing.



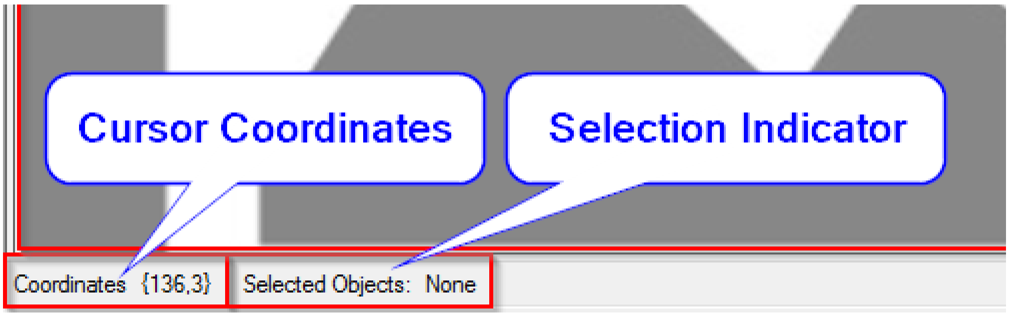
2.5.3 Cursor Coordinates

The current cursor coordinates are shown here. The origin (coordinates (0,0)) is located at the top-left corner of the Screen area. The X axis is positive to the right side of the origin, and the Y axis is positive to the bottom side of the origin, as shown on the following picture:



2.5.4 Selection Indicator

The currently selected Objects are listed here. The number of selected Controls and Display Variables are indicated separately. Only visible Objects can be selected. If a selected Object is no longer visible, it is deselected.

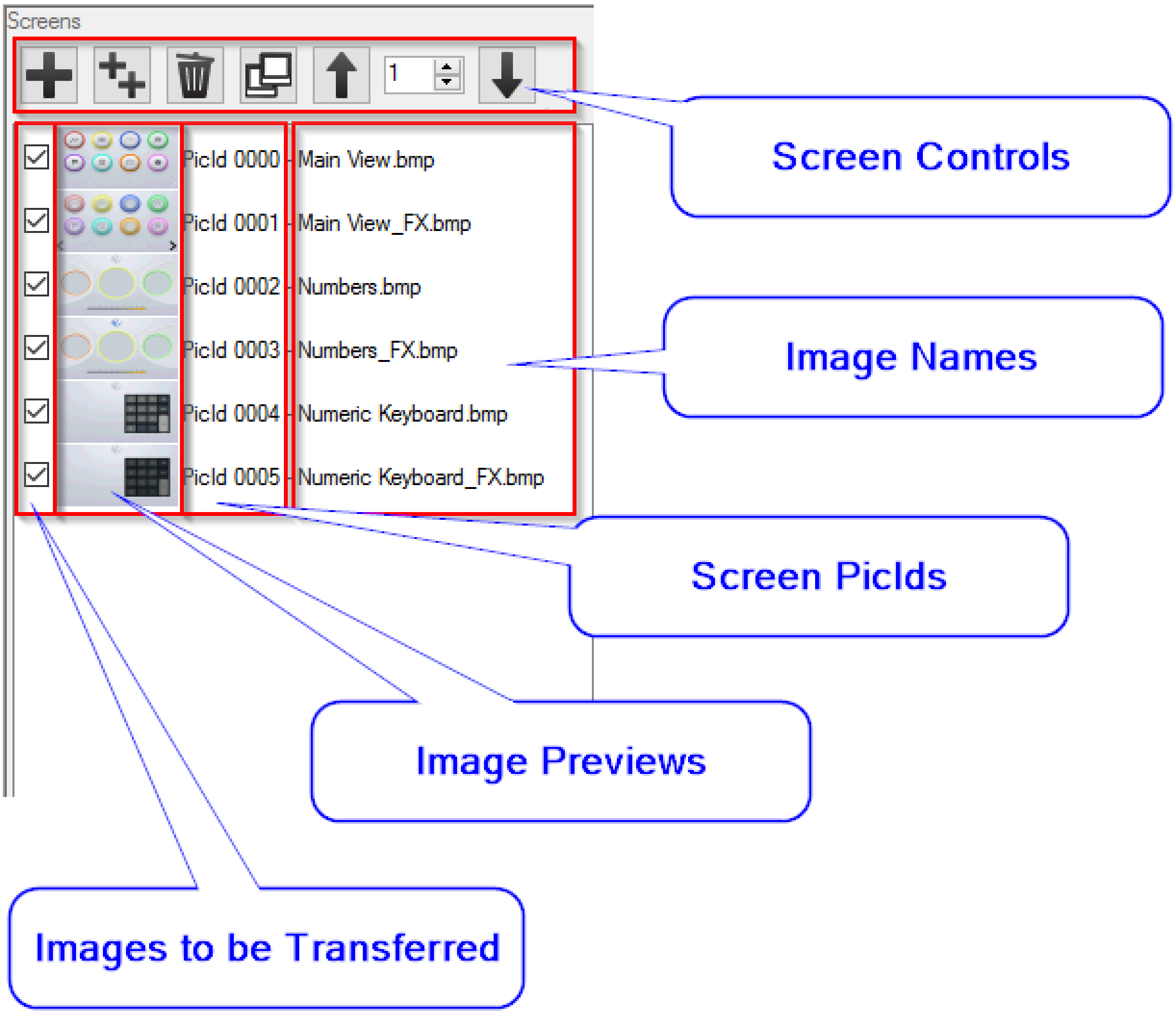


Coordinates {136,3}

Selected Objects: None

2.6 Screens Panel

The Screens panel shows all the Screens on the project. It’s also where you can add, remove, edit and reorder the Screens.


Screens
1
☑ PicId 0000 Main View.bmp
☑ PicId 0001 Main View_FX.bmp
☑ PicId 0002 Numbers.bmp
☑ PicId 0003 Numbers_FX.bmp
☑ PicId 0004 Numeric Keyboard.bmp
☑ PicId 0005 Numeric Keyboard_FX.bmp
Screen Controls
Image Names
Screen PicIds
Image Previews
Images to be Transferred

Info Multiple rows can be selected using **(CTRL+LMB)**, **(SHIFT+LMB)**, or **(SHIFT+[UP/DOWN])**.

The definition of Screens, Background Images and PicIds is given in section “2.11.7 - Screens”.

2.6.1 Adding and Removing Screens

To add a new Screen, click the ➕ button. To add multiple new Screens at once, click the ⁺₊, and choose how many Screens you want to add on the dialog window that will be shown.

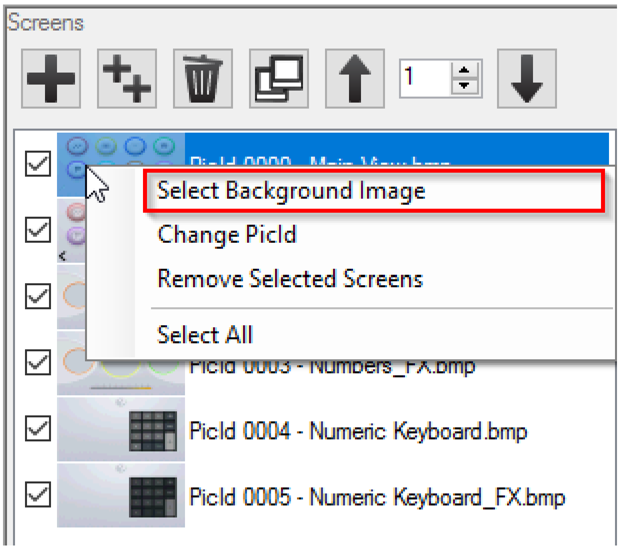
Caution It may take a few minutes to add large quantities of Screens at once.

To remove Screens, select one or more Screens and click the 🗑 button, or press **(DELETE)** on your keyboard.

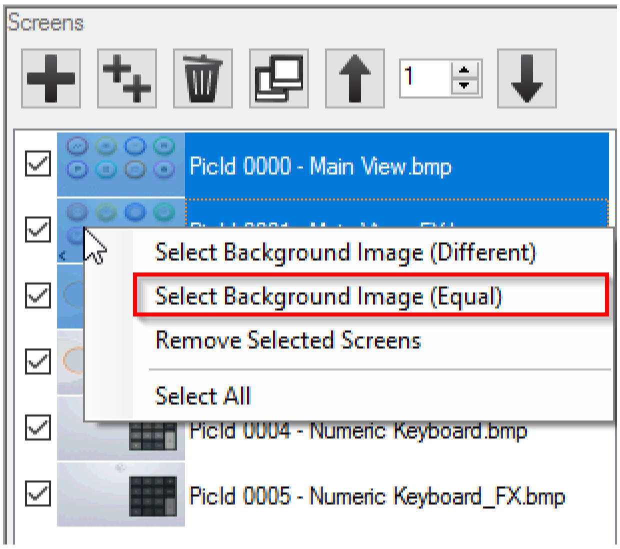
2.6.2 Assigning Background Images

There are a few ways to assign an Image Resource (refer to section "3.1 - Image Resources" for the definition of Image Resources) as the Background Image of a Screen:

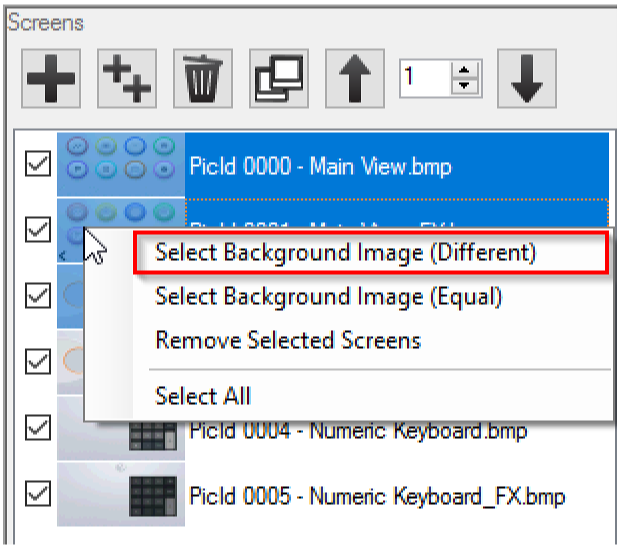
- Right click **(RMB)** a Screen, then click **"Select Background Image"**. Select the desired image on the Image Resources window that will be shown.


Screens
1
PicId 0000 - Main View.bmp
Select Background Image
Change PicId
Remove Selected Screens
Select All
PicId 0003 - Numbers_FX.bmp
PicId 0004 - Numeric Keyboard.bmp
PicId 0005 - Numeric Keyboard_FX.bmp

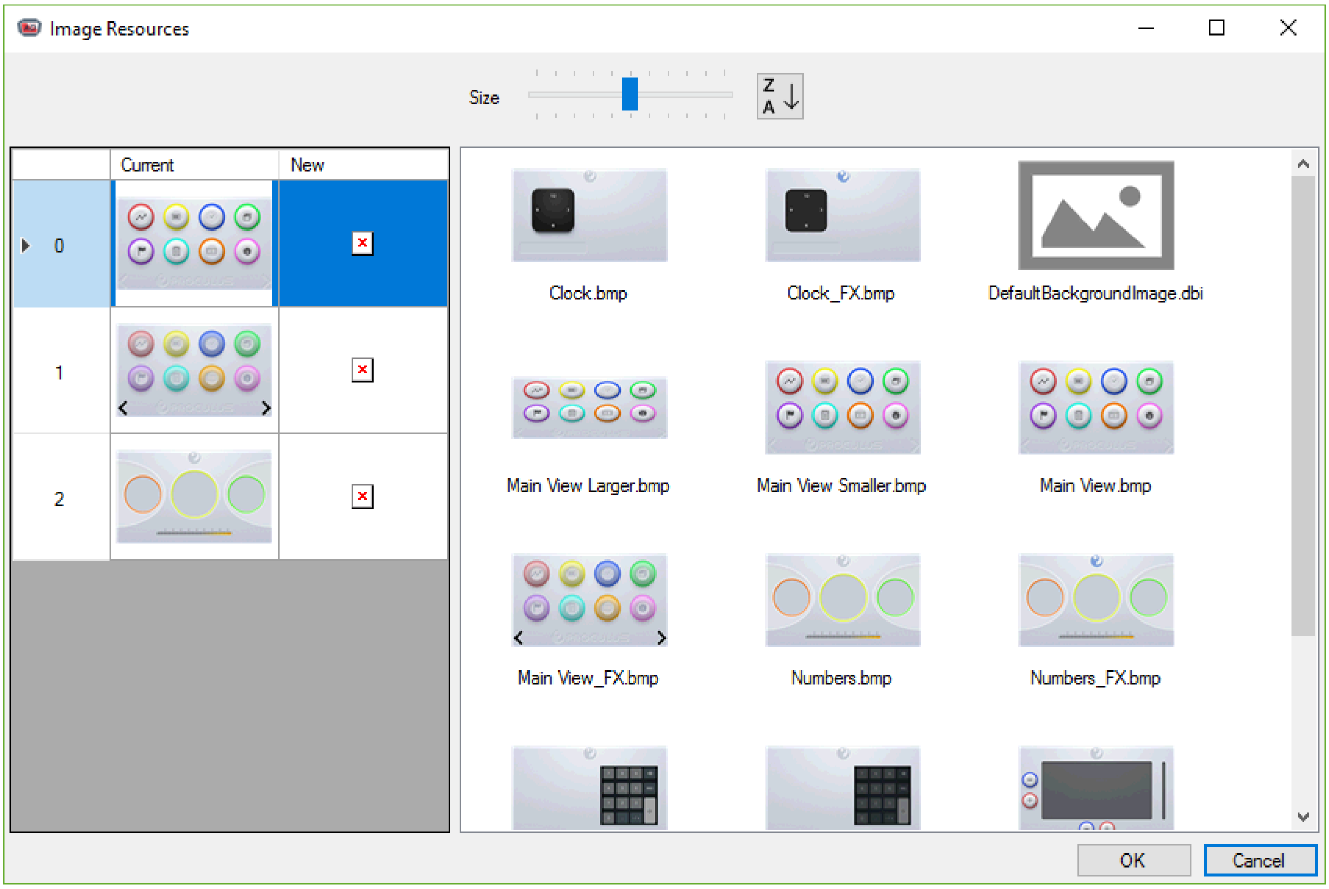
- Select multiple Screens, right click **(RMB)** one of them, then click **"Select Background Image (Equal)".** Select the desired image on the Image Resources window that will be shown. The same image will be assigned to all selected Screens.


Screens
1
PicId 0000 - Main View.bmp
PicId 0001 - Main View_FX.bmp
Select Background Image (Different)
Select Background Image (Equal)
Remove Selected Screens
Select All
PicId 0004 - Numeric Keyboard.bmp
PicId 0005 - Numeric Keyboard_FX.bmp

- Select multiple Screens, right click **(RMB)** one of them, then click **"Select Background Image (Different)".**


Screens
1
PicId 0000 - Main View.bmp
Select Background Image (Different)
Select Background Image (Equal)
Remove Selected Screens
Select All
PicId 0004 - Numeric Keyboard.bmp
PicId 0005 - Numeric Keyboard_FX.bmp

Another Image Resources window will be shown, where you can assign different Background Images to each Screen.



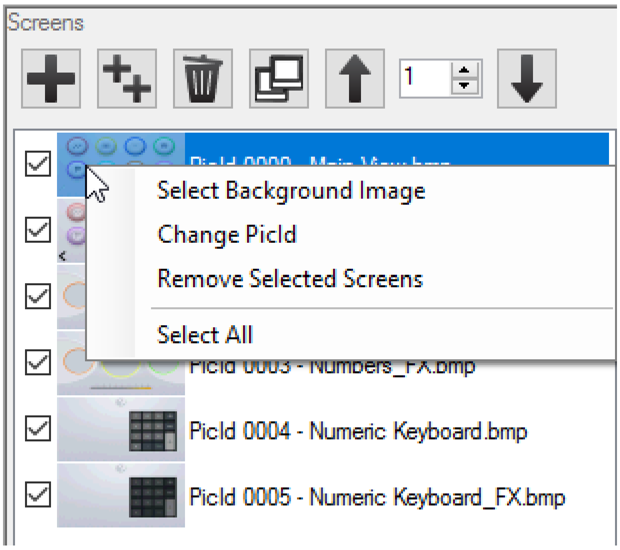
The assignment is made by dragging the images from the right-side panel to the Screen list on the left-side panel.

You can select multiple images at once, and they will be assigned on the same order as they appear on the right-side panel.

Multiple images can be selected using **(CTRL+LMB)**, **(SHIFT+LMB)**, or **(SHIFT+[UP/DOWN])**.

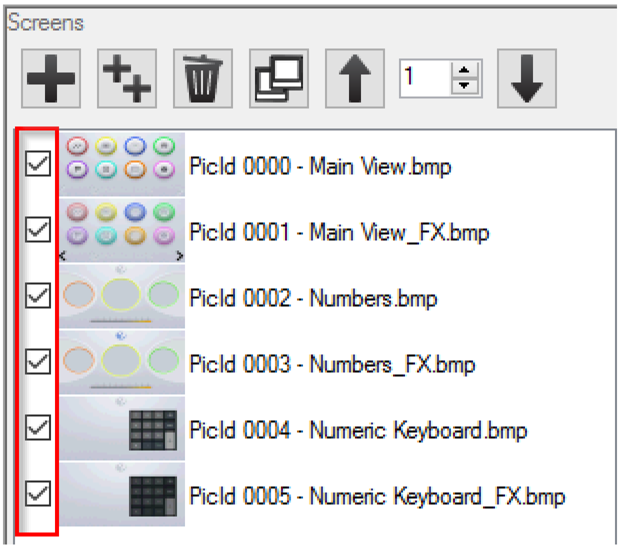
2.6.3 Changing PicId

To change the PicId of a single Screen, right click **(RMB)** the desired Screen, and click "Change PicId", then select the new PicId, which must be free (not allocated to another Screen). You may also select one or more Screens, then click the buttons ↑ to move up (decrease PicId) or ↓ to move down (increase PicId). The numeric control 1 indicates how many PicIds to ascend or descend.


Screens
1
PicId 0000 - Main View.bmp
Select Background Image
Change PicId
Remove Selected Screens
Select All
PicId 0003 - Numbers_FX.bmp
PicId 0004 - Numeric Keyboard.bmp
PicId 0005 - Numeric Keyboard_FX.bmp

2.6.4 Selecting Images to be Transferred

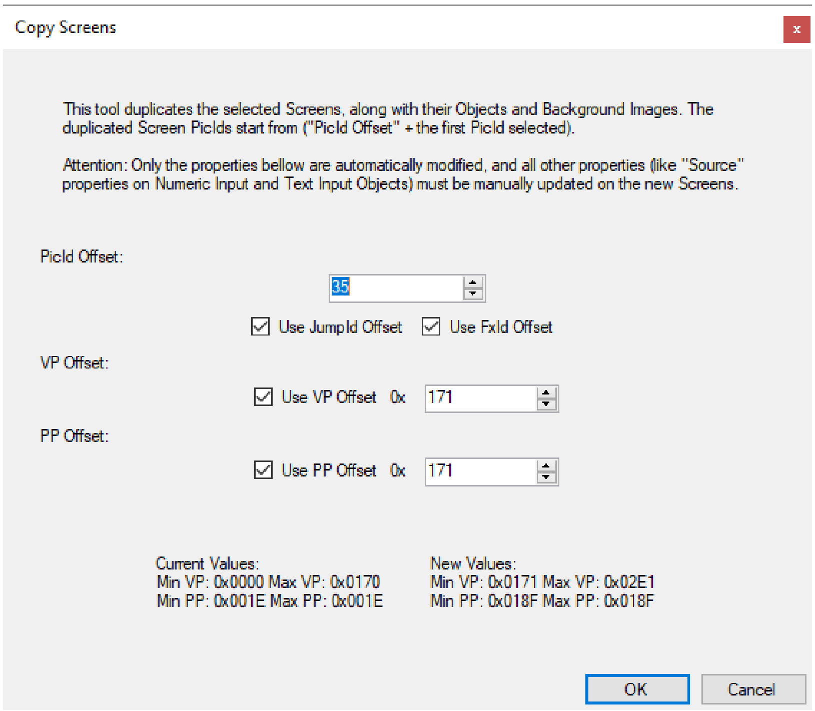
To select which Background Images should be transferred to the LCM (more details in section "2.11.9 - Download to LCM"), simply click the checkboxes of each Screen, on the left side of their Background Image previews. Only the Screens which have the checkbox checked will have their Background Images transferred to the LCM.


Screens
1
☑ PicId 0000 - Main View.bmp
☑ PicId 0001 - Main View_FX.bmp
☑ PicId 0002 - Numbers.bmp
☑ PicId 0003 - Numbers_FX.bmp
☑ PicId 0004 - Numeric Keyboard.bmp
☑ PicId 0005 - Numeric Keyboard_FX.bmp

Info — By default, all Screens are checked to be transferred.

2.6.5 Copying Screens

To copy (duplicate) Screens, first select all the Screens you want to copy, then click the (**Copy Screens)** button, select the offset settings, and accept. This will create copies of all selected Screens, along with their Background Images and Objects. The Objects' VP, PP, JumpId and FxId parameters will change as defined by the selected options on the Copy Screens dialog window, shown below:



Copy Screens

This tool duplicates the selected Screens, along with their Objects and Background Images. The duplicated Screen PicIds start from ("PicId Offset" + the first PicId selected).

Attention: Only the properties bellow are automatically modified, and all other properties (like "Source" properties on Numeric Input and Text Input Objects) must be manually updated on the new Screens.

PicId Offset: 35

☑ Use JumpId Offset ☑ Use FxId Offset

VP Offset:

☑ Use VP Offset 0x 171

PP Offset:

☑ Use PP Offset 0x 171

Current Values:
Min VP: 0x0000 Max VP: 0x0170
Min PP: 0x001E Max PP: 0x001E

New Values:
Min VP: 0x0171 Max VP: 0x02E1
Min PP: 0x018F Max PP: 0x018F

The dialog's options are described on the following table:

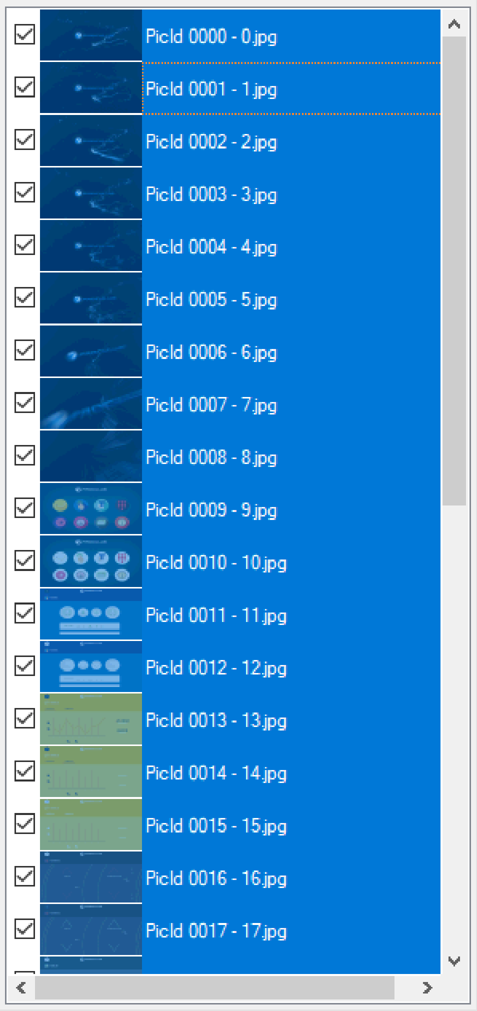
Option	Description	Notes
PicId Offset	Sets how many PicIds to add to each new Screen.	Its automatic minimum value guarantees that the new PicIds won't overlap with any of the current PicIds.
Use JumpId Offset	Sets if the **PicId Offset** value should also be applied to all **JumpId** properties on the Screens' Objects.	
Use FxId Offset	Sets if the **PicId Offset** value should also be applied to all **FxId** properties on the Screens' Objects.	
VP Offset	Sets how many VPs to add to the VP property of the Screens' Objects.	Default value: 1 + (Max VP on the project).
Use VP Offset	Sets if the **VP Offset** value should be applied.	
PP Offset	Sets how many PPs to add to the PP property of the Screens' Objects.	Default value: 1 + (Max VP on the project). All disabled PPs (0xFFFF) are ignored.
Use PP Offset	Sets if the **PP Offset** value should be applied.	

Caution

Only **PicId**, **VP** and **PP** properties are **automatically** changed. All other reference properties must be **manually** changed (**Source PicId** property on Numeric Input, for example).

For example, consider a project with 35 Screens, with PicIds ranging from 0 to 34. The minimum and maximum VP and PP values on this range are as follows (0xFFFF values are ignored):

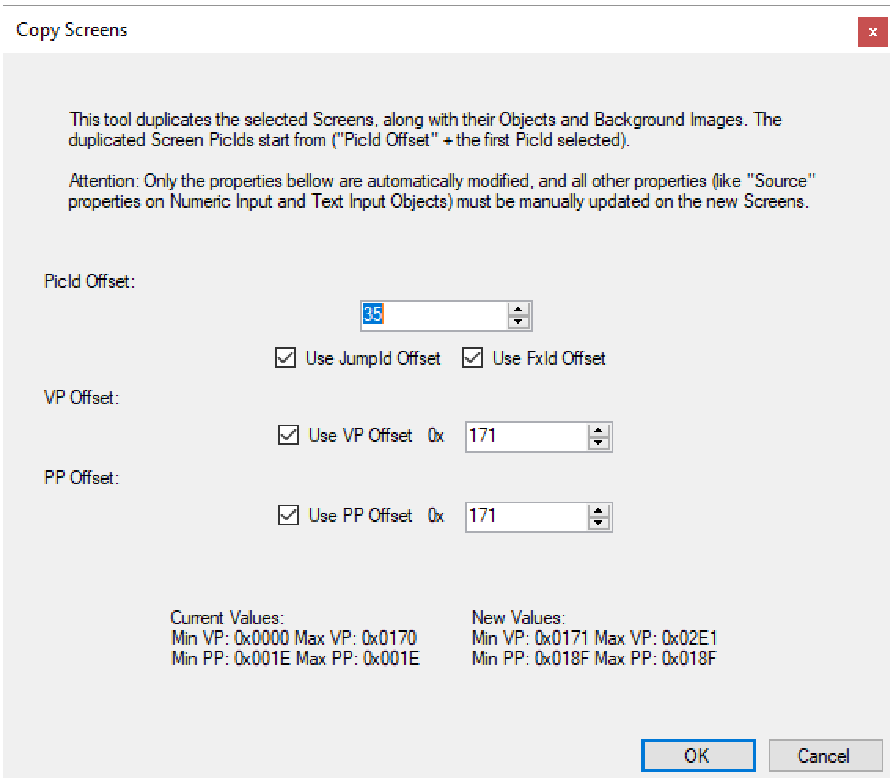
	VP	PP	PicId
Maximum	0x0170	0x001E	34
Minimum	0x0000	0x001E	0



☑ PicId 0000 - 0.jpg
☑ PicId 0001 - 1.jpg
☑ PicId 0002 - 2.jpg
☑ PicId 0003 - 3.jpg
☑ PicId 0004 - 4.jpg
☑ PicId 0005 - 5.jpg
☑ PicId 0006 - 6.jpg
☑ PicId 0007 - 7.jpg
☑ PicId 0008 - 8.jpg
☑ PicId 0009 - 9.jpg
☑ PicId 0010 - 10.jpg
☑ PicId 0011 - 11.jpg
☑ PicId 0012 - 12.jpg
☑ PicId 0013 - 13.jpg
☑ PicId 0014 - 14.jpg
☑ PicId 0015 - 15.jpg
☑ PicId 0016 - 16.jpg
☑ PicId 0017 - 17.jpg

Select all 35 Screens using **(CTRL+A)**, then click on the **Copy Screens** button.

The default PicId, VP and PP offset values can be used for quick duplication.



Copy Screens

This tool duplicates the selected Screens, along with their Objects and Background Images. The duplicated Screen PicIds start from ("PicId Offset" + the first PicId selected).

Attention: Only the properties bellow are automatically modified, and all other properties (like "Source" properties on Numeric Input and Text Input Objects) must be manually updated on the new Screens.

PicId Offset: 35

☑ Use JumpId Offset ☑ Use FxId Offset

VP Offset: ☑ Use VP Offset 0x 171

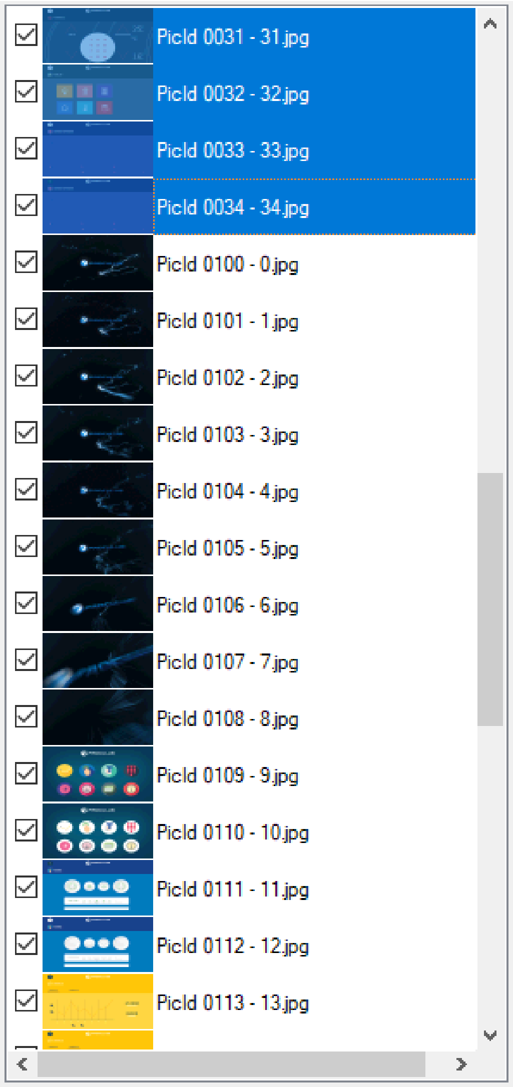
PP Offset: ☑ Use PP Offset 0x 171

Current Values:
Min VP: 0x0000 Max VP: 0x0170
Min PP: 0x001E Max PP: 0x001E

New Values:
Min VP: 0x0171 Max VP: 0x02E1
Min PP: 0x018F Max PP: 0x018F

By using the default values, the new PicId range is 35 to 69, and the minimum and maximum VP and PP values on this range are as follows (0xFFFF values are ignored):

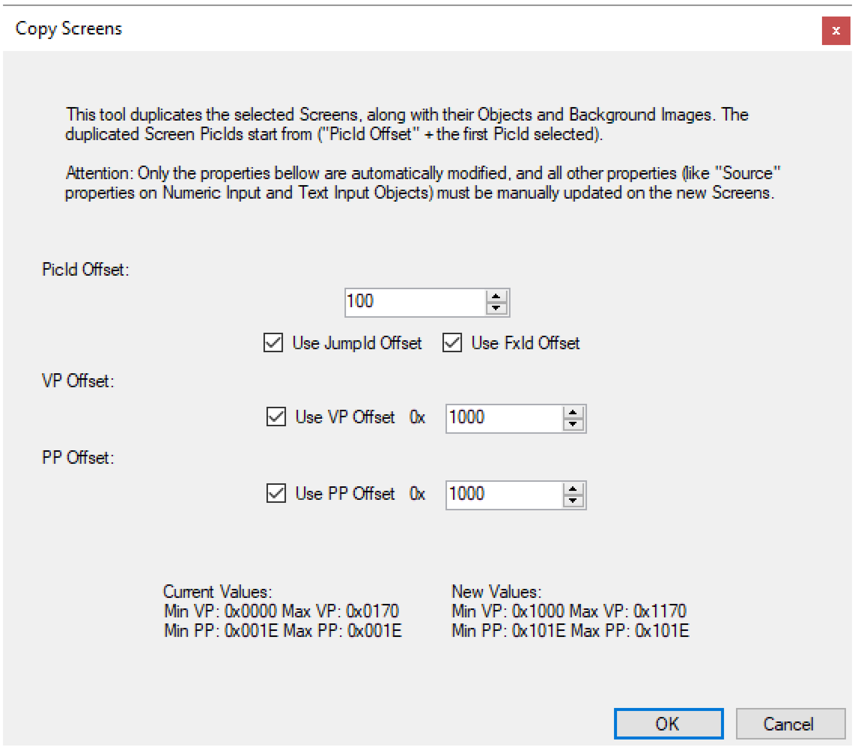
	VP	PP	PicId
Maximum	0x02E1	0x018F	69
Minimum	0x0171	0x018F	35



☑ PicId 0031 - 31.jpg
☑ PicId 0032 - 32.jpg
☑ PicId 0033 - 33.jpg
☑ PicId 0034 - 34.jpg
☑ PicId 0100 - 0.jpg
☑ PicId 0101 - 1.jpg
☑ PicId 0102 - 2.jpg
☑ PicId 0103 - 3.jpg
☑ PicId 0104 - 4.jpg
☑ PicId 0105 - 5.jpg
☑ PicId 0106 - 6.jpg
☑ PicId 0107 - 7.jpg
☑ PicId 0108 - 8.jpg
☑ PicId 0109 - 9.jpg
☑ PicId 0110 - 10.jpg
☑ PicId 0111 - 11.jpg
☑ PicId 0112 - 12.jpg
☑ PicId 0113 - 13.jpg

A common use for the Copy Screens tool is, when designing **multi-language** interfaces, to create all Screens in the **first language**, then **replicate** those Screens for the **other languages**.

In this case, it's usually recommended to use **large offset** values when possible, to make the project **clearly structured, separating the related Screen groups**.



Copy Screens

This tool duplicates the selected Screens, along with their Objects and Background Images. The duplicated Screen PicIds start from ("PicId Offset" + the first PicId selected).

Attention: Only the properties bellow are automatically modified, and all other properties (like "Source" properties on Numeric Input and Text Input Objects) must be manually updated on the new Screens.

PicId Offset: 100

☑ Use JumpId Offset ☑ Use FxId Offset

VP Offset: ☑ Use VP Offset 0x 1000

PP Offset: ☑ Use PP Offset 0x 1000

Current Values:
Min VP: 0x0000 Max VP: 0x0170
Min PP: 0x001E Max PP: 0x001E

New Values:
Min VP: 0x1000 Max VP: 0x1170
Min PP: 0x101E Max PP: 0x101E

Using PicId **Offset = 100** and **VP Offset = PP Offset = 0x1000**, as shown above, for example, keeps everything neatly separated, and all value relations are **immediately recognizable**.

By using the values above, the new PicId range is 100 to 134, and the minimum and maximum VP and PP values on this range are as follows (0xFFFF values are ignored):

	VP	PP	PicId
Maximum	0x1170	0x101E	134
Minimum	0x1000	0x101E	100

It's very easy to notice the address relations when comparing the table above to the original table:

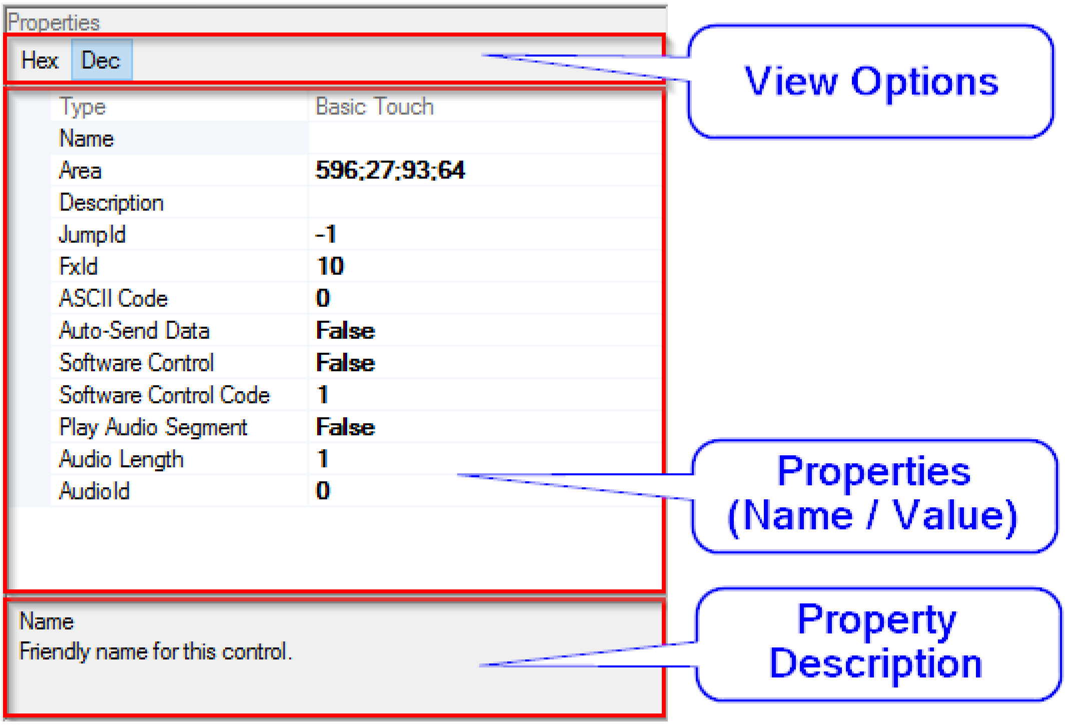
	VP	PP	PicId
Maximum	0x0170	0x001E	34
Minimum	0x0000	0x001E	0

2.7 Properties Panel

The Properties panel shows all the properties of the currently selected Interface Objects.

If multiple Objects of the same type are selected, the value of each property is shown only if they’re the same across all Objects. Otherwise, no value will be shown.

If multiple Objects of different types are selected, only the properties they have in common will be shown.



Type	Basic Touch
Name	
Area	596;27;93;64
Description	
JumpId	-1
FxId	10
ASCII Code	0
Auto-Send Data	False
Software Control	False
Software Control Code	1
Play Audio Segment	False
Audio Length	1
AudioId	0

Name
Friendly name for this control.

At the top of the panel are located the view options, where you can set the display format to decimal or hexadecimal.

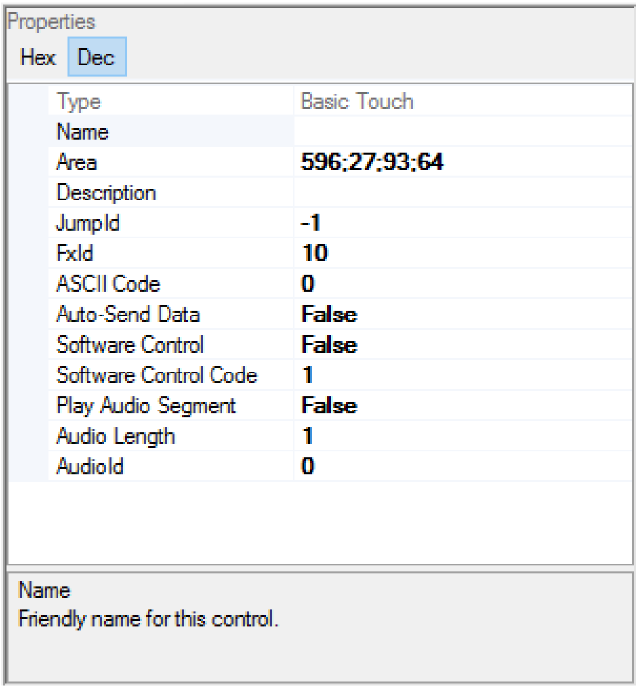
A brief description of each property is available at the bottom of the panel.

2.7.1 Numeric Representation

Most integer values can be typed in both decimal and hexadecimal representations, and are automatically converted to the selected view mode (Hex or Dec).

To type a value in decimal format, simply type normally. To type in a value in hexadecimal format, prefix the number with “0x”. The following images show the Parameters panel in both view modes.

Some properties are represented by an ordered sequence of values. “Area” and “Coordinates”, for instance, are represented as “X;Y;W;H” and “X;Y”, respectively, where X and Y are top-left coordinates, and W and H are Width and Height.

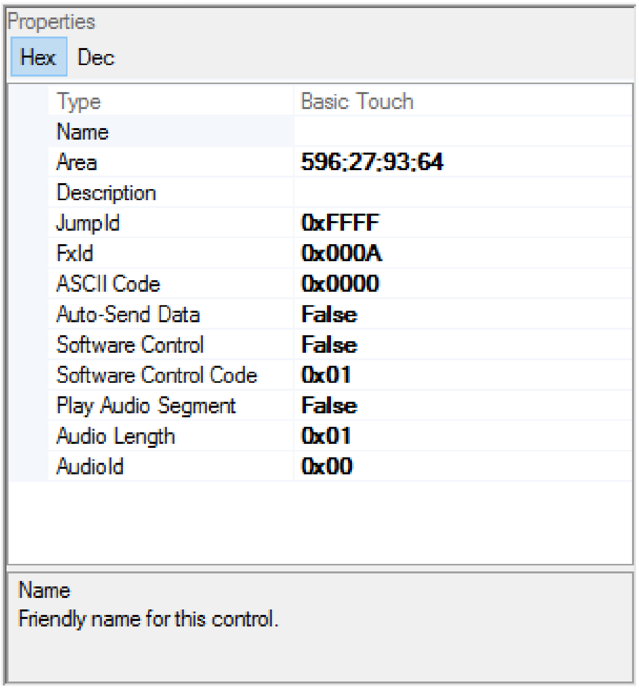


Properties

Hex | Dec

Type	Basic Touch
Name	
Area	596;27;93;64
Description	
JumpId	-1
FxId	10
ASCII Code	0
Auto-Send Data	False
Software Control	False
Software Control Code	1
Play Audio Segment	False
Audio Length	1
AudioId	0

Name
Friendly name for this control.



Properties

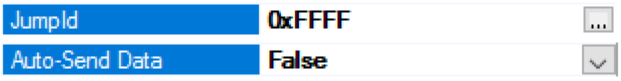
Hex | Dec

Type	Basic Touch
Name	
Area	596;27;93;64
Description	
JumpId	0xFFFF
FxId	0x000A
ASCII Code	0x0000
Auto-Send Data	False
Software Control	False
Software Control Code	0x01
Play Audio Segment	False
Audio Length	0x01
AudioId	0x00

Name
Friendly name for this control.

2.7.2 Selector Windows, Dropdown Lists and Quick Editors

The user inputs most of the parameters manually, but some of them have a **Selector Window**, **Dropdown List** or a **Quick Editor** to choose from predefined values. When a parameter has one of those elements, they display an icon on the right side of the value column when clicked:



- JumpId | 0xFFFF | ... Selector Window
- Auto-Send Data | False | ⌄ Dropdown List or Quick Editor

A **Dropdown List** is a list that shows predetermined values for you to choose from. It ranges from Boolean values (True or False) to Library selection and Value Types options.

A **Selector Window** is a window that provides visual editing tools, making it easier to change the property values than by doing that manually. Examples of Selector Windows are the PicId selector, Coordinate selector, and Color selector.

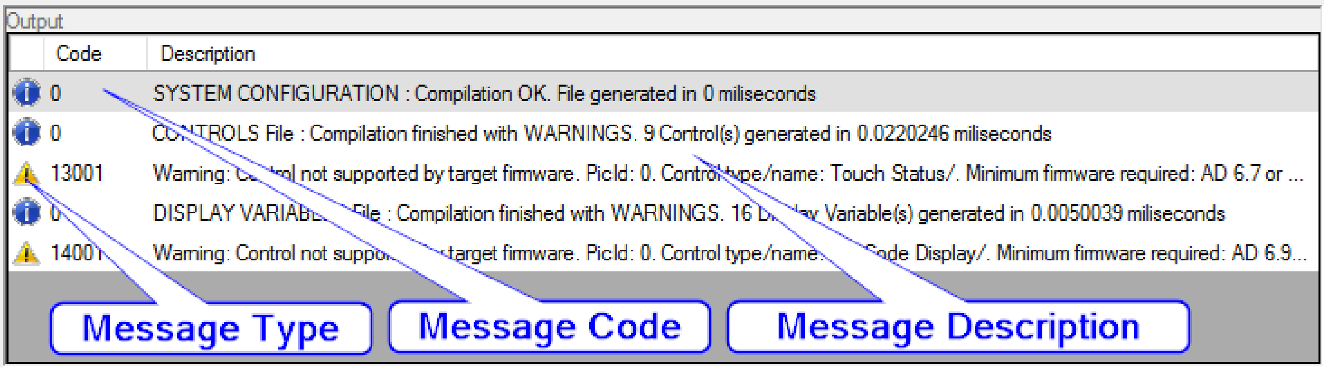
A **Quick Editor** lies between the other two elements. It functions as a Dropdown List, in the sense that it opens right below the property value, and not a new window, providing a quick editing interface, but also functions like a Selector Window, as it provides a more complex editing tool.

Info Section "2.9 - Selector Windows and Quick Editors" contains a list of the available Selector Windows and Quick Editors

2.8 Output Panel

Shows the compilation (or build) result messages. A message has three fields:

- Type: Indicates the severity of the message
- Code: Indicates a reference code that uniquely identifies the message
- Description: Details about the event



```
Output
   Code    Description
ⓘ  0       SYSTEM CONFIGURATION : Compilation OK. File generated in 0 miliseconds
ⓘ  0       CC..TROLS File : Compilation finished with WARNINGS. 9 Control(s) generated in 0.0220246 miliseconds
⚠  13001   Warning: Co..rol not supported by target firmware. PicId: 0. Control type/name: Touch Status/. Minimum firmware required: AD 6.7 or ...
ⓘ  0       DISPLAY VARIAB... File : Compilation finished with WARNINGS. 16 D...lay Variable(s) generated in 0.0050039 miliseconds
⚠  14001   Warning: Control not suppo... target firmware. PicId: 0. Control type/name: ...ode Display/. Minimum firmware required: AD 6.9...
```

There are three types of result messages:

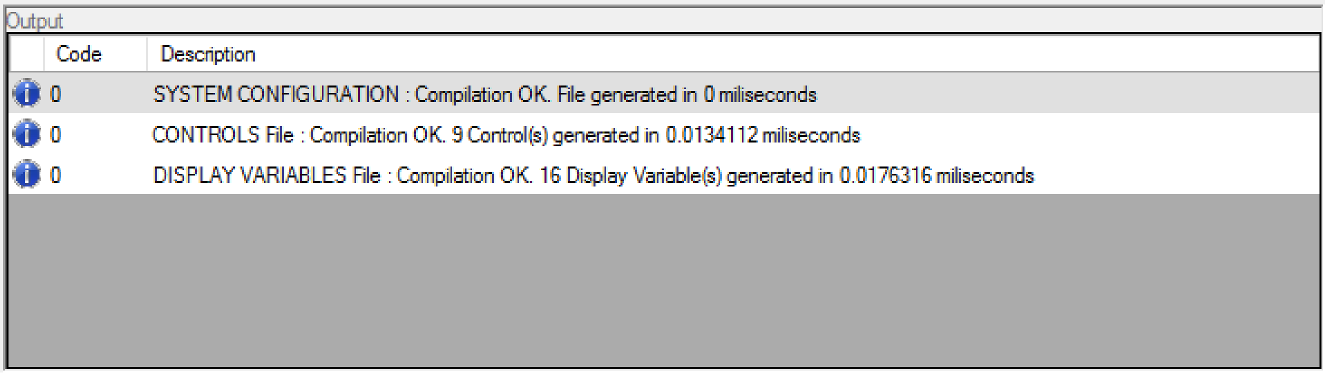
ⓘ Information: Indicates a successful or otherwise expected event.

⚠ Warning: Indicates that something requires attention. Occurs when the compilation was completed, but unexpected effects may result.

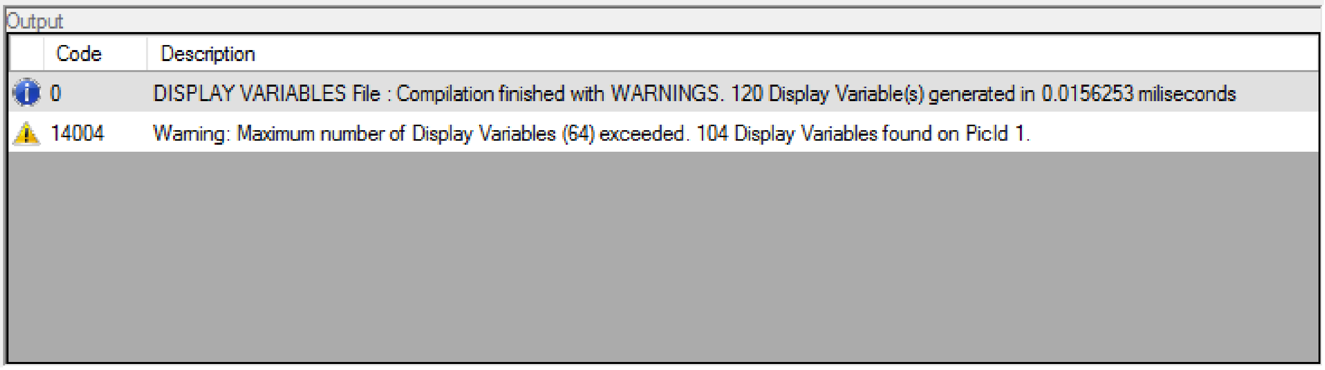
⊗ Error: Indicates that a specific build step was not successfully completed.

⚠	Caution	For the effect of automatic saving after building, a compilation with warnings is considered a successful one.

Examples of Output result messages:

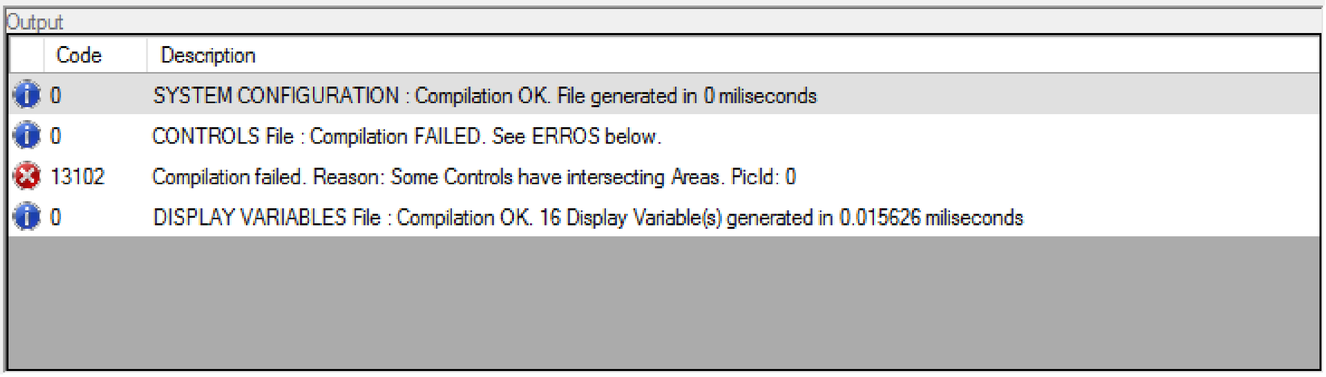


```
Output
   Code    Description
ⓘ  0       SYSTEM CONFIGURATION : Compilation OK. File generated in 0 miliseconds
ⓘ  0       CONTROLS File : Compilation OK. 9 Control(s) generated in 0.0134112 miliseconds
ⓘ  0       DISPLAY VARIABLES File : Compilation OK. 16 Display Variable(s) generated in 0.0176316 miliseconds
```



Output

	Code	Description
ℹ	0	DISPLAY VARIABLES File : Compilation finished with WARNINGS. 120 Display Variable(s) generated in 0.0156253 miliseconds
⚠	14004	Warning: Maximum number of Display Variables (64) exceeded. 104 Display Variables found on PicId 1.



Output

	Code	Description
ℹ	0	SYSTEM CONFIGURATION : Compilation OK. File generated in 0 miliseconds
ℹ	0	CONTROLS File : Compilation FAILED. See ERROS below.
✖	13102	Compilation failed. Reason: Some Controls have intersecting Areas. PicId: 0
ℹ	0	DISPLAY VARIABLES File : Compilation OK. 16 Display Variable(s) generated in 0.015626 miliseconds

ℹ	Info	Section “7.1 - Compilation Results” contains a complete list of result messages.

2.9 Selector Windows and Quick Editors

This section presents all Selector Windows and Quick Editors, which are visual aids designed to make it easier to change the properties of Interface Objects. All properties can still be manually changed on the Properties panel.

⚠ Caution

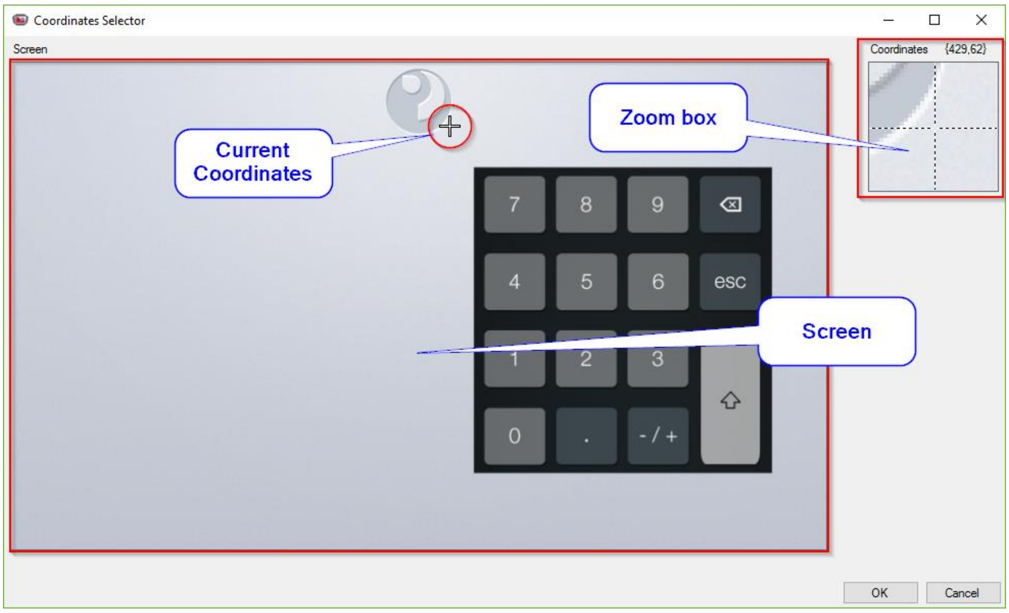
Always compile the project after changing any project resources.

If a **Selector Window**, **Quick Editor** or **Dropdown List** is missing some information, it's likely because the project has not been compiled since the last resource changes.

2.9.1 Coordinates Selector Window

This is the window used to select a point on the Screen (coordinates).

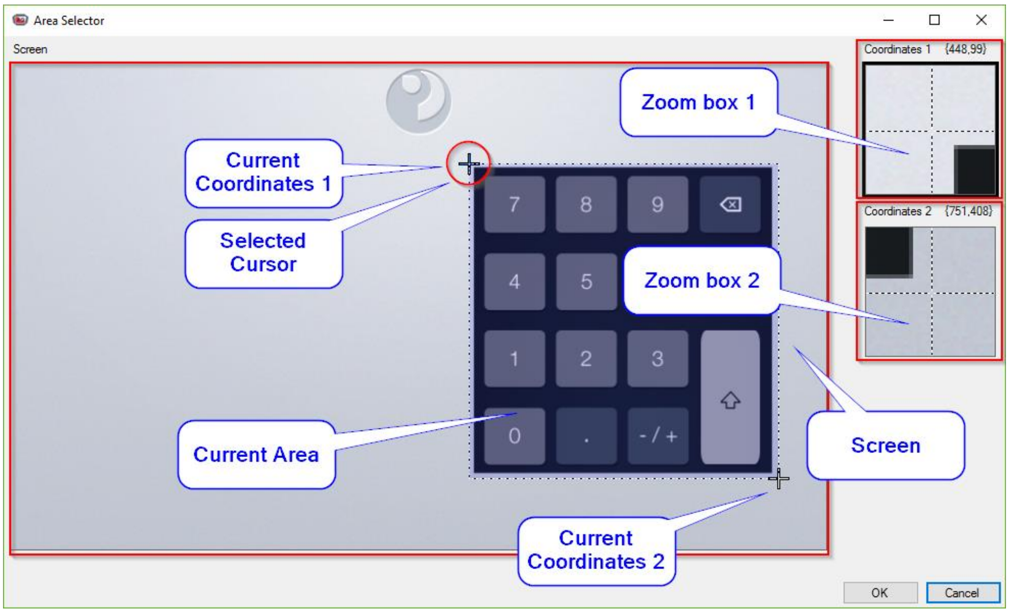
Use the mouse to select a point on the Screen. You can use the arrow keys on your keyboard to move the cursor pixel by pixel, for fine control. At the top-right corner of the window there's a zoom box and the coordinates selected, providing easier selection.



2.9.2 Area Selector Window

This is the window used to select a reference area on a Screen.

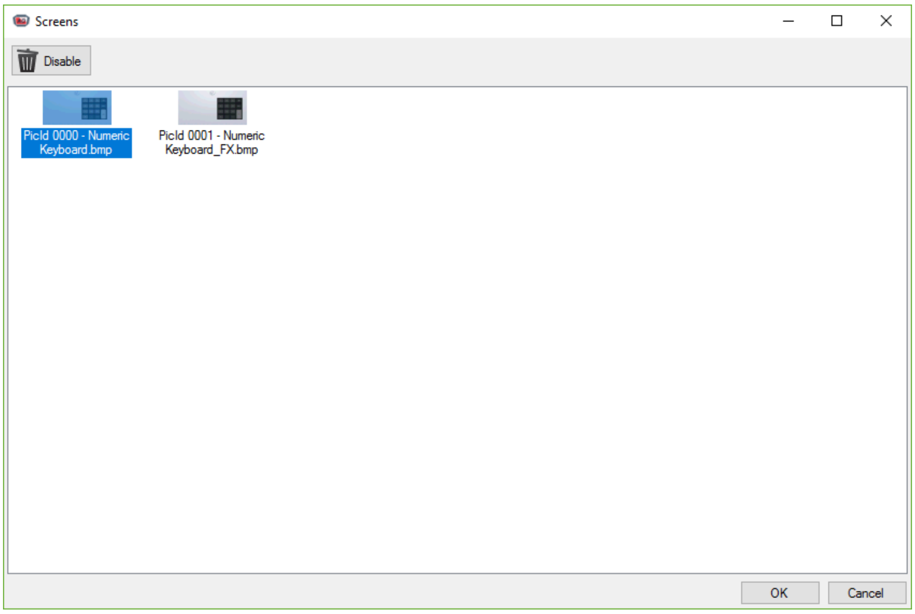
It behaves the same as the **Coordinates Selector Window**, but you can click the zoom boxes to choose what cursor is currently being moved. The selected zoom box will show a black border around it, and the respective cursor will turn blue.



2.9.3 Screen Selector Window

This is the window used to select the PicId of a Screen.

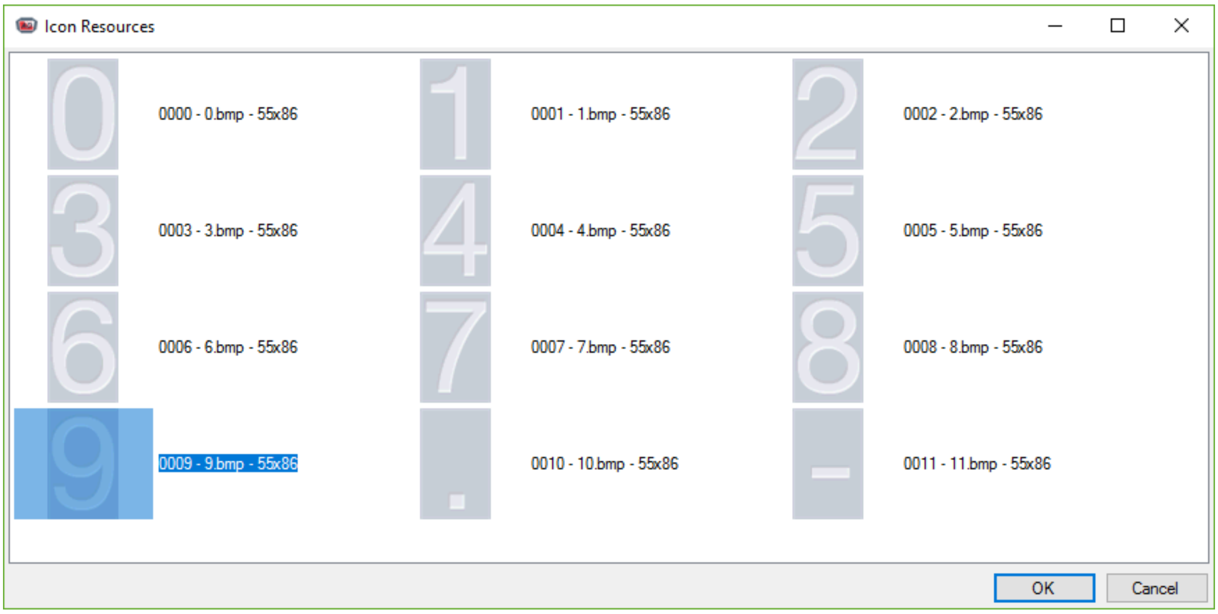
You can select a Screen from all Screens available on the project. Click the Disable button to deselect (set the PicId value to 0) when applicable (like the “**JumpId**” and “**FxId**” properties).


Screens
Disable
PicId 0000 - Numeric Keyboard.bmp
PicId 0001 - Numeric Keyboard_FX.bmp
OK
Cancel

2.9.4 Icon Selector Window

This is the window used to select an Icon.

It shows all the icons on the current Icon Library selected for the concerned Interface Objects.

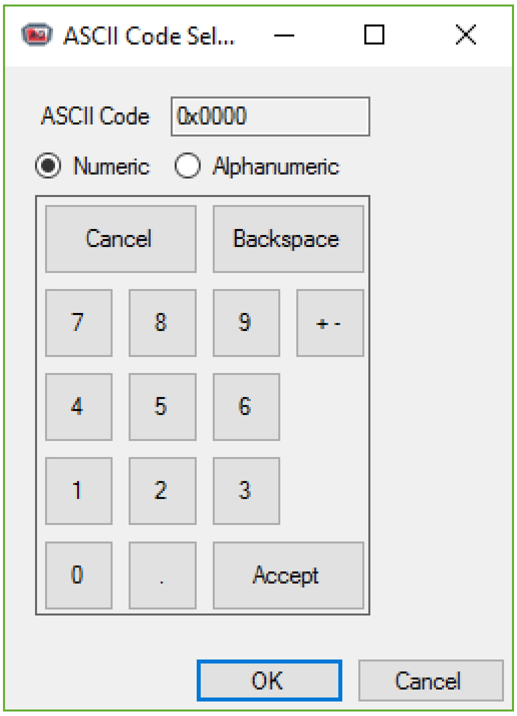


2.9.5 ASCII Code Selector Window

This is the window used to select an ASCII Code.

You can choose between two modes, numeric and alphanumeric, and the keyboard layout will change to reflect your choice.

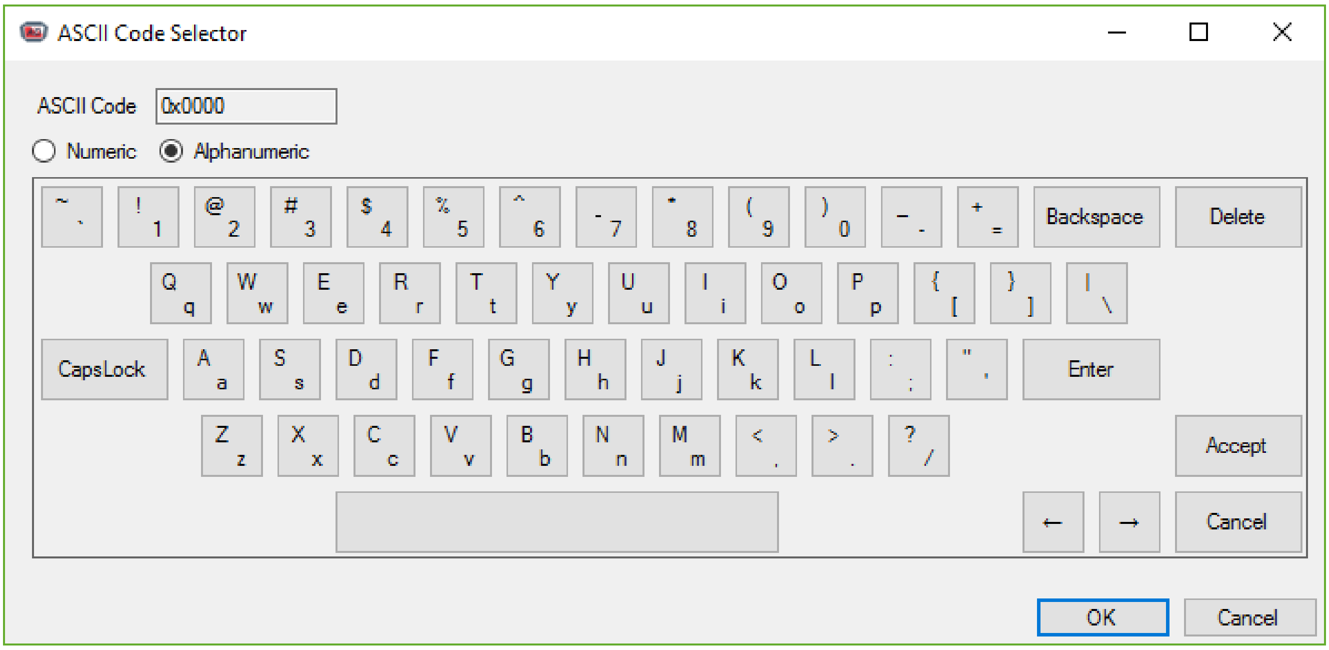
The **numeric** keyboard is typically used when designing Basic Touch Controls for Numeric Inputs and Popups.



ASCII Code: 0x0000

● Numeric ○ Alphanumeric

The **alphanumeric** keyboard is typically used when designing Basic Touch Controls for Text Inputs.



ASCII Code Selector

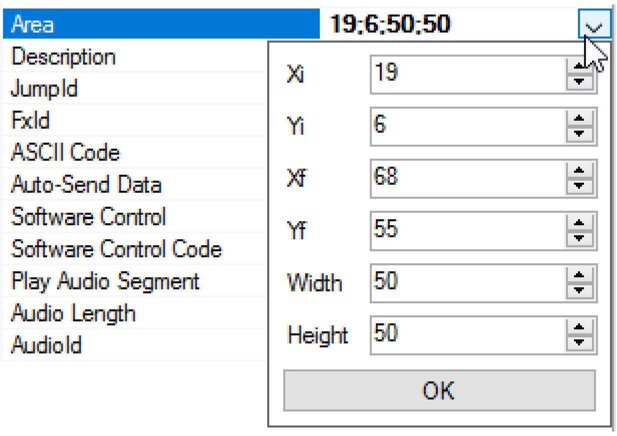
ASCII Code: 0x0000

○ Numeric ● Alphanumeric

~ ` ! 1 @ 2 # 3 $ 4 % 5 ^ 6 - 7 * 8 (9) 0 _ - + = Backspace Delete
Q q W w E e R r T t Y y U u I i O o P p { [}] | \
CapsLock A a S s D d F f G g H h J j K k L l : ; " ' Enter
Z z X x C c V v B b N n M m < , > . ? / Accept
← → Cancel
OK
Cancel

2.9.6 Area Quick Editor

This Editor is used to visually change the "**Area**" property of an Interface Object.



Area	19;6;50;50
Description	
JumpId	
FxId	
ASCII Code	
Auto-Send Data	
Software Control	
Software Control Code	
Play Audio Segment	
Audio Length	
AudioId	

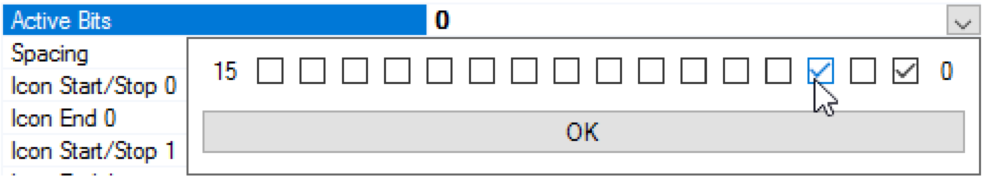
Xi	19
Yi	6
Xf	68
Yf	55
Width	50
Height	50

Values:

- Xi = Left horizontal coordinate (X Initial)
- Yi = Top vertical coordinate (Y Initial)
- Xf = Right horizontal coordinate (X Final)
- Yf = Bottom vertical coordinate (Y Final)
- Width = (Xf - Xi)
- Height = (Yf - Yi)

2.9.7 Bit Quick Editor

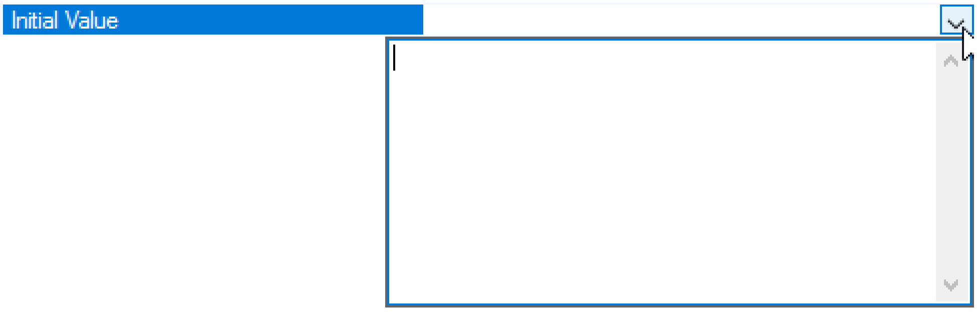
This Editor is used to select the functional bits on a Bitwise Icon.



Active Bits 0

2.9.8 Text Quick Editor

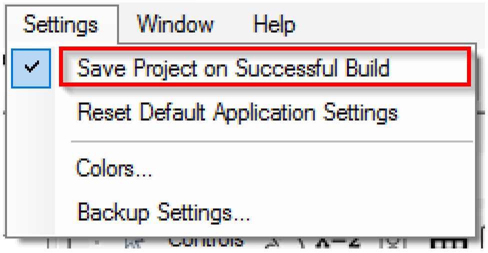
This Editor is used to input strings on properties that have text values.



2.10 Application Settings

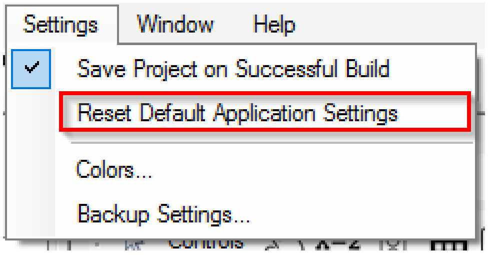
2.10.1 Save Project on Success Build

If this option is set, the project will be saved every time it is successfully compiled.


Settings
Window
Help
✓ Save Project on Successful Build
Reset Default Application Settings
Colors...
Backup Settings...

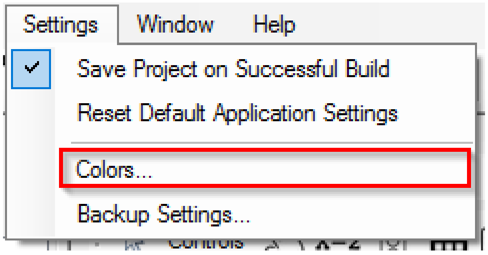
2.10.2 Reset Default Application Settings

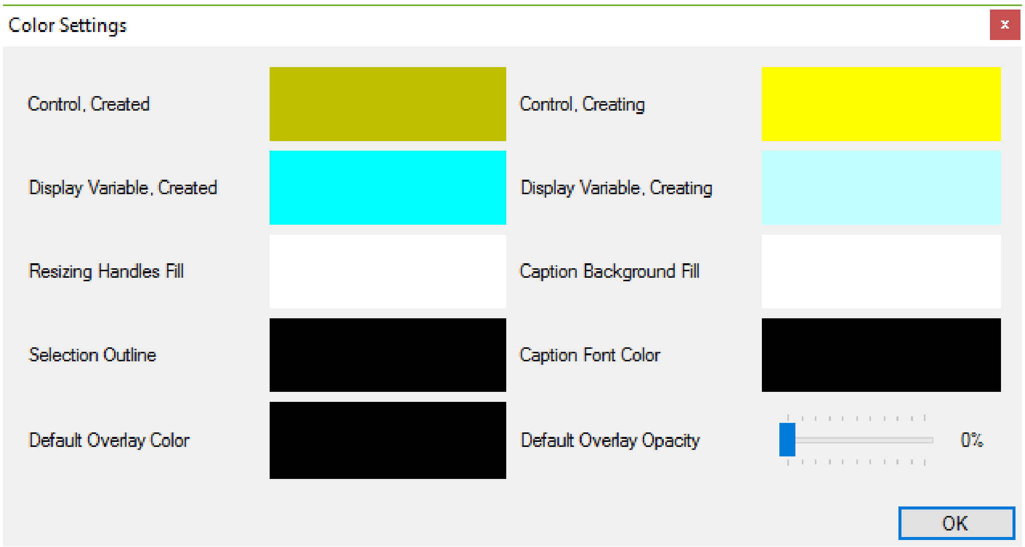
Clicking this button will reset all application settings to their default installation values. The project settings will not be modified.


Settings
Window
Help
✓ Save Project on Successful Build
Reset Default Application Settings
Colors...
Backup Settings...

2.10.3 Colors

Here you can change several color settings for the application


Settings
Window
Help
✓ Save Project on Successful Build
Reset Default Application Settings
Colors...
Backup Settings...



Control, Created

Control, Creating

Display Variable, Created

Display Variable, Creating

Resizing Handles Fill

Caption Background Fill

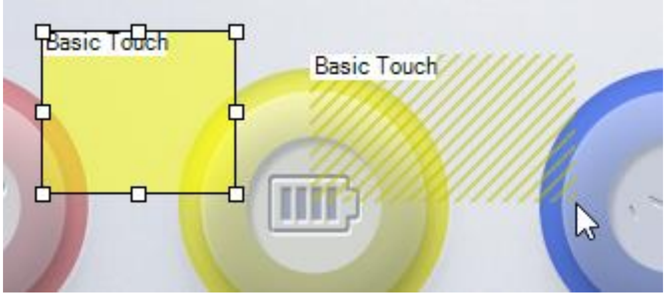
Selection Outline

Caption Font Color

Default Overlay Color

Default Overlay Opacity 0%

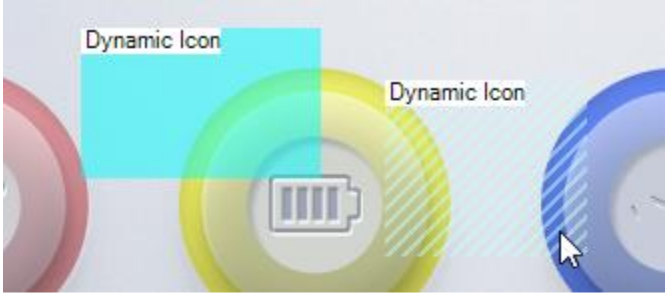
The following pictures show the different color options:



Left: Created Control, with Selection Outline and Resizing Handles.

Right: Creating Control.

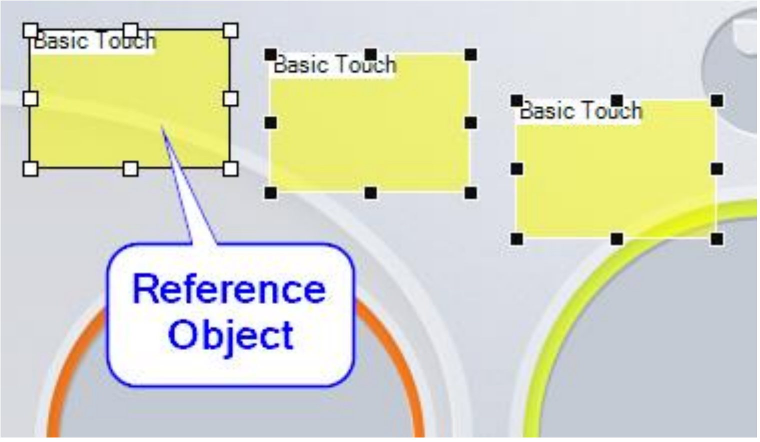
Both: Caption Background Fill and Caption Font.



Left: Created Display Variable.

Right: Creating Display Variable.

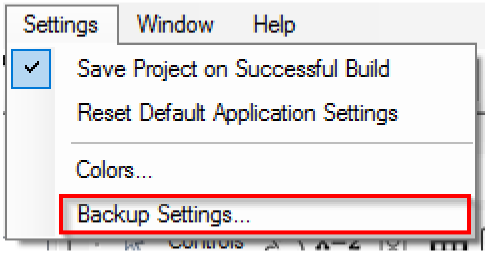
Both: Caption Background Fill and Caption Font.

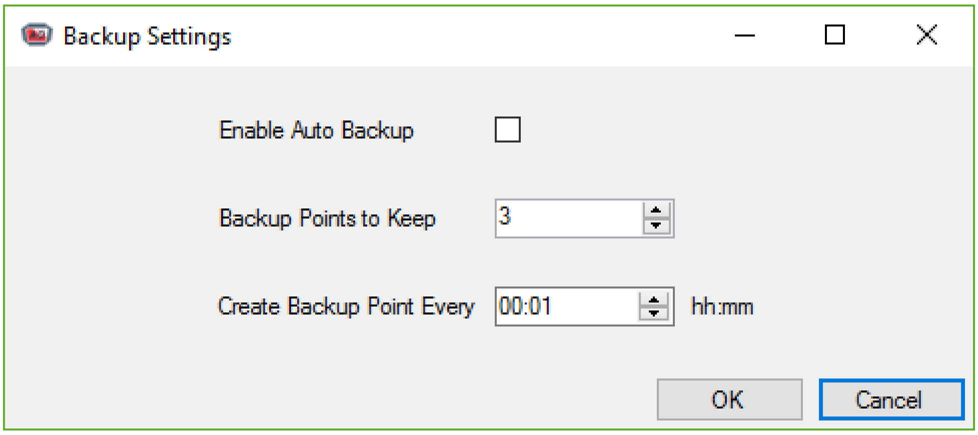


Left: Selected Object with Selection Outline and Resizing Handles (Reference).
Center and Right: Selected Objects with selection Outline and Resizing Handles (inverted colors, not Reference).

2.10.4 Backup Settings

Here you can enable or disable automatic project backup. If enabled, UnicView AD will periodically copy a compressed version of the current project to its “Backup” folder.


Settings
Window
Help
✓ Save Project on Successful Build
Reset Default Application Settings
Colors...
Backup Settings...


Backup Settings

Enable Auto Backup □

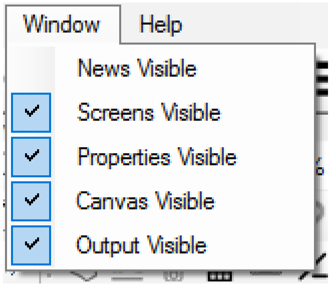
Backup Points to Keep 3

Create Backup Point Every 00:01 hh:mm

You can choose how many backup points to keep, as well as how often they are created.

2.10.5 Window Options

Here you can enable or disable the visibility of all the panels.

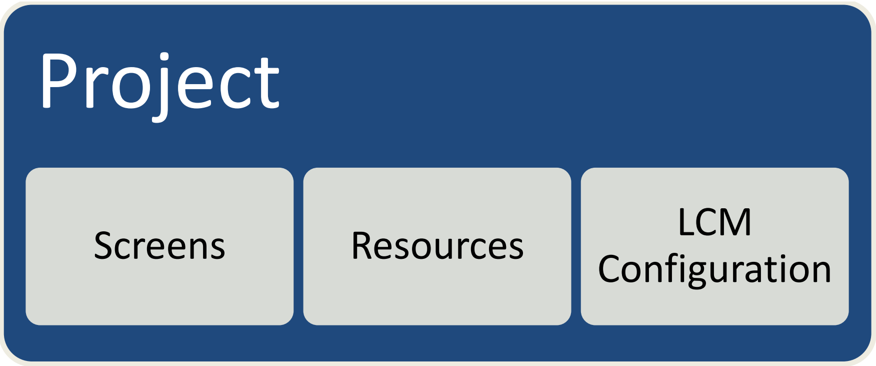


Info The panel proportions are automatically saved, and are considered Application Settings.

2.11 Project

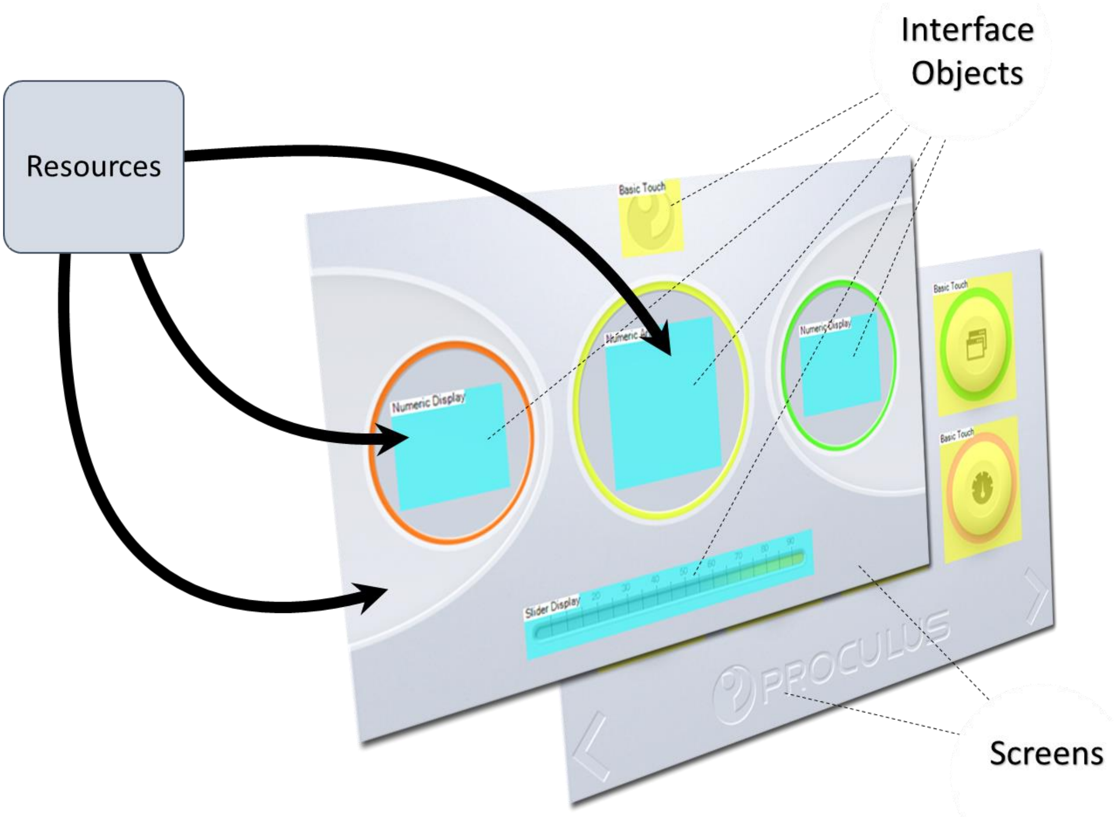
2.11.1 Overview

A project is a collection of Screens, Resources and LCM Configurations.



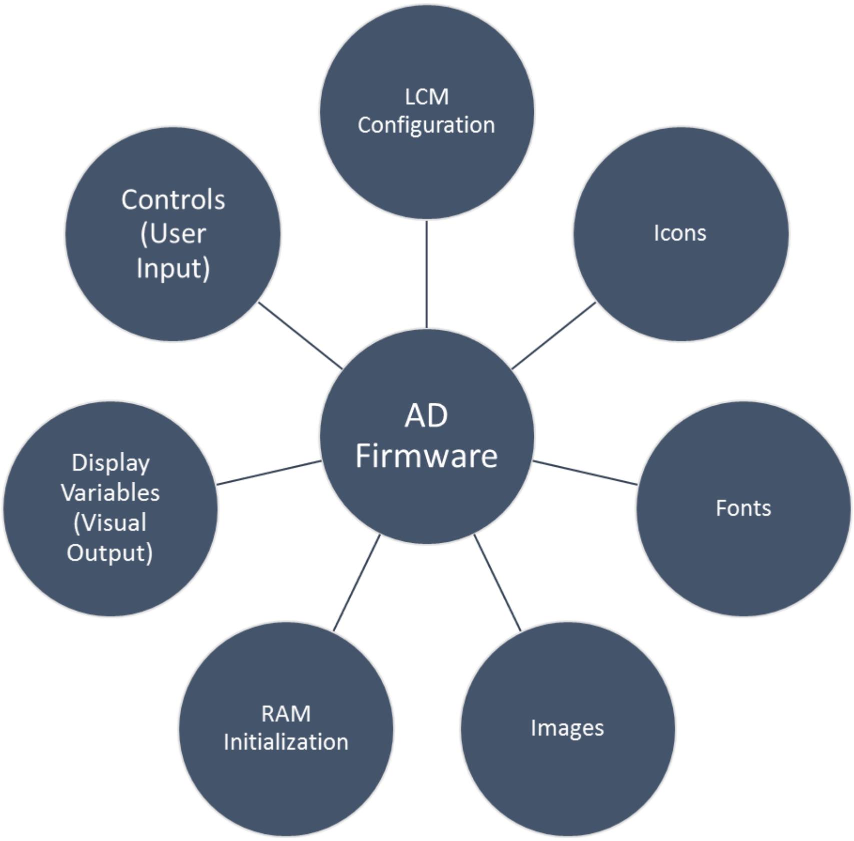
The **Screens** represent the “pages” of the interface on the LCM. Each Screen has **Interface Objects**, which describe the interactive behavior between the LCM and its user. Interface Objects provide input (like a keyboard or a slider) and output (like a text or a graph).

Resources are the images, icons and fonts used to make the Screens graphically appealing.



The **LCM Configurations** are the operational settings, like communication rate, backlight control, etc.

When a project is compiled, it gathers all the information on Screens, Resources and LCM Configuration, and builds the programming files for the LCM.



2.11.2 Folder Structure and File Formats

When a project is created, it generates a folder structure in order to organize all its files. This section describes the folders and files commonly found in a project’s directory.

⚠ Caution **NEVER** manually modify UnicView AD folders or files.

Folder	Content
Audio	Audio reproduction files. Currently not supported.
Backup	Automatic project backup files.
Data	Project data files.
Documentation	Generated documentation files.
Fonts	Generated Font files.
Icons	Generated Icon files, organized by Icon Library.
Images	Copy of the original Image Files.
Log	Application log registries.
OS	AD Assembly (OS) files. Currently not supported.
Output	Compiled output files. Only the files here are transferred to the LCM.
RGB565	Temporary storage for image conversion.
Screenshots	Project screenshots.
Temp	Temporary storage folder. Hidden folder.

File	Description
Project.AdProj	Project information file. Used to open a UnicView AD project.
Data.bin	Main project data storage.
<IconLibraryId>.ICO	Icon Library file.
<FontLibraryId>.DZK	Custom Font Library file.
DefaultBackgroundImage.dbi	Default Background Image file. The underlying format is BMP.
<ImageId>.jpg	Background image (JPEG).
0.hzk	Default Font Library file.
12_QR_CODE.bin	QR Code Library file.
UnicViewAd.bin	Compiled file, containing the Control, Display Variable and RAM Initialization data.
CONFIG.txt	System Configuration file.

2.11.3 File Management

This section describes how to manage project files, including the creation, saving, loading and modification of projects.

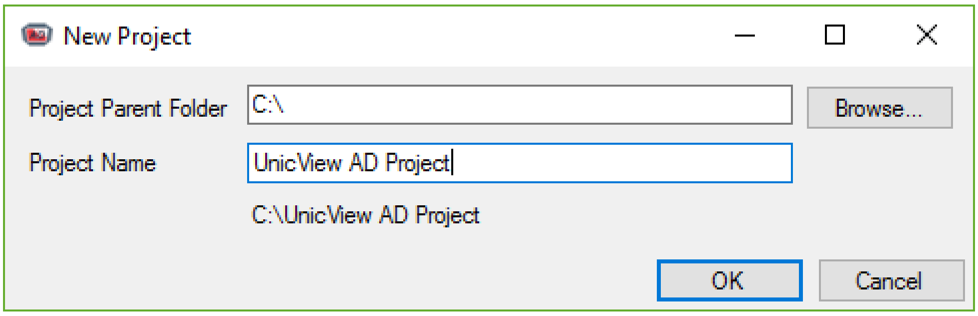
2.11.3.1 Naming Rules

The name of a project may contain any characters, but it's recommended to avoid:

- Special characters, e.g.: # % ¨ $ / \
- Punctuation marks and accentuation, e.g.: ; . , á, ç

2.11.3.2 New Project

To create a new project, click button, on the File menu or File bar. On the New Project dialog that will appear (see the picture bellow), choose the **parent folder** where you want to create the new project, and then type the **project name**. A folder with the project name will be created on the selected path (Project Parent Folder). The complete path is shown on the bottom of the dialog.



New Project

Project Parent Folder: C:\

Project Name: UnicView AD Project

C:\UnicView AD Project

You can't create a project if the selected path already exists. If that's the case, you should first delete the folder.

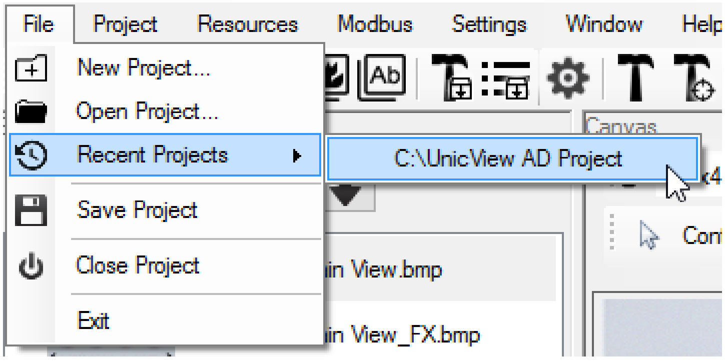
2.11.3.3 Save Project

To save a project, click the button, on the File menu or File bar. Depending on the size of the project, it may take several seconds to complete the saving process. We recommend waiting for the process to complete before proceeding.

Depending on the current Application Settings, the project may be automatically saved after successfully compiling it.

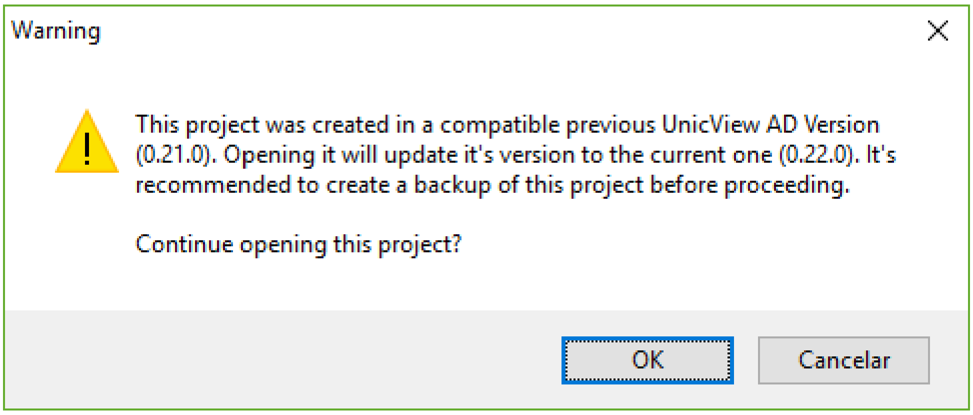
2.11.3.4 Open Project

To open a project, click the 🗀 button, on File menu or File bar. You may also directly open a recent project via the ⟲ menu.



When opening a project from another UnicView AD version, two situations may occur:

- The other version is compatible with the current one, and opening the project will convert it to the current one.

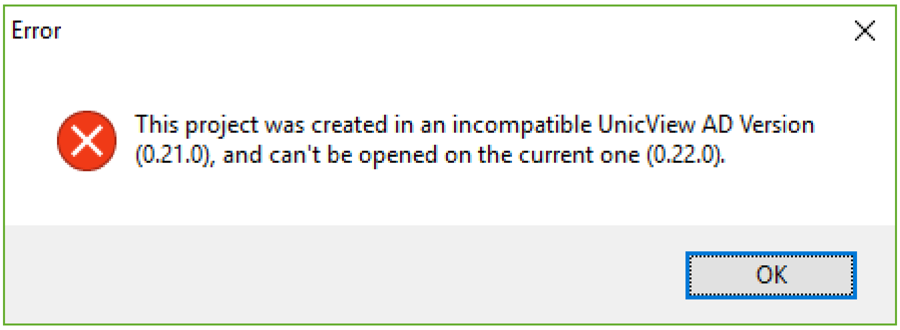


Warning

This project was created in a compatible previous UnicView AD Version (0.21.0). Opening it will update it's version to the current one (0.22.0). It's recommended to create a backup of this project before proceeding.

Continue opening this project?

- The other version is not compatible with the current one, and the project will not be opened.



Error

This project was created in an incompatible UnicView AD Version (0.21.0), and can't be opened on the current one (0.22.0).

ⓘ	Info	If you have compatibility issues, please contact our technical support.

⚠	Caution	Create a backup before trying to open a project from a different UnicView AD version.

If some resource files are missing on a project but it's still able to be opened, a message like the following one will be shown:

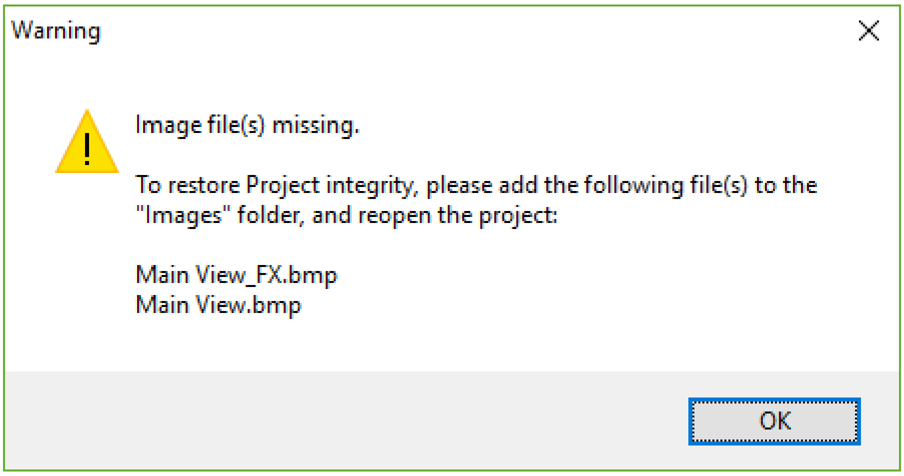
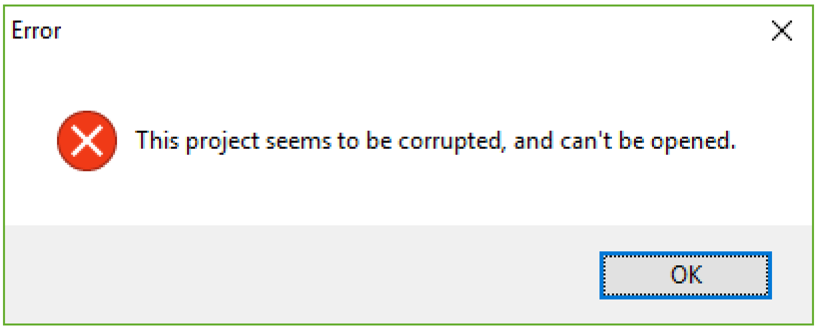


Image file(s) missing.

To restore Project integrity, please add the following file(s) to the "Images" folder, and reopen the project:

Main View_FX.bmp
Main View.bmp

If, by any reason, the project files have been corrupted and can't be read, the project will not be opened, and the following message will be shown:



This project seems to be corrupted, and can't be opened.

2.11.3.5 Close Project

To close a project, click the ⏻ button, on the File menu or File bar, or exit UnicView AD. You'll be asked whether you want to save the project before closing.

ℹ	Info	You may close a project without saving it to undo accidental changes that can't be undone by other means, like removing a Screen.

2.11.3.6 Backup Project

If Automatic Backup is enabled in the Application Settings, you can restore the project to a previous state.

The backup files follow this naming structure:

```
<ProjectName>_<Day>_<Month>_<Year>_<Hours>_<Minutes>_<Seconds>.zip
```

To restore the project, unpack the desired backup file on a new folder, and open the unpacked project. If this backup project is in a satisfactory state, delete the original project and rename the backup project to match the original name.

The backup file doesn't include the “Backup”, “Screenshots” and “Temp” folders.

2.11.3.7 Compress Project

You may want to send a project via e-mail, or keep a version history with all projects compressed to ZIP files. To pack the project, click the button, on the Project menu. A save file dialog will appear, where you can choose where to save the compressed project file.

Caution — Save the project before compressing it.

The compressed project doesn't include the “Backup”, “Screenshots” and “Temp” folders.

2.11.3.8 Rename or Move an Existing Project

To rename a project, close the project (if it's open), and rename the project's main folder. Likewise, to move a project, simply close it, and move its folder to the new location.

Caution — When renaming or moving a project, be sure to comply to the Naming Rules.

Caution — Avoid paths with more than 255 characters, including subfolders and files on the project.

For example:

“C:\UnicView AD Project\Output\26_Agency-FB_20_10x20_AD_Num_50.dzk”

is 65 characters long.

2.11.4 System Configuration

This is the LCM basic configuration settings. Here are located settings like communication options, orientation and backlight control.

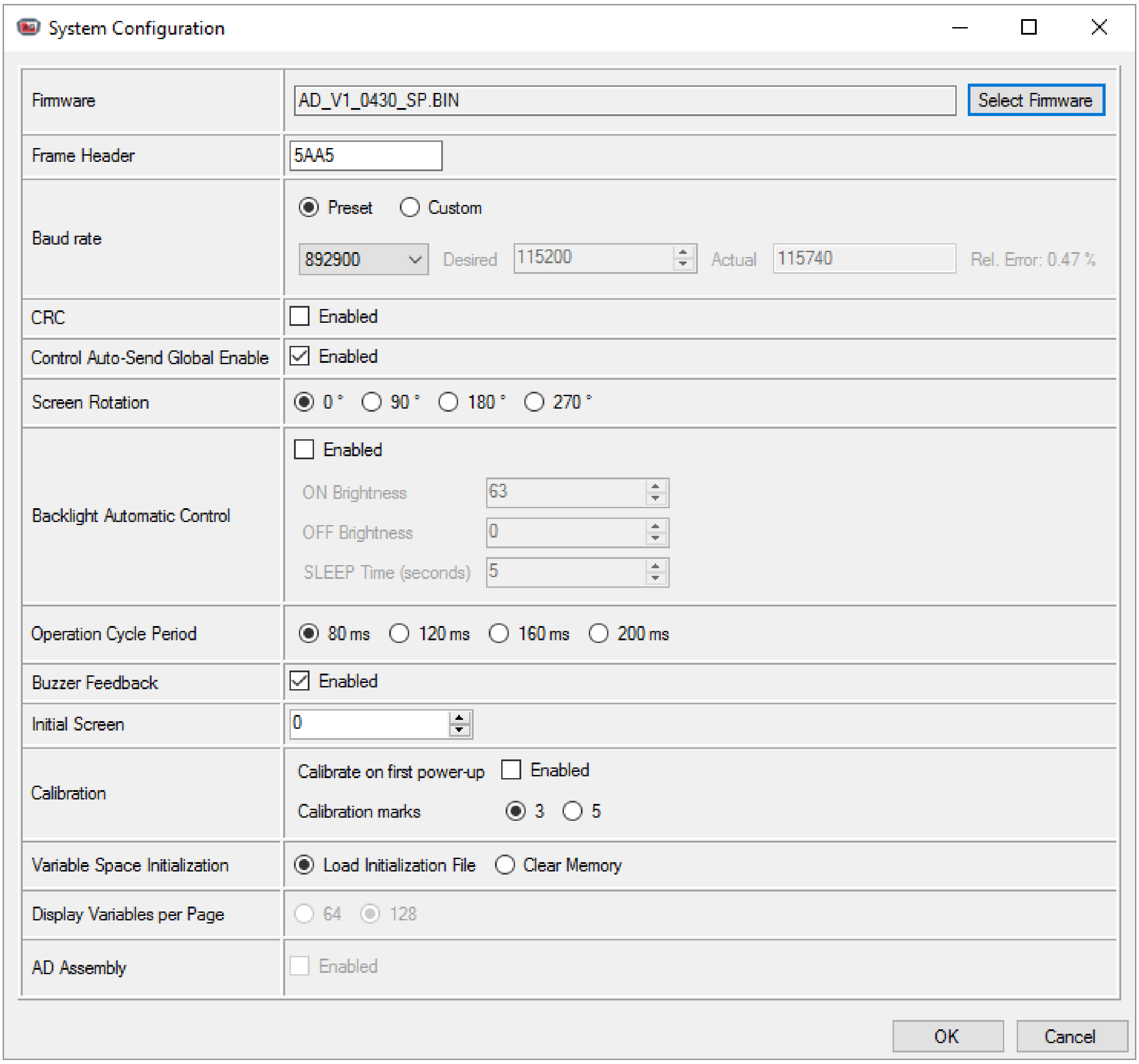
⚠ Caution

Only modify the System Configuration settings through UnicView AD.

Never manually edit the compiled System Configuration file. Doing so may permanently damage the LCM.

ℹ Info

The System Configuration options will automatically change to reflect the currently selected firmware version capabilities.



Firmware	AD_V1_0430_SP.BIN
Frame Header	5AA5
Baud rate	● Preset ○ Custom 892900 Desired 115200 Actual 115740 Rel. Error: 0.47 %
CRC	□ Enabled
Control Auto-Send Global Enable	☑ Enabled
Screen Rotation	● 0 ° ○ 90 ° ○ 180 ° ○ 270 °
Backlight Automatic Control	□ Enabled ON Brightness 63 OFF Brightness 0 SLEEP Time (seconds) 5
Operation Cycle Period	● 80 ms ○ 120 ms ○ 160 ms ○ 200 ms
Buzzer Feedback	☑ Enabled
Initial Screen	0
Calibration	Calibrate on first power-up □ Enabled Calibration marks ● 3 ○ 5
Variable Space Initialization	● Load Initialization File ○ Clear Memory
Display Variables per Page	○ 64 ● 128
AD Assembly	□ Enabled

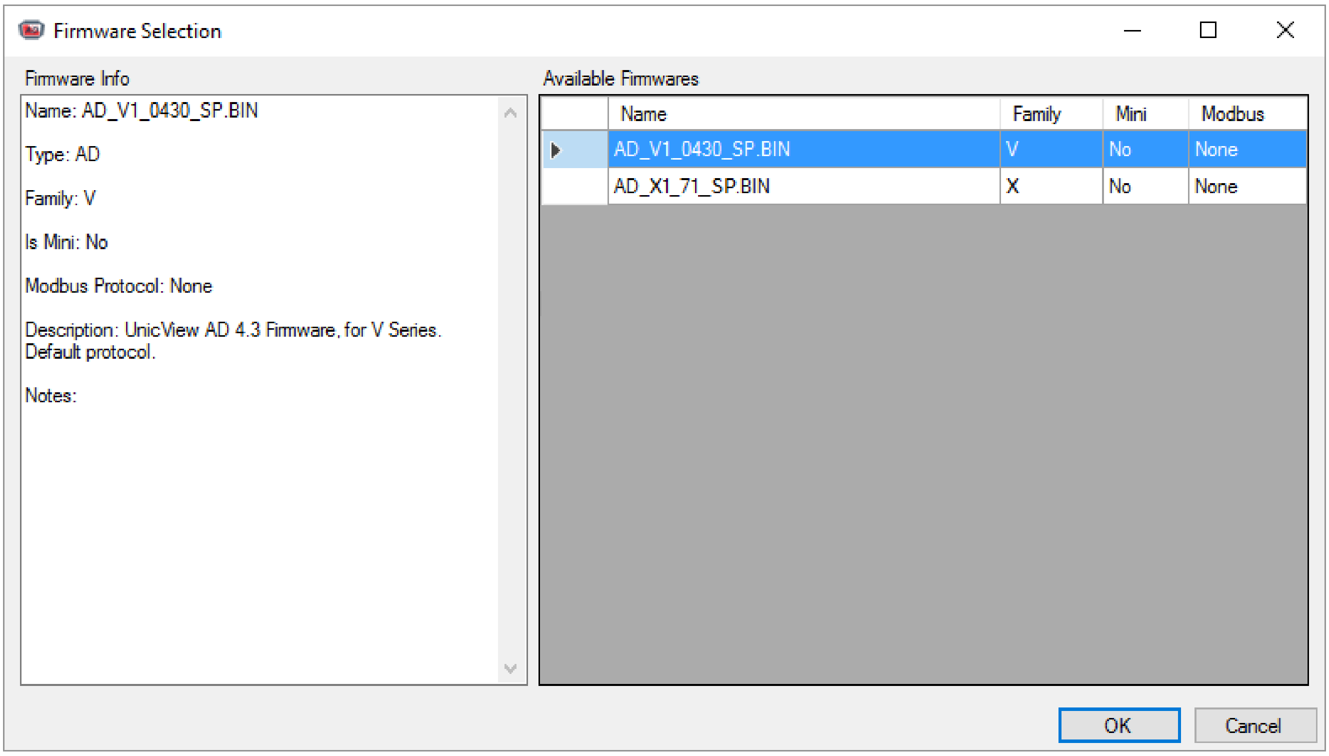
2.11.4.1 Firmware

Here you can choose the target AD firmware. Usually, the best option is the latest version number. The firmware list is automatically updated during UnicView AD's initialization routine.

Info — Always make sure that the correct firmware is selected for your target LCM.

There are several parameters for a given firmware:

Parameter	Description
Name	The name of the firmware file. Also, shows the version number.
Type	Target firmware type. Currently, only show AD types.
Family	Firmware/LCM family.
Is Mini	Indicates if this is a Mini AD firmware.
Modbus Protocol	Indicates if the communication protocol is the default AD Protocol (none), Modbus Master or Modbus Slave.
Description	A brief description of this firmware.
Notes	Additional remarks about this firmware.



Name: AD_V1_0430_SP.BIN

Type: AD

Family: V

Is Mini: No

Modbus Protocol: None

Description: UnicView AD 4.3 Firmware, for V Series.
Default protocol.

Notes:

Name	Family	Mini	Modbus
AD_V1_0430_SP.BIN	V	No	None
AD_X1_71_SP.BIN	X	No	None

If a firmware doesn't support any Interface Objects, the respective Interface Object buttons will be disabled from the Canvas tool bar.

2.11.4.2 Frame Header

The frame header for Serial Communication Protocol. It's the preceding two bytes that indicate a valid AD Protocol packet.

Any 4-digit hexadecimal value is accepted. We recommend 5AA5, which is the default value.

2.11.4.3 Baud rate

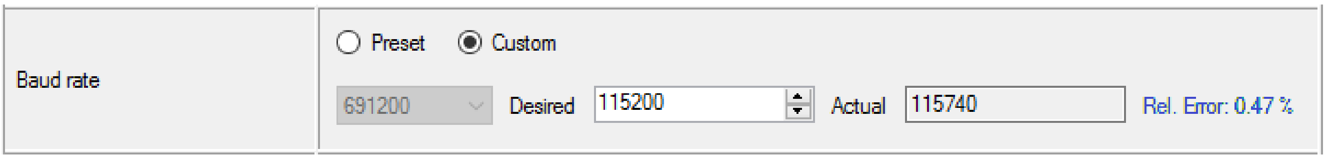
Serial Communication Protocol transfer rate. Should match the intended communication bus baud rate. Refer to section “2.12 - Serial Port Communication” for more information.

You can use a predefined, built-in baud rate, or a custom, software-generated baud rate:

- Preset – Limited range of baud rates. Very accurate frequency values.
- Custom – Wider range of baud rates, implemented by software-divider. Generates less accurate frequency values. The percentage frequency deviation (Relative Error) is color-coded:
    - Blue: Acceptable deviation, less than 2.0%
    - Red: Unacceptable deviation, equal or greater than 2.0%

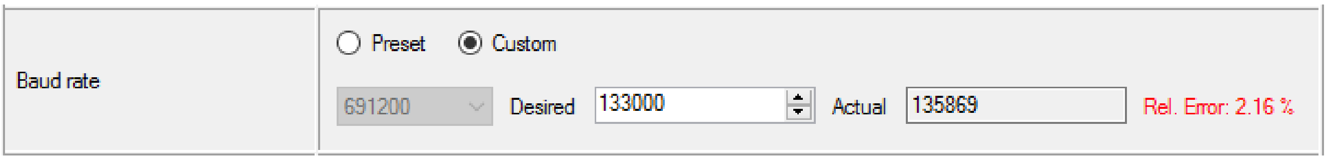
⚠ Caution If a custom baud rate value deviates more than 2.0%, the communication may fail.

Acceptable baud rate deviation:



Baud rate ○ Preset ● Custom 691200 Desired 115200 Actual 115740 Rel. Error: 0.47 %

Unacceptable baud rate deviation:



Baud rate ○ Preset ● Custom 691200 Desired 133000 Actual 135869 Rel. Error: 2.16 %

2.11.4.4 CRC

Enable or disable the CRC (Cyclic Redundancy Check) communication check. Refer to section “2.12 - Serial Port Communication” for more information.

2.11.4.5 Control Auto-Send Global Enable

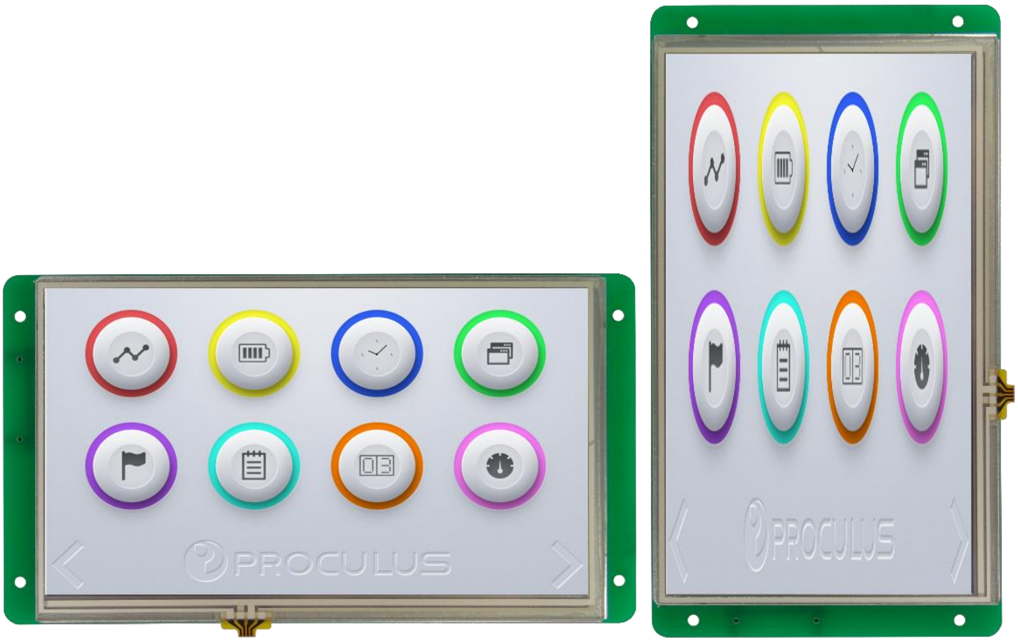
Global enabling setting for automatic data sending on Controls (property **Auto-Send Data**):

- **Disabled** - No Controls will send any data through Serial Port.
- **Enabled** - Each Control functions according to its **Auto-Send Data** property.

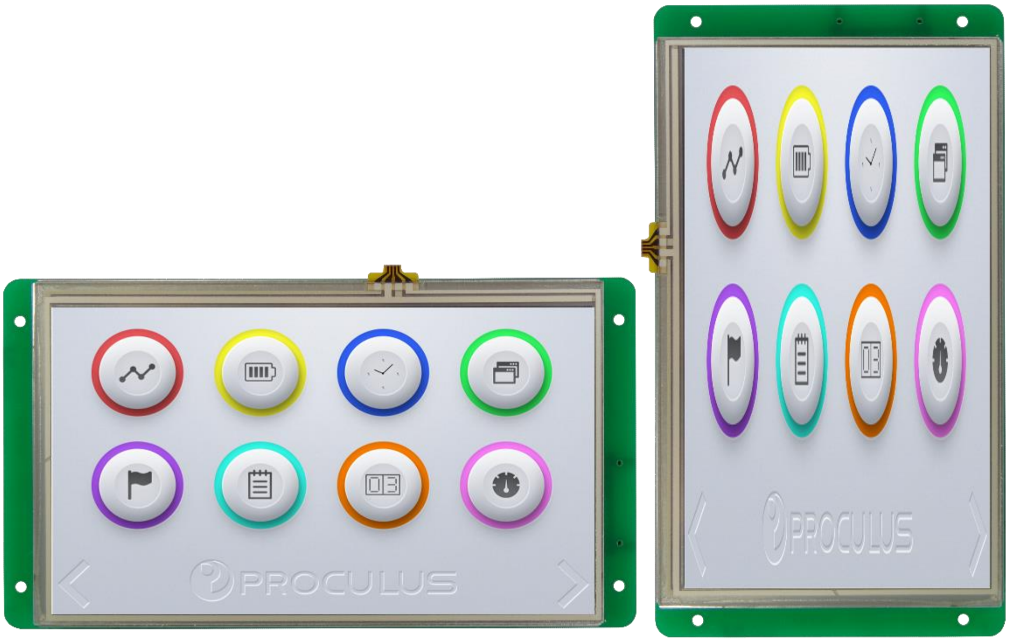
2.11.4.6 Screen Rotation

This setting indicates what is the orientation of the project on the LCM. The rotations are clockwise, regarding the LCM stationary.

⚠ Caution After changing the LCM's orientation, you must calibrate the Touch panel again.



0 degrees and 90 degrees



180 degrees and 270 degrees

2.11.4.7 Backlight Automatic Control

If this setting is enabled, the LCM can be configured to enter low-brightness mode automatically after a set period.

- ON Brightness – Brightness level at normal mode. Range: [0,63]
- OFF Brightness – Brightness level at low-brightness mode. Range: [0,63]
- SLEEP Time – The period, in seconds, after which the LCM will enter low-brightness mode. Counted after the last touch on the Touch Panel. Range: [0,255]

2.11.4.8 Operation Cycle Period

The duration of each operation cycle of the AD firmware. Lower values mean quicker execution.

Available values for standard AD firmwares:

- 80 ms
- 120 ms
- 160 ms
- 200 ms

Available values for Mini AD firmwares:

- 100 ms
- 200 ms

2.11.4.9 Buzzer Feedback

- Enabled - Buzzer sounds when the user touches a Control
- Disabled - Buzzer doesn’t sound when the user touches a Control

2.11.4.10 Calibration

Sets the Touch Panel calibration mode to 3 or 5-point calibration. Also, enables or disables the calibration routine on the first power-up after the System Configuration is written on the LCM.

2.11.4.11 Initial Screen

Sets the Screen that shows up when the LCM is powered on.

2.11.4.12 Variable Space Initialization

Controls the RAM initialization method:

- Load Initialization File – The RAM is initialized according to the Object Table and RAM Map
- Clear Memory – The RAM is initialized to zero on all addresses

2.11.4.13 Display Variables per Page

Sets the maximum number of Display Variables per Page. Despite the maximum number of Screens (PicIds) a given LCM supports, all LCMs have the same limit of Screens with Display Variables on them. The following table shows the two possible settings:

Display Variables per Page	PicId range that accepts Display Variables
64	[0,1023]
128	[0,511]

ⓘ Info This feature is currently not supported.

2.11.4.14 AD Assembly

Enables or disables AD Assembly processing (OS).

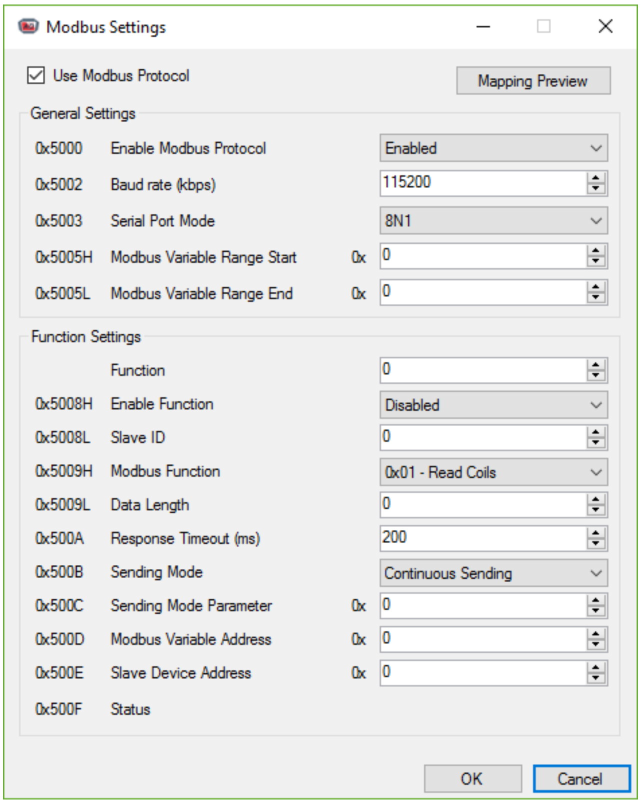
ⓘ Info This is an upcoming feature, currently not supported.

2.11.5 Modbus Settings

When using a **Modbus-ready firmware**, the communication protocol can be changed to Modbus RTU (Master or Slave, depending on the firmware) mode. The Modbus Settings are stored at the high portion of the RAM, and are loaded upon LCM power-up (if the RAM Initialization is set to “**Load Initialization File**”).

⚠ Caution	When using Modbus, the RAM addresses in these ranges are reserved: [0x5000,0x6FFF] for AD [0x0600,0x07FF] for Mini AD

To configure the Modbus protocol settings, click the ⚙MB button, on the Modbus menu or Modbus bar, which will open the Modbus Settings window.



Modbus Settings

☑ Use Modbus Protocol

Mapping Preview

General Settings

0x5000	Enable Modbus Protocol		Enabled
0x5002	Baud rate (kbps)		115200
0x5003	Serial Port Mode		8N1
0x5005H	Modbus Variable Range Start	0x	0
0x5005L	Modbus Variable Range End	0x	0

Function Settings

	Function		0
0x5008H	Enable Function		Disabled
0x5008L	Slave ID		0
0x5009H	Modbus Function		0x01 - Read Coils
0x5009L	Data Length		0
0x500A	Response Timeout (ms)		200
0x500B	Sending Mode		Continuous Sending
0x500C	Sending Mode Parameter	0x	0
0x500D	Modbus Variable Address	0x	0
0x500E	Slave Device Address	0x	0
0x500F	Status		

OK Cancel

On this window, you can set the General Settings, and Function Settings. **The General Settings** control the overall aspects of the protocol, and the **Function Settings** configure each specific Modbus command.

UnicView AD supports up to 1023 different Modbus functions (command presets). Each function is stored in a fixed address range, and can be modified at run-time, as any VP would normally work. For example, you could **pre-configure several functions** with the same Modbus command (let’s take command 0x03, for instance), and each one be used when needed, or you could use **only one function**, but let the user modify its parameters during usage.

⚠	Caution	Since the General and Function Settings address mapping differs between AD and Mini AD Firmwares, you should always make sure that the correct firmware is selected in the System Configuration.

To enable the Modbus, mark the “Use Modbus Protocol” checkbox at the top of the window. Make sure that you’re using a **Modbus-ready firmware**.

The Modbus Command Table, accessed via the MB button, provides a quick view of all Modbus functions in a table format.

You can verify the raw Modbus configuration data by looking their respective addresses in the Object Table and RAM Map.

ℹ	Info	When using Modbus Protocol, the standard AD Protocol is disabled. Project Download to LCM via Serial Communication is also disabled.

2.11.6 Resources

All the resources on the project are allocated after a successful compilation. The allocation table, and details on how to add resources to the project, can be found in section “2.12 - Serial Port Communication”.

2.11.7 Screens

A **Screen** is a page of the LCM. Each Screen is identified by a unique **PicId** (Picture Index).

The **Background Image** is the picture that is painted on the whole LCM’s LCD, over which the other elements (Interface Objects) are painted.

Details on how to design a Screen can be found in section “5 - Screen Composition”.

2.11.8 Compiling

Compiling, or Building, is the process of collecting all project data, interpreting it, them generating the configuration files used to transfer the project to a LCM.

Before transferring the project, it’s necessary to compile it. Compiling may also be required to update the data from all the Resources used in the project.

2.11.9 Download to LCM

After you’ve designed your graphical user interface on UnicView AD, you need to download it to a Proculus LCM.

Transference methods:

- USB Flash Drive
- USB Cable

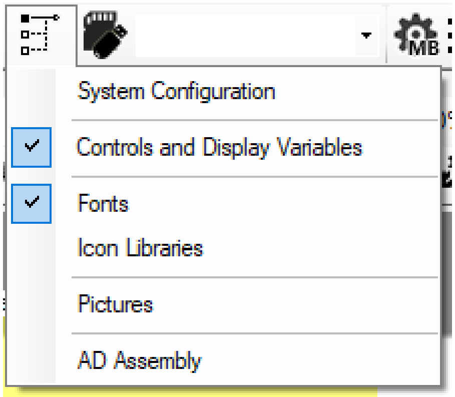
Whatever method you choose, you can specify what files are to be transferred by clicking the button to open the **"Output Transfer Options"**, in the Project menu or Project bar. Then, you can check only the files you need.

ℹ	Info	Always check the Output Transfer Options before starting a project download to the LCM, because some files (like Pictures and Icon Libraries) take a long time to send, even at high baud rates.

ℹ	Info	Refer to section "6.2 - LCM Memory Cleaning" for additional information on Display Variables and RAM Initialization file sizes.

2.11.9.1 Output Transfer Options

The following table describes each item on the Output Transfer Options menu.

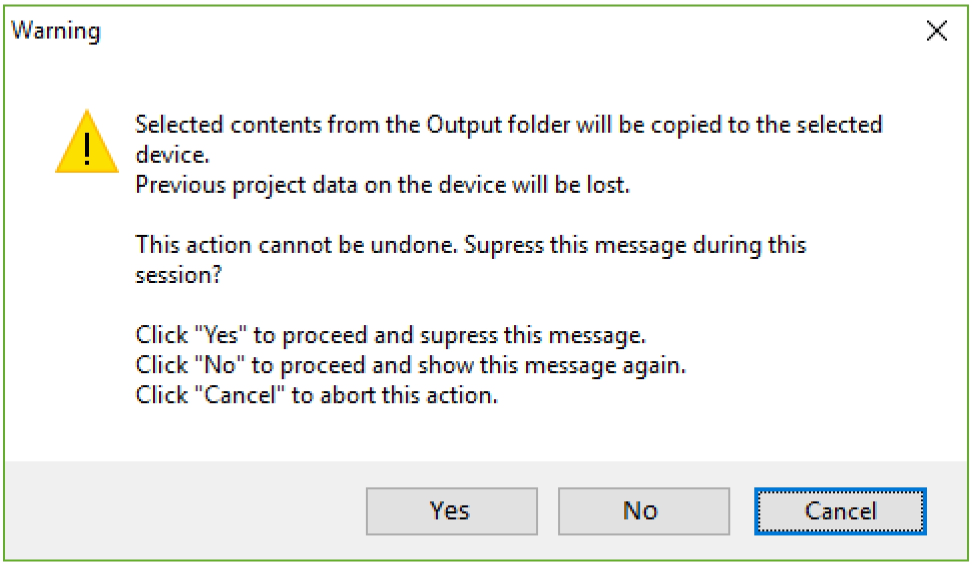


Item	Description	Notes for Serial Communication method
System Configuration	Sends the System Configuration file.	The transference of this file is not acknowledged by the LCM. As such, even if the LCM didn't receive the file, the transference will report success. If you changed the LCM baud rate, you must close the Serial Port, change its baud rate accordingly, then reopen the Serial Port.
Controls and Display Variables	Sends the Control Configuration, Display Variable Configuration and RAM Initialization files.	
Fonts	Sends the Custom Font files.	
Icon Libraries	Sends the Icon Library files.	Large files. Send only when necessary.
Pictures	Sends the checked Background Image files.	Large files. Send only when necessary.
AD Assembly	Sends the AD Assembly file.	

2.11.9.2 USB Flash Drive

To download the project via USB Flash Drive:

1. Insert a Flash Drive on your computer.
2. Wait for it to be recognized by the computer.
3. Select the Flash Drive on the Dropdown list in Project menu or Project bar.
4. Click the button to copy the Output folder to the Flash Drive.
5. A message window will appear, warning you that previous projects on the Flash Drive will be overwritten. You may proceed or abort this operation. If you proceed, you may suppress further warnings during this instance of UnicView AD.



Warning

Selected contents from the Output folder will be copied to the selected device.
Previous project data on the device will be lost.

This action cannot be undone. Supress this message during this session?

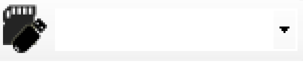
Click "Yes" to proceed and supress this message.
Click "No" to proceed and show this message again.
Click "Cancel" to abort this action.

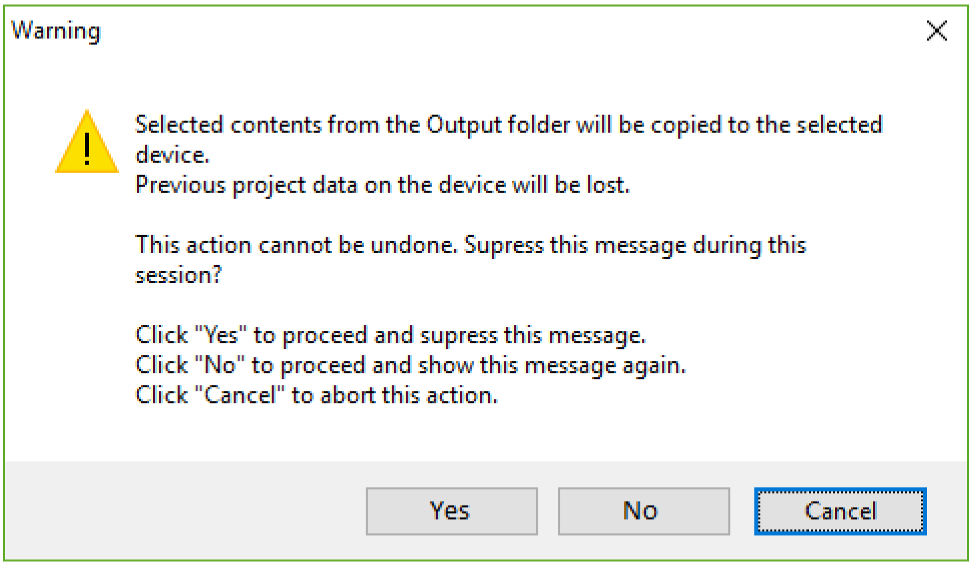
6. Wait for a message informing you that the copy process is complete.
7. Remove the Flash Drive from your computer.
8. Insert it on your LCM. We recommend that you first turn off the LCM.
9. Wait for the transference to complete.
10. Remove the Flash Drive from the LCM.

Caution When downloading the project via USB Flash Drive, the contents of the LCM’s memory will be erased completely, so it’s recommended to **select all items** from the Output Transfer Options menu.

2.11.9.3 USB Cable Method

To download the project via USB Cable:

1. Connect the LCM on your computer via a USB Mini-B cable.
2. Wait for it to be recognized by the computer.
3. Select the LCM on the Dropdown list  in Project menu or Project bar.
4. Click the button to copy the Output folder to the LCM.
5. A message window will appear, warning you that previous projects on the LCM will be overwritten. You may proceed or abort this operation. If you proceed, you may suppress further warnings during this instance of UnicView AD.



Warning

Selected contents from the Output folder will be copied to the selected device.
Previous project data on the device will be lost.

This action cannot be undone. Supress this message during this session?

Click "Yes" to proceed and supress this message.
Click "No" to proceed and show this message again.
Click "Cancel" to abort this action.

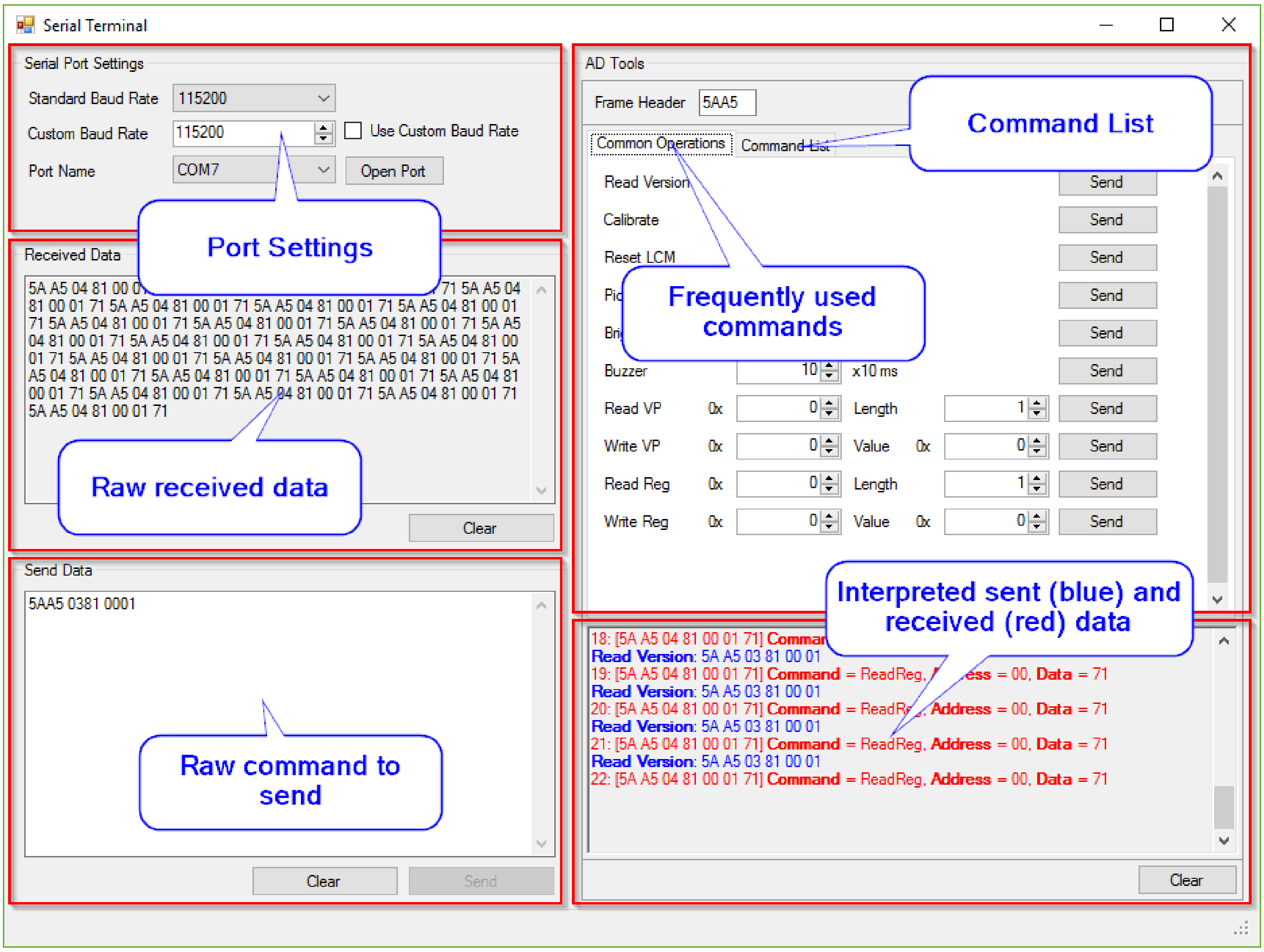
6. Wait for a message informing you that the copy process is complete.
7. Disconnect the USB cable.

2.12 Serial Port Communication

The Serial Port is used to establish bilateral communication with connected LCMs. You can use it to transfer project files and to write and read data from the LCM.

⚠ Caution **NEVER** remove the USB cable while the Serial Port is open. It may crash the application.

Although you can use the built-in **Serial Terminal** to communicate with any serial device, it provides additional functionalities for communicating with AD LCMs.



Serial Port Settings

Standard Baud Rate: 115200

Custom Baud Rate: 115200 □ Use Custom Baud Rate

Port Name: COM7

Received Data

```
5A A5 04 81 00 0                                      71 5A A5 04
81 00 01 71 5A A5 04 81 00 01 71 5A A5 04 81 00 01 71 5A A5 04 81 00 01
71 5A A5 04 81 00 01 71 5A A5 04 81 00 01 71 5A A5 04 81 00 01 71 5A A5
04 81 00 01 71 5A A5 04 81 00 01 71 5A A5 04 81 00 01 71 5A A5 04 81 00
01 71 5A A5 04 81 00 01 71 5A A5 04 81 00 01 71 5A A5 04 81 00 01 71 5A
A5 04 81 00 01 71 5A A5 04 81 00 01 71 5A A5 04 81 00 01 71 5A A5 04 81
00 01 71 5A A5 04 81 00 01 71 5A A5 04 81 00 01 71 5A A5 04 81 00 01 71
5A A5 04 81 00 01 71
```

Send Data

```
5AA5 0381 0001
```

AD Tools

Frame Header: 5AA5

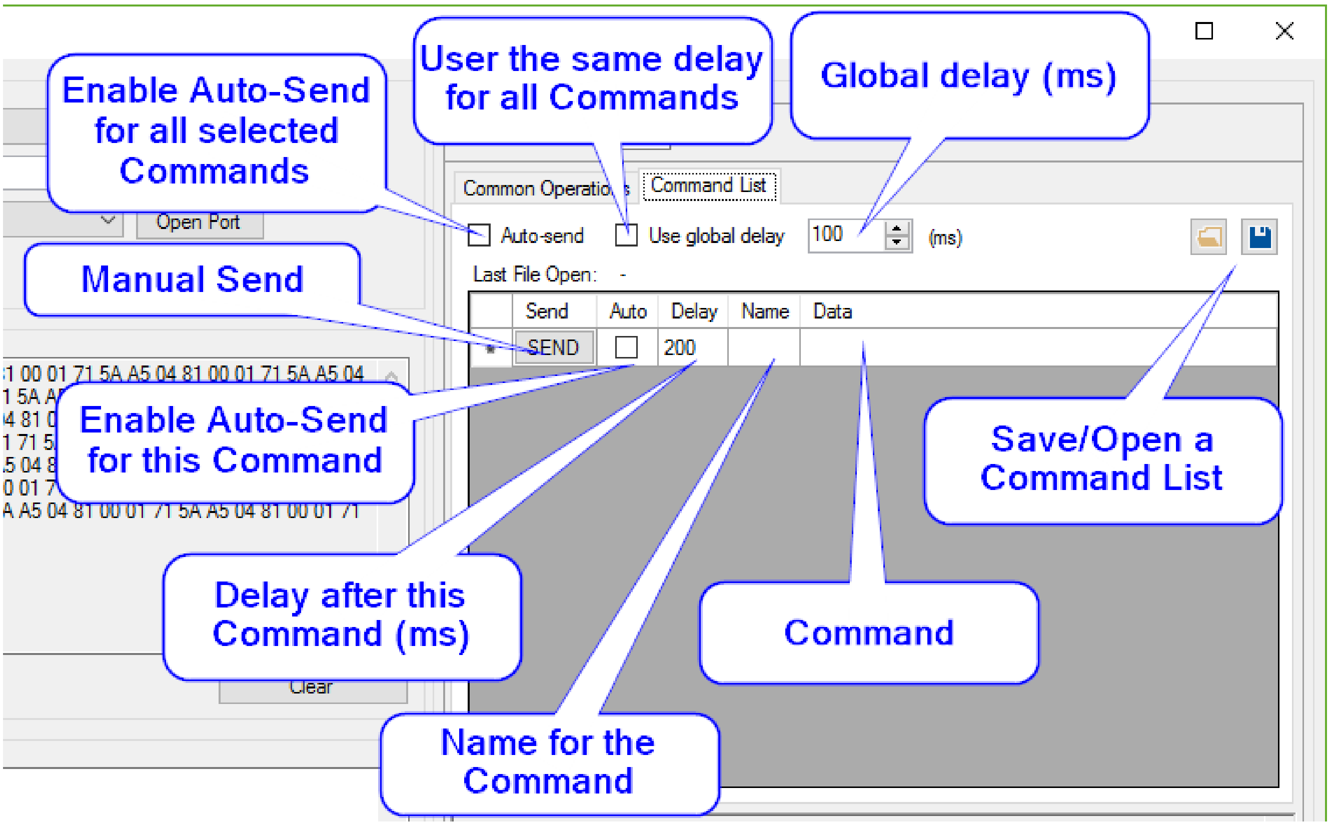
Read Version					Send
Calibrate					Send
Reset LCM					Send
Pic					Send
Bri					Send
Buzzer		10	x10 ms		Send
Read VP	0x	0	Length	1	Send
Write VP	0x	0	Value 0x	0	Send
Read Reg	0x	0	Length	1	Send
Write Reg	0x	0	Value 0x	0	Send

```
18: [5A A5 04 81 00 01 71] Commar
Read Version: 5A A5 03 81 00 01
19: [5A A5 04 81 00 01 71] Command = ReadReg, A     ess = 00, Data = 71
Read Version: 5A A5 03 81 00 01
20: [5A A5 04 81 00 01 71] Command = ReadR  , Address = 00, Data = 71
Read Version: 5A A5 03 81 00 01
21: [5A A5 04 81 00 01 71] Command = ReadReg, Address = 00, Data = 71
Read Version: 5A A5 03 81 00 01
22: [5A A5 04 81 00 01 71] Command = ReadReg, Address = 00, Data = 71
```

You can configure the Serial Port settings, like Port Name and Baud rate, on this window. It also provides AD Protocol parsing (interpretation) and shortcuts for frequently used commands.

2.12.1 Command List

The Command List panel is used to save a list of user-defined commands. Additionally, you can use it to automatically send th e commands.



The following table briefly describes the Command List panel:

Item	Description	Notes
Send	Click to immediately send the command to the LCM.	
Auto	Check to include the command on the auto-send list.	
Delay	Time interval after the command is sent.	Range: [1, 9999]
Name	Description of the command.	Optional

Data	Serial command data.	Hexadecimal format;
		Max. length: 512 digits (256 bytes);
		Spaces are ignored

Auto-send	When checked, all the checked commands are sent sequentially, from top to bottom.	
Use global delay	The same time interval is used after every command, when auto-sending.	

The Command List may be saved and loaded, using the 💾 and 📂 buttons on the top-right region of the Command List panel. The command list file has the extension *.CLF.

3 Resource Management

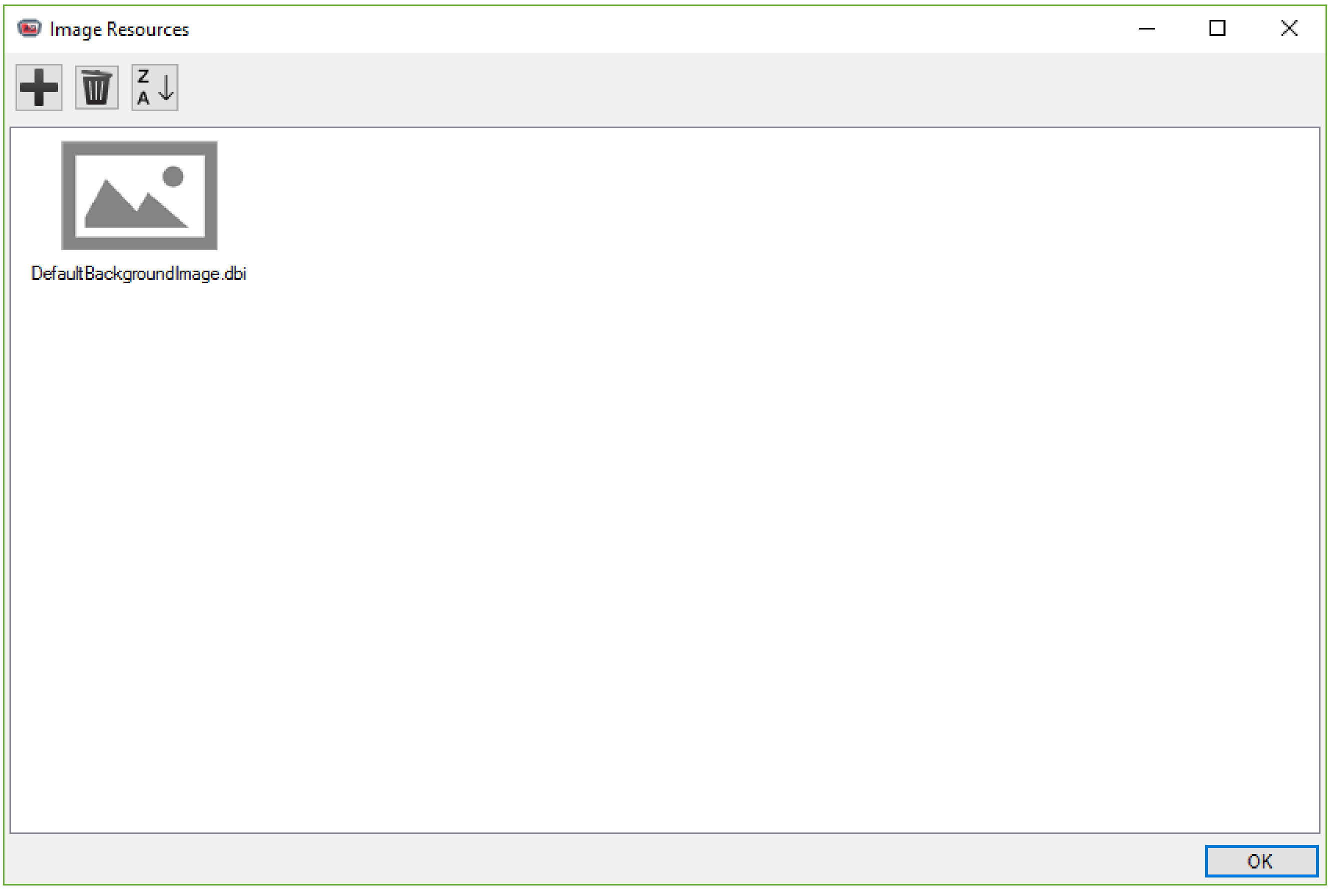
3.1 Image Resources

Each Screen has a background image. The collection of background images on the project is managed on the Image Resources window.

3.1.1 Adding and Removing Image Resources

To view the Image Resources window, click the 🖼 button on the Resources menu or Resources bar. All projects have a default image resource that can't be removed, named "DefaultBackgroundImage.dbi".

ℹ Info After editing any project resource, compile the Resources, using T or T.



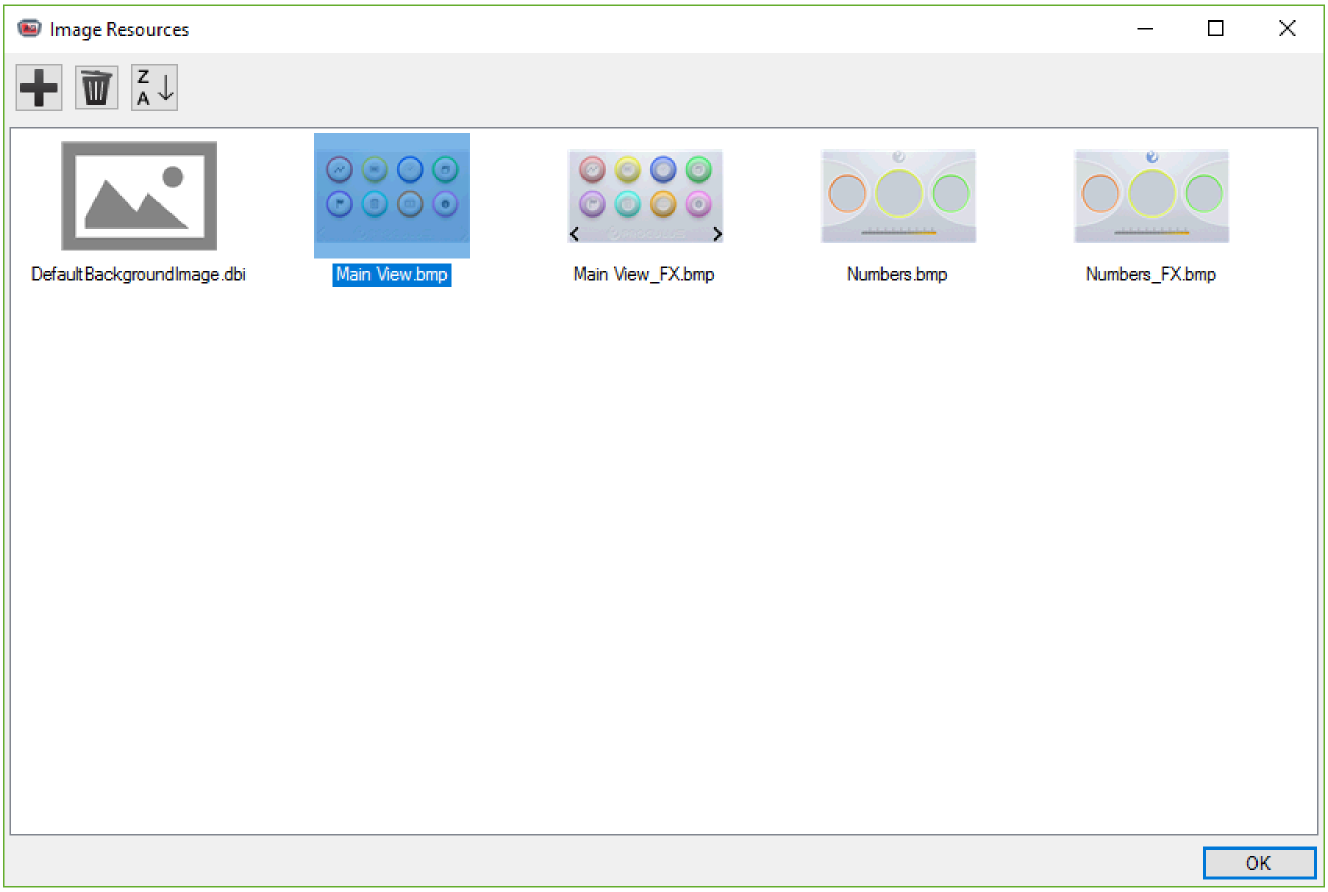
ℹ Info You can sort the images alphabetically by clicking the A Z ↓ / Z A ↓ button.

⚠ Caution Image addition/removal can’t be undone.

To add images to the project, click the ✚ button. A file selection dialog will appear, where you can choose what images to add.

The selected images will be added to the current project, and copied to its “Images” folder.

Images must have unique names, and if you try to add an image with a name already present on the project, a dialog window will appear, asking if you want to overwrite the existing image with the new one.



Multiple images can be selected using **(CTRL+LMB)**, **(SHIFT+LMB)**, or **(SHIFT+[UP/DOWN/LEFT/RIGHT])**.

To remove images from the project, select the images you want to remove, then click the 🗑 button. A dialog window will appear, asking if you’re sure you want to remove the images.

3.1.2 Naming Rules

The name of an image may contain any characters, but it's recommended to avoid:

- Special characters, e.g.: `# % ¨ $ / \`
- Punctuation marks and accentuation, e.g.: `; . , á, ç`

Usually, every image will have a counterpart image with "pressed" effects for the Controls. A **normal image** is an image that is supposed to contain Controls. An **effect image** is an image that is supposed to have no Controls and only provide the "pressed" effect graphics (see more in section 5.2.1 - "Pressed" Effect).

To make the alphabetic sorting of images more intuitive, we recommend using one of the following name structure pairs:

Normal Image	Effect Image
<ImageName>	<ImageName>_FX
<ImageName>	<ImageName>_Effect
<ImageName>_Normal	<ImageName>_FX

If you wish to have your images sortable in a specific order, you can prefix a number to the image names.

⚠ Caution

We don't recommend using number-prefixed file names. It may cause the need of extensive reworks if the prefixes must be changed in the future.

If prefixes *are* used, consider leaving gaps between in the indexing values.

Examples:

- 001 Main Panel
- 002 Main Panel_FX

It's important that the number has enough leading zeros (zeros before the first significant digit), because in an alphabetic sort, "10 Intro" comes before "2 Intro".

3.1.3 Accepted File Formats

Accepted image file formats: BMP, JPEG, JFIF, TIF, PNG, GIF.

ℹ Info

If a project is vertical (90° or 270° rotation), all images must be vertical as well (upright).

3.2 Icon Resources

To view the Image Resources window, click the button on the Resources menu or Resources bar.

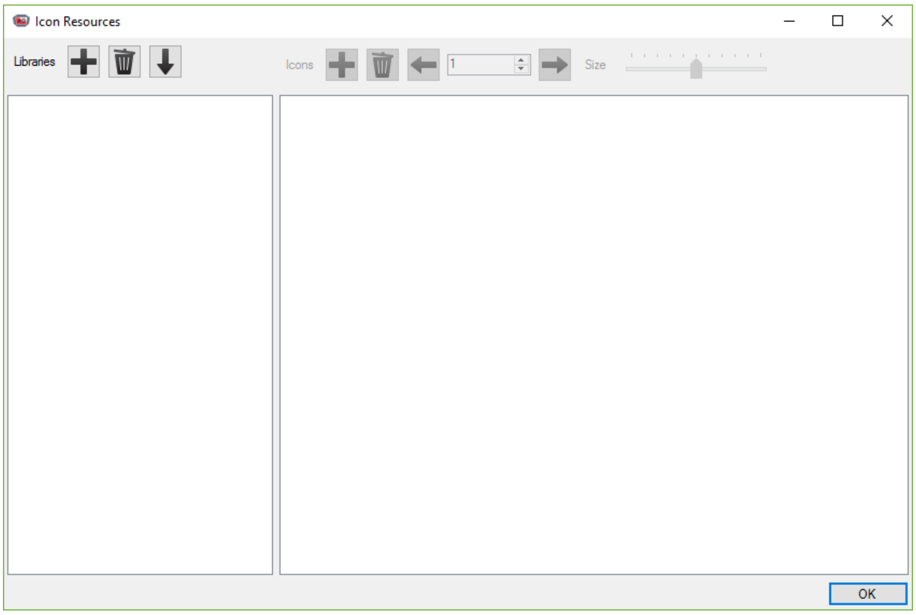
Info After editing any project resource, compile the Resources, using or .

Some definitions regarding Icon Resources:

Item	Description
Icon	An individual image, which is shown on the LCM. They are indexed by IconIds.
IconId	An indexing number for each Icon on an Icon Library.
Icon Library	A collection of Icons, compiled into a single file.
Icon Resource	Refers both to Icons and Icon Libraries, depending on the context.

3.2.1 Adding and Removing Icon Resources

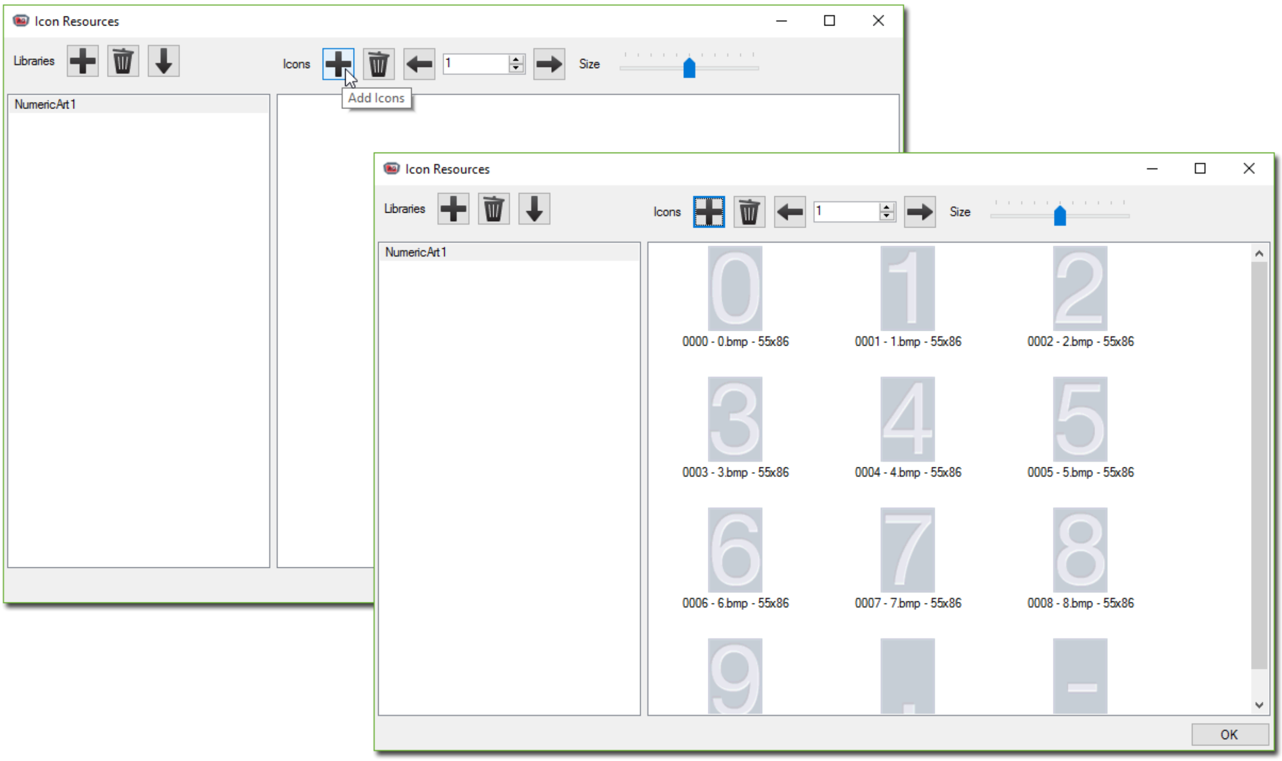
Caution Icon/Icon Library addition/removal can’t be undone.



To add Icon Libraries to the project, click the + button on the **Libraries** panel. A name dialog will appear, where you can choose the name of the new Icon Library. Icon Libraries must have unique names.

The new Icon Libraries are added to the Libraries panel, and a folder with the Library name is created in the “Icons” folder.

To add Icons to the selected Icon Library, click the + button on the **Icons** panel. A file selection dialog will appear, where you can choose what images to add.



Icons must have unique names, and if you try to add an image with a name already present on the project, a dialog window will appear, asking if you want to overwrite the existing Icon with the new one.

The selected images are added to the current Icon Library, and copied to the Library folder.

Info The maximum size of an Icon is 255x255 pixels.

You can also change the preview size of the Icons through the “Size” slider.

If one or more Icons have dimensions greater than 255x255 pixels, a dialog window will appear, giving you three options:

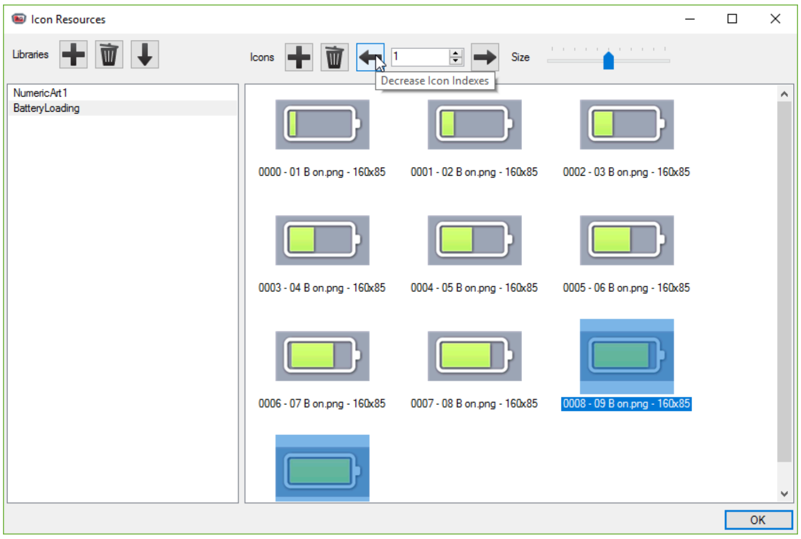
- Auto-scale, preserving aspect ratio – Shrinks the images proportionally, until they fit in 255x255 pixels
- Auto-scale, distorting – Shrinks the images independently on each axis, until the fit in 255x255 pixels
- Skip images – Doesn’t add the images to the Icon Library

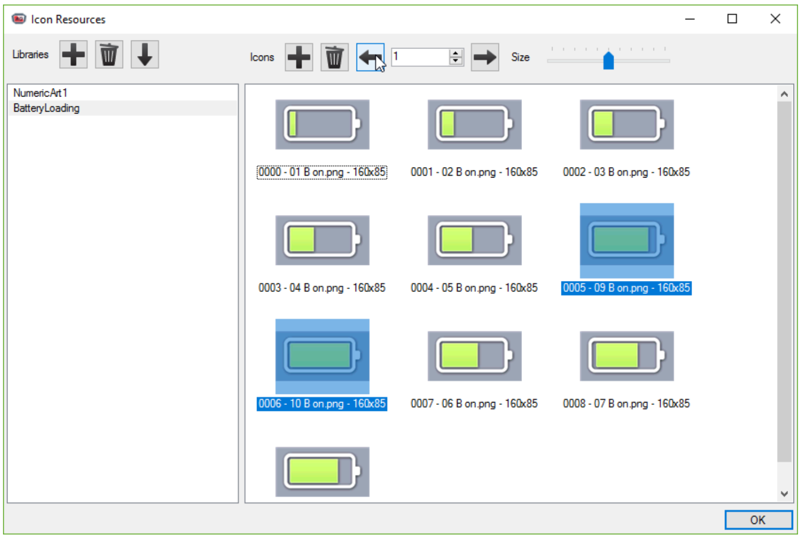
3.2.2 Transparency

The transparency of an icon is determined by its first pixel, i.e., its top-left pixel. The transparency color (or transparency key) is the color of the first pixel.

3.2.3 Index Organization

To change the index of Icons, select one or more Icons, then click ⬅ to decrease or ➡ to increase the selected IconIds. The amount by which the indexes are increased or decreased can be changed on the Up/Down control between the ⬅ and ➡ buttons.


Icon Resources
Libraries
Icons
1
Size
Decrease Icon Indexes
NumericArt1
BatteryLoading
0000 - 01 B on.png - 160x85
0001 - 02 B on.png - 160x85
0002 - 03 B on.png - 160x85
0003 - 04 B on.png - 160x85
0004 - 05 B on.png - 160x85
0005 - 06 B on.png - 160x85
0006 - 07 B on.png - 160x85
0007 - 08 B on.png - 160x85
0008 - 09 B on.png - 160x85
OK


Icon Resources
Libraries
Icons
1
Size
NumericArt1
BatteryLoading
0000 - 01 B on.png - 160x85
0001 - 02 B on.png - 160x85
0002 - 03 B on.png - 160x85
0003 - 04 B on.png - 160x85
0004 - 05 B on.png - 160x85
0005 - 09 B on.png - 160x85
0006 - 10 B on.png - 160x85
0007 - 06 B on.png - 160x85
0008 - 07 B on.png - 160x85
OK

Multiple Icons can be selected using **(CTRL+LMB)**, **(SHIFT+LMB)**, or **(SHIFT+[UP/DOWN/LEFT/RIGHT])**.

3.2.4 Naming Rules

The name of an **Icon** may contain any characters, but it’s recommended to avoid:

- Special characters, e.g.: `# % ¨ $ / \`
- Punctuation marks and accentuation, e.g.: `; . , á, ç`

The name of an **Icon Library** must contain only **alphanumeric** characters and the **underscore** (_) character.

If you wish that the icon images are automatically sorted in a specific order when you add them to an Icon Library, you can prefix a number to the image names.

⚠	Caution	We don’t recommend using number-prefixed file names. It may cause the need of extensive reworks if the prefixes must be changed in the future. If prefixes *are* used, consider leaving gaps between in the indexing values.

Examples:

- 000 Battery Empty
- 001 Battery Half
- 002 Battery Full

It’s important that the number has enough leading zeros (zeros before the first significant digit), because in an alphabetic sort, “10 Battery” comes before “2 Battery”.

3.2.5 Accepted File Formats

Accepted image file formats: BMP, JPEG, JFIF, TIF, PNG, GIF.

ℹ	Info	While Image Resources will automatically resize when the project resolution is changed, Icon Resources are not resized. If you need smaller or larger icons, the original images must be resized.

3.3 Font Resources

Several Interface Objects display information using fonts. Proculus LCMs require a specific font format, and UnicView AD has built-in tools to create such font files. It also comes with a default, multi-sized general-purpose font.

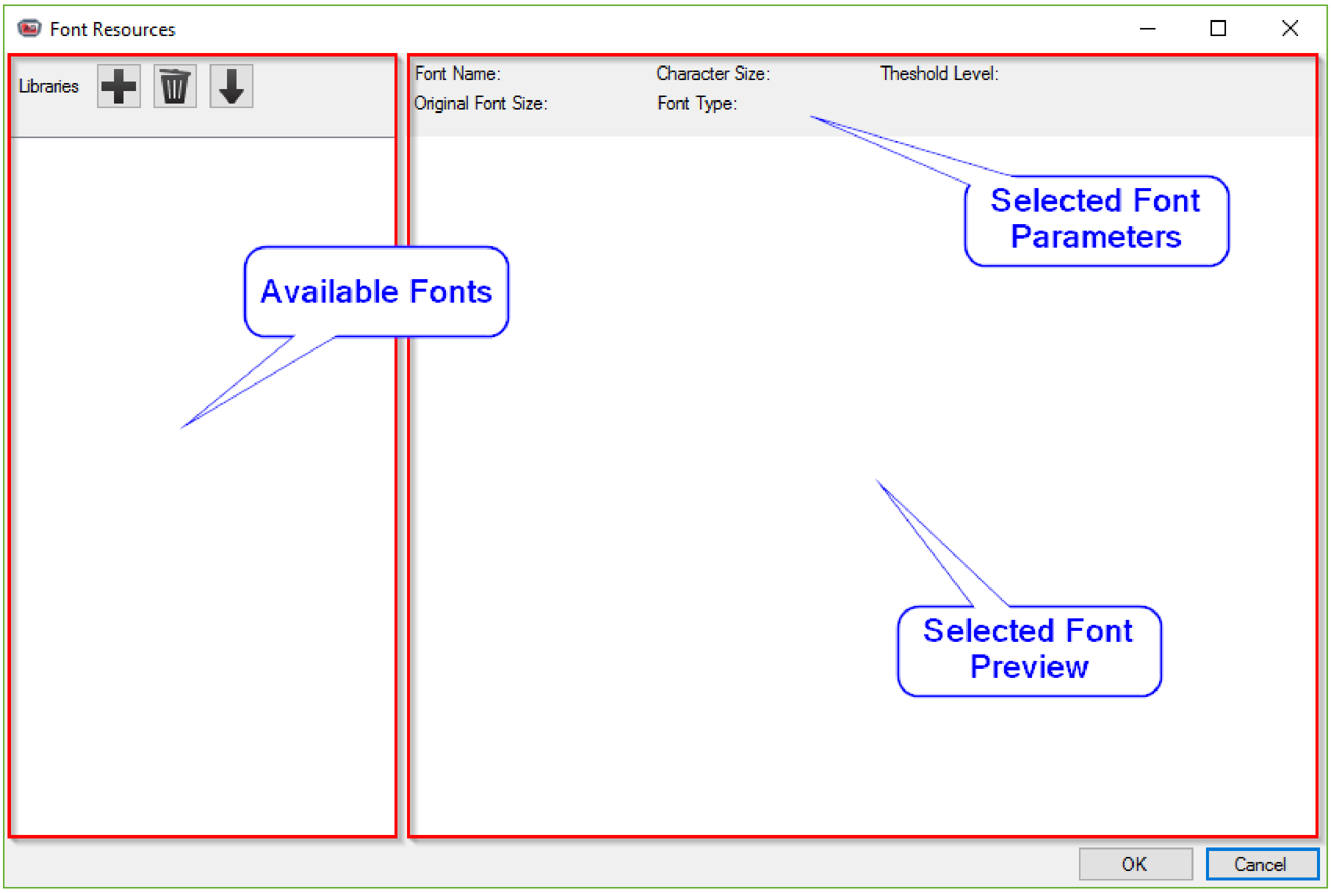
3.3.1 Adding and Removing Font Resources

To view the Font Resources window, click the Ab button on the Resources menu or Resources bar. All projects have a default font resource that can't be removed, named "DefaultFont.hzk". This font is not visible on the Font Resources window.

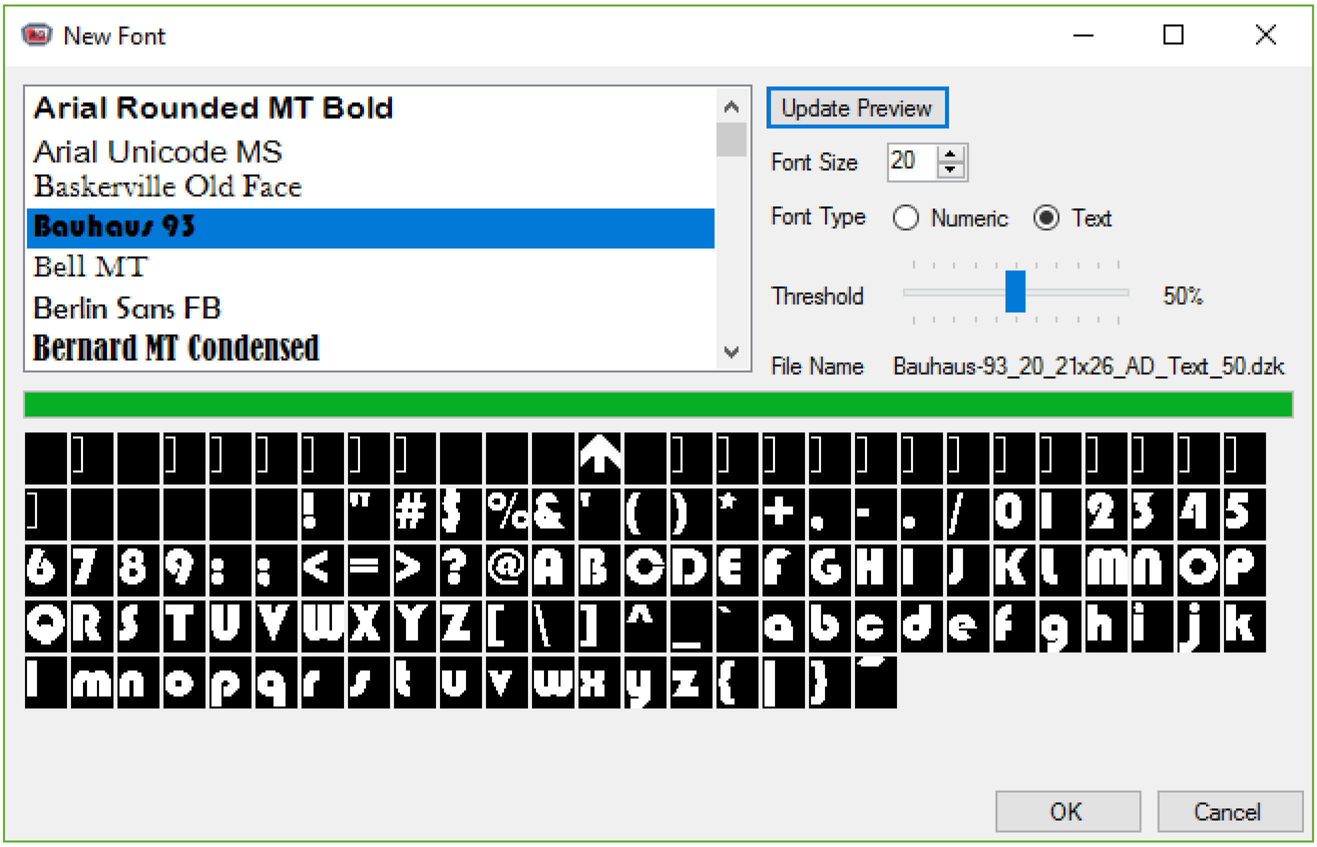
ℹ Info After editing any project resource, compile the Resources, using T or T.

⚠ Caution Image addition/removal can't be undone.

To add fonts to the project, click the ✚ button, on the left panel (Libraries Panel), and the New Font window will appear.



On the New Font window, you can choose the parameters of the font to be created. The Character Size and Font Name parameters are automatically adjusted.



New Font

Font Size: 20

Font Type: ○ Numeric ● Text

Threshold: 50%

File Name: Bauhaus-93_20_21x26_AD_Text_50.dzk

The following table describes each parameter:

Parameter	Description
Font Name	It’s the original font used to create a LCM font file. The left-side panel shows all fonts currently installed on your operating system. If a font is missing, it’s not supported. Only **TrueType** fonts are supported.
Font Size	The Windows font size used to create the font.
Font Type	Numeric or Text font. See details in section “3.3.2 - Numeric and Text font types”.
Threshold Level	The transparency detection level. See details in section “0 - Threshold Level”.
File Name	The name of the generated font file. Automatically generated. See details in section “0 - Naming Rules”.
Character Size	The character box size of the generated font, in pixels. It’s automatically adjusted, based on the selected font.

All the font files are stored in the “Fonts” folder.

Fonts must have unique names, and if you try to add a font with the exact same parameters of an existing one, it will not be added.

ℹ Info Click the [Update Preview] button before creating the font, to ensure it is correct.

To remove a font from the project, click the 🗑 button.

To import existing font files, click the ⬇ button. Important: To successfully import a font file, it must comply to the Font Resource Naming Rules.

3.3.2 Numeric and Text font types

Numeric fonts are used on Numeric Input, RTC Input, Numeric Display, Hex Display and RTC Display.

Since a numeric font's character box is justified to accommodate the digit characters (0 through 9), if alphabetic characters are needed (to show a value's unit, for example), you must use a font that has those characters with similar widths and heights.

ℹ Info When displaying numeric data, you can use the 4.3.7 Numeric Art Display Variable to achieve beautiful results.

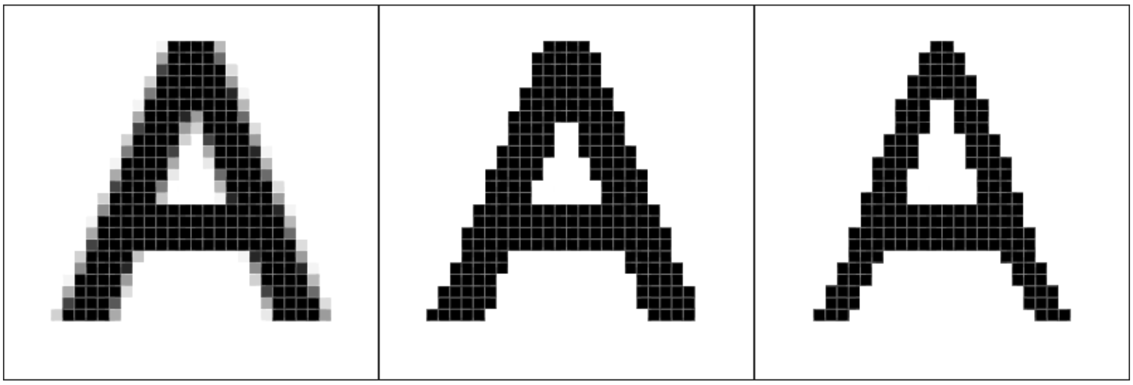
ℹ Info Monospaced fonts are ideal to achieve great results, especially when using alphabetic characters in numeric fonts.

Text fonts are used on Text Input, Text Display and Table Display.

The character box is justified to accommodate all visible characters.

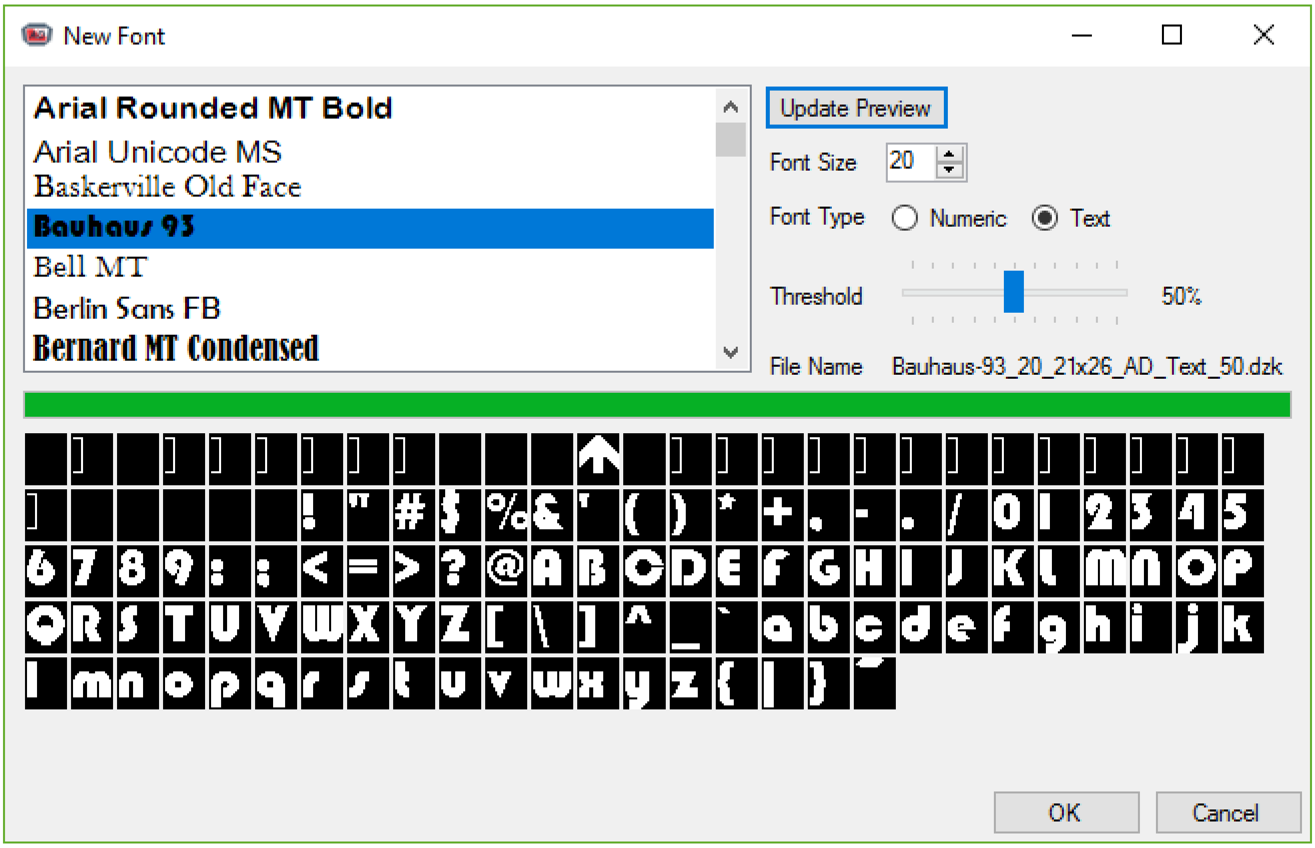
3.3.3 Threshold Level

Proculus LCMs don’t support font anti-aliasing. As such, when creating a new font, the generator must analyze the transparency level of each pixel, and decide if it will be an opaque pixel or an invisible pixel. The level of transparency above which the pixel is considered opaque is the **threshold level**.



Take the “A” character above, for example. The left character is the original, anti-aliased one. It has semi-transparent pixels, and the threshold level will determine which ones will turn to black and which ones will turn to white. The middle character was converted using a **lower** threshold level, making it bolder, darker. The right character was converted using a **higher** threshold level, making it thinner, lighter.

You can use the threshold level to fine-tune the appearance of the font, as seen on the following examples:



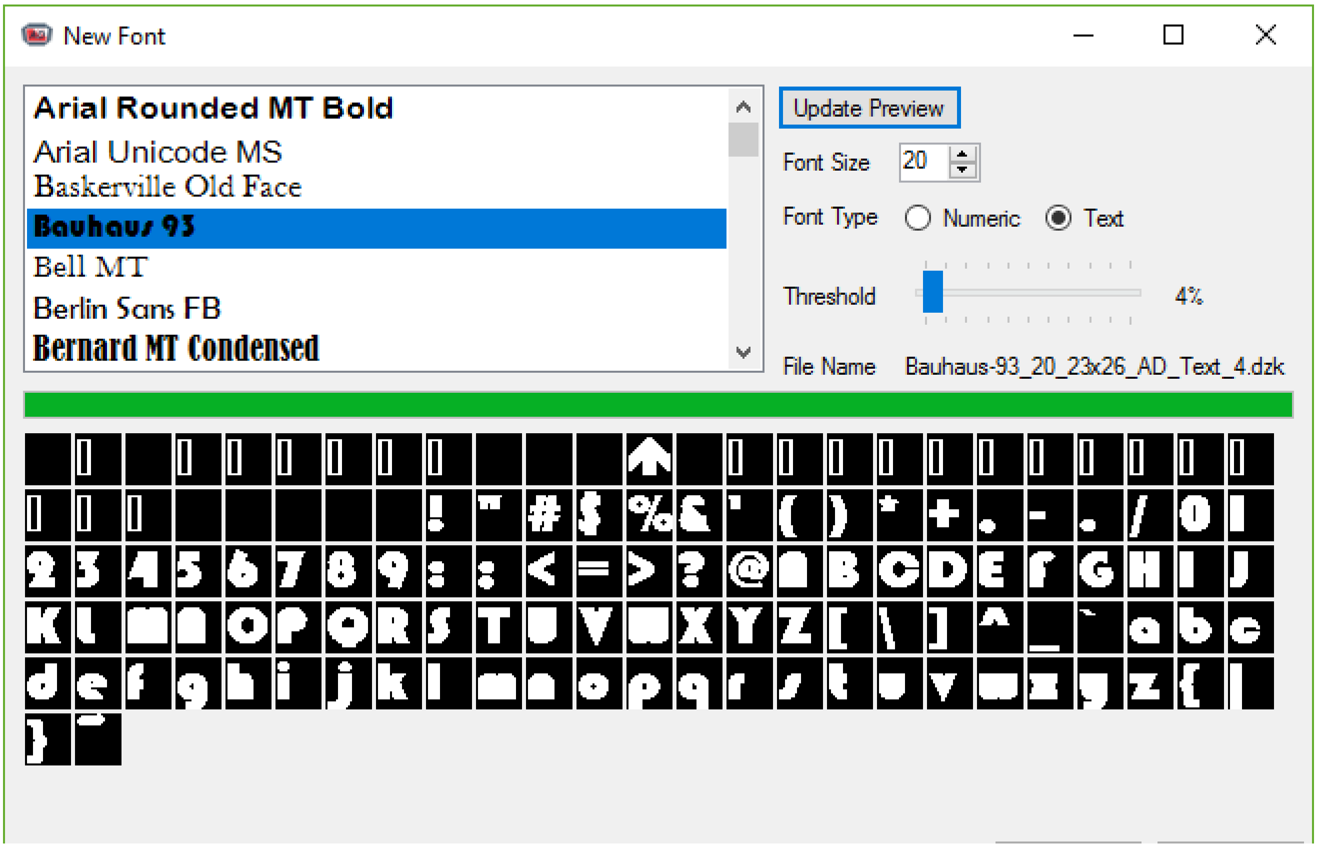
Arial Rounded MT Bold
Arial Unicode MS
Baskerville Old Face
Bauhaus 93
Bell MT
Berlin Sans FB
Bernard MT Condensed

Font Size 20

Font Type ○ Numeric ● Text

Threshold 50%

File Name Bauhaus-93_20_21x26_AD_Text_50.dzk



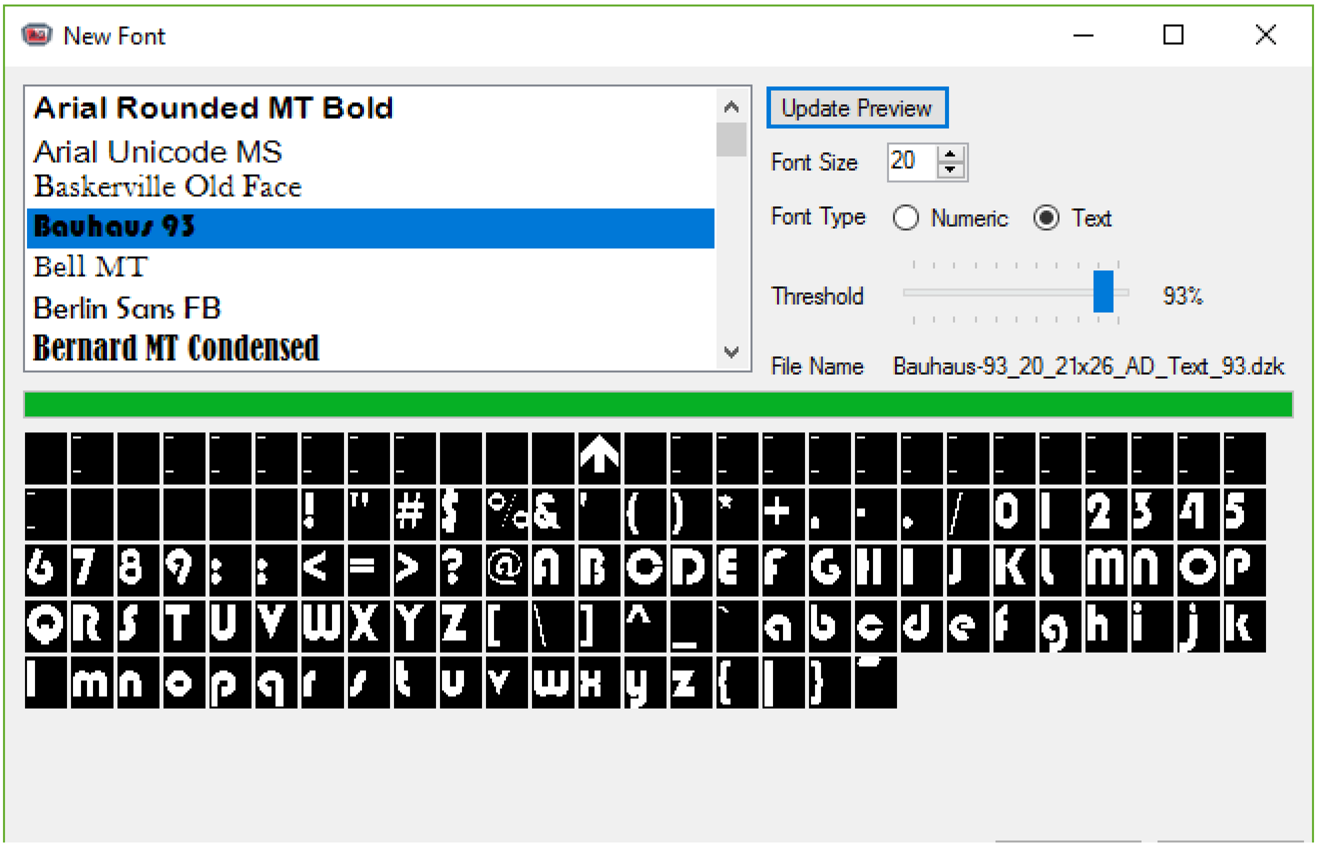
Arial Rounded MT Bold
Arial Unicode MS
Baskerville Old Face
Bauhaus 93
Bell MT
Berlin Sans FB
Bernard MT Condensed

Font Size 20

Font Type ○ Numeric ● Text

Threshold 4%

File Name Bauhaus-93_20_23x26_AD_Text_4.dzk



Arial Rounded MT Bold
Arial Unicode MS
Baskerville Old Face
Bauhaus 93
Bell MT
Berlin Sans FB
Bernard MT Condensed

Font Size 20

Font Type ○ Numeric ● Text

Threshold 93%

File Name Bauhaus-93_20_21x26_AD_Text_93.dzk

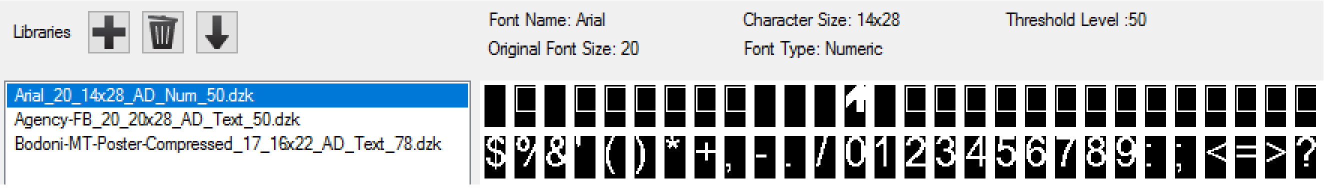
3.3.4 Naming Rules

The default font file is named “DefaultFont.hzk”.

Custom fonts must follow this naming structure:

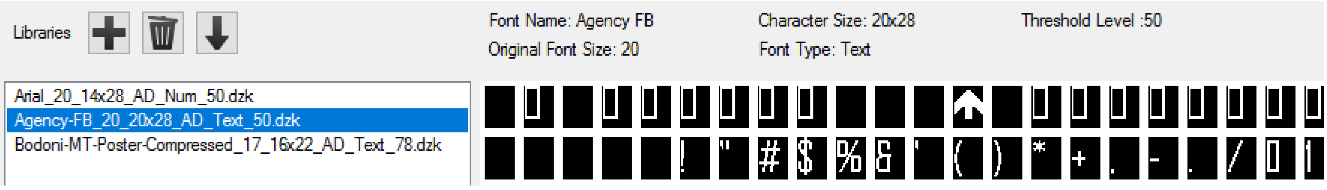
```
<FontName>_<FontSize>_<CharacterSize>_AD_<FontType>_<ThresholdLevel>.dzk
```

Where each field represents the parameters used to create the font. Examples:



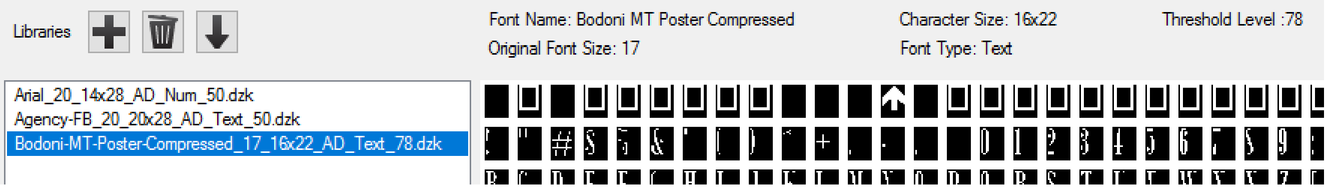
Font Name: Arial
Original Font Size: 20
Character Size: 14x28
Font Type: Numeric
Threshold Level :50

Arial_20_14x28_AD_Num_50.dzk
Agency-FB_20_20x28_AD_Text_50.dzk
Bodoni-MT-Poster-Compressed_17_16x22_AD_Text_78.dzk



Font Name: Agency FB
Original Font Size: 20
Character Size: 20x28
Font Type: Text
Threshold Level :50

Arial_20_14x28_AD_Num_50.dzk
Agency-FB_20_20x28_AD_Text_50.dzk
Bodoni-MT-Poster-Compressed_17_16x22_AD_Text_78.dzk



Font Name: Bodoni MT Poster Compressed
Original Font Size: 17
Character Size: 16x22
Font Type: Text
Threshold Level :78

Arial_20_14x28_AD_Num_50.dzk
Agency-FB_20_20x28_AD_Text_50.dzk
Bodoni-MT-Poster-Compressed_17_16x22_AD_Text_78.dzk

3.3.5 Accepted File Formats

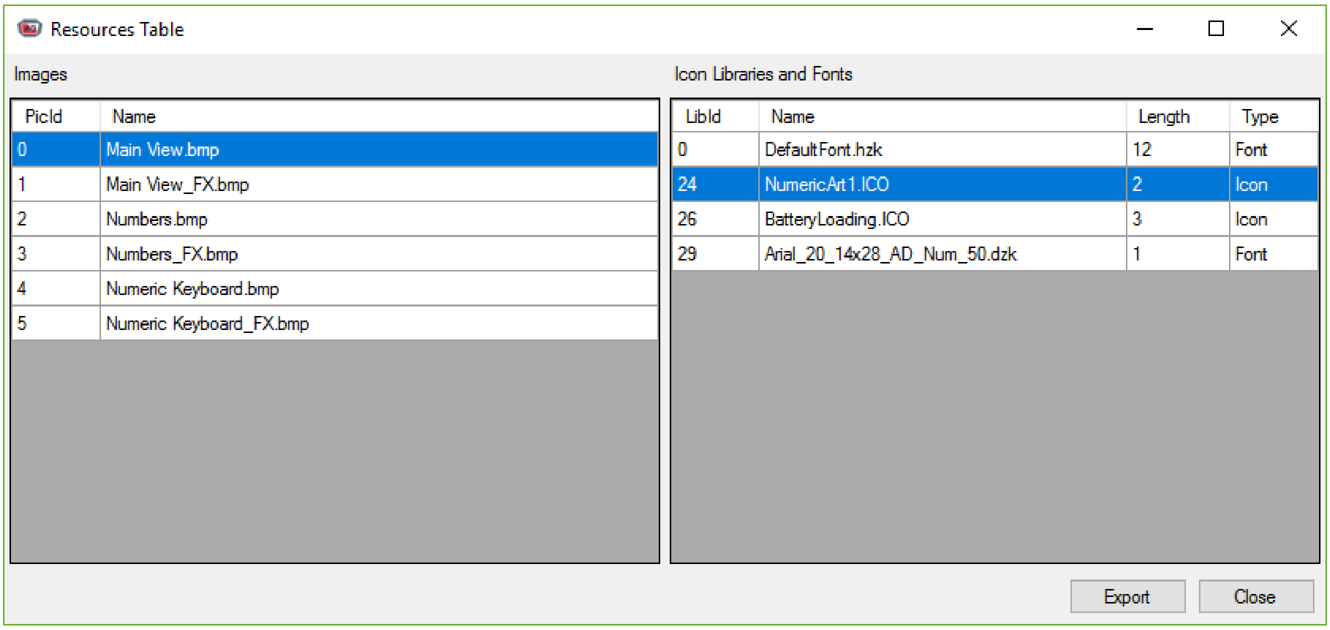
The default font file as the extension HZK. All custom font files have the extension DZK.

3.4 Resource Allocation Table

After compiling the project resources, they're all allocated to certain addresses in their respective memory spaces. These addresses are referred to as **indexes**. The indexes of Image Resources are called **PicIds** (Picture ID), and those of **Icon and Font Resources** are called **LibIds** (Library ID).

Since the control side of an application (a microcontroller of a sale machine, for example) may need to know what are those indexes, UnicView AD provides the Resource Allocation Table, where you can see what resources are allocated to which addresses.

Info For more information on memory spaces and resource placement, refer to the "UnicView AD Development Guide" document.



Images

PicId	Name
0	Main View.bmp
1	Main View_FX.bmp
2	Numbers.bmp
3	Numbers_FX.bmp
4	Numeric Keyboard.bmp
5	Numeric Keyboard_FX.bmp

Icon Libraries and Fonts

LibId	Name	Length	Type
0	DefaultFont.hzk	12	Font
24	NumericArt1.ICO	2	Icon
26	BatteryLoading.ICO	3	Icon
29	Arial_20_14x28_AD_Num_50.dzk	1	Font

PicIds are allocated according to the Screens Panel, and only change when you explicitly change them.

LibIds, on the other hand, are automatically attributed, and may change if you modify the **Icon or Font Resources**.

Caution If your application developer needs to know the PicIds and LibIds, always check the Resource Allocation Table for changes after compiling the project.

4 Interface Objects

On AD LCMs, direct user interaction is provided by **Interface Objects** (or, simply, **Objects**). There are two kinds of Interface Objects:

- Controls – Provide user input.
- Display Variables – Provide visual output to the user.

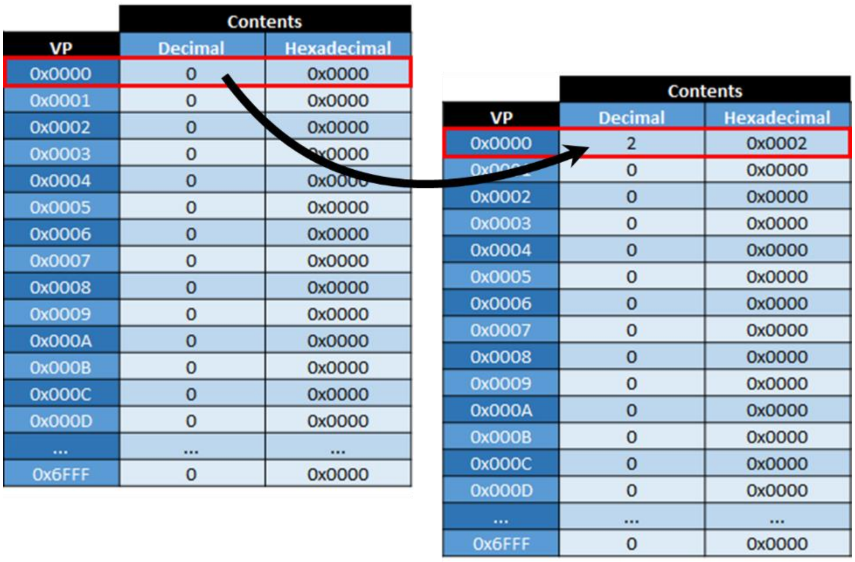
Using both kinds of Interface Objects, you can layout and compose a great human-machine interface (HMI). Each Screen on the LCM has its own set of Objects.

4.1 VP and PP Distribution

Most Interface Objects must be assigned to a **Variable Pointer (VP)**. A VP is an address on the RAM space. Each VP points to a 2-byte (1 word) value.

Info The value stored on a VP is called **VPC (Value Pointer Content)**. It is also denoted as ***VP**.

For example, if an Incremental Input is assigned to VP 0x0000, when it is activated, it will increment the value stored at this VP. Assuming the initial value is 0, after two activations (touches on the Touch Panel), the new value stored on VP x0000 is 2.



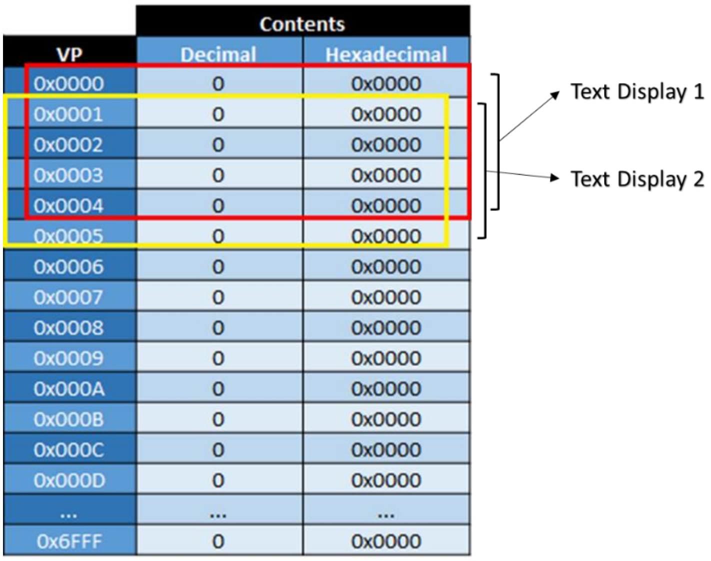
	Contents	
VP	Decimal	Hexadecimal
0x0000	0	0x0000
0x0001	0	0x0000
0x0002	0	0x0000
0x0003	0	0x0000
0x0004	0	0x0000
0x0005	0	0x0000
0x0006	0	0x0000
0x0007	0	0x0000
0x0008	0	0x0000
0x0009	0	0x0000
0x000A	0	0x0000
0x000B	0	0x0000
0x000C	0	0x0000
0x000D	0	0x0000
...	...	...
0x6FFF	0	0x0000

	Contents	
VP	Decimal	Hexadecimal
0x0000	2	0x0002
0x0001	0	0x0000
0x0002	0	0x0000
0x0003	0	0x0000
0x0004	0	0x0000
0x0005	0	0x0000
0x0006	0	0x0000
0x0007	0	0x0000
0x0008	0	0x0000
0x0009	0	0x0000
0x000A	0	0x0000
0x000B	0	0x0000
0x000C	0	0x0000
0x000D	0	0x0000
...	...	...
0x6FFF	0	0x0000

A **Parameter Pointer (PP)** is a VP used to store the parameters of a Display Variable, so that they can be modified by the user at run-time.

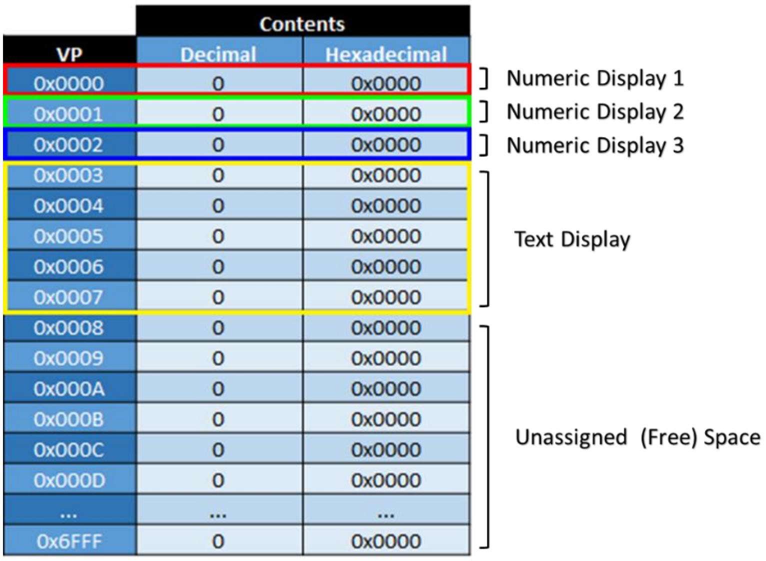
Info If the PP of a Display Variable is set to the **default value** (-1 or 0xFFFF), the parameters of this Display Variable are fixed, and can only be changed by modifying the project. If you set the PP to a valid RAM address, the parameters are modifiable, and are initialized along the rest of the RAM.

Since there are no restrictions on address assignment, conflicts may show up on a project. For example, usually, you should not assign two Text Displays with Text Lengths greater than 2 to subsequent VPs, because their data will overwrite each other:



	Contents	
VP	Decimal	Hexadecimal
0x0000	0	0x0000
0x0001	0	0x0000
0x0002	0	0x0000
0x0003	0	0x0000
0x0004	0	0x0000
0x0005	0	0x0000
0x0006	0	0x0000
0x0007	0	0x0000
0x0008	0	0x0000
0x0009	0	0x0000
0x000A	0	0x0000
0x000B	0	0x0000
0x000C	0	0x0000
0x000D	0	0x0000
...	...	...
0x6FFF	0	0x0000

To avoid this problem, always keep in mind how many VPs an Interface Object takes, space them properly:

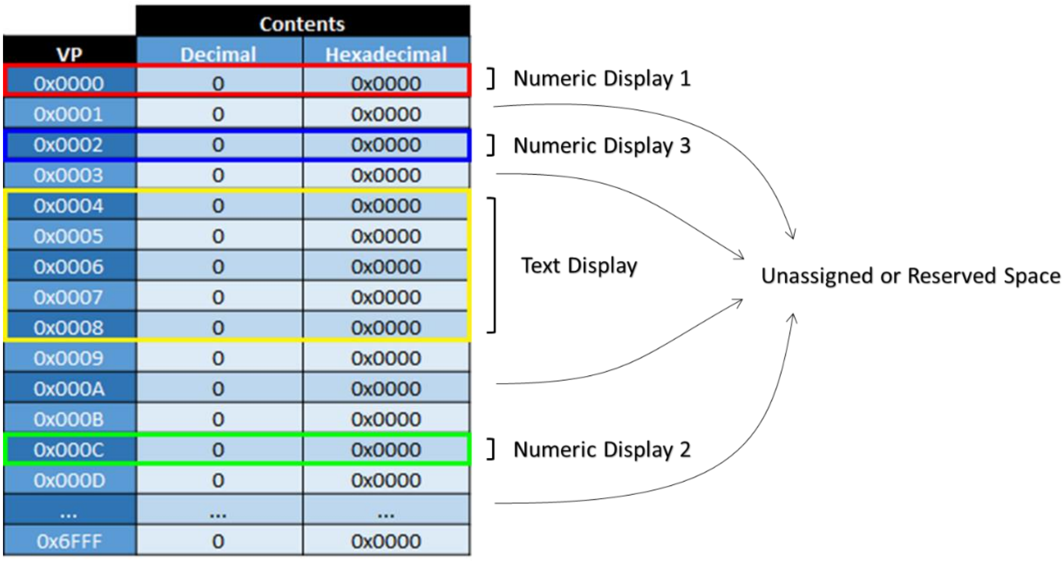


	Contents	
VP	Decimal	Hexadecimal
0x0000	0	0x0000
0x0001	0	0x0000
0x0002	0	0x0000
0x0003	0	0x0000
0x0004	0	0x0000
0x0005	0	0x0000
0x0006	0	0x0000
0x0007	0	0x0000
0x0008	0	0x0000
0x0009	0	0x0000
0x000A	0	0x0000
0x000B	0	0x0000
0x000C	0	0x0000
0x000D	0	0x0000
...	...	...
0x6FFF	0	0x0000

Info

While unintended address overlapping is a project error, it’s sometimes desirable to have overlapping VPs. Multiple Display Variables may be assigned to the same VP, for example, to display the same value in different locations or formats.

To improve scalability and avoid unintended data overlapping, we recommend planning your project's address allocation beforehand, so that you can reserve addresses between Objects, and give them plenty of space to expand:



	Contents	
VP	Decimal	Hexadecimal
0x0000	0	0x0000
0x0001	0	0x0000
0x0002	0	0x0000
0x0003	0	0x0000
0x0004	0	0x0000
0x0005	0	0x0000
0x0006	0	0x0000
0x0007	0	0x0000
0x0008	0	0x0000
0x0009	0	0x0000
0x000A	0	0x0000
0x000B	0	0x0000
0x000C	0	0x0000
0x000D	0	0x0000
...	...	...
0x6FFF	0	0x0000

Each project requires a different strategy for addressing rules, but here are some examples:

- All Objects will have even VPs.
- Text Displays will always have VPs 0x3#00 (0x3100, 0x3200, etc.);
- Other Objects will have VPs 0x0##0 (0x0100, 0x0110, 0x0200, etc.), and have PPs 0x5##0 (0x5100, 0x5110, 0x5200, etc.);

By using addressing rules, you can greatly reduce mapping issues and reworks when scaling up or down your projects.

Info	Always verify how many VPs and PPs (if used) an Interface Object requires.

4.2 Controls

Controls provide direct user input interaction. They can be considered as buttons. All Controls can be activated by a physical touch from the user, and most of them can be activated via Serial Communication. They are usually employed to modify the contents of the RAM space, although they may be used purely as Serial Communication triggers.

Please refer to the document "UnicView AD Development Guide" for details on the address mapping of each Control.

ℹ	Info	Controls may be user-activated or software-activated (via Serial Communication).

⚠	Caution	Controls can't have overlapping areas.

4.2.1 Common Properties

The properties summarized in the following table are common to all Controls, and are omitted in the next sections.

Property	Description	Notes
Type	Type of the object.	Read Only
Name	Friendly name for this control.	
Area	Area of the control.	Uses the notation X;Y;Width;Height Uses: Area Quick Editor
Description	Descriptive text about this control.	
JumpId	PicId of the screen to jump to.	
FxId	PicId of the screen where the "pressed" effects for this control reside.	
Auto-Send Data	Sets whether this control automatically sends data through the serial port when activated.	
Software Control	Sets whether this control can activated by software.	Consult the "UnicView AD Development Guide" document for details.
Software Control Code	Code to be written on Register 0x4F to activate this control via software.	Consult the "UnicView AD Development Guide" document for details.
Play Audio Segment	Sets whether this control plays audio when activated.	
Audio Length	Number of audio segments to play.	
AudioId	Index of the first audio segment to play.	

4.2.2 Basic Touch

Used to implement basic Screen navigation and to create the keys on keyboards.

Property	Description	Notes
ASCII Code	ASCII code for the control. Only needed when creating keyboards.	

4.2.3 Set Value

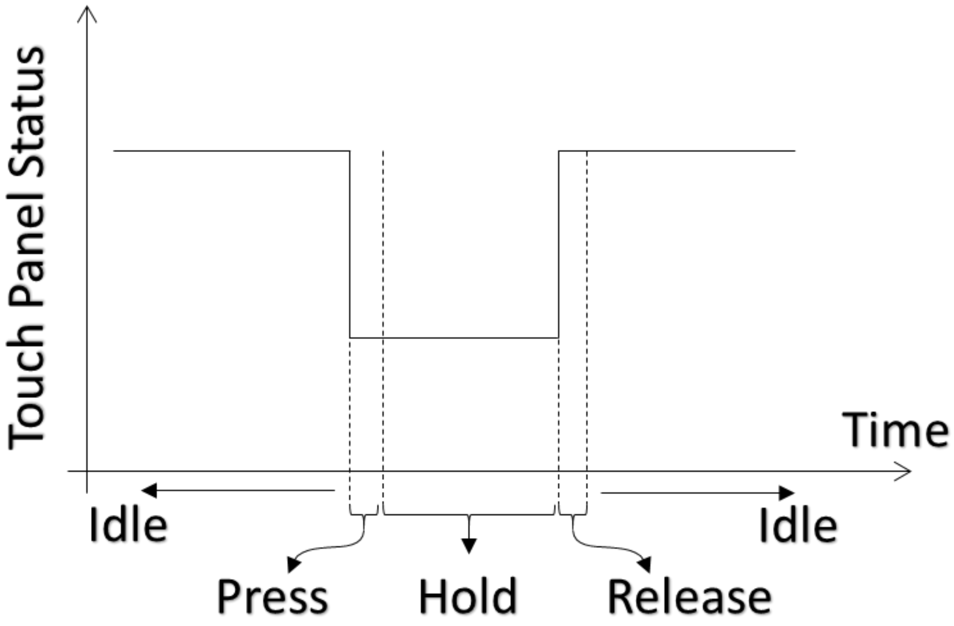
Used to implement a button that writes a value to a VP when pressed, and to signal events to an external controller.

Property	Description	Notes
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
VP Mode	Value Memory Size.	
Return Value	Value to be written on the selected VP.	

4.2.4 Touch Status

Used to implement push-buttons. Can be configured to execute different operations during its three states:

- Pressed
- Held down
- Released



Since it has three possible active estates (Pressed, Held, Released), you can use it as a push-button, and send Serial Commands that activate a motor only while the user is holding the button pressed, for example.

Property	Description	Notes
Op. Mode (Press)	Operation mode for when this control is pressed. 0. Does nothing. 1. Copies *[Source Address (Press)] to RAM at [Target Address (Press)]. 2. Sends *[Source Address (Press)] to the Serial Port. 3. Copies *[Source Address (Press)] to Control Register Memory at [Target Address (Press)].	
Source Address (Press)	Source Address for copying (Press).	
Target Address (Press)	Target Address for copying (Release).	
Length (Press)	Number of bytes to copy (Press). When in Operation Mode 1, this number must be even.	

Op. Mode (Hold)	Operation mode for when this control is held down. 0. Does nothing. 1. Copies *[Source Address (Hold)] to RAM at [Target Address (Hold)]. 2. Sends *[Source Address (Hold)] to the Serial Port. 3. Copies *[Source Address (Hold)] to Control Register Memory at [Target Address (Hold)].	
Source Address (Hold)	Source Address for copying (Hold).	
Target Address (Hold)	Target Address for copying (Release).	
Length (Hold)	Number of bytes to copy (Hold). When in Operation Mode 1, this number must be even.	
Op. Mode (Release)	Operation mode for when this control is released. 0. Does nothing. 1. Copies *[Source Address (Release)] to RAM at [Target Address (Release)]. 2. Sends *[Source Address (Release)] to the Serial Port. 3. Copies *[Source Address (Release)] to Control Register Memory at [Target Address (Release)].	
Source Address (Release)	Source Address for copying (Release).	
Target Address (Release)	Target Address for copying (Release).	
Length (Release)	Number of bytes to copy (Release). When in Operation Mode 1, this number must be even.	

4.2.5 Numeric Input

Opens a keyboard for numeric values input. It uses fixed-point integer values.

Property	Description	Notes
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
Library	Font or Icon Library.	
Font Width	Font width, in pixels.	
Font Color	Color of the font.	
Cursor Color	Color of the input cursor.	
Image from other PicId	Indicates if the image source is in another PicId or in the current one.	
Source PicId	PicId of the Screen used as image source for this control.	
Source Area	Source area. The area where the keyboard is.	
Target Top-Left Point	Top-left coordinates of the pasting area.	
Cursor Origin	Coordinates of the top-left corner of the cursor's origin.	
Integer Digits	Number of digits to the left of the decimal separator.	
Decimal Digits	Number of digits to the right of the decimal separator.	
Show Characters	Sets whether the characters are to be displayed. Otherwise, they'll be masked by asterisks (*).	
Limit Value Range	Limit the control's accepted input values according to the range set.	
Max Value	Maximum value accepted by the control.	
Min Value	Minimum value accepted by the control.	
VP Mode	Value Memory Size.	

4.2.6 Text Input

Opens a keyboard for alphanumeric (text) values input.

Property	Description	Notes
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
Library	Font or Icon Library.	
Font Width	Font width, in pixels.	
Font Height	Font height, in pixels.	
Font Color	Color of the font.	
Cursor Color	Color of the input cursor.	
Re-edit Input	Indicates whether the input is blank or shows the current value when the keyboard is open.	
Update Input Status	If true, the address [VP-1] is used to indicate the input status of this control: [VP-1].High = Has the value 0x5A when the keyboard is closed Other Values indicate it's open. [VP-1].Low = Last successful input length, in bytes.	
Image from other PicId	Indicates if the image source is in another PicId or in the current one.	
Source PicId	PicId of the Screen used as image source for this control.	
Source Area	Source area. The area where the keyboard is.	
Text Area	Area where the text will be displayed.	
Target Top-Left Point	Top-left coordinates of the pasting area.	
Hide Characters	Sets whether the characters are to be masked by asterisks (*). Otherwise, they'll be displayed normally.	
Text Length (words)	Text Length (number of characters = 2*[Text Length]).	

4.2.7 Incremental Input

Used to implement a button that increments the content of a VP.

Property	Description	Notes
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
Increment Sign	Sign of the increment (positive or negative).	
Increment Value	Value (step) of each increment (or decrement).	
Limit Value Range	Limit the control's accepted input values according to the range set.	
Max Value	Maximum value accepted by the control.	
Min Value	Minimum value accepted by the control.	
Continuous Increment	Sets whether the value is incremented (or decremented) while the user holds it pressed.	
Loop	Sets whether the value loops around the range set when the maximum or minimum value is reached.	
VP Mode	Value Memory Size.	

4.2.8 Slider Input

Used to implement a sliding button that dynamically changes the content of a VP.

Property	Description	Notes
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
Slider Orientation	Indicates whether the slider is vertical or horizontal.	
Max Value	Maximum value accepted by the control.	
Min Value	Minimum value accepted by the control.	
VP Mode	Value Memory Size.	

4.2.9 RTC Input

Opens a keyboard to modify the current date and time.

Property	Description	Notes
Library	Font or Icon Library.	
Font Width	Font width, in pixels.	
Font Color	Color of the font.	
Cursor Color	Color of the input cursor.	
Image from other PicId	Indicates if the image source is in another PicId or in the current one.	
Source PicId	PicId of the Screen used as image source for this control.	
Source Area	Source area.	
Target Top-Left Point	Top-left coordinates of the pasting area.	
Cursor Origin	Coordinates of the top-left corner of the cursor's origin.	

4.2.10 Popup

Opens a keyboard in a popup window-style.

Property	Description	Notes
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
Source PicId	PicId of the Screen used as image source for this control.	
Source Area	Source area.	
Target Top-Left Point	Top-left coordinates of the pasting area.	
VP Mode	Value Memory Size.	

4.3 Display Variables

Display Variables provide visual interaction to the users. They function as numeric, textual and graphic indicators. Display Variables are always associated to a memory address, and show the contents of such address in some human-readable form.

Please refer to the document “UnicView AD Development Guide” for details on the address mapping of each Display Variable.

⚠ Caution There is a maximum number of Display Variables per Screen. This number (64 or 128) is set on the System Configuration window.

4.3.1 Common Properties

The properties summarized in the following table are common to all Display Variables, and are omitted in the next sections.

Property	Description	Notes
Type	Type of the object.	Read Only.
Name	Friendly name for this display variable.	
Area	Area of the display variable.	
Description	Descriptive text about this display variable.	

⚠ Caution Make sure to reserve enough VPs for each Display Variable. Consider their data sizes and their PP lengths (when PPs are used).

4.3.2 Dynamic Icon

Used to show an Icon from an Icon Library. The current Icon is determined by the value of the VP.

PP Length: 8 VPs.

Property	Description	Notes
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
PP	Parameter Pointer. RAM Address for run-time parameters; shares the same memory-space as VP.	Consult the “UnicView AD Development Guide” document for details.
Library	Font or Icon Library.	
Transparency	Transparency mode for the icon. Opaque = The icons are shown with a solid background. Transparent = The icons are shown without the background color (the color of the top-left pixel).	
Icon Max	Icon associated to the Maximum Value.	
Icon Min	Icon associated to the Minimum Value.	
Value Max	Maximum value for valid icons.	
Value Min	Minimum value for valid icons.	
Initial Value	Initial Value for this display variable.	

4.3.3 Animated Icon

Used to show a loop animation of Icons from an Icon Library. The animation state is determined by the value of the VP.

PP Length: 10 VPs.

⚠ Caution Animated Icon uses **2 VPs**. Make sure to reserve VPs accordingly. **Do not** write in the second VP.

Property	Description	Notes
Type	Type of the object.	
Name	Friendly name for this display variable.	
Area	Area of the display variable.	
Description	Descriptive text about this display variable.	
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	Animated Icon uses 2 VPs (VP and VP+1). Don’t write in the second VP.
PP	Parameter Pointer. RAM Address for run-time parameters; shares the same memory-space as VP.	Consult the “UnicView AD Development Guide” document for details.
Library	Font or Icon Library.	
Transparency	Transparency mode for the icon. Opaque = The icons are shown with a solid background. Transparent = The icons are shown without the background color (the color of the top-left pixel).	
Value Stop	When VP contains this value, the animation will stop.	
Value Start	When VP contains this value, the animation will start.	
Restart Animation From First Icon	Indicates whether the animation should always start from the first frame when restarting.	
Icon Stop	Icon displayed when the animation is stopped.	
Icon Start	Icon displayed at the first frame of the animation.	
Icon End	Icon displayed at the last frame of the animation.	
Initial Value	Initial Value for this display variable.	

4.3.4 Slider Display

Used to show an Icon that moves along a given axis (horizontal or vertical) based on the value of the VP. Typically used in linear graphs, or in conjunction with Slider Inputs.

PP Length: 10 VPs.

Property	Description	Notes
Type	Type of the object.	
Name	Friendly name for this display variable.	
Area	Area of the display variable.	
Description	Descriptive text about this display variable.	
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
PP	Parameter Pointer. RAM Address for run-time parameters; shares the same memory-space as VP.	Consult the “UnicView AD Development Guide” document for details.
Library	Font or Icon Library.	
Icon	Icon for the slider handle.	
Transparency	Transparency mode for the icon. Opaque = The icons are shown with a solid background. Transparent = The icons are shown without the background color (the color of the top-left pixel).	
Value Max	Maximum value for the slider.	
Value Min	Minimum value for the slider.	
Slider Orientation	Indicates whether the slider is vertical or horizontal.	
Icon Offset	Offset adjustment for the handle icon. Offsets on the X axis for horizontal sliders, and on the Y axis for vertical sliders.	
Value Type	Value Type	
Initial Value	Initial Value for this display variable.	

4.3.5 Rotating Icon

Used to show an Icon that pivots around a given rotation center, based on the value of the VP. Typically used in radial graphs, like speedometers and dials.

PP Length: 12 VPs.

Property	Description	Notes
Type	Type of the object.	
Name	Friendly name for this display variable.	
Area	Area of the display variable.	
Description	Descriptive text about this display variable.	
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
PP	Parameter Pointer. RAM Address for run-time parameters; shares the same memory-space as VP.	Consult the “UnicView AD Development Guide” document for details.
Library	Font or Icon Library.	
Icon	Icon to be rotated.	
Transparency	Transparency mode for the icon. Opaque = The icons are shown with a solid background. Transparent = The icons are shown without the background color (the color of the top-left pixel).	
Icon Center	Center of rotation on the icon.	This center rotates around the **Top-Right** corner of the Rotating Icon Object area.
Value Max	Value associated to the maximum angle.	
Value Min	Value associated to the minimum angle.	
Angle Max	Maximum angle of rotation (given in 0.5°).	
Angle Min	Minimum angle of rotation (given in 0.5°).	
Value Type	Value Type	
Initial Value	Initial Value for this display variable.	

4.3.6 Bitwise Icon

Used to show fixed and/or animated Icons, according to a bit flag value on the VP. The value of each bit represents the state of a single Icon, and many Icons can be shown, in different states, based on the VPC.

Typically used to display several alarms at once, or to implement bar graphs.

PP Length: 12 VPs.

<table>
<tr><th>Property</th><th>Description</th><th>Notes</th></tr>
<tr><td><b>Type</b></td><td>Type of the object.</td><td></td></tr>
<tr><td><b>Name</b></td><td>Friendly name for this display variable.</td><td></td></tr>
<tr><td><b>Area</b></td><td>Area of the display variable.</td><td></td></tr>
<tr><td><b>Description</b></td><td>Descriptive text about this display variable.</td><td></td></tr>
<tr><td><b>VP</b></td><td>Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".</td><td></td></tr>
<tr><td><b>PP</b></td><td>Parameter Pointer. RAM Address for run-time parameters; shares the same memory-space as VP.</td><td>Consult the “UnicView AD Development Guide” document for details.</td></tr>
<tr><td><b>Library</b></td><td>Font or Icon Library.</td><td></td></tr>
<tr><td><b>Transparency</b></td><td>Transparency mode for the icon.
Opaque = The icons are shown with a solid background.
Transparent = The icons are shown without the background color (the color of the top-left pixel).</td><td></td></tr>
<tr><td><b>Display Mode</b></td><td>The following table describes what icons are shown when each bit value is either 0 or 1.
<table>
<tr><th rowspan="2">Mode</th><th colspan="2">Bit Value</th></tr>
<tr><th>0</th><th>1</th></tr>
<tr><td>0</td><td>Icon Start/Stop 0</td><td>Icon Start/Stop 1</td></tr>
<tr><td>1</td><td>Icon Start/Stop 0</td><td>None</td></tr>
<tr><td>2</td><td>Icon Start/Stop 0</td><td>Animation:
Icon Start/Stop 1 ->
Icon End 1</td></tr>
<tr><td>3</td><td>None</td><td>Icon Start/Stop 1</td></tr>
<tr><td>4</td><td>None</td><td>Animation:
Icon Start/Stop 1 ->
Icon End 1</td></tr>
<tr><td>5</td><td>Animation:
Icon Start/Stop 0 ->
Icon End 0.</td><td>Icon Start/Stop 1</td></tr>
<tr><td>6</td><td>Animation:
Icon Start/Stop 0 ->
Icon End 0</td><td>None</td></tr>
<tr><td>7</td><td>Animation:
Icon Start/Stop 0 ->
Icon End 0</td><td>Animation:
Icon Start/Stop 1 ->
Icon End 1</td></tr>
</table>
</td><td></td></tr>
<tr><td><b>Alignment Mode</b></td><td>Alignment Mode.
Horizontal or Vertical, and Reserved or Unreserved Space.</td><td></td></tr>
<tr><td><b>Active Bits</b></td><td>Indicates which bits are displayed.</td><td></td></tr>
<tr><td><b>Spacing</b></td><td>The size reserved for each icon, in pixels.</td><td></td></tr>
</table>

Icon Start/Stop 0	Modes 0, 1, 2; Bit value = 0; Icon shown. Modes 5, 6, 7; Bit value = 0: First icon in animation mode.	
Icon End 0	Modes 5, 6, 7; Bit value = 0: Last icon in animation mode.	
Icon Start/Stop 1	Modes 0, 3, 5; Bit value = 1; Icon shown. Modes 2, 4, 7; Bit value = 1: First icon in animation mode.	
Icon End 1	Modes 2, 4, 7; Bit value = 1: Last icon in animation mode.	
Initial Value	Initial Value for this display variable.	

⚠ Caution Animated Icon uses **3 VPs**. Make sure to reserve VPs accordingly. **Do not** write in the second or third VPs.

4.3.7 Numeric Art

Works like a Numeric Display, using Icons instead of Fonts. Typically used when you need to display numeric information that needs an anti-aliased look. It uses fixed-point integer values.

PP Length: 7 VPs.

Property	Description	Notes
Type	Type of the object.	
Name	Friendly name for this display variable.	
Area	Area of the display variable.	
Description	Descriptive text about this display variable.	
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
PP	Parameter Pointer. RAM Address for run-time parameters; shares the same memory-space as VP.	Consult the “UnicView AD Development Guide” document for details.
Library	Font or Icon Library.	
Icon 0	Icon associated to the digit 0. Each subsequent icon on the library, in order, will be associated to the sequence [0123456789-.]	
Transparency	Transparency mode for the icon. Opaque = The icons are shown with a solid background. Transparent = The icons are shown without the background color (the color of the top-left pixel).	
Text Alignment	Text alignment mode.	
Integer Digits	Number of digits to the left of the decimal separator.	
Decimal Digits	Number of digits to the right of the decimal separator.	
Value Type	Value Type	
Initial Value	Initial Value for this display variable.	

4.3.8 Numeric Display

Used to display numeric information. It uses fixed-point integer values.

PP Length: 8 to 19 VPs.

Property	Description	Notes
Type	Type of the object.	
Name	Friendly name for this display variable.	
Area	Area of the display variable.	
Description	Descriptive text about this display variable.	
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
PP	Parameter Pointer. RAM Address for run-time parameters; shares the same memory-space as VP.	Consult the “UnicView AD Development Guide” document for details.
Library	Font or Icon Library.	
Font Width	Font Width	
Font Color	Color of the font.	
Text Alignment	Text alignment mode.	
Unity	Unity to be displayed after the numeric value.	
Integer Digits	Number of digits to the left of the decimal separator.	
Decimal Digits	Number of digits to the right of the decimal separator.	
Value Type	Value Type	
Initial Value	Initial Value for this display variable.	

4.3.9 Text Display

Used to display textual information.

PP Length: 13 VPs.

Property	Description	Notes
Type	Type of the object.	
Name	Friendly name for this display variable.	
Area	Area of the display variable.	
Description	Descriptive text about this display variable.	
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
PP	Parameter Pointer. RAM Address for run-time parameters; shares the same memory-space as VP.	Consult the "UnicView AD Development Guide" document for details.
Library	Font or Icon Library.	
Font Width	Font Width.	
Font Height	Font Height.	
Font Encoding	Font Encoding.	For **DefaultFont**, use GB2312. For **custom fonts**, the typical encoding is 8 Bit.
Font Color	Color of the font.	
Text Length	Text length, in characters.	
Fixed Spacing	Indicates whether characters have fixed width or automatically remove excess space.	
Character Spacing	Spacing between characters, in pixels.	
Line Spacing	Spacing between lines, in pixels.	
Value Type	Value Type	
Initial Value	Initial Value for this display variable.	

4.3.10 Image Animation

Used to create an animation of Screens. Can be implemented via Serial Communication as a series of Screen jumps.

PP Length: 4 VPs.

Property	Description	Notes
Type	Type of the object.	
Name	Friendly name for this display variable.	
Area	Area of the display variable.	
Description	Descriptive text about this display variable.	
PP	Parameter Pointer. RAM Address for run-time parameters; shares the same memory-space as VP.	Consult the “UnicView AD Development Guide” document for details.
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
First Frame	First animation frame.	
Last Frame	Last animation frame.	
Frame Time	Indicates for how long each animation frame is shown, in multiples of 8 ms.	

4.3.11 Hex Display

Used to display numeric information in hexadecimal format, with optional digit separators.

PP Length: 6 to 21 VPs.

Property	Description	Notes
Type	Type of the object.	
Name	Friendly name for this display variable.	
Area	Area of the display variable.	
Description	Descriptive text about this display variable.	
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
PP	Parameter Pointer. RAM Address for run-time parameters; shares the same memory-space as VP.	Consult the “UnicView AD Development Guide” document for details.
Library	Font or Icon Library.	
Font Width	Font Width	
Font Color	Color of the font.	
Bytes To Show	Indicates how many bytes to show, starting from the high byte in VP.	
Separator Format	Sequence of characters (ASCII) representing the separators for this Hex Display. The current value (contained on VP) will be shown in hexadecimal, and after each byte, a separator character is inserted. Special characters: 0x00 (blank), 0x0D (new line). Example: Current value in the 2 addresses starting at VP: 0x1234AABB. Separator Format: 0x00200D (blank, space, new line). What is shown on the LCM: 1234 AA BB	

4.3.12 RTC Display

Used to display current date and/or time, in digital format. Uses the internal RTC.

PP Length: 6 to 22 VPs.

Property	Description	Notes
Type	Type of the object.	
Name	Friendly name for this display variable.	
Area	Area of the display variable.	
Description	Descriptive text about this display variable.	
PP	Parameter Pointer. RAM Address for run-time parameters; shares the same memory-space as VP.	Consult the “UnicView AD Development Guide” document for details.
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
Library	Font or Icon Library.	
Font Width	Font Width	
Font Color	Color of the font.	
Date Format	Display format string. Use ASCII characters and the Field Codes on the following table. E.g.: Current time = 2012-05-02 12:00:00 Wednesday, Y-M-D H: Q: S 0x00, will display “2012-05-02 12:00:00”. M-D W H: Q 0x00, will display “05-02 WED 12:00”.	
Value Type	Value Type	

RTC Field Codes:

Description	Field Code	Range/Format
Year	Y	2000-2099
Month	M	01-12
Day	D	01-31
Hour	H	00-23
Minute	Q	00-59
Second	S	00-59
Date	W	SUN MON TUE WED THU FRI SAT
Coding end	0x00	

4.3.13 Analog Clock

Used to display current time, in analog format (a radial clock). Uses the internal RTC.

PP Length: 13 VPs.

Property	Description	Notes
Type	Type of the object.	
Name	Friendly name for this display variable.	
Area	Area of the display variable.	
Description	Descriptive text about this display variable.	
PP	Parameter Pointer. RAM Address for run-time parameters; shares the same memory-space as VP.	Consult the “UnicView AD Development Guide” document for details.
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
Library	Font or Icon Library.	
Hour Hand Center	Center of rotation for the hour hand.	
Minute Hand Center	Center of rotation for the minute hand.	
Second Hand Center	Center of rotation for the second hand.	
Hour Hand Icon	Icon for the hour hand.	
Minute Hand Icon	Icon for the minute hand.	
Seconds Hand Icon	Icon for the second hand.	

4.3.14 Table Display

Used to display tabular text.

PP Length: 13 VPs.

Property	Description	Notes
Type	Type of the object.	
Name	Friendly name for this display variable.	
Area	Area of the display variable.	
Description	Descriptive text about this display variable.	
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
PP	Parameter Pointer. RAM Address for run-time parameters; shares the same memory-space as VP.	Consult the “UnicView AD Development Guide” document for details.
Library	Font or Icon Library.	
Font Width	Font Width	
Font Height	Font Height	
Font Encoding	Font Encoding	
Fixed Spacing	Indicates whether characters have fixed width or automatically remove excess space.	
Font Color	Color of the font.	
Border Color	Border Color.	
Column Count	Number of columns.	
Row Count	Number of rows.	
First Column	Column index (0 based) of the first column (left-most) to be shown.	
First Row	Row index (0 based) of the first row(top-most) to be shown.	
Show Column Header	Indicates whether the column headers are to be displayed.	
Show Row Header	Indicates whether the row headers are to be displayed.	
Cell Length	Cell Length, in words (number of characters = 2*[Cell Length]).	
Content Type	Content Type	
Show Border	Indicates whether the table border is to be displayed.	
Value Type	Value Type	
Initial Value	Initial Value for this display variable.	

The following table summarizes the content type custom formats. The indicated values must be written into the first 2 VPs of each cell data.

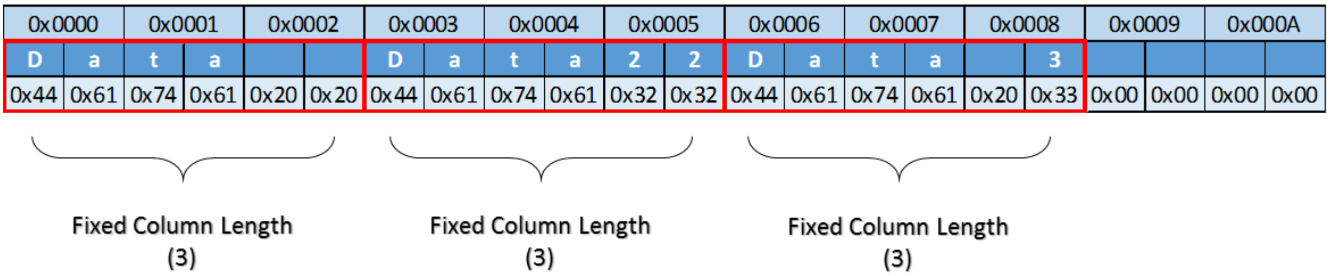
Address	Value	Format Description
First VP, High Byte	0x00	16-bit Integer.
	0x01	32-bit Integer.
	0x02	8-bit Unsigned Integer in High Byte of the VP.
	0x03	8-bit Unsigned Integer in Low Byte of the VP.
	0x04	64-bit Integer.
	0x05	16-bit Unsigned Integer.
	0x06	32-bit Unsigned Integer.
	0x10	Time Format, hh:mm:ss, BCD.
	0x11	Time Format, hh-mm-ss, BCD.
	0x12	Time Format, YYYY-MM-DD hh:mm:ss, BCD
	0xFF	Text
First VP, Low Byte	[0x00,0xFF]	In modes 0x00 - 0x06: High nibble is the number of integer digits. Low nibble is the number of decimal digits. In modes 0x10 - 0x11: BCD length, in bytes. Other modes: Not used.
Second VP	[0x0000,0xFFFF]	Text Color

Info You can use double terminator characters (**0xFFFF**) to indicate the end of text in a cell.

If "**Cell_Length**" is 0, the cell length is specified for each column. Starting from the High Byte of the VP, each byte indicates the Length of each column. In this case, the actual data starts from VP + ("Column Count"/2) (rounded up).

In the following examples, consider a table with VP = 0x0000, and 3 columns of data.

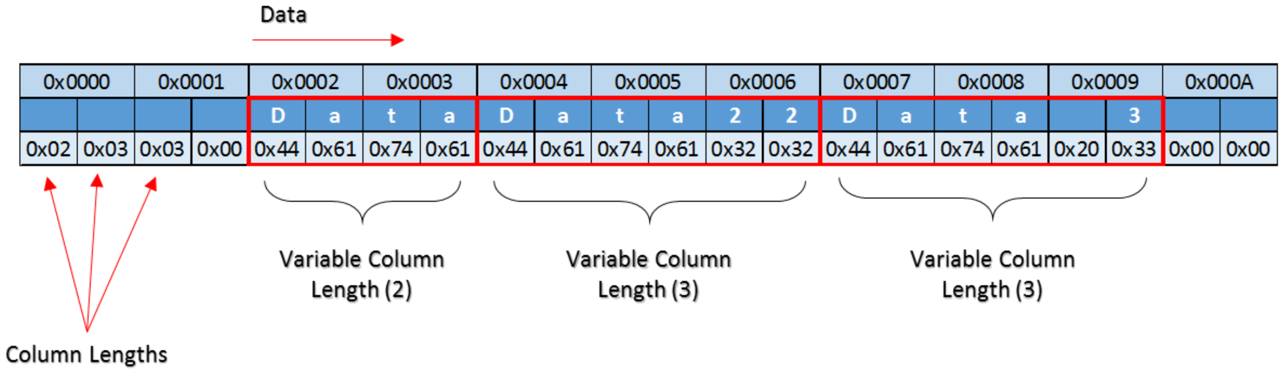
- **Fixed cell length:**



0x0000		0x0001		0x0002		0x0003		0x0004		0x0005		0x0006		0x0007		0x0008		0x0009		0x000A	
D	a	t	a			D	a	t	a	2	2	D	a	t	a		3				
0x44	0x61	0x74	0x61	0x20	0x20	0x44	0x61	0x74	0x61	0x32	0x32	0x44	0x61	0x74	0x61	0x20	0x33	0x00	0x00	0x00	0x00

Fixed Column Length
(3)
Fixed Column Length
(3)
Fixed Column Length
(3)

- **Variable cell length ("Cell Length" == 0x00):**

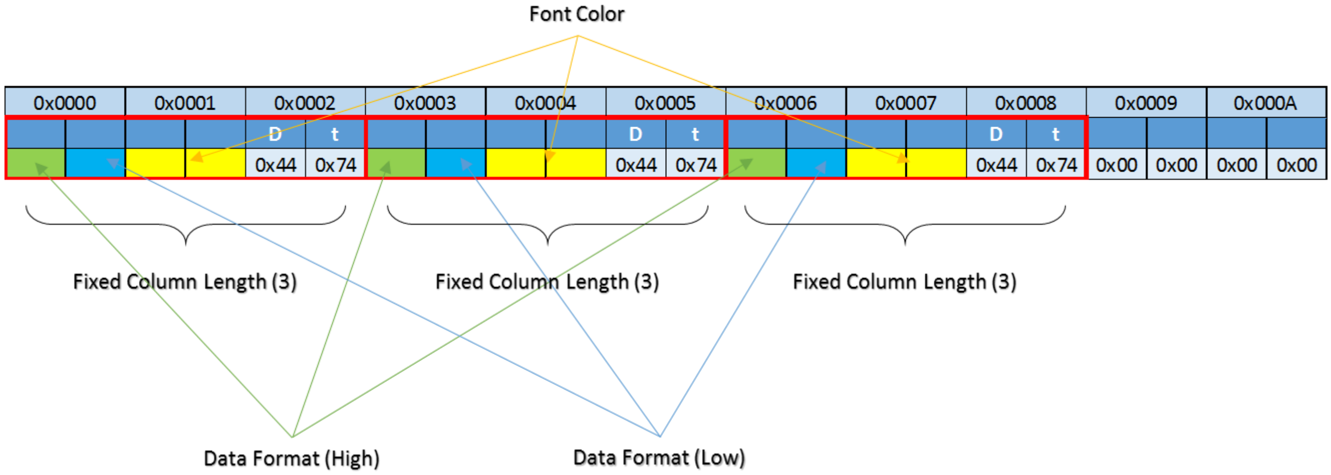


0x0000		0x0001		0x0002		0x0003		0x0004		0x0005		0x0006		0x0007		0x0008		0x0009		0x000A	
				D	a	t	a	D	a	t	a	2	2	D	a	t	a		3		
0x02	0x03	0x03	0x00	0x44	0x61	0x74	0x61	0x44	0x61	0x74	0x61	0x32	0x32	0x44	0x61	0x74	0x61	0x20	0x33	0x00	0x00

Data
Column Lengths
Variable Column Length (2)
Variable Column Length (3)
Variable Column Length (3)

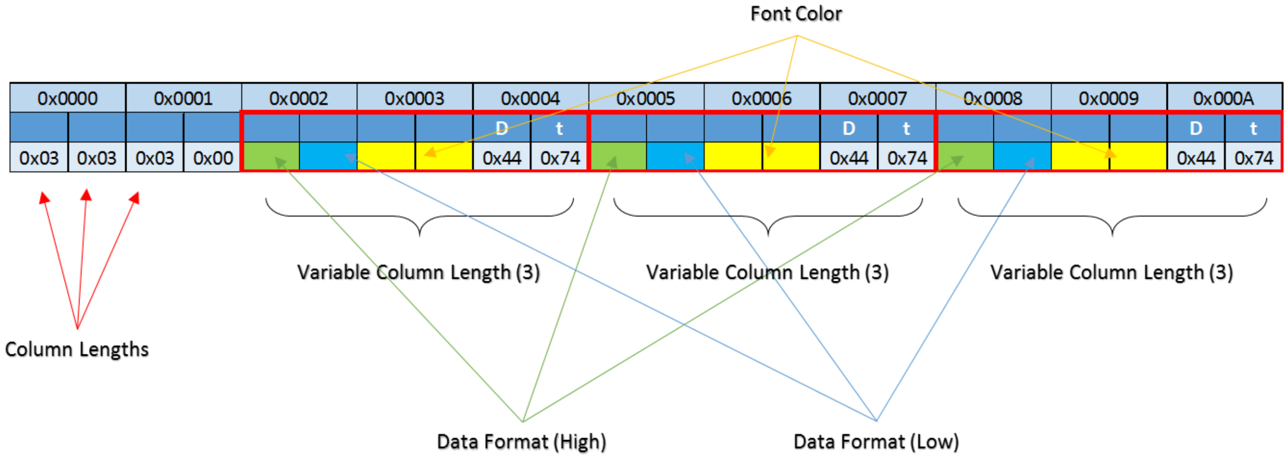
When using data type custom formats, you must reserve at least 3 VPs for each cell, because custom formats require the first 2 VPs of each cell.

- **Fixed cell length:**



0x0000		0x0001		0x0002		0x0003		0x0004		0x0005		0x0006		0x0007		0x0008		0x0009		0x000A	
				D	t					D	t					D	t				
🟩	🟦	🟨	🟨	0x44	0x74	🟩	🟦	🟨	🟨	0x44	0x74	🟩	🟦	🟨	🟨	0x44	0x74	0x00	0x00	0x00	0x00

- **Variable cell length ("Cell Length" == 0x00):**



0x0000		0x0001		0x0002		0x0003		0x0004		0x0005		0x0006		0x0007		0x0008		0x0009		0x000A	
								D	t					D	t					D	t
0x03	0x03	0x03	0x00	🟩	🟦	🟨	🟨	0x44	0x74	🟩	🟦	🟨	🟨	0x44	0x74	🟩	🟦	🟨	🟨	0x44	0x74

Font Color
Column Lengths
Variable Column Length (3)
Variable Column Length (3)
Variable Column Length (3)
Data Format (High)
Data Format (Low)

4.3.15 Trend Curve Display

Used to plot line graphs.

PP Length: 10 VPs.

Property	Description	Notes
Type	Type of the object.	
Name	Friendly name for this display variable.	
Area	Area of the display variable.	
Description	Descriptive text about this display variable.	
PP	Parameter Pointer. RAM Address for run-time parameters; shares the same memory-space as VP.	Consult the "UnicView AD Development Guide" document for details.
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
Curve Color	Color of the curve.	
Channel	Channel of the curve.	
Vertical Origin	Origin point of the vertical axis.	
Vertical Origin Value	Value associated to the vertical axis origin.	
Vertical Zoom	Vertical Zoom. Range: [0x0000,0x7FFF].	
Horizontal Increment	Horizontal Increment. Range: [0x01,0xFF].	

"**Vertical Zoom**" is calculated by this formula:

$$Vertical\_Zoom = \frac{(Y_f - Y_i) * 256}{(V_{max} - V_{min})}$$

Where:

- Y_f = Bottom Y coordinate of the "**Area**".
- Y_i = Top Y coordinate of the "**Area**".
- V_{max} = Maximum value on the plot.
- V_{min} = Minimum value on the plot.

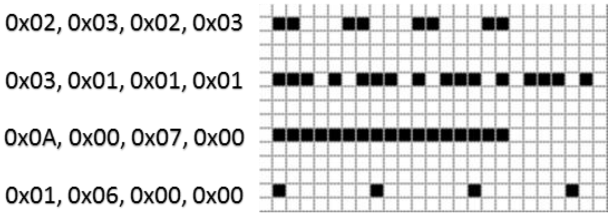
4.3.16 Graphic Primitives Display

Used to access many graphic manipulation functions, like copy/pasting and shape drawing.

PP Length: 6 VPs.

Property	Description	Notes
Type	Type of the object.	
Name	Friendly name for this display variable.	
Area	Area of the display variable.	
Description	Descriptive text about this display variable.	
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
PP	Parameter Pointer. RAM Address for run-time parameters; shares the same memory-space as VP.	Consult the "UnicView AD Development Guide" document for details.
Use Dashed Line	Indicates whether the border is to use a dash pattern.	
Dash Pattern	Dash pattern format. The pattern has 4 segments (bytes): Byte 1: Length of the first opaque segment. Byte 2: Length of the first transparent segment. Byte 3: Length of the second opaque segment. Byte 4: Length of the second transparent segment.	

Examples of dash patterns:

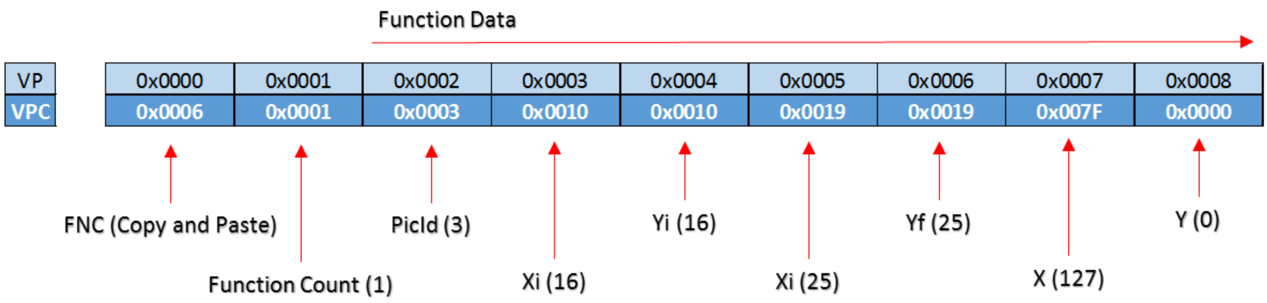


Info Write the dash pattern as a single number (0x02030203, for example).

The graphic primitives available (also called **Functions**) are used by writing their Function Code and Arguments in the VP of the Graphic Primitives Display. The first address (VP) holds the Function Code, which identifies the Function. The second address (VP + 1) indicates how many times this Function should be displayed. From the third address (VP + 2) and onwards, the information on how to draw these Functions is stored.

Address	Definition	Description
VP	Function Code (FNC)	Identifies the desired Function.
VP + 1	Function Count	Number of functions displayed. Only works with the same FNC. For FNC 0x0002, it's the number of vertices - 1.
VP + 2	Function Data	Function arguments, like coordinates, color, etc.

For example, to perform a **Copy and Paste** Function:



VP	0x0000	0x0001	0x0002	0x0003	0x0004	0x0005	0x0006	0x0007	0x0008
VPC	0x0006	0x0001	0x0003	0x0010	0x0010	0x0019	0x0019	0x007F	0x0000

The following table summarizes all available Functions.

FNC	Function	Data			
		Relative Address	Length (word)	Definition	Description
0x0001	Dot	0x00	2	(X,Y)	Dot coordinates.
		0x02	1	Color	Dot color.
0x0002	Line	0x00	1	Color	Line color.
		0x01	2	(X,Y)0	Coordinates of Vertex 0. High byte of X coordinate is the Control Byte.
		0x03	2	(X,Y)1	Coordinates of Vertex 1. High byte of X coordinate is the Control Byte.
		0x01+2*n	2	(X,Y)n	Coordinates of Vertex n. High byte of X coordinate is the Control Byte.
0x0003	Rectangle	0x00	2	(X,Y)i	Top-left coordinates. High byte of X coordinate is the Control Byte.
		0x02	2	(X,Y)f	Bottom-right coordinates.
		0x04	1	Color	Color of the rectangle.
0x0004	Rectangle Area	0x00	2	(X,Y)i	Top-left coordinates. High byte of X coordinate is the Control Byte.
		0x02	2	(X,Y)f	Bottom-right coordinates.
		0x04	1	Color	Fill color.
0x0005	Circle	0x00	2	(X,Y)	Coordinates of the center of the circle. High byte of X coordinate is the Control Byte.
		0x02	1	Radius	Radius of the circle.
		0x03	1	Color	Circle color.
0x0006	Copy and Paste	0x00	1	PicId	Index of the Screen to be copied. High byte of X coordinate is the Control Byte.
		0x01	2	(X,Y)i	Top-left coordinates of the copy area.
		0x03	2	(X,Y)f	Bottom-right coordinates of the copy area.
		0x05	2	(X,Y)	Paste position on current Screen, Top-left coordinates.
0x**07	Icon	0x00	2	(X,Y)	Top-left coordinates where the Icon will be displayed. High byte of X coordinate is the Control Byte.
		0x02	1	IconId	Index of the Icon in the Icon Library that will be displayed. High byte of FNC specifies the LibId. Transparent background.
0x0008	Color Fill	0x00	2	(X,Y)	First pixel to paint. All adjacent pixels with the same color will be also painted. High byte of X coordinate is the Control Byte.
		0x02	1	Color	Fill color.
0x0009	Spectrum Graph	0x00	1	Color	Connects (X, Yi), (X, Yf) with a line segment. High byte of X coordinate is the Control Byte.
		0x01	1	X	
		0x02	1	Yi	
		0x03	1	Yf	

0x000A	Segment	0x00	1	Color	Connects (Xi, Yi), (Xf, Yf) with line segment. High byte of Xi coordinate is the Control Byte.
		0x01	1	Xi	
		0x02	1	Yi	
		0x03	1	Xf	
		0x04	1	Yf	
0x000B	Arc	0x00	1	Color	Arc color.
		0x01	2	(X,Y)	Coordinates of the center of the arc. High byte of X coordinate is the Control Byte.
		0x03	1	Radius	Radius of the arc.
		0x04	1	Initial Angle	Initial angle of the arc. Range: [0,720], in 0.5° increments.
		0x05	1	Final Angle	Final angle of the arc. Range: [0,720], in 0.5° increments.
0x000C	Character	0x00	1	Color	Character color.
		0x01	2	(X,Y)0	Top-left coordinates of the character. High byte of X coordinate is the Control Byte.
		0x03.H	0.5	LibId	Index in the FLASH memory of the Font to use.
		0x03.L	0.5	Encoding	Font encoding: 0x00: 8 bit. 0x01: GB2312. 0x02: GBK. 0x03: BIG5 0x04: SJIS 0x05: UNICODE.
		0x04.H	0.5	Font Width	Font width, in pixels.
		0x04.L	0.5	Font Height	Font height, in pixels.
		0x05	1	Text0	Character code. Only valid on high byte of 8-bit encoding. For 0x01-0x04 encoding and ASCII data, DefaultFont will be used.
0x000D	Color Inverting Rectangle	0x00	2	(X,Y)i	Top-left coordinates. High byte of X coordinate is the Control Byte.
		0x02	2	(X,Y)f	Bottom-right coordinates.
		0x04	1	Inverting Bits	Indicates the bits which will inverse the color. 0x0000: No color inversion. 0xFFFF: all color bits are inversed.
0x000E	Binary Color Graph	0x00	2	(X,Y)i	Top-left coordinates of the graph. High byte of X coordinate is the Control Byte.
		0x02	1	Width	Width of the graph, in pixels.
		0x03	1	Height	Height of the graph, in pixels.
		0x04	1	Color1	Color associated to value 0b1.
		0x05	1	Color0	Color associated to value 0b0. If Color0 == Color1, then Color0 will be transparent.
		0x06	?	Binary Data	Binary data to display. Data is read MSB to LSB, drawing the pixels left to right. A set bit (0b1) is painted with Color1, and a clear bit (0b0) is painted with Color0.
0x000F	Bitmap	0x00	2	(X,Y)i	Top-left coordinates of the bitmap. High byte of X coordinate is the Control Byte.
		0x02	1	Width	Bitmap width, in pixels. Bitmap size must not exceed 28665 pixels.
		0x03	1	Height	Bitmap height in pixels. Bitmap size must not exceed 28665 pixels.
		0x04	?	Color Data	Color data to display. Each pixel occupies 1 word (MSB aligned, RGB565 format).
0x0010	Zoom and Paste	0x00	2	(X,Y)	Paste position on current Screen, Top-left coordinates. High byte of X coordinate is the Control Byte.
		0x02	2	(X,Y)i	Top-left coordinates of the zoom area.
		0x04	2	(X,Y)f	Bottom-right coordinates of the zoom area.

4.3.17 QR Code Display

Used to display QR Codes generated from the value of the VP.

PP Length: 6 VPs.

Property	Description	Notes
Type	Type of the object.	
Name	Friendly name for this display variable.	
Area	Area of the display variable.	
Description	Descriptive text about this display variable.	
VP	Value Pointer. RAM Address; each VP stores a 2-byte (word) value. The term VPC (Value Pointer Content) is used to refer to the actual value stored at the VP address, also denoted by "*VP".	
PP	Parameter Pointer. RAM Address for run-time parameters; shares the same memory-space as VP.	Consult the “UnicView AD Development Guide” document for details.
Pixel Size	Size of each pixel of the QR Code, in pixels.	
Initial Value	Initial Value for this display variable.	

The QR Code automatically reads and updates the data starting from its VP. To identify the end of the text data, append a double terminator (**0x0000** or **0xFFFF**). Data range: 0 to 458 bytes.

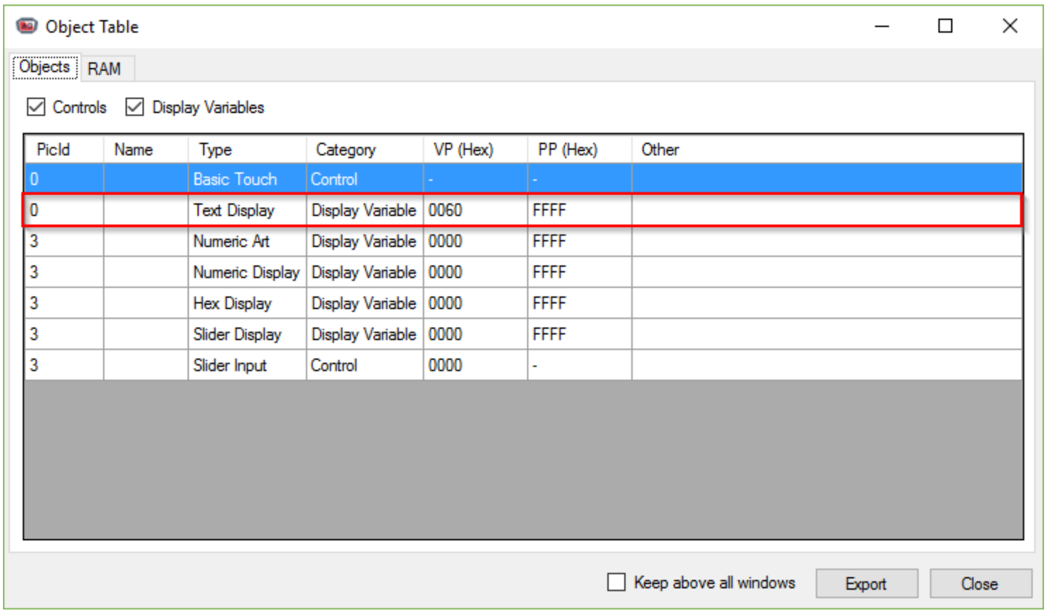
The size of the QR Code is set according to the data length:

- Up to 154 bytes: 45x45 QR pixels.
- Up to 458 bytes: 73x73 QR pixels.

Info QR Code needs the “12_QR_CODE.bin” file to work normally.

4.4 Object Table and RAM Map

When a project is compiled, all Interface Objects are allocated to their respective VPs and PPs. You can view the address mapping clicking the ☰ button, in the Project menu or Project bar, which opens the **Object Table**. The **Objects panel** shows all Controls and Display Variables, along with their PicIds, Names, Types, VPs and PPs.

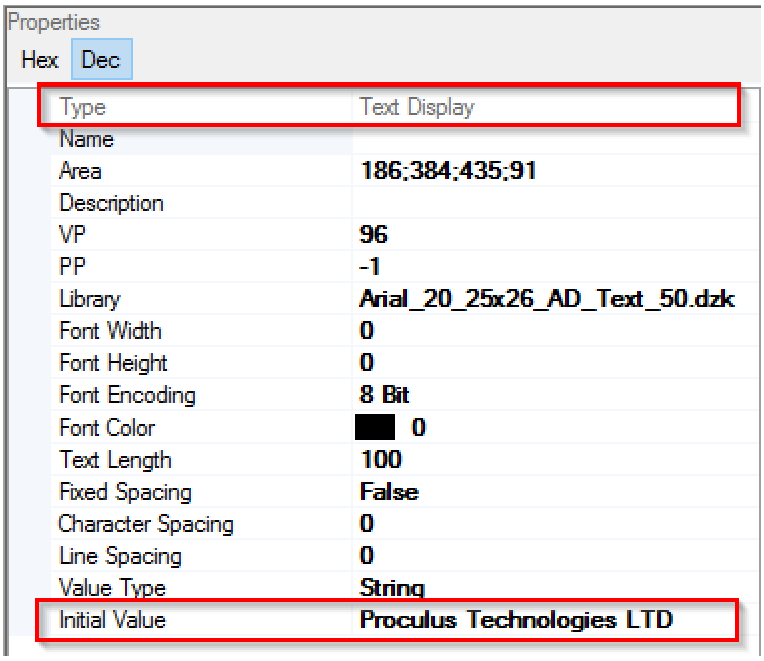


☑ Controls ☑ Display Variables

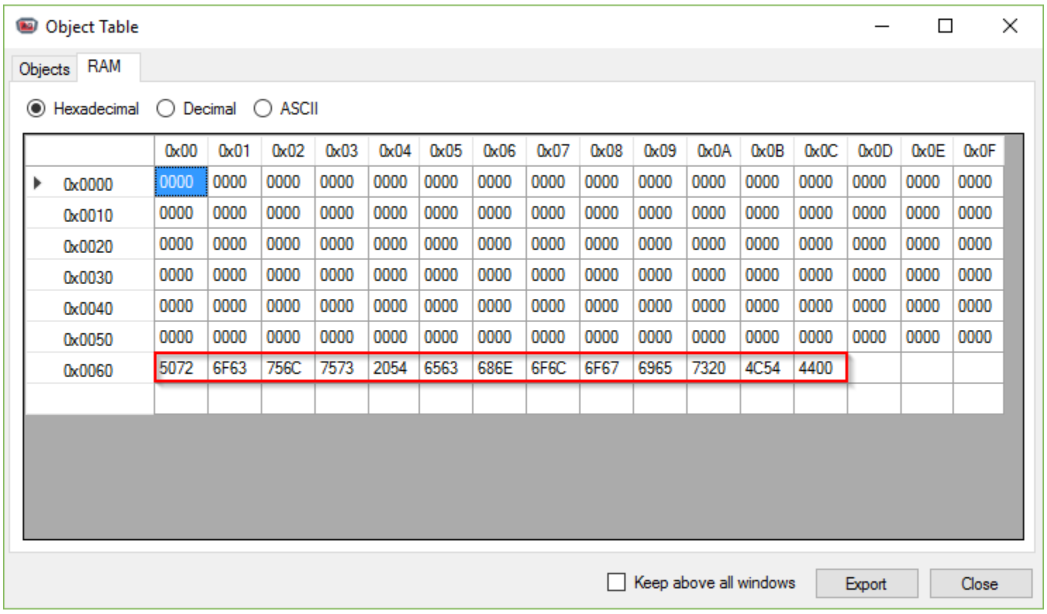
PicId	Name	Type	Category	VP (Hex)	PP (Hex)	Other
0		Basic Touch	Control	-	-	
0		Text Display	Display Variable	0060	FFFF	
3		Numeric Art	Display Variable	0000	FFFF	
3		Numeric Display	Display Variable	0000	FFFF	
3		Hex Display	Display Variable	0000	FFFF	
3		Slider Display	Display Variable	0000	FFFF	
3		Slider Input	Control	0000	-	

□ Keep above all windows

The **RAM panel** shows the complete RAM memory initialization map, in three view formats, to facilitate the visualization of different data types (hexadecimal, decimal and ASCII).

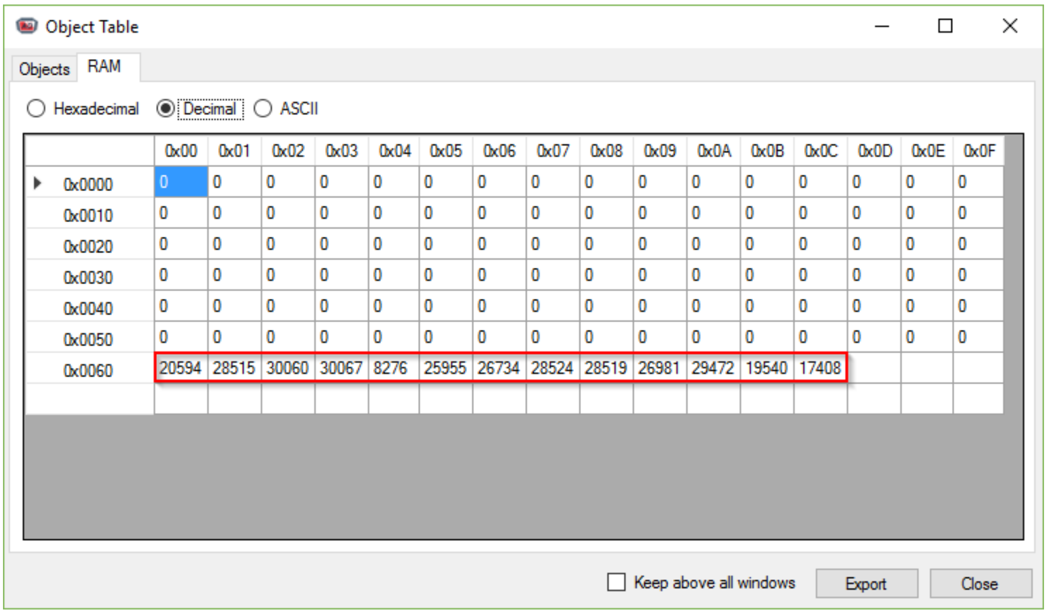


Type	Text Display
Name	
Area	186;384;435;91
Description	
VP	96
PP	-1
Library	Arial_20_25x26_AD_Text_50.dzk
Font Width	0
Font Height	0
Font Encoding	8 Bit
Font Color	■ 0
Text Length	100
Fixed Spacing	False
Character Spacing	0
Line Spacing	0
Value Type	String
Initial Value	Proculus Technologies LTD



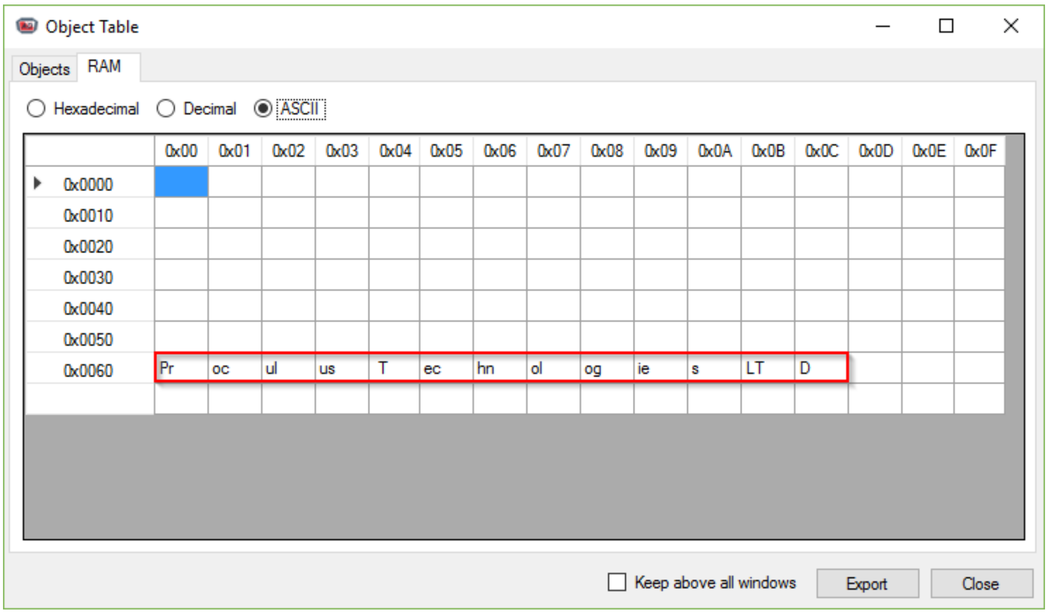
● Hexadecimal ○ Decimal ○ ASCII

	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
0x0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0x0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0x0020	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0x0030	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0x0040	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0x0050	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0x0060	5072	6F63	756C	7573	2054	6563	686E	6F6C	6F67	6965	7320	4C54	4400			



○ Hexadecimal ● Decimal ○ ASCII

	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
0x0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0010	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0020	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0030	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0040	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0050	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0060	20594	28515	30060	30067	8276	25955	26734	28524	28519	26981	29472	19540	17408			



○ Hexadecimal ○ Decimal ● ASCII

	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
0x0000																
0x0010																
0x0020																
0x0030																
0x0040																
0x0050																
0x0060	Pr	oc	ul	us	T	ec	hn	ol	og	ie	s	LT	D			

5 Screen Composition

Screen Composition is the creation of each Screen on the LCM. It’s the choice of background image, layout of buttons and displays, and all other aspects of a visual interface screen.

This section explains the basic aspects on how to compose a Screen.

5.1 Introduction

Here’s the definition of some terms related to Screen Composition:

Item	Description
Screen	The set of elements that are shown on the LCMs screen. A Screen is a “page” that contains a picture and Interface Objects. Each Screen is indexed to a unique identifier, called **PicId** (Picture ID).
LCM screen (screen)	The physical LCD screen on the LCM. It’s the electronic device where the user sees the Screens. It also has the Touch Panel, used to detect user input.
Touch Panel	The physical sensor device that detects when and where the user has touched the screen.
Background Image	Is the picture to be displayed at the background of each Screen.
Screen Orientation	Is the orientation of the elements on the Screen, relative to the LCM screen. The orientation is given in degrees, clockwise rotation.

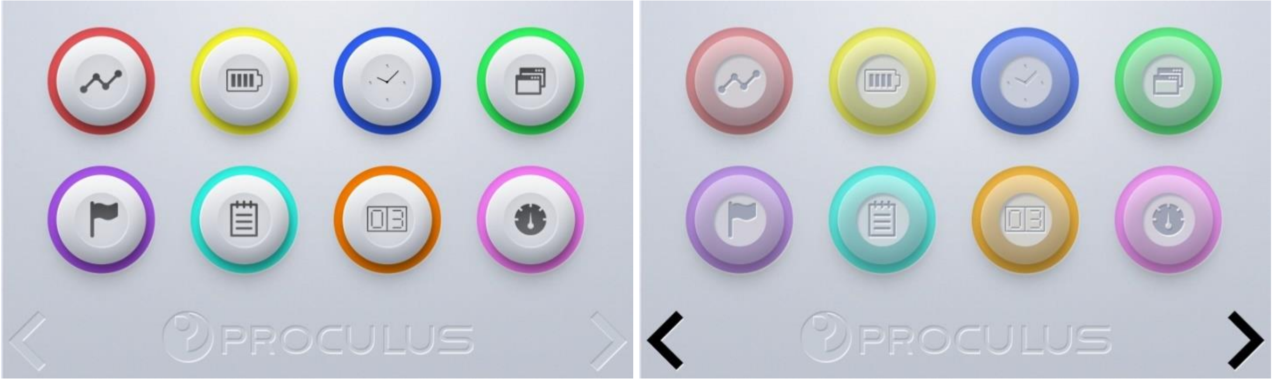
5.2 Screen Background Image

The background image is the picture that will be shown on the LCM screen. This image is automatically loaded when the current Screen (PicId) changes.

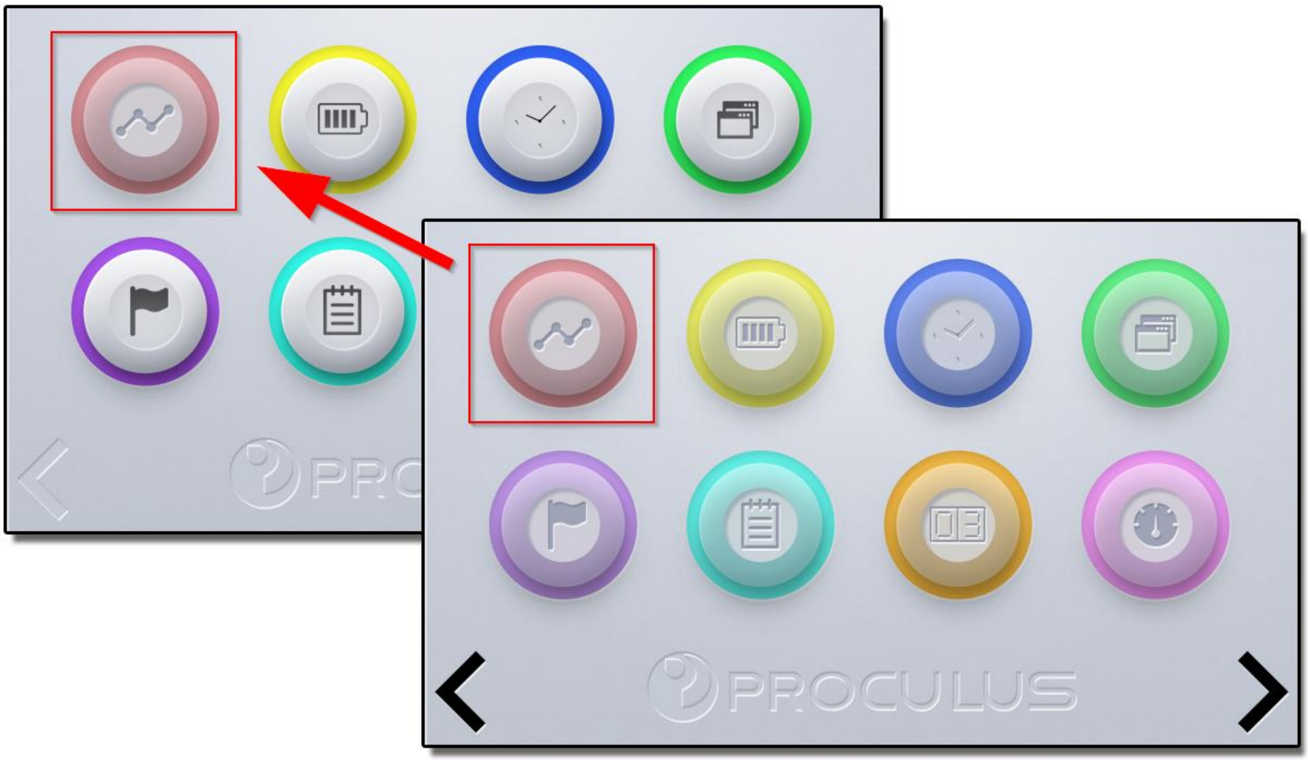
Info Please refer to section “3.1 - Image Resources” for details on how to manage the Image Resources.

5.2.1 “Pressed” Effect

Most Controls provide a visual feedback to indicate that they’re pressed down by the user. To achieve this effect, an additional Screen is used, whose image is designed with the “pressed” effect graphics, as seen on the following images. On the left, the **normal image**. On the right, the “pressed” **effect image**.



Each Control has a "FxId" property, that indicates in what PicId the effect is (set to -1 to indicate that no effect is desired). When the user touches the Control, the region of the Control (Area property) automatically overlays the respective region from the **effect image** on the current Screen.



The picture above represents a Control being pressed, and the respective region from the FxId picture being overlaid on the current Screen.

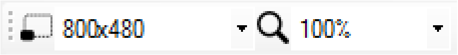
5.3 Screen Resolution, Orientation and Magnification

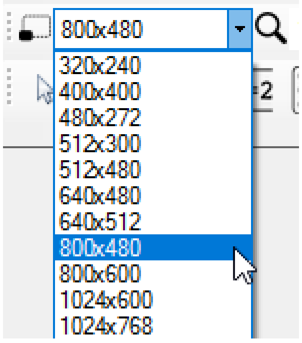
The resolution of the project must match that of the target LCM. If the resolution doesn't match, the background image will appear distorted.

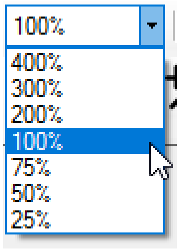
ℹ	Info	When the project resolution is changed, all Interface Objects will be proportionally re-dimensioned to fit the new size.

⚠	Caution	You can change the project resolution at any time, but we recommend **setting it only once, at the start of a new project**. Changing the resolution may dislocate the Interface Objects, due to rounding errors when converting from one resolution to another.

The resolution and magnification (zoom) can be selected on the Screen view bar:





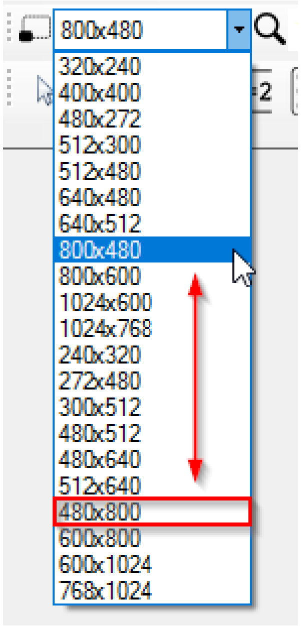


The magnification level is only intended to assist in Object layout, and doesn't interfere on any aspect of the final appearance of the Screen.

The resolution of the background image, on the other hand, may have any dimensions. The picture is automatically stretched to fit the current project resolution. This may cause unwanted blurring or distortion of the picture.

Info Use pictures of the same size as the project resolution to achieve the best results.

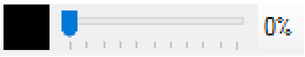
If your project is intended to be vertical (90° or 270° rotation), the resolution should be transposed. For example, the project for an 800x480 LCM, used in 90° or 270° orientation, should have the resolution 480x800 selected.



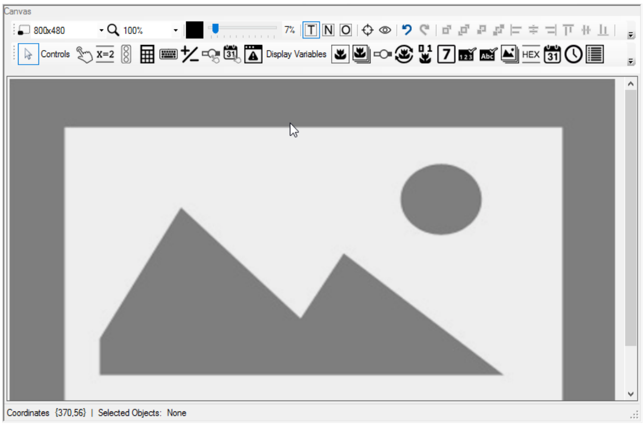
5.4 Screen Image Color Overlay

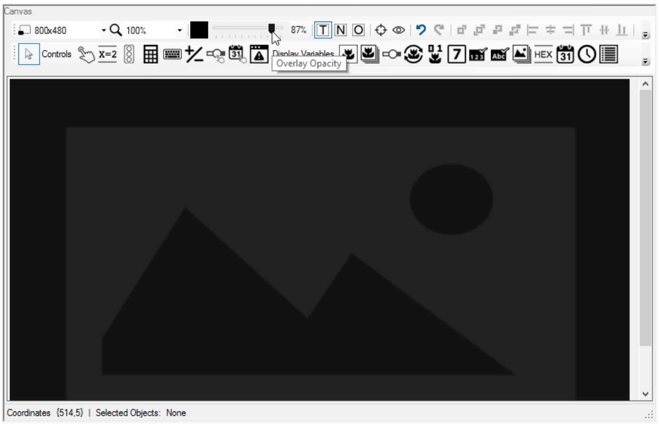
You may overlay a color to the background image, creating a “colorizing” effect on it. For instance, if you add a 2% white overlay, the background image you appear a little brighter. Adding a 5% black overlay, on the other hand, will make the image look darker, and can be used on sequential images, with progressively higher values, to create a “fade out” effect.

To choose the overlay color for the current Screen, click the color box on the Screen effects bar.



The opacity (intensity) of the overlay is a value from 0% to 100%, set by the slider near the color box. The following images exemplify the color overlay. On the left, a 7% black overlay has been applied. On the right, an 87% black overlay was used instead.





Info | The default color and opacity overlay options can be changed on the Colors window.

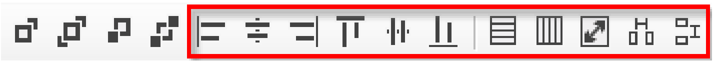
5.5 Object Layout

⚠	Caution	Controls mustn’t have overlapping areas.

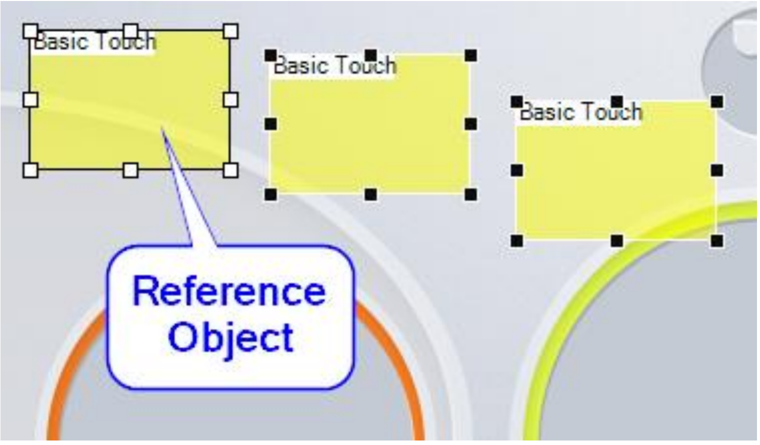
⚠	Caution	There is a maximum number of Display Variables per Screen. This number (64 or 128) is set on the System Configuration window.

5.5.1 Positioning and Sizing tools

The positioning and sizing tools are used to quickly layout the Interface Objects across the Screen.



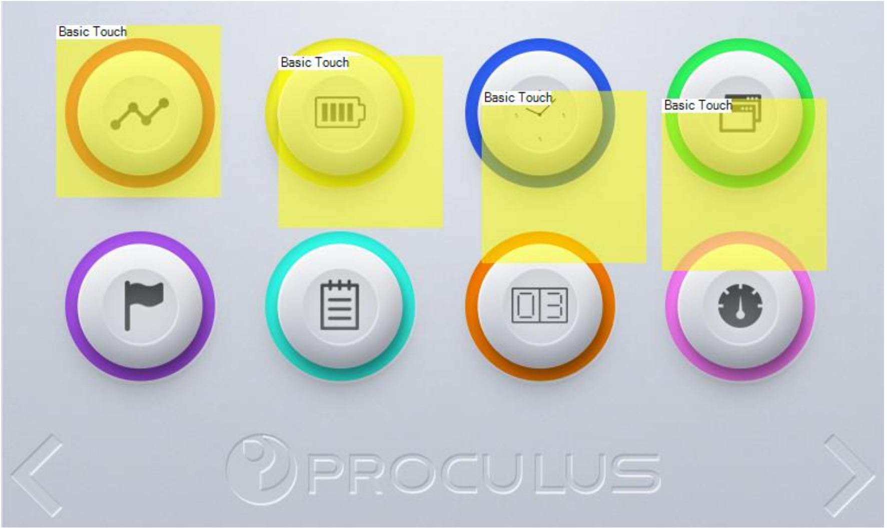
These tools usually are applied using a specific object as reference. This object is called the **Reference Object**. For example, when applying the “Align Top” tool, the Y coordinates of all selected Interface Objects will align to the Y coordinate of the Reference Object. The sizing handles, which indicate that an Object is selected, of the Reference Object are the same of a single selected Object. If multiple Objects are selected, those which are not the Reference Object have the sizing handles colors inverted:



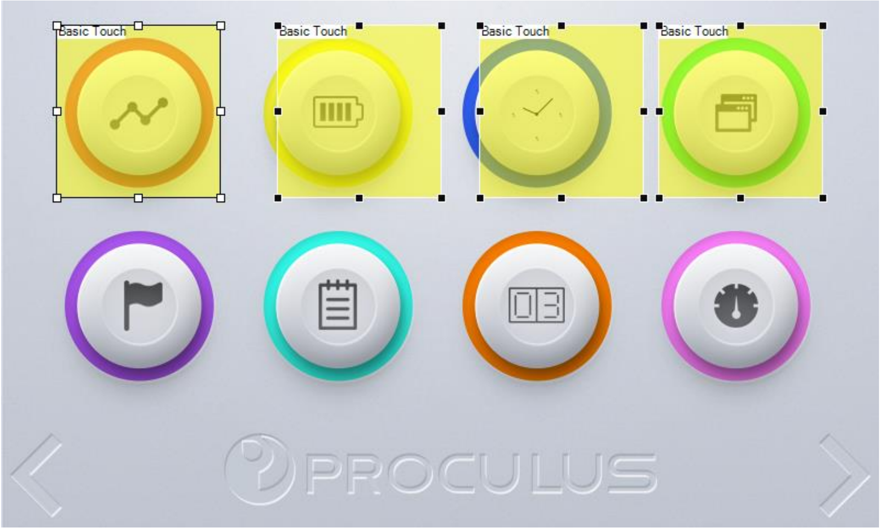
🛈	Info	The alignment and sizing tools are applied using the reference Object. You can select the reference Object using **(ALT+LMB)**.

The following procedure illustrates the usage of the positioning and sizing tools:

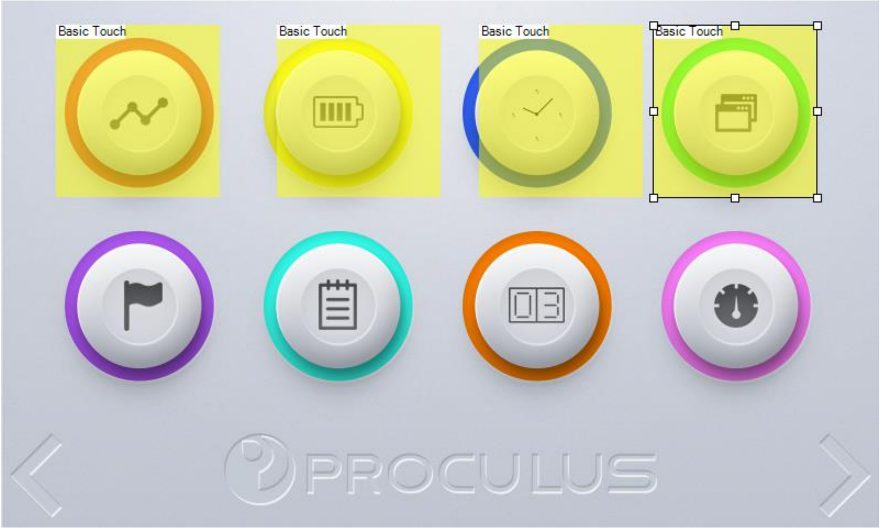
1. Create four Controls. You can either create one Control, then copy **(CTRL+C)** and paste **(CTRL+V)** it three times, or create four Controls and use the ⤢ tool to make them the same size (or ▤ then ▥).



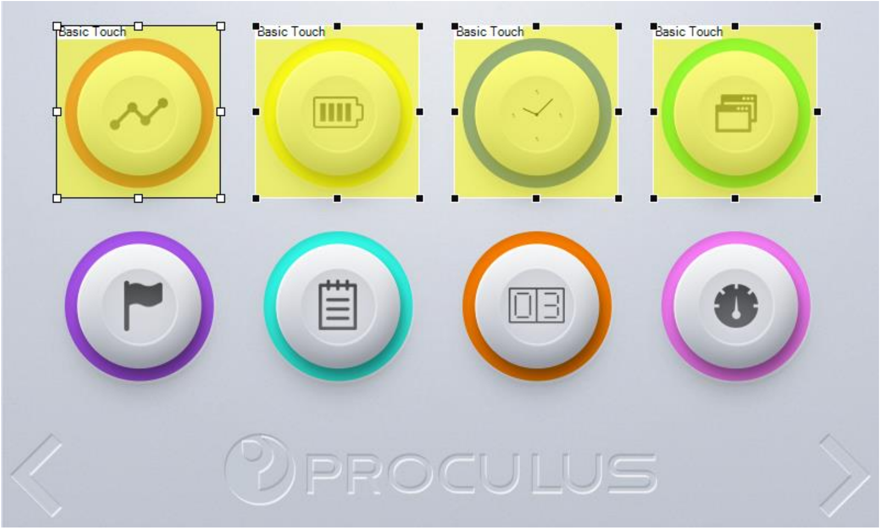
2. Select all controls by dragging a selection box around them or **(CTRL+A)**. Make sure the left-most Control is the reference object **(ALT+LMB)**. Click the ⊤ button to align their top coordinates.



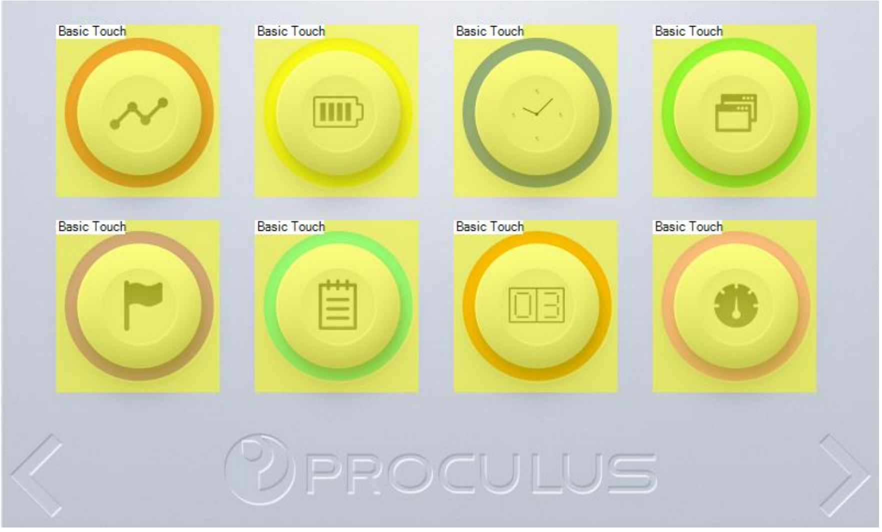
3. Select the right-most Control. Align it to the graphic button on the Screen either moving it with the mouse (LMB+DRAG), or pressing the **(LEFT)**, **(RIGHT)**, **(CTRL+LEFT)** and **(CTRL+RIGHT)** keys.



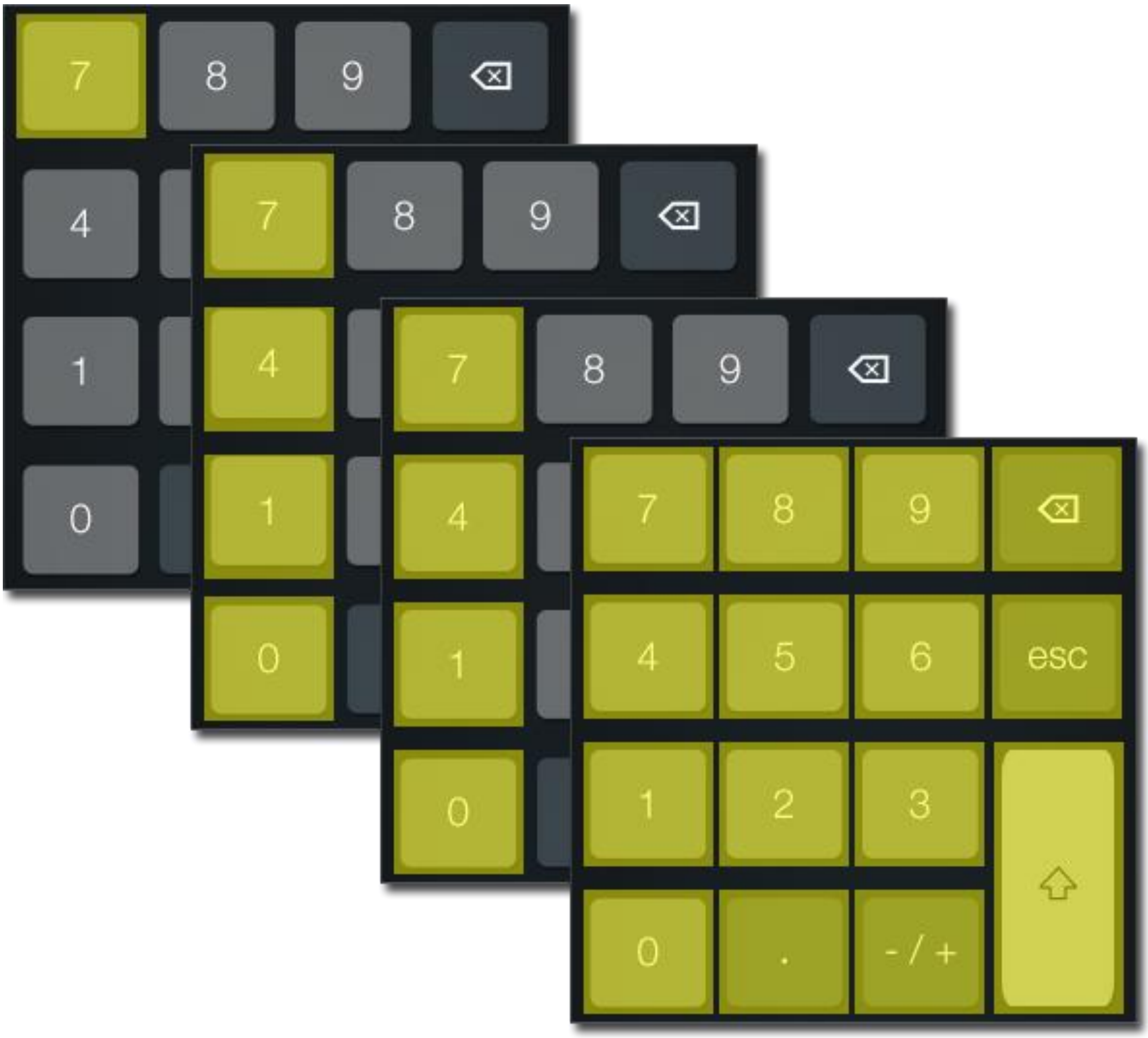
4. Select all the Controls. Click the button to uniformly distribute them.



5. Copy the Controls and move them down until they align to the bottom row of buttons.

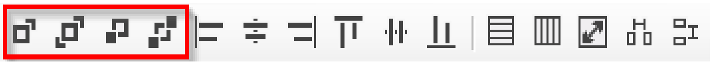


This procedure is particularly useful when designing keyboards:

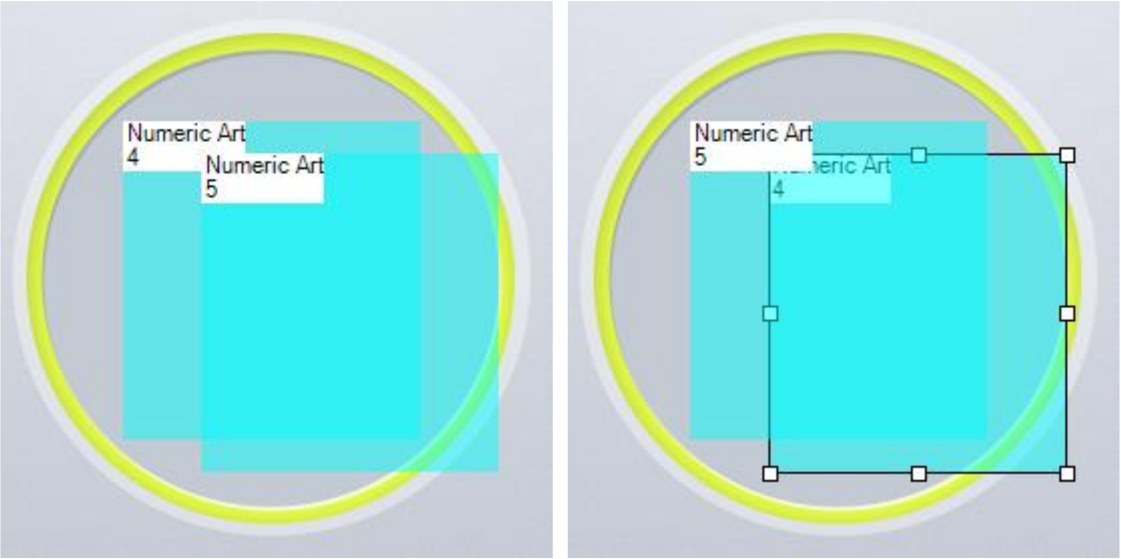


5.5.2 Ordering tools

It’s also possible to change the ordering of the Interface Objects. That’s only useful for Display Variables, since Controls can’t have their areas overlapping.



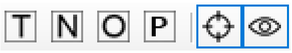
The higher the ordering index of a Display Variable, the “closer” it is to the user. In the example bellow, in the left picture, the Numeric Art with order index 5 is above the other one, and it will appear superimposed to the Numeric Art with order index 4.



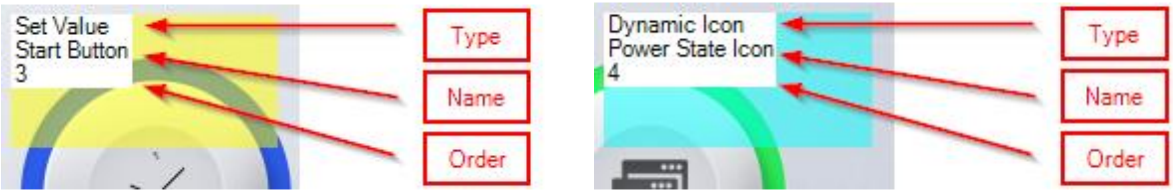
By selecting the first Numeric Art and clicking the “Send Backward” button, the order indexes will be swapped, and the first Numeric Art will now be underneath the other Numeric Art.

5.6 Object Visibility

Object visibility options are changed via the Object Visibility bar:



Interface Objects have captions to help identifying them. These captions can be individually enabled or disabled.



T Toggles the visibility of the “Type” caption.

N Toggles the visibility of the “Name” caption.

O Toggles the visibility of the “Order” caption.

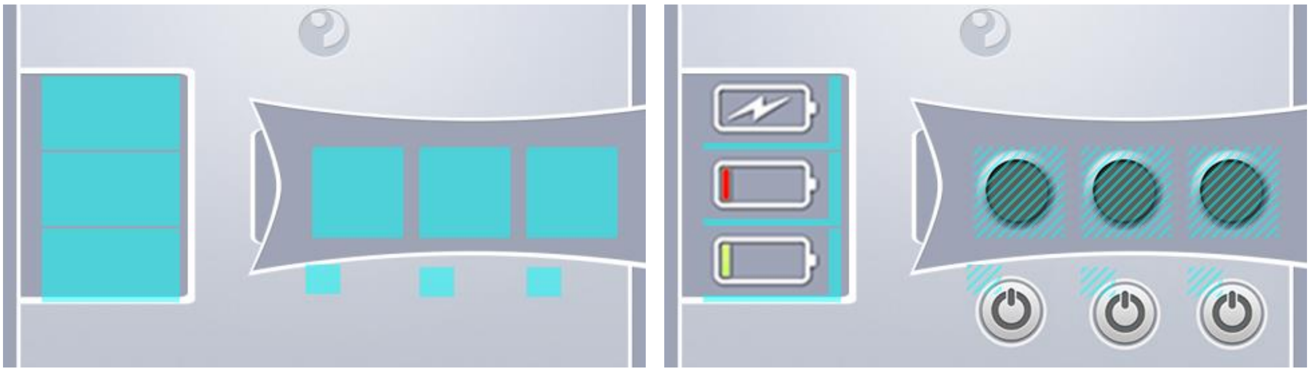
You can also toggle on and off the visibility of Object previews. Currently, the Interface Objects that support Icon previews are:

- Dynamic Icon
- Animated Icon
- Slider Display
- Bitwise Icon
- Numeric Art

And the Interface Objects that support text preview (for custom Fonts only) is:

- Text Display

The following pictures show a Screen with Object Preview off (left) and on (right). When Object Preview is enabled, if the preview of an Icon is larger than the Object’s area, the Icon is shown over the Object. If the Icon is smaller than the Object’s area, the Object is shown over the Icon, in a hatched pattern (usually a Display Variable’s area doesn’t affect the display of the Icon on the actual LCM screen).



P Toggles the visibility of Object previews.

Object visibility can also be toggled, based on their kind:

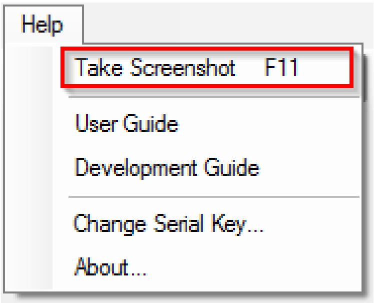
⌖ Toggles the visibility of Control Objects.

👁 Toggles the visibility of Display Variable Objects.

6 Additional Features

6.1 Screenshots

You can easily take screenshots of your project by clicking the “Take Screenshot” button in the Help menu, or pressing the **(F11)** key.



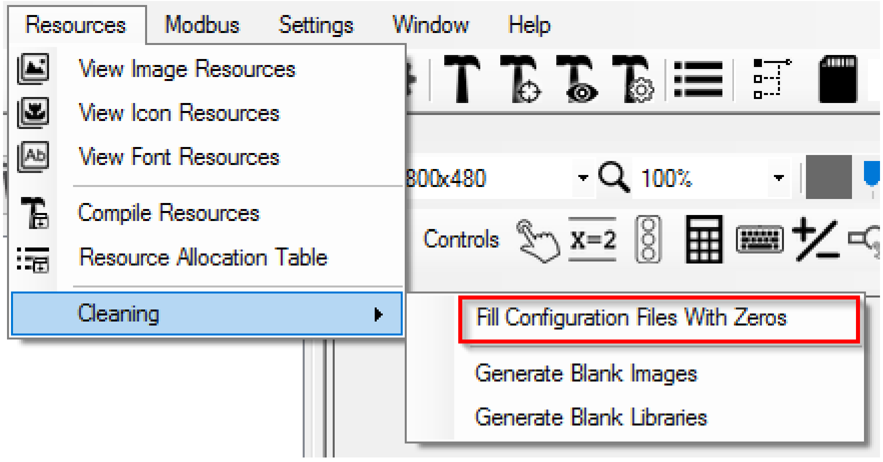
If there’s open project, the screenshot will be saved on the “Screenshots” folder of this project. Otherwise, the screenshot will be saved to UnicView AD’s user application folder.

6.2 LCM Memory Cleaning

You can choose how the compiler handles unused addresses. By default, it generates only the needed data on the Configuration Files. However, it may be necessary to fill the unused addresses with zeros. For example, if, for security reasons, you need to clean previous values on the LCM memory, or if you observe strange behavior, like Display Variables showing where they’re not supposed to.

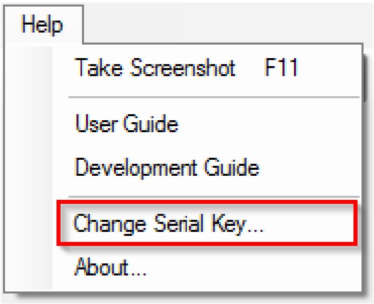
To enable the cleaning (filling with zeros) of unused addresses, enable the “Fill Configuration Files With Zeros” setting, in the Resources menu.

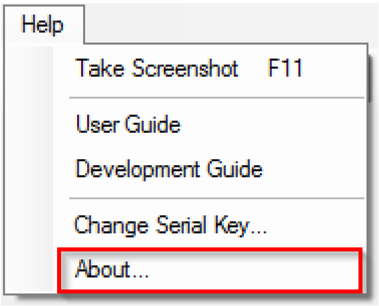
ℹ	Info	Configuration files that clear unused addresses are far larger than normal ones. We recommend transferring these files via SD Card.



6.3 License Management

You can change the registered Serial Key to modify your current License, by clicking the “Change Serial Key” button in the Help menu. To verify your current License, click the “About” button.





7 Troubleshooting

7.1 Compilation Results

7.1.1 Warnings

Controls

Code	Warning	Description	Solution
13001	Incompatible Firmware	A Control is not supported by the current firmware.	Remove the Control, or select a compatible firmware.
13002	Maximum VP Exceeded	The VP value of a Control is greater than the maximum allowed.	Use a lower VP.

Display variables

Code	Warning	Description	Solution
14001	Incompatible Firmware	A Control is not supported by the current firmware.	Remove the Control, or select a compatible firmware.
14002	Maximum VP Exceeded	The VP value of a Display Variable is greater than the maximum allowed.	Use a lower VP.
14003	Maximum PP Exceeded	The VP value of a Display Variable is greater than the maximum allowed.	Use a lower PP.
14004	Maximum Number of Display Variables per Page Exceeded	There are more Display Variables in one or more Screens than the maximum allowed.	Remove Display Variables on the Screens that generated the Warning.
14005	Maximum PicId Exceeded	There are Display Variables in a PicId greater than the maximum allowed.	Remove all Display Variables on the Screens that generated the Warning.

System Configuration

Code	Warning	Description	Solution
10001	Baud rate Error Percentage Too High	The custom baud rate value can't be achieved within 2.0% frequency accuracy.	Use a different baud rate value.

7.1.2 Errors

Controls

Code	Warning	Description	Solution
13101	File Size Overflow	The Control Configuration file length is too large.	Remove some Controls.
13102	Control Area Intersection	There are two or more Controls with overlapping areas.	Remove the are overlaps.
13103	File Inaccessible	Error when writing the Control Configuration file.	Try again. If error persists, contact our Technical Support.

Display Variables

Code	Warning	Description	Solution
14101	File Size Overflow	The Display Variables Configuration file length is too large.	Remove some Display Variables.
14103	File Inaccessible	Error when writing the Display Variables Configuration file.	Try again. If error persists, contact our Technical Support.

System Configuration

Code	Warning	Description	Solution
10101	File Not Written	Error when writing the Configuration File.	Try again. If error persists, contact our Technical Support.

7.2 Error Report

In case of unexpected software failure, UnicView AD will try to log the crash information. This information can be accessed by opening the error report file, located at “**%appdata%/Proculus Technologies Limited/UnicView AD/Log/ExceptionLog.txt**”.

If the failure persists, you may send the error report file to Proculus Technologies Limited Customer Support for analysis.

For feature requests or suggestions, please contact Proculus Technologies Limited Customer Support.

Customer support e-mail address: support@proculustech.com.