ObVerse Manual:
Advanced Processor

ObVerse is North's strategy language and is available within Commander and ObSys. It allows an engineer to define how equipment should link and interoperate.

ObVerse is available within ObSys in two areas: ObvProcessor and ObView.

This manual covers the extra features available within ObView – the advanced processor.

This document relates to the advanced processor within ObView v1.1

Please read the *ObVerse Manual: Standard Processor*, and *ObSys Tutorial* alongside this document, available from *www.northbt.com*

Contents

What is ObVerse?

Sometimes you need to do more than simply transfer a value from one object to another – you need to calculate something, delay something, or perform a more complex function on one or more values. North provides this flexibility with ObVerse, a cause-and-effect programming language.

ObVerse consists of a range of modules. The engineer selects particular modules and links them together to perform a desired strategy.
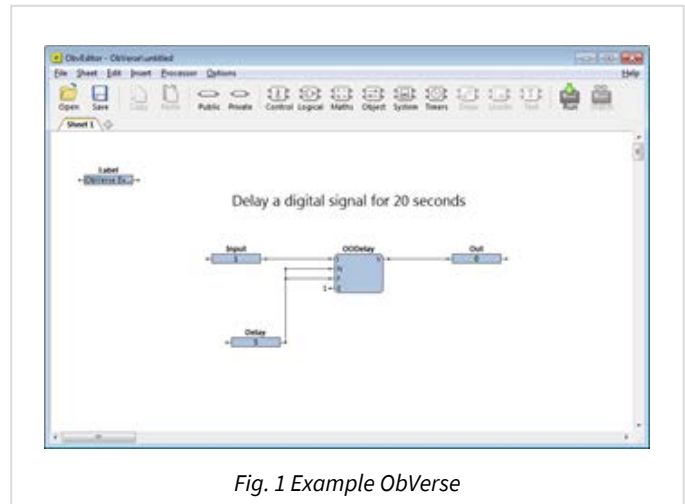

Fig. 1 Example ObVerse

Processors

ObVerse strategy runs in an ObVerse processor within a device.

ObVerse processors come in two types:

- **Standard Processor** – with logic, maths, and control modules
- **Advanced Processor** – with the same features found in a standard processor, plus extended maths and logic, display, application execution, directory and file services, and user-defined modules.

Both Commander and ObServer (part of ObSys) contain several ObvProcessors, which are designed to run in a controller, and so support the standard features of ObVerse, including logic, maths, and timers.

ObView (part of ObSys) is designed to run alongside a PC screen, and so supports both the standard and advanced features of ObVerse, including drawing items on screen, user input from a screen, text manipulation, and engineer-defined modules.

The document ObVerse Manual – Standard Processor covers the standard features in both ObvProcessor and ObView.

This document ObVerse Manual - Advanced Processor covers the advanced features found only in ObView.

Properties, Modules, and Comments

ObVerse strategy consists of properties, modules and comments.

ObVerse properties are containers for storing data values and carry a value from one module to another or between the processor and other tasks in the system.

ObVerse modules perform an operation on one or more input values and calculate one or more output values. The module's inputs can be linked to properties to use their values, and the module's outputs can be linked to properties to store the calculated values.

Comments in ObVerse are short pieces of text used to explain ObVerse strategy and make it easier for others to understand.

Editing ObVerse

You can create and edit ObVerse strategy using North's ObvEditor application, installed as part of the ObSys software. ObvEditor provides drag-and-drop graphical editing of ObVerse, uploading and downloading of ObVerse strategy, and run-time monitoring of the strategy within the processor.

You can also create and edit ObVerse strategy using a text editor.

Quick Start

If necessary, install North Engineering software onto your PC. It is available from *www.northbt.com*

Edit ObVerse in a Processor

🖥 To edit ObVerse for a standard processor, follow these steps:

→ Run the **Start Engineering** application

→ If you are using a processor in the local PC, navigate to **ObServer**. If you are using a processor in Commander, navigate to **North IP Devices**, and select the required Commander.

→ Navigate to **Configuration**, and then select an **ObVerse Processor**

→ Select **ObVerse** to start the ObvEditor application.

When ObvEditor application starts, it communicates with the associated processor, to determine its features, maximums, etc.

🖥 To edit ObVerse for an advanced processor, follow these steps:

→ Locate ObvEditor.exe within the Programs folder of the operating system

→ Double-click to start the application

When ObvEditor application starts, it has no associated processor, so allows all modules to be inserted.

For more information on using ObvEditor, refer to the section 'Introduction to ObVerse Programming' in the *ObSys Tutorial Part 1*.

Processors

A processor runs ObVerse. There are two types of ObVerse processor: one supports a standard set of properties and modules, and the other supports a more advanced set.

Standard Processors

ObVerse standard processors perform essential control for a distributed system – and support modules for object reading and writing, maths, logic, and timing, as well as more advanced energy optimisations.

Commander has two ObVerse standard processors. Whenever Commander is powered-on, these processors run their ObVerse strategy continuously.

ObServer, the core of ObSys, has four ObVerse standard processors. When ObServer is running, these processors run their ObVerse strategy continuously.

The engineer can use the ObvEditor application to create and edit ObVerse strategy.

Each standard processor allows 1000 items within its strategy. These items are any combination of modules, properties and comments.

The engineer can use the ObvEditor application save the strategy as a text file, and to put the strategy into the processor.

Advanced Processors

ObVerse advanced processors provide all the functions of a standard processor, as well as extra maths, logic, drawing, and user input modules. An advanced processor also supports custom modules, which the engineer creates and uses as required.

ObSys includes an application called ObView, several copies of which can be run simultaneously. Each copy of ObView is an advanced processor.

Each copy of ObView can run the same or different ObVerse strategy file when it is started – this could be when Windows starts, when some ObVerse decides, or when the user decides.

The engineer may use ObView or the ObvEditor to create the ObVerse strategy files.

The ObView application, when it is started, is told which ObVerse text file to load and execute.

Properties

ObVerse properties are containers for storing data values. They can carry a value from one module to another, or between the processor and other tasks within the system – similar to a wire in an electronic circuit.

Properties have a data type, to define the type of value they hold – like a number or a text string. The range of types supported depends on the processor. For a complete list of the types supported by a standard processor, refer to the *Property Types in Standard Processors* section below. Properties sometimes hold values passing only between modules in the same processor. In ObVerse, we call these private properties, as their value is private to the processor.

Properties sometimes hold values passing between the processor an external task within the system. A task could be Essential Data, Data Transfer, Time Control, user action, or another ObVerse Processor; either in the same North device or another. In ObVerse, we call these public properties, as their value is publicly available.

A property may be assigned an initial value to use when the processor runs for the first time.

Private Properties

A private property holds a value as it passes between modules. One module assigns a value to it, and one or more modules use this value (Fig. 2). The property's value is not available outside the processor.

Fig. 2 Private property

A private property's reference must start with a lowercase character. The editor usually assigns the reference automatically, as the character 'p' followed by a number.

The engineer should link a module output to a property's left-side connector: the module will then store its output in that property.

The engineer should link a module input to a property's right-side connector: the module will then use the value stored in the property for its input, when it is performing its operation.
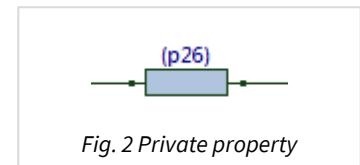
Public Properties

A public property holds a value that both modules and external tasks can access.

Fig. 3 Public property

A public property must have a reference that starts with an uppercase character. This reference also becomes the object reference used by external tasks to access the value from the ObVerse processor.

External tasks can read the value of a public property, and if adjustable, write the value. A property becomes adjustable by an external task when a module is not assigning a value to it within ObVerse.

The engineer may link a module output to the property's left-side connector: the module will then store its output in that property.

If the left-side connector is not linked to a module, the property becomes adjustable by an external task, and would hold its initial value until a task wrote a new value into it.

The engineer may link a module input to the property's right-side connector, the property will then provide the input value to the module when it is performing its operation.

When inserting a public property, in addition to an initial value and reference, other parameters may be included to help describe the property to an external task. These vary depending on the data type, but could include a label, high and low value limits, read rates, etc.
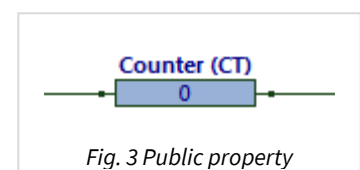
In the example public property shown above (Fig. 3), the reference has been set to 'CT', the label has been set to 'Counter', and its initial value is '0'. Because both left and right-hand sides are linked (but not fully shown), a module to the left will write a value into the property (overwriting the initial value), and a module (or more) to the right will use the value.

For more information on accessing public properties from an external task, refer to the ObVerse object within the *Commander Manual* document.

Type Conversion

If a module assigns a value of one type into a property of a different type, the module automatically converts the value to the type required by the property. Similarly, if a module input reads a value from a property of another type, the module will automatically convert the value to that required by the module input.

Reserved Public Properties

The processor reserves some property references within ObVerse for special purposes.

These properties are used by the ObVerse processor (ObView) as a way to integrate with other tasks, and the platform itself. ObView is a Windows application, and supports visible items in ObVerse

Description	Reference	Type
Label Label for ObVerse. If this property is present, the processor returns its value when scanned by the ObView engineering tool. The value is also used as the System field for alarms sent by the Alarm module.	L	Obj\Text: 31 chars; Adjustable
Remote Object Prefix Prefix to add in the front of the object reference used by ObjRead and ObjWrite modules. If a module has its absolute option set, then this prefix is not used.	O	Obj\Obj; Adjustable
Alarm Object If present, the Alarm module routes alarms to the object reference specified. Typically, this object reference ends '.ALARM'. If this property is not present, then the Alarm module routes alarms to the North device's ALARM object.	AO	Obj\Obj; Adjustable
Grid Size Size of grid onto which items are placed, snapping to the grid	G	Obj\Num; Adjustable Default value: 0 (no grid)
Edit Enable Enables the editing and saving of the ObVerse by the user from the view itself	EE	Obj\NoYes; Adjustable Default value: No
ObServer Channel The number of the channel to connect to ObServer	OC	Obj\Num; Adjustable Default value: 100
Close Inhibit Set to disable the 'close' options from the window itself	CI	Obj\NoYes; Adjustable Default value: No
Min/Max Inhibit Set to disable the minimize and maximize options from the window itself	MI	Obj\NoYes; Adjustable; Default value: No
Idle Rate (ms) Set to the desired idle rate for the ObVerse	I	Obj\Num; Adjustable; Default value: 50ms
Finish View	FV	Obj\Text: 127 chars; Adjustable
Window Style Set the type of window to appear with the ObVerse	WS	Obj\ENum; Adjustable Where: 0=Classic, 1=Maximized, 2=Minimized, 3=Invisible, 4=Topmost Classic, 5=Dialog, 6=Fullscreen, 7=Topmost Fullscreen, 8=Menuless Classic
Window X-coordinate Position of top-left corner of window on the screen	WX	Obj\Num; Adjustable Default: depends on other ObVerse running
Window Y-coordinate Position of top-left corner of window on the screen	WY	Obj\Num; Adjustable; Default: depends on other ObVerse running

Description	Reference	Type
Window Width Width of window on screen. If zero, window width will automatically size to show all items within the ObVerse	WW	Obj\Num; Adjustable; Default: tries to show all visible items, but may display scrollbars
Window Height Height of window on screen. If zero, window height will automatically size to show all items within the ObVerse	WH	Obj\Num; Adjustable; Default: tries to show all visible items, but may display scrollbars

Modules

Modules calculate values. They take one or more inputs and calculate one or more outputs.

Different modules are available to perform different operations. The range of modules supported depends on the processor. For a complete list of the modules supported by a standard processor, refer to the *Module Types* section below.

Fig. 4 Module with inputs and outputs
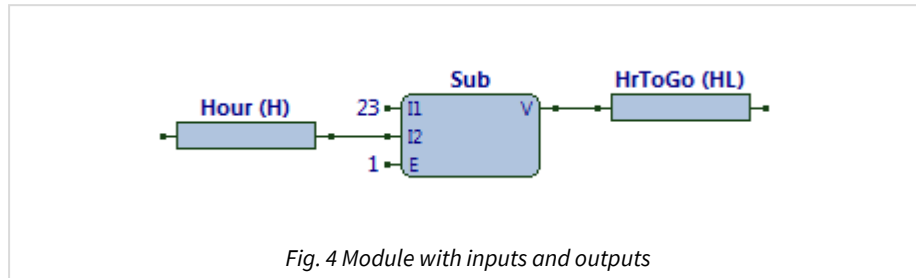
Inputs

Each module has one or more inputs. The engineer can leave the input set to its initial value; set its value to a constant; link it to a property (perhaps from the output of another module); or link it to an Essential Value.

The example strategy above (Fig. 4) shows a subtract module (Sub) with its three inputs – I1, I2, and E. Input I1 has been set to the constant value '23', input I2 has been linked to property H and uses the property's value, and input E has been set to the constant value '1' (the default value).

The engineer may link several module inputs to a property, if those modules all need to use its value.

Outputs

Each module has one or more outputs. The engineer can ignore an output if it is not needed; connect it to a property (so it can be used as input to another module); or connect it to an Essential Value.

The module will update the property or Essential Value whenever it calculates a new value.

The example strategy above (Fig. 4) shows a subtract module (Sub) with its one output – V. Output V has been linked to property HL, so stores its output value in that property.

Only one module can write to a particular property, otherwise it becomes difficult to understand which value the property holds at any time.

Comments

A comment is a short piece of text added to help understand the design of a piece of strategy (Fig. 5). They are optional, but we recommend they are used – it is surprising how quickly we all forget how our strategy works (or should work).

In ObvEditor, a comment can be set as a title, and shown in large text.

You can also use the different sheets, along with a descriptive title, to organise your ObVerse. Public properties should be labelled too, as this will also aid understanding.

Fig. 5 Comments
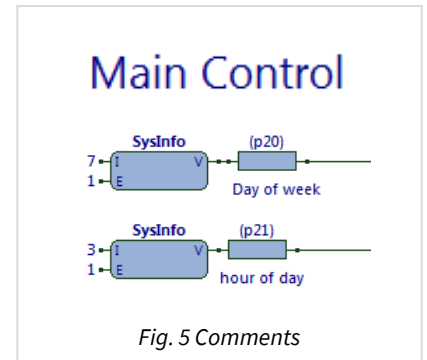
Property Types in Advanced Processors

A property within a processor must have a unique reference and a type. All ObVerse processors support the following types of properties:

ENum
Float
NoYes
Num
Obj
OffOn
Text

Refer to the *ObVerse Manual: Standard Processor* document for more information.

The ObVerse advanced processor additionally supports the following types of properties:

WinClr
WinFont
DateTime
Times
File

Other value types can be handled by using a Text property.

WinClr

Object Type: *[Obj\WinClr]*

A WinClr object holds a colour value, as a numeric value. Available only in Advanced processors.

Description	Reference	Type
Label Label of WinClr object	L	Obj\Text: 20 chars
Initial Value Value after ObVerse download	IV	Obj\Num
Write Inhibit Whether the value can be adjusted by external tasks	WI	Obj\ENum: 0...1; Where: 0=Writable; 1=Write Inhibited

WinFont

Object Type: *[Obj\WinFont]*

A WinFont object holds a font name and size as a text value. Available only in Advanced processors.

The text has the format <font> <size>, where <font> is the name of a font, and size is either the point size, or an asterisk to mean the size of font is to be chosen when an item is drawn.

Description	Reference	Type
Label Label of WinFont object	L	Obj\Text: 20 chars
Initial Value Value after ObVerse download	IV	Obj\Text
Write Inhibit Whether the value can be adjusted by external tasks	WI	Obj\ENum: 0...1; Where: 0=Writable; 1=Write Inhibited

DateTime

Object Type: *[Obj\DateTime]*

A DateTime object holds a date and time, and represent a point-in-time. Available only in Advanced processors.

Description	Reference	Type
Label Label of DateTime object	L	Obj\Text: 20 chars
Initial Value Value after ObVerse download	IV	Obj\Text
Write Inhibit Whether the value can be adjusted by external tasks	WI	Obj\ENum: 0...1; Where: 0=Writable; 1=Write Inhibited

Times

Object Type: *[Obj\Times]*

A Times object holds a series of start-stop times, particular to a single day. Available only in Advanced processors.

Description	Reference	Type
Label Label of Times object	L	Obj\Text: 20 chars
Periods Max number of periods	P	Obj\Num: 1...5
Initial Value Value after ObVerse download	IV	Obj\Text
Write Inhibit Whether the value can be adjusted by external tasks	WI	Obj\ENum: 0...1; Where: 0=Writable; 1=Write Inhibited

File

Object Type: *[Obj\File]*

A File object holds a local file name, including any drive and folder name, sufficient to identify a file. Available only in Advanced processors.

Description	Reference	Type
Label Label of File object	L	Obj\Text: 20 chars
Initial Value Value after ObVerse download	IV	Obj\Text
Write Inhibit Whether the value can be adjusted by external tasks	WI	Obj\ENum: 0...1; Where: 0=Writable; 1=Write Inhibited

Module Types

The ObVerse advanced processor supports the following modules:

Maths

Binary-And
Binary-Inverse
Binary-Or
Binary-Exclusive-Or

User Input

Double-Click
Press

Logic

Greater-or-Equal
Less-or-Equal
Not-Equal

System

Convert-Time
Object-List
Run-ObVerse
Security
Sound
System-Information
View-List
Windows-Execute

Draw

Alarm-Text-Value
Bar
BMP-File
Border
Enumerated-Value
Icon-File
IP-Address-Value
Menu-Item
Numeric-Value
Obj-Value
Rectangle
Static-Text
Text-Value
Times-Value
Trace
Video-Overlay
Variable-Line
WinClr-Value
WMF-File

Alarm-Text-Value

Object Type: *[Obv\AlarmText]*

The Alarm-Text-Value module displays, and optionally allows changes to, an alarm text value.

The value is drawn within a specified rectangle. If not inhibited, the user can click on the value, at which point a dialog window appears to allow them to select a new value. The new value is available for use as necessary.
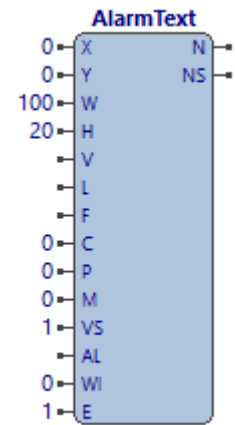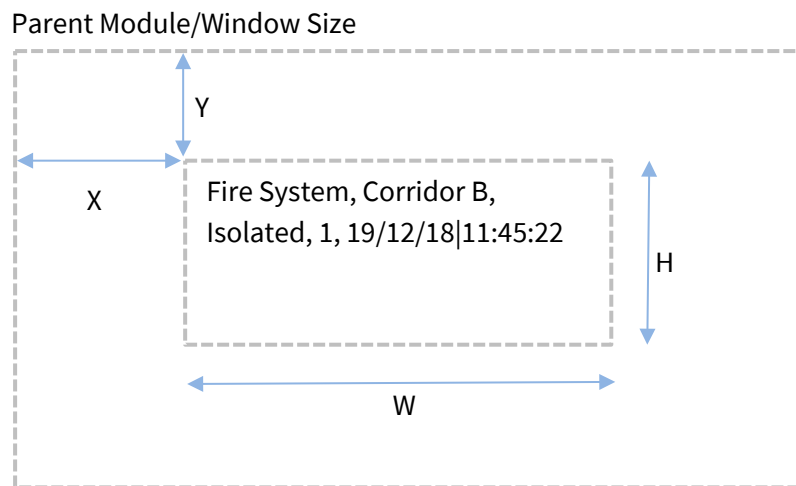

Fig. 6 AlarmText Module

The module contains the following sub-objects:

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner rectangle within which the alarm value is drawn	X	Obj\Num; Adjustable; Default value: 0
Y-Coordinate Y-coordinate of top-left corner of rectangle within which the alarm value is drawn	Y	Obj\Num; Adjustable; Default value: 0
Width Width of rectangle within which the alarm value is drawn	W	Obj\Num; Adjustable; Default value: 100
Height Height of rectangle within which the alarm text is drawn	H	Obj\Num; Adjustable; Default value: 20
Value The current value of the alarm	V	Obj\Alarm; Max chars: 127; Adjustable; Default value: ''
Label The label of the text value – shown on editing window, and within tooltip text	L	Obj\Text; Max chars: 31; Adjustable; Default value: ''
Font The font to display the text value using	F	Obj\WinFont; Adjustable; Default value: ''
Colour The colour of the text	C	Obj\WinClr; Adjustable; Default value: 0 (Black)
Position The position of the text within the rectangle	P	Obj\Num; Adjustable; Default value: 0; Where 0=LeftMiddle, 1=LeftBottom, 2=CentreBottom, 3=RightBottom, 4=LeftMiddle, 5=CentreMiddle, 6=RightomMiddle, 7=LeftTop, 8=CentreTop, 9=RightTop
Multiline Whether the alarm text is to be wrapped at width of rectangle, so taking several lines, or whether the alarm is to be truncated on a single line. Set to 1 for wrap-around multiline format.	M	Obj\NoYes; Adjustable; Default value: ''
Value Set Flag to indicate whether there is value to display – if not set, the box displays asterisks to indicate 'no value'	VS	Obj\NoYes; Adjustable; Default value: 1
Action Label Prepended to the Label to create the tooltip.	AL	Obj\Text; Max Chars:64; Adjustable; Default Value: 'Edit'

Description	Reference	Type
Write Inhibit Flag to inhibit editing of a new value. Set to 1 to inhibit	WI	Obj\NoYes; Adjustable; Default value: 0
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value: 1
New Value Set to the new value after the user has changed the value	N	Obj\Alarm
New Value Set Flag indicating a New Value is available	NS	Obj\NoYes

Screen Appearance

Parent Module/Window Size

Fire System, Corridor B,
Isolated, 1, 19/12/18|11:45:22

The example shows an alarm 'Fire System|Corridor B|Isolated|1|19/12/18|11:45:22' shown in a rectangle with Multiline set to 'Yes'

Related Modules

Enumerated-Value, *IP-Address-Value*, *Numeric-Value*, *Text-Value*, *WinClr-Value*, *Times-Value*

Bar

Object Type: *[Obv\Bar]*

The Bar module draws a dynamic bar, either horizontally or vertically. This could be used to represent a tank fill level, or a thermometer temperature level.
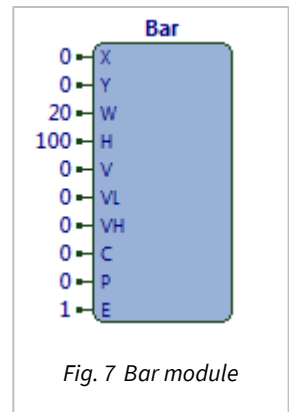

Fig. 7 Bar module

It contains the following sub-objects:

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner of rectangle into which the item is drawn	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle into which the item is drawn	Y	Obj\Num; Adjustable; Default value:0
Width Width of rectangle into which the item is drawn	W	Obj\Num; Adjustable; Default value:20
Height Height of rectangle into which the item is drawn	H	Obj\Num; Adjustable; Default value:100
Value The current value to represent using the bar. The position of the dynamic side of the bar will be proportional to the Value's position between the Value High and Low Limits	V	Obj\Float; Adjustable; Default value: 0
Value High Limit The value that causes the bar to be 'full'	VH	Obj\Float; Adjustable; Default value: 0
Value Low Limit The value that causes the bar to be 'empty'	VL	Obj\Float; Adjustable; Default value: 0
Colour The colour of the bar	C	Obj\WinClr; Adjustable; Default value: 0 (Black)
Position The position/direction that the bar dynamically fills	P	Obj\ENum; Adjustable; Range 0…3; where 0=FromBottom, 1=FromLeft, 2=FromTop, 3=FromRight
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1

Screen Appearance

Parent Module/Window Size
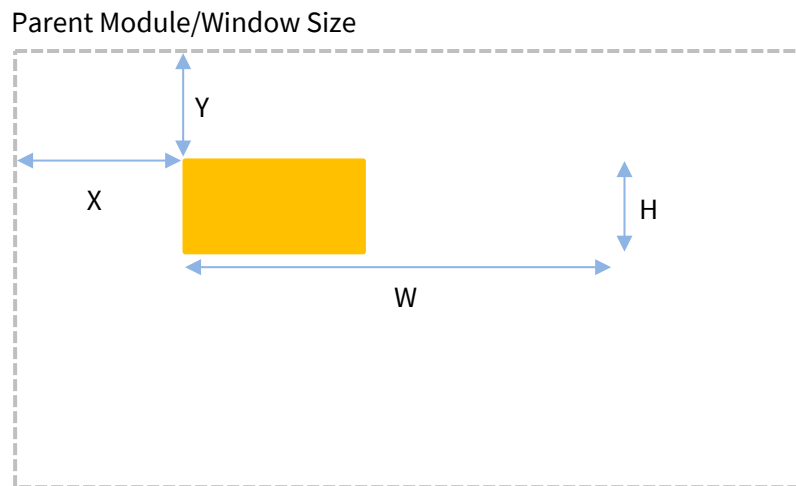
The example shows a **Bar** with a **Value Low Limit** set to '0', a **Value High Limit** set to '100', a **Value** set to '40', **Colour** set to orange, and **Position** set to 'FromLeft' – the rectangle is filled 40/100 from the left side.
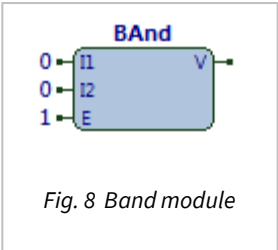
Related Modules

Trace, *Variable-Line*

Binary-And

Object Type: *[Obv\BAnd]*

The Binary-And module 'ands' together two numbers. The numbers are converted to 16-bit binary values, the corresponding bits are 'and'ed, and the result is converted back to a number.
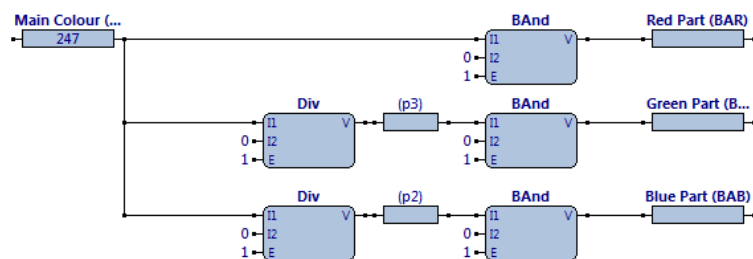

Fig. 8 Band module

Input 1 Number	12803
Input 2 Number	567
Input 1 Binary	0011001000000011
Input 2 Binary	0000001000110111
Bit-wise And	0000001000000011
Result as number	515

It contains the following sub-objects:

Description	Reference	Type
Input *x* Input x to And together, where *x* is in the range 1...2	I*x*	Obj\Float; Adjustable; Default value:0
Enable If set to 0, no calculation happens and V is left unchanged	E	Obj\NoYes; Adjustable; Default value:1
Value Set to the result of the calculation	V	Obj\Float

Example

This strategy determines the different colour components of a Main colour, by dividing and and'ing.

Related Modules

Binary-Or, *Binary-XOr*, *Binary-Inverse*

Binary-Exclusive-Or

Object Type: *[Obv\BXOr]*

The Binary-Exclusive-Or module 'xor's together two numbers. The numbers are converted to integers, then to 16-bit binary values, the corresponding bits are 'xor'ed, and the result is converted back to a number.
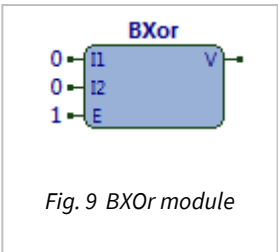
Fig. 9 BXOr module

Input 1	Number	12803
Input 2	Number	567
Input 1	Binary	0011001000000011
Input 2	Binary	0000001000110111
Bit-wise Exclusive Or		0011000000110100
Result as number		12340

It contains the following sub-objects:

Description	Reference	Type
Input *x* Input x to Exclusive-Or together, where *x* is in the range 1…2	I*x*	Obj\Float; Adjustable; Default value:0
Enable If set to 0, no calculation happens and V is left unchanged	E	Obj\NoYes; Adjustable; Default value:1
Value Set to the result of the calculation	V	Obj\Float

Related Modules

Binary-Or, *Binary-And*, *Binary-Inverse*

Binary-Inverse

Object Type: *[Obv\BInv]*

Fig. 10 BInv module

The Binary-Inverse module inverts a number. The number is converted to an integer, as 16-bit binary value, the corresponding bits are inverted, and the result is converted back to a number.

Input 1 number	12803
Input 1 as binary	0011001000000011
Bit-wise Inverse	1100110111111100
Result as number	52732

It contains the following sub-objects:

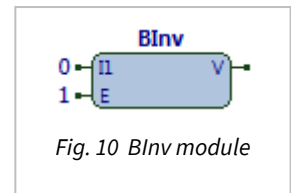| Description	Reference	Type
Input 1 Input 1 to invert	I*1*	Obj\Float; Adjustable; Default value:0
Enable If set to 0, no calculation happens and V is left unchanged	E	Obj\NoYes; Adjustable; Default value:1
Value Set to the result of the calculation	V	Obj\Float

Related Modules

Binary-Or, *Binary-XOr*, *Binary-And*,

Binary-Or

Object Type: *[Obv\BOr]*

Fig. 11 BOr module

The Binary-Or module 'or's together two numbers. The numbers are converted to integers, then to 16-bit binary values, the corresponding bits are 'or'ed, and the result is converted back to a number.

Input 1	Number	12803
Input 2	Number	567
Input 1	Binary	0011001000000011
Input 2	Binary	0000001000110111
Bit-wise Or		0011001000110111
Result as number		12855

It contains the following sub-objects:

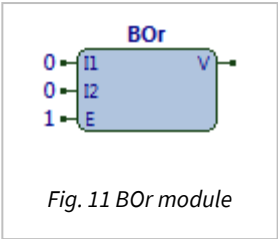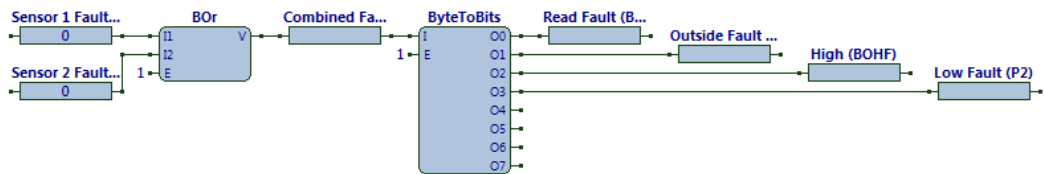| Description	Reference	Type
Input *x* Input x to And together, where *x* is in the range 1..2	I*x*	Obj\Float; Adjustable; Default value:0
Enable If set to 0, no calculation happens and V is left unchanged	E	Obj\NoYes; Adjustable; Default value:1
Value Set to the result of the calculation	V	Obj\Float

Example

This strategy combines fault-bytes from two sensors, before splitting the result to analyse individual bits.

Related Modules

Binary-And, *Binary-XOr*, *Binary-Inverse*

BMP-File

Object Type: *[Obv\BMPFile]*

The BMP-File module draws a .BMP file. Bitmap (BMP) files are the fundamental bitmap files within Windows and are supported by popular Windows-based creation/editing applications.

Fig. 12 BMPFile module

It contains the following sub-objects:

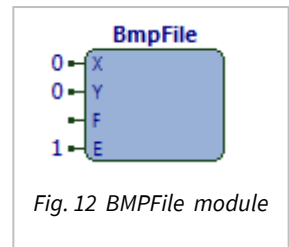| Description	Reference	Type
X-Coordinate X-coordinate of top-left corner of rectangle into which the item is drawn	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle into which the item is drawn	Y	Obj\Num; Adjustable; Default value:0
File The file name of the .BMP file to drawn	F	Obj\File; Adjustable
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1

Screen Appearance

Parent Module/Window Size
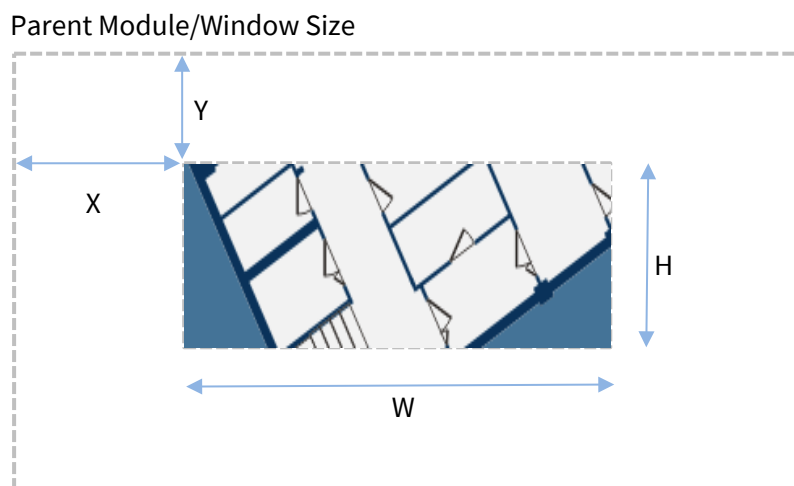
The example shows a .BMP file drawn on the screen. The BMP file was created using Microsoft Paint.
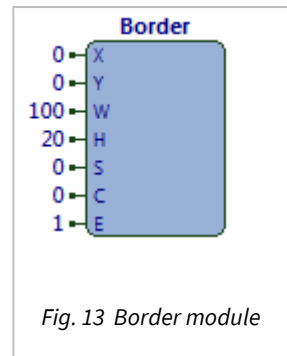
Related Modules

Windows-Meta-File, *Icon-File*
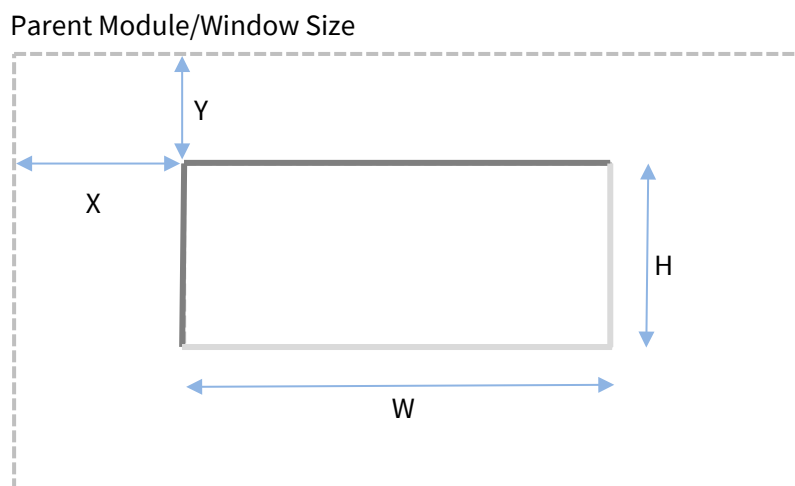
Border

Object Type: *[Obv\Border]*

The Border module draws a rectangular border around a rectangular area. The border can be a simple colour or can be 'raised' or sunk'.

It contains the following sub-objects:

Fig. 13 Border module

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner of rectangle around which the item is drawn	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle around which the item is drawn	Y	Obj\Num; Adjustable; Default value:0
Width Width of rectangle around which the item is drawn	W	Obj\Num; Adjustable; Default value:100
Height Height of rectangle around which the item is drawn	H	Obj\Num; Adjustable; Default value:20
Style The style of the border	S	Obj\ENum; Adjustable; Default value: 0; where 0=sunk, 1=raised, 2=colour
Colour The colour of the border, if Style is colour	C	Obj\WinClr; Adjustable; Default value: 0 (Black)
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1

Screen Appearance

The example shows a Border drawn on the screen. The border was specified 'Sunk'.
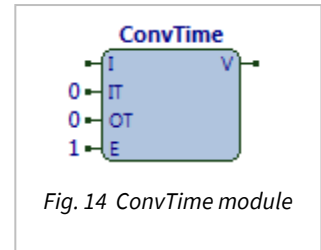
Related Modules

Rectangle

Convert-Time

Object Type: *[Obv\ConvTime]*

The Convert-Time module converts a time in one format to a time in another format

Format Type	Format
0	Seconds since midnight
1	Minutes since midnight
2	Decimal hours
3	hh:mm:ss
4	hh:mm

It contains the following sub-objects:

Description	Reference	Type
Input Input to convert, in the format defined by Input Type	I	Obj\Text; Adjustable; Max chars: 30; Default value:
Input Type The type format that the input is presented	IT	Obj\ENum; Adjustable; Default value:0; where 0=secs; 1=mins; 2=hrs; 3=hh:mm:ss, 4=hh:mm
Output Type The type format that the input is presented	OT	Obj\ENum; Adjustable; Default value:0; where 0=secs; 1=mins; 2=hrs; 3=hh:mm:ss, 4=hh:mm
Enable If set to 0, no calculation happens and V is left unchanged	E	Obj\NoYes; Adjustable; Default value:1
Value The time converted to the format defined by Output Type	V	Obj\Text; Max chars:30

Example
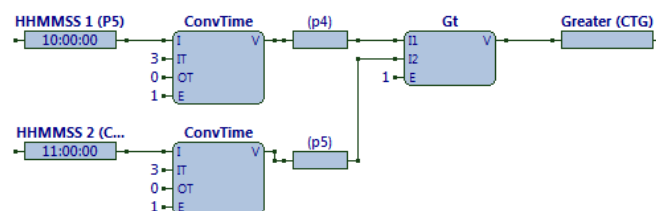
This strategy uses Convert-Time to compare two properties, each with the format hh:mm:ss, and sets a property to 1 if one the time in the first property is greater than the time in the second

Related Modules

Double-Click

Object Type: *[Obv\DblClk]*

The Double-click module monitors an area of the processor's window for mouse movement, and generates outputs if the mouse moves over the area, or the mouse is double-clicked (or if single-clicking is specified, the mouse is single-clicked)

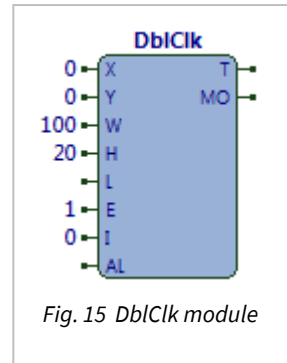It contains the following sub-objects:

Fig. 15 DblClk module

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner of rectangle being monitored	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle being monitored	Y	Obj\Num; Adjustable; Default value:0
Width Width of rectangle being monitored	W	Obj\Num; Adjustable; Default value:100
Height Height of rectangle being monitored	H	Obj\Num; Adjustable; Default value:20
Label The label of the area – shown within tooltip text	L	Obj\Text; Max chars: 64; Adjustable; Default value:
Action Label Prepended to the Label to create the tooltip	AL	Obj\Text; Max Chars:64; Adjustable; Default Value:
Inhibit If set to 1, the clicking is inhibited	I	Obj\NoYes; Adjustable; Default value:0
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1
Trigger Set to 1 momentarily when the mouse is double-clicked, or if the mouse is single-clicked when single-clicking is enabled	T	Obj\NoYes
MouseOver Set to 1 when the mouse is over the rectangle being monitored	MO	Obj\NoYes

Related Modules

Press

Enumerated-Value

Object Type: *[Obv\ENum]*

The Enumerated-Value module displays, and optionally allows changes to, an enumerated value.

The value is drawn within a specified rectangle. If not inhibited, the user can click on the value, at which point a dialog window appears to allow them to select a new value. The new value is available for use as necessary.
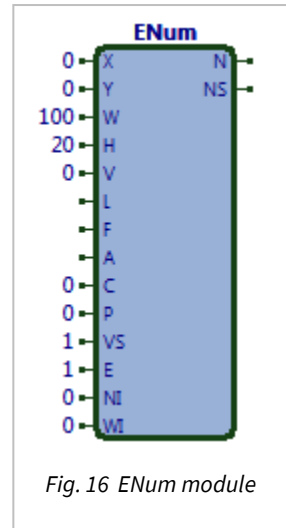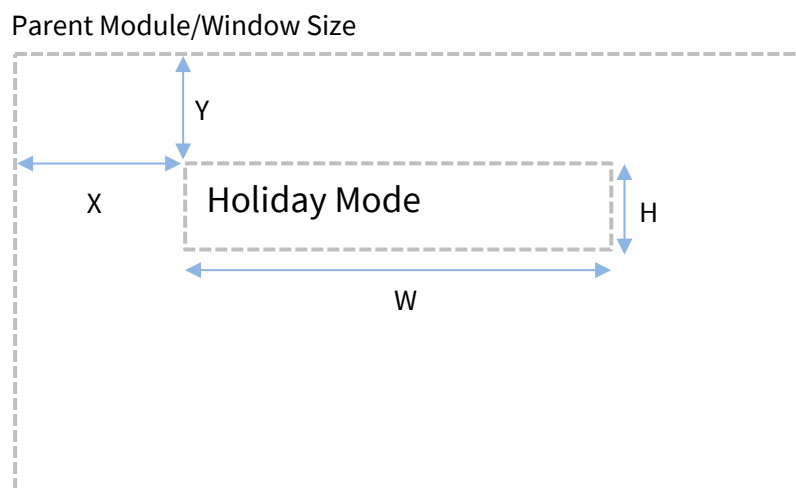

Fig. 16 ENum module

It contains the following sub-objects:

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner rectangle within which the enumerated value is drawn	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle within which the enumerated value is drawn	Y	Obj\Num; Adjustable; Default value:0
Width Width of rectangle within which the enumerated value is drawn	W	Obj\Num; Adjustable; Default value:100
Height Height of rectangle within which the enumerated value is drawn	H	Obj\Num; Adjustable; Default value:20
Value The current value as a number, to display as enumerated text from the Alternative text	V	Obj\ENum; Adjustable; Default value: 0; where the alternatives are defined in the Alternative input
Label The label of the enumerated value – shown on editing window, and within tooltip text	L	Obj\Text; Max chars: 31; Adjustable; Default value:
Font The font to display the enumerated value using	F	Obj\WinFont; Adjustable; Default value:
Alternatives A list of comma-separated text strings, where the first string represents the text to display if the Value is 0, and subsequent strings are used for subsequent values	A	Obj\Text; Max chars:127; Adjustable; Default value:
Colour The colour of the enumerated text	C	Obj\WinClr; Adjustable; Default value: 0 (Black)
Position The position of the enumerated text within the rectangle	P	Obj\Num; Adjustable; Default value: 0; where 0=LeftMiddle, 1=LeftBottom, 2=CentreBottom, 3=RightBottom, 4=LeftMiddle, 5=CentreMiddle, 6=RightomMiddle, 7=LeftTop, 8=CentreTop, 9=RightTop
Value Set Flag to indicate whether there is value to display – if not set, the box displays asterisks to indicate 'no value'	VS	Obj\NoYes; Adjustable; Default value: 1

Description	Reference	Type
New Init Reset from 1 to 0 to clear the New Value Set output. Usually used to indicate the new value has been used.	NI	Obj\NoYes; Adjustable; Default value: 0
Action Label Prepended to the Label to create the tooltip.	AL	Obj\Text; Max Chars:64; Adjustable; Default Value: 'Edit'
Write Inhibit Flag to inhibit editing of a new value. Set to 1 to inhibit	WI	Obj\NoYes; Adjustable; Default value:0
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1
New Value Set to the new value after the user has changed the value	N	Obj\ENum; where the alternatives are defined in the Alternative input
New Value Set Flag indicating a New Value is available	NS	Obj\NoYes

Screen Appearance

Parent Module/Window Size

The example shows a **ENum** with a current **Value** of '2', when the **Alternatives** are set to 'Off,On,Holiday Mode' – the ENum module has selected the text within Alternatives corresponding to 2. The text is shown with **Position** set to 'LeftTop'.

Related Modules

IP-Address-Value, *Numeric-Value*, *Text-Value*, *WinClr-Value*, *Times-Value*, *Alarm-Text-Value*

Greater-or-Equal

Object Type: *[Obv\GtEqual]*

The Greater-or-Equal module determines whether one input is greater than, or equal to, another.

Fig. 17 GtEqual module

The formula is:

> If E then
>> If I2 >= I1 then V = 1
>> Else V = 0

It contains the following sub-objects:

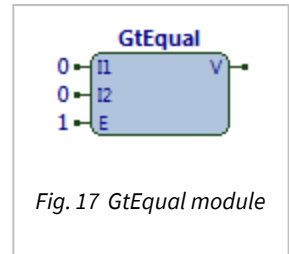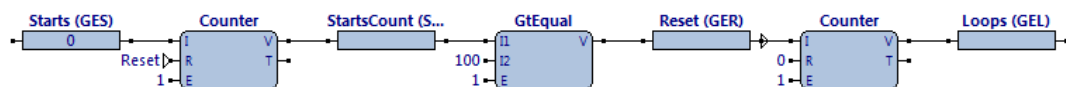| Description	Reference	Type
Input x Input x to compare, where x is in the range 1..2	Ix	Obj\Text; Adjustable; Default value:0
Enable If set to 0, no calculation happens and V is left unchanged	E	Obj\NoYes; Adjustable; Default value:1
Value Set to 1 if Input 1 is greater than Input 2	V	Obj\NoYes

Example

This strategy counts the Starts, and every 100 starts, increments Loops and resets the 100-start counter, using a Greater-or-Equal module

Related Modules

Less-or-Equal, *Greater*, *Equal*

Icon-File

Object Type: *[Obv\IconFile]*

The Icon-File module draws a single icon from within an icon file or can draw a sequence of icons from within one file to produce an animation. Icon files have various extensions (.ICO, .ICL) depending on the number of icons contained within them.
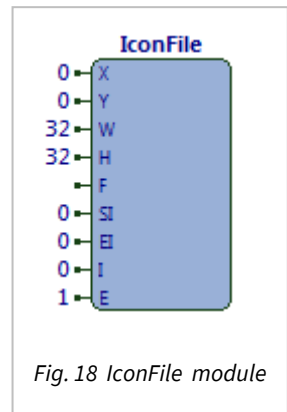

Fig. 18 IconFile module

It contains the following sub-objects:

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner of icon	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of icon	Y	Obj\Num; Adjustable; Default value:0
Width Width of icon	W	Obj\Num; Adjustable; Default value:32
Height Height of icon	H	Obj\Num; Adjustable; Default value:32
File The file name of the icon file containing the icon	F	Obj\File; Adjustable
Start Icon The index of the first icon within the file to draw as part of an animated sequence. Set same as End Icon to disable animation	SI	Obj\Num; Adjustable; Default value:0
End Icon The index of the last icon within the file to draw as part of an animated sequence. Set same as End Icon to disable animations	EI	Obj\Num; Adjustable; Default value:0
Idle Rate (ms) The animation idle rate – the time between each icon in the sequence. Limited by the idle rate of the whole ObVerse	I	Obj\Num; Adjustable; Default value:0
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1

Screen Appearance

Parent Module/Window Size
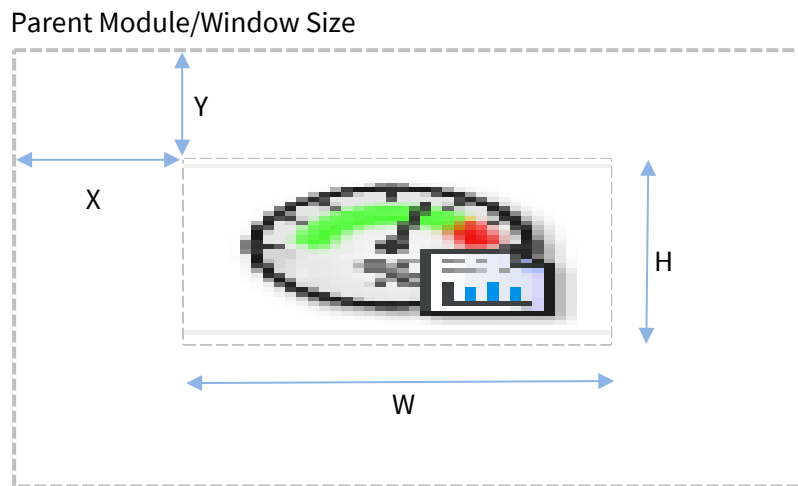
The example shows an Icon, stretched to fit a rectangle on the screen.

Related Modules

BMP-File, *Windows-Meta-File*

IP-Address-Value

Object Type: *[Obv\IPAddr]*

The IP-Address-Value module displays, and optionally allows changes to, an IP address value.

The value is displayed within a specified rectangle. If not inhibited, the user can click on the value, at which point a dialog window appears to allow them to select a new value. The new value is available for use as necessary.

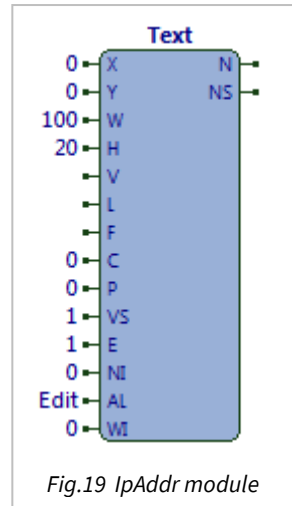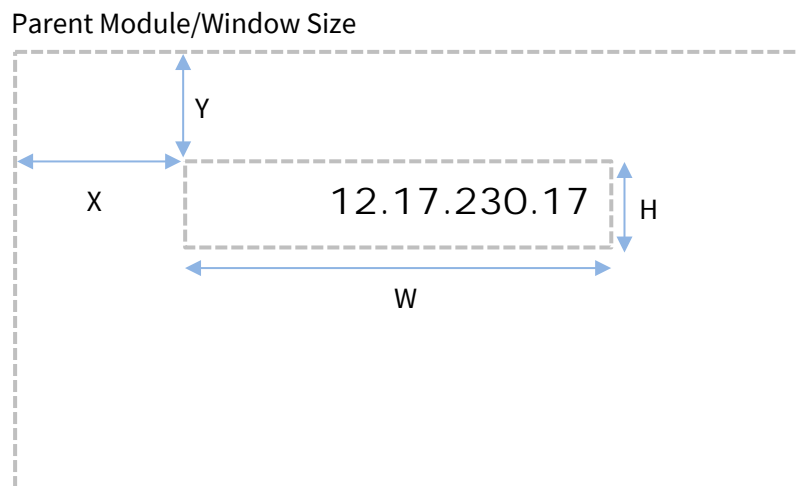It contains the following sub-objects:

Fig.19 IpAddr module

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner rectangle within which the alarm value is drawn	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle within which the alarm value is drawn	Y	Obj\Num; Adjustable; Default value:0
Width Width of rectangle within which the alarm value is drawn	W	Obj\Num; Adjustable; Default value:100
Height Height of rectangle within which the alarm text is drawn	H	Obj\Num; Adjustable; Default value:20
Value The current value of the IP address	V	Obj\IP; Adjustable; Default value:
Label The label of the text value – shown on editing window, and within tooltip text	L	Obj\Text; Max chars: 31; Adjustable; Default value:
Font The font to display the text value using	F	Obj\WinFont; Adjustable; Default value:
Colour The colour of the text	C	Obj\WinClr; Adjustable; Default value: 0 (Black)
Position The position of the text within the rectangle	P	Obj\Num; Adjustable; Default value: 0; where 0=LeftMiddle, 1=LeftBottom, 2=CentreBottom, 3=RightBottom, 4=LeftMiddle, 5=CentreMiddle, 6=RightomMiddle, 7=LeftTop, 8=CentreTop, 9=RightTop
Value Set Flag to indicate whether there is value to display – if not set, the box displays asterisks to indicate 'no value'	VS	Obj\NoYes; Adjustable; Default value: 1
New Init Reset from 1 to 0 to re-initialise the New Value Set output. Usually used to indicate the new value has been used.	NI	Obj\NoYes; Adjustable; Default value: 0
Action Label Prepended to the Label to create the tooltip.	AL	Obj\Text; Max Chars:64; Adjustable; Default Value: 'Edit'
Write Inhibit Flag to inhibit editing of a new value. Set to 1 to inhibit	WI	Obj\NoYes; Adjustable; Default value:0
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1

Description	Reference	Type
New Value Set to the new value after the user has changed the value	N	Obj\IP
New Value Set Flag indicating a New Value is available	NS	Obj\NoYes

Screen Appearance

Parent Module/Window Size

The example shows an **IpAddr** with a current **Value** of '12.17.230.17'. The text is shown with **Position** set to 'RightTop', with **Font** set to 'Arial Black 20'

Related Modules

Enumerated-Value, *Numeric-Value*, *Text-Value*, *WinClr-Value*, *Times-Value*, *Alarm-Text-Value*

Less-or-Equal

Object Type: *[Obv\LessEqual]*

Fig. 20 LessEqual module

The Less-or-Equal module determines whether one input is less than, or equal to, another.

The formula is:

>　　If E then
>　　　　If I2 <= I1 then V = 1
>　　　　Else V = 0

It contains the following sub-objects:

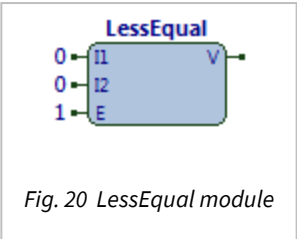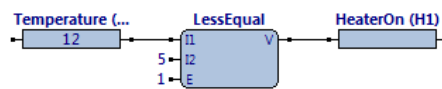| Description	Reference	Type
Input *x* Input x to compare, where *x* is in the range 1..2	I*x*	Obj\Text; Adjustable; Default value:0
Enable If set to 0, no calculation happens and V is left unchanged	E	Obj\NoYes; Adjustable; Default value:1
Value Set to 1 if Input 1 is greater than Input 2	V	Obj\NoYes

Example

Related Modules

Greater-or-Equal, *Less*, *Equal*

Menu-Item

Object Type: *[Obv\MenuItem]*

The Menu-Item module creates a new menu item on the processor's window menu bar. The menu bar has a list of menus, and each menu has one or more items. This module creates and maintains on one item on one menu.

Fig. 21 MenuItem module

It contains the following sub-objects:

Description	Reference	Type
Menu Label The label of the menu on which to place this item. If the menu does not exist, it will be created.	M	Obj\Text; Adjustable; Max chars:64; Default value:
Item Label The label of the item on the menu. If the menu does not exist, it will be created. If the item changes label, the old item will be removed, and the new one created.	I	Obj\Text; Adjustable; Max chars:64; Default value:
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1
Trigger When the menu item is selected, the Trigger is set to 1 for 1 cycle of the processor	T	Obj\NoYes
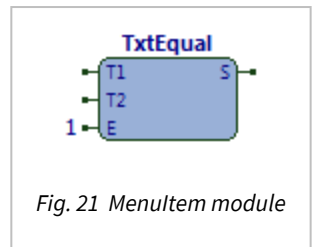
Screen Appearance
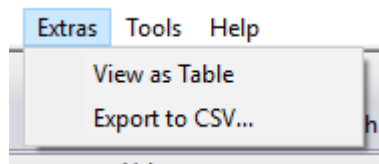
The menu above was created with two **MenuItem** module. The **Menu Label** for both was set to 'Extras'. The **Item Label** of the first was set to 'View as Table', and the second was set to 'Export to SCV…'

Related Modules

Not-Equal

Object Type: *[Obv\NotEqual]*

Fig. 22 NotEqual module

The Not-Equal module compares two inputs and determines whether they are not equal (i.e. different).

The formula is:

> If E then
>> If I1 == I2 then V=0
>> Else V=1
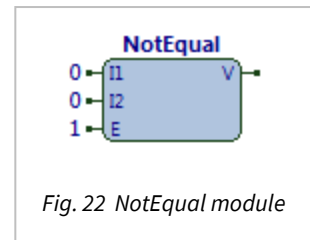
It contains the following sub-objects:

Description	Reference	Type
Input *x* Input to possibly include in calculation, where *x* is in the range 1...2 Inputs 1&2 are compared	I*x*	Obj\Float; Adjustable; Default value:0
Enable If set to 0, no calculation happens and V is left unchanged	E	Obj\NoYes; Adjustable; Default value:1
Value The last calculated value	V	Obj\NoYes

Example

Related Modules

Equal, *Greater*, *Less*

Numeric-Value

Object Type: *[Obv\Num]*

The Numeric-Value module displays, and optionally allows changes to, a numeric value. The value cab be either a floating-point number or an integer number

The value is drawn within a specified rectangle. If not inhibited, the user can click on the value, at which point a dialog window appears to allow them to select a new value. The new value is available for use as necessary.

It contains the following sub-objects:

Fig. 23 Num module

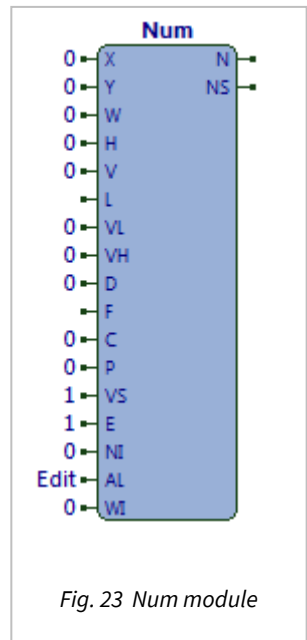| Description	Reference	Type
X-Coordinate X-coordinate of top-left corner rectangle within which the numeric value is drawn	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle within which the numeric value is drawn	Y	Obj\Num; Adjustable; Default value:0
Width Width of rectangle within which the numeric value is drawn	W	Obj\Num; Adjustable; Default value:100
Height Height of rectangle within which the numeric value is drawn	H	Obj\Num; Adjustable; Default value:20
Value The current value	V	Obj\Float; Adjustable; Default value: 0
Value High The high limit for the value	VH	Obj\Float; Adjustable; Default value: 0
Value Low The low limit for the value	VL	Obj\Float; Adjustable; Default value: 0
Decimal Places	D	Obj\Num; Adjustable; Default value: 0; Range 0...4
Label The label of the enumerated value – shown on editing window, and within tooltip text	L	Obj\Text; Max chars: 31; Adjustable; Default value:
Font The font to display the enumerated value using	F	Obj\WinFont; Adjustable; Default value:
Colour The colour of the enumerated text	C	Obj\WinClr; Adjustable; Default value: 0 (Black)
Position The position of the enumerated text within the rectangle	P	Obj\Num; Adjustable; Default value: 0; where 0=LeftMiddle, 1=LeftBottom, 2=CentreBottom, 3=RightBottom, 4=LeftMiddle, 5=CentreMiddle, 6=RightomMiddle, 7=LeftTop, 8=CentreTop, 9=RightTop
Value Set Flag to indicate whether there is value to display – if not set, the box displays asterisks to indicate 'no value'	VS	Obj\NoYes; Adjustable; Default value: 1

Description	Reference	Type
New Init Reset from 1 to 0 to re-initialise the New Value Set output. Usually used to indicate the new value has been used.	NI	Obj\NoYes; Adjustable; Default value: 0
Action Label Prepended to the Label to create the tooltip.	AL	Obj\Text; Max Chars:64; Adjustable; Default Value: 'Edit'
Write Inhibit Flag to inhibit editing of a new value. Set to 1 to inhibit	WI	Obj\NoYes; Adjustable; Default value:0
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1
New Value Set to the new value after the user has changed the value	N	Obj\ENum; where the alternatives are defined in the Alternative input
New Value Set Flag indicating a New Value is available	NS	Obj\NoYes

Related Modules

Enumerated-Value, *IP-Address-Value*, *Text-Value*, *WinClr-Value*, *Times-Value*, *Alarm-Text-Value*

Object-List

Object Type: *[Obv\ObjList]*

The Object-List module displays a list of sub-objects of a specified object. It builds the list using the current *contents.obc* file for the object. It draws each of the items in the list using a preferred entry view. A simple object filter is available if necessary.
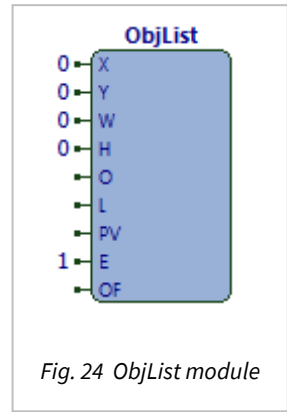

Fig. 24 ObjList module

It contains the following sub-objects:

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner of rectangle into which the item is drawn	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle into which the item is drawn	Y	Obj\Num; Adjustable; Default value:0
Width Width of rectangle into which the item is drawn	W	Obj\Num; Adjustable; Default value:0
Height Height of rectangle into which the item is drawn	H	Obj\Num; Adjustable; Default value:0
Object The object reference to draw the sub-objects for	O	Obj\Obj; Adjustable; Default value:
Label The label of the object	L	Obj\Text; Adjustable; Max chars: 32; Default value:
Preferred View/ObVerse The View/ObVerse file to select (if possible) for displaying sub-object entries – this text will have 'Entry' appended when searching for the preferred view. If the preferred view is not available, Object List will use 'default' as the preferred view	PV	Obj\Text; Adjustable; Max chars:32; Default value:
Object Filter If set, the start of the sub-object references must match this filter string to be displayed	OF	Obj\Obj; Adjustable; Default value:
Enable If set to 0, no calculation happens and V is left unchanged	E	Obj\NoYes; Adjustable; Default value:1

Related Modules

Obj-Value

Object Type: *[Obv\ObjText]*

The Obj-Value module displays, and optionally allows changes to, an object reference value.

The value is drawn within a specified rectangle. If not inhibited, the user can click on the value, at which point a dialog window appears to allow them to select a new value. The new value is available for use as necessary.
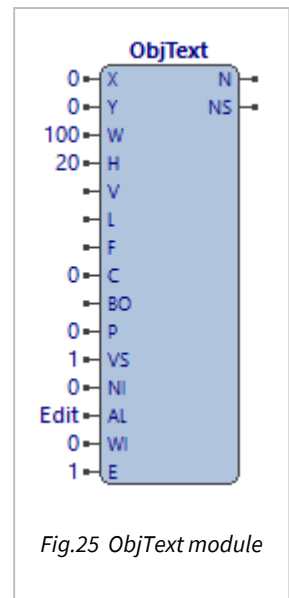

Fig.25 ObjText module

It contains the following sub-objects:

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner rectangle within which the text value is drawn	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle within which the text value is drawn	Y	Obj\Num; Adjustable; Default value:0
Width Width of rectangle within which the text value is drawn	W	Obj\Num; Adjustable; Default value:100
Height Height of rectangle within which the numeric text is drawn	H	Obj\Num; Adjustable; Default value:20
Value The current value of the text	V	Obj\Text; Max chars: 127; Adjustable; Default value:
Label The label of the text value – shown on editing window, and within tooltip text	L	Obj\Text; Max chars: 31; Adjustable; Default value:
Font The font to display the text value using	F	Obj\WinFont; Adjustable; Default value:
Colour The colour of the text	C	Obj\WinClr; Adjustable; Default value: 0 (Black)
Position The position of the text within the rectangle	P	Obj\Num; Adjustable; Default value: 0; where 0=LeftMiddle, 1=LeftBottom, 2=CentreBottom, 3=RightBottom, 4=LeftMiddle, 5=CentreMiddle, 6=RightomMiddle, 7=LeftTop, 8=CentreTop, 9=RightTop
Base Object Reference Object reference in which the actual object reference is located	BO	Obj\Text; Adjustable
Value Set Flag to indicate whether there is value to display – if not set, the box displays asterisks to indicate 'no value'	VS	Obj\NoYes; Adjustable; Default value: 1
New Init Reset from 1 to 0 to re-initialise the New Value Set output. Usually used to indicate the new value has been used.	NI	Obj\NoYes; Adjustable; Default value: 0

Description	Reference	Type
Action Label Prepended to the Label to create the tooltip.	AL	Obj\Text; Max Chars:64; Adjustable; Default Value: 'Edit'
Write Inhibit Flag to inhibit editing of a new value. Set to 1 to inhibit	WI	Obj\NoYes; Adjustable; Default value:0
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1
New Value Set to the new value after the user has changed the value	N	Obj\ENum; where the alternatives are defined in the Alternative input
New Value Set Flag indicating a New Value is available	NS	Obj\NoYes

Related Modules

Enumerated-Value, *IP-Address-Value*, *Numeric-Value*, *WinClr-Value*, *Times-Value*, *Alarm-Text-Value*, *Text-Value*

Press

Object Type: *[Obv\Press]*

The Press module monitors an area of the processor's window for mouse movement, and generates outputs if the mouse moves over the area, or the mouse is single-clicked
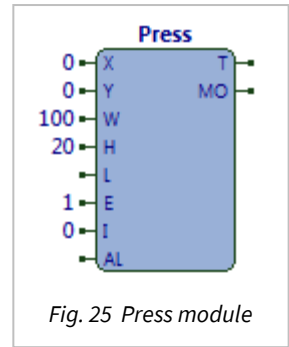

Fig. 25 Press module

It contains the following sub-objects:

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner of rectangle being monitored	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle being monitored	Y	Obj\Num; Adjustable; Default value:0
Width Width of rectangle being monitored	W	Obj\Num; Adjustable; Default value:100
Height Height of rectangle being monitored	H	Obj\Num; Adjustable; Default value:20
Label The label of the area – shown within tooltip text	L	Obj\Text; Max chars: 64; Adjustable; Default value:
Action Label Prepended to the Label to create the tooltip.	AL	Obj\Text; Max Chars:64; Adjustable; Default Value:
Inhibit If set to 1, the clicking is inhibited	I	Obj\NoYes; Adjustable; Default value:0
File The file name of the .WMF file to drawn in the rectangle	W	Obj\File; Adjustable; Default value:
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1
Trigger Set to 1 momentarily when the mouse is single-clicked,	T	Obj\NoYes
MouseOver Set to 1 when the mouse is over the rectangle being monitored	MO	Obj\NoYes

Related Modules

Double-Click

Rectangle

Object Type: *[Obv\Rect]*

The Rectangle module draws a rectangular area

Fig. 26 Rect module

It contains the following sub-objects:

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner of rectangle	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle	Y	Obj\Num; Adjustable; Default value:0
Width Width of rectangle	W	Obj\Num; Adjustable; Default value:100
Height Height of rectangle	H	Obj\Num; Adjustable; Default value:20
Colour The colour of the rectangle	C	Obj\WinClr; Adjustable; Default value: 0 (Black)
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1

Related Modules

Border

Run-ObVerse

Object Type: *[Obv\RunObv]*

The Run-ObVerse module loads and starts executing an ObVerse file. The file is searched for within the folder associated with the type of the specified object

It contains the following sub-objects:
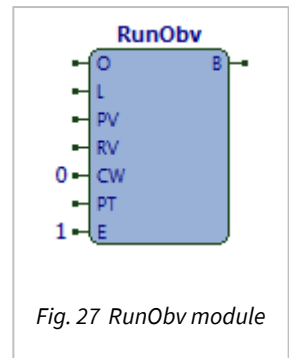

Fig. 27 RunObv module

Description	Reference	Type
Object The object reference of the object that the ObVerse is associated with.	O	Obj\Obj; Adjustable; Default value:
Label The label of the object	L	Obj\Text; Adjustable; Max chars: 32; Default value:
Preferred View/ObVerse The View/ObVerse file to select (if possible) for displaying sub-object entries – this text will have 'Entry' appended when searching for the preferred view. If the preferred view is not available, Object List will use 'default' as the preferred view	PV	Obj\Text; Adjustable; Max chars:32; Default value:
Run View/ObVerse The name of the View/ObVerse file. This will be found in the TypeInfo folder of the object type of the Object specified	RV	Obj\Text; Adjustable; Max chars:31
Close Window Whether the window running this View/ObVerse should be closed, as the specified View/ObVerse is run	CW	Obj\NoYes; Adjustable; Default value: 0
Parameter Text Text to add to the execute instruction before passed to the OS	PT	Obj\Text; Adjustable; Max chars:127
Enable If set to 1, the execution is triggered.	E	Obj\NoYes; Adjustable; Default value:1
Busy Set to 1 when triggering the ObVerse	B	Obj\NoYes

Related Modules

Windows-Execute

Security

Object Type: *[Obv\Security]*

Fig.28 Security module

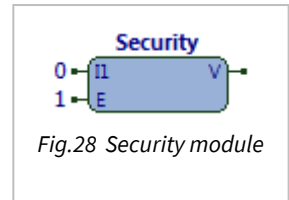The Security module checks whether the current user has privileges to access an item.

Within North products, items requiring security clearance are associated with one of eight security areas. These areas are numbered 1…8 and can be used for any purpose by the engineer.

Each user has a predefined privilege level for each of these areas. Privilege levels are in the range 0…7, where 0 means no privileges, and 7 means maximum privileges.

The standard method used to define which area and level are needed to access a point-of-control is to specify a two-digit access-security code, where the first digit defines the area, and the second defines the minimum level.

For example, if a task has an access security code of 73, then the user must have level 3 privileges in area 7 to perform this task.

The Security module contains the following sub-objects:

Description	Reference	Type
Input The two-digit access security code to check user against. The area digit is in the range 1..8, the level digit is in the range 0..7	I1	Obj\Num; Adjustable; Default value: 0
Enable If set to 0, no calculation happens and V is left unchanged	E	Obj\NoYes; Adjustable; Default value:1
Value Whether the user currently logged in has privilege level equal to (or higher) than the required level in the specified area	V	Obj\Float; value will be set to 0 to indicate user doesn't have clearance, or 1 to indicate user has clearance

Related Modules

Sound

Object Type: *[Obv\Sound]*

Fig. 29 Sound module

The Sound module plays a .WAV file using the PC's speakers. The sound is triggered using the enable input, and plays once.

The Sound module contains the following sub-objects:

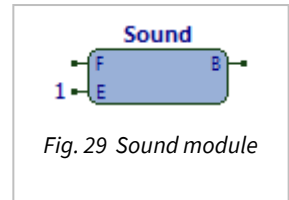| Description	Reference	Type
File The file name of the WAV file	F	Obj\File; Adjustable; Default value:
Enable If set to 1, the WAV file is played once	E	Obj\NoYes; Adjustable; Default value:1
Busy Whether the WAV file is currently playing	B	Obj\NoYes

Related Modules

SysInfo

Object Type: *[Obv\SysInfo]*

The SysInfo module makes fundamental system information available for use in ObVerse. Its main input selects which system information is required:

Fig. 30 SysInfo module
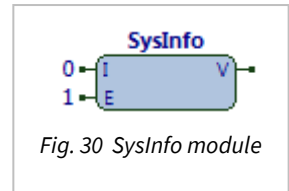
Input	Information
1	Current Seconds
2	Current Minutes
3	Current Hours
4	Current Day-of-Month
5	Current Month
6	Current Year
7	Current Day-of-Week (0=Monday)
10	Pulse on Start-up (1 cycle)
11	Pulse on Second-change (1 cycle)
12	Pulse on Minute-change (1 cycle)
13	Pulse on Hour-change (1 cycle)
14	Pulse on Day-change (1 cycle)
15	Pulse on Month-change (1 cycle)
101	Window background colour
102	Window text colour
103	Window highlight background colour
104	Window highlight text colour
105	Window menu background colour
106	Window menu text colour
107	Window button background colour
108	Window button text colour
121	Window client width
122	Window client height
200	<top> device label
201	Current device label
202	Current object reference, excluding marked object prefix
203	Is current object within marked object
204	Marked object reference
10xx	Window label of previous view (xx=01...10)
11xx	Object reference of previous view (xx=01...10)
12xx	Previous view was a device (xx=01...10)

It contains the following sub-objects:

Description	Reference	Type
Input The information item required	I	Obj\Num; Adjustable; Default value:0
Enable If set to 0, no calculation happens and V is left unchanged	E	Obj\NoYes; Adjustable; Default value:1
Value The value of the specified item	V	Obj\Text

Related Modules

Static-Text

Object Type: *[Obv\SText]*

The Static-Text module draws text within a rectangle. Although the input text can change, the user cannot change the text, unlike the Text module.

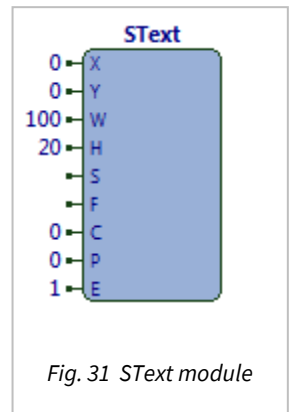Static text is usually used to draw fixed text on a display.

Fig. 31 SText module

It contains the following sub-objects:

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner of rectangle	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle	Y	Obj\Num; Adjustable; Default value:0
Width Width of rectangle	W	Obj\Num; Adjustable; Default value:100
Height Height of rectangle	H	Obj\Num; Adjustable; Default value:20
String The text to draw	S	Obj\Text; Max chars 127; Default value:
Font The font to use to display the text	F	Obj\WinFont; Adjustable; Default value:
Colour The colour of the rectangle	C	Obj\WinClr; Adjustable; Default value: 0 (Black)
Position The position of the text within the rectangle	P	Obj\Num; Adjustable; Default value: 0; where 0=LeftMiddle, 1=LeftBottom, 2=CentreBottom, 3=RightBottom, 4=LeftMiddle, 5=CentreMiddle, 6=RightomMiddle, 7=LeftTop, 8=CentreTop, 9=RightTop
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1

Related Modules

Text-Value

Object Type: *[Obv\Text]*

The Text-Value module displays, and optionally allows changes to, a text value.

The value is drawn within a specified rectangle. If not inhibited, the user can click on the value, at which point a dialog window appears to allow them to select a new value. The new value is available for use as necessary.
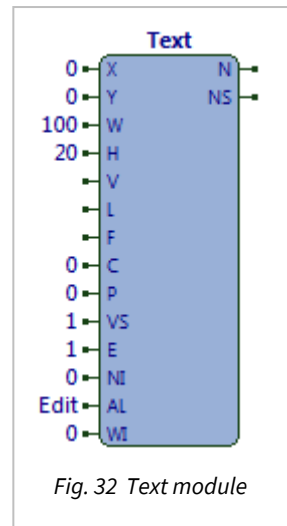

Fig. 32 Text module

It contains the following sub-objects:

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner rectangle within which the text value is drawn	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle within which the text value is drawn	Y	Obj\Num; Adjustable; Default value:0
Width Width of rectangle within which the text value is drawn	W	Obj\Num; Adjustable; Default value:100
Height Height of rectangle within which the numeric text is drawn	H	Obj\Num; Adjustable; Default value:20
Value The current value of the text	V	Obj\Text; Max chars: 127; Adjustable; Default value:
Label The label of the text value – shown on editing window, and within tooltip text	L	Obj\Text; Max chars: 31; Adjustable; Default value:
Font The font to display the text value using	F	Obj\WinFont; Adjustable; Default value:
Colour The colour of the text	C	Obj\WinClr; Adjustable; Default value: 0 (Black)
Position The position of the text within the rectangle	P	Obj\Num; Adjustable; Default value: 0; where 0=LeftMiddle, 1=LeftBottom, 2=CentreBottom, 3=RightBottom, 4=LeftMiddle, 5=CentreMiddle, 6=RightomMiddle, 7=LeftTop, 8=CentreTop, 9=RightTop
Value Set Flag to indicate whether there is value to display – if not set, the box displays asterisks to indicate 'no value'	VS	Obj\NoYes; Adjustable; Default value: 1
New Init Reset from 1 to 0 to re-initialise the New Value Set output. Usually used to indicate the new value has been used.	NI	Obj\NoYes; Adjustable; Default value: 0
Action Label Prepended to the Label to create the tooltip.	AL	Obj\Text; Max Chars:64; Adjustable; Default Value: 'Edit'
Write Inhibit Flag to inhibit editing of a new value. Set to 1 to inhibit	WI	Obj\NoYes; Adjustable; Default value:0

Description	Reference	Type
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1
New Value Set to the new value after the user has changed the value	N	Obj\ENum; where the alternatives are defined in the Alternative input
New Value Set Flag indicating a New Value is available	NS	Obj\NoYes

Related Modules

Enumerated-Value, *IP-Address-Value*, *Numeric-Value*, *WinClr-Value*, *Times-Value*, *Alarm-Text-Value*,

Times-Value

Object Type: *[Obv\Times]*

The Times-Value module displays, and optionally allows changes to, a list of on-off times.

The value is drawn within a specified rectangle. If not inhibited, the user can click on the value, at which point a dialog window appears to allow them to select a new value. The new value is available for use as necessary.
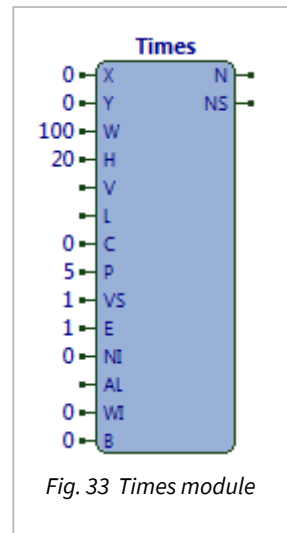
Fig. 33 Times module

It contains the following sub-objects:

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner rectangle within which the on-off-times are drawn	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle within which the on-off-times are drawn	Y	Obj\Num; Adjustable; Default value:0
Width Width of rectangle within which the on-off-times are drawn	W	Obj\Num; Adjustable; Default value:100
Height Height of rectangle within which the on-off-times are drawn	H	Obj\Num; Adjustable; Default value:20
Value The current on-off times	V	Obj\Times; Adjustable; Default value:
Periods The maximum number of on-off periods acceptable in the New Value	P	Obj\Num; Adjustable; Default value: 0; Range 0…5
Label The label of the on-off times – shown on editing window, and within tooltip text	L	Obj\Text; Max chars: 31; Adjustable; Default value:
Colour The colour of the enumerated text	C	Obj\WinClr; Adjustable; Default value: 0 (Black)
Value Set Flag to indicate whether there is value to display – if not set, the box displays asterisks to indicate 'no value'	VS	Obj\NoYes; Adjustable; Default value: 1
New Init Reset from 1 to 0 to re-initialise the New Value Set output. Usually used to indicate the new value has been used.	NI	Obj\NoYes; Adjustable; Default value: 0
Action Label Prepended to the Label to create the tooltip.	AL	Obj\Text; Max Chars:64; Adjustable; Default Value: 'Edit'
Write Inhibit Flag to inhibit editing of a new value. Set to 1 to inhibit	WI	Obj\NoYes; Adjustable; Default value:0
Background Colour	B	Obj\WinClr; Adjustable; Default value: 0 (Black)
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1

Description	Reference	Type
New Value Set to the new value after the user has changed the value	N	Obj\Times
New Value Set Flag indicating a New Value is available	NS	Obj\NoYes

Related Modules

Enumerated-Value, *IP-Address-Value*, *Numeric-Value*, *Text-Value*, *WinClr-Value*, *Alarm-Text-Value*,

Trace

Object Type: *[Obv\Trace]*

The Trace module records a value and displays a trace of that value within a rectangle. The trace builds over time and supports a maximum of 100 readings.

The high and low limits of the value are input, and the trace draws the readings proportionally between the 'sides', like a chart-recorder.
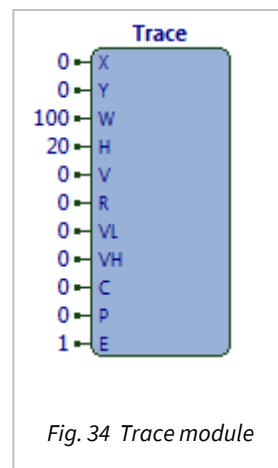

Fig. 34 Trace module

It contains the following sub-objects:

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner of rectangle	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle	Y	Obj\Num; Adjustable; Default value:0
Width Width of rectangle	W	Obj\Num; Adjustable; Default value:100
Height Height of rectangle	H	Obj\Num; Adjustable; Default value:20
Value The value to trace	V	Obj\Float; Adjustable; Default value:0
Refresh Rate (seconds) The rate at which the value is sampled, and the trace is re-drawn	R	Obj\Float; Adjustable; Range 0..300; Default value:5
Value High The high limit for the value	VH	Obj\Float; Adjustable; Default value: 0
Value Low The low limit for the value	VL	Obj\Float; Adjustable; Default value: 0
Colour The colour of the rectangle	C	Obj\WinClr; Adjustable; Default value: 0 (Black)
Position The position from where the trace 'grows'	P	Obj\Num; Adjustable; Default value: 0; where 0=LeftMiddle, 1=LeftBottom, 2=CentreBottom, 3=RightBottom, 4=LeftMiddle, 5=CentreMiddle, 6=RightMiddle, 7=LeftTop, 8=CentreTop, 9=RightTop
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1

Related Modules

Bar, *Variable-Line*

View-List

Object Type: *[Obv\ViewList]*

The View-List module displays a list of alternative views for an object. It builds the list from the folder associated with the type of the object. It draws the list using a preferred entry view for each item.
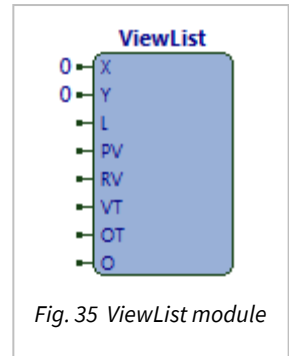

Fig. 35 ViewList module

It contains the following sub-objects:

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner of rectangle into which the object list is drawn	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle into which the object list is drawn	Y	Obj\Num; Adjustable; Default value:0
Object The object reference to draw the list of alternative views	O	Obj\Obj; Adjustable; Default value:
Label The label of the object	L	Obj\Text; Adjustable; Max chars: 32; Default value:
Preferred View/ObVerse The View/ObVerse file to select (if possible) for displaying view entries – this text will have 'Entry' appended when searching for the preferred view. If the preferred view is not available, Object List will use 'default' as the preferred view	PV	Obj\Text; Adjustable; Max chars:32; Default value:
Run View	RV	Obj\Text; Adjustable; Max chars: 127
View Type	VT	Obj\Text; Adjustable; Max chars:127
Object Type If set, the start of the sub-object references must match this filter string to be displayed	OT	Obj\Text; Adjustable; Default value:

Related Modules

Object-List

Video-Overlay

Object Type: *[Obv\VideoOverlay]*

Windows itself supports several sources of video, and the Video-Overlay module overlays one of these 'capture' sources onto a rectangle of the processor's window.

This is typically used to put a camera feed onto an ObVerse window.

Fig. 36 VideoOverlay module

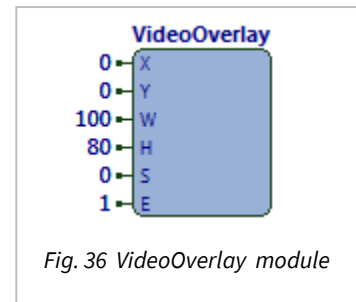It contains the following sub-objects:

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner of rectangle into which the video is overlaid	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle into which the video is overlaid	Y	Obj\Num; Adjustable; Default value:0
Width Width of rectangle into which the video is overlaid	W	Obj\Num; Adjustable; Default value:100
Height Height of rectangle into which the video is overlaid	H	Obj\Num; Adjustable; Default value:20
Source The source of the video	S	Obj\Num; Adjustable; Default value:0
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1

Related Modules

Variable-Line

Object Type: *[Obv\VLine]*

The Variable-Line module draws a straight line between the two sides of a rectangle. The position the line meets each of the two sides is specified used two input values, which are used along with high and low values to determine the actual location of the two ends.

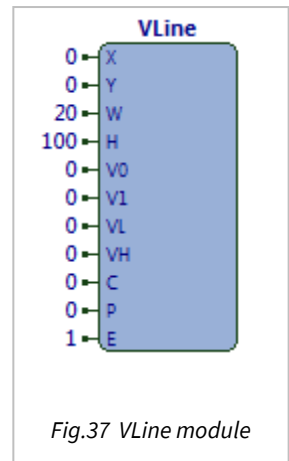It contains the following sub-objects:

Fig.37 VLine module

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner of rectangle	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle	Y	Obj\Num; Adjustable; Default value:0
Width Width of rectangle	W	Obj\Num; Adjustable; Default value:20
Height Height of rectangle	H	Obj\Num; Adjustable; Default value:100
Value 0 Value to position start of line, relative to Value Low and Value High Limits	V0	Obj\Float; Adjustable; Default value: 0
Value 1 Value to position end of line, relative to Value Low and Value High Limits	V1	Obj\Float; Adjustable; Default value: 0
Value Low Limit	VL	Obj\Float; Adjustable; Default value: 0
Value High Limit	VH	Obj\Float; Adjustable; Default value: 0
Colour The colour of the line	C	Obj\WinClr; Adjustable; Default value: 0 (Black)
Position The position/direction that the two sides grow	P	Obj\ENum; Adjustable; Range 0…3; where 0=FromBottom, 1=FromLeft, 2=FromTop, 3=FromRight
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1

Related Modules

Bar, *Trace*

WinClr-Value

Object Type: *[Obv\WinClr]*

The WinClr-Value module displays, and optionally allows changes to a WinClr (Windows Colour) value.

The colour value is displayed as a coloured rectangle. If not inhibited, the user can click on the colour, at which point a dialog window appears to allow them to select a new colour value. The new colour value is available for use as necessary.
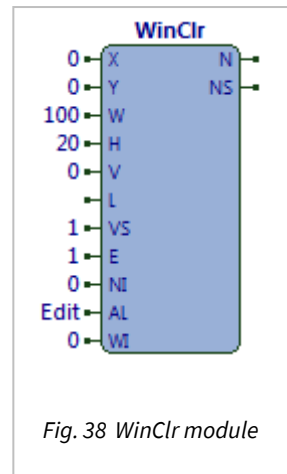
It contains the following sub-objects:

Fig. 38 WinClr module

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner rectangle within which the text value is drawn	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle within which the colour value is drawn	Y	Obj\Num; Adjustable; Default value:0
Width Width of rectangle within which the colour value is drawn	W	Obj\Num; Adjustable; Default value:100
Height Height of rectangle within which the colour text is drawn	H	Obj\Num; Adjustable; Default value:20
Value The current value of the colour	V	Obj\WinClr; Adjustable; Default value: 0 (Black)
Label The label of the colour value – shown on editing window, and within tooltip text	L	Obj\Text; Max chars: 31; Adjustable; Default value:
Value Set Flag to indicate whether there is value to display – if not set, the box displays asterisks to indicate 'no value'	VS	Obj\NoYes; Adjustable; Default value: 1
New Init Reset from 1 to 0 to re-initialise the New Value Set output. Usually used to indicate the new value has been used.	NI	Obj\NoYes; Adjustable; Default value: 0
Action Label Prepended to the Label to create the tooltip.	AL	Obj\Text; Max Chars:64; Adjustable; Default Value: 'Edit'
Write Inhibit Flag to inhibit editing of a new value. Set to 1 to inhibit	WI	Obj\NoYes; Adjustable; Default value:0
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1
New Value Set to the new colour value after the user has changed the value	N	Obj\ENum; where the alternatives are defined in the Alternative input
New Value Set Flag indicating a New Value is available	NS	Obj\NoYes

Related Modules

Enumerated-Value, *IP-Address-Value*, *Numeric-Value*, *Text-Value*, *Times-Value*, *Alarm-Text-Value*,

Windows-Execute

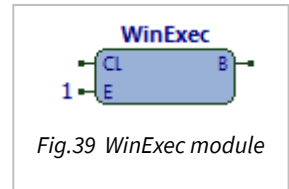Object Type: *[Obv\WinExec]*

Fig.39 WinExec module

The Windows-Execute module allows ObVerse to trigger the start of other programs. There are two main ways this can be done, either specifying the full command line to Windows, or specifying the data file, and allowing Windows to determine the application to use for that file-type.

The command line, by default, will be passed to Windows' own WinExec command. The command line can contain local property references enclosed in % characters (for example %L%), in which case the property reference (and % characters) is replaced with the current value of that property.

If the command line begins 'shell ', the rest of the command line is deemed to be a data filename, and it is passed to the shell for execution with the default verb (usually 'open'). The shell will find the associated application.

If the command line begins 'shellprint ', the rest of the command line is deemed to be a data filename, and it is passed to the shell for execution with the 'print' verb. The shell will find the associated application.

If the command line begins 'help ', the rest of the command line is deemed to be an ObSys help filename within the default ObSys help folder. The complete file name is passed to the shell for execution with the default verb (usually 'open'). The shell will find the associated application.

It contains the following sub-objects:

Description	Reference	Type
Command Line The command line to execute, with %objref% replacements, and special starting texts	CL	Obj\Text; Max chars: 127; Adjustable; Default value:
Enable If set to 1, the command line is triggered	E	Obj\NoYes; Adjustable; Default value:1
Busy This output is set to 1 when the command line is triggered	B	Obj\NoYes

Related Modules

Run-ObVerse

Windows-Meta-File

Object Type: *[Obv\WmFile]*

The Windows-Meta-File module draws a .WMF file. Windows Meta-files are used to hold platform independent vector drawing data and are supported by popular Windows-based creation/editing applications.
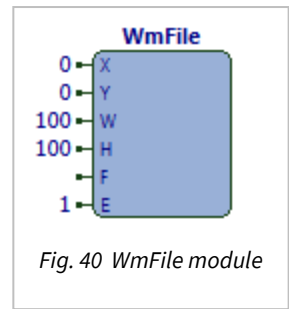

Fig. 40 WmFile module

It contains the following sub-objects:

Description	Reference	Type
X-Coordinate X-coordinate of top-left corner of rectangle into which the item is drawn	X	Obj\Num; Adjustable; Default value:0
Y-Coordinate Y-coordinate of top-left corner of rectangle into which the item is drawn	Y	Obj\Num; Adjustable; Default value:0
Width Width of rectangle into which the item is drawn	W	Obj\Num; Adjustable; Default value:100
Height Height of rectangle into which the item is drawn	H	Obj\Num; Adjustable; Default value:20
File The file name of the .WMF file to drawn in the rectangle	F	Obj\File; Adjustable; Default value:
Enable If set to 0, the item is not drawn	E	Obj\NoYes; Adjustable; Default value:1

Related Modules

BMP-File, *Icon-File*

Object Specifications

Once some ObVerse starts, its processor object becomes available within the top-level object of the device. As with all North objects, this object contains sub-objects, each of which corresponds with a public property defined within the ObVerse. It is possible to navigate around the objects using the ObSys Engineering Software.

Device Top-Level Objects

When ObVerse is started, and the ObServer channel is less than channel 100, the object below becomes available within ObSys, and its contents can be scanned. For example, if ObVerse is started with ObServer Channel 23, then object reference 'S23' becomes available.

When ObVerse is started with an ObServer Channel 100 or greater, the object becomes available, but does not get scanned.

Description	Reference	Type
Label The Label of the ObVerse running in the processor. ObVerse's text property with reference L is used as a label	*Px*	Variable Container: *Type depends on the file name of the ObVerse, but is contains objects listed in ObVerse Contents below [ObVerse\Various]*

ObVerse Contents

Object Type: *[ObVerse\Various]*

ObVerse contains a list of the public properties, which may be any of the types:

Description	Reference	Type
OffOn Label Label of OffOn object, with reference *r*	*r*	Obj\OffOn; Adjustable (if no module stores a value into the property)
NoYes Label Label of NoYes object, with reference *r*	*r*	Obj\NoYes; Adjustable (if no module stores a value into the property)
Num Label Label of Num object, with reference *r*	*r*	Obj\Num; In the range defined in the property; Adjustable (if no module stores a value into the property)
Float Label Label of Float object, with reference *r*	*r*	Obj\Float; In the range defined in the property; Adjustable (if no module stores a value into the property)
ENum Label Label of ENum object, with reference *r*	*r*	Obj\Text; In the range defined in the property; Alternatives defined in the property; Adjustable (if no module stores a value into the property)
Obj Label Label of Obj object, with reference *r*	*r*	Obj\Obj; Adjustable (if no module stores a value into the property)
Text Label Label of Text object, with reference *r*	*r*	Obj\Text; Max Length: 32; Adjustable (if no module stores a value into the property)
DateTime Label ADV Label of DateTime object, with reference *r*	*r*	Obj\DateTime; Adjustable (if no module stores a value into the property)
Times Label ADV Label of Times object, with reference *r*	*r*	Obj\Times; Number of periods defined in property; Adjustable (if no module stores a value into the property)
File Label ADV Label of File object, with reference *r*	*r*	Obj\File; Adjustable (if no module stores a value into the property)
WinClr Label ADV Label of WinClr object, with reference *r*	*r*	Obj\WinClr; Adjustable (if no module stores a value into the property)
WInFont Label ADV Label of WinFont object, with reference *r*	*r*	Obj\WinFont; Adjustable (if no module stores a value into the property)
Label ADV Label of Subroutine of ObVerse	*r*	[Various]

Note. ADV - Only available in Advanced ObVerse Processors

ObView Versions

Version	Build Date	Details
1.1.0.17	15/04/15	Added: Lead/Lag Module
1.1.0.19	07/03/2016	Updated Help Menu
1.1.0.20	17/03/2017	Added: ObjText Module
1.1.0.21	20/11/2018	Mod: MsgPriority 0/1 increased to speed comms
1.1.0.22	01/09/2019	Added <prefView>Footer.obv support Added ParentClass (PC) object to <prefView>Entry.obv Added SysInfo 202-204
1.1.0.24	01/08/2023	Moved to standard processor: Text-Equal, Text-In-Text, Text-Join, Text-Length, and Text-Split module types.
1.1.0.25	23/08/2024	Added Reserved Property WindowStyle = 8 – Menuless classic

Next Steps…

If you require help, contact support on 01273 694422 or visit *www.northbt.com/support*

North Building Technologies Ltd
+44 (0) 1273 694422
support@northbt.com
www.northbt.com

This document is subject to change without notice and does not represent any commitment by North Building Technologies Ltd.

ObSys and Commander are trademarks of North Building Technologies Ltd. All other trademarks are property of their respective owners.

Author: TM
Checked by: JF

Document issued 27/01/2025.