FactoryTalk View ME to Studio 5000 View Designer Conversion Tool User Manual

Rockwell Automation

User Manual

Original Instructions

Contents

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.

These 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).

The following icon may appear in the text of this document.

Tip: Identifies information that is useful and can help to make a process easier to do or easier to understand.

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.

Conversion Overview

The View ME to View Designer conversion tool takes displays exported from View ME in the View ME `.xml` format, and converts them to the View Designer import format introduced in View Designer Version 9.

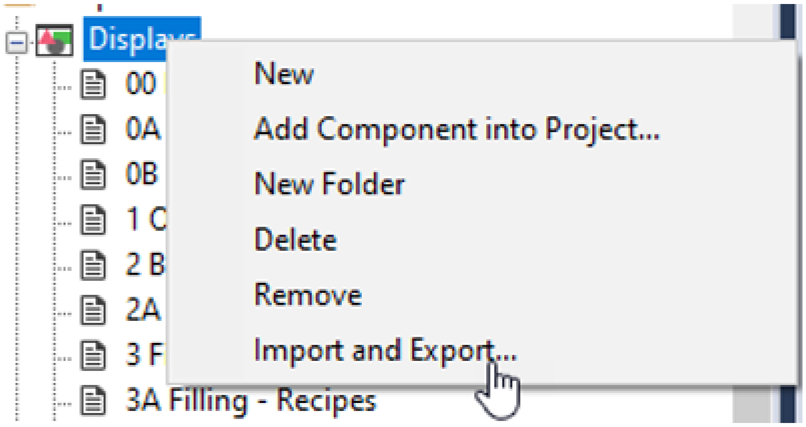
Tip: This is not a 100% conversion. Simple applications and objects will tend to convert more completely but will still require some work in View Designer, and more complex applications and objects will require more work after the conversion to View Designer. The conversion tool is focused on screen content. Other View ME project content such as data logs or users will need to be manually created in View Designer.

The View ME to View Designer tool is tested to work with View ME V13 projects converted to View Designer V9. Other View ME versions may also work but have not been tested. View Designer V9 is the earliest version which supports the import/export feature, so earlier View Designer versions cannot be supported.

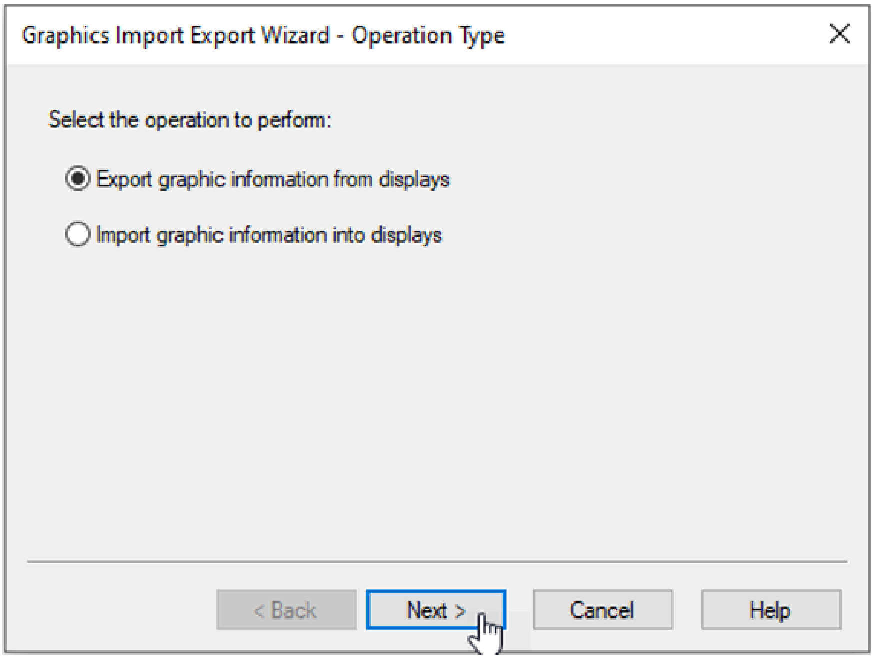
Workflow

Use the export feature of View ME to export your View ME displays.

1. In View ME, right-click on the **Displays** folder and select **Import and Export...**.



2. Select **Export graphic information from displays**, and select **Next**.



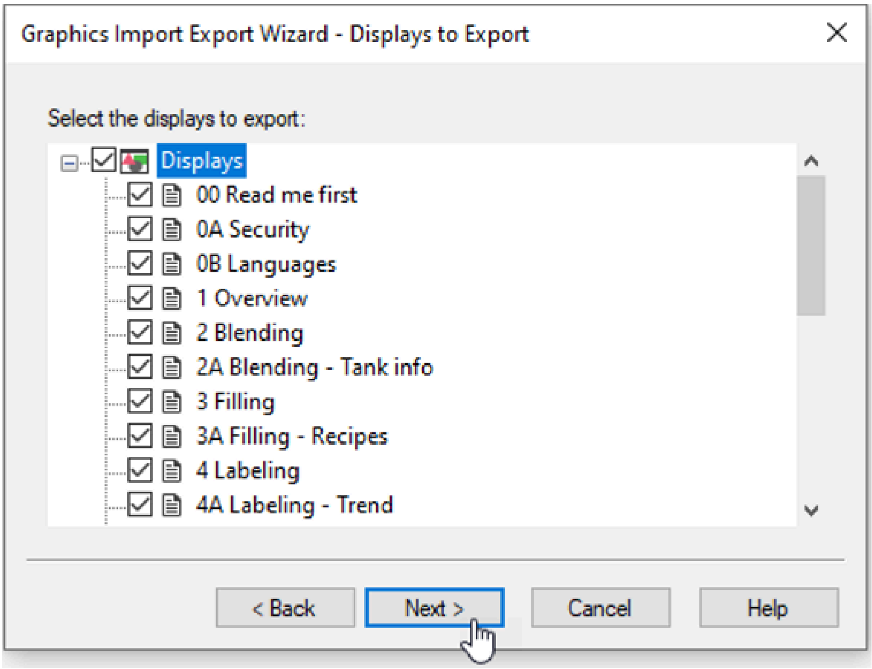
Graphics Import Export Wizard - Operation Type

Select the operation to perform:

● Export graphic information from displays

○ Import graphic information into displays

3. Select the displays that you want to export, and select **Next**.

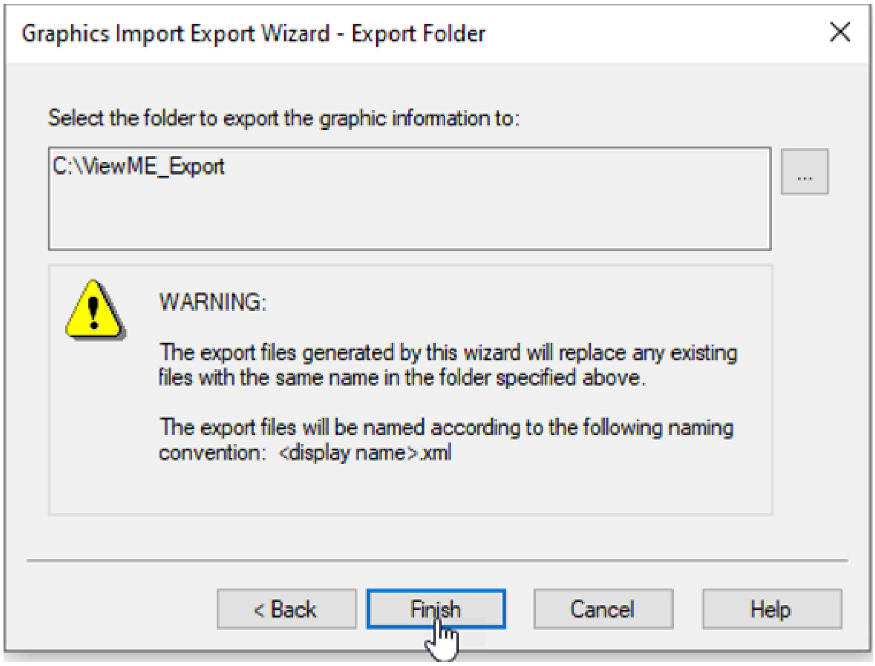


Graphics Import Export Wizard - Displays to Export

Select the displays to export:

- ☑ Displays
  - ☑ 00 Read me first
  - ☑ 0A Security
  - ☑ 0B Languages
  - ☑ 1 Overview
  - ☑ 2 Blending
  - ☑ 2A Blending - Tank info
  - ☑ 3 Filling
  - ☑ 3A Filling - Recipes
  - ☑ 4 Labeling
  - ☑ 4A Labeling - Trend

4. Select a folder where you want to export the View ME displays, and select **Finish**.



Graphics Import Export Wizard - Export Folder

Select the folder to export the graphic information to:

C:\ViewME_Export

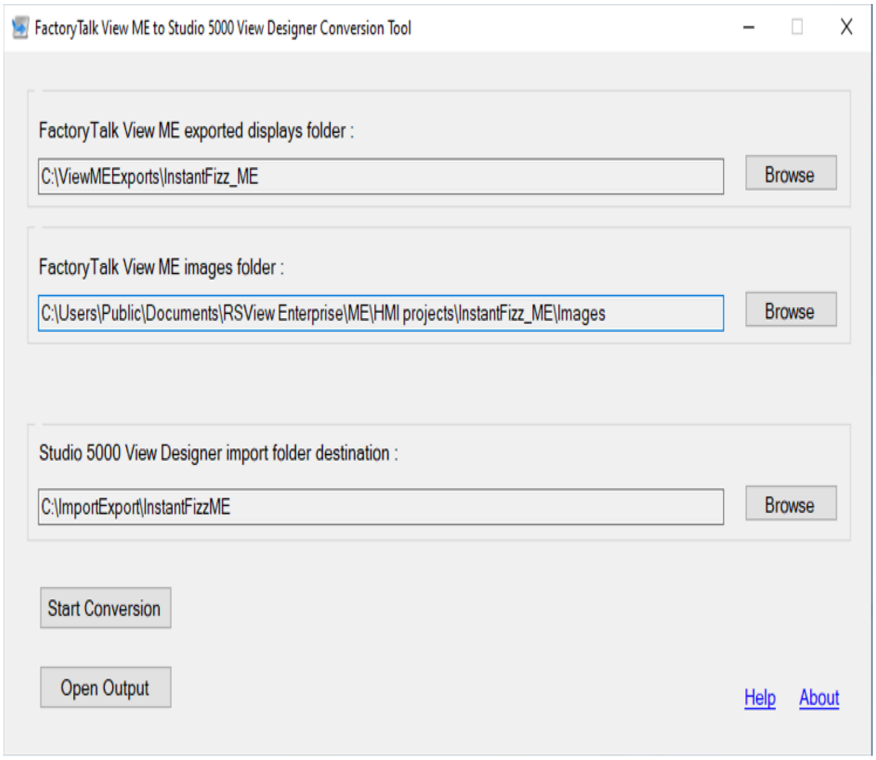
WARNING:

The export files generated by this wizard will replace any existing files with the same name in the folder specified above.

The export files will be named according to the following naming convention: <display name>.xml

You will select this folder in the View ME to View Designer conversion tool.

5. Run the View ME to View Designer conversion tool.



FactoryTalk View ME to Studio 5000 View Designer Conversion Tool

FactoryTalk View ME exported displays folder : C:\ViewMEExports\InstantFizz_ME

FactoryTalk View ME images folder : C:\Users\Public\Documents\RSView Enterprise\ME\HMI projects\InstantFizz_ME\Images

Studio 5000 View Designer import folder destination : C:\ImportExport\InstantFizzME

6. In the **FactoryTalk View ME exported displays folder** field, enter the folder where you exported the View ME displays to their xml format in step 4.

7. In the **FactoryTalk View ME images folder** field, enter your View ME project image directory path.

   The default image directory path is `C:\Users\Public\Documents\RSView Enterprise\ME\HMI projects\`*`My Project Name`*`\Images`

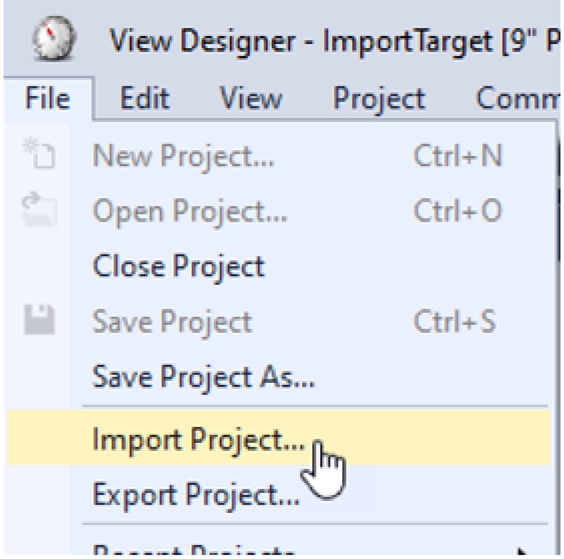
   where *My Project Name* is the name of your View ME project.

IMPORTANT: The image type must match the image file extension for the import to succeed.

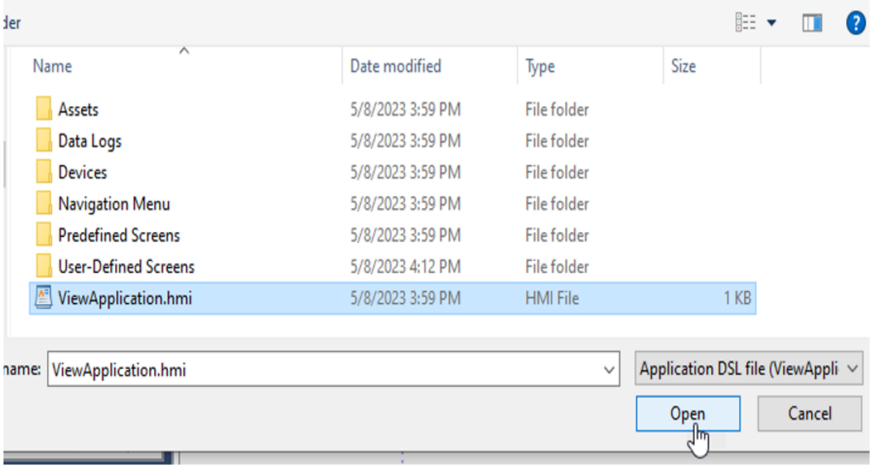
The conversion tool moves these images into the correct folder location to be imported into your View Designer project.

8. In the **Studio 5000 View Designer import folder destination**, enter the location where the conversion tool will create the View Designer import files
9. Press **Start Conversion**.

   The conversion tool displays the progress of the conversion and any error information.
10. Open View Designer and create a target project in View Designer for the panel size you wish to import the converted View ME screens. For best results, the target project should have the same resolution as the source View ME project.
11. In View Designer, select **File > Import Project**.



12. Navigate to the converted View Designer import folder directory selected in step 5. Select the `ViewApplication.hmi` file and select **Open**.



Name	Date modified	Type	Size
Assets	5/8/2023 3:59 PM	File folder	
Data Logs	5/8/2023 3:59 PM	File folder	
Devices	5/8/2023 3:59 PM	File folder	
Navigation Menu	5/8/2023 3:59 PM	File folder	
Predefined Screens	5/8/2023 3:59 PM	File folder	
User-Defined Screens	5/8/2023 4:12 PM	File folder	
ViewApplication.hmi	5/8/2023 3:59 PM	HMI File	1 KB

name: ViewApplication.hmi

Application DSL file (ViewAppli

The View Designer imports the screen content available from the conversion. The import overwrites any existing content with matching name.

Component Conversion Details

This section details how different View ME project components are handled by the converter.

Refer to the *Studio 5000 View Designer Help* and the *Studio 5000 View Designer Getting Results Guide* for more information.

Alarms

Alarms in View Designer and the PanelView 5000 are handled as Logix-based alarms using either the ALMD and ALMA alarm instructions in Logix or using tag-based alarms available starting in Logix V31. The standard View ME alarms will not be converted and will need to be created as alarms in the Logix controller.

Expressions

Simple expressions (such as MyTag*2) generally convert. However, more complex expressions will need to be reworked after the conversion since the expression syntax is different in View Designer and FactoryTalk View ME. Refer to View Designer help and the View Designer Getting Results Guide for more information.

Animations

Visibility animation is converted to a binding on the Visible property of an element. All other animations, such as Fill, Rotation, etc. are not converted and must be reconfigured on the converted element using bindings, state tables, and/or events. Refer to the *View Designer Help* and the *View Designer Getting Results Guide* for more information

Users and Security

Users and security settings will need to be recreated in View Designer. Configure users and Auto-Logoff settings in View Designer under Tools > Security Administration. View Designer uses role-based security where each user is given a role, and each screen defines what the different roles can do with that screen under the screen's Security properties. View Designer does not use the A-P codes as in View ME. Refer to View Designer help and the View Designer Getting Results Guide for more information.

Diagnostics List

Diagnostics on the PanelView 5000 are displayed directly on the screen element which has an error. The element will show a `Circle X` error indication which shows a detailed error message when pressed.

Audit Trail

Audit trail is not supported on the PanelView 5000.

CSV Export

Alarm history and data logs can be exported to `.csv` files using the pre-defined Data Export settings screen on the PanelView 5000.

Global Connections

Global Connections are not supported on the PanelView 5000. However, many of those items can be displayed or configured on the pre-defined settings screens. Remote screen control can also be accomplished by using Project Events.

Startup

The initial display in View Designer is configured as the home screen. Data logging and alarms automatically start at runtime.

HMI Tags

View Designer does not use HMI tags other than local tags which display or set HMI device information. View Designer uses Logix controller tags to display and set data values.

Local Messages

View Designer does not support local messages. You can use a state table on a Text Display to display up to 2500 different text strings.

Information Messages

View Designer does not support information messages.

Macros

View Designer does not support macros. Instead, any element can have events defined which can execute one or more commands to set tag values or perform other actions. Project events can also be used to execute commands when expressions transition from false to true. Refer to View Designer help and the View Designer Getting Results Guide for more information.

Data Logs

View Designer and the PanelView 5000 support up to 3 data logs which can log up to 250 tags total across the three logs. Up to 1.296 billion records can be logged.Refer to View Designer help and the View Designer Getting Results Guide for more information.

RecipePlus

View Designer does not support RecipePlus. Recipes can be accomplished using pre-defined screens and Logix code as described in tech note *QA50469*.

FactoryTalk Linx Communication Setup

The PanelView 5000 does not use FactoryTalk Linx to communicate with the Logix controller. In View Designer, set up controller references under Project > Project Properties > References to define your paths from the HMI to the controllers. The PanelView 5310 supports a single controller, and the PanelView 5510 supports up to four controllers. To get your tag bindings to work with the converted content from View ME, you can name your controller reference the same as the FactoryTalk Linx shortcut name of the controller in View ME.

Displays

Displays in View ME will convert to either a full screen or a popup in View Designer.

NOTE:

- Display Type indicates if the View ME display will convert as a screen or a popup:
  - Replace converts to a screen.
  - On Top converts to a popup. The popup Width and Height will match the View ME screen Width and Height.
  - Cannot Be Replaced is ignored
- Display Number and Security Code are ignored.
- Title Bar converts to the Popup Caption – inserted variables are ignored.
- Disable Initial Input Focus is ignored.
- Maximum Tag Update Rate converts to UpdateRate. If greater than 500 ms, use 500 ms; if less than 100 ms, use 100 ms.
- Size converts to popup Width and Height if Display Type is On Top.
- Position is ignored.
- Background Color converts to FillColor.
- Use Gradient Style is ignored.
- Behavior of Object with Input Focus is ignored.

Global Object conversion

- Global objects screens will convert to standard screens in View Designer.
- The objects in the global object screen convert to elements on the View Designer screen.
- Property passing will need to be re-worked after the conversion.
- Global Object instances on screens convert to elements on the screen. Any grouping information is lost on the conversion.

Conversion by View ME Object Type

The descriptions show what View Designer element the View ME object converts to and how different properties of the View ME object convert to properties of the View Designer element.

- **Text**
  - Converts to a Text Display
  - Alignment converts to TextAlignment
  - BackColor converts to FillColor
  - BackStyle of Transparent converts to a FillColor of "NoBrush"
  - Caption converts to Text property
  - Font converts to FontName using the same named font in View Designer if it exists; otherwise, it converts to Arial Unicode MS
  - ForeColor converts to FontColor
  - Height and Width convert to Height and Width
  - Left and Top convert to X and Y
  - SizeToFit is ignored (no matching property in View Designer)
  - Visible converts to Visible
  - WordWrap is ignored
  - Font size is converted to FontSize

    - Bold and Underline selections are converted to Bold and Underline; other selections (italic, strikeout) are ignored
    - Alignment is converted to TextAlignment
    - Variables inserted are ignored.
- **Image**
    - Converts to an Image
    - Height and Width convert to Height and Width
    - Image converts to a file in the Images folder
    - ImageBackColor, ImageBackStyle, ImageBlink, and ImageColor are ignored.
    - Left and Top convert to X and Y
    - Visible converts to Visible
- **Panel**
    - Depending on BorderStyle:
        - None, Raised, and Line convert to Panel2
        - Inset and RaisedInset convert to Panel1
    - BackColor converts to FillColor
    - BackStyle of Transparent converts to a FillColor of NoBrush
    - Blink is ignored
    - BorderColor, BorderStyle, BorderUsesBackColor, and BorderWidth are ignored
    - Height and Width convert to Height and Width
    - Left and Top convert to X and Y
    - PatternColor and PatternStyle are ignored
    - Visible converts to Visible
- **Ellipse**
    - Converts to Ellipse
    - BackColor converts to FillColor
    - BackStyle of Transparent converts to a FillColor of NoBrush
    - ForeColor converts to Border Color
    - Height and Width convert to Height and Width
    - Left and Top convert to X and Y
    - LineStyle converts to Border Style
    - LineWidth converts to Border Width
    - PatternColor and PatternStyle are ignored
    - Visible converts to Visible
- **Line**
    - Converts to LineElement
    - BackColor is ignored
    - BackStyle is ignored
    - ForeColor converts to Border Color
    - Height and Width convert to Height and Width
    - Left and Top convert to X and Y
    - LineStyle converts to Border Style

    - LineWidth converts to Border Width
    - Visible converts to Visible
- **Rectangle**
    - Converts to Rectangle
    - BackColor converts to FillColor
    - BackStyle if Transparent converts to a FillColor of NoBrush
    - ForeColor converts to Border Color
    - Height and Width convert to Height and Width
    - Left and Top convert to X and Y
    - LineStyle converts to Border Style
    - LineWidth converts to Border Width
    - PatternColor and PatternStyle are ignored
    - Visible converts to Visible
- **Rounded Rectangle**
    - Converts to Rectangle with a CornerRadius of 10
    - BackColor converts to FillColor
    - BackStyle if Transparent converts to a FillColor of NoBrush
    - ForeColor converts to Border Color
    - Height and Width convert to Height and Width
    - Left and Top convert to X and Y
    - LineStyle converts to Border Style
    - LineWidth converts to Border Width
    - PatternColor and PatternStyle are ignored
    - Visible converts to Visible
- **Momentary Button**
    - Converts to Button
        - If ButtonAction is Normally Open, then Button has Button Behavior of “Set a tag to 1 on press, 0 on release”
        - If ButtonAction is Normally Closed, then Button has Button Behavior of “Set a tag to 0 on press, 1 on release”
        - If ButtonAction is Value, then Button has a Touch Press Event with a Write command to write the State1 value when pressed and a Touch Release Event with a Write command to write the State0 value when released.
    - Audio is ignored
    - BackStyle, BorderStyle, BorderUsesBackColor are ignored
    - BorderWidth converts to Border Width
    - CaptionOnBorder is ignored
    - Height and Width convert to Height and Width
    - HighlightColor is ignored
    - HoldTime is converted to the Minimum Hold Time on the Button Behavior
    - HorizontalMargin and VerticalMargin are ignored
    - Left and Top convert to X and Y
    - NumberOfStates is ignored
    - Shape is ignored
    - Touch is ignored

    - Visible converts to Visible
    - For states, create a two-state state table for the button with the following properties where the states map to values of these properties:
        - St_BackColor converts to FillColor
        - St_Blink is ignored
        - St_BorderColor converts to Border Color
        - St_Caption converts to Text. If the Caption contains embedded variables, they are ignored. Caption Bold is mapped to Bold; italics and underlines are ignored.
        - ST_CaptionAlignment converts to TextAlignment
        - ST_CaptionBackColor, St_CaptionBackStyle, and St_CaptionBlink are ignored
        - St_CaptionColor converts to FontColor
        - St_CaptionFont converts to FontName. If View Designer does not support the font, use Arial Unicode MS.
        - St_CaptionWordWrap is ignored
        - St_Image* are all ignored
        - St_PatternColor and St_PatternStyle are ignored
    - The button's Value connection converts to the Tag on the Button Behavior.
- **Maintained Button**
    - Converts to Button with a state table with two states and state enter events with value write commands. Also a Touch Release event with a State Next command configured to "Wrap around":
        - The State Next command runs the button through the two states when it is pressed.
        - The State enter events write the two values to the controller.
    - AllowBlankComment is ignored
    - Audio is ignored
    - AuthorizedGroup, Domain*, Require* (And any other e-signature properties) are ignored
    - BackStyle, BorderStyle, BorderUsesBackColor are ignored
    - BorderWidth converts to Border Width
    - CaptionOnBorder is ignored
    - Height and Width convert to Height and Width
    - HighlightColor is ignored
    - HorizontalMargin and VerticalMargin are ignored
    - Left and Top convert to X and Y
    - NextStateBasedOn is ignored (the converted button will always work in "Based on current state" mode)
    - Shape is ignored
    - Touch is ignored
    - Visible converts to Visible
    - For states, create a two-state state table for the button with the following properties where the states map to values of these properties:
        - St_BackColor converts to FillColor
        - St_Blink is ignored
        - St_BorderColor converts to Border Color
        - St_Caption converts to Text. If the Caption contains embedded variables, they are ignored. Caption Bold is mapped to Bold; italics and underlines are ignored.
        - ST_CaptionAlignment converts to TextAlignment
        - ST_CaptionBackColor, St_CaptionBackStyle, and St_CaptionBlink are ignored

        - St_CaptionColor converts to FontColor
        - St_CaptionFont converts to FontName. If View Designer does not support the font, use Arial Unicode MS.
        - St_CaptionWordWrap is ignored
        - St_Image* are all ignored
        - St_PatternColor and St_PatternStyle are ignored
    - The button's Value connection converts to the Tag on the State Enter write commands.
- **Latched Button**
    - Converts to Button with a state table with two states and an expression of !(HandshakeTag). A Press Event writes the State1 value to the controller and advances the state table to State1 using a State Next command. When the HandshakeTag comes back true, it will set the state back to State0. Another State Enter event will see that and write the State0 value to the controller.
    - AllowBlankComment is ignored
    - Audio is ignored
    - AuthorizedGroup, Domain*, Require* (And any other e-signature properties) are ignored
    - BackStyle, BorderStyle, BorderUsesBackColor are ignored
    - BorderWidth converts to Border Width
    - CaptionOnBorder is ignored
    - Height and Width convert to Height and Width
    - HighlightColor is ignored
    - HorizontalMargin and VerticalMargin are ignored
    - LatchResetStyle is ignored (always unlatches when Handshake is non-zero)
    - Left and Top convert to X and Y
    - Shape is ignored
    - Touch is ignored
    - Visible converts to Visible
    - For states, create a two-state state table for the button with the following properties where the states map to values of these properties:
        - St_BackColor converts to FillColor
        - St_Blink is ignored
        - St_BorderColor converts to Border Color
        - St_Caption converts to Text. If the Caption contains embedded variables, they are ignored. Caption Bold is mapped to Bold; italics and underlines are ignored.
        - ST_CaptionAlignment converts to TextAlignment
        - ST_CaptionBackColor, St_CaptionBackStyle, and St_CaptionBlink are ignored
        - St_CaptionColor converts to FontColor
        - St_CaptionFont converts to FontName. If View Designer does not support the font, use Arial Unicode MS.
        - St_CaptionWordWrap is ignored

        - St_Image* are all ignored
        - St_PatternColor and St_PatternStyle are ignored
    - The button’s Value connection converts to the Tag on the Value write and State Enter write commands. The Handshake connection is used as in the state table expression.
- **Multistate Button**
    - Converts to Button with a state table with a number of states equal to NumberOfStates and state enter events with value write commands. Also a Touch Release event with a State Next command configured to “Wrap around”:
        - The State Next command runs the button through the multiple states when it is pressed.
        - The State enter events write the values to the controller.
    - AllowBlankComment is ignored
    - Audio is ignored
    - AuthorizedGroup, Domain*, Require* (And any other e-signature properties) are ignored
    - BackStyle, BorderStyle, BorderUsesBackColor are ignored
    - BorderWidth converts to Border Width
    - CaptionOnBorder is ignored
    - Height and Width convert to Height and Width
    - HighlightColor is ignored
    - HorizontalMargin and VerticalMargin are ignored
    - Left and Top convert to X and Y
    - NextStateBasedOn is ignored (the converted button will always work in “Based on current state” mode)
    - NumberOfStates is converted to the number of states defined in the Button state table
    - Shape is ignored
    - Touch is ignored
    - Visible converts to Visible
    - For states, create a state table for the button with the following properties where the states map to values of these properties:
        - St_BackColor converts to FillColor
        - St_Blink is ignored
        - St_BorderColor converts to Border Color
        - St_Caption converts to Text. If the Caption contains embedded variables, they are ignored. Caption Bold is mapped to Bold; italics and underlines are ignored.
        - St_CaptionAlignment converts to TextAlignment
        - St_CaptionBackColor, St_CaptionBackStyle, and St_CaptionBlink are ignored
        - St_CaptionColor converts to FontColor
        - St_CaptionFont converts to FontName. If View Designer does not support the font, use Arial Unicode MS.
        - St_CaptionWordWrap is ignored
        - St_Image* are all ignored
        - St_PatternColor and St_PatternStyle are ignored
    - The button’s Value connection converts to the Tag on the State Enter write commands.
- **Interlocked Button**
    - Converts to a RadioButton with ShowMark bound to an expression of the Value tag connection equal to the ButtonValue
    - AllowBlankComment is ignored

    - AuthorizedGroup, Domain*, Require* (And any other e-signature properties) are ignored
    - Audio is ignored
    - BackStyle, BorderStyle, BorderUsesBackColor are ignored
    - BorderWidth is ignored
    - CaptionOnBorder is ignored
    - Height and Width convert to Height and Width is set equal to Height
    - HighlightColor is ignored
    - HorizontalMargin and VerticalMargin are ignored
    - Left and Top convert to X and Y
    - Shape is ignored
    - Touch is ignored
    - Visible converts to Visible
    - The button's Value connection converts to the Tag used in the ShowMark expression.
- **Ramp Button**
    - Converts to Button with a Touch Release event and a Value Increment command
    - If UseVariableRamp is false, then the By value is set equal to the RampValue or RampValue*-1 depending on the value of RampType
    - AllowBlankComment is ignored
    - Audio is ignored
    - AuthorizedGroup, Domain*, Require* (And any other e-signature properties) are ignored
    - BackColor is converted to FillColor
    - If UseVariableRamp is true, then the By value is bound to the same tag used in the Ramp connection or that tag * -1 depending on the value of RampType.
    - BackStyle, BorderStyle, BorderUsesBackColor are ignored
    - Blink is ignored
    - BorderColor converts to Border Color
    - BorderWidth converts to Border Width
    - Caption converts to Text. If the Caption contains embedded variables, they are ignored. Caption Bold is mapped to Bold; italics and underlines are ignored.
    - CaptionAlignment converts to TextAlignment
    - CaptionBackColor, CaptionBackStyle, and CaptionBlink are ignored
    - CaptionColor converts to FontColor
    - CaptionFont converts to FontName. If View Designer does not support the font, use Arial Unicode MS.
    - CaptionOnBorder is ignored
    - CaptionWordWrap is ignored
    - Height and Width convert to Height and Width
    - HighlightColor is ignored
    - HorizontalMargin and VerticalMargin are ignored
    - Image* are all ignored
    - Left and Top convert to X and Y
    - LimitValue is used in the By expression with an IIF such that if the tag to increment is above the Limit, the increment By value is 0.
    - PatternColor and PatternStyle are ignored
    - RampValue is used in the By expression if UseVariableRamp is false
    - Shape is ignored

  - Touch is ignored
  - UseVariableLimit is used in the increment By expression to limit based on the tag on the Limit connection it True; otherwise, limit by the LimitValue
  - UseVariableRamp is used to select the value to increment in the increment By expression
  - Visible converts to Visible
- **Numeric Display**
  - Converts to NumericDisplay
  - Alignment converts to TextAlignment
  - BackColor converts to FillColor
  - BackStyle of Transparent converts to a FillColor of NoBrush
  - Blink is ignored
  - BorderColor converts to Border Color
  - BorderStyle is ignored
  - BorderUsesBackColor is ignored
  - BorderWidth converts to Border Width
  - DecimalPlaces converts to DigitsAfterDecimal
  - FillLeftWith – If "Zeroes", set LeadingZerosFill true; otherwise, set LeadingZerosFill false
  - Font converts to FontName. If View Designer does not support the font, use Arial Unicode MS. Bold is mapped to Bold; italics and underlines are ignored.
  - ForeColor converts to FontColor
  - Height and Width convert to Height and Width
  - Left and Top convert to X and Y
  - NumberOfDigits converts to DigitsBeforeDecimal such that DigitsBeforeDecimal = NumberOfDigits-DecimalPlaces-1
  - PatternColor and PatternStyle are ignored
  - Visible converts to Visible
  - The Value connection is mapped to the Value binding
  - The Polarity connection is ignored
- **Numeric Input Enable**
  - Converts to NumericInput
  - AllowBlankComment, AuthorizedGroup, Domain*, Require*, VariableDomainName, and any other E-Signature properties are ignored
  - Audio is ignored
  - BackColor converts to FillColor
  - BackStyle of Transparent converts to a FillColor of NoBrush
  - Blink is ignored
  - BorderColor converts to Border Color
  - BorderStyle is ignored
  - BorderUsesBackColor is ignored
  - BorderWidth converts to Border Width
  - Caption Alignment converts to TextAlignment
  - CaptionBackColor, CaptionBackStyle, and CaptionBlink are ignored
  - CaptionColor converts to FontColor
  - CaptionFont converts to FontName. If View Designer does not support the font, use Arial Unicode MS.
  - CaptionOnBorder and CaptionWordWrap are ignored

  - DecimalPoint is ignored (the NumericInput always uses a fixed position for display)
  - DigitsAfterDecimalPoint converts to DigitsAfterDecimal
  - EnterKey* are ignored
  - FixedPosition is ignored
  - HandshakeResetType is ignored
  - Height and Width convert to Height and Width
  - HighlightColor is ignored
  - HorizontalMargin and VerticalMargin are ignored
  - Image* are ignored
  - KeyNavigation is ignored
  - Left and Top convert to X and Y
  - NumericPopUp is ignored
  - PatternColor and PatternStyle are ignored
  - RampValue is ignored
  - Shape is ignored
  - TakeFocusOnPress is ignored
  - Touch is ignored
  - UseVariableMinMax is converted if true, bind KeypadMaxValue and KeypadMinValue to the Maximum and Minimum connections. If false, set KeypadMaxValue and KeypadMinValue equal to the maximum and minimum value for the Numeric Input Enable dialog.
  - Visible converts to Visible
  - The Value connection is mapped to the Value binding
  - The Optional Exp connection is ignored
  - The Enter and Enter Handshake connections are ignored
  - The Minimum and Maximum connections are used to configure KeypadMaxValue and KeypadMinValue if UseVariableMinMax is true
- **Numeric Input Cursor Point**
  - Converts to NumericInput
  - Alignment converts to TextAlignment
  - AllowBlankComment, AuthorizedGroup, Domain*, Require*, VariableDomainName, and any other E-Signature properties are ignored
  - Audio is ignored
  - BackColor converts to FillColor
  - BackStyle of Transparent converts to a FillColor of NoBrush
  - Blink is ignored
  - BorderColor converts to Border Color
  - BorderStyle is ignored
  - BorderUsesBackColor is ignored
  - BorderWidth converts to Border Width
  - CaptionOnPad is ignored
  - DecimalPlaces converts to DigitsAfterDecimal
  - DecimalPoint is ignored (the NumericInput always uses a fixed position for display)
  - DigitsAfterDecimalPoint is ignored
  - EnterKey* are ignored
  - FillLeftWith – If "Zeroes", set LeadingZerosFill true; otherwise, set LeadingZerosFill false

  - FixedPosition is ignored
  - Font converts to FontName. If View Designer does not support the font, use Arial Unicode MS. Bold is mapped to Bold; italics and underlines are ignored.
  - ForeColor converts to FontColor
  - HandshakeResetType is ignored
  - Height and Width convert to Height and Width
  - HighlightColor is ignored
  - HorizontalMargin and VerticalMargin are ignored
  - KeyNavigation is ignored
  - Left and Top convert to X and Y
  - NumberOfDigits converts to DigitsBeforeDecimal such that DigitsBeforeDecimal = NumberOfDigits-DecimalPlaces-1
  - NumericPopUp is ignored
  - PatternColor and PatternStyle are ignored
  - RampValue is ignored
  - Touch is ignored
  - UseVariableMinMax is converted if true, bind KeypadMaxValue and KeypadMinValue to the Maximum and Minimum connections. If false, set KeypadMaxValue and KeypadMinValue equal to the maximum and minimum value for the Numeric Input Enable dialog.
  - Visible converts to Visible
  - The Value connection is mapped to the Value binding
  - The Optional Exp connection is ignored
  - The Indicator connection is ignored
  - The Enter and Enter Handshake connections are ignored
  - The Minimum and Maximum connections are used to configure KeypadMaxValue and KeypadMinValue if UseVariableMinMax is true

- **String Display**
  - Converts to TextDisplay
  - Alignment converts to TextAlignment
  - BackColor converts to FillColor
  - BackStyle of Transparent converts to a FillColor of NoBrush
  - Blink is ignored
  - BorderColor converts to Border Color
  - BorderStyle is ignored
  - BorderUsesBackColor is ignored
  - BorderWidth converts to Border Width
  - Font converts to FontName. If View Designer does not support the font, use Arial Unicode MS. Bold is mapped to Bold; italics and underlines are ignored.
  - ForeColor converts to FontColor
  - Height and Width convert to Height and Width
  - Left and Top convert to X and Y
  - PatternColor and PatternStyle are ignored
  - Visible converts to Visible
  - The Value connection is mapped to the Text binding

- **String Input Enable**
    - Converts to TextInput
    - AllowBlankComment, AuthorizedGroup, Domain*, Require*, VariableDomainName, and any other E-Signature properties are ignored
    - Audio is ignored
    - BackColor converts to FillColor
    - BackStyle of Transparent converts to a FillColor of NoBrush
    - Blink is ignored
    - BorderColor converts to Border Color
    - BorderStyle is ignored
    - BorderUsesBackColor is ignored
    - BorderWidth converts to Border Width
    - Caption Alignment converts to TextAlignment
    - CaptionBackColor, CaptionBackStyle, and CaptionBlink are ignored
    - CaptionColor converts to FontColor
    - CaptionFont converts to FontName. If View Designer does not support the font, use Arial Unicode MS.
    - CaptionOnBorder and CaptionWordWrap are ignored
    - EnterKey* are ignored
    - FillCharacter is ignored
    - HandshakeResetType is ignored
    - Height and Width convert to Height and Width
    - HighlightColor is ignored
    - HorizontalMargin and VerticalMargin are ignored
    - Image* are ignored
    - KeyNavigation is ignored
    - Left and Top convert to X and Y
    - MaskScratchpad converts to MaskedValue
    - NumberOfInputCharacters is ignored
    - PatternColor and PatternStyle are ignored
    - Shape is ignored
    - StringPopUp is ignored
    - TakeFocusOnPress is ignored
    - Touch is ignored
    - Visible converts to Visible
    - The Value connection is mapped to the Text binding
    - The Enter and Enter Handshake connections are ignored
- **Goto Display**
    - Converts to Button with a ButtonBehavior of “Navigate to screen on release”.
    - Audio is ignored
    - BackColor converts to FillColor
    - BackStyle of Transparent converts to a FillColor of NoBrush
    - Blink is ignored
    - BorderColor converts to Border Color
    - BorderStyle is ignored
    - BorderUsesBackColor is ignored

  - BorderWidth converts to Border Width
  - Caption converts to Text
  - Caption Alignment converts to TextAlignment
  - CaptionBackColor, CaptionBackStyle, and CaptionBlink are ignored
  - CaptionColor converts to FontColor
  - CaptionFont converts to FontName. If View Designer does not support the font, use Arial Unicode MS.
  - CaptionOnBorder and CaptionWordWrap are ignored
  - Display will need to be re-entered on the Screen property of the Navigate to screen Button Behavior.
  - DisplayLeftPosition, DisplayPosition, and DisplayTop position are ignored
  - Height and Width convert to Height and Width
  - HighlightColor is ignored
  - HorizontalMargin and VerticalMargin are ignored
  - Image* are ignored
  - Left and Top convert to X and Y
  - ParameterFile, ParameterList, and ParameterType are ignored. The user will need to change how they do this by using custom screen properties.
  - PatternColor and PatternStyle are ignored
  - Shape is ignored
  - Touch is ignored
  - UseVariableDisplay and UseVariableDisplayPosition are ignored
  - Visible converts to Visible
  - The Connections are ignored

- **Return To Display**
  - Converts to Button with a ButtonBehavior of “Navigate backward on release”.
  - Audio is ignored
  - BackColor converts to FillColor
  - BackStyle of Transparent converts to a FillColor of NoBrush
  - Blink is ignored
  - BorderColor converts to Border Color
  - BorderStyle is ignored
  - BorderUsesBackColor is ignored
  - BorderWidth converts to Border Width
  - Caption converts to Text
  - Caption Alignment converts to TextAlignment
  - CaptionBackColor, CaptionBackStyle, and CaptionBlink are ignored
  - CaptionColor converts to FontColor
  - CaptionFont converts to FontName. If View Designer does not support the font, use Arial Unicode MS.
  - CaptionOnBorder and CaptionWordWrap are ignored
  - Height and Width convert to Height and Width
  - HighlightColor is ignored
  - HorizontalMargin and VerticalMargin are ignored
  - Image* are ignored
  - Left and Top convert to X and Y
  - PatternColor and PatternStyle are ignored
  - Shape is ignored

  - Touch is ignored
  - Visible converts to Visible
- **Close Display**
  - Converts to Button with a ButtonBehavior of “Close popup on release”.
  - Audio is ignored
  - BackColor converts to FillColor
  - BackStyle of Transparent converts to a FillColor of NoBrush
  - Blink is ignored
  - BorderColor converts to Border Color
  - BorderStyle is ignored
  - BorderUsesBackColor is ignored
  - BorderWidth converts to Border Width
  - Caption converts to Text
  - Caption Alignment converts to TextAlignment
  - CaptionBackColor, CaptionBackStyle, and CaptionBlink are ignored
  - CaptionColor converts to FontColor
  - CaptionFont converts to FontName. If View Designer does not support the font, use Arial Unicode MS.
  - CaptionWordWrap is ignored
  - CloseValue converts to the value written in a Value Write command if WriteOnClose is true
  - Height and Width convert to Height and Width
  - HighlightColor is ignored
  - HorizontalMargin and VerticalMargin are ignored
  - Image* are ignored
  - Left and Top convert to X and Y
  - PatternColor and PatternStyle are ignored
  - Shape is ignored
  - Touch is ignored
  - Visible converts to Visible
  - WriteOnClose if false is ignored; if true, converts to a Press Release event with a Value Write command writing the CloseValue to the tag defined in the Value connection
  - Value connection is used as the tag to write to if WriteOnClose is true.
- **Multistate Indicator**
  - Converts to a Text Display with a state table with a number of states equal to the NumberOfStates property of the Multistate Indicator
  - BackStyle of Transparent converts to FillColors of NoBrush
  - BorderStyle is ignored
  - BorderUsesBackColor is ignored
  - BorderWidth converts to Border Width
  - CaptionOnBorder is ignored
  - Height and Width convert to Height and Width
  - Left and Top convert to X and Y
  - NumberOfStates converts to the number of states in the state table for the Text Display
  - Shape is ignored
  - TriggerType is ignored (always use Value)
  - Visible converts to Visible

    - For states, create a state table for the Text Display with the following properties where the states map to values of these properties:
        - St_BackColor converts to FillColor
        - St_Blink is ignored
        - St_BorderColor converts to Border Color
        - St_Caption converts to Text. If the Caption contains embedded variables, they are ignored. Caption Bold is mapped to Bold; italics and underlines are ignored.
        - St_CaptionAlignment converts to TextAlignment
        - St_CaptionBackColor, St_CaptionBackStyle, and St_CaptionBlink are ignored
        - St_CaptionColor converts to FontColor
        - St_CaptionFont converts to FontName. If View Designer does not support the font, use Arial Unicode MS.
        - St_CaptionWordWrap is ignored
        - St_Image* are all ignored
        - St_PatternColor and St_PatternStyle are ignored
    - Indicator connection is used in the expression of the state table on the Text Display.
- **List Indicator**
    - Converts to a List Box Indicator. The Items in the List Box Indicator each will have a number in parenthesis matching the St_Value properties of the List Indicator.
    - BackColor converts to FillColor
    - BackStyle of Transparent converts to FillColors of NoBrush
    - BorderStyle is ignored
    - BorderUsesBackColor is ignored
    - BorderWidth is ignored
    - CaptionFont converts to FontName. If View Designer does not support the font, use Arial Unicode MS
    - CaptionTruncate is ignored
    - Height and Width convert to Height and Width
    - Left and Top convert to X and Y
    - NumberOfStates converts to the number of Items in the ListBoxIndicator
    - PatternColor and PatternStyle are ignored
    - SelectionBackColor converts to SelectionColor
    - SelectionForeColor is ignored
    - St_Caption converts to the item in the Items list of the ListBoxIndicator. If the Caption contains embedded variables, they are ignored. Caption Bold is mapped to Bold; italics and underlines are ignored.
    - St_CaptionAlignment is ignored
    - St_CaptionBackColor, St_CaptionBackStyle, and St_CaptionBlink are ignored
    - St_CaptionColor converts to FontColor. Use the first state’s CaptionColor as the FontColor for the entire ListBoxIndicator.
    - St_Value converts to the number in parenthesis after each Item in the ListBoxIndicator
    - TriggerType is ignored (always use Value)

  - Visible converts to Visible
  - Indicator connection is used in the expression of the Value property of the ListBoxIndicator.
- **Bar Graph**
  - Converts to a BarGraph.
  - BackColor converts to FillColor
  - BackStyle of Transparent converts to FillColor of NoBrush
  - BorderColor converts to Border Color
  - BorderStyle is ignored
  - BorderUsesBackColor is ignored
  - BorderWidth converts to Border Width
  - FillColor converts to LevelColor
  - FillDirection is ignored
  - FillStyle is ignored
  - Height and Width convert to Height and Width
  - Left and Top convert to X and Y
  - MaxValue converts to MaxValue
  - MinValue converts to MinValue
  - NumberOfThresholds if greater than 0 is converted into a color table on the BarGraph which changes the LevelColor based on ThresholdColor and ThresholdValue
  - ThresholdColor and ThresholdValue convert to values used in the definition of the color table for LevelColor if NumberOfThresholds is greater than 0.
  - ThresholdType is used in the definition of the color table for LevelColor if NumberOfThresholds is greater than 0. The values are either calculated in percent or as a direct value.
  - Visible converts to Visible
  - Value connection is used in the binding of the Value property of the BarGraph.
- **Gauge**
  - Convert to a Gauge1.
  - BackColor converts to FillColor
  - BackStyle of Transparent converts to FillColor of NoBrush
  - DigitsAfterDecimalPoint is ignored
  - FillColor is ignored
  - Font converts to FontName. If View Designer does not support the font, use Arial Unicode MS. Bold is mapped to Bold; italics and underlines are ignored.
  - ForeColor converts to TickMarkColor
  - Height and Width convert to Height and Width
  - Left and Top convert to X and Y
  - LegendColor converts to FontColor
  - LineStyle is ignored
  - MajorTicks is ignored
  - MaxValue converts to MaxValue
  - MinorTicks is ignored
  - MinValue converts to MinValue
  - NeedleColor converts to NeedleColor
  - NeedleWidth is ignored
  - NumberOfThresholds is ignored

- ShowLegend sets TextOpacity to 0 if false
- SweepStyle is ignored
- Threshold* are ignored
- Visible converts to Visible
- Value connection is used in the binding of the Value property of the Gauge1.

- **Trend**
  - Converts to a TrendChart
  - Height and Width convert to Height and Width
  - Left and Top convert to X and Y
  - Visible converts to Visible
  - Pen connections are used in the bindings of the Traces
  - All other Trend properties are ignored and will need to be configured in the converted View Designer trend
- **Login**
  - Converts to a Button with a Button Behavior of “Logon on release”
  - Audio is ignored
  - BackColor converts to FillColor
  - BackStyle of Transparent converts to a FillColor of NoBrush
  - Blink is ignored
  - BorderColor converts to Border Color
  - BorderStyle is ignored
  - BorderUsesBackColor is ignored
  - BorderWidth converts to Border Width
  - Caption converts to Text. If the Caption contains embedded variables, they are ignored. Caption Bold is mapped to Bold; italics and underlines are ignored.
  - Caption Alignment converts to TextAlignment
  - CaptionBackColor, CaptionBackStyle, and CaptionBlink are ignored
  - CaptionColor converts to FontColor
  - CaptionFont converts to FontName. If View Designer does not support the font, use Arial Unicode MS.
  - CaptionWordWrap is ignored
  - DomainName is ignored
  - Height and Width convert to Height and Width
  - HideUserNameEntry is ignored
  - HighlightColor is ignored
  - HorizontalMargin and VerticalMargin are ignored
  - Image* are ignored
  - Left and Top convert to X and Y
  - PatternColor and PatternStyle are ignored
  - Shape is ignored
  - Touch is ignored
  - UseVariableDomainName is ignored

    - Visible converts to Visible
    - DomainName connection is ignored.
- **Logout**
    - Converts to a Button with a Button Behavior of "Logoff on release"
    - Audio is ignored
    - BackColor converts to FillColor
    - BackStyle of Transparent converts to a FillColor of NoBrush
    - Blink is ignored
    - BorderColor converts to Border Color
    - BorderStyle is ignored
    - BorderUsesBackColor is ignored
    - BorderWidth converts to Border Width
    - Caption converts to Text. If the Caption contains embedded variables, they are ignored. Caption Bold is mapped to Bold; italics and underlines are ignored.
    - Caption Alignment converts to TextAlignment
    - CaptionBackColor, CaptionBackStyle, and CaptionBlink are ignored
    - CaptionColor converts to FontColor
    - CaptionFont converts to FontName. If View Designer does not support the font, use Arial Unicode MS.
    - CaptionWordWrap is ignored
    - Display is ignored
    - Height and Width convert to Height and Width
    - HighlightColor is ignored
    - HorizontalMargin and VerticalMargin are ignored
    - Image* are ignored
    - Left and Top convert to X and Y
    - Parameter* are ignored
    - PatternColor and PatternStyle are ignored
    - Shape is ignored
    - ShowDisplayOnLogout is ignored
    - Touch is ignored
    - Visible converts to Visible
- **Control List Selector**
    - Converts to a ListBoxSelector. The Items in the List Box Selector each will have a number in parenthesis matching the St_Value properties of the Control List Selector.
    - AllowBlankComment, AuthorizedGroup, Domain*, Require*, VariableDomainName, and any other E-Signature properties are ignored
    - BackColor converts to FillColor
    - BackStyle of Transparent converts to FillColors of NoBrush
    - Blink is ignored
    - BorderColor, BorderStyle, and BorderUsersBackColor are ignored
    - BorderWidth is ignored
    - CaptionFont converts to FontName. If View Designer does not support the font, use Arial Unicode MS
    - CaptionTruncate is ignored
    - EnterKey* are ignored
    - HandshakeResetType is ignored

  - Height and Width convert to Height and Width
  - KeyNavigation is ignored
  - Left and Top convert to X and Y
  - NumberOfStates converts to the number of Items in the ListBoxSelector
  - PatternColor and PatternStyle are ignored
  - SelectionBackColor converts to SelectionColor
  - SelectionForeColor is ignored
  - St_Caption converts to the item in the Items list of the ListBoxSelector. If the Caption contains embedded variables, they are ignored. Caption Bold is mapped to Bold; italics and underlines are ignored.
  - St_CaptionAlignment is ignored
  - St_CaptionBackColor, St_CaptionBackStyle, and St_CaptionBlink are ignored
  - St_CaptionColor converts to FontColor. Use the first state's CaptionColor as the FontColor for the entire ListBoxIndicator.
  - St_Value converts to the number in parenthesis after each Item in the ListBoxIndicator
  - Visible converts to Visible
  - WrapAround is ignored
  - WriteOnEnter is ignored
  - Value connection is used in the Value property of the ListBoxSelector
  - Enter and Enter Handshake connections are ignored

- **Language Switch Button**
  - Convert to Button with a ButtonBehavior of "Switch language on release".
  - Audio is ignored
  - BackColor converts to FillColor
  - BackStyle of Transparent converts to a FillColor of NoBrush
  - Blink is ignored
  - BorderColor converts to Border Color
  - BorderStyle is ignored
  - BorderUsesBackColor is ignored
  - BorderWidth converts to Border Width
  - Caption converts to Text. If the Caption contains embedded variables, they are ignored. Caption Bold is mapped to Bold; italics and underlines are ignored.
  - Caption Alignment converts to TextAlignment
  - CaptionBackColor, CaptionBackStyle, and CaptionBlink are ignored
  - CaptionColor converts to FontColor
  - CaptionFont converts to FontName. If View Designer does not support the font, use Arial Unicode MS.
  - CaptionWordWrap is ignored
  - Height and Width convert to Height and Width
  - HighlightColor is ignored
  - HorizontalMargin and VerticalMargin are ignored
  - Image* are ignored
  - Language converts to the "Switch language to" property of the Switch Language command.
  - Left and Top convert to X and Y
  - PatternColor and PatternStyle are ignored
  - Shape is ignored

  - Touch is ignored
  - Visible converts to Visible
- **Time and Date**
  - Converts to DateTimeDisplay.
  - Alignment converts to TextAlignment
  - BackColor converts to FillColor
  - BackStyle of Transparent converts to a FillColor of NoBrush
  - Blink is ignored
  - BorderColor converts to Border Color
  - BorderStyle is ignored
  - BorderUsesBackColor is ignored
  - BorderWidth converts to Border Width
  - Font converts to FontName. If View Designer does not support the font, use Arial Unicode MS.
  - ForeColor converts to FontColor
  - Format converts to Format. If no matching format, use default of DateTimeDisplay.
  - Height and Width convert to Height and Width
  - Left and Top convert to X and Y
  - PatternColor and PatternStyle are ignored
  - Visible converts to Visible
  - WordWrap is ignored
- **Clear Alarm History**
  - Converts to Button with a Touch Release event with a command of Clear Alarm History
  - AllowBlankComment, AuthorizedGroup, Domain*, Require*, VariableDomainName, and any other E-Signature properties are ignored
  - Audio is ignored
  - BackColor converts to FillColor
  - BackStyle of Transparent converts to a FillColor of NoBrush
  - Blink is ignored
  - BorderColor converts to Border Color
  - BorderStyle is ignored
  - BorderUsesBackColor is ignored
  - BorderWidth converts to Border Width
  - Caption converts to Text. If the Caption contains embedded variables, they are ignored. Caption Bold is mapped to Bold; italics and underlines are ignored.
  - Caption Alignment converts to TextAlignment
  - CaptionBackColor, CaptionBackStyle, and CaptionBlink are ignored
  - CaptionColor converts to FontColor
  - CaptionFont converts to FontName. If View Designer does not support the font, use Arial Unicode MS.
  - CaptionWordWrap is ignored
  - FilteredTriggers is ignored
  - Height and Width convert to Height and Width
  - HighlightColor is ignored
  - HorizontalMargin and VerticalMargin are ignored
  - Image* are ignored
  - Left and Top convert to X and Y

  - PatternColor and PatternStyle are ignored
  - SendPressTo is ignored
  - Shape is ignored
  - Touch is ignored
  - Visible converts to Visible
- **Alarm List**
  - Converts to Alarm Summary Medium if the screen size width is less than 800; converts to Alarm Summary Large otherwise. Configure the desired Alarm Summary columns after conversion.
  - AckSymbol is ignored
  - Active* is ignored
  - BackColor and BackStyle are ignored
  - BorderColor and BorderStyle are ignored
  - BorderUsesBackColor is ignored
  - BorderWidth is ignored
  - Display* are ignored
  - FilteredTriggers is ignored
  - Font converts to FontName. If View Designer does not support the font, use Arial Unicode MS.
  - FormatAckTime converts to DateTimeFormat. If the format does not exist, convert to the default.
  - FormatAlarmTime is ignored. Use the FormatAckTime.
  - Header* is ignored
  - Height and Width convert to Height and Width
  - Inactive* is ignored
  - KeyNavigation is ignored
  - Left and Top convert to X and Y
  - LinesPerAlarm is ignored
  - SelectedAlarmIndicator is ignored
  - SelectionBackColor and SelectionForeColor are ignored
  - UseAlarmIdentifier is ignored
  - Visible converts to Visible
  - WordWrap and WrapAround are ignored
  - All the Connections are ignored
- **Alarm Status List**
  - Based on ListMode converts to:
    - All Alarms – converts to Alarm Manager
    - Active Alarms – converts to Alarm Summary
    - Past Alarms – converts to Alarm History
    - Converts to medium sized alarm tables for screen widths less than 800, and the large sized alarm tables otherwise. Configure the desired columns on the alarm table after conversion.
  - BackColor and BackStyle are ignored
  - BorderColor and BorderStyle are ignored
  - BorderUsesBackColor is ignored
  - BorderWidth is ignored
  - ColumnWidth is ignored. Use the defaults.
  - Display* are ignored
  - FilteredTriggers is ignored

    - Font converts to FontName. If View Designer does not support the font, use Arial Unicode MS.
    - ForeColor is ignored
    - Header* is ignored
    - Height and Width convert to Height and Width
    - KeyNavigation is ignored
    - Left and Top convert to X and Y
    - LinesPerAlarm is ignored
    - ListMode converts to the use of the Alarm Manager, Alarm Summary, or Alarm History
    - ModeTitle* are ignored
    - SelectionBackColor and SelectionForeColor are ignored
    - UseAlarmColors is ignored
    - UseAlarmIdentifier is ignored
    - Visible converts to Visible
    - WordWrap and WrapAround are ignored
    - AlarmIdentifier Connection is ignored

View ME objects which do not convert:

- Groups. The objects within a group convert, but the group itself will need to be recreated in View Designer.
- Arc
- Freehand
- Polygon
- Polyline
- Wedge
- Display List Selector
- Symbol Indicator
- Scale
- Pause Pen Button
- Next Pen Button
- RecipePlus objects
- Key objects
- User Management objects (except for Login and Logout)
- Piloted Control List Selector
- Display Print
- Local Message Display
- Macro Button
- Shutdown
- Goto Configure Mode Button
- Acknowledge Alarm
- Acknowledge All
- Alarm Status Mode
- Clear Alarm Banner
- Print Alarm History
- Print Alarm Status
- Reset Alarm Status
- Silence
- Sort Alarms

- Alarm Banner
- Clear Diagnostics
- Diagnostics Clear All
- Diagnostics List
- Clear Audit Trail
- Audit Trail List
- Audit Trail Detail
- Acknowledge Info
- Message Display
- ActiveX Object
- Symbol Factory objects only convert if they are copy/pasted as an image in View ME

Rockwell Automation Support

Use these resources to access support information.

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

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.

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.

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