



Studio 5000 View Designer Import/Export Reference Manual

Version 9.01

Rockwell Automation Publication 9324-RM001A-EN-P, June 2023



Important User Information

Read this document and the documents listed in the additional resources section about installation, configuration, and operation of this equipment before you install, configure, operate, or maintain this product. Users are required to familiarize themselves with installation and wiring instructions in addition to requirements of all applicable codes, laws, and standards.

Activities including installation, adjustments, putting into service, use, assembly, disassembly, and maintenance are required to be carried out by suitably trained personnel in accordance with applicable code of practice.

If this equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc., is prohibited.

Throughout this manual, when necessary, we use notes to make you aware of safety considerations.



WARNING: Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.



ATTENTION: Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence.

IMPORTANT Identifies information that is critical for successful application and understanding of the product.

Labels may also be on or inside the equipment to provide specific precautions.



SHOCK HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.



BURN HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.



ARC FLASH HAZARD: Labels may be on or inside the equipment, for example, a motor control center, to alert people to potential Arc Flash. Arc Flash will cause severe injury or death. Wear proper Personal Protective Equipment (PPE). Follow ALL Regulatory requirements for safe work practices and for Personal Protective Equipment (PPE).

Rockwell Automation recognizes that some of the terms that are currently used in our industry and in this publication are not in alignment with the movement toward inclusive language in technology. We are proactively collaborating with industry peers to find alternatives to such terms and making changes to our products and content. Please excuse the use of such terms in our content while we implement these changes.

Import and export View Designer project content	Chapter 1	
	Export a View Designer project.....	5
	Import content into a View Designer project	6
	Edit exported View Designer project content.....	6
	Best practices and limitations	7
Format of exported content	Chapter 2	
	Contents of an exported View Designer project.....	9
	Text .hmi file overview	9
	General syntax rules	10
	Format of ViewApplication.hmi file	13
	Screen .hmi syntax	14
	Screen .hmi file example overview	15
	Popup .hmi syntax.....	20
	Folder Properties .hmi syntax	21
	Add-on Graphics syntax.....	22
	Navigation Menu syntax.....	25
	The Banner syntax.....	26
	Data Log syntax	41
	Event .hmi conventions	41
Common enums	44	

Import and export View Designer project content

Import and export View Designer project content

Use the import and export features in Studio 5000 View Designer® to export View Designer content formatted as text. You can then import the text as content into other View Designer projects. It is also possible to automate project creation or editing (for example with a scripted programming language). A View Designer project exports content in the following formats:

- Screens. Export as .hmi files in a folder that has the same structure as in **Project Explorer**.
- Popups. Export as .hmi files in a folder that has the same structure as **Project Explorer**.
- Add-On Graphics. Export as .hmi files. These files appear in the **Assets > Add-On Graphics** folder.
- Documents. Export as .pdf files. These files appear in the **Assets > Documents** folder.
- Images. Export as .jpg, .png, .bmp, or .svg. These files appear in the **Assets > Images** folder.
- Data Logs. Export as .hmi file. These files appear in the **Data Logs** folder.
- Navigation Menu and Shortcuts. Export as an .hmi file. These files appear in the **Navigation Menu** folder.
- Predefined Screens. The only pre-defined screen supported by import/export is the banner. Exports as an .hmi file. These files appear in the **Predefined Screens** folder.
- Controller References information. Export as .hmi files. These files appear in the **Devices** folder.
- Folder property information. Exports as .hmi files. These files appear inside folders in the **User-Defined Screens** folder.

Export a View Designer project

Export a View Designer project as a set of files. You can import the exported files to another View Designer project, and you can use any text editor to modify the project. Network paths and Sharepoint are not supported, but you can use drive mapping. For more information, see [Best practices and limitations](#) on [page 7](#).

1. With the project open in Studio 5000 View Designer, from the menu bar, select **File > Export Project**.
2. In the **Export project** window, enter or browse to the location where you want to export the project.



Tip: **Name** contains the name of the project as it appears in View Designer.

3. Select **Export**. When export finishes successfully, a prompt appears.
4. To open the location of the exported content, select the link to the file path. The location contains a folder with the same name as the exported View Designer project.
5. Use a text editor to open any exported file and view or edit the content.

Import content into a View Designer project

Import project content, such as screens, popups, and add-on graphics into another View Designer project. The imported elements overwrite the existing elements in the target project if they have the same name. If you must recover any content overwritten by the import, the original target project is saved as a backup file, stored in your project directory. You can import previously exported project content, or other content created in the supported formats.

If you import a screen with different dimensions than the target HMI device, View Designer will scale the screen contents to fit in the space available on the target HMI device screen. If the target HMI device screen is using the system banner, the imported screen contents will scale to fit the available space below the system banner if it is used on that screen. Note that the Arc graphic element is never scaled on import.



Tip: It is recommended to import a project on the same panel type it was created for. Import for different panel type may change the size, position and shapes of graphic elements.

1. Open the target View Designer project you want to import content to.
2. In the menu bar, select **File > Import Project**.
3. Navigate to the folder that contains the View Designer **ViewApplication.hmi** file of the project that you want to import.
4. Select the **ViewApplication.hmi** file and then select **Open**. **Import project** displays a progress bar.

Before the import is complete, you can cancel it by selecting **Cancel Import**.

When the import is complete, the content appears in the target View Designer project.

Edit exported View Designer project content

With a text editor, modify the property values of exported Screens, Popups and Add-On Graphics in .hmi format. You can then import the modified content to View Designer.

1. Go to the exported project directory.
2. Right-click the .hmi file you want to edit.
3. From the context menu, select **Open with...**
4. From the list, select the text editor you want to edit the file with.
5. Select **OK**.
6. In the text editor, find and edit the property values you want to modify.



Tip: You must follow the .hmi file syntax rules, and use valid parameter names and values to edit the file correctly. Wrongly formatted .hmi files cannot be successfully imported to View Designer. See [Format of exported content](#) on [page 9](#).

Specific property values must be entered in appropriate formats, for example enums or hex color codes.

7. Save your changes.

Best practices and limitations

Good practices:

- Do not use folder or file names ending with dot, spaces, or consisting of spaces only.
- If you insert multiple definitions of the same property for one object, the last is taken in account.
- If you want to import a project that uses a different language, you must import the language file after you import the project. For more information, refer to "Switch languages on the HMI device" in the Studio 5000™ View Designer Help.

Limitations:

- You can store and open projects in network locations, but it is not possible to perform export or import to and from network locations. If you require such functionality, you can use the Windows drive-mapping feature.
- The order of screens and shortcuts after an import may differ from the original order in the exported project.
- Project elements that use reserved operating system names (CON, PRN, AUX, NUL, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9) are exported with an underscore (_) added in front of the file name. For example: `_CON.hmi`
- Aliases to enum elements export with integer value instead of enumerated value.

For example: `Access := "1"`

This limitation also applies to enum values of borders that have an alias created.

For example:

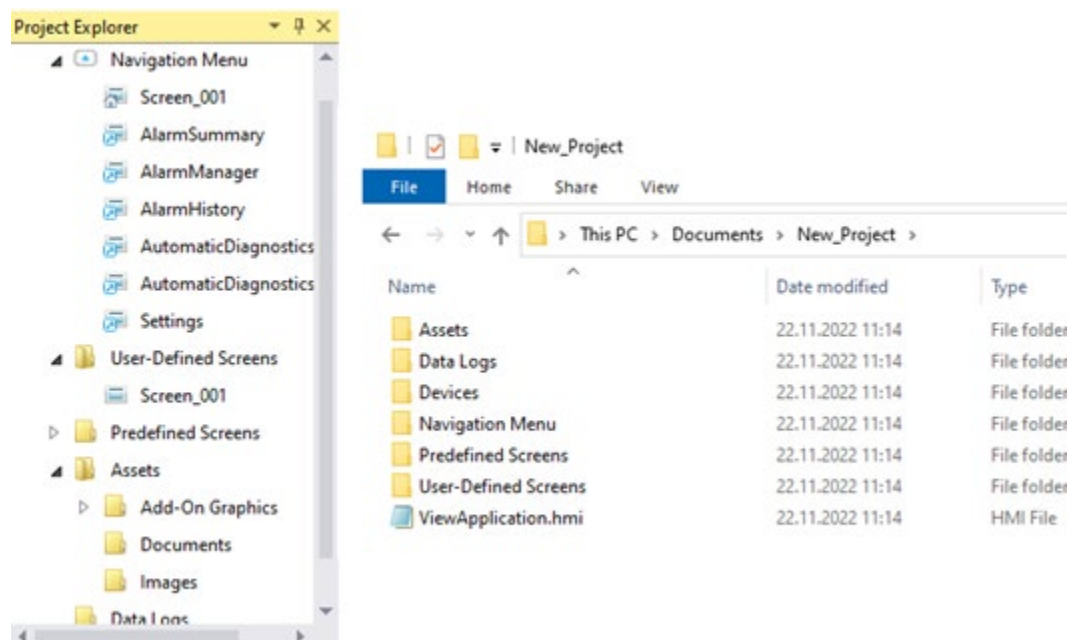
```
Border_001
:=
"<Pen><color>#000000</color><style>4</style><capstyle>0</capstyle><joinstyle>128</joinstyle><cosmetic>false</cosmetic><miterlimit>2</miterlimit><width>8</width><name><name></Pen>",
```


Format of exported content

Contents of an exported View Designer project

An exported project is saved in a new folder in a specified directory.

A text file named **ViewApplication.hmi** contains project settings and is the starting point when performing import. View Designer exports content for screens, popups, and the graphic elements on a screen or popup into specific text .hmi files, which model aspects of visualization in the View Designer. The .hmi files are stored in separate folders, and the exported project folder structure corresponds to the folder structure in the **Project Explorer** in View Designer.



For detailed information about the View Designer object properties, refer to the Studio 5000™ View Designer Help.

Text .hmi file overview

An .hmi file is a text file that contains configured property values of a project element, such a screen. The file follows specific syntax rules, and it can be created or modified with any text editor, but the syntax must be followed.

Correctly formatted .hmi files can be imported to existing projects in View Designer.

Object properties are described consecutively, divided in the following groups (if applicable):

- screen/popup/banner properties - general properties of the object

- state/color tables definition
- events configuration
- graphics elements/Add-On Graphics instances
- definition of user properties

See the general syntax of an .hmi file:

```
Screen | Popup | Banner [Name (only for screen and popup)]
{
    <screen/popup/banner properties>
    <state/color tables definition>
    <events configuration>
    <graphics elements/Add-On Graphics instances>
    <definition of user properties (only for screen or
popup)>
}
```

General syntax rules

Syntax rules for the View Designer .hmi files align with the IEC 61131-3 standard.

Follow these basic syntax rules:

- Syntax is case insensitive. For example, the identifiers abcd, ABCD, and aBCd are treated as identical.
- Element names, or identifiers, can contain up to 40 characters.
- All elements appear in the same general format:

```
[keyword identifying element] <modifiers for
element> [element name];
```

When an element contains content or sets properties, the code format is extended:

```
[keyword identifying element] <modifiers for
element> [element name] {
    <element content>
    < properties := property value >
}
```

- Every statement ends in a semicolon. For example:
- ```
Opacity := 90;
```
- Not every property, setting, or value must be specified. If a setting is not specified, then its default value is used.
  - Multiple statements can exist on the same line when they are separated by a semicolon. For example:

```
Opacity := 90; ^Visible := true;
```

- Bound properties appear in the general format:

```
property := "<Initial_Value>" -> "<Bound_Value>"
```

For example:

```
Text := "Text Display" ->
"::Local:HMIDevice.AutoLogoff.RemainingTime";
```

## Keywords

Keywords are specific identifiers used as individual syntactic elements. Keep these guidelines in mind when using keywords:

- Keywords cannot contain embedded spaces.
- The case of characters is not significant in keywords. For example, the keywords *FOR* and *for* are syntactically the same.
- Keywords cannot be used for any other purpose; for example, as variable or element names.
- Certain keywords can be used as variable or element name only if preceded with a caret (^). For example, you could use the keyword `^Screen` as an element name.

The following keywords can be used by preceding the keyword with a caret:

- action
- addongraphic
- alarm
- and
- aoi
- artifact
- banner
- behavior
- commanddefinition
- const
- continuous
- controller
- converge
- datalog
- datalogentry
- dev
- directedlink
- diverge
- enum
- event
- extends
- extern
- false
- fbd
- function
- graphicselement
- gsv
- gsv\_only
- gsvssv\_able
- gsvssv\_only
- hi

- hidden
- hide
- hihi
- inout
- input
- instruction
- ld
- lo
- lolo
- mathfunction
- motion
- namespace
- nullable
- output
- parameterconnection
- periodic
- phase
- popup
- program
- property
- required
- rll
- roc\_neg
- roc\_pos
- routine
- safety
- screen
- selectbranch
- sfc
- shortcut
- show
- simulbranch
- ssv
- ssv\_only
- st
- statedefinition
- statetabledefinition
- step
- stopstep
- struct
- stx
- tag
- task
- textbox
- thiscontrollerapp
- thisprogram

- thisroutine
- thistask
- transition
- triggerbasedefinition
- trip
- true
- type
- union
- unsupported
- userproperties
- using
- view
- viewfolder
- viewproject
- visible
- wire

Other keywords cannot be used as a variable or element name even if preceded with a caret. These reserved keywords are:

- by
- case
- else
- elsif
- end\_case
- end\_for
- end\_if
- end\_repeat
- end\_while
- exit
- for
- if
- mod
- nod
- of
- or
- repeat
- then
- to
- until
- while
- xor

## Format of ViewApplication.hmi file

The ViewApplication.hmi is a text file that contains the settings of a specific exported View Designer project.

The .hmi file follows the structure:

```
namespace ViewDesigner;
using HMICatalog;

ViewProject [name] {
 <homescreen declaration>
}

```

For example:

```
namespace ViewDesigner;
using HMICatalog;

/* HMI Device Type: 15" PanelView 5510 2715P-T15CD */

ViewProject SampleProject{
 HomeScreen := "Screen_001";
}

```

## Screen .hmi syntax

Each Screen .hmi file contains the properties of an exported screen.

The file follows the structure:

```
namespace ViewDesigner;
using HMICatalog;

screen [name] {
 <bindable setting values>
 <class properties' values>
 <security roles>
 <graphic element and AOG instances>
 <user defined properties>
 <state and color tables>
 <triggers>
}

```

For example:

```
namespace ViewDesigner;
using HMICatalog;

Screen MyScreen {
 ShowDefaultBanner := true;
 FillColor := "#ffffff";
 UpdateRate := EnumUpdateRate._500_ms;
}

```

```

// AOG Instance
PDF_Viewer_Landscape PDF_Viewer_Landscape_001 (
 CurrentPage := "1",
 DocumentName := "",
 Zoom := "Fit width") {

 X := 0;
 Y := 0;
 Width := 800;
 Height := 553;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access := EnumElementAccessFamily.Inherit;
 ForceAnimations := false;
}

// Trigger
OnTouchRelease TouchRelease_001 {
 ReleaseOnOpenClose := false;
 DisplayPopupCommand DisplayPopupCommand_001
(
 Title := "Hello World",
 Width := 500,
 Height := 400,
 PopupName := "User-Defined
Screens\Popup_001"
)
}
}

```

## Screen .hmi file example overview

The following is an example of an .hmi file content of a user-defined screen.

```

namespace viewdesigner;

using HMICatalog;

Screen Sample_Screen {
 ShowDefaultBanner := true;
 FillColor := "#ffffff";
 UpdateRate := EnumUpdateRate._500_ms;
 Button Sample_Button {
 FillColor := "#263a4e";
 Border {

```

```

 Color := "#262324";
 BorderStyle {
 Line := PenStyle.SolidLine;
 Cap := CapStyle.SquareCap;
 Join := JoinStyle.MiterJoin;
 }
 Width := 2;
 }
 CornerRadius := 3;
 TextAlignment := EnumAlignment.HCENTER_VCENTER;
 FontName := "Arial Unicode MS";
 FontSize := 12;
 FontColor := "#ffffff";
 Bold := true;
 Text := "Sample Button";
 UsePredefinedDisabled := true;
 BehaviorNavigateToScreen
BehaviorNavigateToScreen_001 {
 screenName := "Navigation Menu\Settings";
 Key := EnumBezelKeys.VK_NONE;
 RequiresFocus := false;
 AlwaysTriggerReleaseEvent := false;
 }
 X := 15.14;
 Y := 265.26;
 Width := 123.92;
 Height := 42.6;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access := EnumElementAccessFamily.Inherit;
 ForceAnimations := false;
}
TextDisplay Remaining_Time {
 Text := "Text Display" ->
"::Local:HMIDevice.AutoLogoff.RemainingTime";
 FontName := "Arial Unicode MS";
 FontSize := 15;
 Bold := true;
 Underline := false;
 FontColor := "#000000";
 TextAlignment := EnumAlignment.HLEFT_VCENTER;
 Padding := 0;
 CornerRadius := 0;
 Border {

```



```

 Color := "#000000";
 BorderStyle {
 Line := PenStyle.NoPen;
 Cap := CapStyle.SquareCap;
 Join := JoinStyle.MiterJoin;
 }
 Width := 1;
 }
 FillColor := "#00000000";
 StateTable StateTable {
 Expression := "" ->
 "::Local:HMIDevice.AutoLogoff.Enable";
 State State0 (
 ^visible := "False")
 {
 value := "0";
 }
 State State1 (
 ^visible := "true")
 {
 value := "1";
 }
 State DefaultState (
 ^visible := "true")
 {
 }
 }
 X := 183.38;
 Y := 265.26;
 Width := 109;
 Height := 26;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access := EnumElementAccessFamily.Inherit;
 ForceAnimations := false;
}
BarGraph1 Sample_Bar_Graph {
 FillColor := "#4d4d4d";
 FillOpacity := 100;
 TickMarkOpacity := 100;
 TextOpacity := 100;
 LevelColor := "#29abe2";
 Value := 0 ->
 "::Local:HMIDevice.Display.BacklightIntensity";
}

```

```
 MinValue := 0;
 MaxValue := 100;
 HousingColor := "#333333";
 TickMarkColor := "#f2f2f2";
 FontName := "Arial Unicode MS";
 FontSize := 12;
 FontColor := "#ffffff";
 X := 310.79;
 Y := 265.26;
 Width := 108.74;
 Height := 201.56;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access := EnumElementAccessFamily.Inherit;
 ForceAnimations := false;
 }
 Width := 1280;
 Height := 981;
 ForceAnimations := false;
}
```

In this .hmi file, the following graphic elements are described:

- Button
- TextDisplay
- BarGraph1
- Border (as part of the button)

Each graphic element in the above example has properties configured and animations or events applied. All properties and their values and events, behaviors, and animations in the .hmi file match those displayed in the View Designer Properties tab. This .hmi file sample defined the following graphic element properties:

- General properties:
  - ShowDefaultBanner
  - UpdateRate
  - Value
  - MaxValue
  - MinValue
  - Text
- Appearance properties:
  - Opacity
  - FillOpacity
  - TickMarkOpacity
  - TextOpacity

- FillColor
- Color
- TextAlignment
- FontName
- FontSize
- FontColor
- Bold
- Underline
- UsePredefinedDisabled
- Border Style
- Line
- Cap
- Join
- Padding
- CornerRadius
- ^Visible
- Enabled



Tip: If you want you use W3C color keywords instead of hex values, you can configure them as bound properties. For example: `FillColor := "" ->`

`''lightblue''`

- Position and size properties:
  - X
  - Y
  - Width
  - Height
  - Angle
  - ForceAnimations
- Security Properties:
  - Access

For detailed information about the View Designer graphic element properties, refer to "Graphic elements library" in the Studio 5000™ View Designer Help.

The .hmi also contains the `StateTable` animation, and the `BehaviorNavigateToScreen` behavior.

## Bound properties

The following is an example of how bound properties are expressed in the .hmi:

```
Text := "Text Display" ->
"::Local:HMIDevice.AutoLogoff.RemainingTime";
Expression := "" -> "::Local:HMIDevice.AutoLogoff.Enable";
```

## Popup .hmi syntax

Each Popup .hmi file contains the properties of an exported popup.

The file follows the structure:

```

namespace ViewDesigner;
using HMICatalog;

popup [name] {
 <bindable setting values>
 <class properties' values>
 <security roles>
 <graphic element and AOG instances>
 <user defined properties>
 <state/color tables>
 <triggers>
}

```

For example:

```

namespace ViewDesigner;
using HMICatalog;

Popup Popup_001 {
 CaptionVisible := true;
 CloseButtonVisible := true;
 Caption := "Popup Caption";
 UpdateRate := EnumUpdateRate._500_ms;
 FillColor := "#263a4e";
 BackgroundOpacity := 100;
 Cacheable := false;

 Button Button_002 {
 FillColor := "#263a4e";
 Border {
 Color := "#262324";
 BorderStyle {
 Line := PenStyle.SolidLine;
 Cap := CapStyle.SquareCap;
 Join := JoinStyle.MiterJoin;
 }
 Width := 2;
 }
 CornerRadius := 3;
 TextAlignment := EnumAlignment.HCENTER_VCENTER;
 FontName := "Arial Unicode MS";
 FontSize := 12;
 }
}

```

```

 FontColor := "#ffffff";
 Bold := true;
 Text := "Go to screen";
 UsePredefinedDisabled := true;
 BehaviorNavigateToScreen
BehaviorNavigateToScreen_001 (
 Message := "Hello World")
 {
 screenName := "User-Defined
Screens\Screen_002";
 Key := EnumBezelKeys.VK_NONE;
 RequiresFocus := false;
 AlwaysTriggerReleaseEvent := false;
 }
 X := 294.82;
 Y := 166;
 Width := 100;
 Height := 30;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access := EnumElementAccessFamily.Inherit;
 ForceAnimations := false;
 }
}

```

## Folder Properties .hmi syntax

Each `__folder_properties.hmi` file contains the properties of an exported folder.

The file follows the structure:

```

namespace ViewDesigner;
using HMICatalog;

```

```

ViewFolder [name] {
 <SecurityRoles>
}

```

For example:

```

namespace ViewDesigner;
using HMICatalog;

```

```

/* HMI Device Type: 12.1" PanelView 5510 2715P-T12WD
*/ ViewFolder User_Defined_Screens {

```

```

SecurityRoles {
 Supervisor := RoleAccess.FullAccess;
 Restricted := RoleAccess.FullAccess;
 Operator := RoleAccess.FullAccess;
 Maintenance := RoleAccess.ReadOnly;
 Engineer := RoleAccess.NoAccess;
}
}

```

### Security Roles Behavior

- If a role has a value assigned, in the `SecurityRoles` object it displays as an enum, and you can overwrite it with another enum value.

For example: `Engineer := RoleAccess.NoAccess`

- If a role is not listed, it inherits the value the same way it happens in View Designer.
- If the `SecurityRoles` object does not exist:
  - in new folders, all security role values are inherited the same way it happens in View Designer
  - in an existing folder, all security role values are not changed.



Tip: Administrator privileges cannot be modified in View Designer and also cannot be edited by using the .hmi files.

## Add-on Graphics syntax

The Add-On Graphic .hmi file contains the configuration of an exported Add-On Graphic.

The file follows the structure:

```

namespace ViewDesigner::AOG;

using HMICatalog;

/* HMI Device Type: 9" PanelView 5510 2715P-T9WD */

AddOnGraphic [name] {
 <Add-On Graphic properties>
 <user defined properties>
 <graphic element instances>
}

```

For example:

```
namespace ViewDesigner::AOG;

using HMICatalog;

/* HMI Device Type: 9" PanelView 5510 2715P-T9WD */

AddOnGraphic MyAOG {
 Description := "";
 Vendor := "";
 Major := 1;
 Minor := 0;
 Extended := "";
 Note := "";
 Width := 75;
 Height := 78.5;
 TerminalWidth := 800;
 TerminalHeight := 480;
 UserProperties {
 TagInstance {
 DataType := "":LGX.REAL";
 Description := "";
 Category := "General";
 }
 }
}

Button Button_001 {
 FillColor := "#263a4e";
 Border {
 Color := "#262324";
 BorderStyle {
 Line := PenStyle.SolidLine;
 Cap := CapStyle.SquareCap;
 Join := JoinStyle.MiterJoin;
 }
 Width := 2;
 }
 CornerRadius := 3;
 TextAlignment := EnumAlignment.HCENTER_VCENTER;
 FontName := "Arial Unicode MS";
 FontSize := 12;
 FontColor := "#ffffff";
 Bold := true;
 Text := "Button";
}
```

```
UsePredefinedDisabled := true;
X := 0;
Y := 0;
Width := 75;
Height := 22.5;
Angle := 0;
Enabled := true;
^Visible := true;
Opacity := 100;
Access := EnumElementAccessFamily.Inherit;
}
Button Button_002 {
 FillColor := "#263a4e";
 Border {
 Color := "#262324";
 BorderStyle {
 Line := PenStyle.SolidLine;
 Cap := CapStyle.SquareCap;
 Join := JoinStyle.MiterJoin;
 }
 Width := 2;
 }
 CornerRadius := 3;
 TextAlignment := EnumAlignment.HCENTER_VCENTER;
 FontName := "Arial Unicode MS";
 FontSize := 12;
 FontColor := "#ffffff";
 Bold := true;
 Text := "Button";
 UsePredefinedDisabled := true;
 X := 0;
 Y := 31.25;
 Width := 75;
 Height := 22.5;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access := EnumElementAccessFamily.Inherit;
}
NumericDisplay NumericDisplay_001 {
 DigitsBeforeDecimal := 5;
 DigitsAfterDecimal := 0;
 LeadingZero := false;
 TrailingZeros := false;
```



```

Rounding := EnumNumericRounding.Nearest;
LeadingZerosFill := false;
Value := 0 -> "TagInstance";
FontName := "Arial Unicode MS";
FontSize := 15;
Bold := true;
FontColor := "#000000";
TextAlignment := EnumAlignment.HRIGHT_VCENTER;
Padding := 0;
CornerRadius := 0;
Border {
 Color := "#000000";
 BorderStyle {
 Line := PenStyle.NoPen;
 Cap := CapStyle.SquareCap;
 Join := JoinStyle.MiterJoin;
 }
 Width := 1;
}
FillColor := "#00000000";
X := 15.5;
Y := 56.5;
Width := 44;
Height := 22;
Angle := 0;
Enabled := true;
^Visible := true;
Opacity := 100;
Access := EnumElementAccessFamily.Inherit;
}
}

```

## Navigation Menu syntax

The navigation menu exports as a set of folders and .hmi files matching the structure of the navigation menu configured for your project. The .hmi files represent the shortcuts you configured in your navigation menu.

Shortcut .hmi files follow the structure:

```

namespace ViewDesigner;

using HMICatalog;

/* HMI Device Type: 9" PanelView 5510 2715P-T9WD */

Shortcut [name] (

```

```
 <screen property bindings>
 {
 <shortcut property definitions>
 }
```

For example:

```
namespace ViewDesigner;

using HMICatalog;

/* HMI Device Type: 9" PanelView 5510 2715P-T9WD */

Shortcut Mixer1Detail (
 MixerTag := "::LGX.Mixer1")
{
 TargetScreenName := "User-Defined
Screens\MixerDetail";
 UseScreenSecurity := true;
 Caption := "Mixer1
Detail";
}
```

## The Banner syntax

The banner is the only pre-defined content which exports. The export of the banner will include any of your changes. The banner.hmi file follows the structure:

```
namespace ViewDesigner;

using HMICatalog;

/* HMI Device Type: 9" PanelView 5510 2715P-T9WD */

Banner {
 <banner properties>
 <graphic element and AOG instances>
 <state and color tables>
 <triggers>
}
```

For example, the default banner exports as:

```
namespace ViewDesigner;
```

```
using HMICatalog;

/* HMI Device Type: 9" PanelView 5510 2715P-T9WD */

Banner {
 FillColor := "#738596";
 UpdateRate := EnumUpdateRate._500_ms;
 PanelDeviceGroup LogonGroup {
 Button BAN_LogonBtn {
 FillColor := "#263a4e";
 Border {
 Color := "#262324";
 BorderStyle {
 Line := PenStyle.SolidLine;
 Cap := CapStyle.SquareCap;
 Join := JoinStyle.MiterJoin;
 }
 Width := 1;
 }
 CornerRadius := 3;
 TextAlignment :=
EnumAlignment.HCENTER_VCENTER;
 FontName := "Arial Unicode MS";
 FontSize := 8.6;
 FontColor := "#ffffff";
 Bold := true;
 Text := "";
 UsePredefinedDisabled := true;
 OnTouchRelease released_001 {
 ReleaseOnOpenClose := false;
 DisplayLogOnPopupCommand
DisplayLogOnPopupCommand_001 {
 Width := 500;
 Height := 400;
 PopupName := "";
 }
 }
 X := 0;
 Y := 0;
 Width := 36;
 Height := 36;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 }
 }
}
```

```

 Access :=
EnumElementAccessFamily.Inherit;
 }
 LoggedIn LoggedIn {
 FillColor := "#e5e5e5";
 X := 8.78;
 Y := 6.84;
 Width := 18.86;
 Height := 24.25;
 Angle := 0;
 Enabled := true;
 ^Visible := true ->
"::Local:HMIDevice.Security.CurrentUserName=="Guest$"
? 0 : 1";
 Opacity := 0 ->
"::Local:HMIDevice.Security.CurrentUserName=="Guest$"
? 0 : 100";
 Access :=
EnumElementAccessFamily.Inherit;
 }
 LoggedOut LoggedOut {
 FillColor := "#e5e5e5";
 X := 8.17;
 Y := 6.84;
 Width := 18.74;
 Height := 24.09;
 Angle := 0;
 Enabled := true;
 ^Visible := true ->
"::Local:HMIDevice.Security.CurrentUserName=="Guest$"
? 1 : 0";
 Opacity := 100;
 Access :=
EnumElementAccessFamily.Inherit;
 }
 TextDisplay BAN_LogonLabel {
 Text := "Log on" ->
"::Local:HMIDevice.Security.CurrentUserName=="Guest$"
? $"Log on$" :
::Local:HMIDevice.Security.CurrentUserName";
 FontName := "Arial Unicode MS";
 FontSize := 9;
 Bold := false;
 Underline := false;
 FontColor := "#ffffff";
 TextAlignment :=
EnumAlignment.HLEFT_VCENTER;
 Padding := 0;
 CornerRadius := 0;
 }

```

```

 Border {
 Color := "#000000";
 BorderStyle {
 Line := PenStyle.NoPen;
 Cap := CapStyle.SquareCap;
 Join := JoinStyle.MiterJoin;
 }
 Width := 1;
 }
 FillColor := "#00000000";
 X := 42;
 Y := 8.5;
 Width := 259.78;
 Height := 19;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access :=
EnumElementAccessFamily.Inherit;
 }
 X := 261;
 Y := 7.75;
 Width := 301.78;
 Height := 36;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access := EnumElementAccessFamily.Inherit;
}
PanelDeviceGroup NavigationMenuGroup {
 Button BAN_NavMenuBtn {
 FillColor := "#263a4e";
 Border {
 Color := "#262324";
 BorderStyle {
 Line := PenStyle.SolidLine;
 Cap := CapStyle.SquareCap;
 Join := JoinStyle.MiterJoin;
 }
 Width := 1;
 }
 CornerRadius := 3;
 TextAlignment :=
EnumAlignment.HCENTER_VCENTER;

```

```

 FontName := "Arial Unicode MS";
 FontSize := 8.6;
 FontColor := "#ffffff";
 Bold := true;
 Text := "";
 UsePredefinedDisabled := true;
 OnTouchRelease released_001 {
 ReleaseOnOpenClose := false;
 ShowNavMenuCommand
ShowNavMenuCommand_001 {
 }
 }
 X := 0;
 Y := 0;
 Width := 36;
 Height := 36;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access :=
EnumElementAccessFamily.Inherit;
 }
 NavigationMenuIcon NavigationMenuIcon {
 FillColor := "#e5e5e5";
 X := 4.77;
 Y := 10.64;
 Width := 26.47;
 Height := 14.71;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access :=
EnumElementAccessFamily.Inherit;
 }
 X := 213;
 Y := 7.75;
 Width := 36;
 Height := 36;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access := EnumElementAccessFamily.Inherit;
}

```

```

PanelDeviceGroup NavigationForwardGroup {
 Button BAN_NavForwardBtn {
 FillColor := "#263a4e";
 Border {
 Color := "#262324";
 BorderStyle {
 Line := PenStyle.SolidLine;
 Cap := CapStyle.SquareCap;
 Join := JoinStyle.MiterJoin;
 }
 Width := 1;
 }
 CornerRadius := 3;
 TextAlignment :=
EnumAlignment.HCENTER_VCENTER;
 FontName := "Arial Unicode MS";
 FontSize := 8.6;
 FontColor := "#ffffff";
 Bold := true;
 Text := "";
 UsePredefinedDisabled := true;
 OnTouchRelease released_001 {
 ReleaseOnOpenClose := false;
 NavigateFwdCommand
NavigateFwdCommand_001 {
 }
 }
 X := 0;
 Y := 0;
 Width := 36;
 Height := 36;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access :=
EnumElementAccessFamily.Inherit;
 }
 NextPage NextPage {
 FillColor := "#e5e5e5";
 X := 8.36;
 Y := 8.97;
 Width := 19.31;
 Height := 18.06;
 Angle := 0;
 Enabled := true;
 }
}

```

```

 ^Visible := true;
 Opacity := 100;
 Access :=
EnumElementAccessFamily.Inherit;
 }
 X := 165;
 Y := 7.75;
 Width := 36;
 Height := 36;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access := EnumElementAccessFamily.Inherit;
}
PanelDeviceGroup NavigationBackwardGroup {
 Button BAN_NavBackBtn {
 FillColor := "#263a4e";
 Border {
 Color := "#262324";
 BorderStyle {
 Line := PenStyle.SolidLine;
 Cap := CapStyle.SquareCap;
 Join := JoinStyle.MiterJoin;
 }
 Width := 1;
 }
 CornerRadius := 3;
 TextAlignment :=
EnumAlignment.HCENTER_VCENTER;
 FontName := "Arial Unicode MS";
 FontSize := 8.6;
 FontColor := "#ffffff";
 Bold := true;
 Text := "";
 UsePredefinedDisabled := true;
 OnTouchRelease released_001 {
 ReleaseOnOpenClose := false;
 NavigateBackCommand
NavigateBackCommand_001 {
 }
 }
 }
 X := 0;
 Y := 0;
 Width := 36;
 Height := 36;
 }
}

```



```

 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access :=
EnumElementAccessFamily.Inherit;
 }
 PrevPage PrevPage {
 FillColor := "#e5e5e5";
 X := 8.31;
 Y := 9.05;
 Width := 19.31;
 Height := 18.06;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access :=
EnumElementAccessFamily.Inherit;
 }
 X := 122;
 Y := 7.75;
 Width := 36;
 Height := 36;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access := EnumElementAccessFamily.Inherit;
}
AlarmStatusIndicator BAN_AlarmStatusIndicator {
 NavigateTo := "Navigation Menu\AlarmSummary";
 UsePredefinedDisabled := true;
 FontSize := 8;
 X := 12;
 Y := 7.75;
 Width := 36;
 Height := 36;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access := EnumElementAccessFamily.Inherit;
}
AutomaticDiagnosticsStatusIndicator
BAN_AutomaticDiagnosticsStatusIndicator {

```

```
 NavigateTo := "Navigation
Menu\AutomaticDiagnostics";
 UsePredefinedDisabled := true;
 FontSize := 8;
 X := 55;
 Y := 7.75;
 Width := 36;
 Height := 36;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access := EnumElementAccessFamily.Inherit;
 }
PanelDeviceGroup BAN_StatusAndDateTimeGroup {
 DateTimeDisplay BAN_CurDateDisplay {
 Format := EnumDateTimeFormat.M_d_yyyy;
 FontName := "Arial Unicode MS";
 FontSize := 9;
 Bold := false;
 FontColor := "#ffffff";
 TextAlignment :=
EnumAlignment.HRIGHT_VCENTER;
 Padding := 0;
 CornerRadius := 3;
 Border {
 Color := "#738596";
 BorderStyle {
 Line := PenStyle.NoPen;
 Cap := CapStyle.SquareCap;
 Join := JoinStyle.MiterJoin;
 }
 Width := 1;
 }
 FillColor := "#00000000";
 X := 122.4;
 Y := 16.25;
 Width := 85;
 Height := 19;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access :=
EnumElementAccessFamily.Inherit;
 }
}
```

```

DateTimeDisplay BAN_CurTimeDisplay {
 Format := EnumDateTimeFormat.hh_mm_ss_AP;
 FontName := "Arial Unicode MS";
 FontSize := 9;
 Bold := false;
 FontColor := "#ffffff";
 TextAlignment :=
EnumAlignment.HRIGHT_VCENTER;
 Padding := 0;
 CornerRadius := 3;
 Border {
 Color := "#738596";
 BorderStyle {
 Line := PenStyle.NoPen;
 Cap := CapStyle.SquareCap;
 Join := JoinStyle.MiterJoin;
 }
 Width := 1;
 }
 FillColor := "#00000000";
 X := 122.4;
 Y := 0;
 Width := 85;
 Height := 19;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access :=
EnumElementAccessFamily.Inherit;
}

ControllerStatusIndicator
BAN_ControllerStatusIndicator {
 NavigateTo := "Predefined
Screens\ControllersGeneral";
 X := 81.6;
 Y := 0.25;
 Width := 40.8;
 Height := 36;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access :=
EnumElementAccessFamily.Inherit;
}

```

```

NetworkStatusIndicator
BAN_NetworkStatusIndicator {
 NavigateTo := "Predefined
Screens\NetworkDiagnostic";
 X := 40.8;
 Y := 0.25;
 Width := 40.8;
 Height := 36;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access :=
EnumElementAccessFamily.Inherit;
}

ProjectEventsStatusIndicator
BAN_ProjectEventsStatusIndicator {
 StateTable ^StateTable {
 Expression := "" ->
"::Local:HMIDevice.ProjectEvents.ErrorCount != 0&n&&
(nt!::Local:HMIDevice.Alarms.ExportInProgress&n$t&&
::Local:HMIDevice.Alarms.ExportErrorStatus == $"$"n$t&&
!::Local:HMIDevice.DataLogs.ExportInProgress&n$t&&
::Local:HMIDevice.DataLogs.DataLoggingError ==
$" $"n$t&& ::Local:HMIDevice.DataLogs.ExportErrorStatus
== $" $"n$t&&
::Local:HMIDevice.DataLogs.DataLoggingStatus !=
$"SAFELY_STOPPED$"n$t&&
!::Local:HMIDevice.DataLogs.TagsInError$n) $n";
 State State0 (
 opacity := "0",
 ^visible := "False")
 {
 value := "0";
 }
 State State1 (
 opacity := "100",
 ^visible := "true")
 {
 value := "1";
 }
 State DefaultState (
 opacity := "100",
 ^visible := "true")
 {
 }
 }
 X := 0;
 Y := 0.25;
}

```

```

Width := 40.8;
Height := 36;
Angle := 0;
Enabled := true;
^Visible := true;
Opacity := 100;
Access :=
EnumElementAccessFamily.Inherit;
}
DataLogStatusIndicator
BAN_DataLogStatusIndicator {
NavigateTo := "Predefined
Screens\DataLog";
StateTable ^StateTable {
Expression := "" ->
"::Local:HMIDevice.ProjectEvents.ErrorCount == 0$n&& ($n
::Local:HMIDevice.Alarms.ExportInProgress$n ||
::Local:HMIDevice.Alarms.ExportErrorStatus != $"$"n
|| ::Local:HMIDevice.DataLogs.ExportInProgress$n ||
::Local:HMIDevice.DataLogs.DataLoggingError != $"$"n
|| ::Local:HMIDevice.DataLogs.ExportErrorStatus !=
$" $"n ||
::Local:HMIDevice.DataLogs.DataLoggingStatus ==
$"SAFELY_STOPPED$"n ||
::Local:HMIDevice.DataLogs.TagsInError$n) $n";
State State0 (
opacity := "0",
^visible := "False")
{
value := "0";
}
State State1 (
opacity := "100",
^visible := "true")
{
value := "1";
}
State DefaultState (
opacity := "100",
^visible := "true")
{
}
}
X := 0;
Y := 0.25;
Width := 40.8;
Height := 36;
Angle := 0;
Enabled := true;

```

```

 ^Visible := true;
 Opacity := 100;
 Access :=
EnumElementAccessFamily.Inherit;
 }
 PanelDeviceGroup BAN_ErrorIndicator {
 Rectangle ErrorIndicatorBg {
 CornerRadius := 0;
 Border {
 Color := "#000000";
 BorderStyle {
 Line :=
PenStyle.SolidLine;
 Cap :=
CapStyle.SquareCap;
 Join :=
JoinStyle.MiterJoin;
 }
 Width := 0;
 }
 FillColor := "#738596";
 X := 0;
 Y := 0;
 Width := 40.8;
 Height := 36;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 0;
 Access :=
EnumElementAccessFamily.Inherit;
 }
 ErrorIndicator ErrorIndicator {
 X := 5.4;
 Y := 3;
 Width := 30;
 Height := 30;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access :=
EnumElementAccessFamily.Inherit;
 }
 StateTable StateTable_001 {
 Expression := "" ->
"::Local:HMIDevice.ProjectEvents.ErrorCount != 0$n&&

```

```

(nt::Local:HMIDevice.Alarms.ExportInProgressnt||
::Local:HMIDevice.Alarms.ExportErrorStatus != "$"$n$t||
::Local:HMIDevice.DataLogs.ExportInProgressnt||
::Local:HMIDevice.DataLogs.DataLoggingError !=
"$"$n$t|| ::Local:HMIDevice.DataLogs.ExportErrorStatus
!= "$"$n$t||
::Local:HMIDevice.DataLogs.DataLoggingStatus ==
"$SAFELY_STOPPED$"n$t||
::Local:HMIDevice.DataLogs.TagsInError$n) $n";

 State State0 (
 opacity := "0",
 ^visible := "False")
 {
 value := "0";
 }
 State State1 (
 opacity := "100",
 ^visible := "true")
 {
 value := "1";
 }
 State DefaultState (
 opacity := "100",
 ^visible := "true")
 {
 }
 }
 BehaviorOpenPopUp BehaviorOpenPopUp_001 {
 screenName := "Predefined
Screens\WarningsAndErrors";
 Key := EnumBezelKeys.VK_NONE;
 RequiresFocus := false;
 AlwaysTriggerReleaseEvent := false;
 }
 X := 0;
 Y := 0.25;
 Width := 40.8;
 Height := 36;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access :=
EnumElementAccessFamily.Inherit;
}
X := 580.6;
Y := 7.5;
Width := 207.4;

```

```
 Height := 36.25;
 Angle := 0;
 Enabled := true;
 ^Visible := true;
 Opacity := 100;
 Access := EnumElementAccessFamily.Inherit;
 }
 Width := 800;
 Height := 51.5;
}
```



## Data Log syntax

The DataLog.hmi files contain information about the data log configuration values and the tags which it logs. There is a DataLog.hmi file for each data log configured in your project. The file follows the structure:

```
namespace ViewDesigner;

using HMICatalog;

/* HMI Device Type: 9" PanelView 5510 2715P-T9WD */

DataLog [name] {
 <data log configuration values>
 <list of tags to log>
}
```

For example:

```
namespace ViewDesigner;

using HMICatalog;

/* HMI Device Type: 9" PanelView 5510 2715P-T9WD */

DataLog DataLog_001 {
 SamplingRate := EnumSamplingRate._1000_ms;
 Duration := 7;
 DurationUnit := EnumDurationUnit.Days;
 StatusTag := "";
 DataLogEntry DataLogEntry_001 {
 TagName := "::LGX.Mixer1.Level";
 }
 DataLogEntry DataLogEntry_002 {
 TagName := "::LGX.Mixer1.LevelSP";
 }
}
```

## Event .hmi conventions

This section lists the events, behaviors and commands that can be configured in a View Designer project, as they appear in the exported .hmi files.

List of events:

- OnTouchPress
- OnTouchRelease
- OnKeyPress
- OnKeyRelease
- OnStateEnter

- OnStateExit

List of behaviors:

- BehaviorLogonOnRelease
- BehaviorLogoffOnRelease
- BehaviorNavigateBackward
- BehaviorNavigateForward
- BehaviorShowNavigateMenu
- BehaviorClosePopUp
- BehaviorSwitchLanguageOnRelease
- BehaviorOpenPopUp
- BehaviorNavigateToScreen
- BehaviorSetTagTo1OnRelease
- BehaviorSetTagTo0OnRelease
- BehaviorToggleTagOnRelease
- BehaviorHighSpeedWriteTo1OnPress0OnRelease
- BehaviorTagOnPressOnReleaseBase
- BehaviorSetTagTo1OnPress0OnRelease
- BehaviorSetTagTo0OnPress1OnRelease
- BehaviorWriteToTagBase
- BehaviorWriteToTagOnRelease
- BehaviorIncrementTagOnRelease

List of commands:

- Alarm history:
  - AlarmHistoryExportCancellationCommand
  - ClearAlarmHistoryCommand
  - AlarmHistoryExportCommand
- Automatic Diagnostics:
  - ClearADDAHistoryCommand
- Certificate:
  - DeleteCertificateCommand
  - ImportCertificateCommand
- Data Log
  - DLExportCancelCommand
  - ClearDataLogCommand
  - DLExportCommand
- HMI Device Configuration:
  - CalibrateTouchscreenCommand
  - DownloadFromMediaCommand
  - RebootTerminalCommand
  - SafelyRemoveMediaCommand
- Language:
  - LanguageCommand

- Navigation:
  - CloseWebBrowserCommand
  - NavigateBackCommand
  - NavigateFwdCommand
  - OpenWebBrowserCommand
  - ClosePopupCommand
  - DisplayPopupCommand
  - ScreenNavCommand
  - ShowNavMenuCommand
- Notification
  - SendEmailCommand
- PDF:
  - PDFGoToPageCommand
  - PDFNextPageCommand
  - PDFPreviousPageCommand
  - PDFZoomInCommand
  - PDFZoomOutCommand
  - PDFZoomCommand
- Security:
  - DoLogOffCommand
  - DisplayLogOnPopupCommand
- State:
  - NextStateCommand
  - PreviousStateCommand
- Value:
  - IncrementCommand
  - ToggleCommand
  - SetCommand

The following is an example of an event with a command, as seen in an .hmi file:

```
OnTouchPress TouchPress_001 {
 ReleaseOnOpenClose := true;
 AlarmHistoryExportCancellationCommand
AlarmHistoryExportCancellationComman_001 {
 }
}
```

The following is an example of a behavior, as seen in an .hmi file:

```
BehaviorNavigateToScreen BehaviorNavigateToScreen_001 {
 screenName := "Navigation Menu\Settings";
 Key := EnumBezelKeys.VK_NONE;
```

```

RequiresFocus := false;
AlwaysTriggerReleaseEvent := false;
}

```

## Common enums

This table lists enums used in the .hmi files, together with their values. Enum values are normally exported as text (left column), but if you use aliases for enums in View Designer, the values are exported as a corresponding integer (right column). In addition, if these properties are used in a state table, the import/export format will use the integer value in the state table instead of the text.

| Enum Value              | Exported value (integer) |
|-------------------------|--------------------------|
| AlarmDateTimeFormatEnum |                          |
| ISO8601ZZZ              | 0                        |
| Default                 | 1                        |
| NamedMonth              | 2                        |
| ISO8601                 | 3                        |
| Euro                    | 4                        |
| AlarmDataItemEnum       |                          |
| PriorityText            | 1                        |
| PriorityImage           | 2                        |
| Severity                | 3                        |
| AlarmStateText          | 4                        |
| AlarmStateImage         | 5                        |
| InhibitStateText        | 6                        |
| InhibitStateImage       | 7                        |
| EventTime               | 8                        |
| ConditionName           | 9                        |
| AlarmName               | 10                       |
| Message                 | 11                       |
| InAlarmTime             | 12                       |
| AcknowledgeTime         | 13                       |
| OutofAlarmTime          | 14                       |
| EnableTime              | 15                       |
| DisableTime             | 16                       |
| SuppresTime             | 17                       |
| UnsuppresTime           | 18                       |
| ShelveTime              | 19                       |
| UnshelveTime            | 20                       |
| AlarmClass              | 21                       |
| LimitValueExceeded      | 22                       |
| Tag1Value               | 23                       |
| Tag2Value               | 24                       |
| Tag3Value               | 25                       |
| Tag4Value               | 26                       |
| AlarmCount              | 27                       |
| CurrentValue            | 28                       |
| EventCategory           | 29                       |

| Enum Value          | Exported value (integer) |
|---------------------|--------------------------|
| ShelveDuration      | 30                       |
| MaxShelveDuration   | 31                       |
| EnableUser          | 32                       |
| DisableUser         | 33                       |
| ShelveUser          | 34                       |
| AckUser             | 35                       |
| EnableComment       | 36                       |
| DisableComment      | 37                       |
| ShelveComment       | 38                       |
| AcknowledgeComment  | 39                       |
| Quality             | 40                       |
| UnshelveUser        | 41                       |
| UnshelveComment     | 42                       |
| AlarmGroup          | 43                       |
| OffDelay            | 44                       |
| OnDelay             | 45                       |
| ConditionType       | 46                       |
| InputTag            | 47                       |
| TargetTag           | 48                       |
| UseAndEvaluateAlarm | 49                       |
| Expression          | 50                       |
| AHVDataItemEnum     |                          |
| PriorityText        | 1                        |
| PriorityImage       | 2                        |
| Severity            | 3                        |
| AlarmStateText      | 4                        |
| AlarmStateImage     | 5                        |
| InhibitStateText    | 6                        |
| InhibitStateImage   | 7                        |
| EventTime           | 8                        |
| ConditionName       | 9                        |
| AlarmName           | 10                       |
| Message             | 11                       |
| InAlarmTime         | 12                       |
| AcknowledgeTime     | 13                       |
| OutofAlarmTime      | 14                       |
| EnableTime          | 15                       |
| DisableTime         | 16                       |
| SuppressTime        | 17                       |
| UnsuppressTime      | 18                       |
| ShelveTime          | 19                       |
| UnshelveTime        | 20                       |
| AlarmClass          | 21                       |
| LimitValueExceeded  | 22                       |
| Tag1Value           | 23                       |
| Tag2Value           | 24                       |
| Tag3Value           | 25                       |
| Tag4Value           | 26                       |
| AlarmCount          | 27                       |
| CurrentValue        | 28                       |

| Enum Value           | Exported value (integer) |
|----------------------|--------------------------|
| EventCategory        | 29                       |
| ShelveDuration       | 30                       |
| MaxShelveDuration    | 31                       |
| EnableUser           | 32                       |
| DisableUser          | 33                       |
| ShelveUser           | 34                       |
| AckUser              | 35                       |
| EnableComment        | 36                       |
| DisableComment       | 37                       |
| ShelveComment        | 38                       |
| AcknowledgeComment   | 39                       |
| Quality              | 40                       |
| UnshelveUser         | 41                       |
| UnshelveComment      | 42                       |
| AlarmGroup           | 43                       |
| OffDelay             | 44                       |
| OnDelay              | 45                       |
| ConditionType        | 46                       |
| InputTag             | 47                       |
| TargetTag            | 48                       |
| UseAndEvaluateAlarm  | 49                       |
| Expression           | 50                       |
| StateChangeType      | 51                       |
| TableColumnAlignment |                          |
| AlignLeft            | 0                        |
| AlignCenter          | 1                        |
| AlignRight           | 2                        |
| TableColumnEliding   |                          |
| ElideNone            | 0                        |
| ElideLeft            | 1                        |
| ElideMiddle          | 2                        |
| ElideRight           | 3                        |
| ADDDataItemEnum      |                          |
| Message              | 1                        |
| DeviceName           | 2                        |
| StateText            | 3                        |
| DiagnosticCode       | 4                        |
| Resource             | 5                        |
| Catalog              | 6                        |
| EventTime            | 7                        |
| ProductType          | 8                        |
| Vendor               | 9                        |
| ProductName          | 10                       |
| Version              | 11                       |
| StateIcon            | 12                       |
| StateChange          | 13                       |
| EnumAlignment        |                          |
| HLEFT_VTOP           | 33                       |
| HLEFT_VCENTER        | 129                      |

| Enum Value          | Exported value (integer) |
|---------------------|--------------------------|
| HLEFT_VBOTTOM       | 65                       |
| HCENTER_VTOP        | 36                       |
| HCENTER_VCENTER     | 132                      |
| HCENTER_VBOTTOM     | 68                       |
| HRIGHT_VTOP         | 34                       |
| HRIGHT_VCENTER      | 130                      |
| HRIGHT_VBOTTOM      | 66                       |
| PenStyle            |                          |
| NoPen               | 0                        |
| DashDotLine         | 4                        |
| CustomDashLine      | 6                        |
| DotLine             | 3                        |
| DashLine            | 2                        |
| DashDotDotLine      | 5                        |
| SolidLine           | 1                        |
| CapStyle            |                          |
| FlatCap             | 0                        |
| RoundCap            | 32                       |
| SquareCap           | 16                       |
| JoinStyle           |                          |
| MiterJoin           | 0                        |
| BevelJoin           | 64                       |
| RoundJoin           | 128                      |
| EnumBezelKeys       |                          |
| VK_L1               | 0                        |
| VK_L2               | 1                        |
| VK_L3               | 2                        |
| VK_L4               | 3                        |
| VK_L5               | 4                        |
| VK_L6               | 5                        |
| VK_L7               | 6                        |
| VK_L8               | 7                        |
| VK_L9               | 8                        |
| VK_L10              | 9                        |
| VK_R1               | 10                       |
| VK_R2               | 11                       |
| VK_R3               | 12                       |
| VK_R4               | 13                       |
| VK_R5               | 14                       |
| VK_R6               | 15                       |
| VK_R7               | 16                       |
| VK_R8               | 17                       |
| VK_R9               | 18                       |
| VK_R10              | 19                       |
| VK_None             | 20                       |
| EnumNumericRounding |                          |
| Nearest             | 0                        |
| Up                  | 1                        |
| None                | 2                        |
| RoleAccess          |                          |
| FullAccess          | 0                        |
| ReadOnly            | 1                        |
| NoAccess            | 2                        |

| Enum Value                   | Exported value (integer) |
|------------------------------|--------------------------|
| Inherit                      | 8                        |
| EnumUpdateRate               |                          |
| 100 ms                       | 100                      |
| 250 ms                       | 250                      |
| 500 ms                       | 500                      |
| EnumTerminalMediaType        |                          |
| USB                          | 0                        |
| SD Card                      | 1                        |
| EnumHMIttoControllerPathType |                          |
| PathsOnHMIDevice             | 0                        |
| PathsInApplication           | 1                        |
| EnumTopBar                   |                          |
| Normal                       | 0                        |
| Slim                         | 1                        |
| NormalReadOnly               | 2                        |
| NormalEditable               | 3                        |
| EnumSamplingRate             |                          |
| _500_ms                      | 500                      |
| _1000_ms                     | 1000                     |
| _5000_ms                     | 5000                     |
| _10000_ms                    | 10000                    |
| _30000_ms                    | 30000                    |
| _60000_ms                    | 60000                    |
| EnumDurationUnit             |                          |
| Minutes                      | 0                        |
| Hours                        | 1                        |
| Days                         | 2                        |
| EnumDateTimeFormat           |                          |
| yyyy_MM_dd_hh_mm_ss_zzzzzz   | 0                        |
| M_dd_yyyy_h_mm_ss_AP         | 1                        |
| d MMM_yyyy_hh_mm_ss          | 2                        |
| yyyy_MM_dd_hh_mm_ss          | 3                        |
| dd_M_yyyy_hh_mm_ss           | 4                        |
| dddd_MMMM_d_yyyy             | 5                        |
| MMMM_d_yyyy                  | 6                        |
| d_MMMM_yyyy                  | 7                        |
| d_MMM_yyyy                   | 8                        |
| yyyy_MM_dd                   | 9                        |
| M_d_yyyy                     | 10                       |
| d_M_yyyy                     | 11                       |
| dddd                         | 12                       |
| MMMM_d                       | 13                       |
| yyyy                         | 14                       |
| hh_mm_ss_zzz_AP              | 15                       |
| hh_mm_ss_AP                  | 16                       |
| hh_mm_ss                     | 17                       |
| hh_mm_AP                     | 18                       |
| hh                           | 19                       |
| mm                           | 20                       |
| ss                           | 21                       |



| Enum Value                                     | Exported value (integer) |
|------------------------------------------------|--------------------------|
| yyyy_MM_dd_hh_mm_ss.ms.ms.ms.us.us.us          | 22                       |
| yyyy_MM_dd_hh_mm_ss.ms.ms.ms.us.us.us.ns.ns.ns | 23                       |
| IndustrialThermometerEnumType                  |                          |
| Celsius                                        | 1                        |
| Fahrenheit                                     | 0                        |
| KeypadPositionEnum                             |                          |
| TopLeft                                        | 1                        |
| TopCenter                                      | 2                        |
| TopRight                                       | 3                        |
| MiddleLeft                                     | 4                        |
| MiddleCenter                                   | 0                        |
| MiddleRight                                    | 5                        |
| BottomLeft                                     | 6                        |
| BottomCenter                                   | 7                        |
| BottomRight                                    | 8                        |
| EnumSliderThumbShape                           |                          |
| Square                                         | 0                        |
| Octagon                                        | 1                        |
| Diamond                                        | 2                        |
| Circle                                         | 3                        |
| EnumSliderWriteMode                            |                          |
| ContinuousUpdate                               | 0                        |
| UpdateOnRelease                                | 1                        |
| PdfZoomModeEnum                                |                          |
| Fit_Width                                      | 0                        |
| Fit_Page                                       | 1                        |
| Actual Size                                    | 2                        |
| EnumTrendChartTimeFormats                      |                          |
| GeneralTimeDateUS                              | 1                        |
| DateTimeWorld                                  | 2                        |
| ISOFormat                                      | 3                        |
| UniversalFormat                                | 4                        |
| ShortDateUS                                    | 5                        |
| ShortDateWorld                                 | 6                        |
| ShortISOFormat                                 | 7                        |
| ShortUniversalFormat                           | 8                        |
| EnumTraceMarkers                               |                          |
| None                                           | 0                        |
| Star                                           | 1                        |
| Cross                                          | 2                        |
| Circle                                         | 3                        |
| Square                                         | 4                        |
| Triangle                                       | 5                        |
| FilledCircle                                   | 6                        |
| FilledSquare                                   | 7                        |
| FilledTriangle                                 | 8                        |

## Rockwell Automation support

Use these resources to access support information.

|                                                         |                                                                                                            |                                                                    |
|---------------------------------------------------------|------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------|
| <b>Technical Support Center</b>                         | Find help with how-to videos, FAQs, chat, user forums, and product notification updates.                   | <a href="http://rok.auto/support">rok.auto/support</a>             |
| <b>Knowledgebase</b>                                    | Access Knowledgebase articles.                                                                             | <a href="http://rok.auto/knowledgebase">rok.auto/knowledgebase</a> |
| <b>Local Technical Support Phone Numbers</b>            | Locate the telephone number for your country.                                                              | <a href="http://rok.auto/phonesupport">rok.auto/phonesupport</a>   |
| <b>Literature Library</b>                               | Find installation instructions, manuals, brochures, and technical data publications.                       | <a href="http://rok.auto/literature">rok.auto/literature</a>       |
| <b>Product Compatibility and Download Center (PCDC)</b> | Get help determining how products interact, check features and capabilities, and find associated firmware. | <a href="http://rok.auto/pcdc">rok.auto/pcdc</a>                   |

## Documentation feedback

Your comments help us serve your documentation needs better. If you have any suggestions on how to improve our content, complete the form at [rok.auto/docfeedback](http://rok.auto/docfeedback).

## Waste Electrical and Electronic Equipment (WEEE)



At the end of life, this equipment should be collected separately from any unsorted municipal waste.





Rockwell Automation maintains current product environmental information on its website at [rok.auto/pec](http://rok.auto/pec).

Allen-Bradley, expanding human possibility, Logix, Rockwell Automation, and Rockwell Software are trademarks of Rockwell Automation, Inc.

EtherNet/IP is a trademark of ODVA, Inc.

Trademarks not belonging to Rockwell Automation are property of their respective companies.

Rockwell Otomasyon Ticaret A.Ş. Kar Plaza İş Merkezi E Blok Kat:6 34752, İçerenköy, İstanbul, Tel: +90 (216) 5698400 EEE Yönetmeliğine Uygundur

Connect with us.    

**rockwellautomation.com** ————— expanding **human possibility**™

AMERICAS: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444

EUROPE/MIDDLE EAST/AFRICA: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640

ASIA PACIFIC: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846