Proficy CIMPLICITY 11.1

Screens

Proprietary Notice

The information contained in this publication is believed to be accurate and reliable. However, General Electric Company assumes no responsibilities for any errors, omissions or inaccuracies. Information contained in the publication is subject to change without notice.

No part of this publication may be reproduced in any form, or stored in a database or retrieval system, or transmitted or distributed in any form by any means, electronic, mechanical photocopying, recording or otherwise, without the prior written permission of General Electric Company. Information contained herein is subject to change without notice.

© 2021, General Electric Company. All rights reserved.

Trademark Notices

GE, the GE Monogram, and Predix are either registered trademarks or trademarks of General Electric Company.

Microsoft® is a registered trademark of Microsoft Corporation, in the United States and/or other countries.

All other trademarks are the property of their respective owners.

We want to hear from you. If you have any comments, questions, or suggestions about our documentation, send them to the following email address:

doc@ge.com

Chapter 1. CimEdit Configuration..... 10

About CimEdit..... 10

CimEdit Features..... 10

Opening CimEdit Screens..... 12

Estimate the Number of Screens for a Project..... 13

CimEdit Configuration Overview..... 13

CimEdit Configuration Overview..... 13

Open CimEdit through the Workbench..... 13

Open an existing CimEdit Screen..... 14

Look over the CimEdit Window..... 15

Information Requirements for CimView..... 16

CimEdit Configuration Choices..... 19

CimEdit Screen Configuration Example..... 22

CimEdit/CimView Screen Management..... 27

Historian Connections..... 27

CimEdit Workspace Tools and Object Identifiers..... 28

CimEdit Workspace Tools and Object Identifiers..... 28

CimEdit Workspace Tools..... 28

Help for a CimEdit Screen or Object..... 48

CimEdit/CimView File Types and Search Paths..... 55

CimEdit Object Name and Description....62

Ambient Properties (Screen Configuration)....64

Add ActiveX and Other Objects....66

Translation....66

Preliminary Layout....68

Preliminary Layout....68

Available Objects....69

Object Form....126

Object Layout....142

Object Color....165

Object Shadow....212

Points, Variables and Expressions....217

Points, Variables and Expressions....217

Points in CimEdit....217

CimEdit Variables....252

Expressions in CimEdit....269

Linked Objects....328

Linked Objects....328

Basic Principles about Linking Objects....328

Linked Object Configuration....336

Find a Linked Object Source....344

Protected Link Source Objects..... 345

Runtime Movement and Animation..... 352

Runtime Movement and Animation..... 352

1. Object Movement.....353

2. Object Scaling.....359

3. Object Rotation.....363

4. Object Fill..... 368

5. Object Transparency.....373
6. Animated Picture Animation Speed..... 376
7. Attribute Animation.....381

8. Text Value Animation..... 398
9. Frame Animation..... 410

10. Copy/Paste Animation..... 425
11. Angle Animation.....427
Events, Procedures and Scripts.....431
Events, Procedures and Scripts.....431
CimEdit Events..... 432

CimEdit Procedures.....477

CimEdit Scripts..... 530

CimEdit Global Specifications.....558
CimEdit Global Specifications.....558

1. General Features that can be Shared Globally..... 560
2. Global Scripts Load List.....561

3. Before Context Menu/Before Editor Context Menu Global Events....565

4. Compatibility....571

5. Display Options....574

6. Undo Stack Size for a CimEdit Screen....575

7. Web HMI....576

ActiveX Controls....577

ActiveX Controls....577

Special Pasting Objects onto a CimEdit Screen....578

Insert an ActiveX Control on a CimEdit Screen....581

ActiveX Control's Properties....583

Other OLE Objects....585

Metafile Objects to CimEdit Objects....590

Text File Format Syntax....591

Text File Format Syntax....591

BNF Syntax....592

Token Syntax....617

CimEdit/CimView Screen Management....621

Chapter 2. CimLayout....650

About CimLayout....650

1. Open the CimLayout Configuration Editor....652

1. Open the CimLayout Configuration Editor....652

1.1. Open a New CimLayout Configuration Window....652

1.2. Open an Existing CimLayout Configuration....653
2. CimLayout Configuration Editor: Components....654
2. CimLayout Configuration Editor: Components....654
2.1 CimLayout Configuration Editor: Toolbar....655
2.2 CimLayout Configuration Editor: Menus....655
2.3. CimLayout Configuration Editor: Left Pane....659
2.4. CimLayout Configuration Editor: Right Pane....660
3. CimLayout: Monitor Selections....661
4. CimLayout: Container Type Configuration....661

4. CimLayout: Container Type Configuration....661
4.1. CimLayout: Screen Container Type....661
4.2. CimLayout: Splitter Container....665
4.3. CimLayout: Tab Container....678

5. CimLayout: Variables....681
Chapter 3. Navigation Configuration....683

About Navigation Configuration....683
1. Open the CIMPLICITY Navigation Configuration Editor....684
1. Open the CIMPLICITY Navigation Configuration Editor....684
1.1. Create a New Navigation Configuration File....685
1.2. Open an Existing Navigation Configuration File....688
2. Navigation Configuration Editor: Components....690
2. Navigation Configuration Editor: Components....690
2.1. Navigation Configuration Editor: Toolbar....691
2.2. Navigation Configuration Editor: Menus....691
2.3. Navigation Configuration Editor: Left Pane....695
2.4. Navigation Configuration Editor: Right Pane....697
3. Navigation: Global Properties....698
3. Navigation: Global Properties....698
3.1. Navigation: Default Color Properties....699
3.2. Navigation: Bar Display Properties....701

3.3. Navigation: Text Properties....704
3.4. Navigation: Icons on Buttons....706
3.5. Navigation: Force Default Variables....709
3.6. Navigation: Menu Translation....713
4. Add: Configuration Objects....717
4. Add: Configuration Objects....717
4.1. Add: Root Object....718
4.2. Add: Children Objects....721
4.3. Add: Quick Launch Objects....724

Add Configuration Object Dialog Box Options....728
5. Navigation: Child Properties and Variables....728
5. Navigation: Child Properties and Variables....728
5.1. Navigation: Required Variables....729
5.2. Navigation: Child Properties....729
5.3. Navigation: Additional Variables....732
6. Configuration Object: Properties....732
6. Configuration Object: Properties....732
6.1. Configuration Object: Color Properties....733
6.2. Configuration Object: Text Properties....739
6.3. Configuration Object: Image Property....741
6.4. Configuration Object: Inherit Permissions....743
6.5. Configuration Object: Inherit Variables....748
6.6. Configuration Object: Project Name to Qualify Points....753
6.7. Configuration Object: Screen Name....753
6.8. Configuration Object: Script Entry Point....755
7. Navigation: Runtime Tools....758
7. Navigation: Runtime Tools....758
7.1. Navigation Runtime: History....758
7.2. Navigation Runtime: Tree View....759
7.3. Navigation Runtime: Favorites....760
7.4. Navigation Runtime: Previous, Forward and Shift Buttons....763

7.5. Navigation Runtime: Display/Hide the Navigation Bar....763
Technical Reference....764
CimView Screen Navigation Error Messages....764
Chapter 4. CimView....766
About CimView....766
CimView Opened....766
CimView Opened....766
Open CimView on a Configuration Server or Viewer....767
Open CimView on a Runtime Server or Viewer....768
Typical Windows Application Tasks Available in CimView....768
Typical Windows Application Tasks Available in CimView....768
1. Open Other CimView Screens....769
2. Flush Cached Screens....770
3. Distribute Copies of CimView Screens and Files....770
4. Display the Toolbar, Menu and/or Status Bar....771
5. Go Back to Other CimView Screens....774
6. Zoom and Fit a CimView Screen or Window....774
CIMPLICITY Runtime Features in CimView....778
CIMPLICITY Runtime Features in CimView....778
Open the Point Control Panel through CimView....779
Open Quick Trends through CimView....780
Use Expression View in CimView....782

Use Point View in CimView..... 782
Review Screen Object Properties during Runtime..... 784

Set Points or Variables..... 786

Set Points or Variables..... 786

Execute Slider Setpoints..... 786

Enter a Value for a Setpoint..... 788

Enter a Variable Assign Value..... 789

Help for a Selected Object..... 789

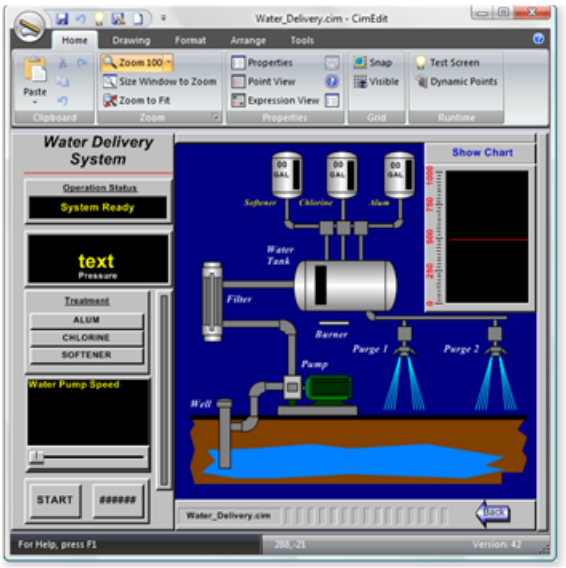
Standard Windows Features..... 790

Chapter 5. About CIMPLICITY Screens..... 792

Chapter 1. CimEdit Configuration

About CimEdit

With CimEdit, you can use rich animation and interactive controls to create or edit simple or complex custom screens that allow you to monitor and control your business processes and manufacturing assets. For example, the custom-designed graphic below shows the process and layout of a water delivery system. After you create and save a screen, you can view and test it in a runtime environment using CimView.



If you are also using Web HMI, you can export the screens to a folder from which you can import, and then display, them in the Web HMI application. You can use the default WebExport folder under the Project/Screens folder, or you can change the path and folder name. For information on exporting CimEdit screens as Web HMI mimics, see the Integrating CIMPLICITY with Web HMI section. For information on importing screens into Web HMI, refer to the Web HMI help system.

CimEdit Features

CimEdit includes the following features:

Interactive, dynamic point configuration allows you to add or modify point configuration data from anywhere within CimEdit.

A Point Browser dialog box lets you access any CIMPLICITY point on the entire network, and use the point to animate an object.

Standard Object shapes such as squares and rectangles, circles and ellipses, lines, polylines, arcs, pies, and cords are included.. You can define position, style, color, and rotation attributes, among others.

ActiveX embedded objects give your screens more power. Trend charts, spreadsheet charts, multimedia presentations, and live motion video can all provide power and extensibility to your system. OLE and ActiveX in-place editing of embedded objects allows you to view your screen as a single document without popping up other application windows. OLE and ActiveX drag-and-drop support means you can just drag an OLE or ActiveX object from one document to another. Drag Excel charts into CimEdit. Drag objects from one CimEdit screen to another.

A **Drag and Drop Library** of over 2000 Symbols and SmartObject™ objects makes creating screens a snap. The CIMPLICITY Object Explorer allows you to easily drag and drop the symbols and SmartObjects into the screens you are creating.

You can also add to the library by creating your own set of SmartObjects. SmartObjects are easily created with standard CIMPLICITY objects through Group Editing and Expression Variables. Group Editing provides the ability to edit properties of objects within a group without ungrouping the objects. Expression Variables provide the ability to use variables anywhere an expression or point can be used. A variable can be replaced with either a string or numeric value. The substitution of a variable can take place at either edit time or at run time.

Movement and **Rotation** are two animations that can be performed on objects.

Filled Objects including fill from top, bottom, left, right, or bi-directional. Bi-directional fill is a unique feature that allows you to configure a single object, which can fill in two directions from a center point. This is ideal for bipolar bar graphs.

Interior and Border Animation provides you the ability to animate the internal and outline aspects of objects. Interior Animation allows for color and pattern changes. Border Animation changes the line surrounding the object.

Events can be configured to handle

- Expression High, Expression Update
- Key Down, Key Up
- Mouse Down, Mouse Up
- Object Inserted, Object Removed
- Periodic
- Screen Close, Screen Open
- SmartObject
- While Key Down

- While Mouse Down

Frame Animation is a compound object that allows a series of frames to be defined. Each frame can consist of different objects. A particular frame is displayed based on the value of an expression. This allows areas of the screen to change like a filmstrip.

Hold Last Value can be defined on a project basis. This feature allows you to configure the system to hold the last known values of points in your CimView screens if the points go into an Unavailable State. Text points in this state will display in a configured color (determined for the entire project).

Metafile Import allows Windows Metafile objects produced by programs such as AutoCad and PowerPoint to be cut and pasted into CimEdit. The imported images can be decomposed into CIMPLICITY objects and can be fully animated. This is in contrast to bitmap imports that remain as single static objects.

Point Search & Replace allows you to search the screen for a point and highlight all objects that contain the point. You can then replace point identifiers within a CimEdit screen by simply typing over the name in a list of points used in the screen.

Scalable Objects provides the ability to change the size of an object based on the value of a point. The object can be scaled independently in the X and Y directions.

Scripting allows Basic Control Engine Scripts to be run from a CimView procedure.

Undo/Redo allows you to undo and redo a series of modifications to graphic screens.

Visibility Animation allows an object's visibility to be controlled by an expression. If an object is invisible, it cannot be selected.

Online Help provides comprehensive, indexed documentation, which is just a keystroke away at any time.

Object Help can be configured for any object on the screen. The operator can then access this help at any time using the right mouse button.

Dynamic Screen Testing allows you to test screen-editing changes in CimEdit without changing your original screen. By using the Test button on the standard menu bar, you can automatically start a CimView window to view your edits without committing to them.

Export Capability to Web HMI allows you to export a screen you created in CIMEdit and its compatible functionality, to Web HMI.

Opening CimEdit Screens

To open CimEdit, do one of the following:

- Click the Start > All Programs > CimEdit. Click the CimEdit button , then select New Window to create a new screen or select a screen for editing.
- From the Workbench, click File > New > Object to create a new screen.
- From the Workbench, click Edit > Properties to edit an existing screen.

Estimate the Number of Screens for a Project

When you have determined your information requirements, you can estimate the number of screens that need to be designed. The number will be influenced by the:

- Anticipated number of viewer destinations that require different screens
- Type and number of monitoring and regulating tasks that can be divided into logical units
- Amount of information to be displayed, keeping in mind that displays need to be clear to view in a runtime environment

Using frames is another layout option if you have several objects on a screen that will change during runtime, based on a similar set of conditions.

CimEdit Configuration Overview

CimEdit Configuration Overview

CimEdit combines the features commonly found in high-powered graphics applications, with an abundant number of state of the art configuration tools. They all help you take advantage of CIMPLICITY's extensive runtime capabilities. Consequently, you can create CimView screens that are clear, easy, and robust.

- Look over the CimEdit window.
- Information requirements for CimView.
- CimEdit configuration choices.
- CimEdit screen configuration example.

Open CimEdit through the Workbench

The CIMPLICITY Workbench offers you familiar features (also found in Windows Explorer) for opening new and existing CimEdit screens.

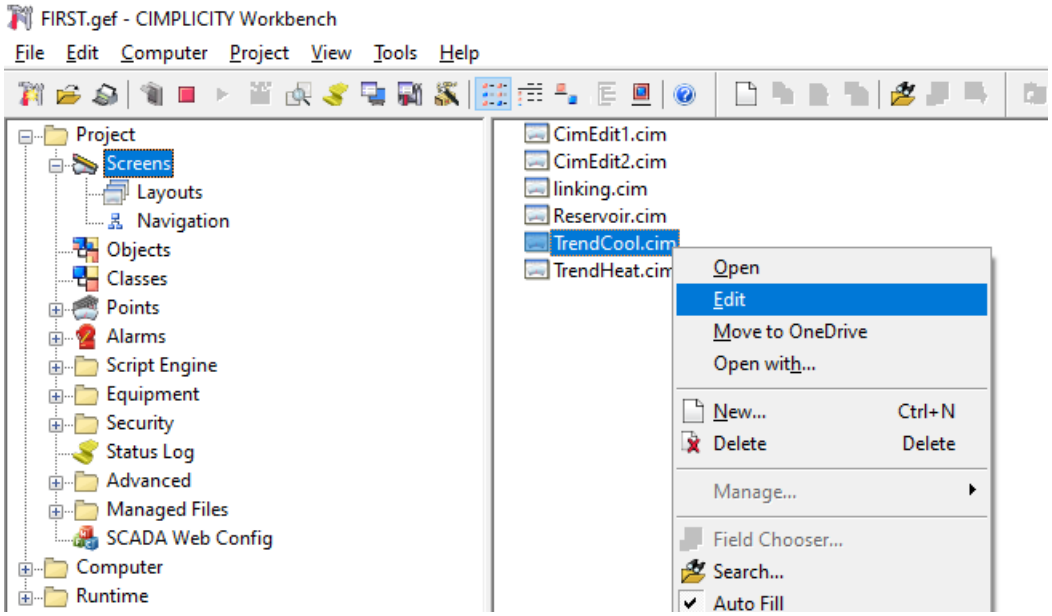
Option 1 *(page 12)*	Open a new CimEdit screen.
Option 2 *(page 14)*	Open an existing CimEdit screen.

Open an existing CimEdit Screen

- Through the Workbench
- In CimEdit

Through the Workbench

1. Select **Screens** in the Workbench left pane.
2. Right-click on a screen in the Workbench right pane.
3. Select **Edit** on the Popup menu.



Result: The CimEdit window you selected opens when you use any of the three methods.

Note:

To open a screen in CimEdit on double-click, set CimEdit as the default program to open .cim files.

In CimEdit:

a. Click the CimEdit button.
b. Select one of the following.
   • Open Window.
   • A recently opened screen. Result:

The selected screen opens in a new window.

Look over the CimEdit Window

As you configure your CimEdit screen, you will work on two or more levels.

The levels are as follows:

Level	Description
Screen *(page 28)*	Primary workspace in which you create the CimEdit/CimView configuration.
Frame container *(page 410)*	Two or more frames. Each frame contains groups and/or objects that display when the frame's pre-configured conditions evaluate to True.
Group *(page 143)*	A collection of several objects that are combined to act like a single object in certain instances.
Object *(page 142)*	Primary item with which you work.

Example

A CimEdit screen that displays the status of a kiln has the following tools and objects.

1	CimEdit button

2	Quick Access toolbar
3	CimEdit Ribbon bar
4	Frame
5	Group
6	Screen workspace
7	Text object
8	Status bar

In addition to providing you with high-powered graphics tools in its Ribbon bar, CimEdit provides you with easy to use dialog boxes. These dialog boxes are powerful enough for the most sophisticated programmers and easy enough for screen designers whose abilities lean more toward design.

Information Requirements for CimView

How you design your CimEdit configuration depends on your project's information requirements. Therefore, your planning begins with determining those requirements. The requirements include:

1 *(page 16)*	Choose data sources for CimView.
2 *(page 18)*	Determine user interaction with CimView screens.
3 *(page 18)*	Determine where runtime Information will be used.

Choose Data Sources for CimView

Information that CimView displays, calculates, and monitors can come from a wide variety of sources.

Sources include:

- Point data.
- Variables.
- Expressions.
- Other data sources.

Point Data

Point values *(page 218)* are at the core of CimView monitoring and regulation of processes. Points defined in CIMPLICITY's Point Configuration application are readily available whenever you need them in CimEdit. In addition, if you need to create new points you can easily open the Point Configuration application through CimEdit.

Device Point	Values from PLCs or other devices provide the CimView viewer with the ability to monitor a process represented by several points.
Virtual Points	Provide a flexible way to create calculated values.

Two common uses for a point are:

- Set points to affect a process.
- Alone or in expressions to display information about a process.

Variable Data

Variables *(page 252)* are flexible containers for information and do not add to your project's overhead. A variable has a unique name, a variable ID, and represents a value that can be assigned to it during configuration of the CimEdit screen or at runtime.

Some of the data sources it can represent are:

- Full Point ID
- Partial Point ID
- Text string in an expression.

A variable does not communicate with the PLC but is wholly contained in CimEdit.

Example of a Variable

Different values (types of values) are assigned to a variable ID **{var_value}** for different items on a CimEdit screen.

Variable ID	Value
var_level	**40**
var_level	**tank_level** (point)
var_level	**tank_level+50**
var_level	Assigned during runtime

In addition to providing you with several options for assigning values, variables can streamline your configuration time. For example, when you create an object that uses variable IDs, you can use the same object in several locations on a CimEdit screen, or on several different screens and assign different text strings to each instance of the object.

Expression Data

Expressions *(page 272)* provide a valuable and flexible way to help you evaluate, compare, and use the information gathered by points or variables or both.

An expression includes points, variables, or both along with any of the following CimEdit operations:

- Arithmetic
- Logical
- Alarm functions
- Bitwise
- Conversion
- Relational
- Scientific

Other Data Sources

CimEdit provides you with an easy to use Expression dialog box, in which you build complicated expressions with just a few clicks of the mouse.

Although points, variables, and expressions are sources for the most current information in CimView, you may need to view logged, historical, or other types of information. These values can come from vast number of sources, including:

- Logged files
- Text files
- Other database type files

Note: These data sources are available through features, such as Trending, or scripts.

Determine User Interaction with CimView Screens

CimView provides you with the framework to take full advantage of CIMPLICITY's powerful setpoint capabilities.

These capabilities include enabling CimView users to change device point values in a PLC to turn a machine on/off, open/close valves, or increase/decrease values to control how processes will function.

Determine where Runtime Information will be Used

If you have more than one CimView viewer, it is a good idea to lay out what information needs to be displayed and what type of user interaction is required at each location. The number and type of locations may be a major factor in how the screen design will be most effective.

For example, locations can be as disparate as being on:

- On one or more Viewers in the network.
- Remote, with a user accessing the CIMPLICITY project through PocketView on a handheld PC.

CimEdit Configuration Choices

CimEdit offers you a number of tools and features to help you design the CimView screens that will work most effectively in your organization. The choices you make depend, of course, on your project's information requirements. You also have the flexibility to cater to your own and the users' design preferences.

1 *(page 19)*	Estimate the number of screens for a project.
2 *(page 20)*	Select objects to convey CimView data.
3 *(page 20)*	Unleash the power of CimView.
4 *(page 21)*	Add ActiveX and other objects.
5 *(page 21)*	Trigger actions in CimView.
6 *(page 22)*	Monitor processes in CimVew.

Estimate the Number of Screens for a Project

When you have determined your information requirements, you can estimate the number of screens that need to be designed. The number will be influenced by the:

- Anticipated number of viewer destinations that require different screens
- Type and number of monitoring and regulating tasks that can be divided into logical units
- Amount of information to be displayed, keeping in mind that displays need to be clear to view in a runtime environment

Note: Using frames is another layout option if you have several objects on a screen that will change during runtime, based on a similar set of conditions.

Select Objects to Convey CimView Data

CimEdit provides you with a huge variety of objects to contain, display, and let the CimView user act on the data flowing to, from, and within it.

Objects include:

- Basic graphic shapes that you create.
- Text objects.
- Graphic images.
- Wide variety of SmartObjects from CimEdit's Object Explorer.
- ActiveX controls including the CIMPLICITY Alarm Viewer and Trend chart and OLE objects.

These objects may be a:

- Single element, such as a text message
- Pre-configured group of objects, such as most objects in CimEdit's Object Explorer
- Group of objects combined by you

CimEdit also provides you with a huge variety of tools as toolbar buttons and menu items to change the object's appearance. This includes changing its:

- Size.
- Shape.
- Angle of rotation.
- Color and fill.

Another time saving feature provided by CimEdit is the ability to create linked objects.

Linked objects *(page 328)* save you valuable time by providing you with the ability to change a single object, the linked object, and then let CIMPLICITY finish the job of updating every link to that linked object, in your entire project.

In addition, this single source capability insures that any specification or change made to the linked object will be reproduced exactly in every link to that linked object.

By mapping out the data flow, you can narrow down your choice of objects for each CimEdit task. Because each configuration is composed of a series of small steps, you can easily change your mind as you begin configuration.

Unleash the Power of CimView

When objects are laid out you are ready to make them work for you.

- Anything on the screen, including the screen, can be involved in actions that occur

- Within CimEdit, for example, another screen can be opened.
- A machine or process can be turned on and off or regulated.
- Objects can also be used to monitor plant processes, including:
- Normal operations
- Alarm states
- Trends

It is in this phase of configuration that you will involve various types of data. What you want each object to represent and do will determine which type you use.

You may have placed some of the objects on your CimEdit screen to promote one or more actions. The screen, itself, can also be involved.

Add ActiveX and Other Objects

Because CimEdit provides you with ActiveX objects, such as Trend and XY charts, as well as the ability to use such diverse items as entire CAD drawings you may need to do more configurations. Simply continue in the same systematic manner.

Trigger Actions in CimView

The process is straightforward, involving a:

Trigger

An event *(page 432)* triggers a procedure or calls a script. CimEdit provides a long list of events from which you can choose the best one for your requirements.

Result

A procedure *(page 477)* is one or more actions that are triggered in the specified order when an event occurs and while the screen is displayed in **CimView**. CimEdit provides several actions from which a screen designer can easily compile a meaningful list.

A script *(page 530)* , which is usually written by a system administrator, uses the same Editor and Basic language as the Basic Control Engine. Anything you can do in a normal script, you can do in a **CimEdit** script. CimEdit provides additional extensions to give you a wider range of screen development choices. However, **CimEdit** scripts are only accessible from the screen in which you create them.

Monitor Processes in CimView

CimEdit provides several choices to create activity on your screens that make if easy for a CimView user to quickly determine the status of a point or expression.

Items can:

- Move.
- Rotate.
- Change in size.
- Fill.
- Change color and text through animation.

Items can be:

- Groups.
- Objects.
- Text.
- Lines and shapes.

Each configuration involves a few simple arithmetic calculations. Combining or nesting activities requires only the simple configuration that you do for each.

Your options depend only on the requirements for your project, your ingenuity, and system resources.

CimEdit Screen Configuration Example

Following is a simple CimEdit configuration example. It begins with determining some information requirements for a process and continues with configuring screen for CimView that will address the information requirements.

1 *(page 23)*	Identify PLC's and points.
2 *(page 23)*	Determine any required user Interaction.
3 *(page 23)*	Determine the Information destinations.
4 *(page 24)*	Estimate what screens are needed.

5 *(page 24)*	Map out CimEdit representations to report the values.
6 *(page 25)*	Select and configure CimEdit objects to meet the requirements

1. Identify PLC's and Points

- There are three PLCs.

PLC	Connected to:
1	Tank 1.
2	A throttling valve regulating flow to Tank 1.
3	Tank 2.

- CIMPLICITY has the following configured points:
- Tank1_Level
- Tank1_Temp
- Tank1_Flow
- Valve_Piston
- Tank2_Flow

1. Determine any Required User Interaction

On Viewer 1

Users need to regulate the valve piston opening through PLC 2.

1. Determine the Information Destinations

There are two Viewers.

CimView users need to:

On Viewer 1

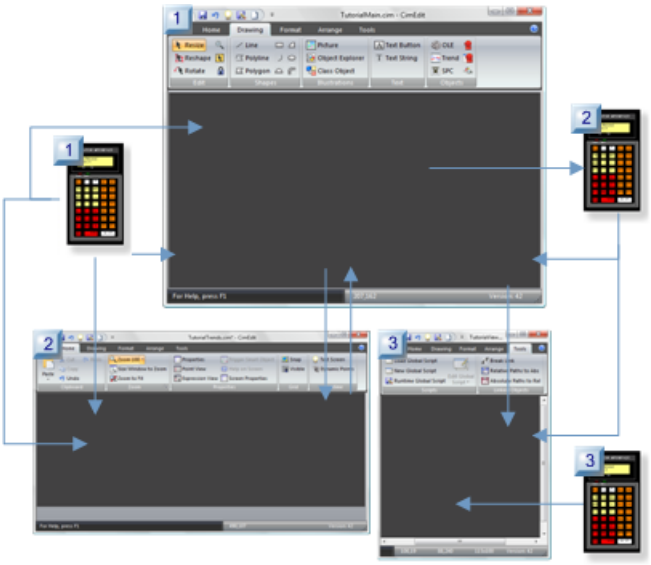
1. Monitor the current level of Tank 1.
2. Monitor the current temperature of Tank 1.
3. Monitor the rate of flow out of the Tank 1 valve (into Tank 1).

4. View a Tank 1 level trend.
5. View a Tank 1 temperature trend.

   On Viewer 3

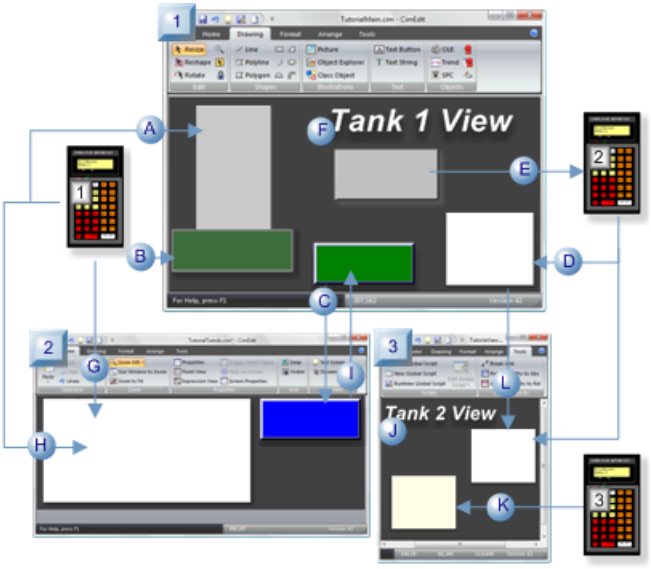
6. View the percent level in Tank 2.
7. View the rate of flow to Tank 1.
   a. Estimate what Screens are Needed

   There will be three screens.



Screen	Actions
1	• Display as the main screen on Viewer 1. • Receive device data from PLC 1 and PLC 2. • Send setpoint input to PLC 2.
2	• Display on Viewer 1. • Be opened through Screen 1. • Receive device data from PLC 1.
3	• Display on Viewer 2. • Receive data from Screen 1 (Tank 1 flow). • Receive device data from PLC 3 (Tank 2 flow)

   a. Map out CimEdit Representations to report the Values



Screen 1

	CimEdit Representation	Point
A	Fill	Tank1_Level
B	Attribute animation	Tank1_Temp
C	Mouse up event	Open Screen 2 action
D	Rotation/linked object source	Tank1_Flow
E	Setpoint/Movement	Valve_Piston (setpoint)
F	Linked source object	None

Screen 2

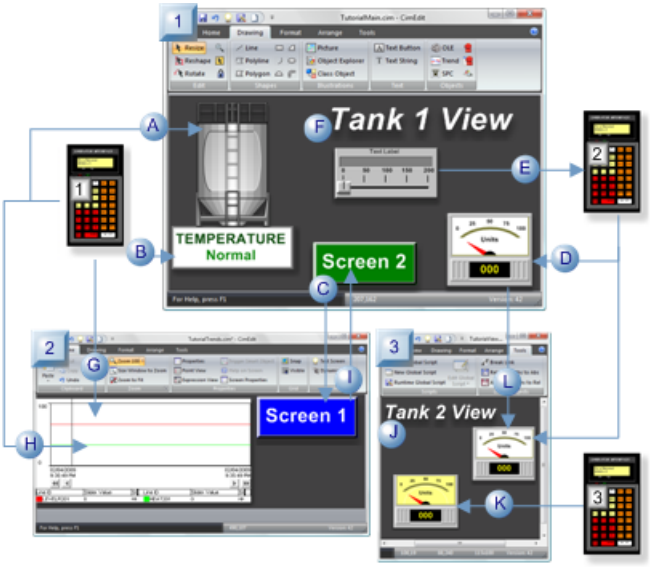
	CimEdit Representation	Point
G	Trend	Tank1_Temp
H	Trend	Tank1_Level
I	Mouse up event	Open Screen 1 action

Screen 3

	CimEdit Representation	Point
J	Linked container	Public variable
K	Rotation/copied object	Tank2_Flow
L	Rotation/linked container	Tank1_Flow

a. Select and Configure CimEdit Objects to Meet the Requirements

The objects are configured using some of CimEdit's powerful configuration features.



Screen 1

	CimEdit Object	CimEdit Representation	Point
A	Tank	Fill	Tank1_Level
B	Text	Attribute animation	Tank1_Temp
C	Button	Mouse up event	Open Screen 2 action
D	Gauge	Rotation/linked object source	Tank1_Flow
E	Slider	Setpoint/Movement	Valve_Piston (setpoint)
F	Text	Linked source object	None

Screen 2

	CimEdit Object	CimEdit Representation	Point
G	ActiveX control	Trend	Tank1_Temp
H	ActiveX control	Trend	Tank1_Level
I	Button	Mouse up event	Open Screen 1 action

Screen 3

	CimEdit Object	CimEdit Representation	Point
J	Text	Linked container	Public variable
K	Gauge	Rotation/copied object	Tank2_Flow

	CimEdit Object	CimEdit Representation	Point
L	Gauge	Rotation/linked container	Tank1_Flow

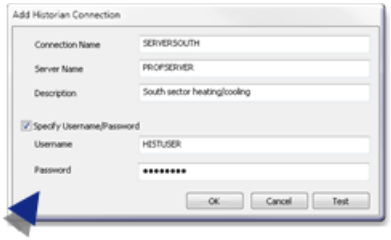
CimEdit/CimView Screen Management

- Advanced topics for the system administrator include:
- Screen performance issues.
- Create a CimView screen shortcut.
- Add arguments to the command line for a CimView shortcut.
- Command line arguments
- CimEdit/CimView GMMI_SCREENS Libraries.

Historian Connections

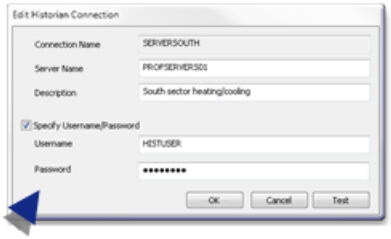
Add Historian Connection

Click to review details about adding an Historian connection to CIMPLICITY.



Edit Historian Connection

Click to review details about editing an Historian connection to CIMPLICITY.



CimEdit Workspace Tools and Object Identifiers

CimEdit Workspace Tools and Object Identifiers

CimEdit provides you with several tools to help you configure your work.

- CimEdit workspace tools.

You can also insure that the screen and objects are easily found and identified by including

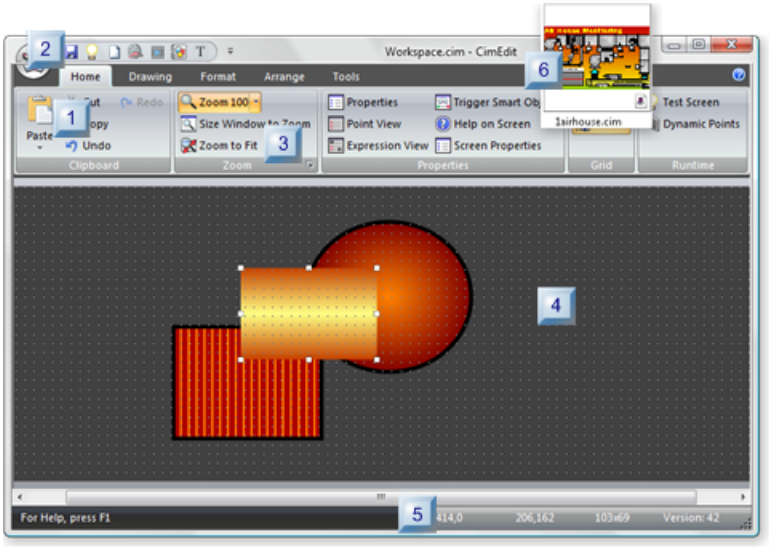
- Help for a CimEdit screen or object.
- CimEdit screen or object name
- CimEdit/CimView file types and search paths.

CimEdit Workspace Tools

CimEdit Workspace Tools

While you are working on your CimEdit screen you can choose several aids to help

They include:



rect 245, 305, 273, 330 5. Status Bar Information *(page 43)*
rect 332, 175, 363, 201 4. Horizontal and Vertical Grid *(page 42)*
rect 162, 75, 188, 99 3. CimEdit Screen Display Size *(page 37)*
rect 14, 18, 40, 42 2. CimEdit Quick Access Tools *(page 31)*
rect 29, 58, 55, 82 1. Ribbon Bar: Overview *(page 29)*
rect 325, 29, 356, 55 6. Screen Compile and Thumbnails for Viewers *(page 45)*

1 *(page 29)*	Ribbon bar overview
2 *(page 31)*	CimEdit quick access tools
3 *(page 37)*	Zoom the CimEdit screen display size
4 *(page 42)*	Horizontal and vertical grid
5 *(page 43)*	Status bar information
6 *(page 45)*	Screen compile and thumbnails for Viewers

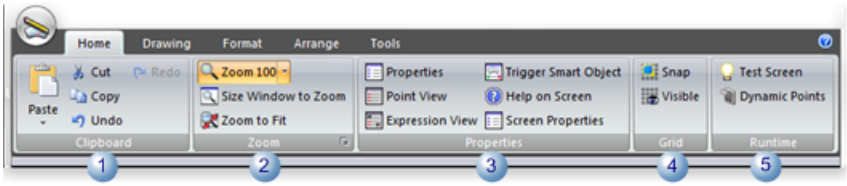
1. Ribbon Bar: Overview

The CimEdit Ribbon bar provides fast, easy to use access to CimEdit's numerous configuration tools.

- The Ribbon bar contains five tabs.

- Each tab displays logically grouped tools.
- The tools are described in detail throughout this documentation.

Home Tab

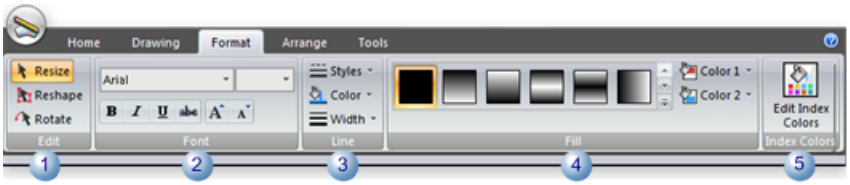


1	Clipboard
2	Zoom
3	Properties
4	Grid
5	Runtime

Drawing Tab

1	Edit
2	Shapes
3	Illustrations
4	Text
5	Objects

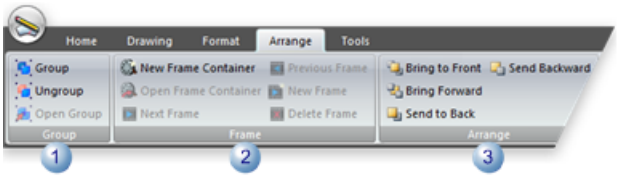
Format Tab

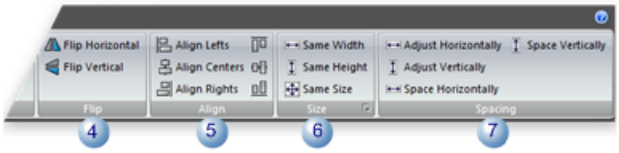


1	Edit
2	Font
3	Line
4	Fill

5	Index Colors

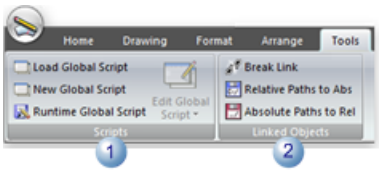
Arrange Tab





1	Group
2	Frame
3	Arrange
4	Flip
5	Align
6	Size
7	Spacing

Tools Tab



1	Scripts
2	Linked Objects

2. CimEdit Quick Access Tools

2. CimEdit Quick Access Tools

CimEdit provides you with a:

- Quick Access toolbar that you can customize for your convenience enabling you to quickly access frequently performed tasks
- Quick way to display or hide the Ribbon bar, as needed.

Note: The default Quick Access toolbar:

- Is located above the Ribbon bar.
- Displays the Save and Undo buttons

2.1 *(page 32)*	Quick Access Toolbar buttons
2.2 *(page 35)*	Quick Access Toolbar position and Ribbon bar display.
2.3 *(page 36)*	Quick Access: Default Keyboard Shortcuts

2.1. Quick Access: Toolbar Buttons

You can add and/or remove buttons to/from the Quick Access toolbar at anytime.

Buttons can be selected in either of two locations.

- All available buttons: Options Dialog box
- Ribbon bar buttons: Customize dialog box

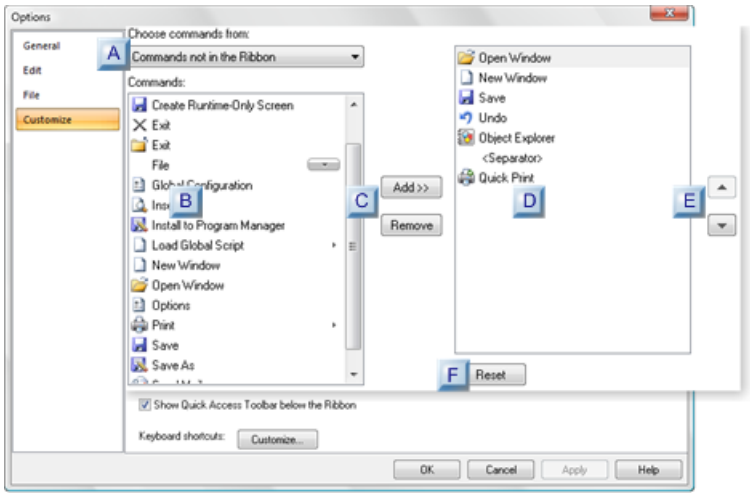
All Available Buttons: Options Dialog Box

1. Click the CimEdit button .
2. Click the Options button.

   The Options dialog box opens.

3. Select Customize.

   Options to choose or remove buttons are as follows.



rect 266, 216, 294, 242 *(page 34)*
rect 414, 108, 442, 134 *(page 33)*
rect 314, 111, 342, 137 *(page 33)*
rect 208, 109, 236, 135 *(page 33)*
rect 100, 108, 128, 134 *(page 33)*
rect 54, 17, 82, 43 *(page 33)*

A	Categories of buttons are listed in the **Choose commands from:** dropdown list. Categories are as follows.		
	Popular Commands		
	Commands not in the Ribbon		
	All Commands		
	File		
	Print Preview		
	Home		
	Drawing		
	Format		
	Arrange		
B	Available buttons based on the selected category.		
C	Add/Remove buttons add and remove buttons to/from the Quick Access toolbar.		
D	Buttons that will be included on the Quick Access toolbar.		
E	Up/down buttons change the button location on the Quick Access toolbar.		
	Button		
	Up		
	Down		

F	Reset button resets the toolbar to the two default buttons: Save and Undo. **Notes** • A message opens asking you to confirm the reset. • If you confirm the reset, you can click the Cancel button in the Options dialog box and revert to the toolbar you selected before the reset.

4. Click OK when you have completed configuring the toolbar.

   Result: The Quick Access toolbar displays the buttons you selected.

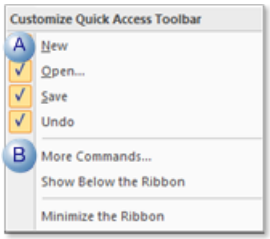
   Ribbon Bar Buttons: Customize Dialog Box

   Commands can be quickly added or removed to/from the Quick Access toolbar using the Quick Access toolbar Popup menu.

5. Left-click or right-click the Down arrow to the right of the Quick Access toolbar.

   Select one of the following on the Popup that opens.

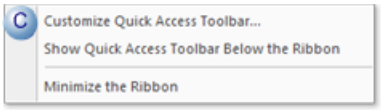
   Left-click Quick Access Popup

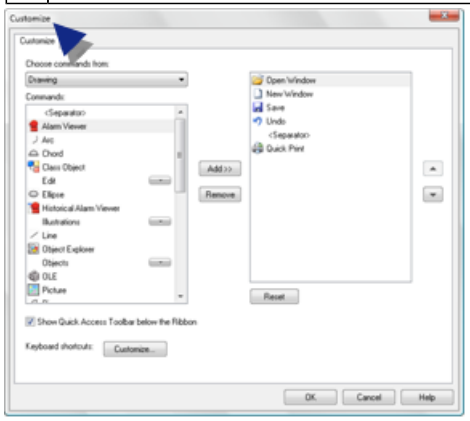
A	Standard buttons	
	On the Popup menu standard buttons can be quickly included on the Quick Access toolbar. Standard buttons are: • New • Open • Save • Undo	
	Check	
	Clear	
B	More Commands	

	Opens the Customize dialog box.

Right-click Quick Access Popup



C	Customize Quick Access Toolbar
	Opens the customize dialog box. The Customize Dialog box includes the following button categories. • File • Print Preview • Home • Drawing • Format • Arrange



6. Add, remove and position buttons the same way you do in the Options *(page 33)* dialog box.
7. Click OK when you have completed configuring the toolbar.

The Quick Access toolbar displays the buttons you selected.

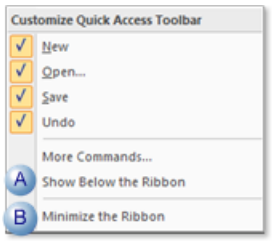
2.2. Quick Access: Toolbar Position and Ribbon Bar Display

You can select where to position the Quick Access toolbar and whether or not the Ribbon Bar should display.

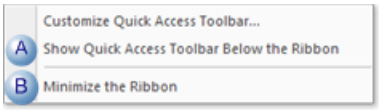
Left-click or right-click the Down arrow to the right of the Quick Access toolbar.

Select one of the following on the Popup that opens.

Left-click Quick Access Popup



Right-click Quick Access Popup



A	Check or clear the item that displays on the selected Popup menu. • Show Below the Ribbon • Show Quick Access Toolbar Below the Ribbon	
	Select	
	Check	
	Clear	
B	Check or clear Minimize the Ribbon	
	Select	
	Check	
	Clear	

2.3. Quick Access: Default Keyboard Shortcuts

1. Click Customize in the Options dialog box.
2. Click the Keyboard shortcuts Customize button.

   A Customize Keyboard dialog box opens.

3. Select a category in the **Categories** box.
4. Select a command in the **Commands** box.

The assigned shortcut key combination displays in the Current Keys box for commands that have shortcut keys.

Note: Currently only default shortcut key combinations will be available to use.

3. Zoom the CimEdit Screen Display Size

3. CimEdit Screen Display Size

CimEdit offers you several methods to magnify or reduce your CimEdit display size during configuration.

You can zoom a screen and fit the window to the zoom or change the window size and fit the screen to the window. Tools for changing the screen display and window size are as follows.

2.1 *(page 37)*	Quick CimEdit screen zoom tools
2.2 *(page 39)*	Precise CimEdit screen zoom tool.
2.3 *(page 40)*	Fit the CimEdit Window to Zoom
More *(page 254)*	CimEdit system variable: $ZOOM_LEVEL

Note: When the CimEdit screen display is magnified hold down the mouse middle button (wheel) and move the mouse in any direction to scroll (pan) the screen in that direction.

3.1. CimEdit Screen Quick Zoom Tools

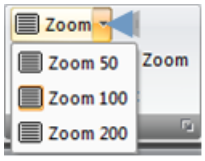
Quick zoom tools are the:

- Zoom button
- Mouse-click zoom

Zoom Button

The Zoom button in the Home>Zoom group provides three zoom percents.

1. Click the down arrow to the right of the Zoom button.



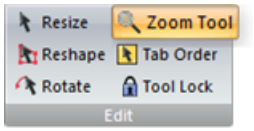
2. Select any of the three common zooms.
   - Zoom 50
   - Zoom 100
   - Zoom 200

   Result: The screen zooms to your selection.

   Note: When you re-open the CimEdit screen after you have used one of these buttons to zoom it, the CimEdit screen will re-open at the zoomed size.

   Mouse-click Zoom

3. Click Zoom Tool in the Drawing>Edit group on the CimEdit Ribbon bar.



   The cursor changes to a magnifying glass.

4. Choose any of the following to zoom the image in or out.
   - Click mouse-buttons, as follows.

Button	Result
Right	Enlarges the image by 5% increments.
Left	Decreases the image by 5% increments.

   - Use the mouse scroll wheel to zoom in and out.
   - Select an area to zoom.

   a. Hold down the left-mouse button to drag the cursor around an area

   A box surrounds the selected area.

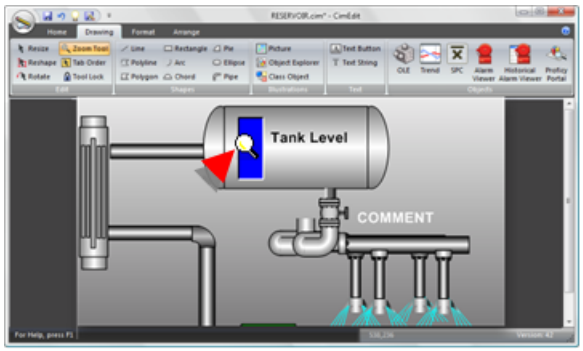
   a. Release the mouse button.

   The screen zooms to maximize the selected area display size.

   - Use keyboard keys, as follows.

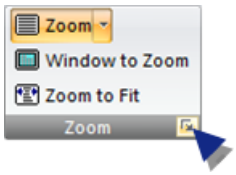
   Note: Keyboard zooms include both location and percent.

Keys	Result
Ctrl+K	Zooms to the next zoom in the zoom stack. Ctrl+K functionality is only available if you have already used Ctrl+J.
Ctrl+J	Zooms to the previous zoom.



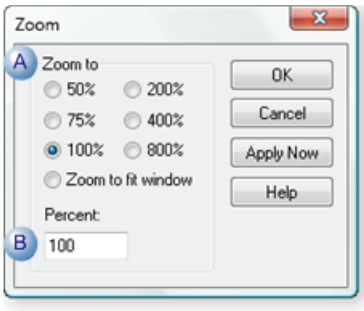
3.2. Precise CimEdit Screen Zoom Tool

1. Click the tab on the right of the Drawing>Zoom group bar on the CimEdit Ribbon bar.



   The Zoom dialog box opens.

2. The options are as follows.



	Option	Description
A	Zoom to	The screen will zoom to the checked percent.
B	Percent	Enter the precise percent of its size to which the screen should zoom. **Note:** CimEdit automatically fills this field if one of the Percent radio buttons is checked.

3. Click any of the following buttons.

Button	Description
OK	Close the dialog box and save the selections.
Apply Now	See the change before the dialog box is closed

Button	Description
Cancel	Cancel the changes

The selections take effect or are cancelled based on the button you click.

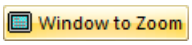
Note: When you re-open the CimEdit screen after using this zoom tool, it will re-open at a 100% zoom.

3.3. Window to Zoom/Zoom to Fit

You can fit the CimEdit window to the screen size or the screen size to the window.

- Fit window to zoom
- Zoom to fit window

Fit Window to Zoom

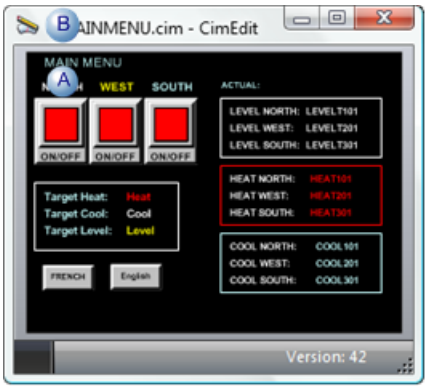
Click the Window to Zoom button 

in the Home>Zoom group on the CimEdit Ribbon bar.

Result: The CimEdit window immediately expands or contracts to fit the workspace display size.

Example

A CimEdit screen was much smaller than the window size. The window was zoomed to fit the screen.

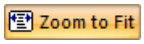


A	The screen maintains its proportion and size.
B	The window is zoomed to fit the screen. When the screen size is smaller than the CimEdit Ribbon bar, the CimEdit Ribbon bar is contracted or removed to accommodate the screen size.

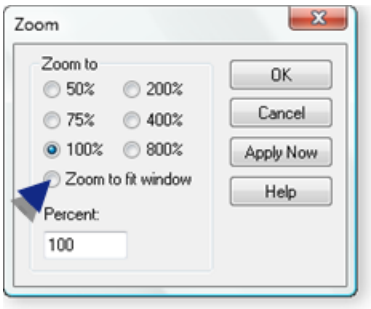
Zoom to Fit Window

If you make the CimEdit window larger or smaller you can zoom the screen to fit the window.

Do one of the following.

- Click the Zoom to Fit button 

in the Drawing>Zoom group on the CimEdit Ribbon bar.
- Check to fit window on in the Zoom dialog box *(page 39)*.



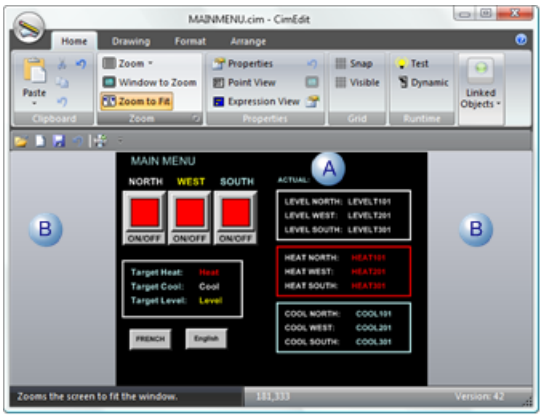
Result: If the window is smaller than the screen, the screen decreases to fit in the window.

If the window is larger, the screen increases to fit in the window.

In either case, screen retains its static dimensions so the fit is the best fit.

Example

A CimEdit screen has been zoomed to fit in its window.



A	The screen maintains its proportion when zoomed to its best fit in the CimEdit window.
B	The window (with a static length/height that is longer than the screen's) displays space around the screen

Note: When you re-open the CimEdit screen after using this zoom tool, it will re-open at a 100% zoom.

4. Horizontal and Vertical Grid

CimEdit provides you with the option to display a grid on your CimEdit screen and features of the display. The grid is one of many tools that help you place and align objects exactly where you want them.

- Detailed Grid Options
- Grid/Snap quick toggle

Detailed Grid Options

You can define the grid size and the default grid display in the Options dialog box.

1. Click the CimEdit button .

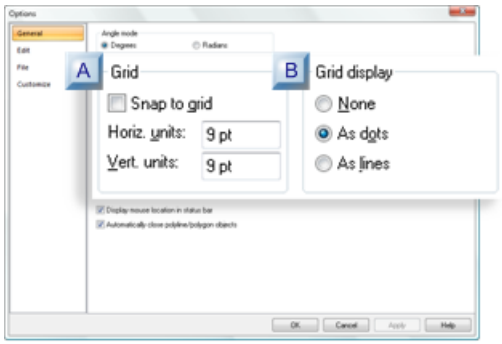
   The CimEdit menu opens.

2. Click the Options button.

   The Options dialog box opens.

3. Select General.

   Grid options are as follows.



rect 38, 28, 65, 56 *(page 42)*
rect 166, 27, 195, 55 *(page 43)*

A	Grid

Configuration for the grid includes the following.

Feature	Description	
Snap to grid	Checked	
	Clear	
Horiz. units	Horizontal space (points) between grid lines or dots.	
Vert. units	Vertical space (points) between grid lines or dots.	

B	Grid display

Check one of the Grid display options.

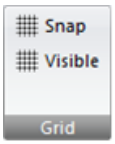
Radio button	Description
None	No grid displays.
As dots	Grid displays as dots.
As lines	Grid displays as dashes.

Note: Using your mouse is a convenient way to move an object from one location on the screen to another. However, the keyboard allows you to move an object by grid units, when you need to be more precise.

Grid/Snap Quick Toggle

You can quickly toggle Snap to grid and the grid on/off as follows.

Click one or both buttons in the Home>Grid group.

Grid Group	Button is:	Current Status	Click to:
	Highlighted	• Grid displays	Hide grid.
	• Snap to grid is enabled.	Disable snap.	
	Not highlighted	• Grid is hidden.	Display grid.

Note: Using your mouse is a convenient way to move an object from one location on the screen to another. However, the keyboard allows you to move an object by grid units, when you need to be more precise.

5. Status Bar Information

- Display mouse location on the screen.
- Status bar information.

Display mouse location on the screen

You can select to display the cursor coordinates in the CimEdit screen status bar.

1. Click the CimEdit button .

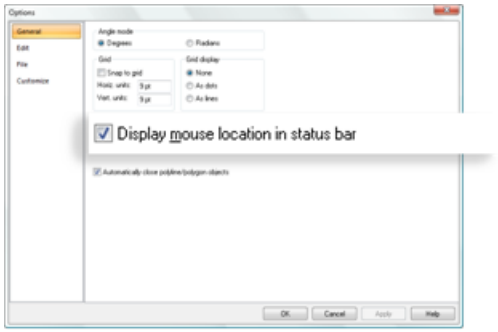
   The CimEdit menu opens.

2. Click the Options button.

   The Options dialog box opens.

3. Select General.

4. Check **Display mouse location in status bar**.



Status bar information

The Status bar provides the following information.

A	A phrase that describes the status or a suggested action based on what the cursor is over or what is selected on the screen.
B	The cursor location's screen coordinates, if this feature has been selected to display.
C	Selected object's left, top coordinates
D	Selected object's width, height
E	The version number reports what CIMPLICITY release the format of the CimEdit file was saved in. When the Version number is in red, e.g. , the CimEdit file you are working with was created in an earlier CIMPLICITY release. If you save the file, it will be upgraded to the CIMPLICITY version you are working in; the file format version number in the status bar will increase. As a result, the saved CimEdit file may not work if you try to use it on systems that still use an earlier version of CIMPLICITY. If you want to save the file, it is recommended that you save it to a different name to create a separate upgraded file.

6. Screen Compile and Thumbnails for Viewers

1. Click the CimEdit button .

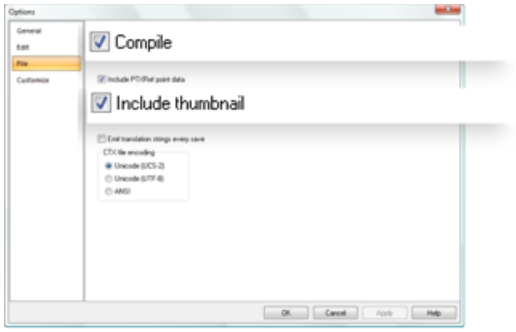
   The CimEdit menu opens.

2. Click the Options button.

   The Options dialog box opens.

3. Select File.

4. Compile and thumbnail options are as follows.



Option	Description
Compile	Checked
	Clear
	Default

Option	Description
Important: If you clear Compile, you will only be able to read the screen with CimEdit. Essentially you create an edit only screen.	
Option	Description
Include Thumbnail	Checked
	Clear
	Default

UI Elements

CimEdit Colors Tab

- Review screen fill and a border options.
- Configure fills and borders.

CimEdit Edit Options Dialog

- Stack size limit.
- Point qualification.

CimEdit File Options Tab

- Compile.
- Include thumbnail.
- Include PTXRef point data.
- Emit translation strings every save.

Note: Strings that are identified to be translated will be sent to the selected language mapper each time the CimEdit screen is saved. For more information on translation, see Translation *(page 66)*.

- CTX file encoding.

CimEdit General Options Tab

- Angle mode–Degrees or Radians.
- Configure the screen grid.
- Polyline/polygon objects.
- Enter Object Explorer directories.

CimEdit General Tab

- Name and description of a screen or object.
- Provide Help for a screen or object.
- View an object's tab order.

CimEdit Geometry Tab

Change the:

- Size of an object.
- Rotation angle of an object.
- Position of an object.

Make an object:

- Move during runtime.
- Scale during runtime.

Arc details

- Direction of arc, pie or chord object

CimEdit Procedures Tab

- Create procedures for the screen.
- Create procedures for groups and objects.

CimEdit Rotation/Fill Tab

- Object rotation during runtime.
- Object fill during runtime.

CimEdit Text Box

- Advanced actions.
- Enter a text string.
- Enter a text button string.

CimEdit Text Button Tab

- Text button: Detailed formatting
- Text value animation

CimEdit Text Tab

- Text object: Detailed formatting
- Text value animation

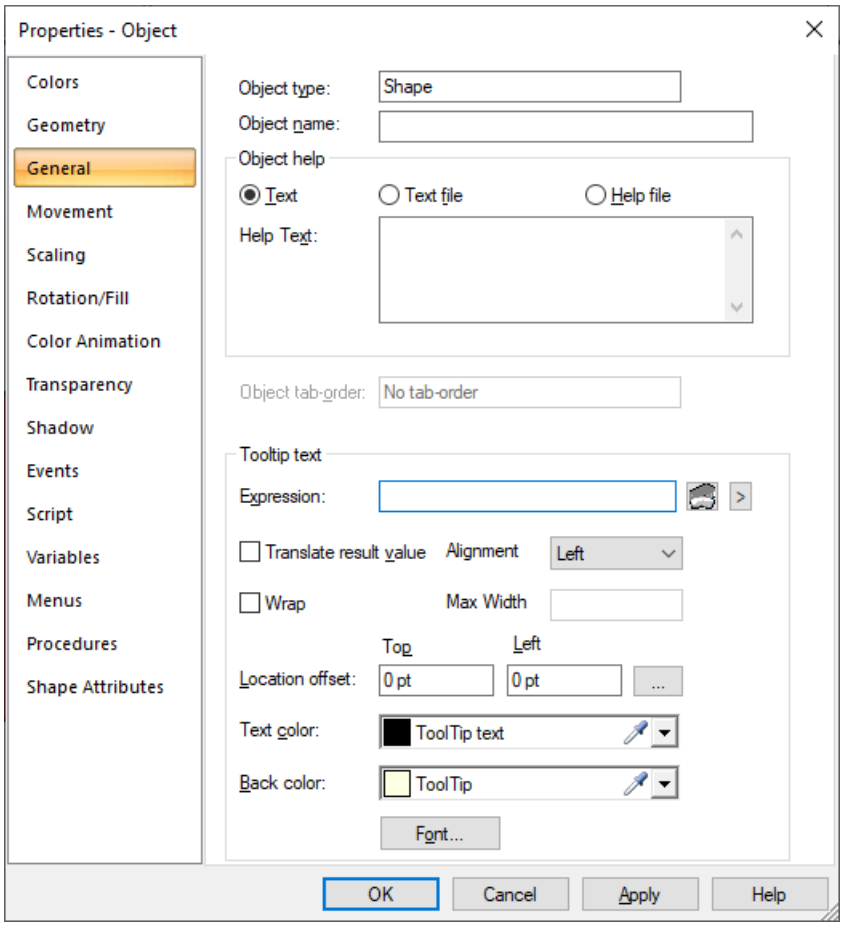
Help for a CimEdit Screen or Object

Create a Tooltip for a CimEdit Object

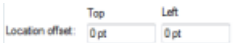
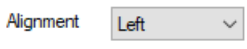
CimView allows you to create tooltip text for many objects on the screen. Tooltips are shown when you pause the mouse on top of an object without selecting it. Tooltips are not allowed for Windowed ActiveX or .net controls.

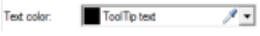
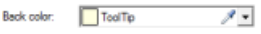
1. Select an object on the screen.
2. Right-click the object and select Properties.
3. Select the **General** tab.

   The Properties screen opens. The bottom portion of the screen is where you create a tooltip.



When creating a tootip in CimView, you can:

Component/ Option	Name	Description
	Expression	Enter an expression that evaluates to the text you want displayed in the tooltip.
	Expression Editor	Opens the Expression Editor *(page 272)*.
	More Popup	Enter operations, functions and calls to the tooltip. Click on a displayed expression to add it to the tooltip.
	Translate Result Value	Decide if the tooltip text and results of any expressions in the tooltip will be translated into the language set in CimView, and configured in the language mapper. For more information on translation, see Translation *(page 66)*.
	Location offset	Control where the tooltip appears in relation to the graphic object by enter X,Y coordinates. Use this field to keep tooltips from being obscured by other objects on the screen. The default location is to the top-left of the object.
	Alignment	Aligns the tooltip text towards the left, right, or center.

Component/ Option	Name	Description
	Wrap	Adjusts the tooltip text within the max width provided.
	Max Width	Enter the width within which the tooltip text will be adjusted. This is enabled only when the Wrap check box is selected.
	Offset Helper	Control the location of the tooltip by dragging its location rather than entering coordinates.
	Text Color	Set the color of the text.
	Back Color	Set the background color of the tooltip.
	Font	Open a screen with font choices, for example, font type, bold, italics, etc.

Help for a CimEdit Screen or Object

You can provide users with help at the screen, frame, group, and object levels.

Important: A user sees help that

You can:

- Designate a Help file for a screen or object.
- Create popup menu items for the screen or objects.

Designate a Help File for a Screen or Object

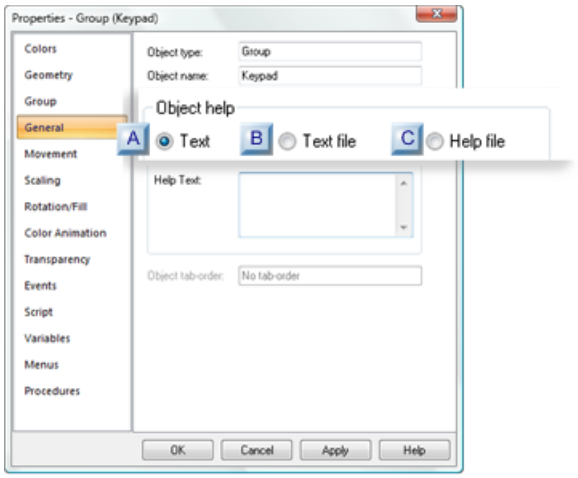
CimView supports three types of Help files that can be created for the screen or objects on the screen.

- Configure a custom help file
- Open a custom help file

Configure a Custom Help File

1. Open the Properties dialog box for a CimEdit object or screen.
2. Select the General tab.

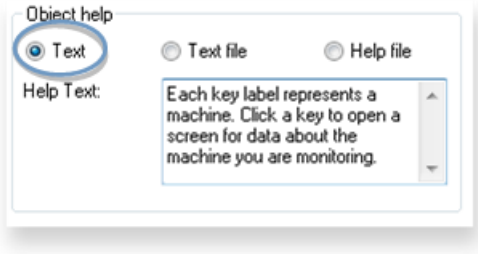
   The following help objects can be associated with the object or screen.



rect 64, 67, 98, 99 *(page 51)*
rect 142, 69, 176, 101 *(page 51)*
rect 233, 65, 267, 97 *(page 52)*

A *(page 51)*	Text
B *(page 51)*	Text file
C *(page 52)*	Help file

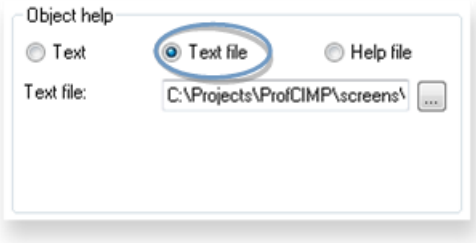
A	Text



Enter up to 30,000 characters in the scrolling field.

Result: The text you enter will display when help is selected.

B	Text file

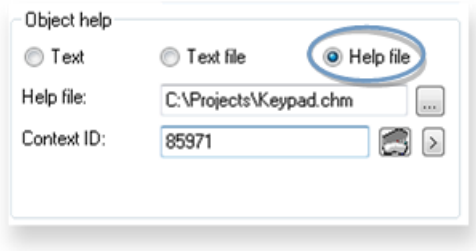


Enter the path and name of a text (.txt) file in the **Text file** field.

Note: Use the Browse button to find the file and enter it in the **Text file** field..

Result: The selected text file opens in Notepad.

C	Help file



3. Enter the name of a **Help file** generated by a Microsoft HELP compiler.

   File types include:

   - .hlp

   Note: Winhlp32.exe is required In order for .hlp files to work on Vista and Windows 2008. This file is not shipped as a Windows feature.

   You can download WinHlp32.exe from the Microsoft Website.

   - .chm

4. Enter the map number in the **Context ID** field for the Help topic you want to display.

   Result: The selected help file opens displaying the topic associated with the context ID..

   Tip: If you need to browse for a file:

5. Click the Browse button to the right of the Help file field.
6. Open the folder that has the file.
7. Do one of the following in the **File name** field.

.hlp file	Select the required .hlp file.
.chm file	Select the required .chm file.

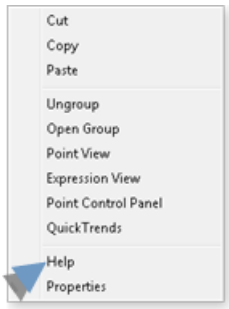
Open a Custom Help File

Note: If there is no custom help for your selection, Help is disabled on the Popup menu.

If help has been written, the following procedures will open it.

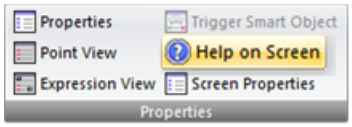
- CimEdit Object or Screen

8. Right-click a CimEdit/CimView object or screen.
9. Select Help on the Popup menu.



- CimEdit screen

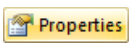
Click Help on Screen in the Home>Properties group on the CimEdit Ribbon bar.



The help for the selected object or screen opens.

Create Popup Menu Items for the Screen or Objects

All menu items currently defined for the screen or objects are listed in the list box along with the procedures or scripts they execute and any additional parameters.

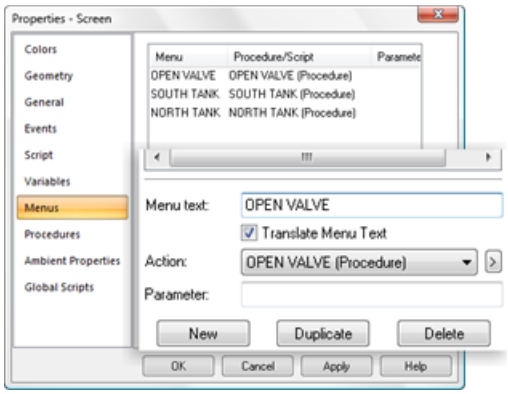
1. Select the screen or object that will display the Popup menu.
2. Do one of the following.
   - Click Properties 

in the Home>Properties group on the CimEdit Ribbon bar.
   - Right-click the selection; select Properties on the Popup menu.
   - Double-click an object (except SmartObjects) or group.

   The Properties dialog box for you selection opens.

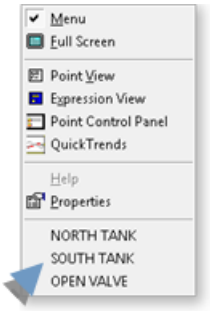
3. Select Menus.

   Options to configure Popup menu items are as follows.



Option	Description
Menu list	List of menu items that have been created and configured.
Menu text	**Important:** Enter text in the **Menu text** field the first time you select Menus for an object. This will enable the other fields. Text that will display when an authorized CimView user displays the CimView Popup menu.
Translate Menu Text	When checked, the menu text string becomes available for translation. For more information on translation, see Translation *(page 66)*.
Action	An existing or new procedure *(page 477)* or script *(page 530)* that will be executed when a CimView user selects the Popup menu item.
Parameter	Pass a string to the script if the procedure in the **Action** field invokes a script. The script must use the `CimGetEventContext().UserParameter` property to accept the parameter,
New	Click to create a new menu item
Duplicate	Click to duplicate a selected menu item. The menu text and configuration are duplicated. There cannot be duplicate text strings for the Popup menu entry so the text must be changed. The action and other selections can remain as duplicates.
Delete	Delete a selected menu item.

The configured menu items display when the Popup menu for the selected object or screen opens on a CimView screen.



CimEdit/CimView File Types and Search Paths

CimEdit/CimView File Types and Search Paths

CimEdit enables you to save screens in three formats.

Format	Type
.cim	Binary, editable.
.cimrt	Binary, protected.
.ctx	Text, editable.

CimEdit can also read files that are in **.asc** text format.

The format you choose determines:

- Whether the screen is protected or editable.
- When the screen is looked for in a CimEdit/CimView screen search sequence. CimEdit searches for the file when you instruct CimEdit/CimView to open a screen, e.g., through an Open Screen action or Overlay Screen action.

Decide the Format for a CimEdit Screen

Decide the Format for a CimEdit Screen

The two most common file formats you will work with are the binary formats:

- .cim
- .cimrt

In some instances you may want to save the file in a text format.

- .ctx

Note: CimEdit/CimView can read **.** **asc** files.

CimEdit Binary, Editable .cim Screen

The **.cim** is the binary CimEdit screen format that enables you and any designer who opens the screen to edit the configuration.

Do the following to save an editable *.cim file.

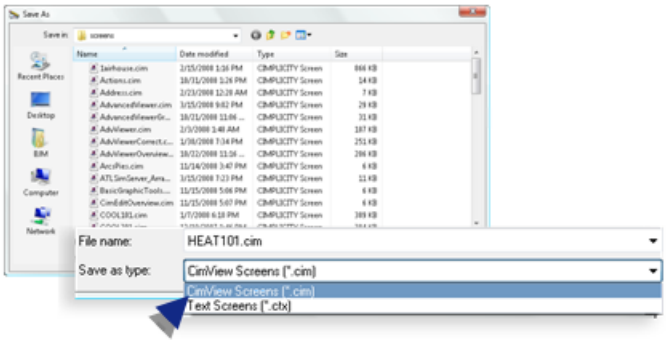
1. Do one of the following:

   - Click the CimEdit button ; select Save on the CimEdit menu.
   - (If selected for the Quick menu) Click the **Save** button

   The Save As dialog box opens prompting you for the file name and location.

2. Enter a name for the screen in the selected folder.

3. Select CimView Screens (*.cim) in the **Save as type** field.



4. Click **Save**.

The screen is saved as a binary, editable .cim file.

CimEdit Binary, Protected .cimrt Runtime-Only Screen

You can protect your CimEdit screen by saving it as a runtime-only screen. .

The runtime-only screen, which has a .cimrt extension, is a snapshot of the editable .cim screen. There is no information stored about the name or location of the source .cim screen.

- Runtime-only screen advantages
- Guidelines for runtime-only screen and objects
- Save a runtime-only screen

Runtime-only Screen Advantages

Runtime-only screens provide you with two major advantages.

You can create protected:

1. Screens that cannot be modified or decomposed.
2. SmartObjects that cannot be modified or decomposed. When a screen is saved in the .cimrt format, a designer can only link the named SmartObjects. This enables you to distribute the object for development purposes, while protecting your proprietary development information.

   guide: Guidelines for Runtime-only Screen and Objects

3. You can open a runtime-only screen using CimEdit. However, you cannot modify the screen.
4. You cannot modify an object on a runtime-only screen. The Properties dialog box for an object on a runtime-only screen contains a read-only General tab. It does not contain any other tabs.
5. You can select named objects on the runtime-only screen and copy them or drag them onto other screens. However, doing so does not move or copy the objects, it creates links to the objects on the runtime-only screen.
6. You can configure the appearance and potential runtime behavior (e.g. rotation and fill) of the linked container on an editable (.cim) screen.
7. The link container retains the properties of the source object. However, you cannot see code or configuration for the source object when you open the link container's Properties dialog box.
8. You cannot link unnamed objects from a runtime-only screen because you cannot create a link to an unnamed object.

   Warning: Keep the original .cim file because if it is lost, there is no way to recover the editable screen from the runtime screen.

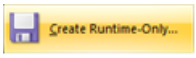
   The prompts you receive when you first save a screen as a runtime-only screen depend on the saved condition of the screen as follows:

   - The screen has never been saved,
   - A saved editable .cim screen has been modified, or
   - A saved editable .cim screen has been saved and not modified.

   Save a Runtime-only Screen

9. Click the CimEdit button .

The CimEdit menu opens.

10. Select Create Runtime-Only 

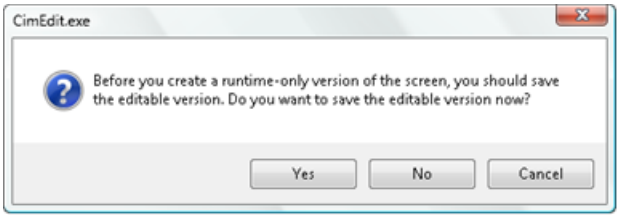
.

The remainder of procedure that you follow depends on the saved state of the screen.

A *(page 58)*	The screen has not been saved
B *(page 59)*	A saved .cim screen has been modified.
C *(page 59)*	A saved .cim screen has not been modified.

1. The screen has not been saved.

A message box opens advising you to save an editable version of the screen.

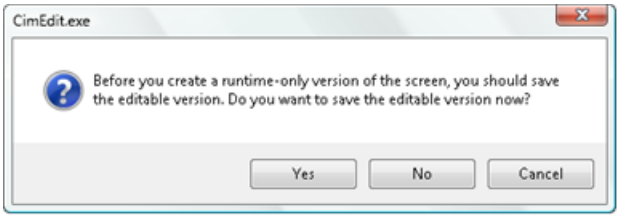


Click one of the following.

Button	Result
Yes	(Recommended) You must save a .cim screen if you want to edit your configuration. You are prompted for a name in a Save As dialog box. Result: • Two new files now exist in the folder you select. • One screen is an editable .cim or .ctx file. • The other screen is runtime only .cimrt file. • The editable CimEdit screen displays. If you make further changes, save both screens again.
No	You are prompted for a name in a Save Copy As dialog box. • The screen is saved as a runtime only .cimrt file with the name entered in the Save Copy As dialog box. • The editable .cim file is not saved.
Cancel	Cancels the runtime save and returns to the editable screen.

1. A saved .cim screen has been modified.

A message opens advising you to save an editable version of the screen.



Click one of the following.

Button	Result
Yes	• The modifications are saved for the cim (or ctx) and .cimrt file. • The editable CimEdit screen displays. If you make further changes, save both screens again.
No	The latest version is saved as a runtime-only screen with the same name as the .cim screen. The latest changes are not saved for the editable screen.
Cancel	Cancels the runtime save and returns to the editable screen.

1. A saved .cim screen has not been modified.

There is nothing further for you to do.

CimEdit saves the unmodified screen as .cim and .cimrt files.

Note: Only the **. cimrt** file needs to reside on the target machine. Save the .cim file where only authorized persons can open it.

CimEdit, Text, Editable .ctx Files

Text **.ctx** files are slower in operation than the binary **. cim** files. However, there may be times when you want to do a massive character search and replace, for example changing a **tr** in a Point ID to an **RT**. You can do this easily by saving the file as a **.ctx** file, opening it in a text editor such as Notepad and making your changes.

You save the **.ctx** file the same way you save a **.cim** *(page 56)* file, except you select the (**Text screens *.ctx)** file format option.

Select the .ctx file encoding

1. Click the CimEdit button .

   The CimEdit menu opens.

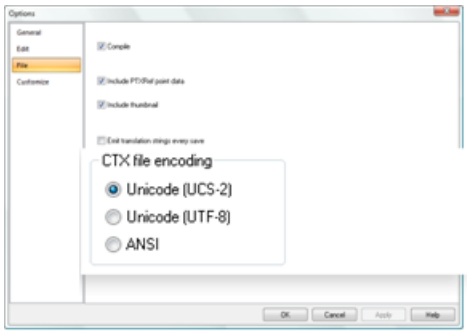
2. Click the Options button.

   The Options dialog box opens.

3. Select File.

4. Check one of the three encoding options.

   The options are:

   - Unicode (UCS-2)
   - Unicode (UTF-8)
   - ANSI

5. Click OK.

Ctx files will be saved in the selected encoding.

CimEdit/CimView Screen Search Sequence

CimEdit/CimView Screen Search Sequence

When an action, such as **Open Screen** , opens a screen that is identified with a CimEdit extension (e.g., **.cim**), in an absolute path, CimEdit/CimView opens that screen.

More frequently, commands for CimEdit/CimView to find another screen include a static path and may or may not provide a CimEdit extension to the file name. CimEdit/CimView follows a specific search sequence to find the file.

Factors affecting when CimEdit/CimView will look for the screen in the sequence include:

- Whether the screen from which the action is triggered is editable or protected.
- The directory in which the current screen resides (current directory).
- The file name extension that was entered for the screen to be opened.

CimEdit/CimView Screen Search from an Editable Screen

CimEdit/CimView goes through the following search sequence for another screen when the current screen file is in a **. cim** or **. ctx** format. When a match is found, CimEdit/CimView uses the screen file and stops the search..

CimEdit/CimView looks for:

1st	The exact specified file in the current directory.
2nd	A **. cim** or **. cimrt** extension in the current directory. If the specified extension is a CimEdit extension (.cim, .cimrt, .ctx or .asc), for its search, CimEdit substitutes in the following order: 1. **. cim** . 2. **. cimrt** . If the specified extension is not a CimEdit extension (.cim, .cimrt, .ctx or .asc), for its search, CimEdit appends in the following order: 1. **. cim** . 2. **. cimrt** .
3rd	A **. cim** or **. cimrt** extension in the directory path specified by the GMMI_SCREENS *(page 646)* library directories. If the specified extension is a CimEdit extension (.cim, .cimrt, .ctx or .asc), CimEdit substitutes: 1. **. cim** . 2. **. cimrt** . If the specified extension is not a CimEdit extension (.cim, .cimrt, .ctx or .asc), CimEdit appends: 1. **. cim** . 2. **. cimrt** .
4th	A **. ctx** extension, per the 2nd and 3rd search sequence.
5th	A **. asc** extension, per the 2nd and 3rd search sequence.

CimEdit/CimView Screen Search from a Protected Screen

CimEdit goes through the following search sequence for another screen when the current screen file is in a **. cimrt** format. When a match is found, CimEdit/CimView uses the screen file and stops the search.

CimEdit/CimView looks for:

1st	The exact specified file in the current directory.

2nd	A **.cimrt** or **.cim** extension in the current directory. If the specified extension is a CimEdit extension (.cim, .cimrt, .ctx or .asc), for its search, CimEdit substitutes in the following order: 1. **.cimrt** . 2. **.cim** . If the specified extension is not a CimEdit extension (.cim, .cimrt, .ctx or .asc), for its search, CimEdit appends in the following order: 1. **.cimrt** . 2. **.cim** .
3rd	A **.cimrt** or **.cim** extension in the directory path specified by the GMMI_SCREENS *(page 646)* library directories. If the specified extension is a CimEdit extension (.cim, .cimrt, .ctx or .asc), CimEdit substitutes: 1. **.cimrt** . 2. **.cim** . If the specified extension is not a CimEdit extension (.cim, .cimrt, .ctx or .asc), CimEdit appends: 1. **.cimrt.** 2. **.cim.**
4th	A **.ctx** extension, per the 2nd and 3rd search sequence.
5th	A **.asc** extension, per the 2nd and 3rd search sequence.

CimEdit Object Name and Description

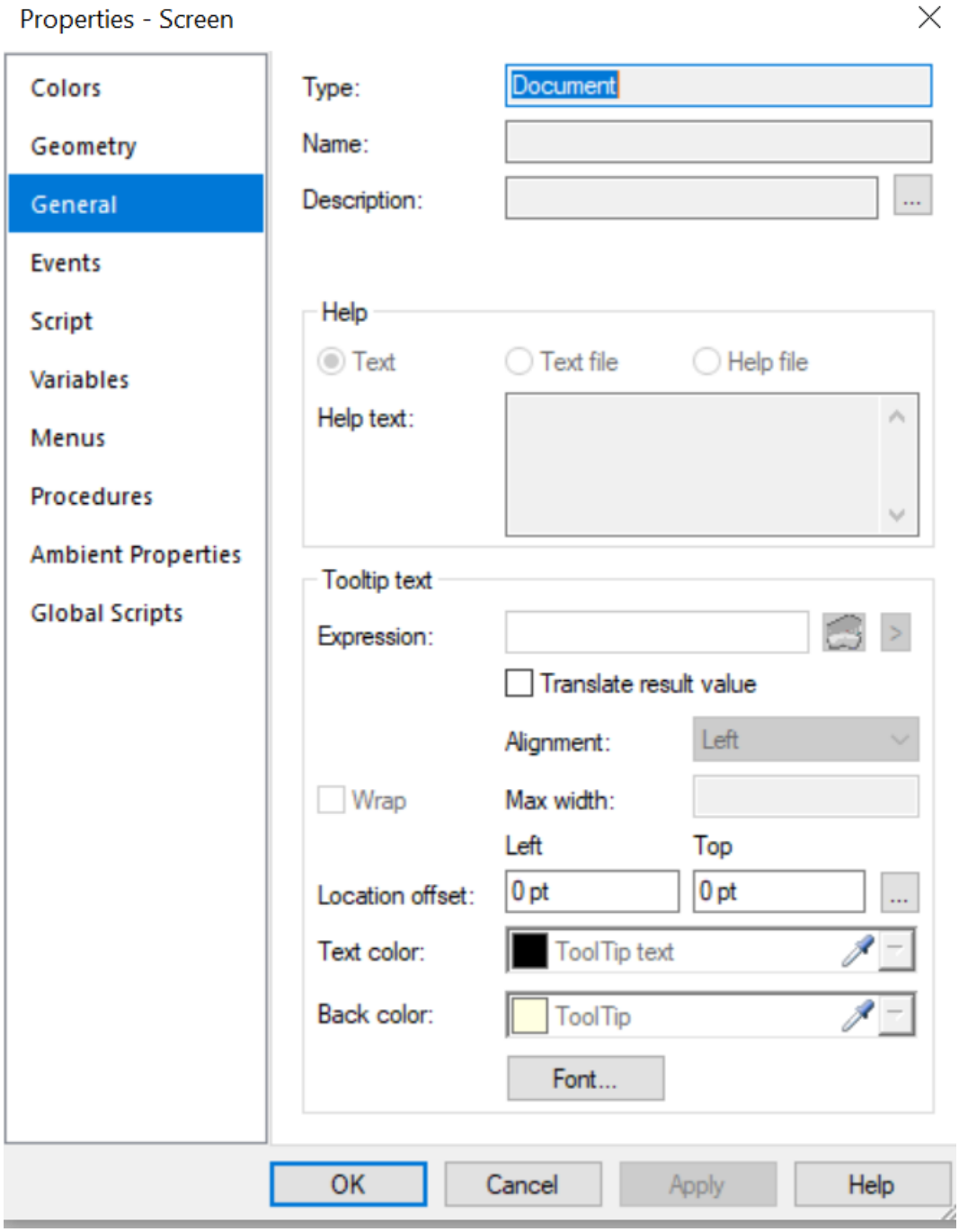
You can add the name and description of a CimEdit screen or object in its associated Properties dialog box.

Important: Objects need to be named for some scripting and procedure operations.

1. Open the Properties dialog box.
2. Select General.

   A read-only **Object type** field displays the object type, e.g. Document, Group, Object.

3. Enter a name in the **Object name** field.



a. Select ... .
b. Add the description.
c. Select **Close**.

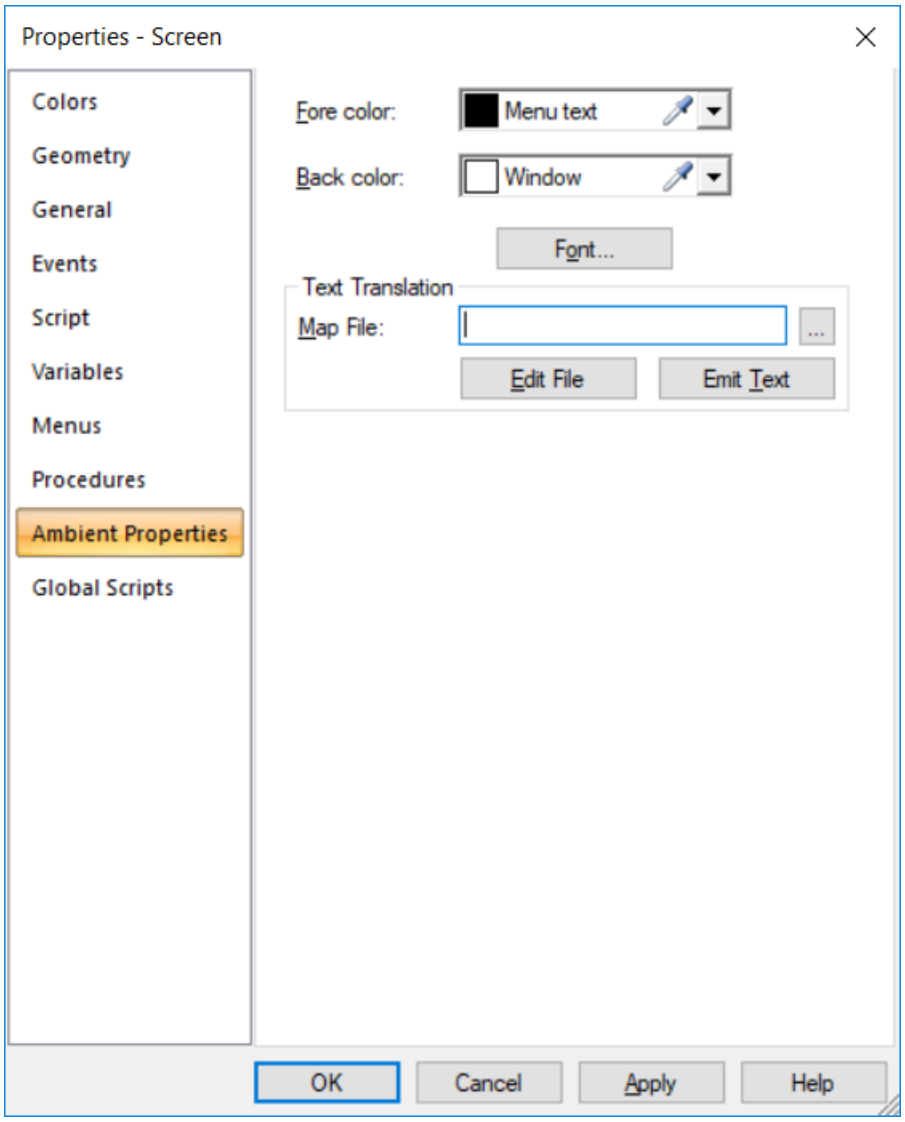
Ambient Properties (Screen Configuration)

When you insert ActiveX control objects into your screen, the default ambient properties used are defined at the screen level. Ambient properties are the window foreground color, background color, text font, and text translation. To access the ambient properties of a screen.

1. Place your cursor on the screen.
2. Do one of the following.
   - In the CimEdit Ribbon bar, select **Properties**.
   - Right-click the screen, and select **Properties**.

   The Properties - Screen dialog box opens.
3. Select Ambient Properties.

   The ambient properties color, font, and text translation options appear.



Option	Description
Fore color	Select the color for text or foreground lines. ActiveX controls (also called OLE controls, OCX controls) may let you override this color in the control property pages.
Back color	Select the background color of the control. ActiveX controls (also called OLE controls, OCX controls) may let you override this color in the control property pages.
Font	Select the font for text. ActiveX controls (also called OLE controls, OCX controls) may let you override this color in the control property pages.

Option	Description
Map File	Select the CIMPLICITY Language Map file to be translated. For more information on translation, see Translation *(page 66)*.
Emit Text	Write the translatable strings from the screen into the CIMPLICITY Language Map file.
Edit File	Edit the translated strings in the CIMPLICITY Language Mapper tool.

4. Click **OK** to activate the new defaults.
   The new default values are activated and will be reflected in all ActiveX controls currently embedded in the screen.

Add ActiveX and Other Objects

Because CimEdit provides you with ActiveX objects, such as Trend and XY charts, as well as the ability to use such diverse items as entire CAD drawings you may need to do more configurations. Simply continue in the same systematic manner.

Translation

CIMPLICITY allows you to translate your application into different languages and to switch the displayed language at run time. Use the CIMPLICITY Language Mapper to define the languages and translate the strings to other languages.

Translating CimEdit Screens

Text objects (including Text Button objects), message text, menu text, and tooltip text contain strings. Their Translate property determines whether the text is available for translation.

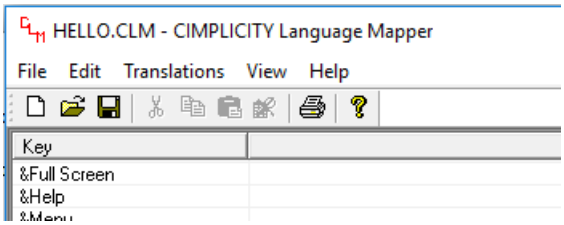
In the Ambient Properties section, when you select Emit Text, strings from your CimEdit screen will be made available for translation.

The following sections guide you through an example.

Create a CimEdit screen

1. Open a new CimEdit window.
2. Select Text String and create two text objects, Hello and everyone!.
3. Go to the properties of each of the objects and in the Text tab, select the Translate check box.
4. Save the screen as c:\temp\hello.cim.
5. Select Screen Properties, and then select the Ambient Properties tab.
6. In the Map Properties field, enter c:\temp\hello.clm.
7. Select Emit Text, and then select Edit File. The CIMPLICITY Language Mapper window appears.

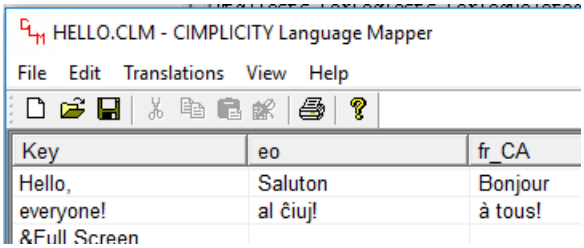
Note: If you select the Main Button, select Options, select the File tab, and then select the Emit translation strings every save check box, you do not have to select Emit Text.



Translate the screen

1. In the CIMPLICITY Language Mapper window, select Translations, and then select Language Editor.
2. In the Language Editor dialog box, select Add, enter eo as the Language with the default Locale ID, and then select OK.
3. In the Language Editor dialog box, select Add, enter fr_CA as the Language, select French (Canada) (0xc0c) as the Locale ID, and then select OK.
4. Select Done to exit the Language Editor.
5. Double-click the row containing the Hello key to open the Translation Editor.
6. Double-click the row containing the eo language, and enter Saluton,.
7. Double-click the row containing the fr_CA language, and enter Bonjour,.
8. Select Done to exit the Translation Editor.
9. Repeat steps 5 to 8 for the everyone! key, and enter al ĉiuj! in step 6 and à tous! in step 7.

Note: If the characters entered in the Translation Editor do not appear as they are, select Edit, and then select Select Font to select Arial or Calibri as the font.



10. Select Save, and exit the CIMPLICITY Language Mapper.

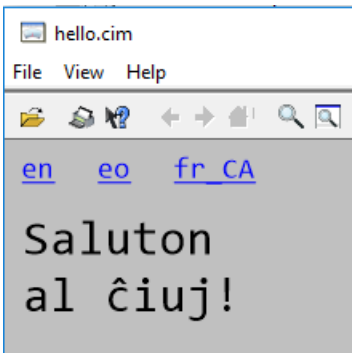
Add language switching controls

1. In the CimEdit screen, select Screen Properties.
2. In the Properties - Screen dialog box, select the Script tab, select Edit, and enter the following script:

```
Option Explicit
```

```
Sub SwitchLanguage(x As Long, y As Long, flags As Long)
 Dim newLang As String
 Dim newLangStr As String
 Dim curLangStr As String
 ' if current language is blank, that's the screen default
 curLangStr = CimGetScreen.Application.Language
 If curLangStr = "" Then curLangStr = "en"
 ' if the new language in English, use the screen default
 newLangStr = CimGetEventContext().TriggerObject.TextFormat.Text
 If newLangStr = "en" Then newLang = "" Else newLang = newLangStr
 ' switch the language
 MsgBox "Switching from " & curLangStr & " to " & newLangStr
 CimGetScreen.Application.Language = newLang
 CimGetScreen().Refresh False
End Sub
```

3. Select OK.
4. In the CimEdit screen, select Text String, and create a text object, en.
5. Select Properties, select the Colors tab, and under Fill, select Blue as the Color.
6. In the Properties - Object dialog box, select the Text tab, select Font, select the Underline check box, and then select OK.
7. In the Properties - Object dialog box, select the Events tab, select Mouse Up as the Event, select SwitchLanguage (Script) as the Action, and then select OK.
8. Copy and paste the en text object twice, name them eo and fr_CA, and save the screen.
9. Select Test Screen. The hello.cim screen appears.
10. Select the eo and fr_CA text objects to view the translated text.



Preliminary Layout

Preliminary Layout

CimEdit offers you a wide assortment of objects *(page 69)* and object types to place on your CimEdit screen.

You can place objects that deal with data from any source you specify and display the data or evaluation results in a manner that is most effective for your project's runtime requirements.

- Available objects.
- Object form.
- Object layout
- Object color in CimEdit.
- Object shadow.

Tip: Place several of the objects you think you need before you start in-depth configuration. This way you can easily make changes in your choice of objects as you experiment with layout. This will also help you decide if you want to use the same object in more than one place on the same screen or on other screens. If that is the case, there are shortcuts available to you.

In CimEdit, you can easily create multiple copies of objects you create and configure. You can place the additional copies in different locations on the same screen or different screens.

1. The most powerful method is to create as many link containers as you need from a linked object *(page 328)*.

   Link containers created from a linked object initially contain all the properties of the linked object. In addition, when a component designer updates a linked object, all of its link containers are automatically updated. A basic method is to copy an object from one screen to another, leaving the original object on the first screen.

2. If you want to change the location of an object, you can simply cut it from one screen and paste it on another.

   While you are determining what objects to use, take advantage of CimEdit's tools to experiment with the layout by moving the objects around, flipping them and rotating them.

Available Objects

Available Objects

CimEdit provides you with several types of objects to deal with the different requirements you will have as you design your screens.

- Available objects include:
- Basic graphic objects
- Object Explorer
- Text objects
- Graphic images
- Class object graphics.
- ActiveX and OLE controls.

Basic Graphic Objects

Basic Graphic Objects

As you design your CimEdit screen, you will probably create one or more basic shapes for any number of reasons.

Basic shapes include:

rect 2, 1, 61, 34 *(page 71)*	Straight Lines
rect 2, 1, 61, 34 *(page 71)*	Polylines
rect 2, 2, 64, 36 *(page 73)*	Rectangles (square corners)
rect 5, 3, 63, 36 *(page 72)*	Rectangles (round corners)
rect 2, 1, 61, 34 *(page 74)*	Polygons
rect 2, 1, 61, 34 Basic Objects: Ellipses *(page 75)*	Ellipses
rect 2, 1, 61, 34 Basic Objects: Arc Lines, Chords and Pies *(page 76)*	Arcs, chords and pies.
rect 2, 1, 61, 34 Basic Objects: Pipe Tool *(page 79)*	Pipes

CimEdit offers several tools *(page 84)* to speed up the process.

Basic Objects: Straight Lines

- Straight lines.

- Polylines

Straight lines

1. Click the Line button in the Drawing tab>Shapes group on the CimEdit Ribbon bar.

   Click Tools>Line on the menu bar.

   The cursor changes to a crossbar.

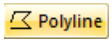
2. Move the cursor to the starting position for the line.
3. Hold down the left mouse key and drag the mouse to the ending location for the line.

4. Release the mouse button.

   Result: A straight line displays on the CimEdit screen surrounded by a Selected box. Handles are available for moving and re sizing the object.

   Polylines

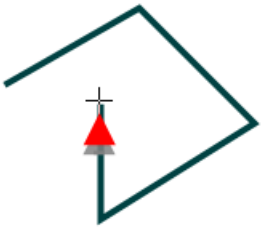
   A polyline can have as many sides as you need.

5. Click the Polyline button  in the Drawing tab>Shapes group on the CimEdit Ribbon bar.

   The cursor changes to a crosshair.

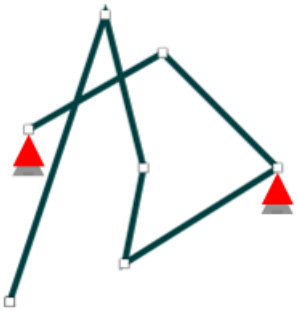
6. Move the cursor to the starting position for the object.
7. Click the left mouse button to set the first point.
8. Move the cursor to the next point you want to set.

   As you move the cursor, a line will be drawn between the last point you set and the cursor.



5. Click the left mouse button to set the next point.
6. Repeat 4 and 5 until you reach the last vertex.
7. At the last vertex, double-click the left mouse button to finish drawing your polyline.

Result: A polyline in the shape you designed displays on the CimEdit screen surrounded by a Selected box. Handles are available (at each location where you clicked the mouse button) for moving and resizing the object.



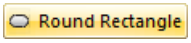
Note: If you attach the end of last line to the beginning end of the first line when you finish the polyline, you will create a closed polygon. You can choose to have CimEdit automatically close the polyline/polygon objects.

9. Click the CimEdit button
10. Select Options.
11. Check **Automatically close polyline/polygon objects**.

Basic Objects: Rectangles, Polygons

- Rectangles (Rounded corners)
- Rectangles (Square corners)
- Polygons

Rectangles (Round Corners)

1. Click the Round Rectangle button 

in the Drawing tab>Shapes group on the CimEdit Ribbon bar..

   The cursor changes to a crossbar.

2. Move the cursor to the starting position for the object.

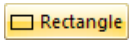
3. Hold down the left mouse button; drag the crossbar cursor to the ending location for the rectangle.

4. Release the mouse button.

   Result: A rectangle that represents the size you drew appears. Handles are available for moving and resizing the object.

Tip: When you click the **Rectangle** button, then click the left mouse button a duplicate of the last rectangle appears.

Rectangles (Square Corners)

5. Click the Rectangle button 

in the Drawing tab>Shapes group on the CimEdit Ribbon bar..

   The cursor changes to a crossbar.

6. Move the cursor to the starting position for the object.

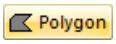
7. Hold down the left mouse button; drag the crossbar cursor to the ending location for the rectangle.

8. Release the mouse button.

   Result: A rectangle that represents the size you drew appears. Handles are available for moving and resizing the object.

Tip: When you click the **Rectangle** button, then click the left mouse button a duplicate of the last rectangle appears.

Polygons

9. Click the Polygon button 

in the Drawing tab>Shapes group on the CimEdit Ribbon bar..

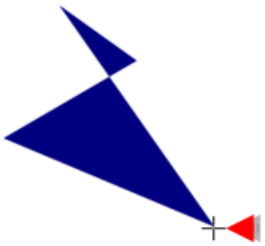
   The cursor changes to a crossbar.

10. Move the cursor to the starting position for the object.
11. Click the left mouse button to set the first point (or vertex) of the polygon.
12. Move the cursor to the second point you want to set.

    As you move the cursor, a line will be drawn between the last point you set and the cursor.

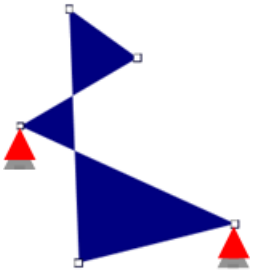
13. Click the left mouse button to set the next point.
14. Move the cursor to the next point you want to set.

    As you move the cursor, lines will be drawn between the last point you set and the cursor and between the cursor and the first point to create a closed polygon.



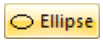
15. Click the left mouse button to set the next point.
16. Repeat 6 and 7 until you reach the last point in your polygon.
17. Double-click the left mouse button to finish drawing the polygon.

A polygon that represents the shape you drew appears. Handles are available for moving and resizing the object.



Tip: You can easily turn your polygon into a single straight line. Double-click the left mouse button after you create the first line of the polygon. You will create a single straight line.

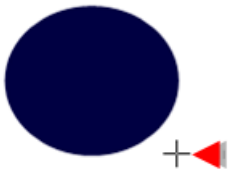
Basic Objects: Ellipses

1. Do one of the following.
   - Click the Ellipse button 

in the Drawing tab>Shapes group on the CimEdit Ribbon bar..
   - Click Tools>Ellipse on the menu bar.

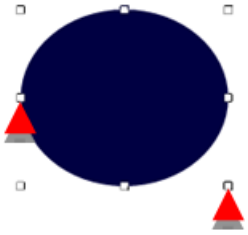
   The cursor changes to a crossbar.

2. Move the cursor to the starting position for the object.
3. Hold down the left mouse button and drag the mouse to the ending position for the ellipse.



4. Release the mouse button.

The ellipse will appear as you drew it. Handles are available for moving and resizing the object.



Tip: When you click the **Ellipse** button, then click the left mouse button a duplicate of the last ellipse appears.

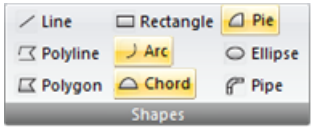
Basic Objects: Arc Lines, Chords and Pies

Important: The arc object first uses the visual properties of the object that was created last. Therefore, make sure that an object that includes a line was created so a line will display when you create the arc.

- Place a shape on the screen.
- Configure the shape.
- Object angles/direction and style example

Place a Shape on the Screen

1. Click one of the following in the Drawing tab>Shapes group on the CimEdit Ribbon bar.
   - **Arc**
   - **Chord**
   - **Pie**

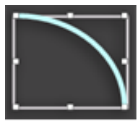
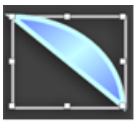
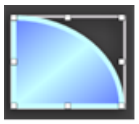


   The cursor changes to a crossbar.

2. Move the cursor to the starting position for the object.
3. Hold down the left mouse button.

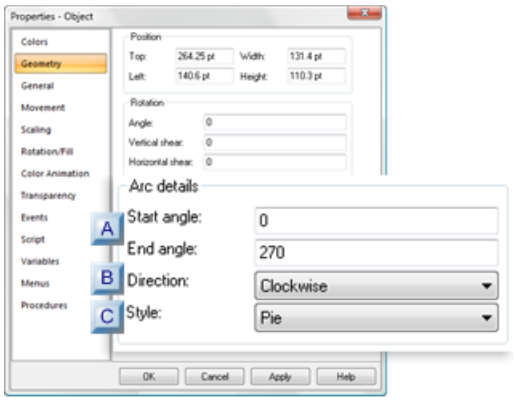
   The cursor location becomes the starting location of the arc.

   Result: The shape that represents what you drew displays. The shape can be changed at anytime.

Arc	Chord	Pie
		

Configure the Shape

4. Open the Properties dialog box for the selected shape.
5. Select **Geometry**.



rect 49, 125, 83, 153 *(page 77)*
rect 49, 153, 83, 181 *(page 78)*
rect 48, 182, 82, 210 *(page 78)*

A *(page 77)*	Start angle/End angle
B *(page 78)*	Direction
C *(page 78)*	Style

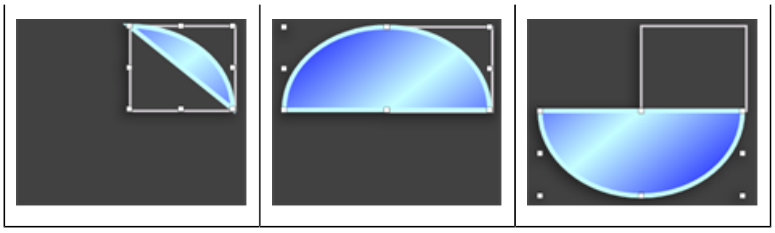
A	Start angle/End angle

The start and end position combined with the direction determine the shape of the selected style.

Example

- A chord is draw on the screen inside a rectangle.
- The Start and End angles only are changed in its Properties dialog box.
- The chord changes shape based on the Start and End angle specifications.

Start 90/End 0	Start 180/End 0	Start 0/End 180



Up *(page 77)*

B	Direction

The direction choices are:

- Clockwise
- Counter-clockwise

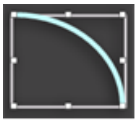
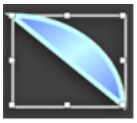
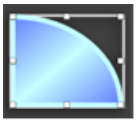
The shape and size the of the selected style is affected by the start and end angles, combined with the selected direction.

Up *(page 77)*

C	Style

The style can be changed at any time. during or after the object is configured.

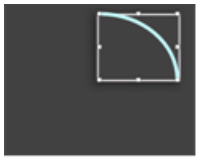
Options are:

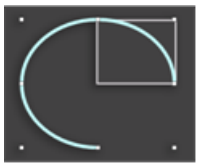
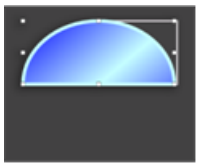
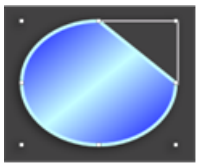
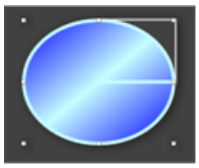
Arc	Chord	Pie
		

Object Angles/Direction and Style Example

An arc is placed on the screen within a rectangle.

Following is an example of the arc's first display and changes in its display when modifications are made.

Selections	Example
Start 90 End 0 Clockwise Arc	

Selections	Example
Start 270 End 0 Clockwise Arc	
Start 0 End 180 Counterclockwise Pie	
Start 0 End 90 Counterclockwise Chord	
Start 0 End 0 Either Pie	

Basic Objects: Pipe Tool

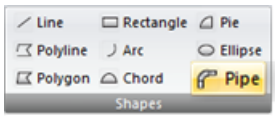
The CimEdit Pipe is a customized tool that enables you to draw pipes quickly.

The pipe tool provides you with:

- On screen formatting
- Detailed pipe formatting

On Screen Formatting

1. Click Pipe in the Drawing>Shapes group on the CimEdit Ribbon bar.



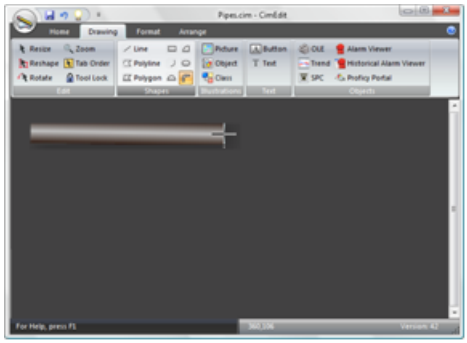
   The cursor changes to cross-hairs.

2. Click the CimEdit screen where you want the pipe to start.

Note: The pipe will be an object so you can move it at any time.

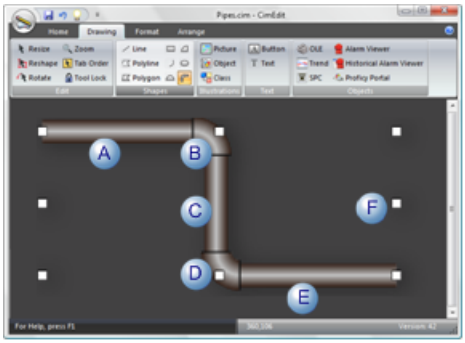
3. Drag the mouse across the screen.

   A pipe object is created as the cursor moves.



4. Click the left-mouse button when you want the pipe to bend.
5. Drag the pipe in the direction you want.
6. Click the left-mouse button if you want the pipe to bend again.
7. Double-click the left-mouse button when you have completed drawing the pipe.

The pipe displays based on your actions.



A	The pipe is straight for the first time you drag it.
B	An elbow is added to the pipe when you click the right-mouse button.
C	The pipe continues in the direction you drag the mouse
D	An elbow is added each time you click the right-mouse button.
E	The pipe continues in the direction you drag the mouse.
F	The pipe is completed as an object when you double-click the mouse.

Detailed Pipe Formatting

If necessary, you can refine the pipe's look in several ways.

Do one of the following.

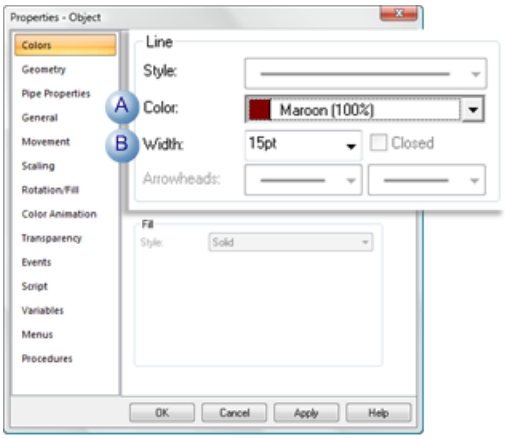
- Double-click the pipe object.
- Right-click the pipe object; select Properties on the Popup menu.
- Click Properties in the Home>Properties group on the CimEdit Ribbon bar.

The pipe's Properties dialog box opens.

1 *(page 81)*	Select the pipe's width and color.
2 *(page 82)*	Select the elbow and pipe shading properties.
3 *(page 84)*	Complete configuration.

1. Select the Pipe's Width and Color

1. Select Colors in the Properties dialog box.
2. Change either of the following.

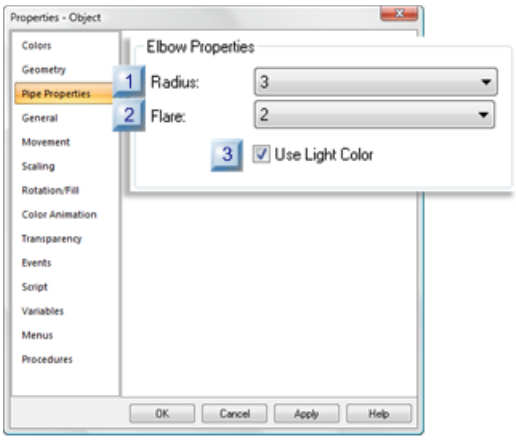


A	Color	can conform to your system needs.
B	Width	Can be different from what was drawn. The dropdown maximum is 11pt; however you can enter a higher width.

1. (Optional) Click Apply to apply and review your changes.

1. Select the Elbow and Pipe Shading Properties

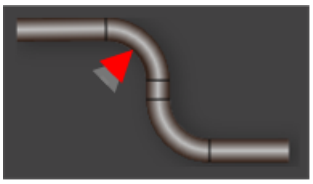
1. Select Piping Properties in the pipe's Properties dialog box.
2. Change either of the following.



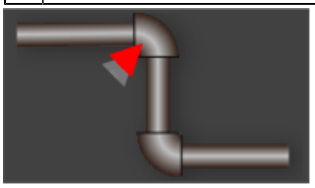
rect 65, 36, 92, 56 *(page 82)*
rect 65, 56, 92, 77 *(page 83)*
rect 129, 83, 156, 104 *(page 83)*

1 *(page 82)*	Radius
2 *(page 83)*	Flare
3 *(page 83)*	Use Light Color

1	Radius	
	Elbow radius dimensions can be as follows.	
	Minimum	
	Maximum	
	Example	
	A pipe properties have been changed. • The radius is set to 8. • The flare is 0.	



2	Flare	
	Flare dimensions can be as follows.	
	Minimum	
	Maximum	
	Example	
	A pipe's properties have been changed. • The radius is set to 1. • The flare is set to 7.	

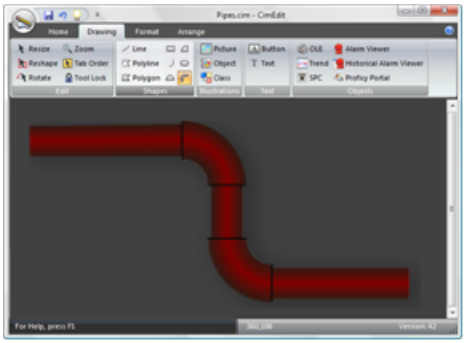


3	Use Light Color	
	Check or clear the checkbox for the following affects.	
	Checked	
	Clear	
	Note: The exact rendering of the display depends on the color and shade that were originally selected.	
	Examples	
	Maroon Originally Selected Maroon (100%)	
	Checked	
	Clear	
	Yellow Originally Selected 255,255,132 (100%)	
	Checked	
	Clear	

1. Complete Configuration

Click OK when you have completed your configuration.

The pipe changes to reflect your specifications.



Tools to Create Several Objects Quickly

You may want to make several of the same type of graphics object or create duplicates of the same object. You can do either easily.

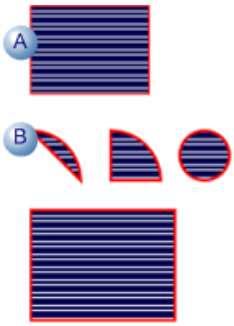
- Last object configured.
- Ctrl key.
- Multiple duplicates.

Last object configured

1. Create an object.

2. Click any of the following buttons.
    - Ellipse
    - Chord
    - Pie
    - Rectangle
3. Click the left mouse button anywhere on the screen.

    Result: CimEdit places a visual configuration duplicate of the last object that was created on the screen.



A	Last shape configured.
B	Shapes with the same fill and line configuration are placed on CimEdit screen when their button is clicked and the left-mouse button is clicked.

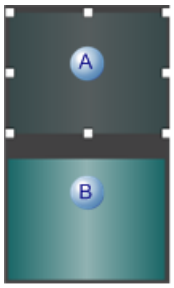
Note: : You can also use this method to create the first of an object on the screen. When you do, they will appear as follows:

Ellipse Size	36 pt by 36 pt (in other words, a circle)
Rectangle Size	36 pt by 36 pt.
Arc Shape	One quarter of an underlying 36 pt by 36 pt ellipse.
Text Button Size	36 pt by 36 pt.

4. Select the object by placing the cursor anywhere in the object.
5. Hold down the left mouse button.
6. Press the **Ctrl** key while holding down the left mouse button.
7. Drag the cursor in any direction. A duplicate of the object moves to where you drag it.

   Ctrl key

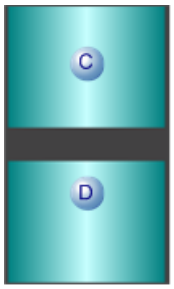
8. Select an object on the CimEdit screen.
9. Hold down the Ctrl key.
10. Hold down the left-mouse button while you drag the object to where you want the duplicate object.



A	Transparency that displays the object's original position.
B	The original object being moved when the mouse button and Ctrl key are pressed.

11. Release the mouse button.

    Result: CimEdit duplicates the moved object.



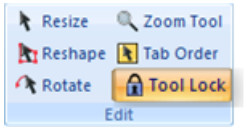
C	Duplicate object.
D	Original object.

Multiple Duplicates

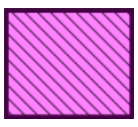
By default, the Tools toolbar enables an object tool for only the one time you create the object.

If you have several objects with the same visual configuration that you want to create use the Tool Lock feature to duplicate the first configuration.

12. Click the Tool Lock button in the Drawing>Edit group on the CimEdit Ribbon bar.



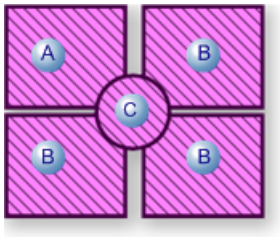
13. Create an object.



14. Continue to select the created shape or select another shape.
15. Click the left-mouse button as many times as you need to create duplicate shapes.

    Note: You can perform other tasks, e.g. moving and aligning objects while the Tool Lock button is enabled. When you select another shape, it will continue to duplicate the original formatting..

    Result: CimEdit places objects with the same visual configuration on the screen.



A	Last shape configured while or just before Tool Lock was enabled.
B	Rectangle shape still selected, while left-mouse is clicked.
C	Circle shape is selected; left-mouse is clicked.

16. Click Tool Lock off when you are done duplicating the object configuration.

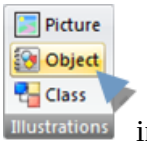
Object Explorer

Object Explorer

The Object Explorer provides you with access to thousands of objects that have been drawn by artists and enable you to create professional looking screens with a minimum of graphic effort. In addition to helping you with the CimView screen's graphic presentation, the Object Explorer provides you with several SmartObjects that are already automated and simply prompt you for configuration values.

1 (page 87)	Display the symbols libraries.
2 (page 88)	Place a Symbol object on a CimEdit screen.
3 (page 91)	Add Symbols Libraries to the Object Explorer.

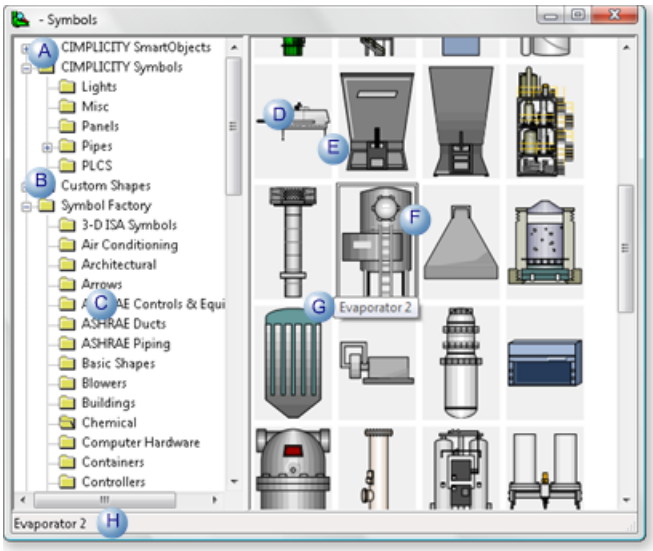
1. Display the Symbols Libraries

Click **Object** 

in the Drawing>Illustrations group.

The Symbols window opens.

Features in the Symbols window include.



A	Default folders in the left pane. • CIMPLICITY Smart Objects. • CIMPLICITY Symbols. • Symbol Factory
	Note: Other (custom) folders may display that were added through the CimEdit screen.
B	User created folders.
C	Subfolders that contain the objects.
D	Objects in the selected subfolder.
E	Grid for organizing the objects.
	Note: All objects are scaled to the same size.
F	Black box surrounding a selected object.
G	Tool tip, when configured for the object.
H	Tip in the Symbols status bar, when configured for the object.

2. Place a Symbol Object on a CimEdit Screen

- Place an Object Explorer object on a CimEdit screen.
- Place a SmartObject on a CimEdit screen.

Place an Object Explorer object on a CimEdit screen

1. Open the Object Explorer through CimEdit.
2. Find and select an object using the mouse or the keyboard.

   Mouse

   Click any object to select it.

   Keyboard

   Keyboard keys operated in the Symbols window as follows.

Keys	Function in Symbols window
Tab	Move to the portion (the category tree or the object grid) of the Object Explorer that you want to affect
Arrows	Select the object that is next to the current object in the direction of the arrow when an object is already selected
Home	Select the top-left most object
End	Select the bottom-right most object.

3. Do one of the following to insert an object on a CimEdit screen.

   Note: When the object is inserted on the CimEdit screen, it may be linked or stand-alone based on whether or not the Link named objects as default *(page 561)* checkbox is checked or clear in the Global Configuration dialog box.

   - Drag a selected object to where you want it in the CimEdit workspace.

   You can place a copy of or link to an Object Explorer object on the CimEdit screen.

   - Double-click the left mouse button on the desired object.
   - Select an object; press **Enter**.

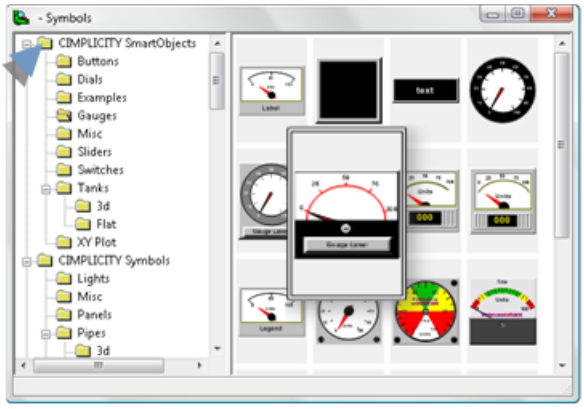
   Result

   - If the object is not a SmartObject, depending on the method you used, it is placed:
   - Where you dragged it or
   - In the center of the last active CimEdit window; that window becomes the foreground window.
   - If the object is a SmartObject:
   - A SmartObject Configuration dialog box may open for pre-placement configuration.

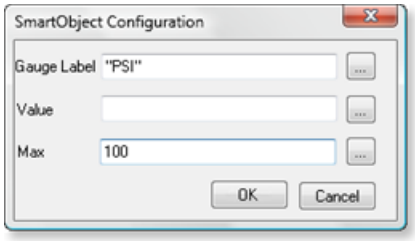
Note: If the SmartObject Configuration dialog box does not open for pre-placement, to open it, double-click the object after it is placed on the screen

Place a SmartObject on a CimEdit screen

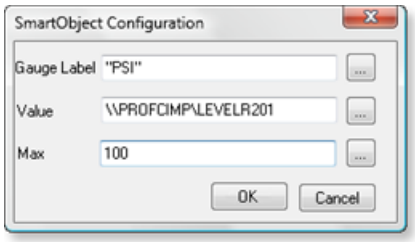
4. Select an object that is in the CIMPLICITY SmartObjects folder.



The SmartObject Configuration dialog box for the selected object opens.

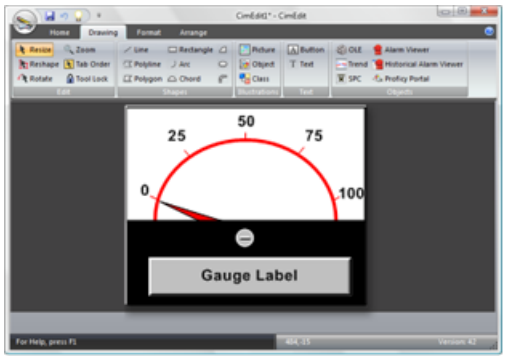


5. Fill in the fields that display for the selected object.



6. Click OK.

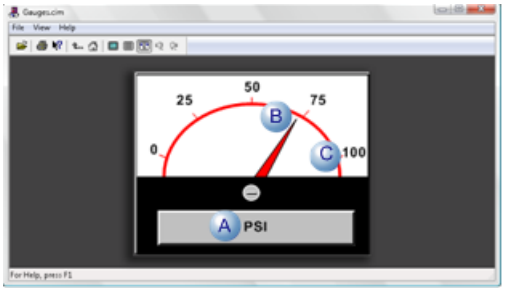
The SmartObject is placed on the CimEdit screen.



7. Make sure the project is running.

8. Click Test Screen to test the SmartObject.

The SmartObject display reflects the specifications entered in the SmartObject Configuration dialog box.



A	Gauge Label
B	Value
C	Max

3. Add Symbols Libraries to the Object Explorer

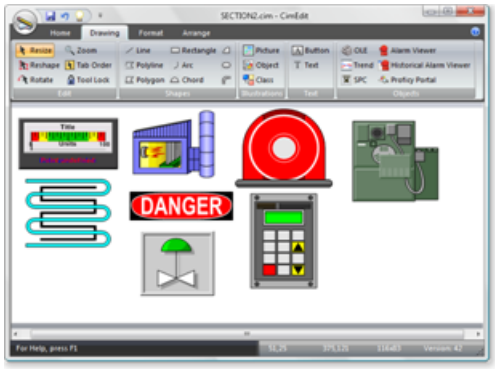
3. Add Symbols Libraries to the Object Explorer

The Symbols Libraries in the Object Explorer are made up of CimEdit screens that contain a variety of graphic objects. The CimEdit screens

3.1 *(page 92)*	Create custom screens for the Symbols library.
3.2 *(page 92)*	Specify folders for the Symbols library.

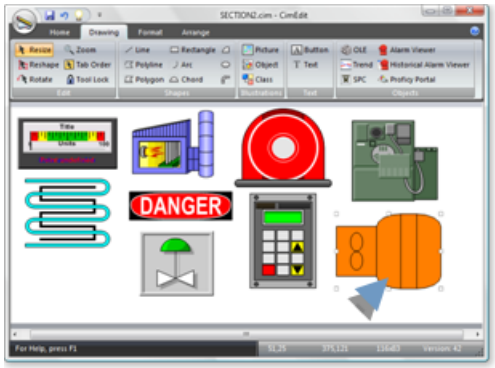
3.1. Create Custom Screens for the Symbols Library

1. Do one of the following.
   - Create a new CimEdit Symbols screen.
   - Open an existing CimEdit Symbols screen.



2. Add new objects to the screen.

   Note: Use any method available to CimEdit.



3. Save the screen in the default location or in a location identified *(page 92)* in the Global Configuration dialog box.

The new objects are now available through the Object Explorer.

3.2. Specify Folders for the Symbols Library

- Select symbols libraries for the Object Explorer
- Add additional symbols libraries to the default location

Select symbols libraries for the Object Explorer

By default, the Object Explorer uses the directory structure in the default Symbols directory on your computer.

```
..\Program Files\Proficy\Proficy CIMPLICITY\Symbols
```

Where

```
...\Program Files\Proficy\Proficy CIMPLICITY
```

is the location of the CIMPLICITY installation directory

If the files that should be viewed through the Object Explorer are not located in the default directory or are both in the default directory and a directory in a different location, you need to define the location(s) as follows.

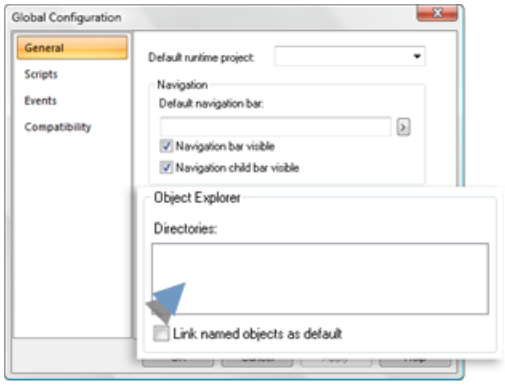
1. Click the CimEdit button .
2. Select the Global Configuration button.

   The Global Configuration dialog box opens.

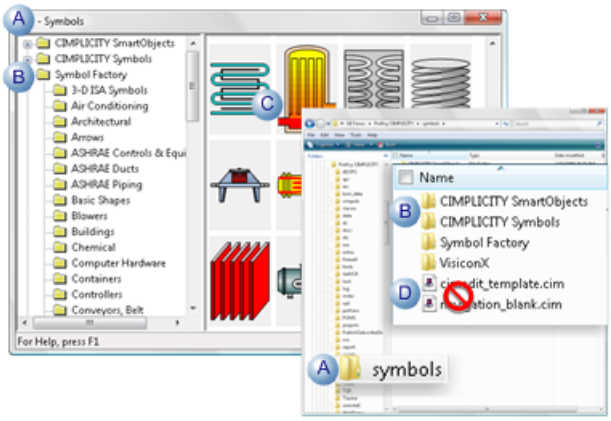
3. Enter specifications in the **Object explorer directories** box to display any of the following.
   - Default Symbols directory only
   - Symbols folders in a location other than the default
   - Default Symbols directory and symbols folders in a different location

   Default Symbols directory only

   Leave the Object Explorer directories box blank.

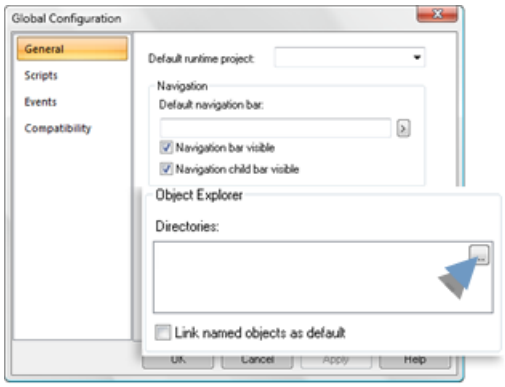
   Result: The following happens.



A	The top level folder (Symbols) acts as the container for sub-folders with symbols screens.
B	Symbols sub-folders display in the Symbols library left pane.
C	Symbols on screens in the sub-folders display when the sub-folder is selected.
D	Screens that are in the top-level folder do not display in the Object Explorer.

Symbols folders in a location other than the default

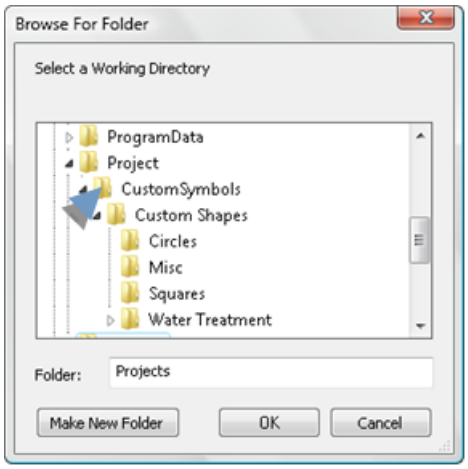
- Place the cursor in the **Object explorer directories:** box. A Browse button displays on the right side of the box.
- Click the Browse button.



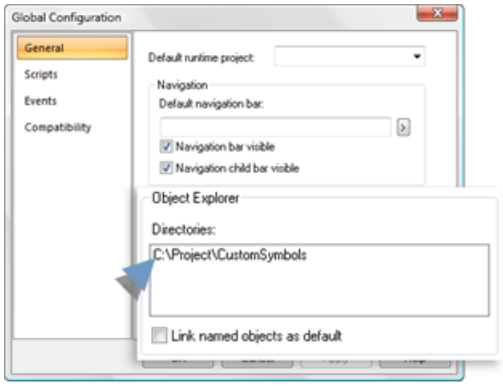
A Browse For Folder browser opens.

Select the folder the contains the sub-folders with symbol screens.

Important: The folder you select will not be listed in the Object Explorer's left pane. The folder's sub-folders will be listed.

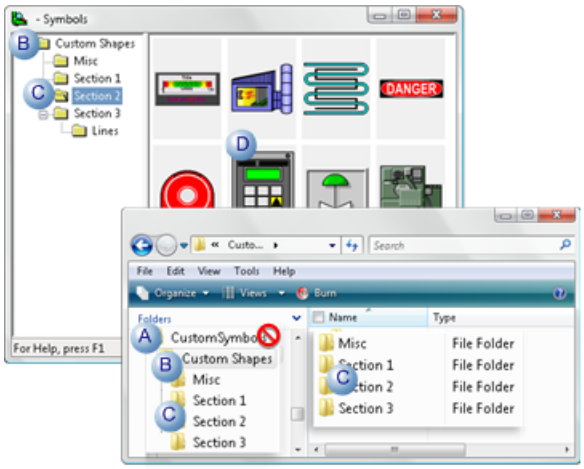


The selected path displays on the first line in the **Object explorer directories** box.



- Click the next line in the Object explorer directories box.
- Follow the same procedure until all of the folders that contain sub-folders with symbols screens are selected.
- Click OK.

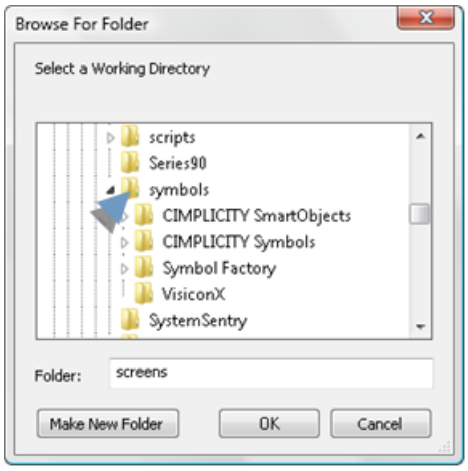
Result: When the Object Explorer is opened the subfolders display in the Object Explorer left pane. Folders (and CimEdit screens) that are located in Windows Explorer display in the Symbols window, as follows



	In Windows Explorer:	Symbols Window
A	The selected top-level folder	Does not display in the Symbols window. **Note:** Objects on any CimEdit file at the top level will not display either.
B	The first subfolder under the top level folder	Displays as the top folder.
C	The subfolder tree at the next level	Displays as a tree.
D	Objects on every CimEdit file that is in a single folder	Display when that folder is selected.

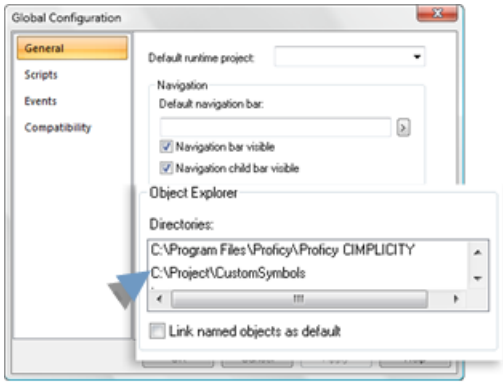
Default Symbols directory and symbols folders in a different location

- Place the cursor in the Object explorer directories: box. A Browse button displays on the right side of the box.
- Click the Browse button. A Browse For Folder browser opens.
- Select the default Symbols folder in the CIMPLICITY directory.

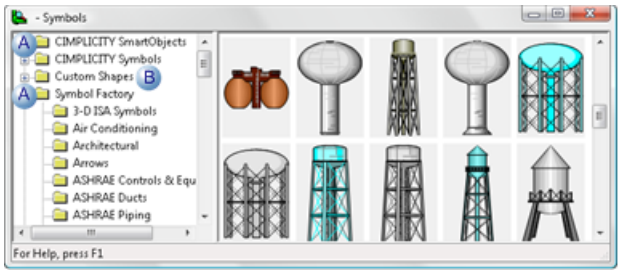


- Click OK. The default Symbols folder path displays on the first line.
- Select the second line in the Object explorer directories box.
- Follow the same procedure to select all the (top-level) folders that are in other locations.

- Click OK.



Result: The next time the Object Explorer is opened both the Symbols libraries that are in the default location and the Symbols libraries that are not in the default location will display in the Object Explorer.



A	Subfolders in the default location.
B	Subfolders in a custom location are located in the list in alphabetical order.

Add additional symbols libraries to the default location

Select ...\Program Files\Proficy\Proficy CIMPLICITY\Symbols in Windows Explorer.

Where ...\Program Files\Proficy\Proficy CIMPLICITY\ is the location of the default installation directory.

Create one or more new folders.

Results: The new folder will be listed in the :

- Object Explorer the next time you open it.
- Workbench.

Note: If the Workbench Symbols feature is open when you create the folder, press F5 to refresh the screen.

Text Objects

Text Objects

CimEdit provides several tools to format and configure text and text button objects.

The text object appearance has tools for on screen and detailed formatting.

Text rect 2, 3, 54, 30 Text Object: On Screen Formatting *(page 98)*	Text: On screen formatting.
Text rect 2, 3, 54, 30 Text Object: Detailed Formatting 2 *(page 100)*	Text: Detailed formatting.
Button rect 0, 2, 54, 30 Text Button: On Screen Formatting *(page 106)*	Text Button: On screen formatting.
Button rect 0, 2, 54, 30 Text Button: Detailed Formatting *(page 108)*	Text Button: Detailed formatting.

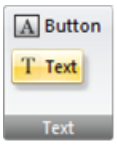
Text Object: On Screen Formatting

Text can be quickly edited and formatted without opening its Properties dialog box.

1 *(page 98)*	Place text on the CimEdit screen.
2 *(page 99)*	Quickly edit the text.

Place text on the CimEdit screen.

1. Click Text in the Drawing>Text group on the CimEdit Ribbon bar.

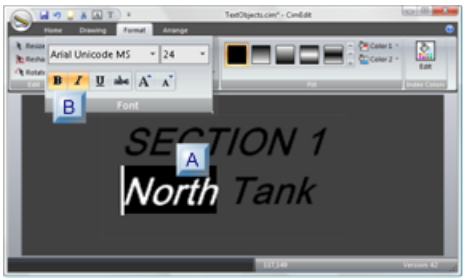


1. Click the CimEdit screen to place the text object.

Result: The word Text displays on the CimEdit screen in Edit mode.

Quickly edit the text.

You can quickly change the text and text format, as follows.



rect 105, 83, 133, 110 *(page 99)*
rect 26, 49, 54, 74 *(page 99)*

A *(page 99)*	Text content.
B *(page 99)*	Font group options.

A	Text Content

The text display can be edited on screen.

1. Type the text you want to display.
2. (To go to a new line) Press Enter.
3. (If your typing needs to be changed) do any of the following.

- Press the Backspace key to delete text; resume typing.
- Highlight a portion of the text; resume typing.

B	Font Group Options

Select any of the options in the Format>Font group.

Font selections include the following.

1	Type.
2	Size (in points) **Note:** As the cursor moves over the sizes in the dropdown **Size** list, the text on the screen changes to the corresponding size.
3	Style.
4	Size increase/decrease.

Note: When you click the Decrease Font Size button

- The text decreases in size.
- CimEdit displays an indication of the previous size.

- Clicking the Increase Font Size button returns the text to a previous larger size.

Click the text object when you have completed the quick format.

Text Object: Detailed Formatting 2

This task provides an overview and information on custom text properties and font colors in CimEdit/CimView.

Overview

The text object's Properties dialog box provides additional formatting options.

1. Select the text object.

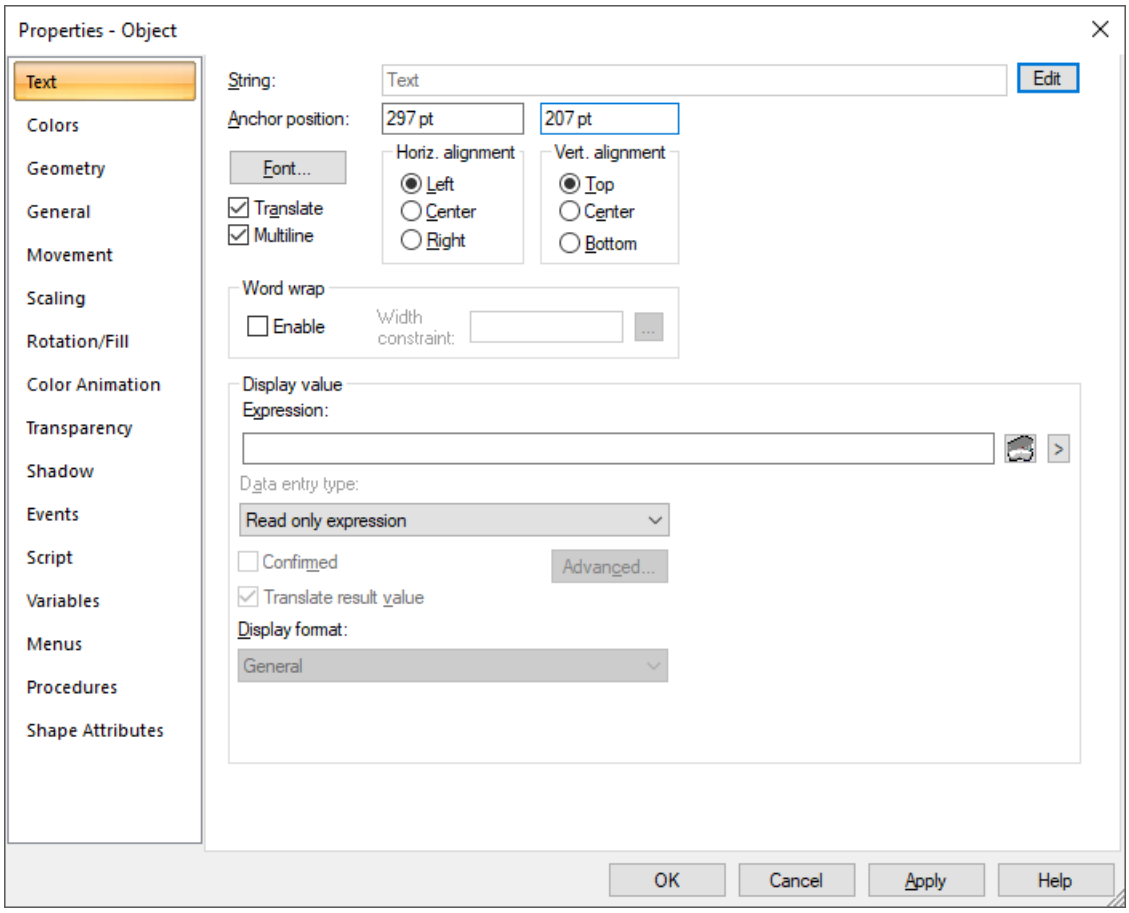
2. Do one of the following.
   - Double-click the text object.
   - Click Properties in the Home>Properties group on the CimEdit Ribbon bar.
   - Right-click the text object; select Properties on the Popup menu.

Result: The Properties dialog box opens when you use any of these methods.

Custom Text Properties

Select Text in the Properties dialog box.

Text formatting options are as follows.



rect 201, 34, 222, 54 *(page 102)*
rect 97, 59, 123, 80 *(page 103)*
rect 98, 93, 122, 115 *(page 103)*
rect 97, 121, 124, 138 *(page 104)*
rect 208, 120, 238, 143 *(page 104)*
rect 320, 121, 344, 142 *(page 105)*
rect 102, 195, 131, 219 *(page 105)*

A *(page 102)*	String/Multiline.
B *(page 103)*	Anchor position.

C *(page 103)*	Font.
D *(page 104)*	Translate.
E *(page 104)*	Horiz alignment.
F *(page 105)*	Vert. alignment.
G *(page 105)*	Wrap Text.

For information on Display value, see 8. Text Value Animation *(page 398)*.

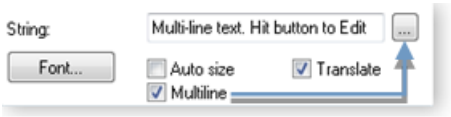
A	String/Multiline

A text string can be used in CimView in either of two ways as:

- A string that is a place holder for an expression or setpoint.
- Text that will display for as long as CimView is open.

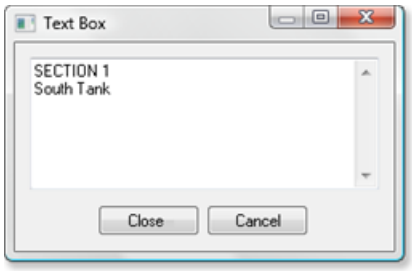
Multiline is checked by default when you open a text Properties dialog box.

When Multiline is checked the String field displays the text "Multi-line text. Hit button to Edit."



When the Multiline checkbox is:

Option	Description
Clear	The multiline text clears. The String field is available for a single line entry.

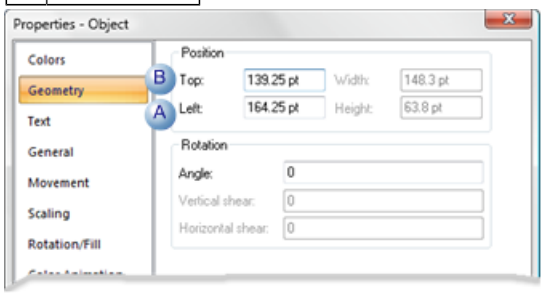
Option	Description
Checked	Click the Text Box button to the right of the String field. A text box opens, displaying the text that was written on the screen.  Multiline text can be entered. • If you entered multiline text on the screen and want to keep it, make sure you leave Multiline checked. • Press Ctrl+Enter to create a multiline entry to display on the CimEdit/CimView screen.

B	Anchor position

(Manual entry optional) X/Y location for the left top corner of the string.

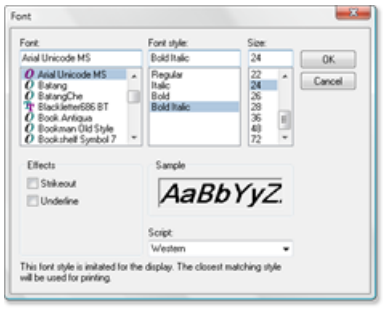
Note: The anchor positions display as the Geometry **Left** and **Top** positions.

A	**Left** position
B	**Top** position



C	Font

Opens the Font dialog box to select the font, style and size that displays on the CimEdit/ CimView screen for the selected string.



Important: Select a font that will accommodate the characters for all languages that will be used on the CimView screen. For example, Arial Unicode MS is a very complete Unicode font.

D	Translate

Checked makes this string available for translation. For more information on translation, see Translation *(page 66)*.

Tip: Check Emit translation strings every save on the File tab in the Options dialog box to emit new text strings to the CIMPLICITY Language Mapper each time CimEdit is saved.

Click Tools>Options on the CimEdit Menu bar; select the File tab.

Click **OK** to save your changes and close the dialog box.

Default	Checked

E	Horiz. alignment

The horizontal alignment positions the text based on the stationary anchor.

Options are:

Option	The text:
Left	Left side aligns with the anchor.
Center	Center aligns with the anchor.

Option	The text:
Right	Right side aligns with the anchor.

F	Vert. alignment

The horizontal alignment positions the text based on the stationary anchor in the **Top** position.

Options are:

Option	The text:
Top	Top aligns with the anchor.
Center	Center aligns with the anchor.
Bottom	Bottom aligns with the anchor.

Result: The object will be the entered text string with the selected font configuration.

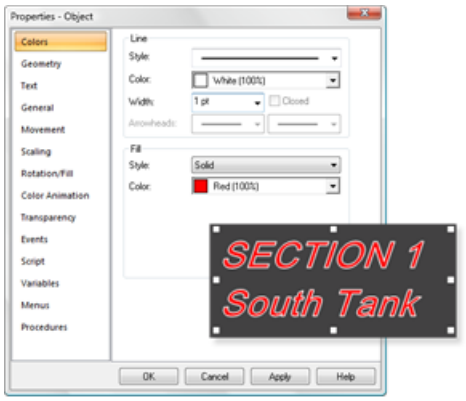
G	Text Wrap

You can set the width of Text Strings (text objects) so that they auto wrap and do not extend off of the screen or intrude onto other objects on the screen.

- Select **Text String**.
- With the text object selected, right click on the text object and select Properties in the Popup menu.
- The text object properties screen opens.
- In the Word wrap section, click Enable.
- Do one of the following: Enter a width. Width is measured in points and 72 points equal 1 inch.Select the helper, a graphical tool that lets you drag the text object to the width you want. Click Apply or OK.
- Click **Apply** or **OK**.

Font Color

Select **Colors** in the Properties dialog box. Text objects can:



- Display any selected color. *(page 165)*
- Have a border in any selected color. *(page 194)*

Text Button: On Screen Formatting

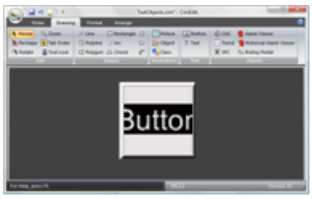
1 *(page 106)*	Place a text button on the CimEdit screen.
2 *(page 107)*	Quickly edit the text button.

Place a text button the CimEdit screen.

1. Click Button in the Drawing>Text group on the CimEdit Ribbon bar.

1. Click the CimEdit screen to place the text button.

Result: A button is placed on the CimEdit screen; the word Button displays on the button in Edit mode.

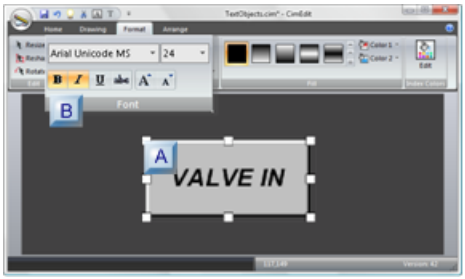


Note: The button style when the button is placed on the screen depends on what is selected in the Global Configuration *(page 574)* dialog box.

Quickly edit the text.

Note: The button object size can be quickly increased or decreased the same as any object.

You can quickly change the text and text format on the button, as follows.



rect 88, 83, 108, 106 *(page 107)*
rect 28, 55, 53, 78 *(page 108)*

A *(page 107)*	Text content.
B *(page 108)*	Font group options.

A	Text Content

The text content can be edited on screen.

1. Type the text you want to display.
2. (To go to a new line) Press Enter.
3. (If your typing needs to be changed) do any of the following.

- Press the Backspace key to delete text; resume typing.

- Highlight a portion of the text; resume typing.

B	Font Group Options

Select any of the options in the Format>Font group.

Font selections include the following.

1	Type.	
2	Size (in points) **Note:** As the cursor moves over the sizes in the dropdown **Size** list, the text on the screen changes to the corresponding size.	
3	Style.	
4	Size increase/decrease.	
	When you click the button:	
	Decrease Font Size	
	Increase Font Size	

Text Button: Detailed Formatting

This task provides an overview and information on custom font properties, styles, colors, and font colors for buttons in CimEdit/CimView.

Overview

If you assign a procedure or a setpoint to a text button, and a user moves the cursor over the button in CimView and clicks the left mouse button, the button on the screen goes down.

The text button's Properties dialog box provides additional formatting options.

1. Select the text button.

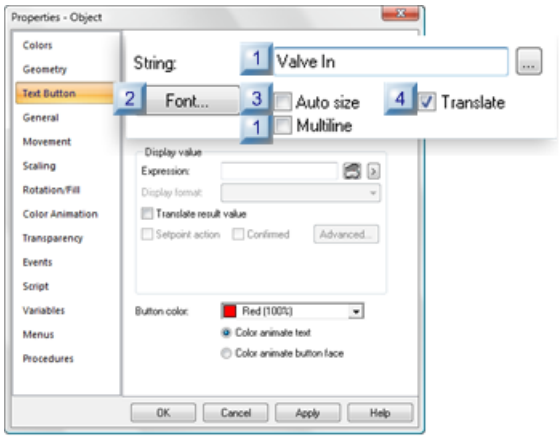
2. Do one of the following.
   - Double-click the button.
   - Click Properties in the Home>Properties group on the CimEdit Ribbon bar.
   - Right-click the button; select Properties on the Popup menu.

   Result: The Properties dialog box opens when you use any of these methods.

Custom Button Font Properties

Select Text Button in the Properties dialog box.

Text button formatting options are as follows.



rect 236, 47, 260, 71 *(page 111)*
rect 145, 44, 169, 68 *(page 111)*
rect 67, 45, 91, 69 *(page 110)*
rect 146, 21, 170, 45 *(page 109)*
rect 146, 66, 170, 88 *(page 109)*

1 *(page 109)*	String/Multiline.
2 *(page 110)*	Font.
3 *(page 111)*	Auto size.
4 *(page 111)*	Translate.

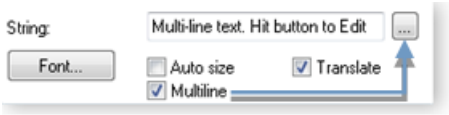
1	String/Multiline

A text string can be used in CimView in either of two ways as:

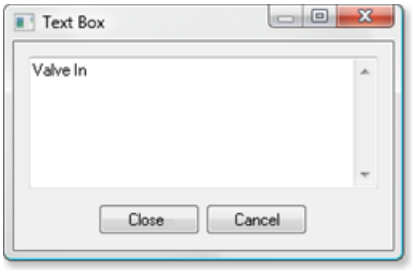
- A string that is a place holder for an expression or setpoint.
- Text that will display for as long as CimView is open.

Multiline is checked by default when you open a text Properties dialog box.

When Multiline is checked the String field displays the text "Multi-line text. Hit button to Edit."

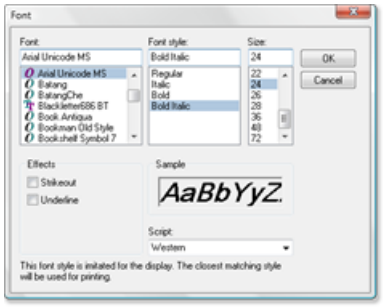


When the Multiline check box is:

Option	Description
Clear	The multiline text clears. The **String** field is available for a single line entry.
Checked	Click the Text Box button to the right of the String field. A text box opens, displaying the text that was written on the screen.  Multiline text can be entered. • If you entered multiline text on the screen and want to keep it, make sure you leave Multiline checked. • Press Ctrl+Enter to create a multiline entry to display on the CimEdit/CimView screen.

2	Font

Opens the Font dialog box to select the font, style and size that will display on the CimEdit/ CimView screen for the selected string



Important: Select a font that will accommodate the characters for all languages that will be used on the CimView screen. For example, Arial Unicode MS is a very complete Unicode font.

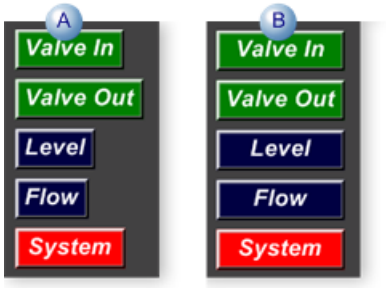
3	Auto size

When checked, the button automatically re-sizes to fit the text length.

There are some situations where auto-size can help in configuration. However, more often than not, it is recommended that you leave Auto size clear.

Auto-sizing will create a jagged effect is words are different lengths. If you are translating the screen auto-sizing can create different size buttons for the same word in different languages.

Example:



A	Auto size is checked.
B	Auto size is clear.

4	Translate

Checked makes the text string available for translation. For more information on translation, see Translation *(page 66)*.

Tip: Check Emit translation strings every save on the File tab in the Options dialog box to emit new text strings to the CIMPLICITY Language Mapper each time CimEdit is saved.

Click Tools>Options on the CimEdit Menu bar; select the File tab.

Default: Checked

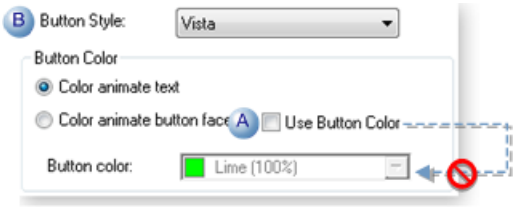
Custom Button Style

CimEdit provides four button styles; each style affects the button's shape and selected color shade *(page 114)* and application.

The options in their default state are as follows.

Button Style	Displays:
Use Global *(page 574)*	The Button style option selected in the Global Configuration dialog box.
Classic	Classic
Standard	Standard
XP	XP
Vista	Vista

Do the following to assign one of the default button styles

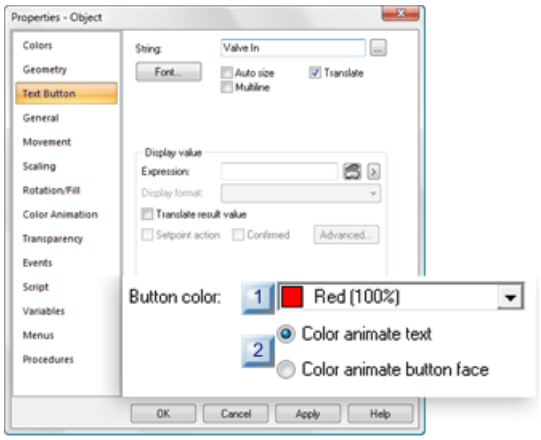


A	Use Button Color	Clear Use Button Color. The **Button color** field is disabled.
B	Button Style	Select a style in the **Button Style** field.

Result: The button will display the selected default style when Apply or OK is clicked.

- The button style when the button is placed on the screen depends on what is selected in the Global Configuration *(page 574)* dialog box.
- When a color is applied to the button, the color application *(page 114)* emulates the selected style.

Custom Button Color



rect 144, 204, 171, 228 *(page 113)*
rect 144, 170, 171, 194 *(page 113)*

1 *(page 113)*	Button color.
2 *(page 113)*	Color animate.

1	Button color

The selected button color may be either of the following.

- The color that displays in CimView.
- A placeholder for button color animation configuration.

Notes

- Click the Down Arrow to the right of the Button color field to open the CimEdit color palette.
- If you use color animation the radio buttons select whether the text of the button face will be animated.

2	Color animate

The CimEdit color animation *(page 381)* feature can be applied to CimEdit buttons.

The Color animate selection you make determines whether the button face or the font will be affected.

Radio Button Checked	Display during runtime
Color animate text	Text changes color to the color selected in the Color animation dialog box.

Radio Button Checked	Display during runtime
Color animate button face	Button face changes color to the color selected in the Color animation dialog box.

Result: The text button object will display your selections on the CimEdit screen.

- You can make duplicates of text buttons the same way you do basic graphic objects.
- If the button display does not seem to represent the selected style, follow the procedure to apply the default style and then select the color.

Example: Button Styles when Color Applied

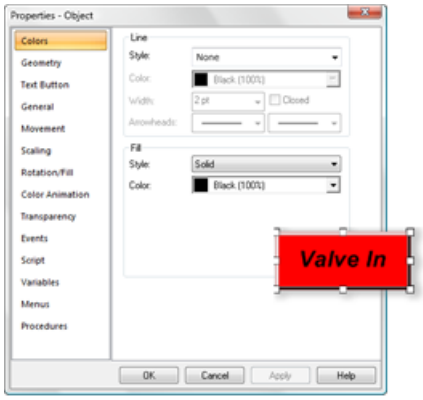
When Lime Green is applied to a available button styles, the buttons display is similar to the following.

Button Style	Display
Classic	Classic
Standard	Standard
XP	XP
Vista	Vista

Button Font Color

Select **Colors** in the Properties dialog box.

Button fonts can:



- Display any selected color. *(page 165)*
- Have a border in any selected color. *(page 194)*

Graphic Images and Transparency

Graphic Images and Transparency

You can insert several types of graphic files into a CimEdit screen and change one of its colors, either by assigning another color or making it transparent.

The file types are:

- .bmp
- .gif
- .jpg
- .png
- .tif

Note: If you use a graphic, such as a .png file that already has an Intrinsic transparency, CimEdit recognizes that transparency in addition to a color that you select in CimEdit.

Steps to work with images are as follows.

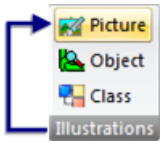
Step 1 *(page 115)*	Place a graphic (picture) object on a CimEdit screen.
Step 2 *(page 118)*	Choose to link or embed the graphic file.
Step 3 *(page 120)*	Change the color in the graphic object.

Step 1. Place a Graphic (Picture) Object on a CimEdit Screen

- Procedure: Insert a graphic.
- Guidelines: File selection.

Procedure: Insert a Graphic

1. Select the Drawing tab on the CimEdit ribbon bar.
2. Click the Picture button in the Illustrations group.

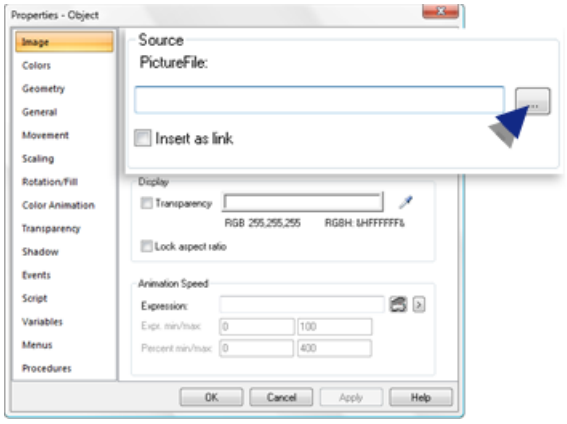


The cursor turns into a crosshair.

3. Place the crosshair in the location you want the graphic object's top left corner to be placed.
4. Click the left-mouse button.

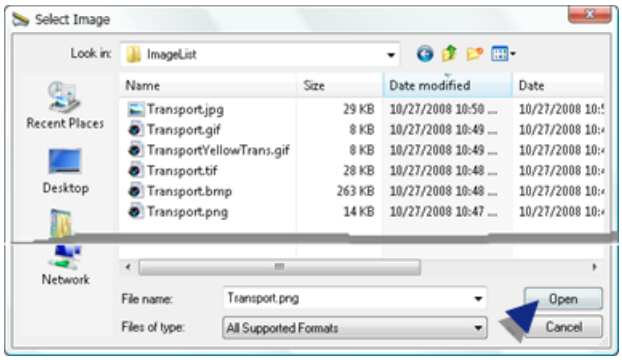
   A Properties - Object dialog box opens; image options display.

5. Click the Open file button to the right of the **PictureFile** field.



   An Open dialog box opens.

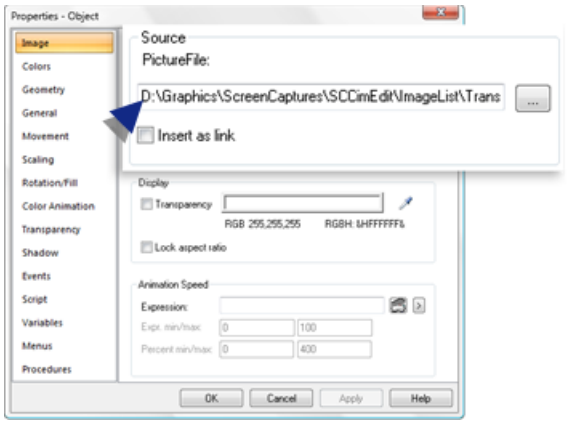
6. Find the image you want to insert.



7. Click Open.

   Results

   • The path and file name you select displays in the PictureFile field.



- The selected image is placed on the CimEdit screen at the specified location.

Note: The graphic is inserted at its actual size; it does not automatically zoom even if the CimEdit screen has been zoomed, e.g. is 50% of the actual size.

Guidelines: File Selection

- You have the option to link or embed the graphic when you assess the file size in you selection.
- If a transparency was applied to a color when a file, e.g. a .png file, was created in another graphics application, CimEdit will

8. Maintain the transparency when the image is first inserted into the CimEdit screen.

9. Treat that transparency the same way it treats the color you select in CimEdit when you enable transparency in CimEdit.

   As a result, you will be able to change or make transparent in CimEdit areas that were different colors in the original graphic.

   Example

   Three graphics are inserted on CimEdit screens (or the same screen) that have a charcoal gray background.

   Two of the graphic objects were created with intrinsic transparencies in another graphics application.

   The images display as follows when they are first placed on the CimEdit screen

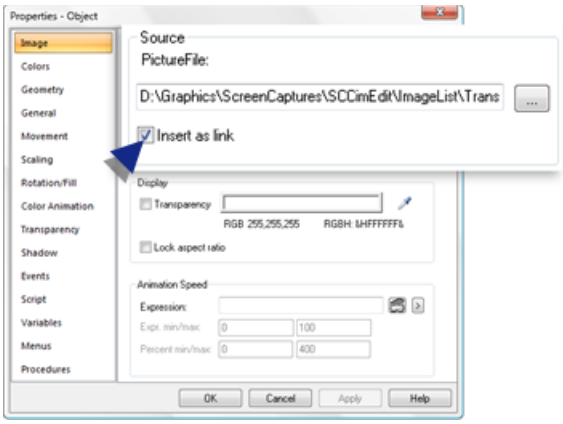
Image/Transparency	Display
.bmp No transparency	
.png When created applied transparency to red pixels.	
.gif When created applied transparency to black pixels.	

- You can select graphics program and copy/paste the image into CimEdit. However, if you use that technique, you will not be able to manipulate the color

Step 2. Choose to Link or Embed the Graphic File

The graphic file can be inserted as a link or embedded in the CimEdit screen.

Whether or not you check Insert as link depends on your system requirements.



When Insert as link is:

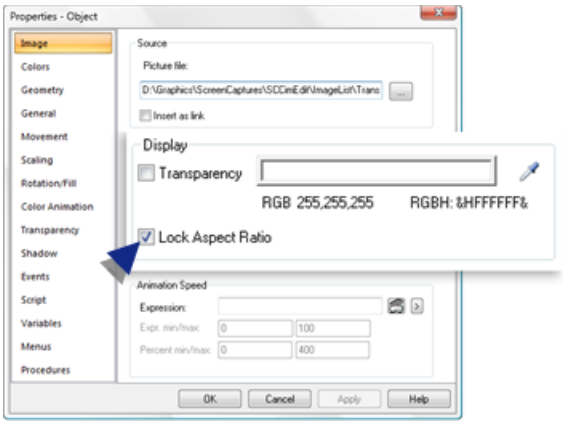
Cleared	(Default) The graphic file is embedded in the CimEdit screen.
Checked	The file is linked to the CimEdit screen.
	Advantage
	Issue

Step 3. Lock Aspect Ratio

You can lock a graphic's aspect ratio to make the graphic retain its shape within the boundaries of the inserted image. when the graphic size is changed.

Lock Aspect Ratio is particularly useful when an image is linked to keep the aspect ratio if the image changes or in a script setting.

Do one of the following



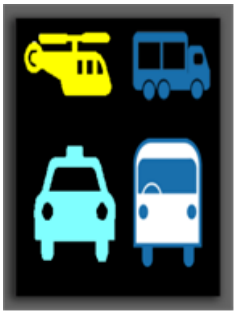
Check/Clear	The image's aspect ratio
Check	Will be locked.
Clear	May change if the image size changes.

Example

1. A linked transport.png file is inserted on a the CimEdit screen.
2. The transport.png graphic image changes at the linked location.

   Results: The revised image:

   - Is sized within the dimensions of the original image.
   - Has a height to width ration base on whether or not the aspect ratio was locked.

Image Inserted	Image Changed	
	Aspect Ratio Locked	Aspect Ratio Not Locked
The height of a revised image is greater than the original image.		
The height of the revised image is less than the original image.		
		

Step 4. Change the Color in the Graphic Object

Step 4. Change the Color in the Graphic Object

Note: You can perform the following procedures in either order and can change your selections as many times as you want to obtain that effect that works best for your screen.

Step 4.1 *(page 121)*	Select how the color will be changed
Step 4.2 *(page 122)*	Select the color that will be changed.

Step 4.1. Select how the Color will be Changed

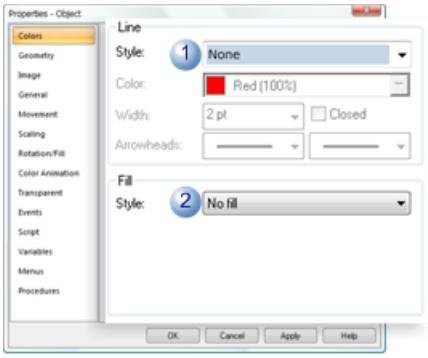
Select Colors in the Properties - Object dialog box.

Options that can be applied to the selected graphic object's color (and intrinsic transparencies) are as follows.

- Transparent
- Solid color

Transparent

Select the following to make the graphic object's color (and intrinsic transparencies) transparent.



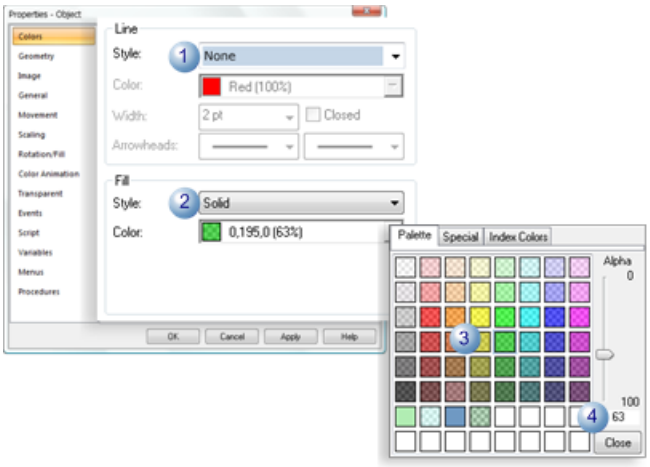
	Field	Select
1	Line Style	None (unless you want the image to have a border).
2	Fill Style	No fill

Solid Color

Any solid color can be assigned to the object's color (and intrinsic transparencies).

The color can be from 0% (transparent) to 100% opaque.

Select the following.



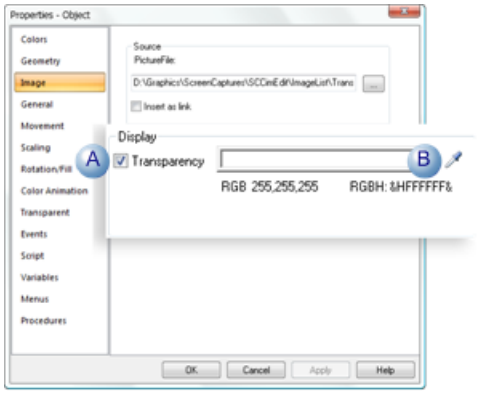
	Field	Select
1	Line Style	None (unless you want the image to have a border).
2	Fill Style	Solid
3	Palette	Any color
4	Alpha	Percent color opacity.

Step 4.2. Select the Color that will be Changed

The graphic object is placed on the CimEdit screen as soon as you select it. Depending on the size of the screen and/or the size of the object, you may have to move the dialog box to the side in order to have an effective view of the object.

Important: This feature does not apply to animated .gif files *(page 376)*.

1. Do the following when you can see the color you want to change.



A	Check Transparency.

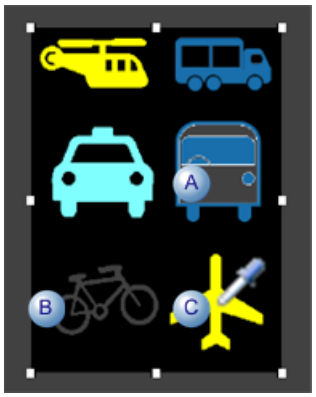
	Note: As soon as you check Transparency you will see that all the areas in that graphic object and embedded transparencies display what you selected as the color option. Selection, No line/no fill, Color
B	Click the eye dropper to the right of the **Transparency** field.

2. Drag the eye dropper onto the image color that should be changed.

   Example

   A .png file with an embedded transparency has been placed on the CimEdit screen.

   The color selection was No line/no fill.

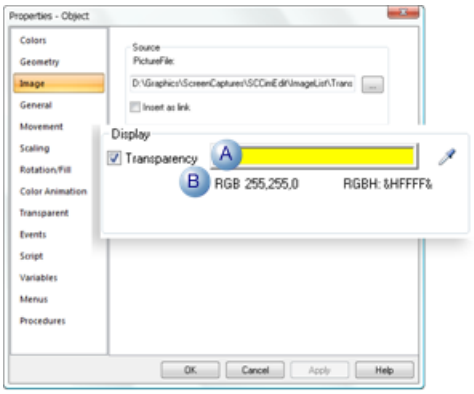


A	Areas that were white on the original graphic are transparent.
B	Areas that had an intrinsic transparency continue to be transparent.
C	The eyedropper selects the color that will be transparent (instead of white).

3. Click the left-mouse button when the eyedropper is over the color that will be selected.

The following occurs.

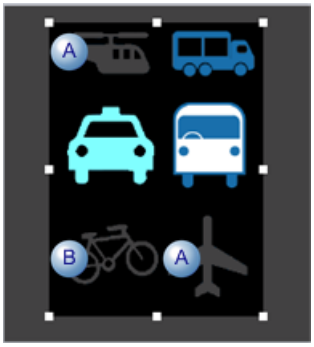
- In the Properties - Object dialog box



A	The color, e.g. yellow, is entered in the Transparency field.
B	RGB and RGBH change to (to reflect the color selection Example • RGB changes to 255, 255, 0 • RGBH changes to &HFFFF&).

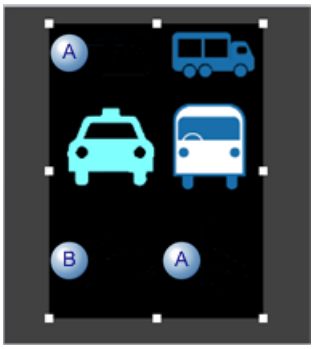
- On the graphic object (originally a .png file with and intrinsic transparency)

If Colors options are No Line/No fill



A	Areas that were the selected color, e.g. yellow, are now transparent.
B	The intrinsic transparency remains transparent.

If Colors options are No Line/Color (e.g. Black)



A	Areas that were the color selected in CimEdit, e.g. yellow, are now black.
B	The intrinsic transparency is now the selected color on the Colors page, e.g. black.

Class Object Graphic

The Workbench enables a developer to create a class that includes a class CimEdit screen with a specified default class object graphic, and possibly other class object (sometimes referred to as OpenObject™) graphics. An object designer can create objects based on the class–class objects.

You can then:

- Place a default class object graphic on a new CimEdit screen,
- Use the default or other available class object graphics,
- Do any other custom configuration for the screen you are configuring,
- Save the screen and
- Work with the class object's values in CimView.

The complexity of the class object screen depends on the configuration when the source object was created for the class.

1. Use any of the following methods to place a class object on a CimEdit screen.
   - Workbench to CimEdit.
   - Object Browser.
   - Properties dialog box.

   Workbench to CimEdit

   a. Select Objects in the Workbench left pane.
   b. Select the object you want to place on the CimEdit screen.
   a. Drag the object onto a new CimEdit screen.

   Result: An object that is linked to the class source object displays on the screen.

   Object Browser

   a. Click Drawings>Illustrations>ClassObject on the CimEdit Ribbon bar.

b. Place the cursor ⌜ where you want the object to be placed.
c. Click the left-mouse button.
d. The Select an Object browser opens.
e. Select the object.

Result: The object you select displays on the CimEdit screen.

Project dialog box

a. Change the class object graphic, if more than one object is available, as follows:
a. Right-click the class object graphic.
b. The Properties dialog box opens.
c. Select the Class Object tab.
d. Select an object from the **Graphic Name** field drop-down list.
a. Click **OK** or **Apply**.

Result: The selected object replaces the default object on the CimEdit screen.

2. (Optional) Modify the object placement or size.
   a. Resize or move the class object so it will appear the way you want during runtime.
   b. Place, create and configure other non-class objects on the screen.
   c. Save the CimEdit screen.

Object Form

Object Form

CimEdit provides you with several tools to adjust an object's form.

The tools include several ways to change its:

- Size.
- Shape.
- Display angle.

Note: The default unit of measure for screens and objects is points (pt). A point is a printer's basic unit of type measurement. There are 72 points in one inch.

There are many places in CimEdit where you are asked to enter location, size, or units as a number and a unit of measure. In addition to points, you can use any of the following for a unit of measure in CimEdit:

"	Inches
in	Inches

mm	Millimeters
cm	Centimeters
twips	1/20th of a point

If you enter any of these other units of measure, the measurement will be automatically converted to points by CimEdit.

For example, if you enter horizontal and vertical grid units of .5" in the Options dialog box for Tools, when you apply the change, the values will be displayed as 36 pts.

Change an Object Size

Change an Object Size

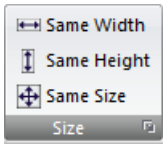
You can change the size of a closed frame container, group, or object by:

- Resize several items to the same dimensions.
- Resize a selected item.
- Scale a selected item.

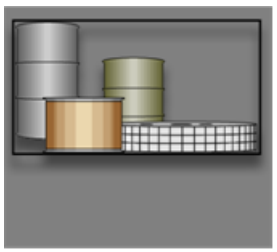
Important: When you change the appearance of a closed frame container, you also change the appearance of the groups and objects in that container.

Resize Several Items to the Same Dimensions

1. Select the objects that will be resized. The last object you select is the object to which the others will conform.
2. Click the button in the Arrange>Size group that will resize the objects according to your requirements.



Example: Four different size objects display on the CimEdit screen.

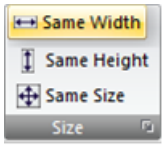


Re-sizing occurs as follows.

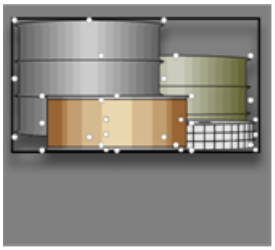
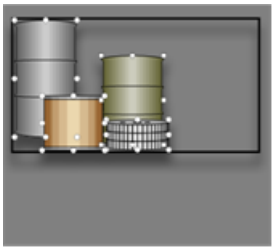
Same Width

Do one of the following.

- Click the Same Width button in the Arrange>Size group.

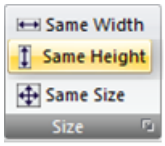


- Right-click the objects; select Size>Same Width on the Popup menu. When the object selected last (to make all objects the same width) is the:

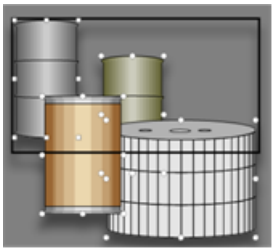
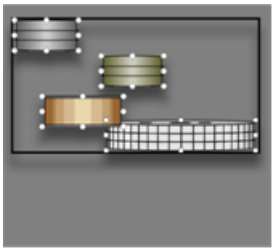
Widest	Most Narrow
Objects expand left to right.	Objects contract left to right.
	

Same Height: Do one of the following.

- Click the Same Height button in the Arrange>Size group.

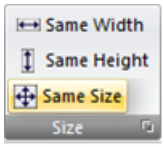


- Right-click the objects; select Size>Same Height on the Popup menu. When the object selected last (to make all objects the same height) is the:

Tallest	Shortest
Objects expand Top to Bottom	Objects contract Bottom to Top.
	

SameSize: Do one of the following.

- Click the Same Size button in the Arrange>Size group.



- Right-click the objects; select Size>Same Size on the Popup menu. Objects expand and/or contract according to the same width/same height directions so the size will conform to the last selected object.

Resize a Selected Item

Resize an item using any of the following.

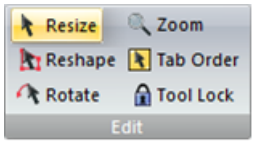
- Resize tool.
- Properties dialog box (precisely).
- Grid units

Resize Tool

The Resize tool can:

- Expand/contract an object width or height.
- Expand/contract an object width and height.

1. Click Resize in the Drawing>Edit group.



2. Select the object to resize.

   The object is surrounded by resize handles.

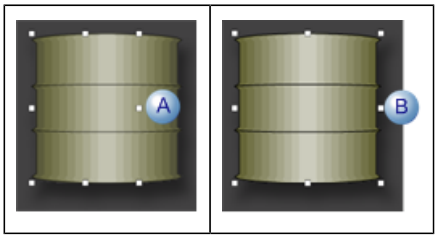
Expand/Contract an Object (Width or Height)

- Expand an object width or height.
- Reduce the object width or height.

Expand an Object Width or Height

3. Select a handle on the side, top or bottom of the object.
4. Drag the selected handle to its new expanded location.
5. Release the mouse button when the new size is correct.

   Result: The object size expands as follows.

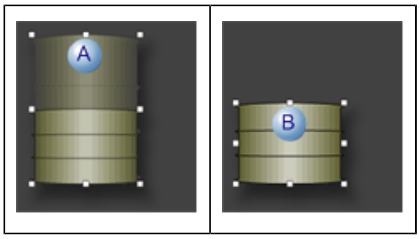


	Size Status	Object Handles
A	Expanding	The object handles remain in the original position. Note: Depending on the object selected, a expanded portion may be transparent.
B	Expanded	The handles move to surround the expanded object.

Reduce the Object Width or Height

6. Select a handle on the side, top or bottom of the object.
7. Drag the selected handle to its new contracted location.
8. Release the mouse button when the new size is correct.

   Result: The object size contracts as follows.



	Size Status	Object Handles
A	Contracting	• The object handles remain in the original position. • A transparent image of the original size displays.
B	Reduced	• The handles move to surround the contracted object. • The transparent image no longer displays.

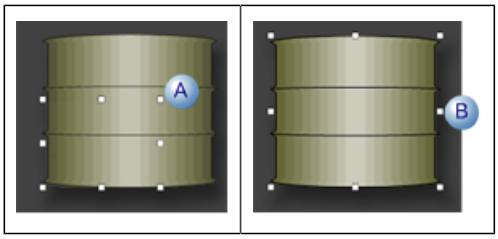
Expand or Contract an Object (Width and Height)

- Expand an object width and height.
- Reduce the object width and height.

Expand an Object (Height and Width)

9. Select a handle on one of the object's corners.
10. Drag the selected handle to its new expanded location.
11. Release the mouse button when the new size is correct.

Result: The object size expands as follows.

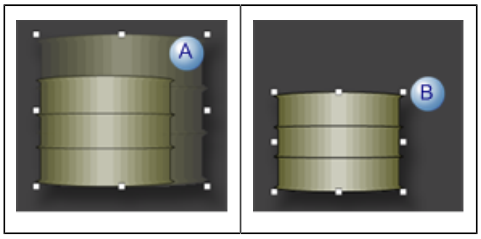


	Size Status	Object Handles
A	Expanding	The object handles remain in the original position.
B	Expanded	The handles move to surround the expanded object.

Reduce an Object Width and Height

12. Select a handle on one of the object's corners.
13. Drag the selected handle to its new contracted location.
14. Release the mouse button when the new size is correct.

Result: The object size contracts as follows.



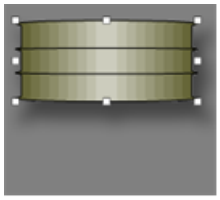
	Size Status	Object Handles
A	Contracting	• The object handles remain in the original position. • A transparent image of the original size displays.
B	Reduced	• The handles move to surround the contracted object. • The transparent image no longer displays.

Properties dialog box (precisely)

The object can be precisely sized in its Properties dialog box.

15. Select an object.

    Example

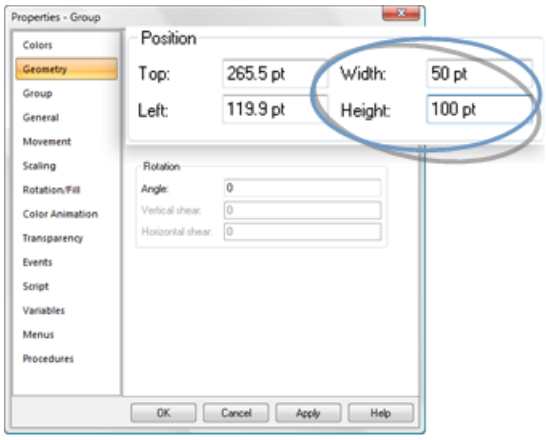
    

    Note: The object width and height are measured from object handle to handle.

16. Open the Properties dialog box.
17. Select Geometry

    The current dimensions display in the **Width** and **Height** fields.

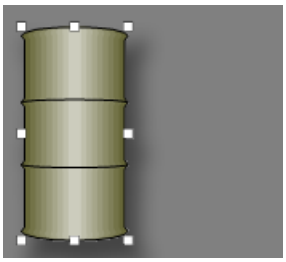
18. Enter a new **Width** and/or **Height** of the object.



19. Click OK.

The object is resized based on your entries.

Example



The re-sizing anchors are as follows.

Re-sizing	Anchor	The object	Direction
Width	Left	Expands	Left to right
		Contracts	Right to left
Height	Top	Expands	Top Down
		Contracts	Bottom Up

Note: If a shape has been rotated, changing its size will change its rotation angle and shearing.

Grid Units

You can use the techniques to resize an object using the grid as follows.

Select an object.

Re-sizing options are as follows.

- Do either of the following.
- Press **Shift+<arrow>** on the keyboard. The object expands or contracts in the specified direction by:

One grid unit when the grid is on *(page 42)*

One pixel when the grid is off

- Press **Ctrl+Shift+<arrow>** on the keyboard. The object expands or contracts in the specified direction by:

Two grid units when the grid is on *(page 42)*

Five pixels when the grid is off

When you use either of the above methods, the upper left corner of the object remains fixed. Each arrow key does the following:

Up	Decreases the height of the object.
Down	Increases the height of the object.
Left	Decreases the width of the object
Right	Increases the width of the object.

For example, if snap to grid is off and you press **Ctrl+Shift+<Down>** the height of the object increases in size by five pixels, instead of grid units.

- Press the **Ctrl** key and hold down the right mouse button; drag a side of the object to expand it or contract it.

The object expands or contracts in two directions, on the side you are dragging and on the opposite side.

Tip: Click Visible in the Nome>Grid group on the CimEdit Ribbon bar to display the grid. Press the Alt key to toggle Snap to Grid on and off.

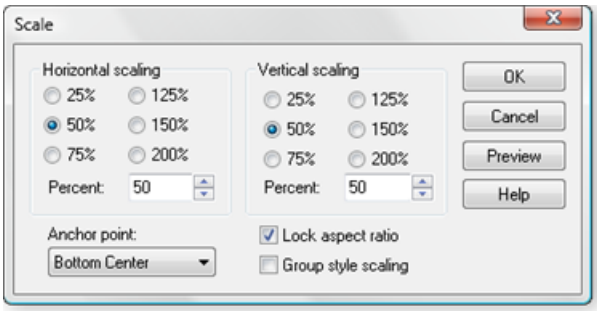
Scale a Selected Item

Object size can be increased or decreased by a specified percent. Scaling can be horizontal, vertical, or both.

Click the Dropdown button in the Arrange>Size group on the CimEdit Ribbon bar.

The Scale dialog box opens.

Scaling options are as follows.



Field	Description
Horizontal scaling	Radio buttons
	Percent
Vertical scaling	Radio buttons
	Percent
Anchor point	Choose one: • Top Left • Top Center • Top Right • Middle left • Center • Middle Right • Bottom Left • Bottom Center • Bottom Right
Lock aspect ratio	Do one of the following.
	- Check
	- Clear
Group style scaling	For a group of objects:
	- Check
	- Clear

Reshape Objects

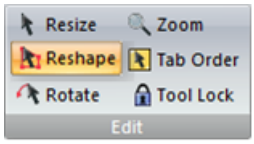
The Reshape tool is particularly useful for changing the shape of a polyline or polygon. Several keys on the keyboard and the CimEdit grid provide you with additional aids when you are reshaping the object.

- Reshape tool

- Object reshaping
- Handles deleted or added

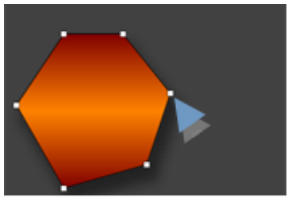
Reshape Tool

1. Click Reshape in the Drawing>Edit group on the CimEdit Ribbon bar.



2. Select the object to reshape.

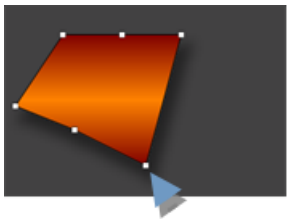
   The object is surrounded by reshape handles.



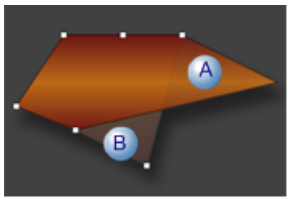
   Object Reshaping

   The object handles can be moved in any direction.

   As the object is reshaped, its handles remain connected to the same position that they were at on the original shape; that position may have changed location.



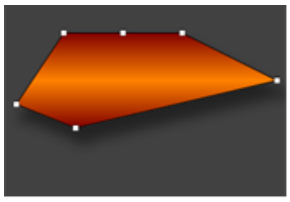
3. Drag any handle to continue reshaping the object.



A	A Transparent image displays how the object is being reshaped.
B	A contrasting shade of color displays the original object shape.

4. Release the left-mouse button.

Result: The object displays in its new shape.



Handles Deleted or Added

You can delete or add handles for polyline or polygon objects to further refine the reshaping.

Choose one of the following.

5. Press **Ctrl** on the keyboard.
6. Position the cursor at the edge of the object to be revised.

   For a **polyline** or **polygon**, the cursor turns to a crosshair.

   Do one of the following.

   - Click a vertex.

   a. The vertex is deleted.
   b. The line straightens between the vertex to the right and left of the deleted vertex.

   - Click a point in the line.

   A vertex is added to the line in the location that was clicked..

Change an Object's Display Angle

Change an Object's Display Angle

CimEdit provides you with complete versatility to rotate and flip items on the screen so they display in the angle and face the direction you want.

Options include:

- Precise object rotation.
- Visual object rotation.
- Techniques for flipping.

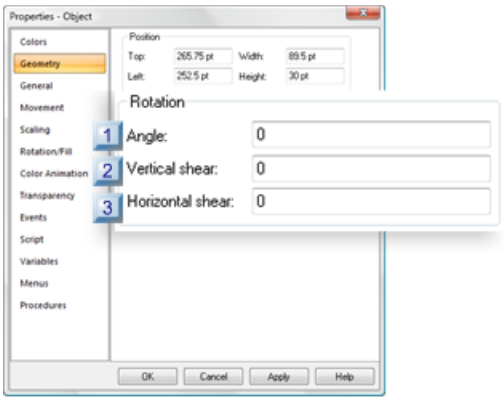
Precise Object Rotation (available for Lines, Text, Shapes or Groups)

Select Geometry in an object's Properties dialog box.

- Precise rotation configuration
- Precise rotation example

Precise Rotation Configuration

Rotation options are as follows.



rect 54, 116, 77, 138 *(page 139)*
rect 55, 91, 78, 113 *(page 138)*
rect 54, 71, 77, 93 *(page 138)*

1 *(page 138)*	Angle
2 *(page 138)*	Vertical shear
3 *(page 139)*	Horizontal shear

1	Angle

Available for:	Lines, text, shapes, groups	
Purpose	Rotates the object to the entered angle. Once you apply the angle the value you entered for:	
	Shape or Text	
	Group or Line	
Valid values:	Positive or negative • 0 and 360 degrees • -6.28139 and +6.28139 radians	

2	Vertical shear

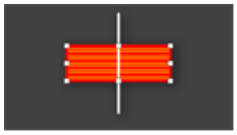
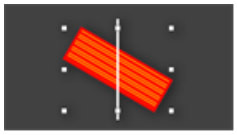
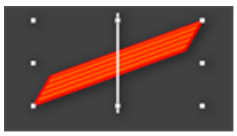
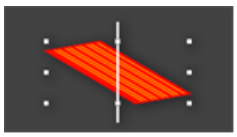
Available for:	Rectangles, ellipses, chords, pies
Purpose	Vertical shear slides the sides of the rectangle enclosing the shape in opposite directions
Valid values:	• -85 and + 85 degrees. • -1.48353 and +1.48353 radians.

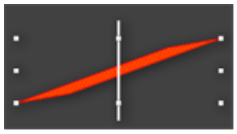
3	Horizontal shear

Available for:	Rectangles, ellipses, chords, pies
Purpose	Horizontal shear slides the top and bottom of the rectangle enclosing the shape in opposite directions.
Valid values:	Positive or negative • -85 and + 85 degrees. • -1.48353 and +1.48353 radians.

Precise Rotation Example

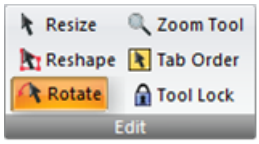
A rectangle displays rotation and sheer values as follows.

		Sheer
Example	Angle	Vertical
	0	0
	90	0
	150	0
	150	50
	150	0

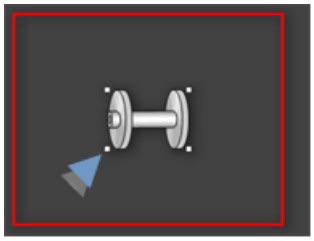
	150	50

Visual Object Rotation

1. Click Rotate in the Drawing>Edit group on the CimEdit Ribbon bar.



2. Click the object to be rotated.



   The selected object is surrounded by four handles.

3. Select one of the object's four handles.

   Rotation anchor options are:

   - Around one of the object's handles
   - Around the object's center point.

   Rotation Anchor: An Object's Handle.

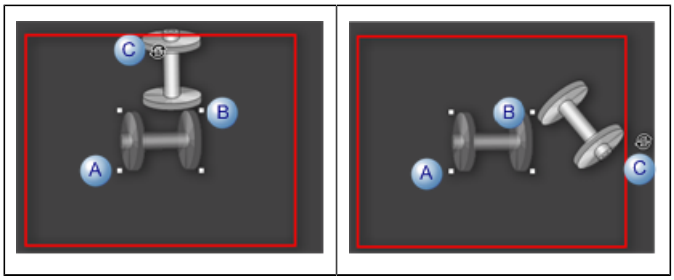
4. Select one of the object's four handles.

5. Drag the handle.

   Result: The object rotates around the diagonally opposite handle in the direction you drag the cursor.

   Example

   A wire spool is selected to rotate from left to right around its top, right handle.



A	The lower left handle is selected. A transparency of the original position displays while the object is being rotated.
B	The spool rotates around the top, right handle.
C	The selected handle displays as a rotation cursor.

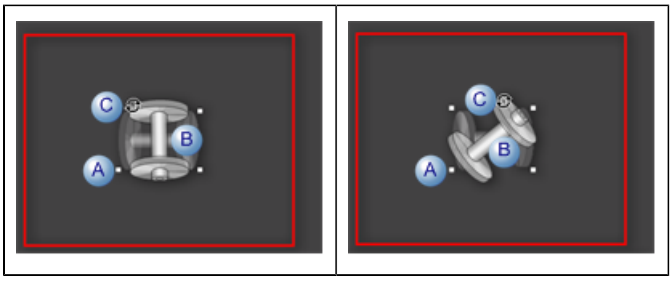
Rotation Anchor: An Object's Center Point

6. Select one of the object's four handles.
7. Hold down the Ctrl key while you drag the handle.

The object rotates around its center point in the direction you drag the cursor.

Example

A wire spool is selected to rotate from left to right around its center point.



A	The lower left handle is selected. A transparency of the original position displays while the object is being rotated.
B	The spool rotates around its center.
C	The selected handle displays as a rotation cursor.

Tip: Hold down the **Shift** key to constrain the rotation to 45-degree increments.

Techniques for flipping

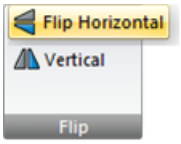
A selected object can be quickly flipped horizontally or vertically.

- Flip horizontal

- Flip vertical

Flip Horizontal

Click Flip Horizontal in the Arrange>Flip group on the CimEdit Ribbon bar.



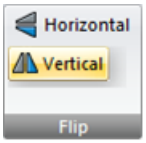
The selected object flips horizontally within the same x,y coordinates.

Example

Original	Flipped Horizontal.	Flipped Horizontal

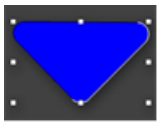
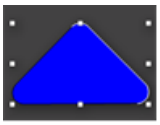
Flip Vertical

Click Flip Vertical in the Arrange>Flip group on the CimEdit Ribbon bar.



The selected object flips vertically within the same x,y coordinates.

Example

Original	Flipped Vertical	Flipped Vertical
		

Object Layout

Object Layout

As you experiment with what objects to use on your CimEdit screen, there are several ways to specify how and where they will be placed and to modify their size and shapes.

You can:

- Create
  - Single objects.
  - A set of several objects.
  - A group of objects.

- Move objects around the screen and:
  - In front of/to the back of other objects.
  - Aligned with other objects.
  - Adjust the spacing between objects.
  - Change objects tab order.

Object Sets

More than one object can make up a temporary set in CimEdit. The set breaks up when the mouse clicks on anything other than the set.

- Select object sets.
- Dominant objects.

Select object set methods

- Quick

Select

Hold down the left-mouse button and sweep a rectangle over the objects that you want to include in the set.

De-select

Click the mouse button anywhere on the screen.

- Precise

Select

Hold down the Shift key and click on each object that you want to include in the set.

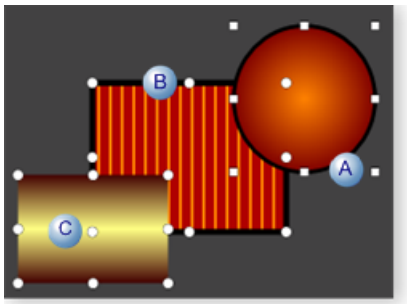
De-select one object

Click a selected object while holding down the Shift key.

Dominant object

When you select a set of objects the last object selected becomes the dominant object.

1. The dominant object has square handles while the other objects have round handles.



A	Square handle for a dominant object.
B	Round handle for other objects.
C	Handles display through an object that is in front of another object.

2. When you align objects, all other objects in the set align to the dominant object.
3. When you size objects, all other objects in the set take their new size from that of the dominant object.

Tip: To change the dominant object:

4. Hold down the Shift key.
5. Click once on the object to de-select it.
6. Click the object again to select it. The object just selected is now the dominant object.

Groups of Objects

Groups of Objects

You can take any one object and combine it with other objects to form a Group. The group remains as a group permanently (or for as long as you want).

If you want two or more objects to be connected so you can manipulate them, e.g. resize them, as one object, you can group them. In addition, you can split a group back into its individual components by ungrouping it.

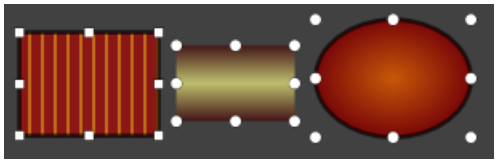
Five basic procedural steps you can follow when you work with a group are:

Step 1 *(page 145)*	Create a group.
Step 2 *(page 146)*	Configure the group's properties.
Step 3 *(page 146)*	Select an object in a group
Step 4 *(page 150)*	(Optional) Ungroup a group

Step 1. Create a Group

1. Hold down the Shift key.
2. Click the objects you want to include in the group.

   A set of objects is created.

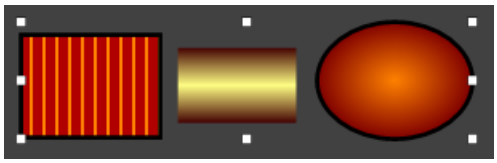


3. Do one of the following to create the group.
   - Click the Group button

 in the Arrange>Group group on the CimEdit Ribbon bar.
   - Right-click the set; select Group on the Popup menu.

   A group is created.

   Note: The handles surround the group the same way they surround one object.

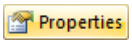


   At any time, you can split the group into its individual objects

   When you create a group, you can still configure properties for its individual objects. However, you can make the objects uniform in their border and interior attributes by creating group properties that override the individual properties.

Note: Several objects in the Object library are actually groups.

Step 2. Configure the Group's Properties

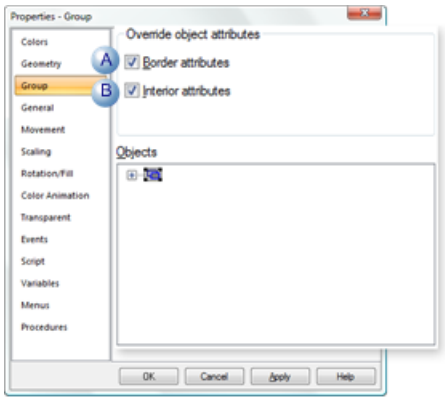
1. Do one of the following.
   - Double-click the group object.
   - Click the Properties button 

In the Home>Properties group on the CimEdit Ribbon bar.
   - Right-click the group; select Properties on the Popup menu.

   The Properties - Group dialog box opens.

2. Select the Group tab.

3. (Optional) check either of the following.



	Check box	Description
A	Border attributes	Borderlines configured for the group override borders configure for the objects in the group.
B	Interior attributes	Fill properties configured for the group override fills configured for the objects in the group.

4. Configure other properties for the group the same as you configure them for objects.

Note: Events, procedures and variables configured at the group level are available to the objects within the group.

Step 3. Select an Object in a Group

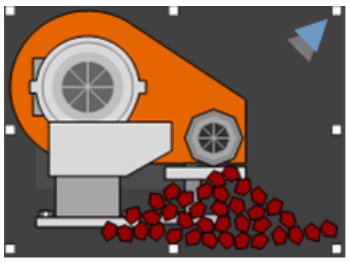
Step 3. Select an Object in a Group

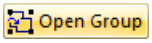
Methods to select an object in a group for configuration are as follows.

Option 3.1 *(page 147)*	Configure an object in an open group.
Option 3.2 *(page 149)*	Select an object through the Properties Group dialog box.

Option 3.1. Configure an Object in an Open Group

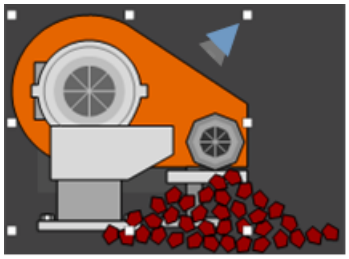
1. Select a group to open.



2. Do one of the following.
   - Click the Open Group button 

in the Arrange>Group group on the CimEdit Ribbon bar.
   - Right-click the group; select Open group on the Popup menu.

   The selected group is now open; an object in the group can now be selected.



3. Navigate around the group

guide: Guidelines

- You can only access objects in an open group that are on the same nested level as a selected object.

For example, if you have opened a nested group, you will only be able to open that group or objects in that group. You will have to close the nested group to select other objects in the next level up group.

- The Ribbon bar buttons are enabled/disabled based on what type of object is selected in the group, e.g. if a nested group is selected Ungroup and Open Group buttons on enabled on the Ribbon bar and display on the Popup menu.

The CimEdit title bar displays the name or the open group next to the screen name.

Example

- A top-level group is open.

The CimEdit title bar displays the following.

- A nested group is selected within the top-level group.

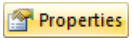
A	Screen name
B	Top-level group name.

The CimEdit title bar displays the following.

A	Screen name
B	Nested-level group name.

- A toolbar that displays at the left of the CimEdit screen provide the following buttons.

A	Frame buttons are enabled if a frame that is included in the group has been opened.
B	Close Group button closes the selected group.

4. Do one of the following when the object you want to configure is selected.
   - Double-click the object.
   - Click the Properties button 

In the Home>Properties group on the CimEdit Ribbon bar.
   - Right-click the object; select Properties on the Popup menu.

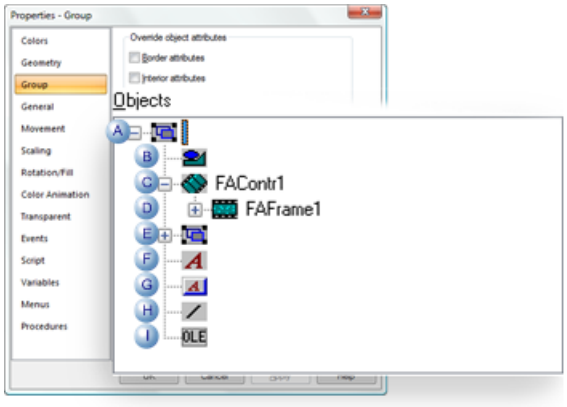
The Properties dialog box opens for the selected object.

5. Close the Properties dialog box when you have completed your configuration.

You can now navigate to an select another object in the group to configure or close the group to move up to the next level.

Option 3.2. Select an Object through the Properties - Group Dialog Box

1. Select the Group tab in the Properties - Group dialog box.
2. Expand the tree in the **Objects** box.
3. Select what objects are in the group as follows:
   a. (If it is not displaying) select the Group tab in the Group Properties dialog box.
   b. Click the + to expand nested groups.
   c. Click the – to collapse nested groups.
   d. Review the objects which may include:



A *(page 145)*	Group	
B *(page 69)*	Shape or object	
C *(page 410)*	Frame container	
D *(page 410)*	Frame object	
E *(page 145)*	Group within a group	

F *(page 98)*	Text object	
G *(page 98)*	Button object	
H *(page 70)*	Line	
I *(page 577)*	OLE or ActiveX control	

4. Open an object's properties dialog box:
    a. Select the object to be configured.
    b. Click the right mouse button.
    c. Select Properties from the popup menu.

Step 4. Ungroup a Group

You can break a group apart at any time. However, if you have included configuration that applies to the group, that configuration may be lost.

Do any of the following to "ungroup" the group.

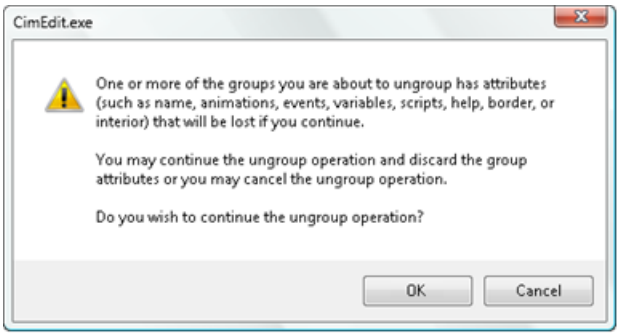
- Click the Ungroup button in the Arrange>Group group on the CimEdit Ribbon bar.
- Right-click the group; select Ungroup on the Popup menu.

Result: One of the following happens.

- Objects in the group immediately become stand-alone objects.

Note: This occurs if the group is a simple group with no configuration that might effect the group's functionality.

- A message box opens to warn you, if you have configured attributes, such as events, for the group object, that the configuration may be lost.



Click one of the following.

OK	The group is ungrouped. **Note:** Nested frames and groups continue to be frames and groups; they simply are no longer nested in the group that is being broken apart.
Cancel	The group remains a group.

When a group is ungrouped, instead of group handles, you will see the individual object handles on the screen.

Move Objects on a CimEdit Screen

Move Objects on a CimEdit Screen

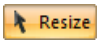
CimEdit provides you with several ways to move and position an object on the screen.

- Move a selected object.
- Object position in front of/back of other objects.
- Objects spacing configured on a CimEdit screen.
- Objects aligned on a CimEdit screen.

Move a Selected Object

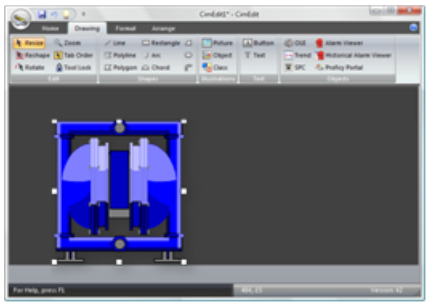
- Select (resize) tool.
- Properties dialog box (precisely).
- Grid units

Select (resize) tool

1. Click **Resize** 

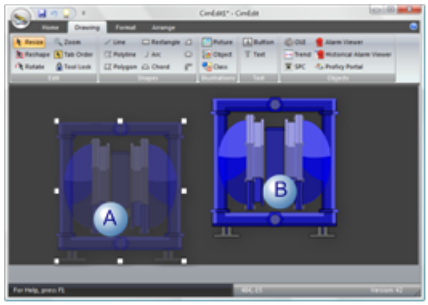
in the Drawing>Edit group on the CimEdit Ribbon bar.
2. Click the object to be moved.

   The object is selected.



3. Hold the left-mouse button down while you drag the object to another position.

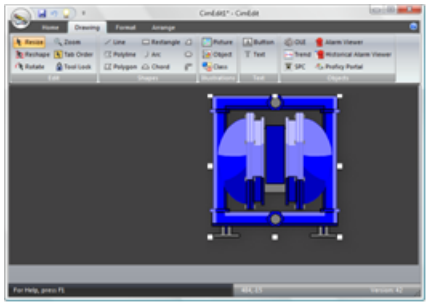
   The following happens.



A	A "phantom" object displays in the original position.
B	The object moves as you drag it.

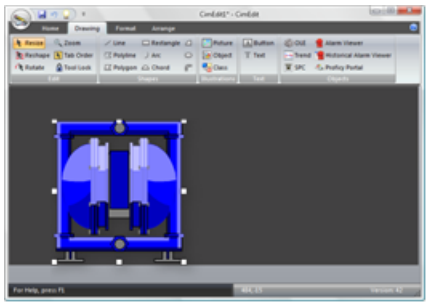
4. Release the left-mouse button where you want the object to be located.

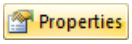
   Result: The object is placed in its new location; the "phantom" object no longer displays.



   Properties dialog box (precisely)

5. Select the object.

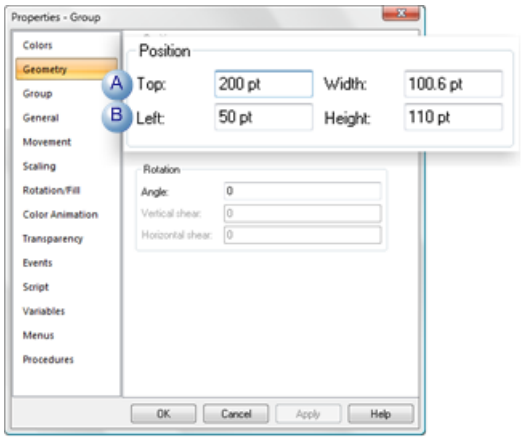


6. Do one of the following.
   - Click Properties  in the Home>Properties group on the CimEdit ribbon bar.
   - Right-click the object; select Properties on the Popup menu.

   The Properties dialog box opens.

7. Select Geometry.

8. Enter the following.



	Field	Description
A	Top	Number of points from the bottom of the screen defines where the top of the object will be located.
B	Left	Number of points from the left of the screen defines where the left of the object will be located.

Grid Units

Choose one of the following.

Method 1. Use the keyboard

Press an arrow key to move the object in the specified direction by one grid unit when the grid is on or one pixel when the grid is off.

Method 2. Use the keyboard

Press Ctrl+<arrow> to move the object in the specified direction by:

- Two grid units when the grid is on
- Five pixels when the grid is off.

Tip: You can toggle the grid on or off temporarily by holding down the **Alt** key while pressing the arrow keys. For example, if the grid is on, and you press **Alt+<arrow>** the object will move one pixel to the right.

Object Position in Front of/In Back of Other Objects

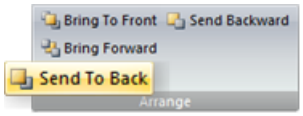
You can use the overlay tools on the Layout toolbar to adjust the order in which objects are overlaid on the screen. You can bring an object to the front, or one place forward, or send it to the back, or one place back, in a set of objects.

- Send one object to the back.
- Bring one object to the front.
- Bring one object forward one place.
- Send an object back one place.

Send one object to the back

1. Select the object you want to send to the back.

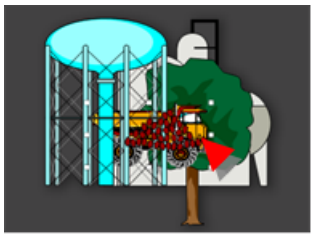
2. Do one of the following.
   - Click Send to Back in the Arrange>Arrange group on the CimEdit Ribbon bar.



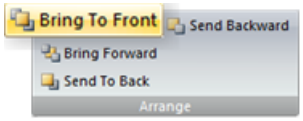
   - Right-click the object; select Arrange>Send to Back on the Popup menu.

Bring one object to the front

3. Select the object you want to bring to the front.



4. Do one of the following.
    - Click Bring to front in the Arrange>Arrange group on the CimEdit Ribbon bar.

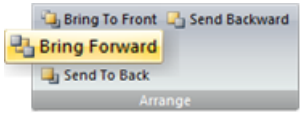


    - Right-click the object; select Arrange>Bring to Front on the Popup menu.

Bring one object forward one place

5. Select the object you want to bring forward one place.

6. Do one of the following.
    - Click Bring Forward in the Arrange>Arrange group on the CimEdit Ribbon bar.

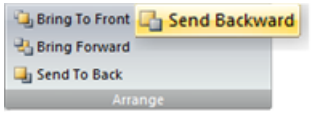


    - Right-click the object; select Arrange>Bring Forward on the Popup menu.

Send an object back one place:

7. Select the object you want to send back one place.

8. Do one of the following.
    - Click Send Backward in the Arrange>Arrange group on the CimEdit Ribbon bar.



- Right-click the object; select Arrange>Send Backward on the Popup menu.

The objects have moved forward and back to their correct location.

Objects Spacing Configured on a CimEdit Screen

Click each object that will be spaced while you hold down the Shift key.

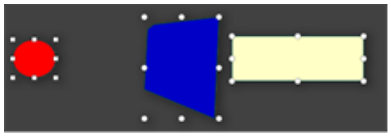
Spacing options are as follows.

- Objects spaced horizontally.
- Objects spaced vertically.

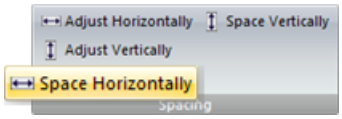
Objects Spaced Horizontally

Even Horizontal Spacing

1. Click each object that will be spaced while you hold down the Shift key.



2. Do one of the following.
   - Click Even Horizontally in the Arrange>Spacing group on the CimEdit Ribbon bar.



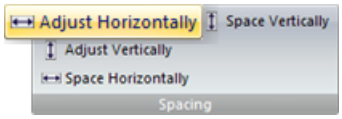
   - Right-click the objects; select Spacing>Space Horizontally on the Popup menu.

   Result: The objects are spaced horizontally evenly between the far left and right object sides.

   Adjust Horizontal Spacing

3. Click each object that will be spaced while you hold down the Shift key.

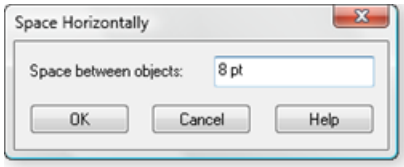
4. Do one of the following.
   - Click Adjust Horizontally in the Arrange>Spacing group on the CimEdit Ribbon bar.



   - Right-click the objects; select Spacing>Adjust Horizontally on the Popup menu.

   The Horizontal Space dialog box opens.

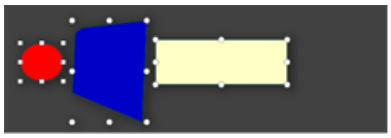
5. Enter the number of points for the space between each of the objects.



6. Click OK.

   Result: The objects move:

   - To the left to reduce the total space size.

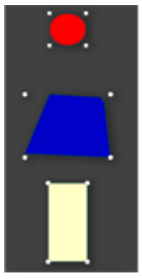


   - To the right to increase the total space size.

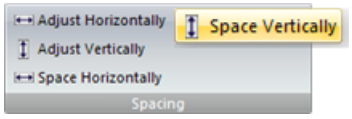
   Objects Spaced Vertically

   Even Vertical Spacing

7. Click each object that will be spaced while you hold down the Shift key.



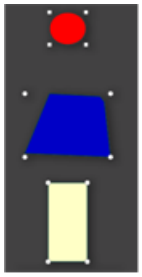
8. Do one of the following.
   - Click Space Vertically in the Arrange>Spacing group on the CimEdit Ribbon bar.



- Right-click the objects; select Spacing>Space Vertically on the Popup menu.

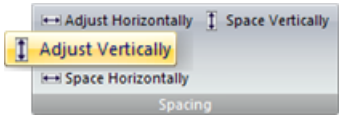
Result: The objects are spaced evenly between the top and bottom end points.

For example, if three objects are spaced evenly, the midpoint of the middle object is the midpoint between the top and bottom ends.



Adjust Vertical Spacing

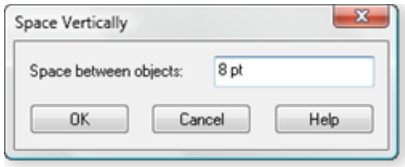
9. Click each object that will be spaced while you hold down the Shift key.
10. Do one of the following.
    - Click Adjust Vertically in the Arrange>Spacing group on the CimEdit Ribbon bar.



    - Right-click the objects; select Spacing>Adjust Vertically on the Popup menu.

    A Space Vertically dialog box opens.

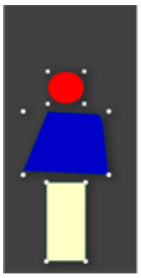
11. Enter the number of points for the space between each of the objects.



12. Select Horizontal from the extended menu.
13. Click OK.

The objects move:

- To the bottom to reduce the total space size.



- To the top to increase the total space size.

Objects Aligned on a CimEdit Screen

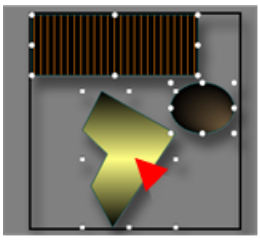
When you have placed two or more objects in the workspace and need align them, use CimEdit's Layout toolbar and Format menu options to automatically align the objects.

Hold down the shift key to select the objects that will be aligned.

All the selected objects will be aligned to the last selected item.

Example

A polygon in a group of three objects is selected last.

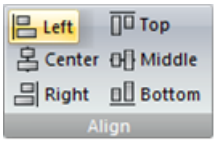


Alignment selections are as follows.

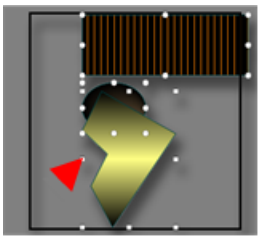
- Left align
- Center align
- Right align
- Top align
- Middle align
- Bottom align

Left Align

Click Left in the Arrange>Align group.

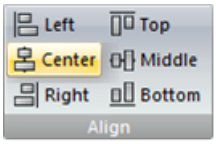


Result: The objects' left side aligns with the last selected object's left side.

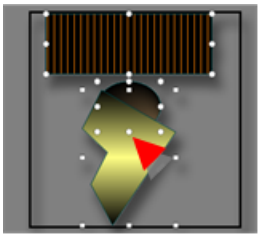


Center Align

Click Center in the Arrange>Align group.

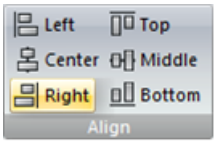


Result: The objects' center aligns with the last selected object's center.

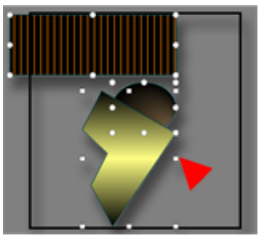


Right Align

Click Right in the Arrange>Align group.

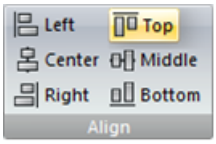


Result: The objects' right side aligns with the last selected object's right side.

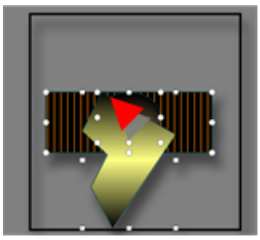


Top Align

Click Top in the Arrange>Align group.

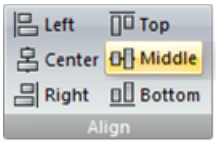


Result: The objects' top aligns with the last selected object's top side.

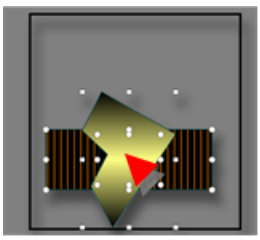


Middle Align

Click Middle in the Arrange>Align group.

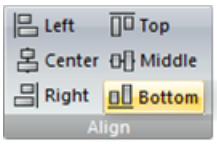


Result: The objects' middle aligns with the last selected object's middle.

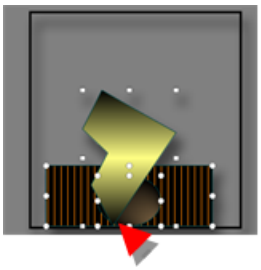


Bottom Align

Click Bottom in the Arrange>Align group.



Result: The objects' bottom aligns with the last selected object's bottom.



Change Objects Tab Order

- Tab order overview.
- Tab order identification.
- Tab order configuration.

Tab Order Overview

Users can tab to action objects during runtime in CimView.

- Tab order criteria

Action objects have to meet one or more of the following criteria:

- The object is a Text object with a setpoint action.
- One or more of the following events are configured for the object:
- Mouse Down
- While Mouse Down
- Mouse Up
- A Slider setpoint is defined for the object.
- The object is an OLE object that has a primary verb. However the verb cannot modify the object (for example, play the object, for an embedded sound).

CimEdit keeps track of how many objects can be included in the tab order. If you attempt to

- Tab range

The tab range for a screen is 0 - the number that equals the total number of objects that meet the tab order criteria.

Note: A group is counted as one object in determining the range even if it includes more than one object that CimView users can tab to.

- Action objects found in CimView

Users in CimView can find action objects by:

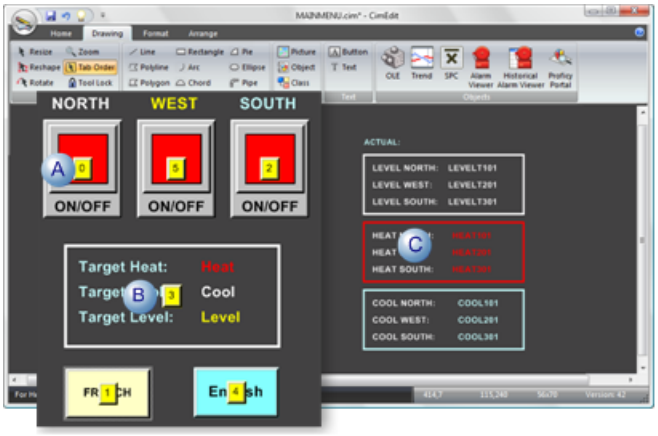
- Using the tab and arrow keys to highlight the objects, then using the Enter key to invoke the actions on the highlighted object.
- Moving the mouse around the screen. When the user moves the mouse over one of these objects, the object is highlighted with a rectangle.

Tab Order Identification

The order in which you create objects that meet the tab order criteria *(page 162)* determines the initial tab order.

Click Tab Order in the Drawing>Edit group on the CimEdit Ribbon bar.

Objects are identified as follows.



A	Numbers display on objects that meet the tab order criteria. Each number identifies its object's tabbing order.
B	A single number displays on each group with objects that meet the tab order criteria. That number identifies the group's tabbing order. The group must be opened to see the tab order for each of its objects that meet the criteria.

C	Objects that do not meet the tab criteria are ignored.

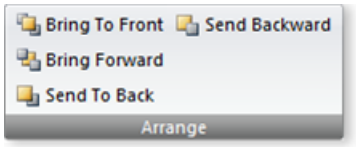
Tab Order Configuration

When you change the order of one object on the screen, the order of the other objects will automatically be adjusted to accommodate your selection.

Do either of the following to change the tab order of a selected object.

Arrange Group

1. Select the Arrange>Arrange group on the CimEdit Ribbon bar.



2. Click one of the following to change the tab order of the selected object.

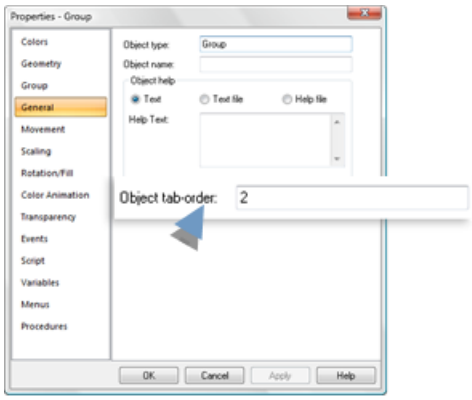
Button	Tab number changes to:	Tabbing order
Bring To Front	0	First.
Bring Forward	One number lower.	One tab sooner than the position.
Send to Back	Highest number in the tab range.	Last
Send Backward	One number higher	One tab after the current position.

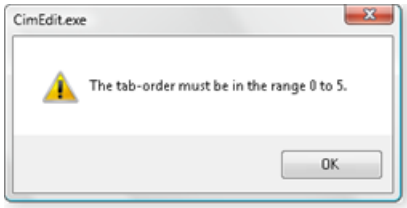
Properties Dialog Box

3. Open the selected object's Properties dialog box.
4. Select **General**.

   The object's tab order displays in the **Object tab-order** field.

5. (Optional) change the tab order, as follows.



Entry	Object's Tabbing Order
0	First.
Lower number	Moves toward the beginning of tabbing as many places as the number is decreased.
Higher number	Moves toward the end of tabbing as many places as the number is increased. **Note:** If you attempt to enter a number that is higher than the screens range, a message reports the range of numbers that can be entered. 

Note: While you are using the Tab Order tool, the Bring to Front, Send to Back, Bring Forward, and Send Backward operations only affect the tab order of the object. They do not affect the position of the object in relation to other objects in the screen.

Object Color

Object Color

CimEdit lines can display in selected colors and styles. Polygons, rectangles, ellipses, chords, pies, buttons, and text objects support several fill styles:

- Color picker.
- Color palettes.
- Borders and lines.
- No fill for an object
- Solid fill
- Pattern fill

- Gradient fill

Display the Colors Section in the Properties Dialog Box

1. Select an object.

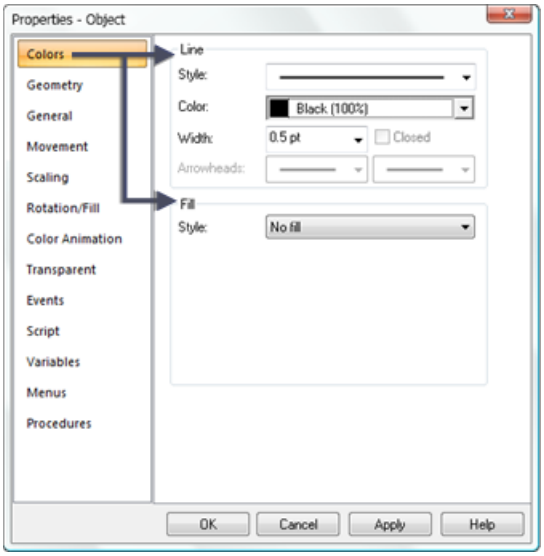
   Note: The object can be a stand-alone object, a component in a group or the screen.

2. Do one of the following.
   - Right-click the object; select Properties on the Popup menu.
   - Select the Home tab on the CimEdit Ribbon bar; click the Properties button in the Properties Group.

   The Properties dialog box opens for the selected object.

3. Select Colors.

Colors can be selected for a line or fill. The line can be any of the several line types or a border.

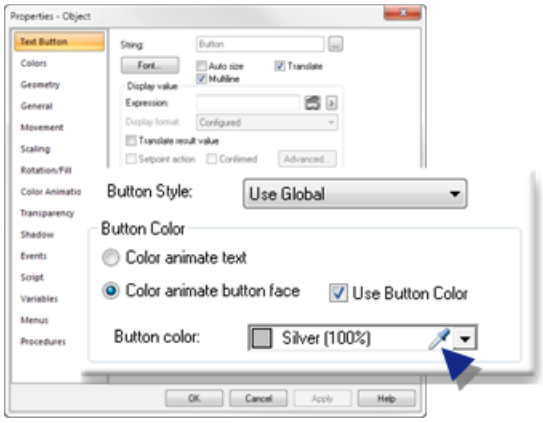


Color Picker

CimEdit provides an eyedrop color picker that can select a color anywhere on available monitors and apply it to a selected **color** field.

1. Select an object that will have its color changed.

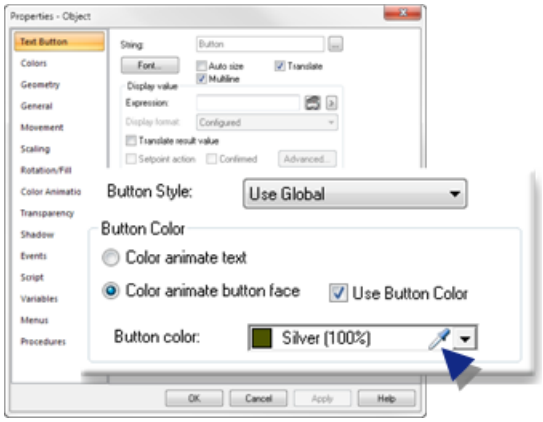
2. Open the Properties - Object dialog box.
3. Select the appropriate **color** field.
4. Click the eye dropper in the object's **color** field.



5. Drag the eye dropper anywhere on any available monitor to find the color that should be used.

6. Release the mouse button when the color is found.

   The selected color displays in the **color** field.



7. Close the Properties - Object dialog box.

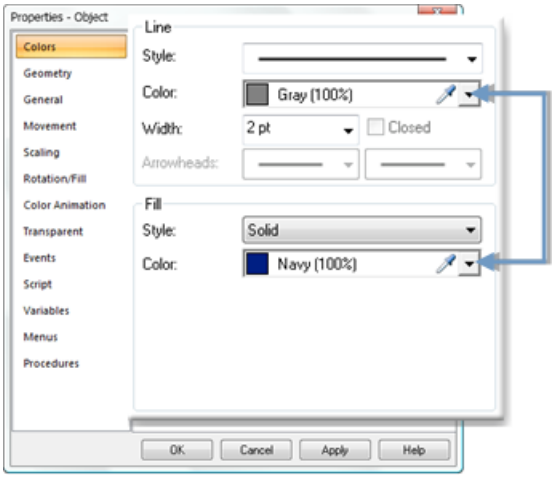
The object displays the selected color.

Color Palettes

Color Palettes

The CimEdit color palette is the basic building block applying color to objects in CimEdit. You can open a color palette in any field that requires a color specification. If you are applying a more complex color combination, e.g. a gradient or a pattern, you will start with selecting single colors in separate color fields. Each color field will have its own color palette.

Click the Down arrow to the right of any color field.



The CimEdit Color palette opens.

The CimEdit color palette provides a wide variety of options for selecting a single color.

The options are accessed through any of three tabs.

Each of the three tabs enables you to provide color specifications that CimEdit will find from different sources.

rect 91, 17, 120, 40 3. Index Colors Tab in the CimEdit Color Palette *(page 178)*
rect 48, 16, 77, 39 2. Special Tab in the CimEdit Color Palette *(page 177)*
rect 8, 16, 37, 39 1. Palette Tab in the CimEdit Color Palette *(page 169)*

	Palette Tab	CimEdit finds the colors from:
1 *(page 169)*	Palette tab in the Color palette	RGB color designations
2 *(page 177)*	Special tab in the Color palette	System Ambient colors
3 *(page 178)*	Index Color	An RGB.dat file that lists Index numbers with their assigned colors.
4 *(page 191)*	Fixed color palette options	Color picker options.

1. Palette Tab in the CimEdit Color Palette

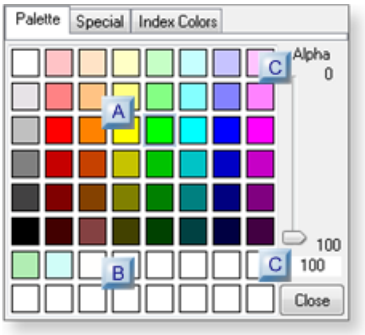
The Palette tab enables you to select and create colors that are based on RGB values.

Each RGB color is made up of a balance of red, green, and blue luminosity values ranging from 0 to 255.

Example

Red	Green
0	0
128	0
255	255

The Palette tab provides the following options for this selection.



rect 156, 146, 182, 174 *(page 175)*
rect 155, 26, 181, 54 *(page 175)*
rect 59, 153, 86, 183 *(page 171)*
rect 58, 52, 91, 81 *(page 170)*

A *(page 170)*	Basic color selection
B *(page 171)*	User defined color
C *(page 175)*	Color transparency

1. Basic Color Selection

You can pick a basic color very quickly as follows.

Note: You can also quickly make the color transparent.

1. Select the Palette tab.
2. Click any one of the available colors.

   Available colors include the following.

1	48 fixed colors

2	16 user definable custom colors

Result: The palette closes and the selected color displays in the color field.

Note: When you move the mouse pointer over one of the colors, the following information displays.

- Color name, or the RGB values for the color.
- Percent color transparency.

a. User Defined Color

Open the (Advanced) Color palette

3. Click a basic or custom color.

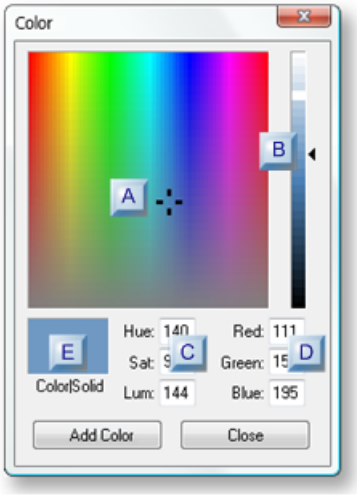
4. Do one of the following.
   - Click a blank custom color box.
   - Right-click any custom color box; select Define Custom Color on the Popup menu.

Note: The custom color will display in the selected box.

Result: The (Advanced) Color palette opens.

Define a custom color

5. Change one or more of the following until you have created the color you want.



rect 29, 204, 57, 235 *(page 175)*
rect 179, 206, 207, 237 *(page 174)*
rect 104, 206, 132, 237 *(page 173)*
rect 161, 79, 189, 110 *(page 172)*
rect 67, 110, 95, 141 *(page 172)*

A *(page 172)*	Color picker
B *(page 172)*	Luminosity picker
C *(page 173)*	Hue/Sat/Lum
D *(page 174)*	Red/Green/Blue
E *(page 175)*	Color|Solid

A	Color picker

The Color picker can be moved around to quickly change the color.

CimEdit automatically changes the hue, saturation, red, green and blue values as the picker is moved.

B	Luminosity picker

The luminosity picker can slide up and down to change the amount of light (absence of black) in the color.

CimEdit automatically changes the luminosity, red, green and blue values as the slider is moved.

C	Hue/Sat/Lum

The hue, saturation and luminosity can be manually adjusted.

CimEdit automatically changes relevant color values.

Field	Description
Hue	Balance of red, green, and blue that you want in the **Hue** field. This balance is calculated against the values that are already in the **Color** fields. The color and color picker position change as the Hue value changes.
Sat	Vividness (amount of gray) the color specified by the balance of the three basic colors.

Field	Description
	0
	240
	The color and color picker position change as the Hue value changes.
Lum	Exact value that specifies the amount of light in the color in the **Lum** field.
	0
	240
	The color and luminosity picker position change as the Hue value changes.

D	Red/Green/Blue

Color fields hold the color's RGB index value. Changing the value changes the selected color's proportion in the mix.

The color, color picker position and luminosity picker position change to reflect any changed value.

Overview	Description
Zero for all colors	Black

Overview	Description
Zero for one color	Black instead of the selected color is included in the mix
255 for all colors	White
255 for one color; zero for two colors	Pure red, green or blue for the color with the 255 value.

E	Color|Solid

If your monitor does not support a selected color CimEdit displays the exact color and the solid color that is closest to it in the **Color|Solid** box.

Selection possibilities are:

- The exact color is the same as the solid color. The box will be entirely solid.
- The exact color is dithered and is different from the solid color. The dithered color appears on the left.
- The solid color that is closest to the exact color appears on the right.

Double-click the Solid side of the **Color|Solid** box to select the solid color. The box changes to the solid color.

a. Do one of the following.
- Click Close.

The Color dialog box closes, re-displaying the palette. No color is selected.

- Click Add Color to select a color.

If the exact and solid colors are different in the Color|Solid box and both display, the exact color will be selected. If only the solid color displays. The solid color will be selected. The selected color is added to the user defined colors in the Palette.

1. Color Transparency

When an object is in front of another, it might be important to make it transparent so the object in the back will also be seen.

You can make any selected color transparent.

1	Click any color on the Palette tab in the CimEdit Palette. The color can be: • Basic • Custom
2	Do one of the following. • Slide the Alpha slide on the right of the palette up/down to select the percent transparency. • Enter a value that represents the percent transparency in the Alpha field. **Note:** The lower the number, the more transparent the color will be.

The color transparency displays as follows.

A custom color was clicked.

The following displays.

1	The custom color displays the selected transparency.
2	Custom colors that were not selected display the transparency that they were set to.
3	.All of the basic colors are opaque.
4	The **Alpha** field displays the selected color's transparency level.

A Basic color was clicked.

The following displays.

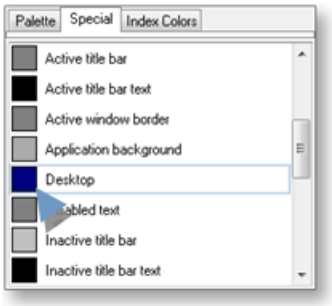
1	All of the basic colors display the selected transparency.
2	Custom colors display the transparency that they were set to.
3	The **Alpha** field displays the selected color's transparency level.

2. Special Tab in the CimEdit Color Palette

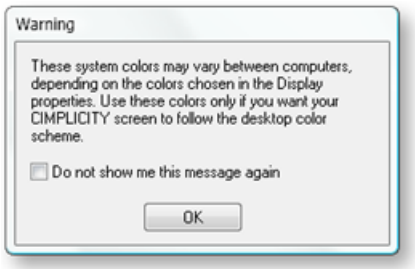
1. Select the Special tab in the Color Palette.

   Options listed on the Special tab display ambient colors that correspond to the selected Windows display colors.

2. Select a Windows color/item .



   Because the system special effects follow the color scheme selected through the Windows Display Properties they may vary from computer to computer. If you pick one of these colors, which are controlled by the Windows selection, you are warned that the color may change from computer to computer depending on the color scheme the user selects for that computer.



3. Click OK.

4. Result: The color/item that was selected in the Windows Display Properties dialog box is entered in the Color field.

3. Index Colors Tab in the CimEdit Color Palette

3. Index Colors Tab in the CimEdit Color Palette

An Index number associates a specified RGB color with an RGB number. When the index number is associated with an object, its color is applied to the object. If the RGB color that is associated with the Index number changes, the color applied to that object changes.

Example

Index No.	Red	Green
1	255	0

The Index Colors tab provides easy access to RGB Index color specifications. You can also open an RBG Editor through the tab to identify, and create if necessary, colors for your system.

3.1 *(page 178)*	Select colors on the Index Colors tab.
3.2 *(page 179)*	Open the RGB Editor.
3.3 *(page 180)*	Index custom colors and entries.

You can open an RGB Editor through this tab and add additional index numbers and assignments to the list. The list can total 1024 assignments, numbers that can be used are 0 -1023.

Note: Existing assignments can be changed; however it is recommended that you do not change them.

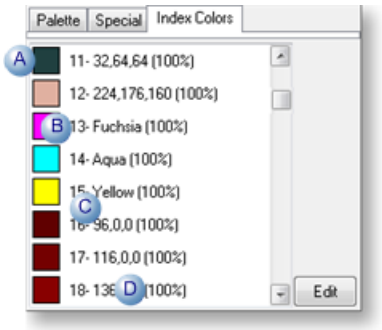
3.1. Select Colors on the Index Colors Tab

- Index colors tab overview.
- Select an index color.

Index Colors Tab Overview

When you first select the Index Colors tab in the Color palette, the tab lists the colors that CIMPLICITY has assigned to Index numbers.

The information that displays for its corresponding color is as follows.



A	A swatch that displays the color.
B	Index number.
C	Description of the color. The description can be either: • A text description. • RGB designations that together create the color.
D	The color's percent transparency.

Select an Index Color

Click the color you want to use.

Result: The palette instantly closes: the selected color displays in the Color field.

3.2. Open the RGB Editor

Open the RGB Editor

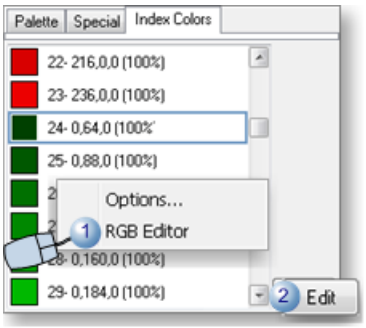
Do one of the following to open the RGB Editor

On the CimEdit Ribbon Bar

If you have selected an object on the CimEdit screen,

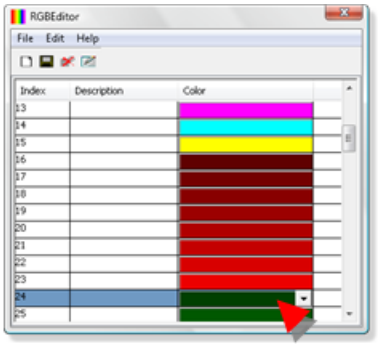
Click Edit in the Format>Index Colors group on the CimEdit Ribbon bar.

In the CimEdit Palette



1	Right-click any color on the Index Colors tab; select RGB Editor on the Popup menu.
2	Click the Edit button.

Result: The RGB Editor opens The color that was selected in the palette is selected in the RGB Editor.



3.3. Index Custom Colors and Entries

3.3. Index Custom Colors and Entries

index colors:

- Can be used in the screen anywhere a color can be configured.
- Include the:
- Entire RGB color palette.
- Alpha channel (transparency) of the colors

Note: 255 is opaque, 0 is completely transparent

The RGB Editor enables you to:

3.3.1 *(page 181)*	Add additional colors to the RGB Editor.
3.3.2 *(page 183)*	Modify an existing color index.
3.3.3 *(page 186)*	Sort, delete RGB indexes.

Tip: Right-click the scroll bar to open a Scroll Popup menu and quickly navigate through the RGB Editor.

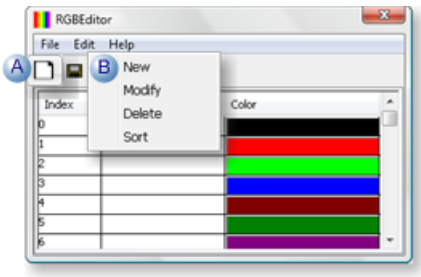
3.3.1. Add Additional Colors to the RGB Editor

The default RGB Editor displays colors for Index numbers 0-95: this range cannot be deleted.

You can add additional colors, including Index numbers for transparent colors.

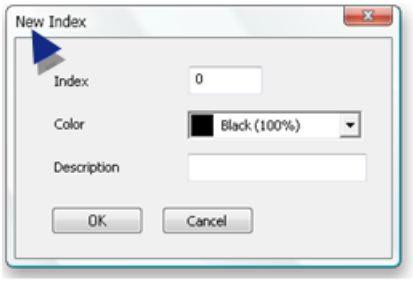
The highest Index number you can use is 1023.

1. Do one of the following.

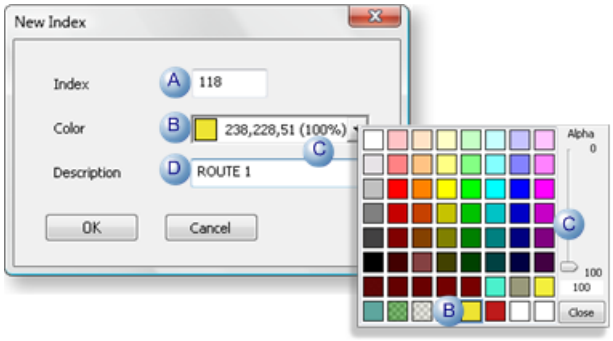


A	Click the New button on the RGB Editor toolbar.
B	Click Edit>New on the RGB Editor menu bar.

A New Index dialog box opens.



2. Enter and select the following.



A	Index	Index number that the color will be assigned to.
B	Color	Color assigned to the index number. a. Click the Down arrow to the right of the Color field to open a color palette. b. Specify a color the same way you do using the palette on the RGB tab. Note: Colors (100% transparency) in the standard palette have assigned index numbers.
C	Transparency	Standard or custom colors that have less than 100% transparency may be added to the indexed list.
D	Description	Enter a description that will display in the: • Description field in the RGB Editor • As the color description on the Index Colors tab If the description is left blank the RGB values and percent opacity in the **Color** field display as the color description on the Index Colors tab. The values do not display in the **Description** field in the RGB Editor.

3. Click one of the following.

OK	• Closes the New Index dialog box. • Adds the new index to the end of the list in the RGB Editor.
Cancel	• Closes the New Index dialog box. • Does not add the new index to the list.

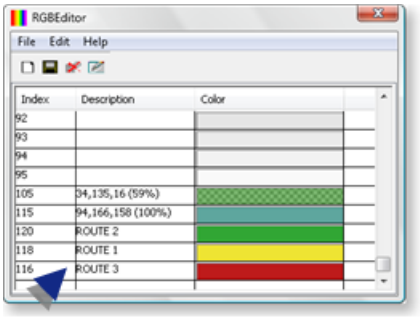
4. Do one of the following.

   Click the Save button.

   Click File>Save on the RGB Editor menu bar.

5. Continue to add the custom colors you need for your system's requirements.

The added colors are listed at the bottom of the RGB Editor.



3.3.2. Modify an Existing Color Index

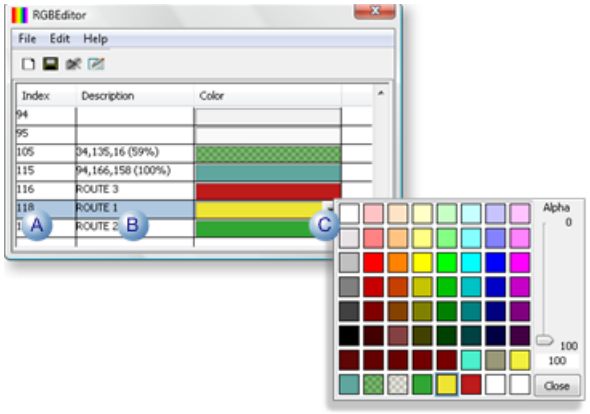
You can modify any aspect of a selected index.

- Edit a color index directly in the RGB Editor.
- Use a Modify Edit dialog box.
- Apply different file encodings.
- Modify indexes guidelines.

Edit a color index directly in the RGB Editor

1. Select an index in the RGB Editor list.

2. Do any of the following.



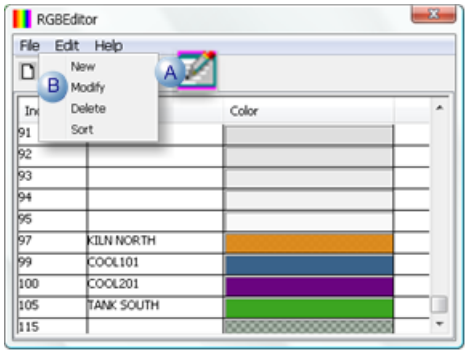
A	Change the Index number.
B	Change or remove the description.
C	Click the Down arrow to the right of the **Color** field; select a new color in the palette. **Note:** You can open the Advanced palette to create a new color.

Result: The changes will be written to the RGB.dat file when you save them.

Use a Modify Edit dialog box

You can change any of the three entries for an index color, as follows.

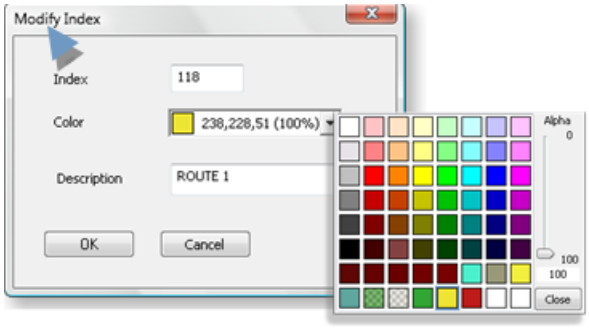
3. Select an index color in the RGB Editor list.
4. Do one of the following.



A	Click the Modify button on the RGB Editor toolbar.
B	Click Edit>Modify on the RGB Editor menu bar.

A Modify Index dialog box opens.

An advantage of opening the Modify Index dialog box is that you can see the exact RGB numbers in the **Color** field.



5. Change entries in any of the fields.
6. Click OK.

Result: The changes display in the RGB Editor. When you save the file, the changes will be applied to the RGB.dat file.

Apply different file encodings

The RGB Editor supports right to left reading and the following Unicode control characters.

7. Right-click any field in the **Index** or **Description** column.
8. Check any of the following options.

   Options on the Popup menu are.

Field	Description
Right to left Reading order	
Display one of the following.	
	Show Unicode control characters
	Hide Unicode control characters
Insert Unicode control character	
	The extended menu has the following options.
	LRM
	RLM
	ZWJ
	ZWNJ
	LRE
	RLE
	LRO
	RLO
	PDF
	NADS
	NODS
	ASS
	ISS
	AAFS
	IAFS
	RS
	US

guide: Modify Indexes Guidelines

- If you change the index number directly or in the Modify Index dialog box, the Index remains in the same location in the list. It may now be out of order. You can resort the list to put the new number in the correct order.
- When you select the Index Colors tab after you save your changes, if:
- No description has been entered for a color in the RGB Editor, the RGB value and percent opacity display next to the color;
- A description has been entered, that description displays next to the color.

3.3.3. Sort, Delete RGB Indexes

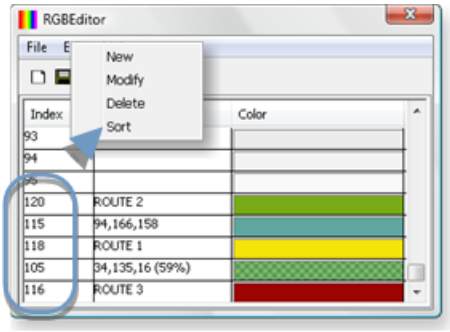
- Sort indexes
- Delete indexes

Sort Indexes

When you add additional indexes, new index numbers may not be in ascending order.

Do one of the following.

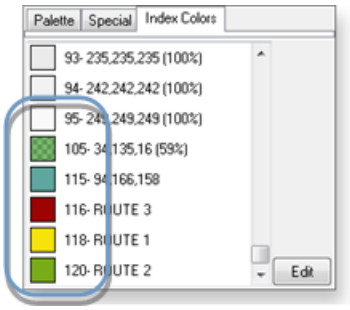
- Click Edit>Sort on the RGB Editor menu bar.



Result: The indexes are sorted in ascending order.

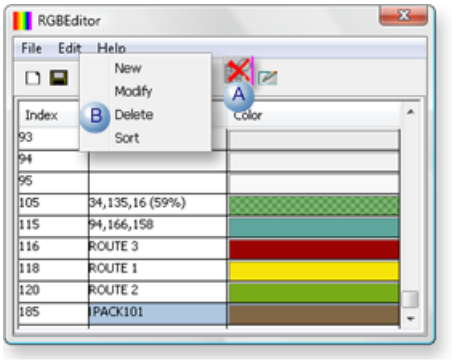
- Save and close the RGB Editor.

Result: The indexes display in ascending order on the palette's Index Colors tab. They will also be sorted when you re-open the RGB Editor.



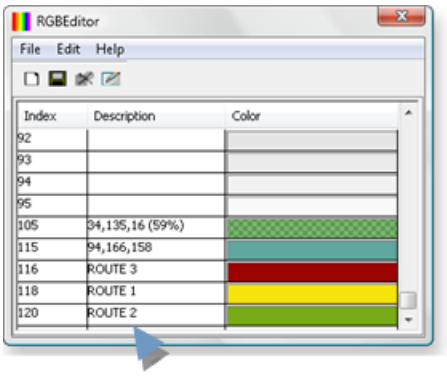
Delete Indexes

1. Select the color index you want to delete.
2. Do either of the following.



A	Click the Delete button.
B	Click Edit>Delete on the RGB Editor menu bar.

The selected color is deleted from the list.



RGB Technical Reference

- Overview
- RGB.dat file

- Default color mappings for alarm class colors.
- Alarm State and Color Index Animation and RGB.dat Files
- Import Screens from a CIMPLICITY System… and RGB.dat Files

Overview

- The first time CimView or CimEdit needs to access the RGB.dat file it will search directories in the following order.

1. <CimView or CimEdit current directory>\RGB.dat
2. SITE_ROOT:data\RGB.dat

   SITE_ROOT is defined to be the project directory. This is usually not defined on a Viewer. Its only defined if you launch CimView from the Workbench or a command prompt prepared by the Workbench.

3. BSM_ROOT:data\RGB.dat

   BSM_ROOT is the installation directory of the product.

   The RGB.dat Editor launched from CimEdit will always perform the above search when it is launched; if deployment is configured CimView will watch if that file is being deployed.

   If the RGB.dat Editor finds a different file from the last time CimEdit was launched, then:

   - CimEdit will reload its colors with this new file when the RGB Editor closes.
   - You cannot cause the RGB.dat Editor to save to a different location than the file it loaded
   - If the RGB.dat file does not exist in the location it is expected to be in, an RGB.dat file will display with a default color palette and be saved to BSM_ROOT:data\RGB.dat..
   - Each RGB color is:
   - Made up of a balance of

   ◆ Red, green, and blue luminosity values ranging from 0 to 255.

   ◆ Alpha value (transparency) ranging from 0 to 255.

   An Alpha value is required to include a description and/or a lock specification.

   - Identified by an index number.
   - (Optional) Identified in a description.
   - (Optional) Locked so it cannot be modified.

   The lock feature is available when the RGB.dat file is edited in a text editor (e.g. Notepad).

   Lock values are:

Locked	1

Not locked	0

- The syntax of the line format in the RGB.dat file is:

<Index><tab><RedValue><tab><GreenValue><tab><BlueValue><tab><AlphaValue><tab><description><tab

Example

Index No.	Red	Green	Blue	Alpha
101	0	65	0	169

RBG.dat file

The RBG Editor lists the CIMPLICITY default entries in an RGB.dat file; when you change default entries or add entries in the RGB Editor.

- You may need to edit the RGB Editor to:
- Change the default indexes for

- Alarm State animation

- Color Index animation

- Add additional indices.
- You will need the RGB.dat file to:
- Import screens from a CIMPLICITY System-I/U, -D/V, -H/U, or -RS/U system that do not use the default color mappings.
- If you change the file on a:

- CIMPLICITY Server, you will need to store a copy of the RGB.dat file in each project's **\data** directory.

- CIMPLICITY Viewer, you will need to store it in the **\data** directory under the main directory where your CIMPLICITY software is installed.

- The RGB.dat file is stored with a UTF-8 (Unicode) encoding so you can use descriptions with any language without dependency on the current code page of the operating system.

Default Color Mappings for Alarm Class Colors

The default color mappings used by the Alarm Viewer and Alarm Class Configuration to assign foreground and background colors to alarms in an RGB.dat file are:

Number	Red	Green	Blue	Total Color
0	0	0	0	Black
1	255	0	0	Red
2	0	255	0	Lime
3	0	0	255	Blue

Number	Red	Green	Blue	Total Color
4	128	0	0	Maroon
5	0	128	0	Green
6	128	0	128	Purple
7	255	255	255	White
8	0	128	128	Teal
9	128	128	128	Gray
10	128	128	0	Olive Green
11	32	64	64	(Dark)
12	224	176	160	(Rose)
13	255	0	255	Fuchsia
14	0	255	255	Aqua
15	255	255	0	Yellow

Alarm State and Color Index Animation and RGB.dat Files

The default colors used by Alarm State and Color Index animation are:

Index Numbers	Description
0–15	Default colors used (also used by the Alarm Viewer and Alarm Strings).
16–63	Rainbow colors.
64–95	Gray tones.
96-255	Available for other purposes

Note:

- For Alarm State and Color Index animation, if an index value does not have a defined color and an object's color has been assigned that index value, the color displays as black.
- You can open an RGB Editor through this tab and add additional index numbers and assignments to the list. The list can total 1024 assignments, numbers that can be used are 0 -1023.

Import Screens from a CIMPLICITY System… and RGB.dat Files

Imported Screens from a CIMPLICITY System…

When you import screens from a CIMPLICITY System-I/U, -D/V, -H/U, or -RS/U system that do not use the default color mappings you will need to create an RGB.dat file.

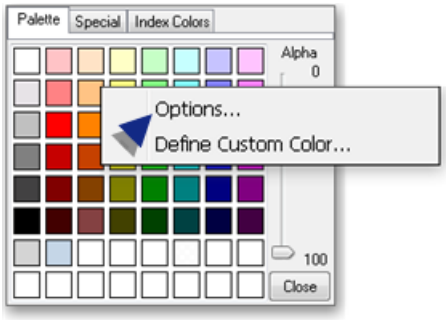
If you are importing screen files from a CIMPLICITY System-D/V, -H/U, -I/U, or -RS/U system, the color mappings should match the RGB.dat file on that system.

If the system you are importing from does not use the default color mappings, you will have to create an RGB.DAT file that matches the one on the source system to display its screens correctly.

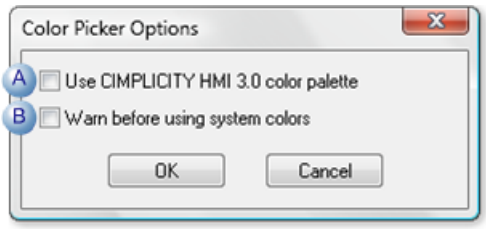
You should use the same color mappings for all projects.

4. Fixed Color Palette Options

1. Right-click a color in the Palette tab, the following popup menu displays:
2. Select Options…



The Color Picker Options dialog box opens.



A	Use CIMPLICITY 3.0 color palette Options are:
	- Clear
	- Check
B	Warn before using system colors. Options are:
	- Clear
	- Check

Borders and Lines

Borders and Lines

CimEdit provides several tools to format borders and lines.

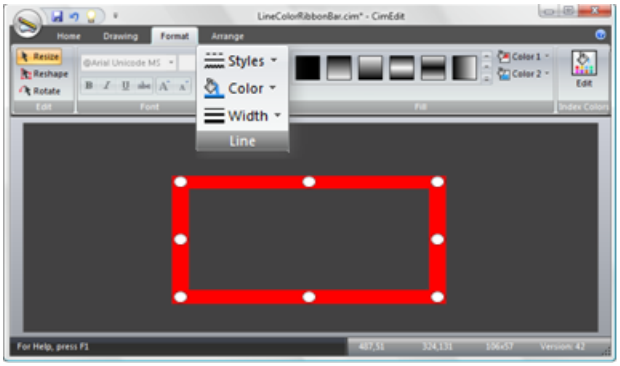
- Borders and lines: On screen formatting
- Borders and lines: Detailed formatting

Borders and Lines: On Screen Formatting

On Screen

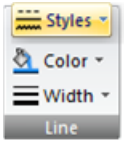
Select the object for line configuration.

On screen formatting options are as follows.



Styles

Click Styles in the Format>Line group on the CimEdit Ribbon bar.

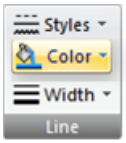


A Pattern dropdown menu provides the following options.

1	Patterns	As you move the cursor over the line patterns, the line selected on the CimEdit screen changes. Click a style to select it.
2	More lines	Opens the object's Properties dialog box. **Colors** is selected.

Color

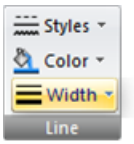
Click Color in the Format>Line group on the CimEdit Ribbon bar.



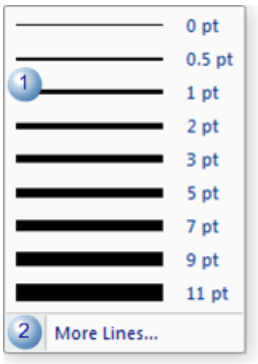
A CimEdit palette opens.

Width

Click Width in the Format>Line group on the CimEdit Ribbon bar.



A Width dropdown menu provides the following options.

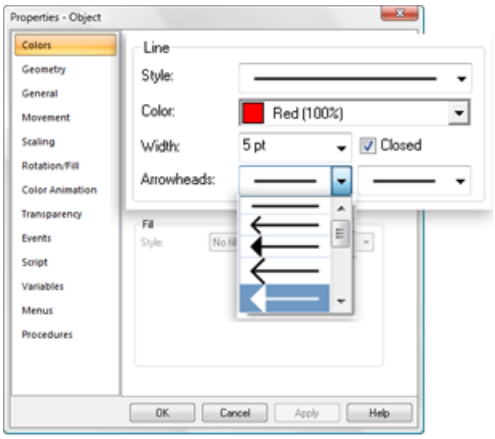


1	Width	As you move the cursor over the line widths, the line selected on the CimEdit screen changes. Click a width to select it.
2	More lines	Opens the object's Properties dialog box. **Colors** is selected

Borders and Lines: Detailed Formatting

Many objects you place on your screens, including text, can be outlined with a border. In addition, you can change the style and shape of any line.

Options for selecting the line display are as follows.



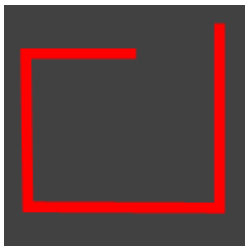
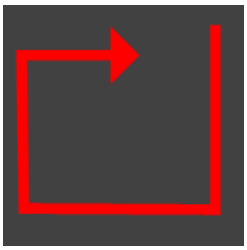
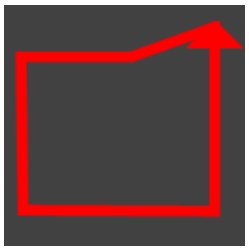
Option	Description
Style	Any of several dotted, dashed and a solid line.
Color	Line color *(page 168)*.
Width	Line width, which can be entered or selected from the drop-down list.
Closed	(Disabled for object borders, e.g. ellipses and rectangles)
	Checked
	Clear

Option	Description
Arrowheads	Left
	Right

Result: The line displays on the screen according to your selections.

Example

An open polygon is drawn on the CimEdit screen. If it is closed and an arrow is added at the start or end the shape will be modified per specifications. Once closed, the object can be filled.

Open Line	Open Line Arrow Start	Closed Line Arrow End
		

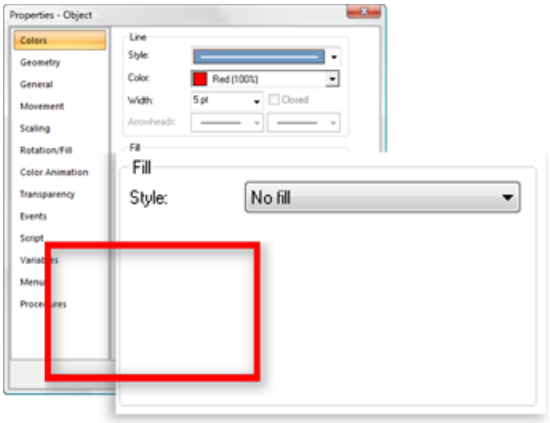
No Fill for an Object

Select Colors in the Properties dialog box.

Because there is nothing to configure when you select **No fill** for the fill **Style**, the Fill section only displays your selection.

You can see any objects behind the unfilled object. Only the line you select as the border (in the Line section above the Fill section) defines the object.

Note: When you select No fill, you must define the line.



Solid Fill: Options

CimEdit provides several tools to fill objects with a solid color.

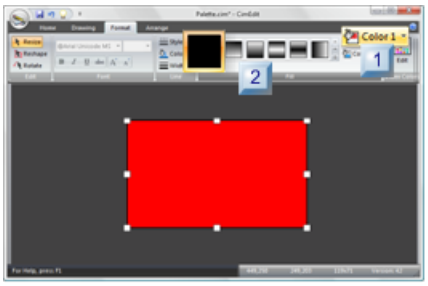
- On screen formatting
- Detailed formatting

On Screen Formatting

An object can be quickly filled with a solid color without opening its Properties dialog box.

Select the object to be filled.

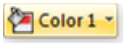
The Ribbon bar provides the following tools.



rect 219, 25, 244, 46 *(page 196)*
rect 145, 34, 170, 55 *(page 197)*

1 *(page 196)*	Color 1
2 *(page 197)*	Fill type

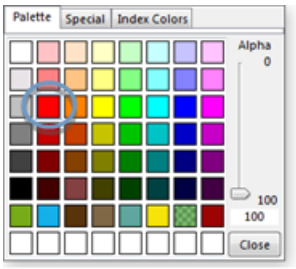
1	Color 1

1. Click Color 1  in the Format>Fill group on the CimEdit Ribbon bar.

A CimEdit Color palette opens.

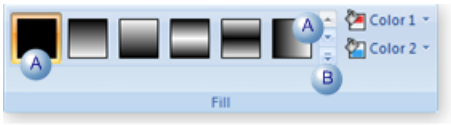
Note: As the cursor moves over colors, the object fill changes color accordingly.

1. Click the color that will fill the solid filled object.
2. (Optional) Select the color's transparency percent.



2	Fill type

Use either of the following to select a solid fill.



A	Solid button

1. (If a solid button does not display on the Fill Button bar) scroll up and down to find it.
2. Click the solid button.

B	Dropdown menu

1. Click the Dropdown menu button in the Fill group.

A Dropdown menu opens.

1. Do either of the following.

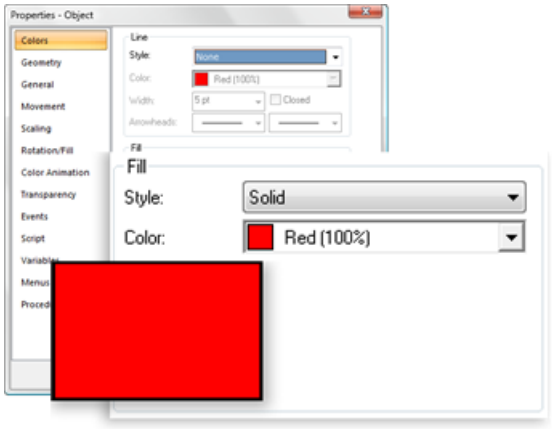
A	Click the solid pattern box on the dropdown menu.
B	Click **More Info**. The object's Properties dialog box will open displaying your color choice.

Solid Color Detailed Formatting

Select Colors in the Properties dialog box.

When you select **Solid** for the fill **Style**, the Fill section changes so you can select the color of the fill.

1. Click the down arrow in the **Color** field to display the Color Palette.
2. Select the color you want.



Pattern Fill: Options

Pattern Fill: Options

CimEdit provides several tools to fill objects with a pattern.

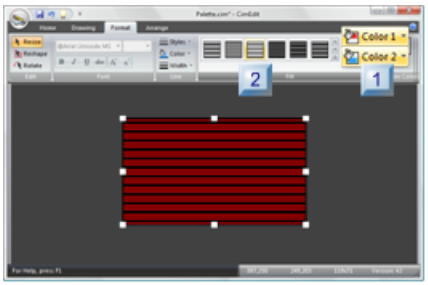
- On screen formatting
- Two opaque colors.
- One transparent and one opaque color.

Pattern Fill: On Screen Formatting

An object can be quickly filled with a pattern without opening its Properties dialog box.

Select the object to be filled.

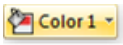
The Ribbon bar provides the following tools.



rect 144, 34, 170, 61 *(page 200)*
rect 220, 33, 246, 60 *(page 199)*

1 *(page 199)*	Color 1
2 *(page 200)*	Fill type

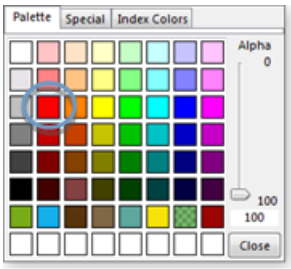
1	Color 1, Color 2

1. Click Color 1  in the Format>Fill group on the CimEdit Ribbon bar.

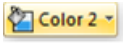
A CimEdit Color palette opens.

Note: As the cursor moves over colors, the object fill changes color accordingly.

1. Click the color that will fill the object's pattern background.
2. (Optional) Select the color's transparency percent.



The palette closes; the selected color displays as the pattern background.

1. Click Color 2 

A second CimEdit Color palette opens.

1. Click the color that will fill the object's pattern.

2. (Optional) Select the color's transparency percent.

The palette closes; the selected color displays as the pattern color.

2	Fill Type

Use either of the following to select a pattern fill.

A	Pattern button

1. (If the pattern you want does not display on the Fill Button bar) scroll up and down to find it.
2. Click the pattern button when you find the pattern you are looking for.

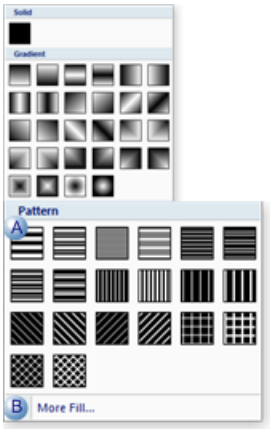
Note: As you move the cursor over pattern buttons, the pattern fill in the object changes accordingly.

B	Dropdown menu

1. Click the Dropdown menu button in the Fill group.

A Dropdown menu opens.

1. Do either of the following.

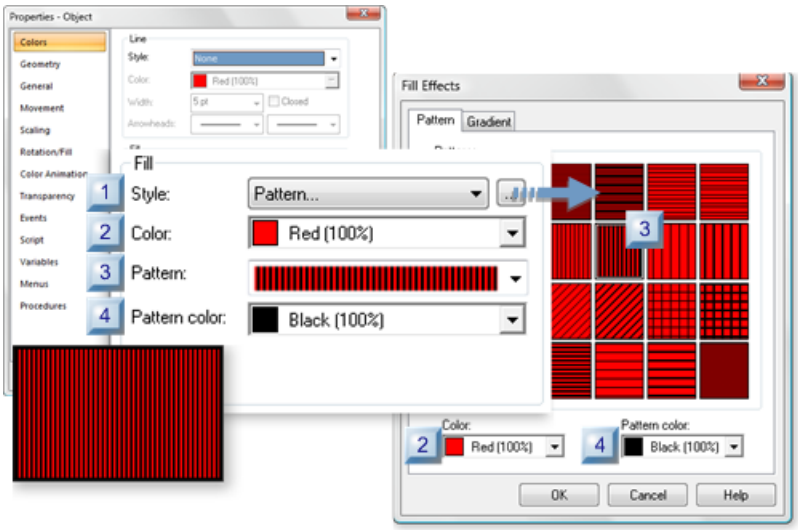


A	Click the pattern box on the dropdown menu that represents the pattern you want.
B	Click **More Info**. The object's Properties dialog box will open displaying your color and pattern choices.

Pattern Fill: Two Opaque Colors

An object's fill is configured on the Colors tab of its Properties dialog box.

Options for a two-color pattern fill are as follows.



rect 51, 106, 77, 127 *(page 201)*
rect 385, 127, 419, 156 *(page 202)*
rect 248, 261, 281, 287 *(page 202)*
rect 50, 126, 78, 152 *(page 202)*
rect 50, 151, 76, 180 *(page 202)*
rect 48, 179, 77, 205 *(page 202)*
rect 360, 261, 389, 287 *(page 202)*

1 *(page 201)*	Style
2 *(page 202)*	Color
3 *(page 202)*	Pattern
4 *(page 202)*	Pattern Color

1	Style

Select Pattern...

1. (Optional) Click the Browse button ... to the right of the Style drop-down list.

The Fill Effects dialog box opens.

1. Select the Pattern tab.

2	Color

The first color selected is the background color for the pattern.

The Down arrow to the right of the Color field opens the CimEdit palette.

3	Pattern

Available pattern options using the selected colors.

Do one of the following.

- Select the pattern from the drop-down list in the Properties dialog box.
- Click a pattern in the Fill Effects dialog box.

4	Pattern Color

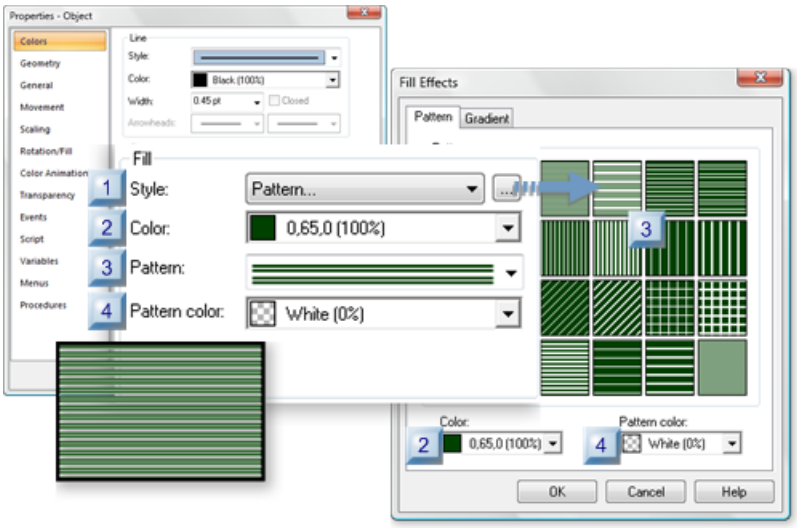
Second color that creates the pattern when applied to the background color.

The Down arrow to the right of the Color field opens the CimEdit palette.

Pattern Fill: One Opaque and one Transparent Color

An object's fill is configured on the Colors tab of its Properties dialog box.

If you want an object to show behind another object, but you want the front object to have more than substance than just a border, you may want a pattern with a transparent affect.



rect 51, 176, 81, 204 *(page 204)*
rect 359, 260, 389, 288 *(page 204)*
rect 49, 149, 78, 177 *(page 204)*
rect 384, 126, 413, 154 *(page 204)*
rect 49, 123, 79, 151 *(page 204)*
rect 249, 262, 279, 290 *(page 204)*
rect 49, 98, 80, 126 *(page 203)*

1 *(page 203)*	Style
2 *(page 204)*	Color
3 *(page 204)*	Pattern
4 *(page 204)*	Pattern Color

1	Style

Select Pattern...

1. (Optional) Click the Browse button ... to the right of the Style drop-down list.

The Fill Effects dialog box opens.

1. Select the Pattern tab.

2	Color

The first color selected is the background color for the pattern.

The Down arrow to the right of the Color field opens the CimEdit palette.

When the color is set to 0%, the background is completely transparent.

Note: Any transparency percent can be applied.

3	Pattern

Available pattern options using the selections in the Color fields.

Do one of the following.

- Select the pattern from the drop-down list in the Properties dialog box.
- Click a pattern in the Fill Effects dialog box.

4	Pattern Color

Second color that creates the pattern when applied to the background color.

The Down arrow to the right of the Color field opens the CimEdit palette.

When the color is set to 0%, the pattern color is completely transparent.

Note: Any transparency percent can be applied.

Gradient Fill: Options

Gradient Fill: Options

CimEdit provides several tools to fill objects with a gradient.

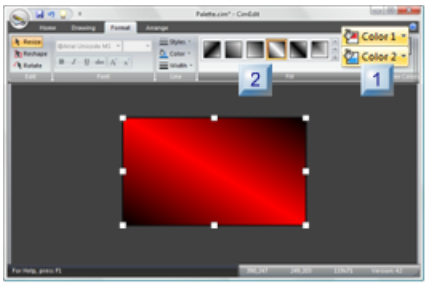
- On screen formatting.
- Two colors.
- One color.

Gradient Fill: On Screen Formatting

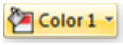
An object can be quickly filled with a gradient without opening its Properties dialog box.

Select the object to be filled.

The Ribbon bar provides the following tools.



1	Color 1, Color 2

1. Click Color 1 

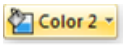
in the Format>Fill group on the CimEdit Ribbon bar.

A CimEdit Color palette opens.

Note: As the cursor moves over colors, the object fill changes color accordingly.

1. Click the color that will fill the object's gradient background.
2. (Optional) Select the color's transparency percent.

The palette closes; the selected color displays as the gradient background.

1. Click Color 2 

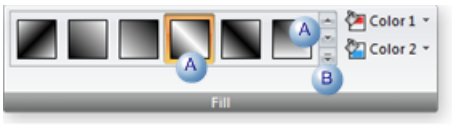
A second CimEdit Color palette opens.

1. Click the color that will fill the object's gradient foreground.
2. (Optional) Select the color's transparency percent.

The palette closes; the selected color displays as the gradient foreground.

2	Fill Type

Use either of the following to select a pattern fill.



A	Gradient button

1. (If the gradient you want does not display on the Fill Button bar) scroll up and down to find it.
2. Click the gradient button when you find the style you are looking for.

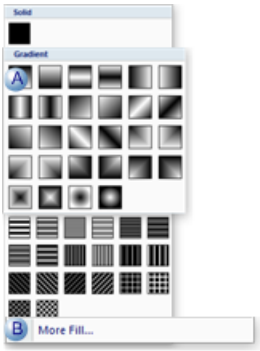
Note: As you move the cursor over gradient buttons, the gradient fill in the object changes accordingly.

B	Dropdown menu

1. Click the Dropdown menu button in the Fill group.

A Dropdown menu opens.

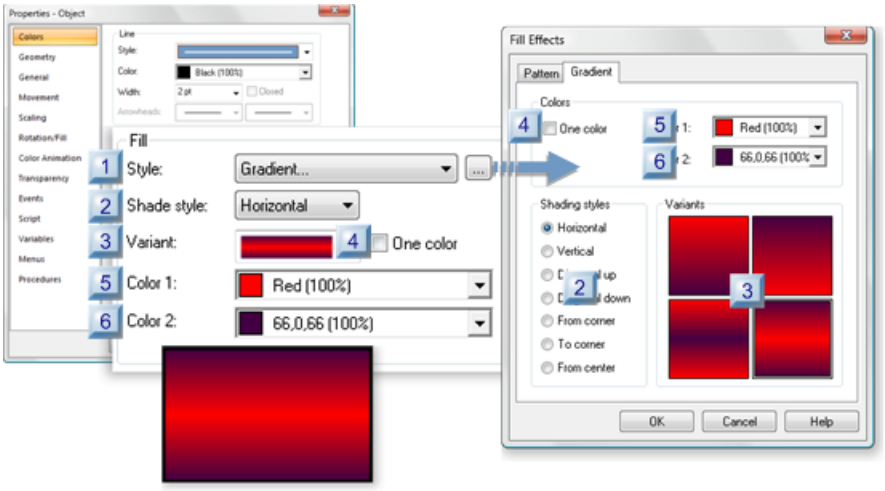
1. Do either of the following.



A	Click the pattern box on the dropdown menu that represents the gradient you want.
B	Click **More Info**. The object's Properties dialog box will open displaying your color and pattern choices.

Gradient Fill: Two Colors

An object's fill is configured on the Colors tab of its Properties dialog box.



rect 52, 89, 77, 113 *(page 207)*
rect 51, 114, 78, 138 *(page 208)*
rect 346, 165, 373, 189 *(page 208)*
rect 51, 135, 78, 159 *(page 208)*
rect 450, 166, 477, 190 *(page 208)*
rect 52, 161, 79, 185 *(page 209)*
rect 398, 64, 425, 88 *(page 209)*
rect 51, 186, 77, 210 *(page 209)*
rect 396, 86, 422, 110 *(page 209)*
rect 206, 136, 228, 160 *(page 209)*
rect 315, 63, 337, 87 *(page 209)*

1 *(page 207)*	Style
2 *(page 208)*	Shade Style
3 *(page 208)*	Variant
4 *(page 209)*	One color
5 *(page 209)*	Color 1
6 *(page 209)*	Color 2

1	Style

Select Gradient...

1. (Optional) Click the Browse button to the right of the Style drop-down list.

   The Fill Effects dialog box opens.

2. Select the Gradient tab.

2	Shade Style

   CimEdit provides several gradient styles for filling the selected object.

   Style types include:

   - Horizontal
   - Vertical
   - Diagonal up
   - Diagonal down
   - From corner
   - To corner
   - From center.

   Do one of the following.

   - Select the style from the drop-down list in the Properties dialog box.
   - Check a gradient style in the Fill Effects dialog box.

3	Variant

   Variations of the selected shade style.

   Do one of the following.

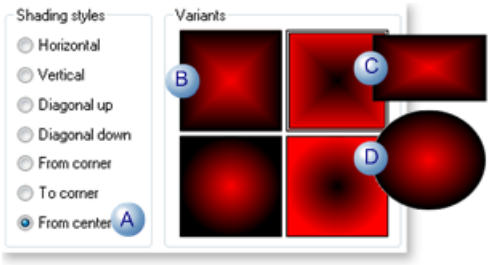
   - Select the style from the drop-down list in the Properties dialog box.
   - Click a gradient style in the Fill Effects dialog box.

Note: The variants in the two top two boxes indicate that the fill should conform to the shape of the object, e.g. if the shape is a polygon the fill will follow the lines in the polygon; if it is an ellipse the shape will follow the ellipse.

As a result, when an ellipse is selected, for some shading styles, the top and bottom variant fills will actually look the same. For other shapes, e.g. a rectangle) they will look exactly like the images indicated in the **Variants** boxes.

Example

A Red/Black gradient has been applied to a rectangle and circle, as follows.



A	From center is selected as the shading style.
B	The top left variant is selected.
C	The rectangle object fill looks like the selection.
D	The ellipse object fill conforms to the shape; as a result it resembles the bottom left box.

4	One color

Clear One color to configure a two-color gradient.

5	Color 1

One color in the gradient. Where it is positioned depends on the selected gradient.

The Down arrow to the right of the Color field opens the CimEdit palette.

6	Color 2r

One color in the gradient. Where it is positioned depends on the selected gradient.

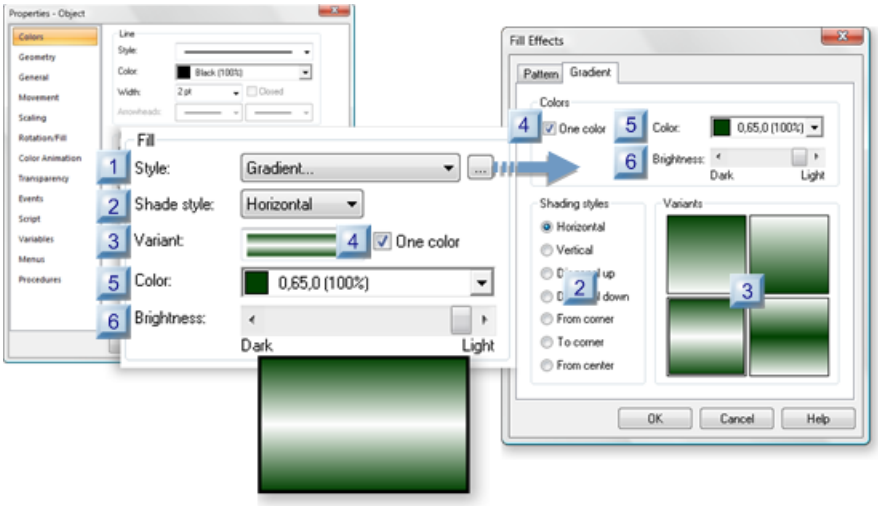
The Down arrow to the right of the Color field opens the CimEdit palette.

Note: If you decide you want to use one color, check the **One color** box.

Gradient Fill: One Color

An object's fill is configured on the Colors tab of its Properties dialog box.

If you want a fill to be one color that becomes lighter or darker in a specified manner, you can easily.



rect 57, 88, 83, 113 *(page 210)*
rect 313, 62, 340, 88 *(page 211)*
rect 205, 136, 232, 162 *(page 211)*
rect 451, 166, 478, 189 *(page 211)*
rect 54, 137, 81, 160 *(page 211)*
rect 55, 113, 81, 136 *(page 211)*
rect 348, 165, 374, 188 *(page 211)*
rect 57, 160, 84, 186 *(page 212)*
rect 380, 61, 407, 87 *(page 212)*
rect 54, 185, 81, 211 *(page 212)*
rect 378, 85, 405, 111 *(page 212)*

1 *(page 210)*	Style
2 *(page 211)*	Shade Style
3 *(page 211)*	Variant
4 *(page 212)*	One color
5 *(page 212)*	Color
6 *(page 212)*	Brightness

1	Style

Select Gradient...

1. (Optional) Click the Browse button to the right of the Style drop-down list.

   The Fill Effects dialog box opens.

2. Select the Gradient tab.

2	Shade Style

   CimEdit provides several gradient styles for filling the selected object.

   Style types include:

   - Horizontal
   - Vertical
   - Diagonal up
   - Diagonal down
   - From corner
   - To corner
   - From center.

   Do one of the following.

   - Select the style from the drop-down list in the Properties dialog box.
   - Check a gradient style in the Fill Effects dialog box.

3	Variant

   Variations of the selected shade style.

   Do one of the following.

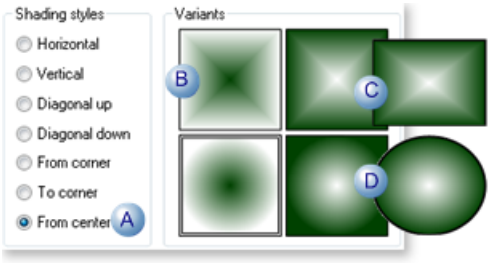
   - Select the style from the drop-down list in the Properties dialog box.
   - Click a gradient style in the Fill Effects dialog box.

Note: The variants in the two top two boxes indicate that the fill should conform to the shape of the object, e.g. if the shape is a polygon the fill will follow the lines in the polygon; if it is an ellipse the shape will follow the ellipse.

As a result, when an ellipse is selected, for some shading styles, the top and bottom variant fills will actually look the same. For other shapes, e.g. a rectangle) they will look exactly like the images indicated in the **Variants** boxes.

Example

A dark green one color gradient has been applied to a rectangle and circle, as follows.



A	From center is selected as the shading style.
B	The top left variant is selected.
C	The rectangle object fill looks like the selection.
D	The ellipse object fill conforms to the shape; as a result it resembles the bottom left box.

4	One color

Check one color to configure a gradient that had one selected color shaded from light to dark based on its configuration.

5	Color

The one color in the gradient. Where it is positioned static to light/dark shading depends on the selected gradient.

The Down arrow to the right of the **Color** field opens the CimEdit palette.

6	Brightness

Slide the Brightness tab to the spot that defines the desired intensity of the fill.

- The left most position corresponds to the minimum possible intensity of zero (black).
- The right most position corresponds to the maximum possible intensity of 255 (white).

Object Shadow

Shadows can help give objects a 3-D effect and make them stand out on a CimView screen.

CimEdit options provide you with the flexibility to create normal or diffuse shadows at selected distances and angles in custom selected transparent or opaque colors.

1 *(page 213)*	Display shadow properties.

2 *(page 213)*	Select shadow properties.

Display Shadow Properties

1. Select the object that will have the shadow.

1. Do one of the following.

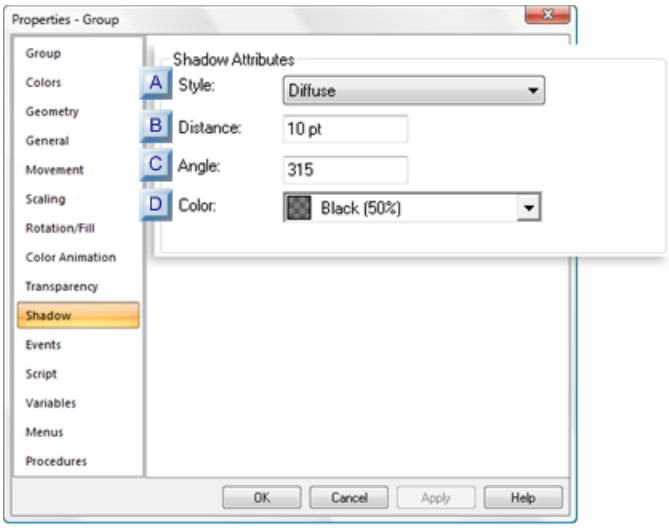
- Double-click the object
- Right-click the object; select Properties on the Popup menu.
- Click the Properties button in the Home>Properties group.

Result: The object's Properties dialog box opens.

Select Shadow Properties

Select Shadows in the Properties dialog box.

Shadows specifications are as follows.



rect 82, 114, 109, 135 *(page 216)*
rect 81, 87, 108, 112 *(page 216)*
rect 82, 62, 109, 87 *(page 215)*
rect 81, 36, 108, 61 *(page 214)*

A *(page 214)*	Style
B *(page 215)*	Distance
C *(page 216)*	Angle
D *(page 216)*	Color

A	Style

The shadow style provides a solid or more blurred effect.

Note: The exact display is affected by the selected color and transparency.

Shadow style options are:

Option	The shadow attribute:

None	Is not applied to the object. **Note:** If the object included a shadow when it was inserted, that shadow will continue to display.
Normal	Includes all of the pixels assigned to the selected color transparency, giving the shadow a more solid appearance.
Diffuse	Presents a gradient type, blurred view.

B	Distance

The entered distance causes the shadow to:

- Move closer to or away from the object.
- Be more or less prominent.
- Influence the 3-D effect.

Note: The distance is tied to the angle, which determines where the shadow will be placed relative to the object.

Example

The shadow display for an object with a diffuse 50% transparent black (at a 315 degree (-45 degree) angle) depends on its specified distance.

Distance	Shadow Display
5 Pt.	When the shadow distance is small, the shadow is almost hidden.

10 Pt.	As the shadow distance increases, the shadow becomes more noticeable.
15 Pt.	A higher shadow distance continues to make the shadow more noticeable.

C	Angle

The angle influences the object's perspective.

0°	45°	90°	135°
			180°
360°/0°	315°/-45°	270°/-90°	225°/135°

D	Color

The shadow can be any color at any transparency.

Example

Points, Variables and Expressions

Points, Variables and Expressions

You can create screens using points, variables, and expressions to monitor the status of your processes.

Points report specific conditions in the system and are configured in the Point Properties dialog. In CimEdit, you can:

- Find and use any point that is already in any broadcasting project on your network.
- Create new points by opening the Point Properties dialog box through CimEdit.

Variables are a container for a piece of data, such as a full or partial Point ID or a text string in an expression. Variables are completely contained in CimEdit, where the application can evaluate them and act on the result in the manner you specify.

Expressions are similar to formulas, and can include points, variables, and constants.

Points in CimEdit

Points in CimEdit

Basic guidelines about defining and configuring points to use in CimEdit are as follows.

- Points Configured through CimEdit.
- Point Attributes in CimEdit
- Point by Address in CimEdit
- PointView
- CimEdit Point Data in Point Cross Reference

Points Configured through CimEdit

Points Configured in CimEdit

1 *(page 218)*	Edit and create points through CimEdit.
2 *(page 220)*	Specify unqualified or fully qualified points for information source.

1. Create and Edit Points through CimEdit

You can create and edit points in CimEdit when the project is stopped or running.

Important:

- When a project is stopped, you must perform a Configuration Update to copy the configuration from the Master directory to the Data directory so they are both the same and the changes are applied to runtime.
- If the project is running, you must ensure dynamic point configuration is enabled for any point changes to immediately take effect.

Click Dynamic Points in the Runtime menu.

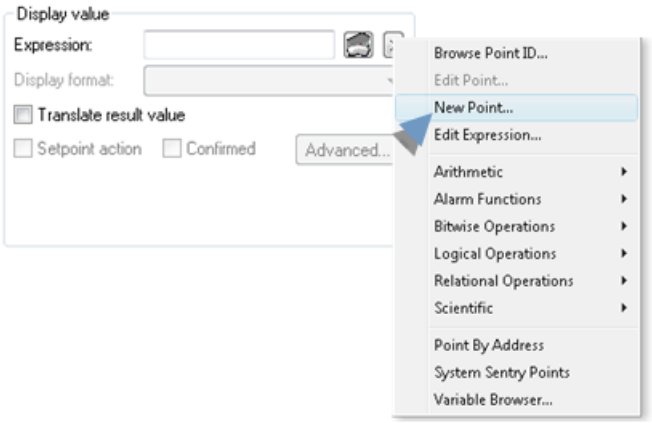
You can create or edit points through any field that allows points as entries.

- Create a new point.
- Edit a selected point.

Create a new point

1. Open the Properties dialog box for a selected object.
2. Click the Popup Menu button to the right of an **Expression** field.

   The Popup menu opens.
3. Select **New Point**.



The New Point dialog box opens.

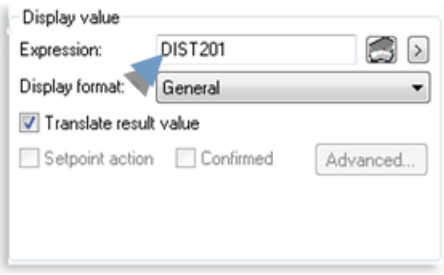
4. Create a new point.

   Important: Make sure to adhere to point naming rules if you create a new point through CimEdit.

5. Configure the new point.

6. Close the Point Properties dialog box for the new point.

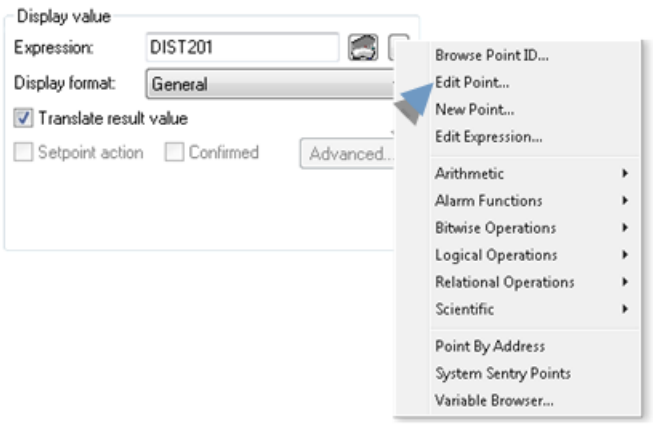
   Result: The new point displays in the Expression field.



   Edit a selected point.

7. Select a point in the **Expression** field.

8. Select Edit Point... on the Popup menu.



The Point Properties dialog box for the selected point opens.

9. Make any modifications to the point configuration.
10. Close the Point Properties dialog box.

The point modifications are applied:

- Immediately if the project is running and dynamic point configuration is enabled.
- After the next Configuration Update.

2. Specify Unqualified or Fully Qualified Points for Information Source

CimView can gather point information from any project on your network that is broadcasting.

You will need to identify where the information comes from through either of the following.

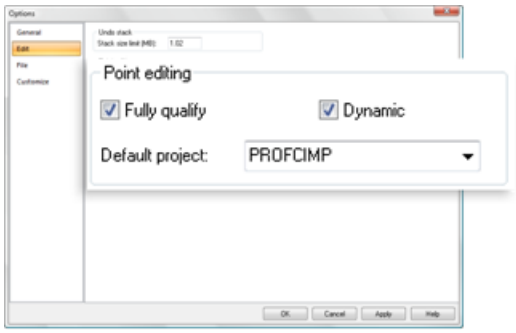
- Point editing specification in CimEdit
- Unqualified points.
- Qualified points.

Point editing specification in CimEdit

1. Click the CimEdit button .
2. Click Options button.

   The Options dialog box opens.

3. Select **Edit**.
4. Choose from the following options.



Checked Box/Field	Description
Fully qualify	Makes the points fully qualified *(page 222)*.
Dynamic Points	(Enabled when the project is running) enables dynamic point configuration.
Default project	From drop-down list, default project for which the points are fully qualified.

Unqualified Point

An unqualified point just specifies the Point ID.

Example

MYPOINT

Information about an unqualified point can come from the locally running project or from the default project determined by the `/project <project>` command line option used.

Unqualified points are best used when you:

- Have a single-node application.
- Are creating CimView screens in a project that will be copied.
- Want to create a screen to view points with the same IDs from different projects.

To get information about the point, CimEdit and CimView use the following steps to fully qualify an unqualified Point ID:

5. An `Open Screen` or `Overlay Screen` action can specify a project to be used to fully qualify any unqualified Point IDs. When a user opens a screen using one of these actions, the project specified in the action is used to fully qualify any unqualified Point IDs.
6. If a command line is used to open CimEdit or CimView, and the `/project` option is used, that project is used to fully qualify any unqualified Point IDs.
7. If the screen is in a project directory, that project is used to fully qualify any unqualified Point IDs.

8. If you are on a Server and a single project is currently running, that project is used to fully qualify any unqualified Point IDs.
9. If you are on a Server and multiple projects are running, the project that was started first is used to fully qualify any unqualified Point IDs. However, this behavior can be modified by using the PROJECT_ID global parameter.
10. If you are on a Viewer, unqualified Point IDs are unavailable.

    Fully Qualified Point

    A fully qualified point is prefaced with a project name, node name or IP address.

    Example:

    `\\MYPROJ\MYPOINT`

    `\\SERVER1\MYPOINT`

    `\\111.36.2.13\MYPOINT`

    Information about a fully qualified point always comes from the project specified in the Point ID. The project can be identified by its name or by its node or IP address.

    Fully qualified points are best used when you have multiple projects running on the network at the same time. When you create a CimView screen using fully qualified Point IDs and use that screen on any computer running CIMPLICITY software, the point information always comes from the projects you specified in the Point IDs.

Point Attributes in CimEdit

1. Select or create an object to which the point attribute will be applied.
2. Open the object's Properties dialog box.
3. Click the Edit Expression button to the right of the **Expression** field that applies to the object's runtime behavior.

   The Edit Expression dialog box opens.

4. Enter the Point ID and attribute you want to display using the following syntax.

   '< POINT_ID >.< ATTRIBUTE_ID >'

   Where

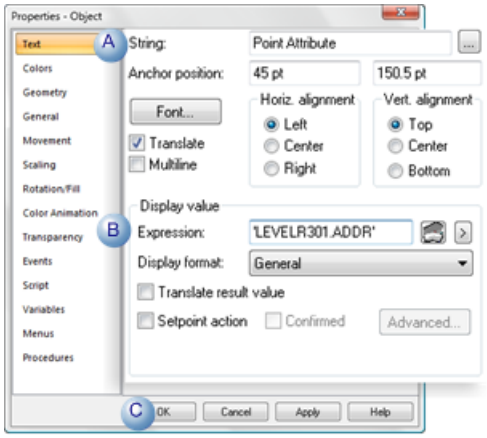
< POINT_ID > is the Point ID of any configured point

< ATTRIBUTE_ID > is one of the configuration attributes described in the point attributes documentation.

Important: You must enclose the syntax in single quotes, because the '.' character is a special character.

Example: Point Attribute Configuration

5. Open a Properties dialog box for a text object.
6. Select **Text**.
7. Enter the following.



A	Enter a default string in the **String** field.
B	Enter a '<POINT_ID>.<ATTRIBUTE_ID>' expression in the Expression field.
	Tip: Click the Expression Editor button to use the Expression Editor.
C	Click OK.

The string displays as follows.

In CimEdit	In CimView
Point Attribute	%R456

Point Attribute Guidelines

- You can force an expression to return a numeric value, if the value is a number but the attribute is listed as returning a character. You use the operator **VAL** to do this.

Example

VAL('DEV1750.ALARM_HIGH')+VAL(DEV1900.ALARM_HIGH')

- Some of the attributes are specific to points of a particular type. Valid point types, which are specified in the attribute list, are:
- All (point types)
- Device
- Derived (Virtual)
- Global (Virtual)
- All field names and enumerated data are case insensitive.

Point by Address in CimEdit

Point by Address in CimEdit

You can use points to reference data in CIMPLICITY software. CimEdit and CimView support both fully qualified and unqualified point addressing..

In many circumstances, you may want to view raw device information without the overhead of configuring a CIMPLICITY point. For example, you may want to view low-level diagnostic information, which is seldom used. You can do this through the use of Point by Address descriptions specified within a CimView screen file.

Point by Address descriptions provide a mechanism for storing the configuration for a CIMPLICITY point as an inline macro within a CimView document. When a user opens the document in CimView, the point is dynamically created in memory. When the user closes the document, the point is dynamically removed from memory.

- Point by address restrictions
- Point by address security
- Point by address configuration
- Point by address syntax

Point by address restrictions

Because Point by Address descriptions only exist while they are being viewed, they have the following restrictions:

- They may only be used in applications, like **CimView,** which directly display point data.
- They cannot be alarmed.
- They cannot be logged.
- They cannot be used in the configuration of another point.
- They cannot have engineering units conversion.
- In deciding whether to use a Point by Address description, also consider the following:

- Because they are created and destroyed dynamically, Point by Address descriptions do not take up memory while they are not in use. However, there is a small delay during startup for these points to be created.
- Because the configuration of a Point by Address is specified in the CimView document rather than the CIMPLICITY project, changes to Point by Address configuration will be more complex.

In general, a point that is frequently used or is referenced by multiple CimView screens should be configured as a CIMPLICITY point. For diagnostic information, which is seldom used and is included in only one screen, a Point by Address is appropriate.

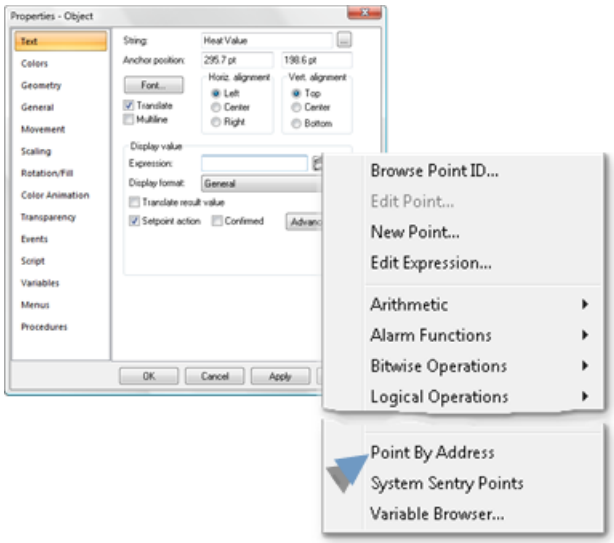
Point by address security

Because the Point by Address feature could potentially give users complete access to any portion of your device memory, this feature is protected by a special privilege. Users who are assigned roles that have the Point by Address privilege disabled will be unable to read or write Point by Address data.

You may also use the CIMPLICITY Setpoint Security feature to enforce read access on Point by Address data. For this purpose, the resource used will be the resource of the CIMPLICITY device from which the Point by Address is collected.

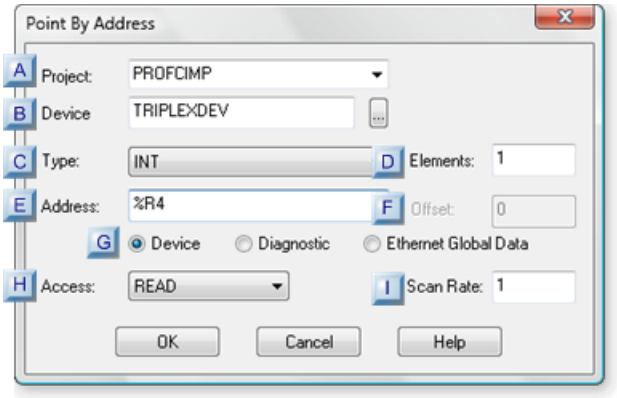
Point by Address Configuration

1. Display the Point ID popup menu in any CimEdit **Expression** fields.
2. Select Point by Address.



A Point by Address dialog box opens.

3. Configure the point by address as follows.



rect 226, 164, 255, 190 *(page 227)*
rect -2, 28, 24, 56 *(page 226)*
rect -5, 53, 21, 80 *(page 226)*
rect -2, 86, 27, 113 *(page 226)*
rect 227, 84, 251, 111 *(page 226)*
rect -4, 112, 24, 139 *(page 226)*
rect 227, 113, 255, 140 *(page 226)*
rect 52, 134, 75, 161 *(page 226)*
rect -2, 164, 26, 190 *(page 227)*

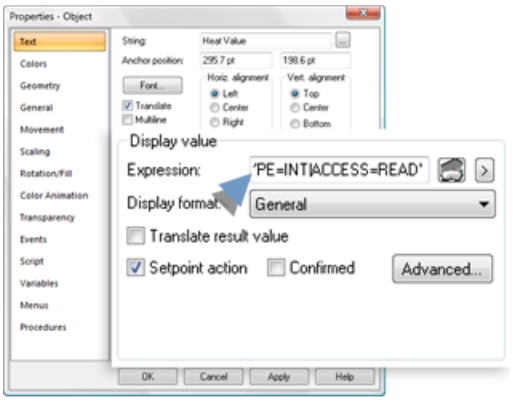
	Field	Description
A	Project	(Optional) Remote Project name for fully qualified projects.
B	Device	Device from which the data will be collected.
C	Type	Data type that will be collected.
		Options are:
		• BOOL • BYTE • DINT • DWORD • INT • REAL
D	Elements	Number of elements to be retrieved
		Default
E	Address	Valid address for the specified device.
F	Offset	Enter the bit offset for the address if the point is one of the following:
		• BOOLEAN • BYTE • WORD
G	Address button	Appropriate radio button to select the data associated with the address:

	Field	Description
		• Device • Diagnostic • Ethernet Global Data
		Default
H	Access	Select one of the following.
		WRITE
		READ
		Default
I	Scan Rate	Multiple of the device Scan rate at which the data will be collected.
		Default

4. Click OK.

CimEdit displays the point address in the Expression field.

Example Address



An example address in the Expression field is:

```
'\\CIMPDEMO\@DEVICE=TRIPLEX|ADDR=%R4|TYPE=INT|ACCESS=READ'
```

Note:

- If the Point by Address entry is dimmed on the popup menu, it is not supported.
- Open the Expression Editor to check the full expression.

Point by Address Syntax

After you specify a Point by Address description using the Point by Address dialog box, a completed description will be returned to the **Edit** field. This description is specified as a string of keywords

and values, proceeded by the at sign (@) and delimited by the vertical bar character (|). Point by Address descriptions may be fully qualified with a project name before the @ sign. The following are some sample Point by Address descriptions:

@DEVICE=MY_DEVICE|ADDR=%R100

\\MYPROJECT\DEVICE=DEV1|ADDR=%M100|TYPE=BOOL

- Keywords supported for Point by Address descriptions.
- Keywords not acceptable as Point by Address descriptions.

Keywords supported for Point by Address descriptions

The following keywords are supported for Point by Address descriptions:

Keyword	Description
DEVICE	Required
ADDR	Required
TYPE	Optional
SCAN	Optional
OFFSET	Optional
ACCESS	Optional
ELEM	Optional
ORIGIN	Optional

If you do not use this keyword, the default is DEV.

The keywords may be specified in any order, but the required keywords must be included. Thus, the following are acceptable Point by Address descriptions:

`@TYPE=BOOL|DEVICE=DEV03|ADDR=%R100|OFFSET=3|ELEM=4`

`@ADDR=%R100|DEVICE=DEV04|TYPE=REAL|SCAN=4|ACCESS=WRITE`

`@DEVICE=DEV05|ADDR=%M100`

Keywords not acceptable as Point by Address descriptions

The following are not acceptable Point by Address descriptions:

`@ADDR=%R100|TYPE=INT`	(no device specified)
`@DEVICE=MYDEV`	(no device address specified)
`DEVICE=MYDEV|ADDR=%R100`	(missing @ sign)

`@DEVICE=DEV99|ADDR=%R1|OFFSET=4`	(integer point types cannot have a device address offset)

Within a point expression, Point by Address descriptors must be quoted; thus, you could display the sum of two registers in CimView:

```
'@DEVICE=MYDEV|ADDR=%R10' + '@DEVICE=MYDEV|ADDR=%R11'
```

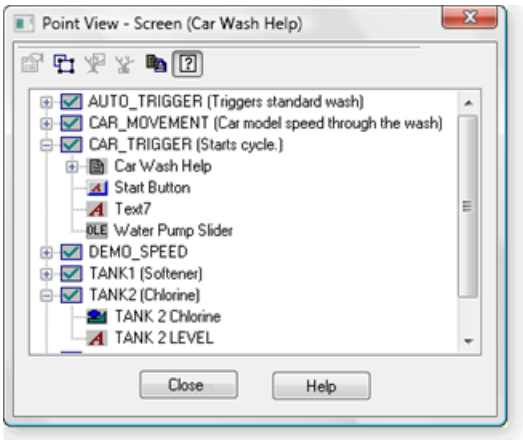
Point View

Point View

CimEdit provides you with an invaluable editing tool, the Point View window.

Two major benefits are that Point View:

- Eliminates guessing about what Point IDs you are using on your screen and with what items they are associated and
- Speeds up editing time by letting you access any item's Properties dialog box with one easy click.



1 *(page 230)*	Open the Point View window.
2 *(page 232)*	Review the Point View tree.
3 *(page 234)*	Display options in the Point View window.
4 *(page 243)*	Make changes through Point View.

5 *(page 248)*	Open runtime tools through Point View.
6 *(page 251)*	Copy point lists to the Windows clipboard.

Note:

- Currently the Point View window does not display information about Point ID's used by scripts, variables or CIMPLICITY ActiveX controls.
- The Point Control Panel button and Quick Trend button display only when you open Point View through CimView; this includes testing a CimEdit screen.
- The Select Objects Using Point button is enabled only in CimEdit.

1. Open the Point View Window

When the Point View window opens, the information it displays depends on what you select before you open it. Once opened, you can expand or collapse your view at any time.

Tip: To select an object or group that is embedded in a group or frame container before Point View is opened, open the group or frame container before you make your selection. When Point View opens, its contents will reflect exactly what is selected. Once opened Point View enables you to review the expression for any object in the group that has an expression.

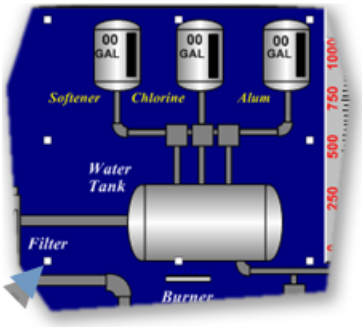
1 *(page 230)*	Select what object will be reviewed.
2 *(page 231)*	Open Point View for your selection.

Select what object will be reviewed.

Do one of the following.

- Review the points associated with only one object or group.

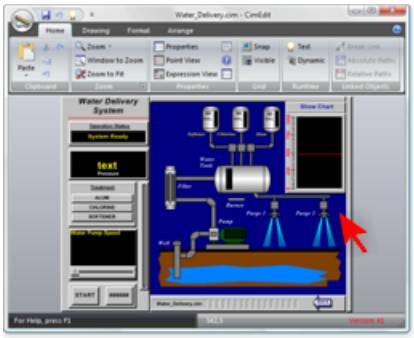
Select an object or group on the CimEdit screen before you open the Expression View window.



This will bring you directly to the information you are looking for. You can expand your view when the Expression View window is open.

- Display the points for the entire CimEdit screen.

Place the cursor anywhere on the screen.

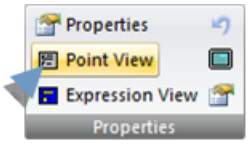


Open Point View for your selection.

Ribbon bar

CimEdit only

Click the Point View button in the Home>Properties group on the CimEdit Ribbon bar.



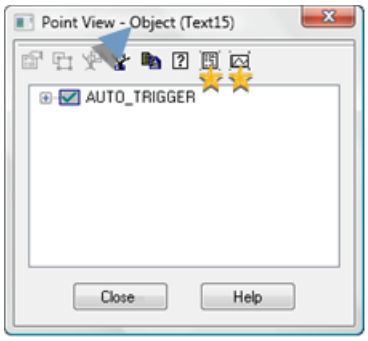
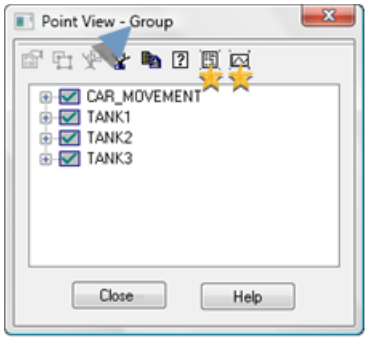
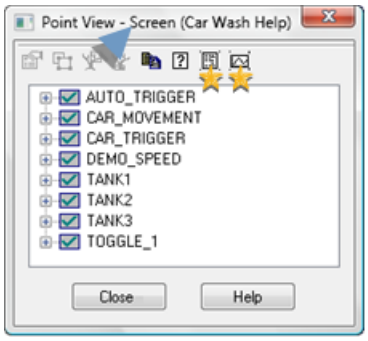
Popup menu

CimEdit and CimView

1. Click the right-mouse button.

2. Select Point View on the Popup menu.

Result: When the Point View window opens, the expressions you see are being used by the object, group or screen you selected.

Object	
Group	
Screen	

Note:

Point Control Panel and Quick Trend buttons display

- In CimView only.
- Only if the user is authorized to use those features.
- The window is re-sizable. The Title bar always reflects the object type and name of the current base object.

2. Review the Point View Tree

When you open *(page 230)* the Point View window you see a list of all the point ID's associated with the object you selected.

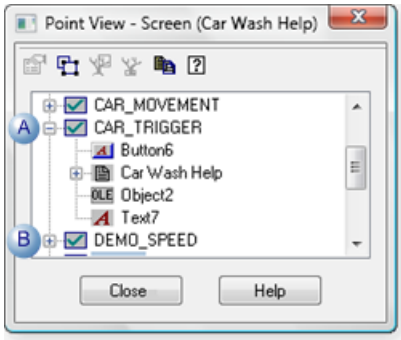
- Expand/collapse the tree.
- Icons in the Point View tree.

Expand/Collapse the Tree

Choose one of the following.

Tree Icons

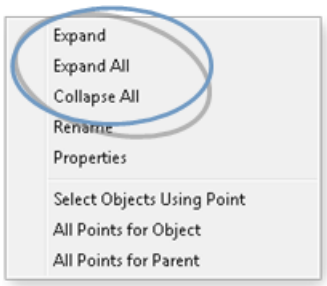
1. Click ⊞ to expand the view.
2. Click ⊟ to collapse the view.



A	Click ⊟ to collapse the view.
B	Click ⊞ to expand the view.

Popup Menu

3. Select any item in the Point View list.
4. Click the right-mouse button.
5. Select from the available expand/collapse options on the Point View Popup menu.



Note: Currently, the Point View window does not display information about Point IDs used by scripts, variables, or ActiveX controls.

Icons in the Point View Tree: Point View displays the following icons to represent objects on the CimEdit/CimView screen.

	Screen
	Group
	Line object
	Shape object
	Procedure **Note:** Procedures display for base objects only, when there is a single object selected. They will not display if they are nested in a group.
	Button
	Text

3. Display Options in the Point View Window

3. Display Options in the Point View Window

Expression View provides the tools to quickly find and review the:

3.1 *(page 234)*	Points associated with a selected object.
3.2 *(page 241)*	Objects associated with a selected point.
3.3 *(page 242)*	Point Descriptions in Point View.
3.4 *(page 243)*	Invalid points being used for a CimView screen.

3.1. Display Points Associated with an Object

3.1. Display Points Associated with an Object

When the Expression View window is open, you can display expressions for any object or group on the CimEdit/CimView screen.

- Display points for the object or parent: Overview

- Display points for the object or parent: Detailed

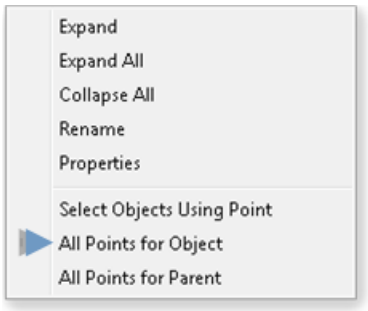
Display Points for the Object or Parent: Overview

- All points used by the highlighted object.
- All points used by the parent of the base object.

All expressions used by the highlighted object

Do one of the following.

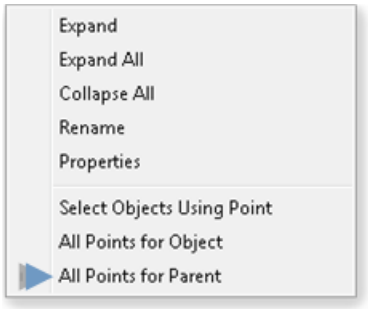
- Click the **All points for object** button to display the selected item as the base object.
- Right-click the object; select **All POints for Object** on the Popup menu.



All expressions used by the parent of the base object.

Do one of the following.

- Click the **All Points for Parent** button to display the parent as the new base object.
- Right-click the base object; select **All Points for Parent** on the Popup menu.



Display Expressions for the Object or Parent: Detailed

3.1.1 *(page 318)*	Base object/parent levels defined.
3.1.2 *(page 320)*	All expressions by a highlighted object.

3.1.1. Base Object/Parent Levels Defined

By incorporating base object/parent functionality, in addition to its expand/collapse features, the Point View window displays exactly the information you are looking for.

The two categories of objects are as follows.

Object	Description
Base	Object that is selected. The first object selected as the base can range from one object to the entire screen.
Parent	Item that is one level above the base object. The parent can range from a group of two items to the entire screen.

Select a parent of a base object as follows.

1 *(page 236)*	Select a single object as the base object.
2 *(page 237)*	Select the parent object of the base object.
3 *(page 237)*	Continue selecting "up" until the screen becomes the base object.
	Guidelines for frames as base and parent objects.

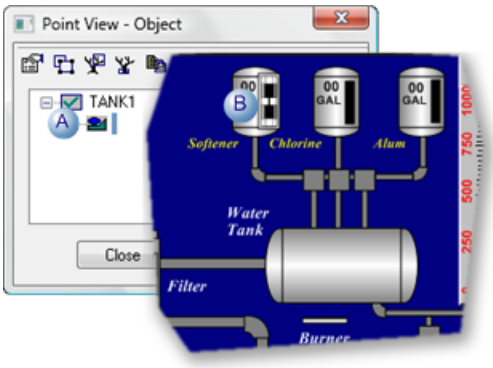
Select a single object as the base object.

1. Select an object.

Note: The object can be in an open group *(page 144)* .

1. Open the Point View window.
2. Select the object in the window.

CimEdit does the following.



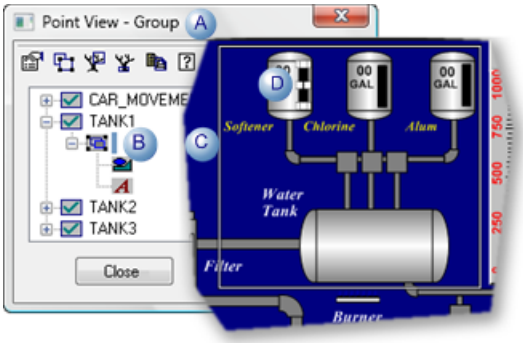
A	The Point View window opens at the Object level.
B	A frame will surround the object if it is selected in the Point View window.

Select the parent object of the base object.

1. Click to make the object's parent object the base object.

CimEdit does the following.

The next parent object up becomes the base object.



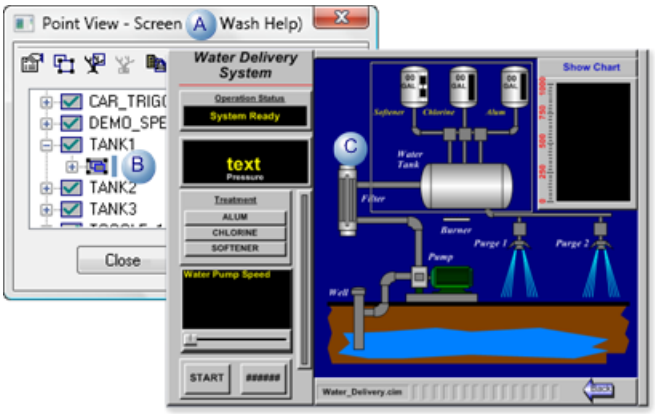
A	The Point View window changes to the Group level.
B	Any object or group can be selected in the Point View window.
C	A frame will surround the object or group that is selected in Point View. **Note:** if the object extended to the group's borders the frame will be the same size as the frame surrounding the object.
D	The originally selected object may still be selected in addition to the selection through Point View.

Continue selecting "up" until the screen becomes the base object.

1. Select the new base object.
2. Click if you want the next parent object to become the base object.

CimEdit continues to select the next parent object.

When is clicked and the screen is the parent of the base object CimEdit does the following.



A	The Point View window changes to the Screen level.
B	Any object or group can be selected in the Point View window.
C	A frame will surround any object or group on the CimEdit screen that is selected in the Point View window.

Guidelines for Frames as Base and Parent Objects

Following are guidelines for dealing with frames *(page 410)* as base objects and parents.

In CimEdit, when you select a frame object on your screen and then enter Point View, if the:

- Frame container is closed and you:
- Select to view the parent of the frame object, the frame container becomes the base object.
- Select an object in the frame container as the base object, the selected frame becomes visible and the object is highlighted on the CimEdit screen.
- CimEdit Screen is in frame mode and you:
- Select to view the parent of the Frame Container as the base object in Point View, frame mode is closed on the CimEdit screen.
- Select the Frame Container as the current object, **Select Objects Using Point** in Point View is disabled.
- Use **Select Objects Using Point** in Point View, only the top-level objects for the current frame are selected.
- Select to view the parent of a Frame Container, and then go back into it, you will not return to frame mode in the CimEdit window.

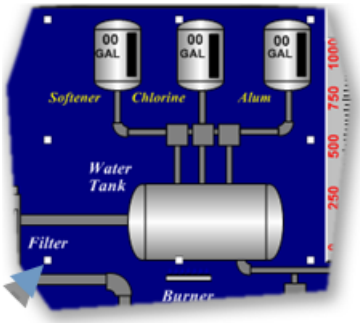
In CimView, if you select a frame container as the base object and open the Point View window, then select any object in the frame container, the selected frame object on your CimView screen does not change.

3.1.2. All Points for a Highlighted Object

You can use the Point View window to quickly find all objects on the CimEdit screen that use a particular Point ID. You can then manipulate the selected objects on the CimEdit screen after you close the Point View window.

Point View provides an easy way to quickly review all of the points that may be associated with a selected object. This is particularly useful if the object is several levels down in a group. It enables you to avoid taking the time to open each group level to access the object's Properties dialog box in order to see the associated points.

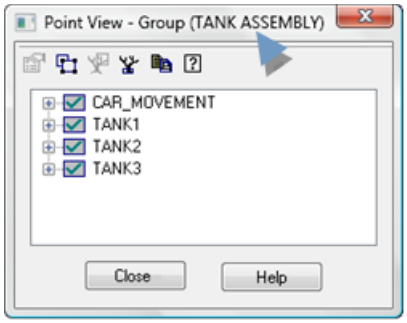
1. Select the group that includes the object to be reviewed.



   Tip: Select the screen if you are not sure what group to select.

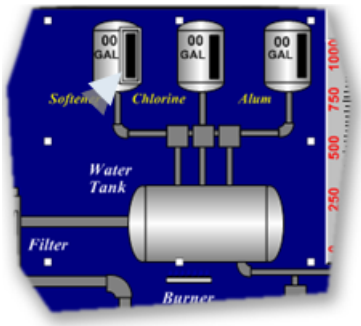
2. Open *(page 230)* the Point View window.

   Point View displays the points for the selected object/group.

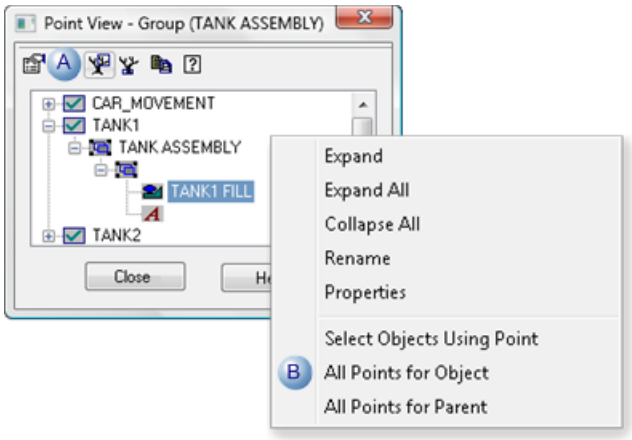


3. Expand points in the list to find and select the object or group to be reviewed.

   Note: Items that you select in the list are surrounded by a frame on the CimEdit screen.

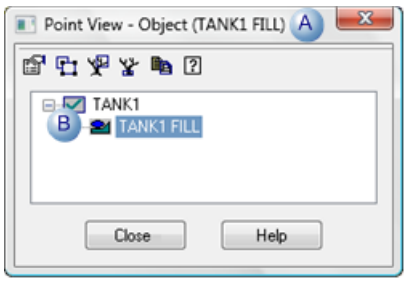


4. Do one of the following.



A	Click the All Points for Object button on the Expression View toolbar.
B	a. Right-click the selected object. b. Select **All Points for Object** on the Popup menu.

Result: The Point View window does the following.



A	Changes to the selected object display.
B	Displays the points associated with the selected object/group.

Important: If the Point ID is being used by an object in a Group, the Group is selected. If the Point ID is being used by a frame in a Frame Container, the Frame Container is selected.

Example

You want to delete all the objects that use Point ID TANK1:

5. Right-click the selected object.
6. Select **All Points for Object** on the Popup menu.
7. Select ROBOT_MOVEMENT in the Point View window.
8. Click the Select Objects Using Point button .
9. Close the Point View window.
10. Press Delete on the keyboard.

    Remember that if TANK1 is being used by an object in a group or frame container, the entire group or frame container will be deleted.

3.2. Display Objects used by a Selected Point

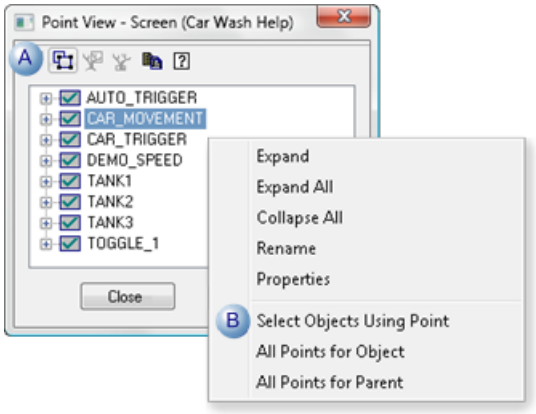
(CimEdit only) Point View lists all the objects that use a point and frames a selected object on the screen.

1. Open *(page 230)* the Point View window for the screen or group you want to review.
2. Select the point to be reviewed.
3. Do either of the following.

   Select all objects that are used by the point.

   Select a point.

   Do one of the following.



A	Click the Select Objects Using Point button.
B	a. Right-click the expression. b. Choose Select Objects Using Expression on the Popup menu.

4. Right-click the expression.

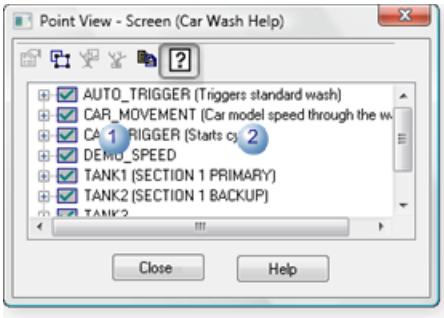
5. Choose Select Objects Using Expression on the Popup menu.

3.3. Display Point Descriptions in Point View

You can display descriptions for point ID's that were entered in the Point Properties dialog box when you review points in the Point View window.

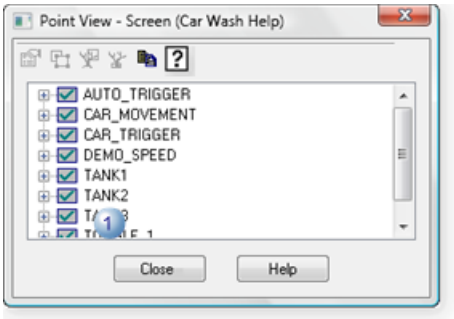
Click the Point Description button on the Point View toolbar.

The point descriptions display next to the Point ID's.



1	Point ID
2	Descriptions entered in the point's Point Properties dialog box.

Click the Point Description button again to hide the point descriptions.

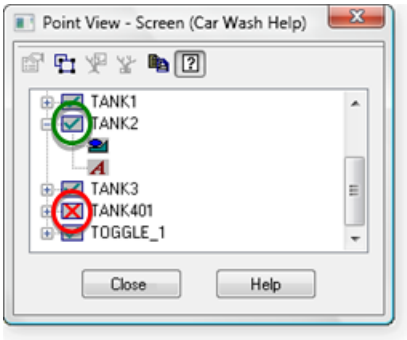


1	Point ID

3.4. Find Invalid Points being used for a CimView Screen

Point View enables you to easily find any invalid points on the CimEdit screen.

The Point View tree displays the following for the points in the object, group or screen that you select.



☑	Valid point
☒	Invalid point

4. Make Changes through Point View

4. Make Changes through Point View

4.1 *(page 244)*	Rename a non point item through Point View.
4.2 *(page 245)*	Modify a highlighted non-point item in Point View.
4.3 *(page 247)*	Replace a point through Point View.

Note: In CimView, you can look at the item's properties, but you cannot display the properties of a Point ID or a frame object.

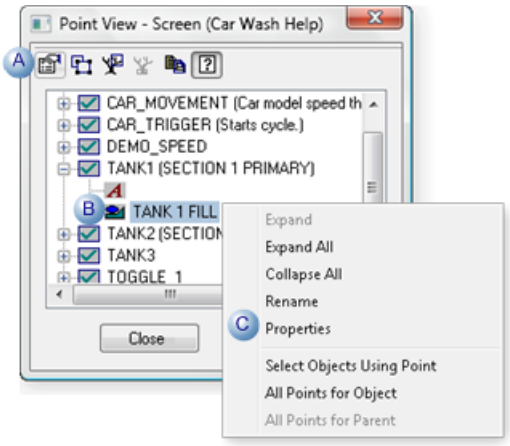
4.1. Open an Object's Properties Dialog Box through Point View

1. Select an object in the Point View window.

   Note:

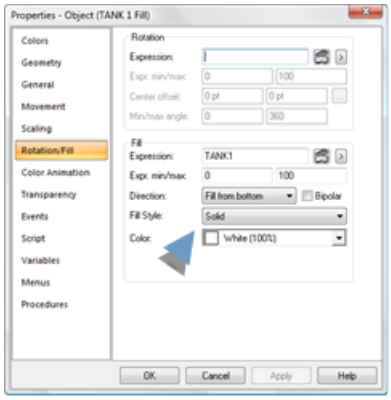
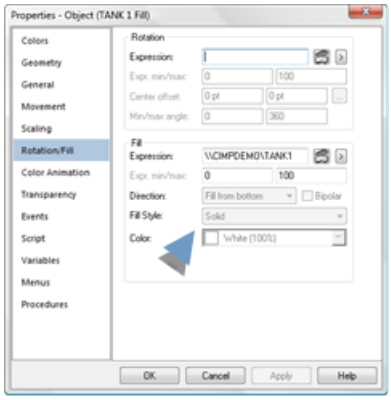
   - The object that is at the bottom of an expanded tree contains the selected point. However, an object above the bottom may also include that point. You can check for duplications and other possible inconsistencies by copying *(page 327)* the point list to the clipboard and reviewing it in an application such as Excel.
   - You can open the Properties dialog box for any object in the tree.

2. Do one of the following.



A	Click the Properties button on the Point View menu bar.
B	Double-click the object.
C	a. Right-click the object. b. Select Properties on the Popup menu.

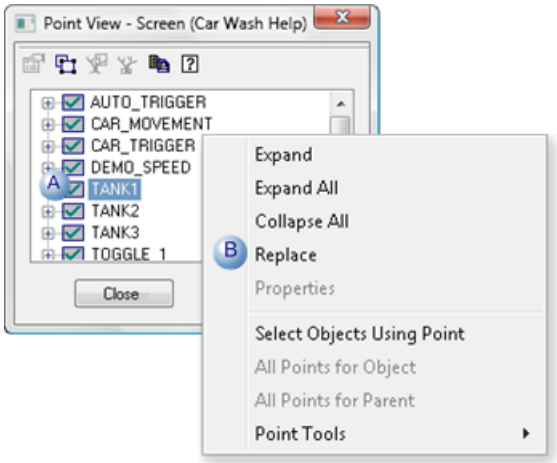
The Properties dialog box opens for the selected object.

CimEdit	Any of the object's properties, including points, can be configured. 
CimView	The properties can be reviewed, but are read-only. 

4.2. Replace a Point through Point View

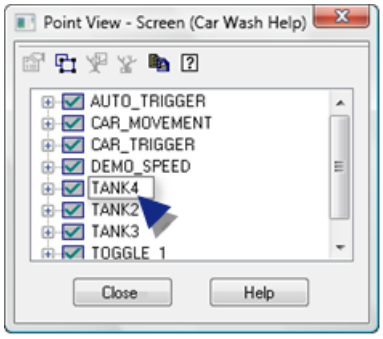
(CimEdit only)

1. Select a point in the Point View window.
2. Do the following.



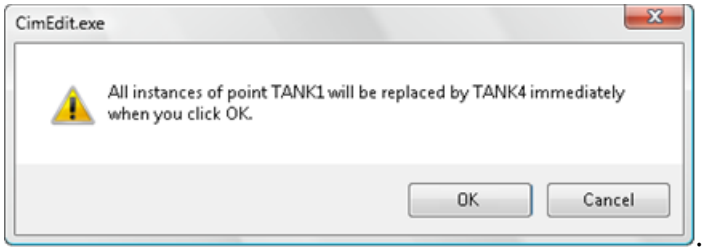
A	Right-click the point in the Point View window.
B	Select Replace on the Popup menu.

3. Enter the new name.



4. Press Enter on the keyboard.

   A messages displays asking you to confirm the replacement.



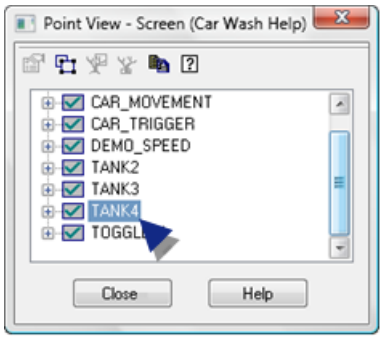
5. Click OK to confirm.

   Note: Clicking Cancel cancels the replacement.

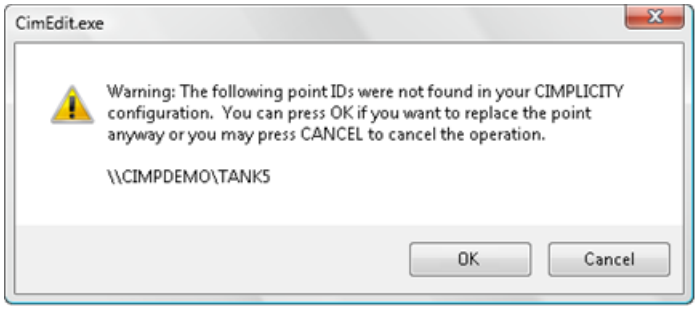
The following happens when you click OK.

If the new point

Exists	CimEdit proceeds with the replacement. The replacement point location in the Point View ascending list is based on its name static to other points in the list.



Does not exist	A message displays warning that the point does not exist.



Important: If you change the name of a Point ID to an array point element:

- An array point element cannot be used in alarm state color animation.
- You may get invalid expressions if the point you are replacing
- Already has an element specified
- Is used in alarm state functions

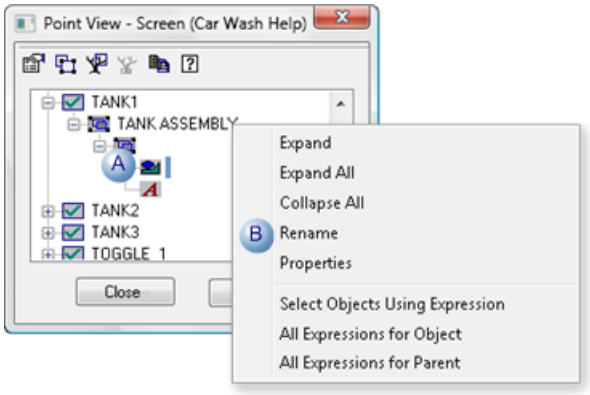
A Point Replace Warning dialog box reviews the array point restrictions and asks you if you want to replace the Point ID anyway.

Click to Replace the Point ID or Cancel your request.

4.3. Rename a Non-point Item through Point View

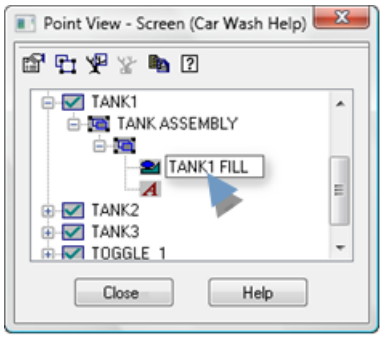
(CimEdit only)

1. Select an item in the Point View window.



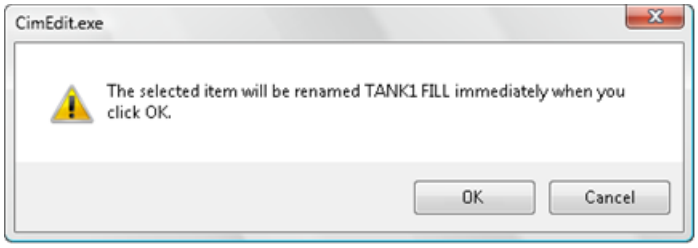
A	Right-click an object in the Point View window.
B	Select Rename on the Popup menu.

2. Type the new name over the old.



3. Press Enter on the keyboard.

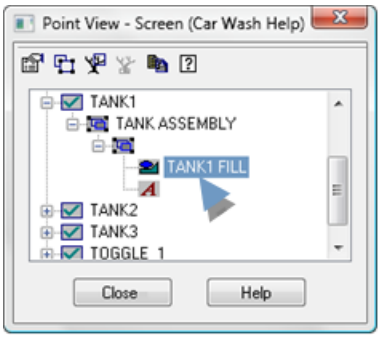
   A messages displays asking you to confirm the change.



4. Click OK to confirm.

   Note: Clicking Cancel cancels changing the name.

Point View renames the item.



Note: The changed name also displays in the General>**Object name** field in the objects Properties dialog box.

5. Open Runtime Tools through Point View

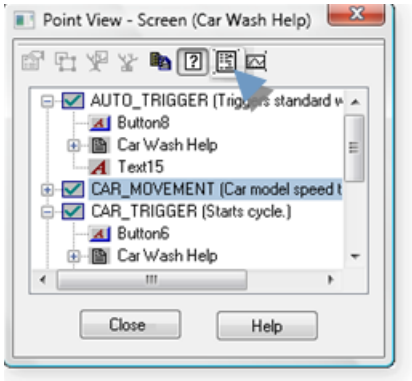
5. Open Runtime Tools through Point View

5.1 *(page 249)*	Display point values in the Point Control Panel
5.2 *(page 250)*	View quick trends through Point View

5.1. Display Point Values in the Point Control Panel

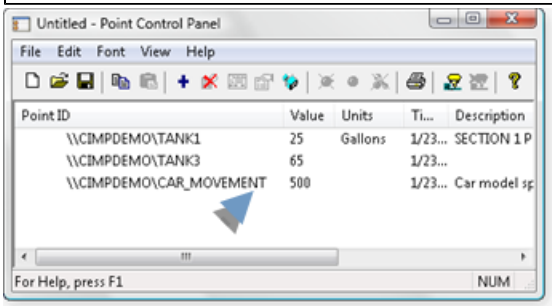
If your role has authorization , you can monitor point values in the Point Control Panel through Point View.

1. Make sure the project is running.
2. Open Point View *(page 230)* through a CimView screen.
3. Select a point ID.
4. Click the Point Control Panel button.



One of the following occurs:

If the Point Control Panel is not open,	The Point Control Panel opens displaying the point data.
When the Point Control Panel has already been open through Point View.	The point data is added to the end of the list.

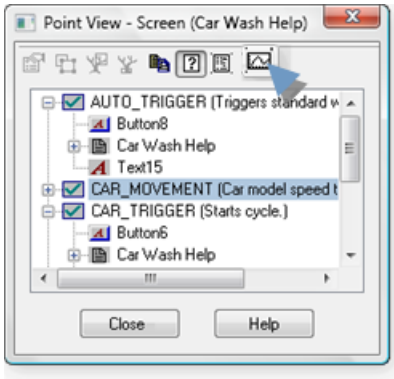


Note: If you do not have authorization to view Quick Trends, the Quick Trends button will not display on the Point View toolbar.

5.2. View Quick Trends through Point View

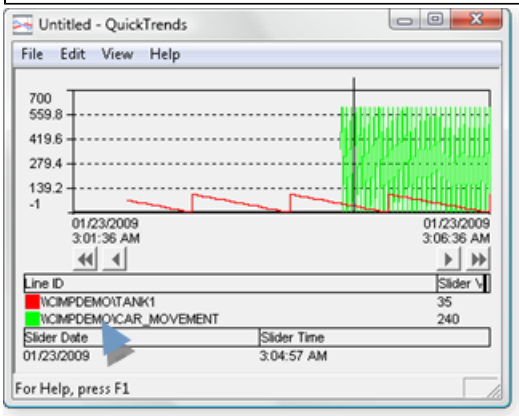
If your role has authorization , you can review point quick trends through Point View.

1. Make sure the project is running.
2. Open Point View *(page 230)* through a CimView screen.
3. Select a point ID in the Point View window.
4. Click the Quick Trend button.



One of the following occurs:

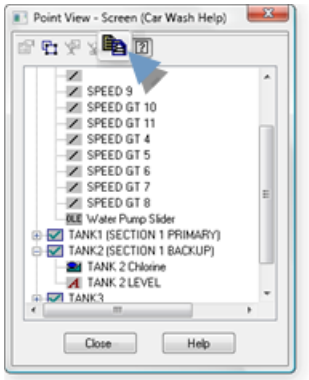
If Quick Trend is not open	A Quick Trend opens and starts displaying a trend of the point's data.
When Quick Trend has already been opened through Point View	The selected point's trend line is added to the trend chart.



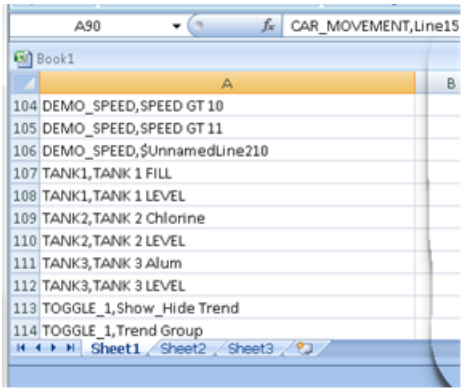
Note: If you do not have authorization to view Quick Trends, the Quick Trends button will not display on the Point View toolbar.

6. Copy Point Lists to the Windows Clipboard

1. Open Point View for a selected object, group or the screen.
2. Click the Paste to Clipboard button on the Point View toolbar.



3. Paste the data into a text editing application, e.g. Excel.



Each item in the expression list provides the information displayed in Expression View.

Note:

- You can paste lists from several base objects into Excel; sort the list and review where expressions may be missing, duplicated or inconsistent.
- If an object on the screen invokes a script that replaces one point with another during runtime, e.g. TANK201 replaces TANK101, both points display in Point View. However, if you copy Point View data to the clipboard, only the second point, e.g. TANK201 will be included.

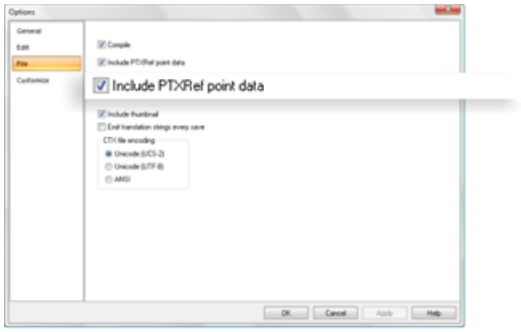
CimEdit Point Data in Point Cross Reference

1. Click the CimEdit button .

   The Options dialog box opens.

2. Select **File**.

3. Check or clear Include PTXRef point data to do the following.



Check	• CimEdit saves the list of points the screen uses in a stream that Point cross reference can read from the .cim file. • The data is available through Point Cross Reference.
Clear	• The screen size is slightly smaller than when Include PTXRef point data is checked. • The data is not available through Point Cross Reference.

CimEdit Variables

CimEdit Variables

A variable can be used in an expression to represent different types of values. When you create an object that uses variable IDs, you can use the same object in several locations on a CimEdit screen, or on several different screens. You can then assign the variable a different value, for example different point IDs, for each instance of the object. This can streamline your configuration time.

Variables becomes a particularly powerful tool when you use a variable in linked objects.

A variable ID's chameleon-like functionality behaves as follows:

1. A screen designer enters text to identify a Variable ID (on the Variables tab) in CimEdit.

   The text may represent any of several types of values.

2. Any of several options can assign a value to the variable.

   Options include:

- Scripts, which can include variable IDs to which values are assigned.
- Values assigned by a user during runtime.
- Values assigned by a screen designer while configuring the screen.

Values that a variable ID can represent include a:

- Full Point ID
- Partial Point ID
- Text string in an expression.

3. CimView substitutes the text into the appropriate expression.

4. CimView evaluates the expression.

Example of Variable Behavior

Step 1	CimView substitutes specified values for the Variable ID's text.		
	Variable ID Expression		
	{myvar} * 2		
	tank{myvar}_level * 2		
	{myvar}		
Step 2	CimView evaluates the substituted expression.		
	Resulting Expression		
	tank1_level * 2		
	tank1_level * 2		
	tank1_level * 2		

Variables as Data Sources within CimView

Variables *(page 252)* are flexible containers for information and do not add to your project's overhead. A variable has a unique name, a variable ID, and represents a value that can be assigned to it during configuration of the CimEdit screen or at runtime.

Some of the data sources it can represent are:

- Full Point ID
- Partial Point ID
- Text string in an expression.

A variable does not communicate with the PLC but is wholly contained in CimEdit.

Example of a Variable

Different values (types of values) are assigned to a variable ID **{var_value}** for different items on a CimEdit screen.

Variable ID	Value
var_level	**40**
var_level	**tank_level** (point)
var_level	**tank_level+50**
var_level	Assigned during runtime

In addition to providing you with several options for assigning values, variables can streamline your configuration time. For example, when you create an object that uses variable IDs, you can use the same object in several locations on a CimEdit screen, or on several different screens and assign different text strings to each instance of the object.

Variable Evaluation Hierarchy

A Variable ID can be created at any level of the CimEdit screen hierarchy. This includes:

- The screen
- A group
- A subgroup (nested groups)
- An object

If you define the same Variable ID at more than one level of the hierarchy, CimEdit:

1. First looks for the Variable ID in the object you are configuring.
2. Goes to the next level up in the hierarchy, if the Variable ID is not there.
3. Continues to the next levels, to find an object that references that variable, if necessary.

CimEdit System Variable: $FILENAME

CimEdit provides the following system variable

$FILENAME

Purpose	To display the name of the screen. **Note:** The value only contains the name of the root file and does not include its path or extension.
Available	For any object on the screen in any field that accepts variables.
Comments	The variable is read-only at the screen level.

CimEdit System Variable: $ZOOM_LEVEL

CimEdit provides the following system variable

$ZOOM_LEVEL

Purpose	To display the CimView zoom percent. **Note:** The value updates every time the screen zoom changes.
Available	For any object on the screen in any field that accepts variables.
Comments	The variable is read-only at the screen level.

Variables Configured and Used

Variables Configured and Used

Step 1 *(page 255)*	Configure public or private variables.
Step 2 *(page 257)*	Select a Variable ID.
Step 3 *(page 259)*	Set the value of the Variable ID.

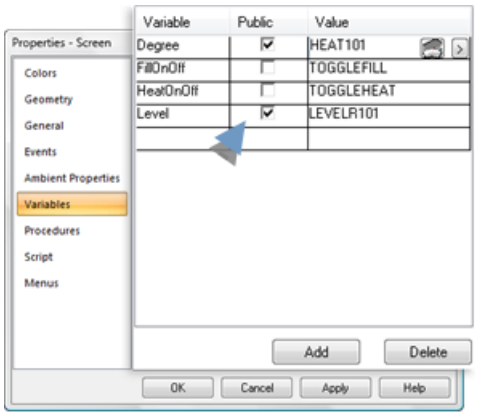
You can then use the Variable ID in your configuration and set its value, if it is still open, either during configuration or runtime.

Step 1. Configure Public or Private Variables

- Variable configuration.
- Example: CIMPLICITY gauge with predefined variable IDs.

Variable Configuration

1. Open the Properties dialog box for the object that is at the appropriate level for creating a variable.
2. Select the Variables tab.
3. Click Add and enter the following for each variable you want to create.



Field	Description
Variable	Unique name
Public	Declares variable for a linked object as public or private.
	Check
	Clear
Value	(Optional) Value can be a: • Point • Partial point • Text String in an expression

Note: You can delete any variable by selecting it and clicking the **Delete** button.

4. Click **Apply**.

   Result: You now have the Variable IDs available for the selected item and any item below it in the configuration hierarchy.

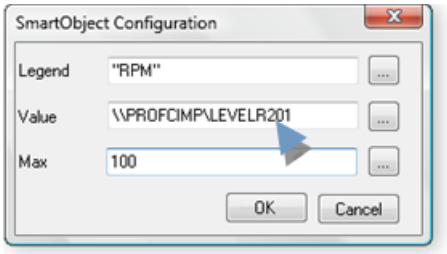
   Tip: Take advantage of CIMPLICITY's many graphic objects in the Symbols and Smart Objects Library that come with predefined Variable IDs.

   Example: CIMPLICITY Gauge with Predefined Variable IDs

5. Select a pressure gauge from the Smart Object Gauges library.

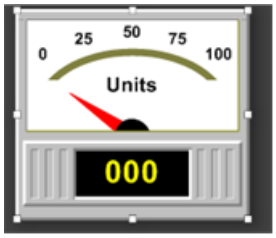
   A SmartObject Configuration dialog box opens.

6. Enter a **Value**.



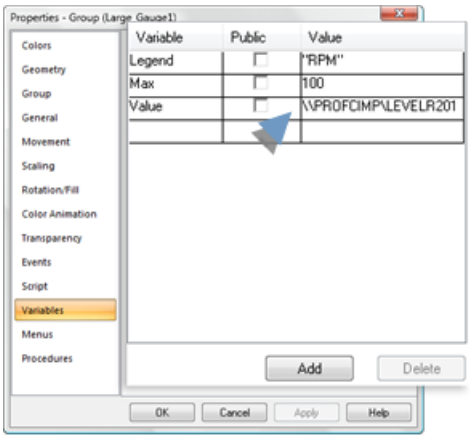
7. Click OK.

   The gauge is placed on the CimEdit screen.



8. Open the gauge's Properties dialog box.
9. Select the Variables tab.

The pre-defined variables display on the Variables tab.



You can edit and / or set the values according to your needs.

Step 2. Select a Variable ID

- Open the Select Variable window.
- Example: Variable hierarchy.

Wherever you can enter a Variable ID in a CimEdit Properties dialog box, you can browse through the list of Variable IDs that you defined in the Variables tab.

Open the Select Variable window

1. Click the Popup Menu button to the right of any field that allows variable ID's.

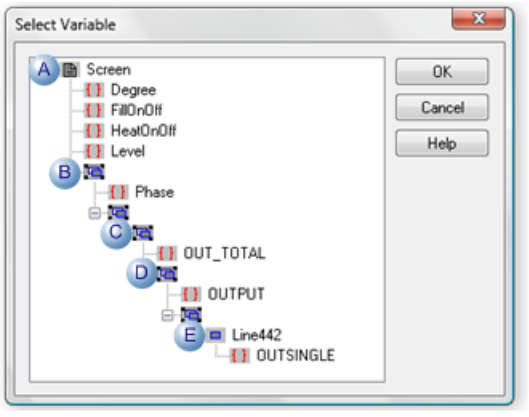
   A Popup menu opens.

2. Select Variable Browser.

   The Select Variable browser opens.

   Variables that are available for the selected object display.

   Example



	Variable Configuration Level
A	Screen
B	Group
C	Sub-group
D	Sub-sub-group

	Variable Configuration Level
E	Object

The following icons define the item in the list.

Icon	Item
	Screen in which the variable is defined
	Variable ID
	Group in which the variable is defined
	Object in which the variable is defined

When you select a variable ID for an object from the tree, make sure that you select one that is above the object and in its path.

Example: Variable hierarchy

An object displays a single line that represents water output from a tank.

Variables that are available to it are variables that are directly above it in the screen's hierarchy.

Hierarch Levels	Available Variables
Screen	`{Degree}`, `{FillOnOff}`, `{HeatOnOff}` and `{Level}`
Group Top Level	{Phase}
Sub-group	`{OUT_TOTAL}`
Sub-sub-group	`{OUTPUT}`
Object	`{OUTSINGLE}`

Step 3. Set the Value of a Variable ID

Once you select a Variable ID, there are several ways to set its value and several ways to use it.

- Value set for a variable ID.
- Variables in setpoint actions.

Value Set for a Variable ID

One way is to assign values to the variable ID's on the Variables tab of a Properties dialog box. In addition, you can create a:

- Variable Assign action

- Assign values to the Variable ID's automatically when the user executes the action at runtime.
- Let the user assign values to the Variable ID's when the user executes the action at runtime.

- Screen Open event that assigns values to the Variable ID's when a user opens the screen in CimView.

- Invoke Script procedure and have the script assign values to the Variable ID's. This procedure can be assigned to any event.

Variables in Setpoint Actions

When you use a Variable ID in a **Setpoint** action, the association between the **Setpoint** action and the Variable ID is made every time the action is executed.

A major benefit of using variables in setpoint actions is that the **Setpoint** action can be set to different values at different times.

CAUTION: If you do use Variable IDs in **Setpoint** actions, especially if you are using nested *(page 254)* Variable IDs, make sure that the Variable ID is referencing the correct object and value.

Variables Examples

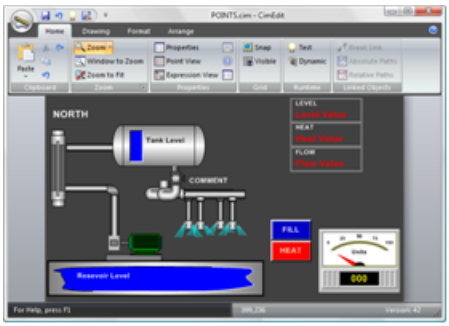
Variables Examples

Example 1 *(page 260)*	Use a variable ID in a toggle setpoint action
Example 2 *(page 265)*	Use variables in scripts.

Example 1: Use a Variable in a Toggle Setpoint Action

Example 1: Use a Variable in a Toggle Setpoint Action

A CimEdit/CimView screen uses several points and variables to display a reservoir level and temperature.



A new variable needs to be added and applied to

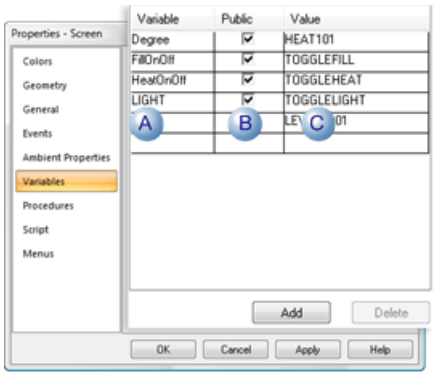
- Turn the area lights on and off.
- Display the lights' state.

Step 1 *(page 261)*	Create a variable to hold the selected value.
Step 2 *(page 262)*	Configure a button to toggle the lights on/off.
Step 3 *(page 263)*	Apply color animation to the On/Off button.
Step 4 *(page 263)*	Configure a rectangle object to reflect lights on/off.
Step 5 *(page 264)*	Test the new variable configuration.

Step 1. Create a Variable to Hold the Selected Value

1. Opens the screen's Properties dialog box.
2. Selects Variable.
3. Adds a new variable.

   The variable is:

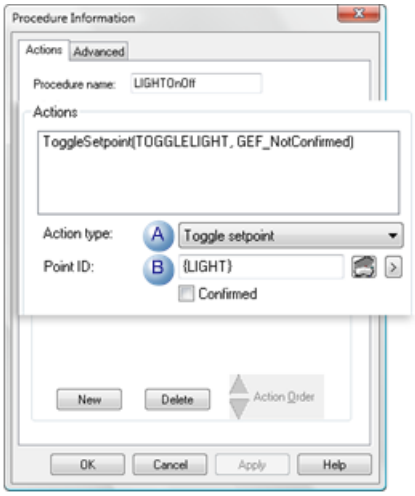


	Column	Entry
A	Variable	LIGHT
B	Public	Checked **Note:** The designer will distribute the screen to other systems. Because variables are being used, the only configuration that will have to change will be the value assignments, where necessary.
C	Value	Point named TOGGLELIGHT.

Step 2. Configure a Button to Toggle the Lights On/Off

1. Places a button on the CimEdit Screen.
2. Labels the button LIGHT.

3. Opens the button's Properties dialog box.
4. Does the following.



A	Creates a `Toggle setpoint` procedure to turn the light on and off.

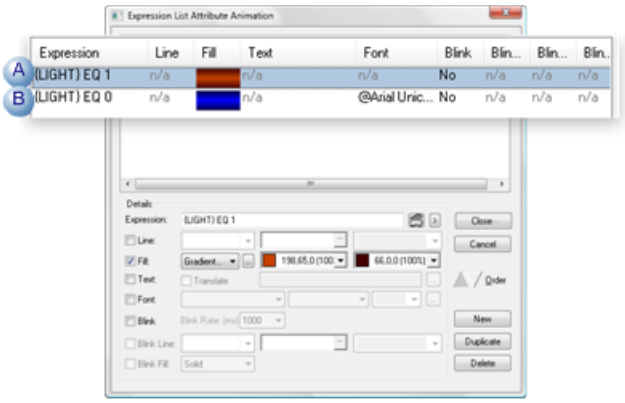
B	Selects the `{LIGHT}` variable in the **Point ID** field.

Step 3. Apply Color Animation to the On/Off Button

1. Selects Color Animation in the button's Properties dialog box.
2. Clicks Edit in the Expression List section.

   The Expression List Attribute Animation window opens.

3. Enters expressions for the animation

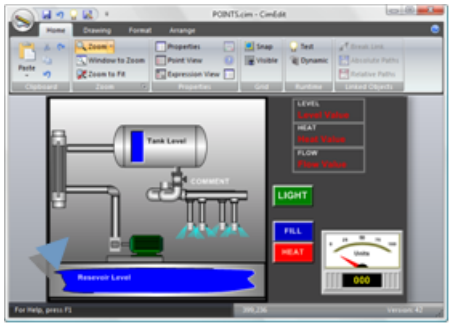


A	`{LIGHT} EQ 1`	Turns the lights on.
B	`{LIGHT} EQ 0`	Turns the lights off.

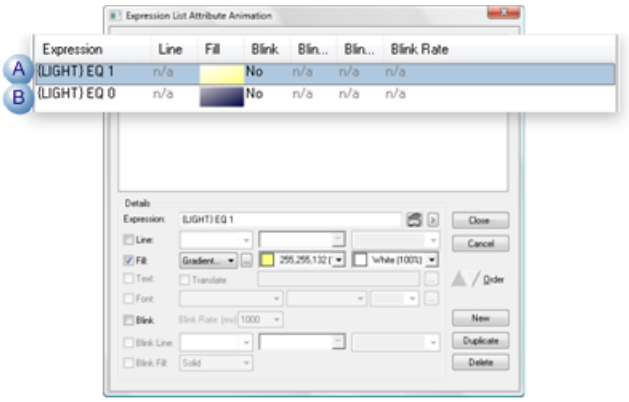
4. Completes the button configuration.

Step 4. Configure a Rectangle Object to Reflect Lights On/Off

1. Inserts a rectangle object in the reservoir group.



2. Opens the rectangle object's Properties dialog box.
3. Enters expressions that use the variable `{LIGHT}` to apply color animation to the rectangle.



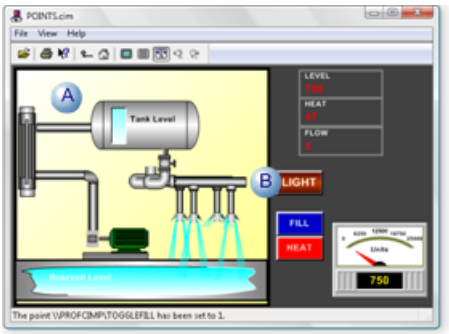
A	`{LIGHT} EQ 1`	Turns the lights on.
B	`{LIGHT} EQ 0`	Turns the lights off.

Step 5. Test the New Variable Configuration

1. Opens the screen in CimView.

   The TOGGLELIGHT point value is 1.

   The following CimView screen objects that were assigned the `{LIGHT}` variable display the configuration for the value 1.

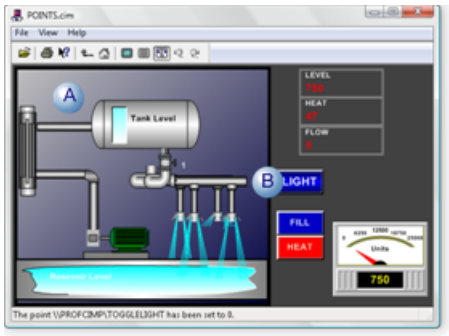


A	Rectangle
B	Button

2. Clicks the LIGHT button.

   The TOGGLELIGHT point value changes to 0.

   The following CimView screen objects that were assigned the `{LIGHT}` variable display the configuration for the value 0.



Example 2: Use Variables in Scripts

You can use and assign Variable IDs in CimEdit scripts.

The following example shows you how to use a Variable ID in a script.

```
Sub OnMouseUp(x As Long, y As Long, flags As Long)
   Dim var As CimObjectVariable
   Set var = CimGetScriptOwner().GetVariable("PointValue")
   i = PointGet(var.Value)
   PointSet var.Value, i+1
End Sub
```

The following example shows you how to assign a Variable ID in a script.

```
Sub OnMouseUp(x As Long, y As Long, flags As Long)
   Dim var As CimObjectVariable
   Set var = CimGetScriptOwner().GetVariable("PointValue")
   Var = "DEMO_SAWTOOTH"
   i = PointGet(var.Value)
   PointSet var.Value, i+1
End Sub
```

Recursive Variables

CIMPLICITY's ability to evaluate variables that are used in expressions that are contained in screens include the following.

- Overview
- Parent Variables

Overview

1. When evaluating an expression, there will now be up to 100 passes over the expression until there are no more variable references in that expression.

Note: A variable reference is indicated by enclosing a variable name in {}.

2. Because it is common for a variable value to be used as both a label in a text object's display expression and as part of a point ID, in some animations a new mechanism to allow a variable to have its value substituted into an expression but still be double quoted at the final pass of variable substitution in an expression is now available. The new mechanism is using a pipe character | in the expression at the location where the double quote would be used.

   Note: The pipe substitution only occurs if variables were used in the expression. However, pipes can still be contained in double quoted strings.

   Example: Pipes in Recursive Variables

3. A linked object has a public variable that contains 2 objects.

   The display expression is as follows.

   Note: These are the configured expressions of the inner most linked objects.

Expression 1	`{unit}.pv`
Expression 2	`|{unit}|`

4. This linked object is contained inside another linked object with its own public variable `area`.

5. The public variable `unit` is assigned the following value.

unit	`{area}.vlt`

6. The variable `area` is given the value as indicated below.

area	Bay12

7. The objects indicated in the inner most linked object would evaluate with the following expressions.

Final Expression 1	Bay12.vlt.pv
Final Expression 2	"Bay12.vlt"

   No lines of code are needed to accomplish the recursive variable evaluation when the pipe is used.

   Parent Variables

   When recursive variables are used, they frequently will have a common variable that communicates some part of the system configuration. The common variable could occur in

several layers in a deeply nested hierarchy. For example, a group of objects might have a variable `OBJECTNAME` at more than one level to represent the object instance that will be used.

If the parent object has the same name as one of the variables in the hierarchy, a direct reference must be made to the parent's definition of the variable, not the lower variable's definition. If the referenced definition is not the parent, issues, such as circular references (e.g. VAR1=VAR1+3) can occur.

Note: The most common use case where these issues can occur is with linked objects *(page 328)*. With other objects you have complete control over the configuration, including variables, variable names and expressions that include variables. With linked objects you do not have complete control.

A relative notation is available to automatically, skip over one or more definitions in the object grouping's hierarchy to find the desired variable definition with the given name.

Begin the variable name with ..\, or ..\..\, or ..\..\..\, etc.

Example

In an expression:

Enter:	`{..\variable}`

Example

An `OBJECTNAME` (object ID) variable represents the object instance that will be used.

`OBJECTNAME` will occur at more than one level in a group of nested variables that are set up for a conveyor line section.

Setup

Conveyor Line Section

The conveyor line section, which is the top of its hierarchy:

- Is identified using the `OBJECTNAME` variable.
- Has the name: Conveyor South.
- Has two motors: Motor1 and Motor2.

Motors

The two motors are linked objects that

- Have the same variables to provide identification.
- `OBJECTNAME` variable out of the parent's variable
- `Motorn` variable suffix that identifies the specific motor.
- Have unique names: one is Motor1; the other is Motor2.

Motor	Variable ID
Motor 1	{..\OBJECTNAME}.{Motorn}
Motor 2	{..\OBJECTNAME}.{Motorn}

Parts

Each motor has two parts, which can be identified using a `.suffix`.

The parts are identified as a continuation of the variable concatenation.

Part	Variable ID
Part 1	{..\OBJECTNAME}.{..\Motorn}.Part1
Part 2	{..\OBJECTNAME}.{..\Motorn}.Part2

Results

Conveyor South is identified for each motor that is in its hierarchy.

Motors

Motor	Identification
Motor 1	Conveyor South.Motor1
Motor 2	Conveyor South.Motor2

Parts

Conveyor South is identified for each motor part that is in its hierarchy.

The correct motor (Motor1 or Motor2) is identified for the parts that are in its group

Motor1 Parts

Part	Variable ID
Part 1	Conveyor South.Motor1.Part1
Part 2	Conveyor South.Motor1.Part2

Motor 2 Parts

Part	Variable ID
Part 1	Conveyor South.Motor2.Part1

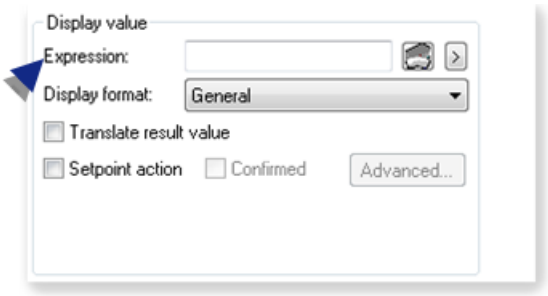
Part	Variable ID
Part 2	Conveyor South.Motor2.Part2

Expressions in CimEdit

Expressions in CimEdit

There are several places in CimEdit where you enter an expression as part of your configuration.

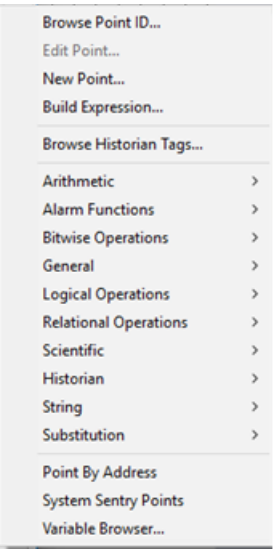
The expression is entered in an Expression box.



You can use the point, variable, and operation selections located to the right of an Expression box to create or edit an expression.

In the Properties window, to the right of the Expression box, select .

A menu opens with the following tools that help build expressions.



Option	Description
Browse Point ID	Opens the Select a Point window.
Edit Point	Opens the Point Properties window for a selected point.
New Point	Opens a New Point window.
Build Expression	Opens the Build Expression window.
View Expression	Opens the View Expression window. Note: • ◦ The View Expression option is available only during run time. It replaces the Build Expression option. ◦ The View Expression window is a read-only version of the Build Expression window, with all its fields disabled.
Browse Historian Tags	Opens the Select a Tag window.
Arithmetic	Displays extended menus with the expression operations and functions that are associated with the selected category.
Alarm Functions	
Bitwise Operations	
General	
Logical Operations	

Option	Description
Relational Operations	
Scientific	
Historian	
String	
Substitution	
Point By Address	Opens the Point By Address window.
System Sentry Points	Opens the System Sentry Address Builder window.
Variable Browser	Opens the Select Variable browser.

Select an operation to place it at the current position of the cursor in the input box. If the operation requires an argument, the cursor is positioned for you to type the argument.

Insert a point ID, point attribute, or variable either after the operator or between parentheses, if they appear with an operator.

Guidelines

- A CimEdit or CimView expression can be up to 3000 characters long.
- When you use a device point ID in the Expression Builder for virtual points, the raw value of the point is used by default.
- If you want to use the engineering units value, enter EU_CONV(<point_id>) in the expression.

CimEdit provides you with two powerful tools for creating and reviewing expressions.

They are:

1 *(page 271)*	Expression Editor
2 *(page 312)*	Expression View

1. Build Expressions

1. Build Expressions

1.1 *(page 272)*	Access the Expression Builder.

1.2 *(page 280)*	Use Expression Functions Window.

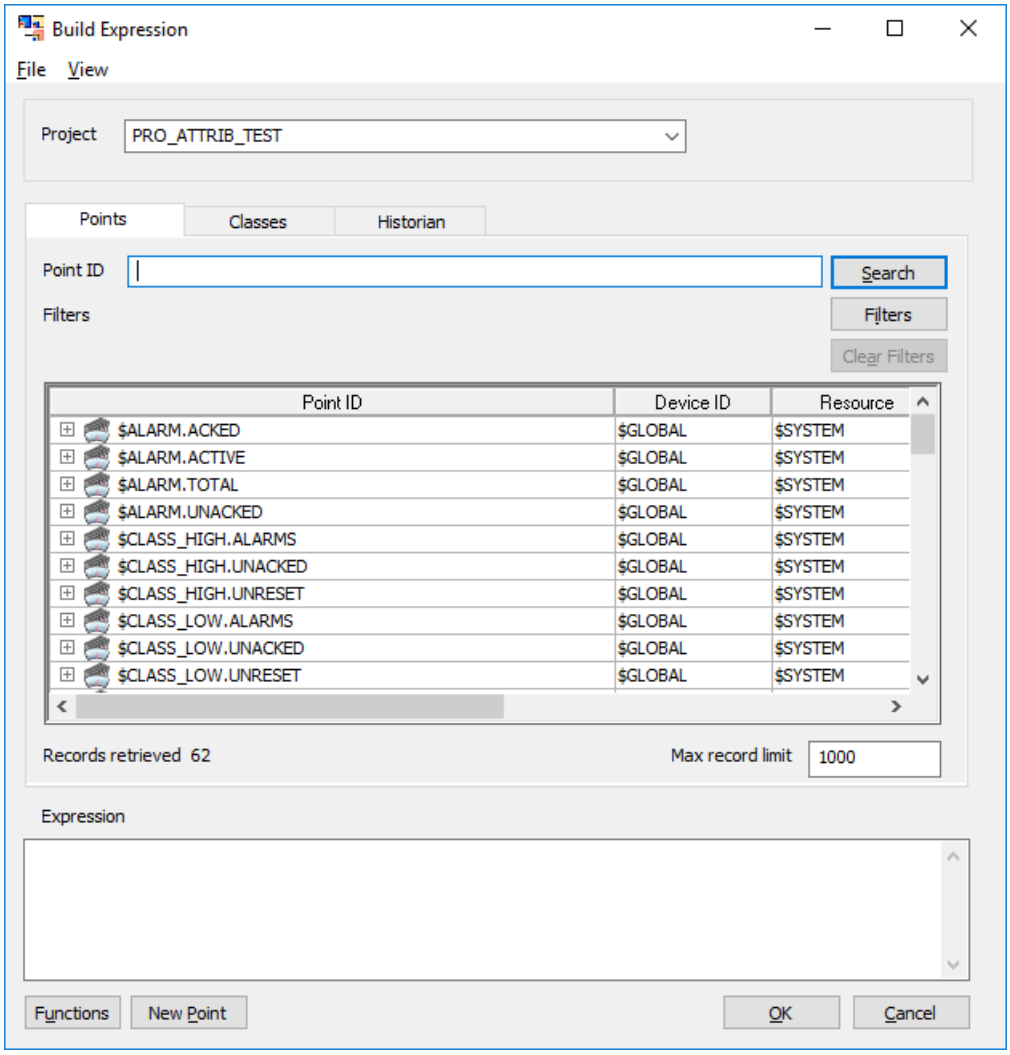
1.1. Access the Expression Builder

1.1. Access the Expression Editor

The Build Expression window provides an area to create an expression using points, class data items, or Historian tags.

To open the Build Expression window, select any of the buttons to the right of the Expression box.

Item	Description
A	Opens the Build Expression window.
B	Displays a menu; select Build Expression to open the Build Expression window.



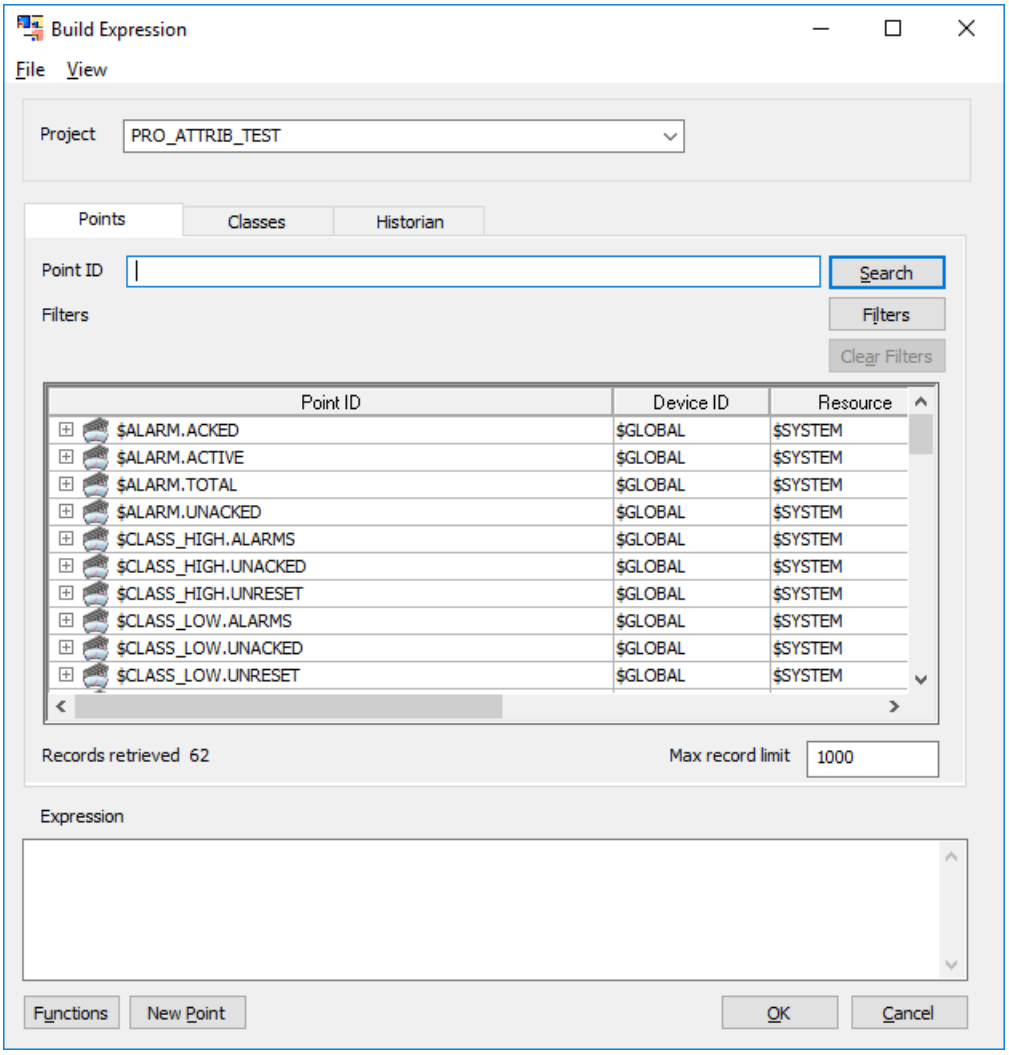
The Build Expression window contains the following boxes and sections:

Box/ Section	Description
Project	Select the project for creating the expression.
Points	See Build an Expression through Points *(page 274)*.
Classes	See Build an Expression through Class Data Items *(page 274)*.
Historian	See Build an Expression through Historian Tags *(page 278)*.
Expression	Displays the expression created by selecting the point IDs, class IDs, Historian tags, or a combination of all of them.
Functions	See Use Function Window *(page 280)*.

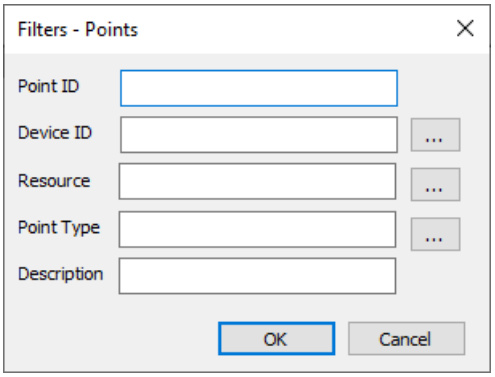
Box/ Section	Description
New Point	Creates a new point to be added to the expression. **Note:** This button is only enabled for the Points section.

1.1.1. Build an Expression through Points

The Points section in the Build Expression window provides you with options to create an expression using points.



The Points section contains the following boxes and buttons:

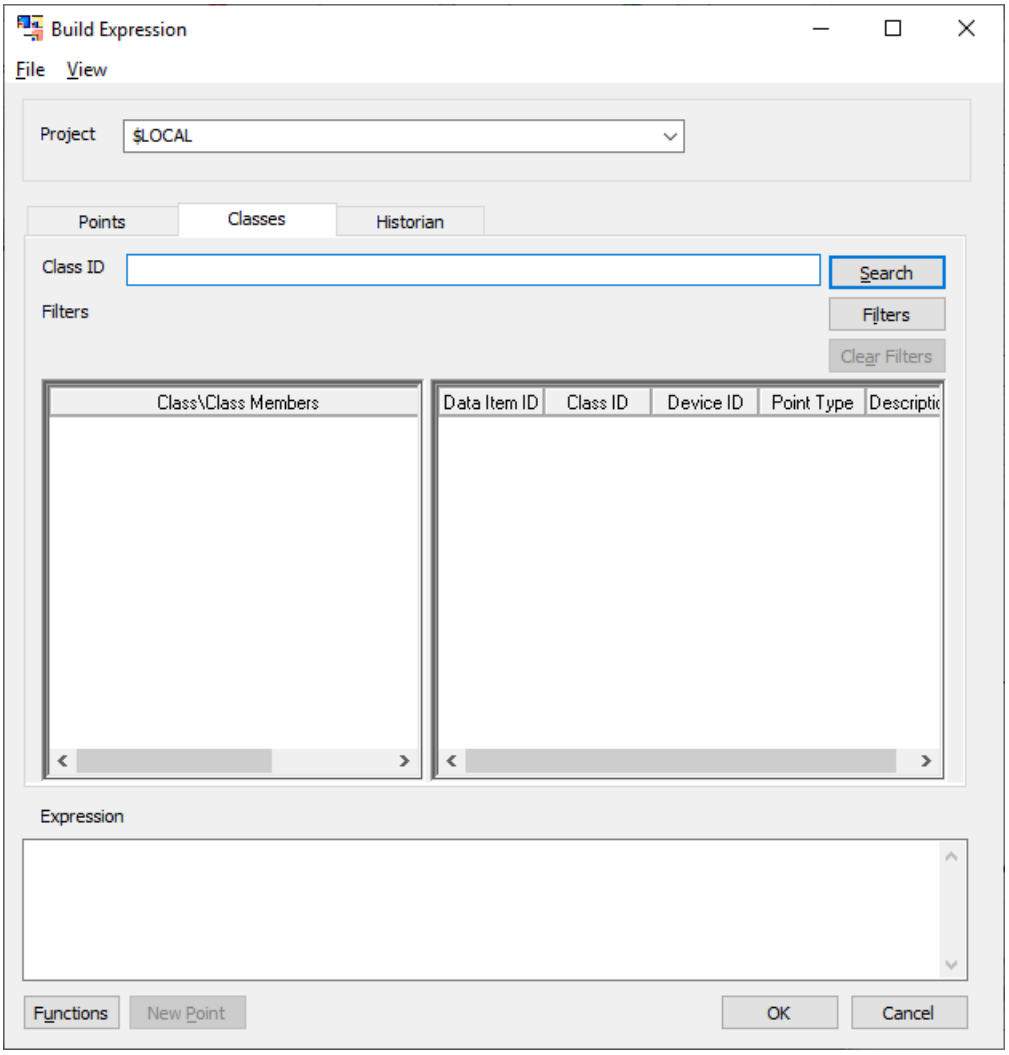
Box/Button	Description
Point ID	The point ID that you want to use to create an expression.
Search	Retrieves and displays records based on the point ID provided.
Filters	 Filters the displayed records based on the following parameters: • Point ID • Device ID • Resource • Point Type • Description
Clear Filters	Removes any filters applied to the displayed records and triggers a new unfiltered search. **Note:** This button is disabled if no filters are applied.
Max record limit	Limits the number of records displayed.

To create an expression using points:

1. In the Build Expression window, in the Points section, in the Point ID box, enter a point ID.
2. Select Search to display all the records associated with a point ID. You can also select Filters to filter the displayed records based on certain parameters.
3. In the list of displayed records, double-click the record that you want to create the expression with. The record is added to the Expression box.
4. Select Functions to add functions to the expression.
5. Select OK. An expression is created using points.

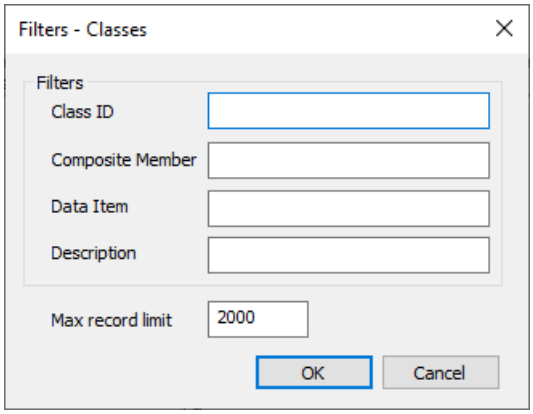
1.1.2. Build an Expression through Class Data Items

The Classes section in the Build Expression window provides you with options to create an expression using class data items.



The Classes section contains the following boxes and buttons:

Box/ Button	Description
Class ID	The class ID that you want to use to create an expression.
Search	Retrieves and displays records based on the class ID provided.

Box/ Button	Description
Filters	 Filters the displayed records based on the following parameters: • Class ID • Composite Member • Data Item • Description • Max Record Limit
Clear Filters	Removes any filters applied to the displayed records. Note: This button is disabled if no filters are applied.

Note: For classes, the Max Record Limit box is located in the Filters window.

To create an expression using class data items:

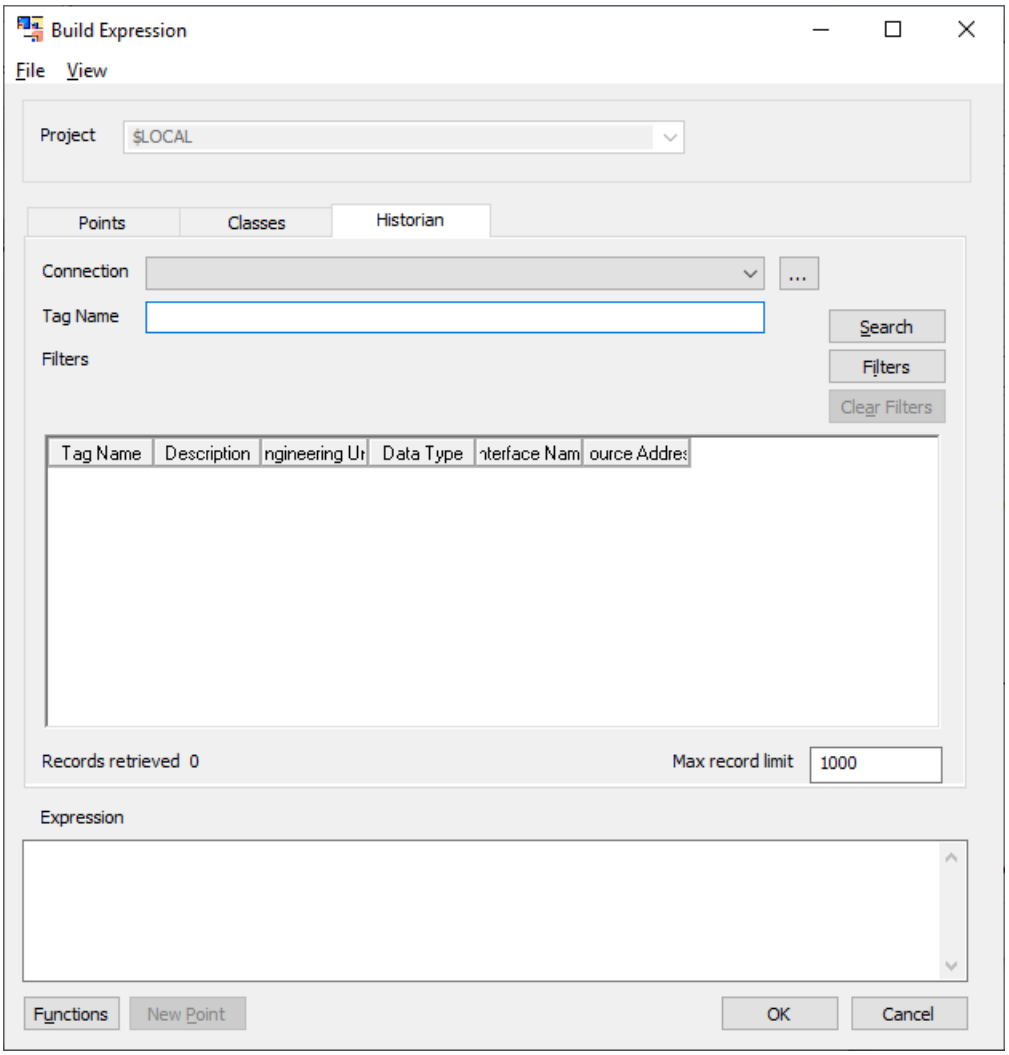
1. In the Build Expression window, in the Classes section, in the Class ID box, enter a class ID.
2. Select Search to display all the classes and class members associated with a class ID. You can also select Filters to filter the displayed classes and class members based on certain parameters.
3. In the list of displayed classes and class members, select a class or class member and the corresponding data item is displayed. Double-click the data item that you want to create the expression with.

   The data item is added to the Expression box.
4. Select Functions to add functions to the expression.
5. Select OK.

An expression is created using class data items.

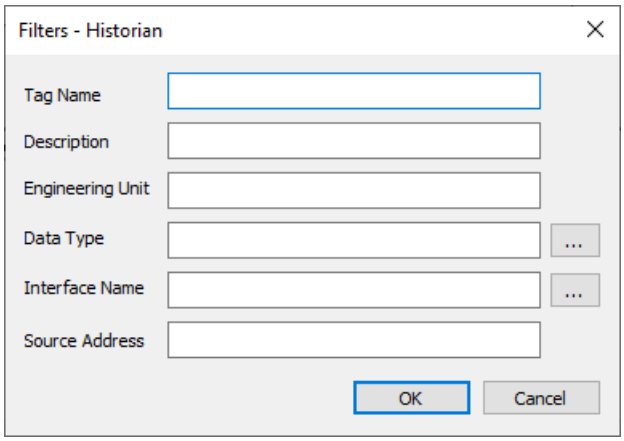
1.1.3. Build an Expression through Historian Tags

The Historian section in the Build Expression window provides you with options to create an expression using Historian tags.



The Historian section contains the following boxes and buttons:

Box/ Button	Description
Connection	Select an existing Historian connection or add a new Historian connection.
Tag Name	The Historian tag that you want to use to create an expression.

Box/ Button	Description
Search	Retrieves and displays records based on the Historian connection established.
Filters	 Filters the displayed records based on the following parameters: • Tag Name • Description • Engineering Unit • Data Type • Interface Name • Source Address
Clear Filters	Removes any filters applied to the displayed records. **Note:** This button is disabled if no filters are applied.
Max record limit	Limits the number of records displayed.

To create an expression using Historian tags:

1. In the Build Expression window, in the Historian section, in the Connection drop-down list box, select an existing Historian connection or add a new Historian connection.
2. After the connection is established, enter the name of the Historian tag that you want to use to create an expression.
3. Select Search to display all the records associated with a Historian tag. You can also select Filters to filter the displayed records based on certain parameters.

4. In the list of displayed records, double-click the record that you want to create the expression with. The record is added to the Expression box.
5. Select Functions to add functions to the expression.

   Note: Historian tags can be configured in the expression only when Historian functions are used.

6. Select OK.

   Result: An expression is created using Historian tags.

*1.2. **Use Expression Functions Window***

The following expression building operations are available in the Expression Functions window.

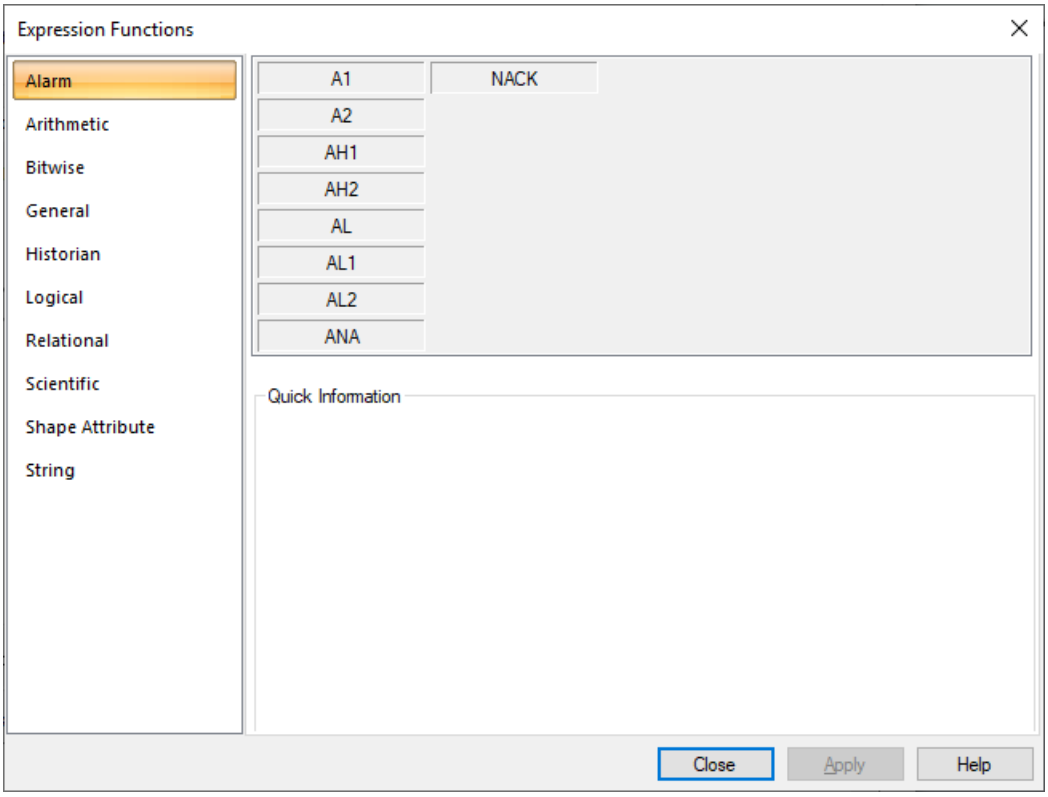
- Alarm
- Arithmetic
- Bitwise
- General
- Historian
- Logical
- Relational
- Scientific
- Shape Attribute
- String

  For all functions,
  - int represents the int64 data type and real represents the double data type.
  - for regions where the decimal separator is a comma (,), include a space after the parameter and before the comma. For example:

    ```
    FormatReal(123,456,1,9,1,0,0) returns an error
    FormatReal(123,456 ,1 ,9 ,1 ,0 ,0) returns 0000123.5
    ```

  - you can enter variables, points, or expressions as the input parameters

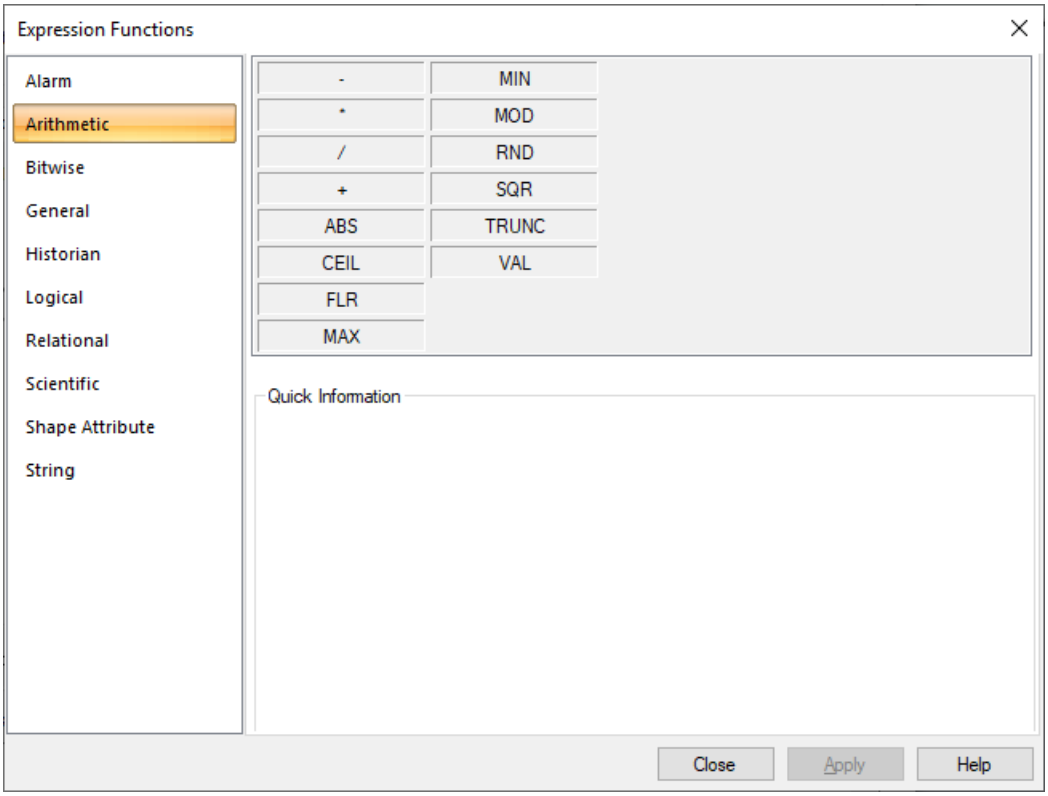
Alarm



Item	Value	Result	
A1	WARNING	Returns	TRUE if the point is in a Warning High or Warning Low state.
		Format	A1(<point id>)
A2	ALARM	Returns	TRUE if the point is in an Alarm High or Alarm Low state.
		Format	A2(<point id>)
AH1	WARNING_HIGH	Returns	TRUE if the point is in a Warning High state.
		Format	AH1(<point id>)
AH2	ALARM_HIGH	Returns	TRUE if the point is in an Alarm High state.
		Format	AH2(<point id>)
AL		Returns	TRUE if the point is in any Alarm or Warning state.
		Format	AL(<point id>)
AL1	WARNING_LOW	Returns	TRUE if the point is in a Warning Low state.
		Format	AL1(<point id>)
AL2	ALARM_LOW	Returns	TRUE if the point is in an Alarm Low state.

Item	Value	Result	
		Format	AL2(<point id>)
ANA	ALARM_NOT_ACKED	Returns	TRUE if the point is in alarm and the alarm has not been acknowledged. Note: The ANA behavior was changed in CIMPLICITY V.6.1. Prior to V.6.1 it behaved as NACK *(page 282)*. The global parameter PTEXP_ANA_EQ_NACK_AND_AL is available to revert ANA behavior to pre-version 6.
		Format	ANA(<point id>) Important: ANA is not supported for Enterprise points.
NACK		Returns	TRUE if the alarm has not been acknowledged, whether or not the point is in an alarm state.
		Format	NACK (<point id>)

Arithmetic

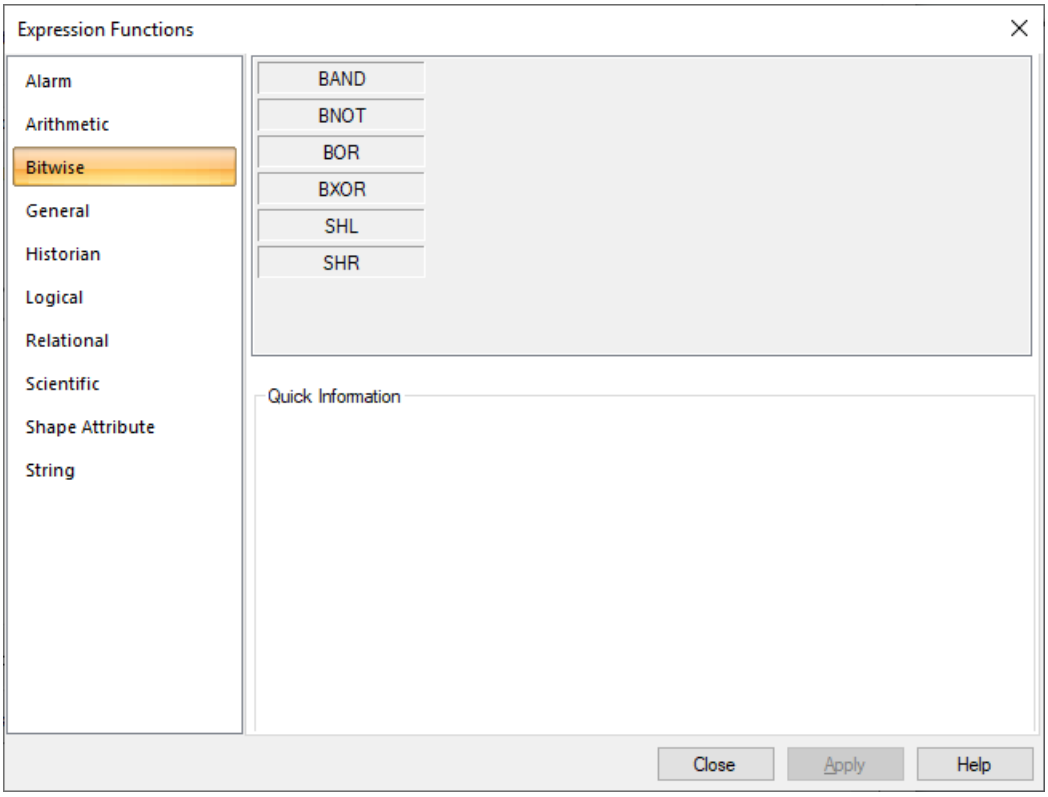


Operation	Result	
-	Returns	Difference between X and Y.

Operation	Result	
	Format	<expr1>-<expr2>
*	Returns	Product of X and Y.
	Format	<expr1>*<expr2>
/	Returns	Quotient of X and Y.
	Format	<expr1>/<expr2>
		Note: The result of dividing two integers in an expression will be an integer. If you want the result to be a floating point, multiply the numerator by 1.0 before dividing. Example POINTA is set to 6. POINTB is set to 4. Results: POINTA/POINTB = 1 (POINTA*1.0)/POINTB = 1.5
+	Returns	Sum of X and Y.
	Format	<expr1>+<expr2>
ABS	Returns	Absolute value of X.
	Format	ABS(expr) Example ABS(-2.6) returns 2.6
CEIL	Returns	Nearest integer greater than or equal to X.
	Format	CEIL (expr) Example CEIL(2.3) returns 3 CEIL(-2.3) returns -2
FLR	Returns	Nearest integer less than or equal to X.
	Format	FLR(expr) Examples FLR(2.6) returns 2 FLR(-2.6) returns -3

Operation	Result	
MAX	Returns	Maximum comparing X and Y.
	Format	<expr1> MAX <expr2> Example 3 MAX 4 returns 4
MIN	Returns	Minimum comparing X and Y.
	Format	<expr1> MIN <expr2> Example 3MIN4 returns 3.
MOD	Returns	Value of X modulo Y.
	Format	<expr1> MOD <expr2> Example 9 MOD 8 returns 1
RND	Returns	Integer nearest to X.
	Format	RND(expr) Example RND(2.6) returns 3 RND(-2.6) returns -3
SQR	Returns	Square root of X.
	Format	SQR(<expr>)
TRUNC	Returns	Value of X with the fractional part removed; the result is its integer value.
	Format	TRUNC(expr) Examples TRUNC(2.6) returns 2 TRUNC(-2.6) returns -2
VAL	Returns	Numeric value of a text string. Note: VAL converts a variable that consists of numbers in text string format to a numeric format that can be included in calculations.
	Format	VAL(expr)

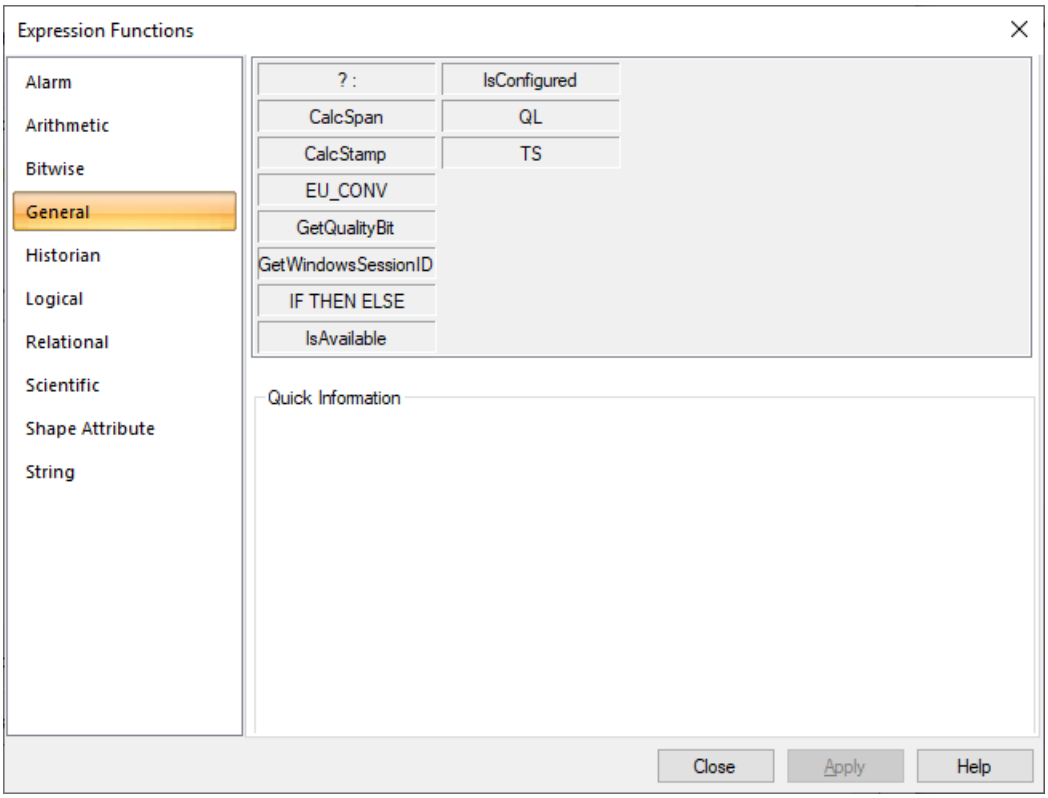
Bitwise



Operation	Result	
BAND	Performs	Bitwise AND of X and Y.
	Format	<expr1>AND<expr2>
BOR	Performs	Bitwise OR of X and Y.
	Format	<expr1>OR<expr2>
BNOT	Performs	Bitwise NOT of X and Y.
	Format	NOT<expr>
BXOR	Performs	Bitwise XOR of X and Y.
	Format	<expr1>XOR<expr2>
SHL	Returns	Value of X shifted left by Y bits.
	Format	<expr1>SHL<expr2> Example 2 SHL 1 returns 4 **Note:** When SHL goes out of range the: Point becomes unavailable. Point Control Pane is starred. Core status log notes that it is unavailable.
SHR	Returns	Value of X shifted right by Y bits.

Operation	Result	
	Format	<expr1>SHR<expr2> Example 2 SHR 1 returns 1 **Note:** When SHR goes out of range the: Point becomes unavailable. Point Control Pane is starred. Core status log notes that it is unavailable.

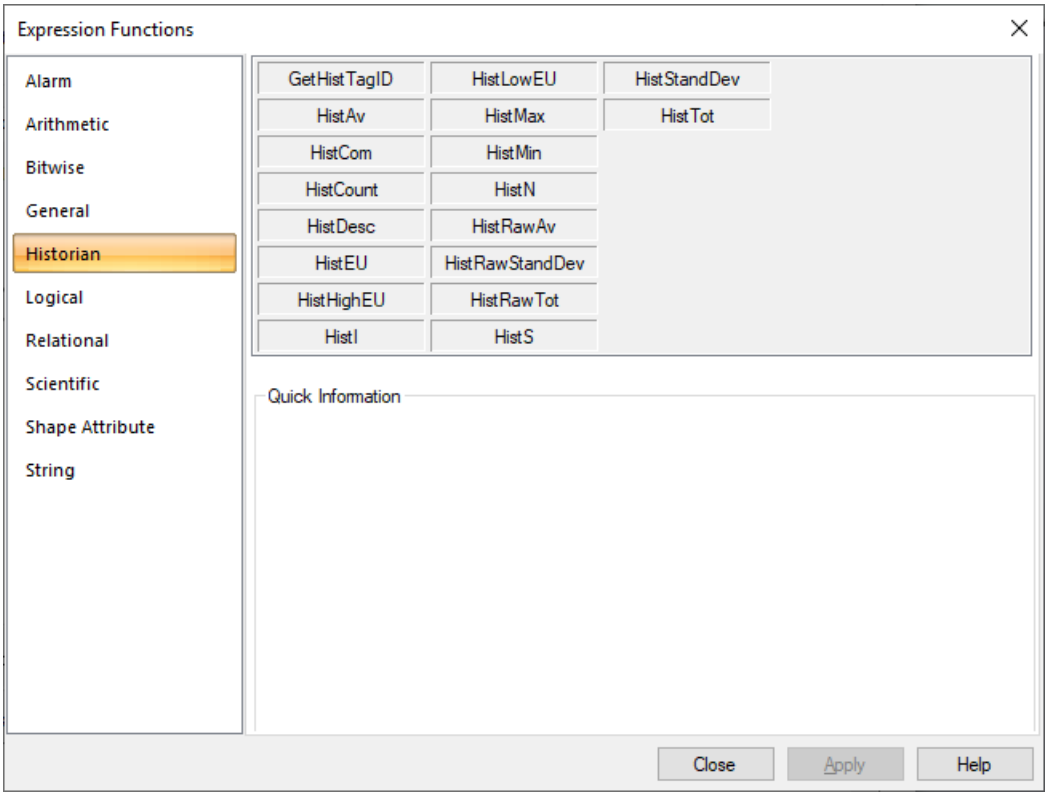
General



Operation	Result			
?:IF/THEN/ELSE	Returns	The value of <True_Expr> if the value of <Cond_Expr> is true. Otherwise, it returns the value of <False_Expr>. The ternary operator (?:) is shorthand for the IF/THEN/ELSE operator.		
		Name	Type	The Input is the:
	Inputs	<Cond_Expr>	Boolean	Expression to be tested.
		<True_Expr>	Any	Value to return if <*Cond_Expr*> is true.
		<False_Expr>	Any	Value to return if <*Cond_Expr*> is false.
	Format	IF <*Cond_Expr*> THEN <*True_Expr*> ELSE <*False_Expr*> or <*Cond_Expr*> ? <*True_Expr*> : <*False_Expr*>		

Operation	Result	
CalcSpan	Returns	An unsigned 64-bit integer containing a duration of time (in decimicroseconds).
	Format	CalcSpan[<days>,<hours>,<minutes>,<seconds>,<fracsec>]
CalcStamp	Returns	An unsigned 64-bit integer containing a time stamp (in decimicroseconds).
	Format	CalcStamp[<year>,<month>,<day>,<hour>,<minute>,<second>,<fracsec>]
EU_CONV	Returns	Converts the default raw value of a device point ID in an expression for virtual points to the engineering units value.
	Format	EU_CONV(Point_ID)
GetQualityBit	Gets	Quality Bit for the specified bit index. Note: GetQualityBit retrieves the quality of the specified expression for a given bit number. The bit number ranges from 0 to 31.
	Format	GetQualityBit(<expression>,<bit index>)
GetWindowsSessionID	Returns	The Windows Session ID of a CimView instance based on the type of session it is running.
	Format	GetWindowsSessionID()
IsAvailable	Returns	If expression is available.
	Format	IsAvailable(<expression>)
IsConfigured	Returns	True if point is configured.
	Format	IsConfigured(P)
QL	Retrieves	The quality (64-bit integer) returned for the wrapped expression.
	Format	QL[<expr>]
TS	Retrieves	Time stamp returned for the wrapped expression.
	Format	TS(<expr>)

Historian



Historian data calls and special functions that retrieve data can be used inside standard expressions.

If you are using the local Historian server, you can enter \\\ instead of the server name.

However, the Historian API only connects to the default server if it is on the local machine. The server name must be entered if the default server is not the local machine.

Example:

```
\\\PROFCIMP.LEVELTSOUTH
```

where:

\\\ indicates the local Historian server, which would be entered as \\<local server>\ if the abbreviation was not used.

ProfCIMP.LEVELTSOUTH is a Historian tag that will be included in the expression.

When or when not to use double-quotes in a Historian expression is as follows.

Use double-quotes when Historian tags are used in the expression.

Note: Because Historian tags will be used most frequently, CIMPLICITY automatically includes double-quotes when you select a Historian function in the Expression Editor.

Do not use double-quotes:

- If you do not use a Historical tag ID, but use either a point ID or GetHistTagID(P).

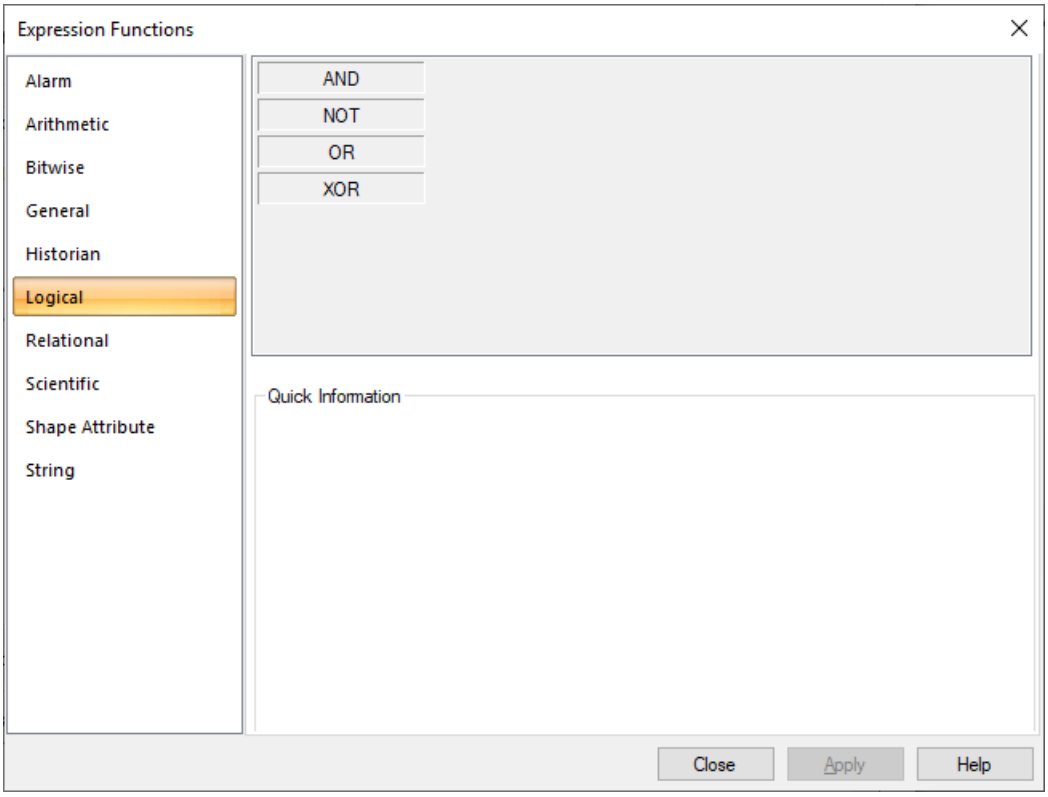
  Note: You must remove the double quotes that are added automatically.

- A global parameter, EXPRESSION_TRACE_LEVEL, is available to trace Historian related problems with connections and expressions

Operation	Result	
GetHistTagID	Returns	A string that may be used to look up a Historian tag from the inputted point symbol.
	Format	GetHistTagID(P) Where P is a CIMPLICITY point ID that has a corresponding tag ID in Historian.
HistAv	Retrieves	Average numeric value of a Historian tab at or before the specified timestamp over the specified previous span.
	Format	HistAv("\\<Historian connection ID>\<TagID>",<Timestamp>,<Span>)
HistCom	Retrieves	Comment (String) of a Historian tag.
	Format	HistCom("\\<Historian connection ID>\<TagID>",<Timestamp>)
HistCount	Retrieves	Count value of a Historian tag at or before the specified timestamp over the specified previous span.
	Format	HistCount("\\<Historian connection ID>\<TagID>",<Timestamp>,<Span>)
HistDesc	Retrieves	Description (string) of a Historian tag.
	Format	HistDesc("\\<Historian connection ID>\<TagID>",<Timestamp>)
HistEU	Retrieves	Engineering Unit (string) of a Historian tag.
	Format	HistEU("\\<Historian connection ID>\<TagID>",<Timestamp>)
HistHightEU	Retrieves	High engineering Unit (string) of a Historian tag.
	Format	HistHightEU("\\<Historian connection ID>\<TagID>",<Timestamp>)
HistI	Retrieves	64-bit integer value of a Historian tag at or before the specified timestamp.
	Format	HistI("\\<Historian connection ID>\<TagID>",<Timestamp>)
HistLowEU	Retrieves	Low engineering Unit (string) of a Historian tag.
	Format	HistLowEU("\\<Historian connection ID>\<TagID>",<Timestamp>)
HistMax	Retrieves	Max numeric value of a Historian tag at or before the specified timestamp over the specified previous span.

Operation	Result	
	Format	HistMax("\\<Historian connection ID>\<TagID>",<Timestamp>,<Span>)
HistMin	Retrieves	Min numeric value of a Historian tag at or before the specified timestamp over the specified previous span.
	Format	HistMin("\\<Historian connection ID>\<TagID>",<Timestamp>,<Span>)
HistN	Retrieves	Raw numeric value (double float) of a Historian tag at or before the specified timestamp.
	Important:	Historian batches and writes values. HISTN reports the value whenever it sees it. Due to the batch writing, an update may not be there at the time of the request from the HISTN function. Example $LOCAL.DATETIME)INTERVAL is set to 10 causing $LOCAL.DATETIME_VARUPDATE to update 10 times per second. The HistN expression updates when it sees a reported value from Historian, which is one time every two seconds. Several $LOCAL.DATETIME_VARUPDATE updates can be missed. You can avoid seeing this behavior if you change the request by deferring for a specified amount of time, e.g. 5 seconds. Example HistN("\\\Simulation00001", $LOCAL.DATETIME_VARUPDATE - CalcSpan(0,0,0,5,0))
	Format	HistN("\\<Historian connection ID>\<TagID>",<Timestamp>)
HistRawAv	Retrieves	Raw average numeric value of a Historian tag at or before the specified timestamp over the specified previous span.
	Format	HistRawAv("\\<Historian connection ID>\<TagID>",<Timestamp>,<Span>)
HistRawStandDev	Retrieves	Standard deviation raw numeric value of a Historian tag at or before the specified timestamp over the specified previous span.
	Format	HistRawStandDev("\\<Historian connection ID>\<TagID>",<Timestamp>,<Span>)
HistRawTot	Retrieves	Total numeric value of a Historian tag at or before the specified timestamp over the specified previous span.
	Format	HistRawTot("\\<Historian connection ID>\<TagID>",<Timestamp>,<Span>)
HistS	Retrieves	String value of a Historian tag at or before the specified timestamp.
	Format	HistS("\\<Historian connection ID>\<TagID>",<Timestamp>)
HistStandDev	Retrieves	Standard deviation numeric value of a Historian tag at or before the specified timestamp over the specified previous span.
	Format	HistStandDev("\\<Historian connection ID>\<TagID>",<Timestamp>,<Span>)
HistTot	Retrieves	Raw total numeric value of a Historian tag at or before the specified timestamp over the specified previous span.
	Format	HistTot("\\<Historian connection ID>\<TagID>",<Timestamp>,<Span>)

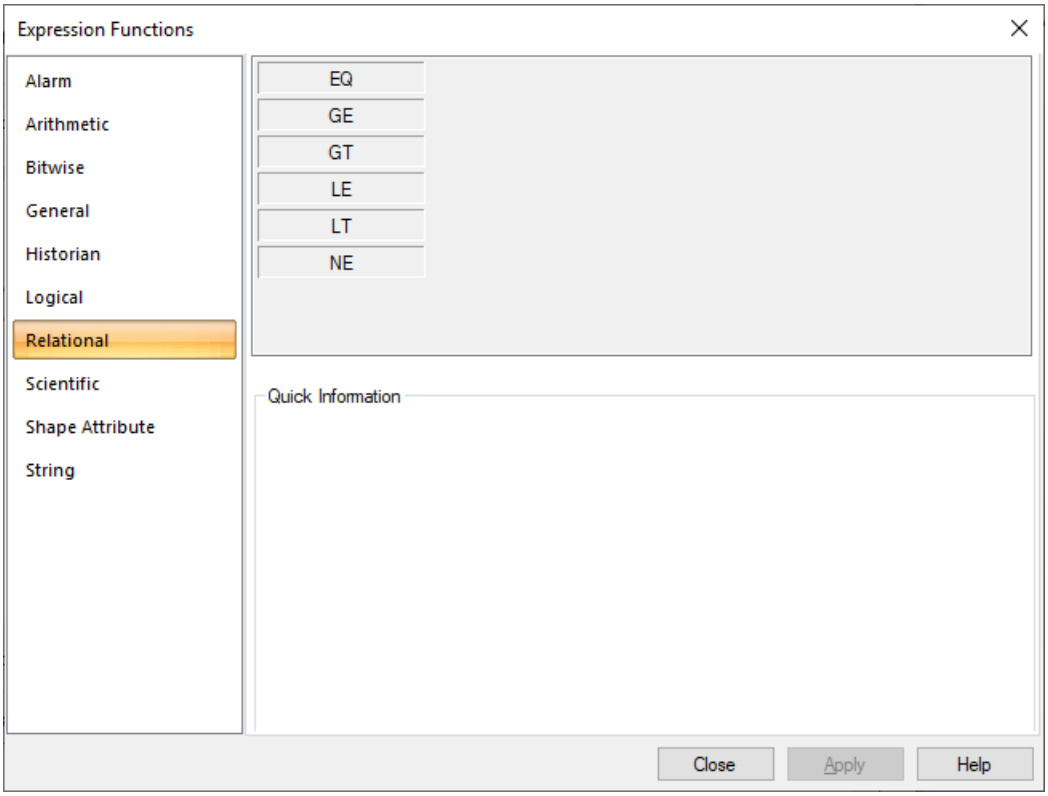
Logical



You can use Boolean, integer or floating point numbers for logical operations. If an expression has a non-zero value, it is TRUE; if the value is zero, it is FALSE.

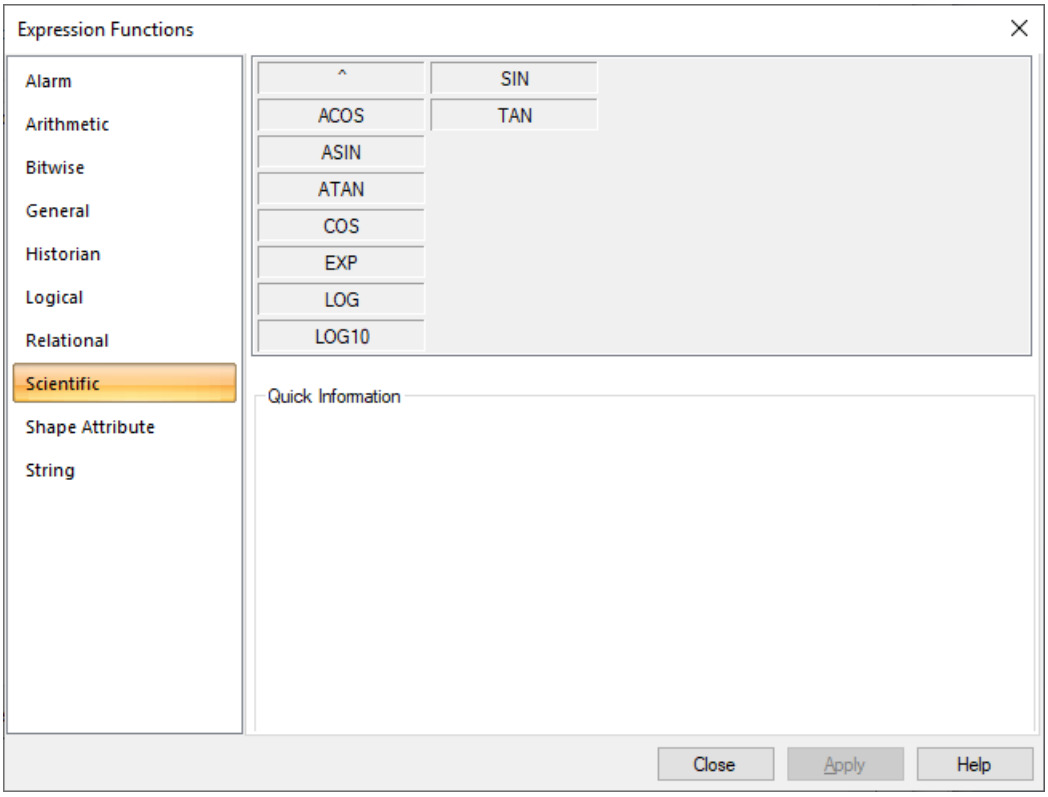
Operation	Expression	Result
AND	Performs	Logical AND of X and Y.
	Format	<expr1>AND<expr2>
NOT	Performs	Logical NOT of X.
	Format	NOT<expr>
OR	Performs	Logical OR of X and Y.
	Format	<expr1>OR<expr2>
XOR	Performs	Logical XOR of X and Y.
	Format	<expr1>XOR<expr2>

Relational



Operation	Expression	Result
EQ	Returns	True if X is equal to Y.
	Format	<expr1>EQ<expr2>
GE	Returns	True if X is greater than or equal to Y.
	Format	<expr1>GE<expr2>
GT	Returns	True if X is greater than Y.
	Format	True if X is greater than Y.
LE	Returns	True if X is less than or equal to Y.
	Format	<expr1>LE<expr2>
LT	Returns	True if X is less than or below Y.
	Format	<expr1>LT<expr2>
NE	Returns	True if X is not equal to Y.
	Format	<expr1>NE<expr2>

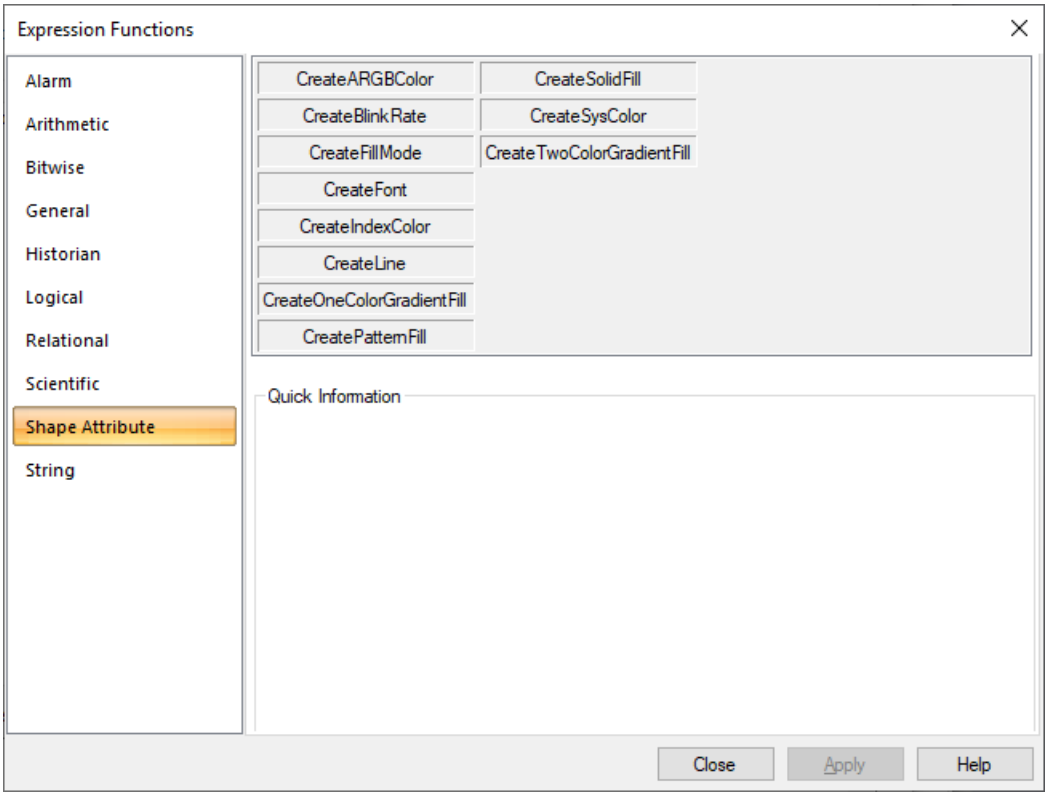
Scientific



Operation	Result	
^	Returns	Value of X raised to the power of Y.
	Format	<expr1>^<expr2>
ACOS	Returns	Arc cosine (angle in radians) of X radians.
	Format	ACOS(<expr>)
ASIN	Returns	Arc sine (angle in radians) of X radians.
	Format	ASIN(<expr>)
ATAN	Returns	Arc tangent (angle in radians) of X radians.
	Format	ATAN(<expr>)
COS	Returns	Cosine (angle in radians) of X radians.
	Format	COS(<expr>)
EXP	Returns	Value of e raised to the power of X. Note: EXP is the exponential (ex) value of an expression where X is the expression.

Operation	Result	
	Format	EXP(<expr>)
LOG	Returns	Natural logarithm (base e) of X.
	Format	LOG(<expr>)
LOG10	Returns	Base 10 logarithm of X.
	Format	LOG10(<expr>)
SIN	Returns	Sine (angle in radians) of X radians.
	Format	SIN(<expr>)
TAN	Returns	Tangent (angle in radians) of X radians.
	Format	TAN(<expr>)

Shape Attribute



Operation	Result			
CreateARGBColor	Returns	A color value string based on the transparency value and RGB color codes provided.		
		Name	Type	The Input is the:

Operation	Result			
	Inputs	<COLOR_ALPHA>	int	Value ranging from 0 to 255 denoting the level of transparency.
		<COLOR_RED>	int	Value ranging from 0 to 255 denoting the shade of red.
		<COLOR_GREEN>	int	Value ranging from 0 to 255 denoting the shade of green.
		<COLOR_BLUE>	int	Value ranging from 0 to 255 denoting the shade of blue.
	Format	CreateARGBColor(<COLOR_ALPHA>,<COLOR_RED>,<COLOR_GREEN>,<COLOR_BLUE> Example CreateARGBColor(255,135,255,132) returns {"ARGB": [255,135,255,132]} This function can be used to provide the color attribute to other shape attribute functions that take a color string as one of their input parameters.		
CreateBlinkRate	Returns	An integer that represents the time interval in which an object will blink at run time.		
		Name	Type	The Input is the:
	Inputs	<BlinkRate>	int	Value ranging from 1 to 5 representing the time interval in which an object will blink. The values are: 1 - 200 ms 2 - 400 ms 3 - 600 ms 4 - 800 ms 5 - 1000 ms
	Format	CreateBlinkRate(BlinkRate) Example CreateBlinkRate(BLINKRATEPT) If BLINKRATEPT is set to 2, the object blinks every 400 ms. If BLINKRATEPT is set to a value that is beyond the valid range of 1 to 5, the object does not blink.		
CreateFillMode	Returns	A formatted string containing information on the fill direction and bipolar properties of an object.		
		Name	Type	The Input is the:

Operation	Result			
	Inputs	<FillDirection>	int	Value ranging from 1 to 4 denoting the direction of the fill. The values are: 1 - Fill from bottom 2 - Fill from top 3 - Fill from left 4 - Fill from right
		<Bipolar>	Bool/ int	Value denoting the polarity of the fill. This is a flag whose value is either 0 or 1. 0 denotes a normal polarity of the fill. 1 denotes a bipolar fill.
	Format	CreateFillMode(FillDirection,Bipolar) Example CreateFillMode(FILLDIRPT,BIPOLARPT) If FILLDIRPT is set to 1, the fill direction is Fill from Bottom. If FILLDIRPT is set to a value that is beyond the valid range of 1 to 4, the fill direction is set to the values entered manually in the Rotation/Fill section of the Properties - Object window..		
CreateFont	Returns	A font based on the parameters provided. Note: This is applicable to Text and TextButton objects only.		
		Name	Type	The Input is the:
	Inputs	<FontName>	string	Value representing the name of the font.
		<FontStyle>	int	Value ranging from 1 to 4 denoting the style of the font. The values are: 1 - Regular 2 - Italic 3 - Bold 4 - Bold italic

Operation	Result			
		<FontSize in TWIPS>	int	Value representing the size of the font in TWIPS.
		<FontEffects>	int	Value ranging from 1 to 4 denoting the effect that can be applied to the font. The values are: 1 - None 2 - Strikethrough 3 - Underline 4 - Strikethrough and underline
	Format	CreateFont(FontName,FontStyle,FontSize in TWIPS,FontEffects) Example CreateFont("Arial",0,200,1) returns {"FontName": "Arial", "FontStyle": 0,"FontSize" : 200, "FontEffects": 1}}"		
CreateIndexColor	Returns	A formatted string representing a color based on the color indices supported by CIMPLICITY.		
		Name	Type	The Input is the:
	Inputs	<IndexValue>	int	Value denoting the index number of the color. Note: For the value of the index, access the RGB Editor.
	Format	CreateIndexColor(IndexValue) Example CreateIndexColor(5) returns {"IndexColor": 5}		
CreateLine	Returns	A formatted string representing a line based on the parameters provided.		
		Name	Type	The Input is the:

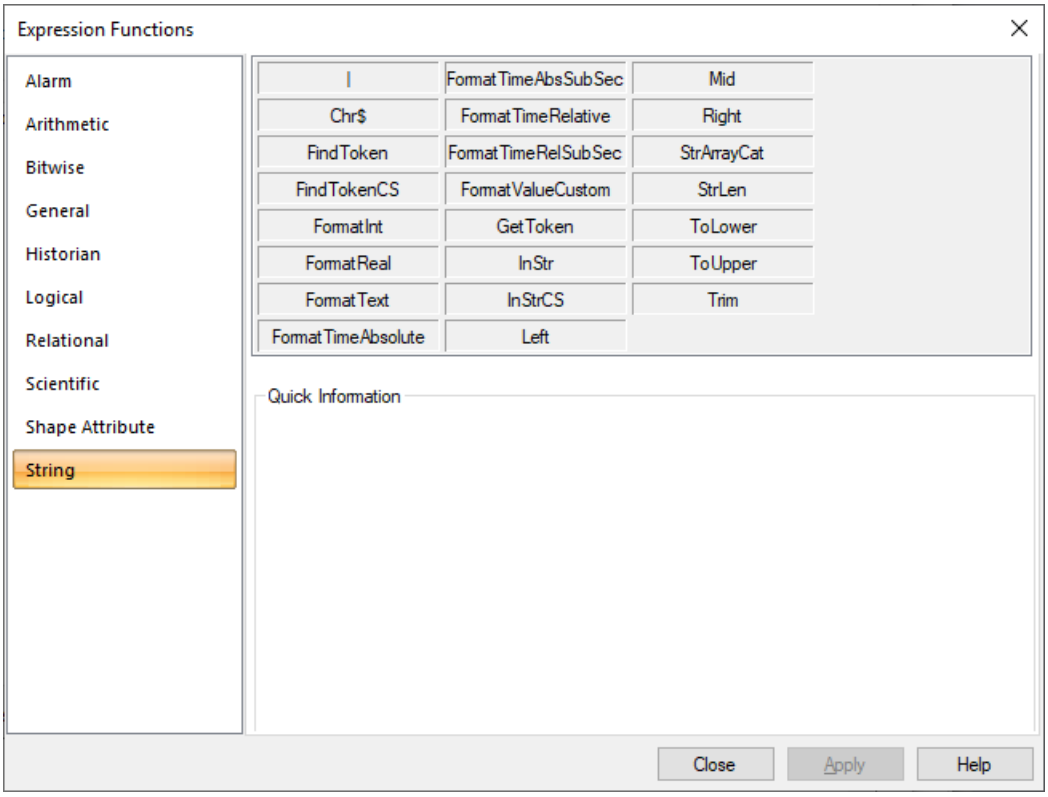
Operation	Result			
	Inputs	<LineStyle>	int	Value ranging from 0 to 11, which represents the style of the line. The values are: 0 - Mixed value 1 - Solid 2 - Long dash 3 - Long dash - space - dot 4 - Dash 5 - Long dash - dot - dot 6 - Round dot 7 - Dot - dot - space 8 - Long dash - long space - dot 9 - Dash - dot - space 10 - Long dash - dot 11 - None
		<Color>	string	Formatted string representing the color of the line. This string can be created by the CreateARGBColor, CreateIndexColor, or CreateSysColor functions.
		<Width>	float	Value specified in twips.
	Format	CreateLine(LineStyle,Color,Width) Example CreateLine(LineStylePT,CreateARGBColor(255,100,100,100),11) If the value of LineStylePT is changed from 1 to 2, the line style will change from solid to long dash. If no value is provided to LineStylePT, the line style is set to the value entered manually in the Colors section of the Properties - Object window.		
CreateOneColorGradientFill	Returns	A formatted string representing a gradient fill type that uses one color to create a gradient effect.		
		Name	Type	The Input is the:

Operation	Result			
	Inputs	<ShadingStyles>	int	Value ranging from 1 to 7 representing the style of the shade. The values are: 1 - Horizontal 2 - Vertical 3 - Diagonally upwards 4 - Diagonally downwards 5 - From a corner 6 - To a corner 7 - From the center
		<ShadeVariant>	int	Value ranging from 1 to 4 representing the variant of the shade.
		<Color>	string	Formatted string that represents the color of the gradient. This string can be created by the CreateARGBColor, CreateIndexColor, or CreateSysColor functions.
		<Brightness>	int	Value ranging from 1 to 20 representing the brightness of the color gradient.
	Format	CreateOneColorGradientFill(ShadingStyles,ShadeVariant,Color,Brightness) Example CreateOneColorGradientFill(1,1, CreateIndexColor(5),15) returns {"Fill": {"FillType": 2,"ShadingStyles": 1,"ShadeVariant": 1 ,"Color1": {"IndexColor": 5} ,"Brightness": 15}}		
CreatePatternFill	Returns	A formatted string representing a pattern fill type that uses two colors to create different fill patterns.		
		Name	Type	The Input is the:
	Inputs	<PatternType>	int	Value ranging from 1 to 24 representing the type of pattern to be used. Note: For details of the index to be used, access the palette.
		<Color>	string	Formatted string that represents the primary color of the pattern. This string can be created by the CreateARGBColor, CreateIndexColor, or CreateSysColor functions.

Operation	Result			
		<PatternColor>	string	Formatted string that represents the secondary color of the pattern. This string can be created by the CreateARGBColor, CreateIndexColor, or CreateSysColor functions.
	Format	CreatePatternFill(PatternType,Color,PatternColor) Example CreatePatternFill(1,CreateIndexColor(5), CreateARGBColor(255,100,132,135)) returns {"Fill": {"FillType": 2,"PatternType":1,"Color": {"IndexColor": 5},"PatternColor": {"ARGB": [255,100,132,135]}}}		
CreateSolidFill	Returns	A formatted string representing a solid fill type.		
		Name	Type	The Input is the:
	Inputs	<Color>	string	Formatted string that represents a color. This string can be created by the CreateARGBColor, CreateIndexColor, or CreateSysColor functions.
	Format	CreateSolidFill(Color) Example CreateSolidFill(CreateARGBColor(255,100,132,135)) returns {"Fill": {"FillType": 1, "Color": {"ARGB": [255,100,132,135]}}}		
CreateSysColor	Returns	A formatted string representing a color based on the system index colors supported by the operating system. Note: Depending on the colors chosen in the Display properties, system colors may vary between computers. Use these colors only if you want your CIMPLICITY screen to follow the desktop color scheme.		
		Name	Type	The Input is the:
	Inputs	<SysColorIndex>	int	Zero-based value denoting a valid system color index.
	Format	CreateSysColor(SysColorIndex) Example CreateSysColor(9) returns {"SystemColor": 9}		
CreateTwoColorGradientFill	Returns	A formatted string representing a gradient fill type that uses two colors to create a gradient effect.		
		Name	Type	The Input is the:

Operation	Result			
	Inputs	<ShadingStyles>	int	Value ranging from 1 to 7 representing the style of the shade. The values are: 1 - Horizontal 2 - Vertical 3 - Diagonally upwards 4 - Diagonally downwards 5 - From a corner 6 - To a corner 7 - From the center
		<ShadeVariant>	int	Value ranging from 1 to 4 representing the variant of the shade.
		<Color1>	string	Formatted string that can be generated using any of the color functions. This string can be created by the CreateARGBColor, CreateIndexColor, or CreateSysColor functions.
		<Color2>	string	Formatted string that can be generated using any of the color functions. This string can be created by the CreateARGBColor, CreateIndexColor, or CreateSysColor functions.
	Format	CreateTwoColorGradientFill(ShadingStyles,ShadeVariant,Color1,Color2) Example CreateTwoColorGradientFill(1,1, CreateIndexColor(5), CreateIndexColor(2)) returns {"Fill": {"FillType": 4,"ShadingStyles": 1,"ShadeVariant": 1,"Color1": {"IndexColor": 5} , "Color2": {"IndexColor": 2} }}		

String



Operation	Result			
|	Will be substituted by double quotes in an expression.			
Chr$	Returns	A one-character UTF-16 encoded UNICODE string. The integer can be any valid 16-bit UNICODE code point from the Basic Multilingual Plane.		
		Name	Type	The Input is the:
	Inputs	<Code_Point>	integer	Integer value of the UNICODE code point.
	Format	Chr$(<Code_Point>) Example The expression Chr$(12461)+Chr$(12486)+Chr$(12451) will display the Japanese text ###. Note: You do not need to use the Chr$ function to display the text. You can enter 16-bit UNICODE characters directly in CimEdit.		

Operation	Result			
FindToken	Returns	A one-based index for the given token. If the token indicated is not found, the function returns 0.		
		Comparison	Case Insensitive	
		Name	Type	The Input is the:
	Inputs	<Source_String>	string	Collection of tokens separated by one of the characters in the Token_Separators input.
		<Token_Separators>	string	List of all the valid characters (for example, ",.") used to separate tokens. Any character in the Token_Separators string will separate tokens. If the separators are ",.", then both comma and period will separate tokens. Empty tokens are ignored. Note: If a space is used in the token separators, it cannot be the last character in the separators.
		<Token_To_Find>	string	A token in the Source_String.
	Format	FindToken(<Source_String>, <Token_Separators>, <Token_To_Find>) Example FindToken("Green.Red,Black.White",",.","black") returns 3. FindToken("a,b.c,,d..e,.f", ",.", <token>) returns 1, 2, 3, 4, 5, and 6 for tokens A, B, C, D, E, and F.		
FindTokenCS	Returns	A one-based index for the given token. If the token indicated is not found, the function returns 0.		
		Comparison	Case Sensitive	
		Name	Type	The Input is the:
	Inputs	<Source_String>	string	Collection of tokens separated by one of the characters in the Token_Separators input.
		<Token_Separators>	string	List of all the valid characters (for example, ",.") used to separate tokens. Any character in the Token_Separators string will separate tokens. If the separators are ",.", then both comma and period will separate tokens. Empty tokens are ignored. Note: If a space is used in the token separators, it cannot be the last character in the separators.

Operation	Result			
		<Token_To_Find>	string	A token in the Source_String.
	Format	FindTokenCS(<Source_String>, <Token_Separators>, <Token_To_Find>) Example FindTokenCS("Green.Red,Black.White",",.","black") returns 0. FindTokenCS("Green.Red,Black.White",",.","Black") returns 3. FindTokenCS("a,b.c,,d..e,.f", ",.", <token>) returns 1, 2, 3, 4, 5, and 6 for tokens a, b, c, d, e, and f.		
FormatInt	Returns	An integer prefixed with trailing zeroes based on the parameters provided.		
		Name	Type	The Input is the:
	Inputs	<Value>	int/ real/ string	Value to be formatted.
		<Width>	int	If zero_filled is 1, Width is the length of the output after the trailing zeroes are prefixed to Value. If zero_filled is 0, ignore Width.
		<zero_filled>	BOOL/ int	Value that indicates if the trailing zeroes will be prefixed to Value.
		<UseDigitGrouping>	BOOL/ int	Value that indicates if digit grouping will be applied to the formatted value.
	Format	FormatInt(<Value>, <Width>, <is zero_filled>, <UseDigitGrouping>) Example FormatInt(123,6,1,0) returns 000123. FormatInt(123456789,6,0,1) returns 123,456,789.		
FormatReal	Returns	A real value prefixed with trailing zeroes based on the parameters provided.		
		Name	Type	The Input is the:
	Inputs	<Value>	int/ real/ string	Value to be formatted.
		<Precision>	int	Number of digits in the value after the decimal to be retained.
		<Width>	int	If zero_filled is 1, Width is the length of the output after the trailing zeroes are prefixed to Value. If zero_filled is 0, ignore Width.

<table>
<tr><th>Operation</th><th colspan="4">Result</th></tr>
<tr><td></td><td></td><td><zero_filled></td><td>BOOL/ int</td><td>Value that indicates if the trailing zeroes will be prefixed to Value.</td></tr>
<tr><td></td><td></td><td><scientific></td><td>BOOL/ int</td><td>Value that determines the scientific notation in which the Value is displayed.</td></tr>
<tr><td></td><td></td><td><UseDigitGrouping></td><td>BOOL/ int</td><td>Value that indicates if digit grouping will be applied to the formatted value.</td></tr>
<tr><td></td><td></td><td colspan="3">FormatReal(<RealValue>, <Precision>, <Width>, <is zero_filled>, <isscientific>, <UseDigitGrouping>)
Example
FormatReal(123.456,1,9,1,0,0) returns 0000123.5
FormatReal(123.456,1,9,1,1,0) returns 001.2e+002
FormatReal(123456789.456,1,9,1,0,1) returns 123,456,789.456</td></tr>
<tr><td>FormatText</td><td>Returns</td><td colspan="3">A string that is truncated based on the parameters provided.</td></tr>
<tr><td></td><td></td><td>Name</td><td>Type</td><td>The Input is the:</td></tr>
<tr><td></td><td>Inputs</td><td><Value></td><td>int/ real/ string</td><td>Value to be truncated.</td></tr>
<tr><td></td><td></td><td><Max Width></td><td>int</td><td>The number of characters to be displayed.</td></tr>
<tr><td></td><td>Format</td><td colspan="3">FormatText(<Value>, <MaxWidth>)
Example
FormatText("GEDigital",3) returns GED.</td></tr>
<tr><td>FormatTimeAbsolute</td><td>Returns</td><td colspan="3">A string that represents a specific number of seconds converted to a format of time calculated since 00:00 UTC on January 1, 1970.</td></tr>
<tr><td></td><td></td><td>Name</td><td>Type</td><td>The Input is the:</td></tr>
<tr><td></td><td>Inputs</td><td><Value></td><td>int/ real/ string</td><td>The number of seconds to be converted.</td></tr>
<tr><td></td><td></td><td><Format Type></td><td>int</td><td>The number assigned to a format based on which the Value will be converted.
<table><tr><th>Number</th><th>Format</th></tr><tr><td>1</td><td>h:mm tt
Note: tt represents a.m. or p.m.</td></tr><tr><td>2</td><td>HH:mm</td></tr></table></td></tr>
</table>

Operation	Result				
				3	HH:mm:ss
				4	M/d/yy
				5	MM/dd/yy
				6	MM/dd/yyyy HH:mm:ss
				7	YYYY:MM:DD:HH:MM:SS Note: If you choose this format, you can only display values after 1995:01:01:00:00:00 (that is, January 1, 1995 at midnight). Values before this date are displayed as ----:--:--:--:--:--.
				8	yyyy:MM:dd:hh:mm:ss
	Format	FormatTimeAbsolute(<Value>, <Format Type>) Example FormatTimeAbsolute(123456789,4) returns 11/30/73.			
FormatTimeAbsSubSec	Returns	A string that represents a specific number of sub seconds converted to a format of time calculated since 00:00 UTC on January 1, 1970.			
		Name	Type	The Input is the:	
	Inputs	<Value>	int/ real/ string	The number of sub seconds to be converted.	
		<Format Type>	int	The number assigned to a format based on which the Value will be converted.	
				Number	Format
				1	h:mm tt Note: tt represents a.m. or p.m.
				2	HH:mm
				3	HH:mm:ss
				4	M/d/yy
				5	MM/dd/yy
				6	MM/dd/yyyy HH:mm:ss
				7	YYYY:MM:DD:HH:MM:SS
				8	yyyy:MM:dd:hh:mm:ss

Operation	Result				
				9	MM-dd-yyyy HH:mm:ss.TTT Note: TTT represents milliseconds
				10	MM-dd-yyyy HH:mm:ss.TTTTTT Note: TTTTTT represents microseconds
	Format	FormatTimeAbsSubSec(<Value>, <Format Type>) Example FormatTimeAbsolute(123456789,4) returns 01/01/70.			
FormatTimeRelative	Returns	A string that represents a specific number of seconds converted to a format of time.			
		Name	Type	The Input is the:	
	Inputs	<Value>	int/ real/ string	The number of seconds to be converted.	
		<Format Type>	int	The number assigned to a format based on which the Value will be converted.	
				Number	Format
				1	DD:HH:MM:SS
				2	HH.tenths
				3	HH:MM:SS
				4	MM.Tenths
				5	MM:SS
	Format	FormatTimeRelative(<Value>, <Format Type>) Example FormatTimeRelative(123456789,4) returns 2057613:09.			
FormatTimeRelSubSec	Returns	A string that represents a specific number of sub seconds converted to a format of time.			
		Name	Type	The Input is the:	
	Inputs	<Value>	int/ real/ string	The number of sub seconds to be converted.	

<table>
<tr><th>Operation</th><th colspan="4">Result</th></tr>
<tr><td></td><td></td><td><Format Type></td><td>int</td><td>The number assigned to a format based on which the Value will be converted.<table><tr><td>Number</td><td>Format</td></tr><tr><td>1</td><td>DD:HH:MM:SS</td></tr><tr><td>2</td><td>HH.tenths</td></tr><tr><td>3</td><td>HH:MM:SS</td></tr><tr><td>4</td><td>MM.Tenths</td></tr><tr><td>5</td><td>MM:SS</td></tr><tr><td>6</td><td>DD:HH:MM:SS:TTT

Note: TTT represents milliseconds</td></tr><tr><td>7</td><td>DD:HH:MM:SS:TTTTTT

Note:TTTTTT represents microseconds</td></tr></table></td></tr>
<tr><td></td><td>Format</td><td colspan="3">FormatTimeRelSubSec(<Value>, <Format Type>)
Example
FormatTimeRelSubSec(123456789,4) returns 2057613:09.</td></tr>
<tr><td>FormatValueCustom</td><td>Returns</td><td colspan="3">A customized string that is formatted based on the specified format.</td></tr>
<tr><td></td><td></td><td>Name</td><td>Type</td><td>The Input is the:</td></tr>
<tr><td></td><td>Inputs</td><td><Value></td><td>int/ real/ string</td><td>Value to be formatted.</td></tr>
<tr><td></td><td></td><td><Format_Specifier></td><td>string</td><td>String representing the format to which the Value will be converted.<table><tr><td>String</td><td>Custom format</td></tr><tr><td>%d</td><td>Signed decimal integer</td></tr><tr><td>%u</td><td>Unsigned decimal integer</td></tr><tr><td>%f</td><td>Decimal floating point number</td></tr><tr><td>%o</td><td>Octal representation</td></tr><tr><td>%x</td><td>Lowercase unsigned hexadecimal integer</td></tr><tr><td>%X</td><td>Uppercase unsigned hexadecimal integer</td></tr></table></td></tr>
</table>

Operation	Result				
				%e	Lowercase scientific notation (mantissa/ exponent)
				%E	Uppercase scientific notation (mantissa/ exponent)
				%c	Equivalent ASCII character
				%%	Inserts a percent (%) symbol after the Value
	Format	FormatValueCustom(<Value>, <Format_Specifier>) Example FormatValueCustom(123.45,"%d") returns 123 FormatValueCustom(123,"%x") returns 7b			
GetToken	Returns	A substring that is the zero-based token element in the string.			
		Name	Type	The Input is the:	
	Inputs	<Source_String>	string	String containing a collection of tokens separated by one of the characters in the Token_Separators input.	
		<Token_Separators>	string	String containing all the valid characters used to separate tokens (for example,","). Any character in the Token_Separators string will separate tokens. If the separators are ",.", then both comma and period will separate tokens. Empty tokens are ignored. Note: If a space is used in the token separators, it cannot be the last character in the separators.	
		<TokenIndex0Based>	int	Zero-based index of the token to be retrieved from the Source_String.	
	Format	GetToken(<source_string>, <Token_Separators>, <TokenIndex0Based>) Example GetToken("Green.Red,Black.White",",.",2) returns Black. GetToken("a,b.c,,d..e,.f", ",.", <index>) returns a, b, c, d, e, and f for indexes 0, 1, 2, 3, 4, and 5.			
InStr	Returns	An integer that is the one-based character position of the first occurrence of the substring that is searched for (after the start search position) in the source string.			
		Comparison	Case Insensitive		
		Name	Type	The Input is the:	
	Inputs	<Start_Pos>	integer	One-based character position where the search begins.	

Operation	Result			
		<Source_String>	string	String to be searched.
		<Search_SubStr>	string	String to search for.
	Format	InStr(<Start_Pos>, <Source_String>, <Search_SubStr>) Example InStr(1, "abba", "a") returns 1 InStr(1, "abba", "A") returns 1 InStr(2, "abba", "a") returns 4		
InStrCS	Returns	An integer that is the one-based character position of the first occurrence of the substring that is searched for (after the start search position) in the source string.		
		Comparison	Case Sensitive	
	Inputs	Name	Type	The Input is the:
		<Start_Pos>	integer	One-based character position where the search begins.
		<Source_String>	string	String to be searched.
		<Search_SubStr>	string	String to search for.
	Format	InStrCS(<Start_Pos>, <Source_String>, <Search_SubStr>) Example InStrCS(1, "abba", "a") returns 1 InStrCS(1, "abba", "A") returns 0 InStrCS(2, "abba", "a") returns 4		
Left	Returns	Extracts the first <Count> characters from the source strings and returns a copy of the extracted substring.		
	Inputs	Name	Type	The Input is the:
		<Source_String>	string	String to copy from.
		<Count>	integer	Number of characters to copy.
	Format	Left(<Source_String>, <Count>)		
Mid	Returns	The substring that is the specified number (<Count>) of characters long and starts at the one-based character position in the source string.		
	Inputs	Name	Type	The Input is the:
		<Source_String>	string	String from which the substring is being extracted.
		<StartPos>	int	One based character position where the search begins.
		<Count>	int	Number of characters to count.

Operation	Result			
	Format	Mid(<Source_String>, <StartPos>, <Count>)		
Right	Returns	Extracts the last <Count> characters from the source string and returns a copy of the extracted substring.		
	Inputs	Name	Type	The Input is the:
		<Source_String>	string	String from which the substring is being extracted.
		<Count>	integer	Number of characters to copy.
	Format	Right(<Source_String>,<Count>)		
StrArrayCat	Returns	The concatenation of the elements of the string array point into a single string.		
		Name	Type	The Input is the:
	Inputs	<Point_ID>	string array point	String array point which is a concatenation of the elements of a string array point.
	Format	StrArrayCat(<Point_ID>) Example If MYSTR is a string array point, where MYSTR[0] = "abc", MYSTR [1] = "de", and MYSTR [2] = "fgh", StrArrayCat(MYSTR) will return abcdefgh.		
StrLen	Returns	The length of the string.		
		Name	Type	The Input is the:
	Inputs	<Source_String>	string	String whose length will be returned.
	Format	StrLen(<Source_String>) Note: If a string literal is used as an argument, the trailing spaces (if any) will not be included in the string length. Example StrLen(“ABC “) will return 3, not 4.		
ToLower	Returns	The string in lowercase.		
	Inputs	Name	Type	The Input is the:
		<Source_String>	string	String that will be lowercase.
	Format	ToLower(<Source_String>)		
ToUpper	Returns	The string in uppercase.		
	Inputs	Name	Type	The Input is the:
		<Source_String>	string	String that will be uppercase.
	Format	ToUpper(<Source_String>)		

Operation	Result			
Trim	Returns	A string with the leading and trailing white space removed.		
	Inputs	Name	Type	The Input is the:
		<string>	string	String being trimmed.
	Format	Trim(<string>)		

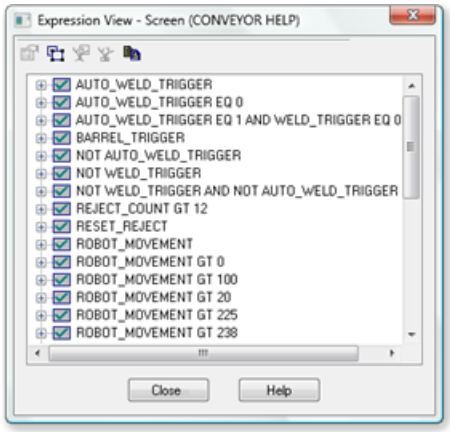
2. Expression View in CimEdit/CimView

2. Expression View in CimEdit/CimView

CimEdit provides you with two invaluable View tools, Expression View and the Point View *(page 229)* .

Two major benefits are that they:

- Eliminate guessing about what Point IDs or expressions you are using on your screen and with what items they are associated.
- Speed up editing time by letting you access any item's Properties dialog box with one easy click.



2.1 *(page 313)*	Open the Expression View window.
2.2 *(page 315)*	Review the Expression View tree.
2.3 *(page 316)*	Display associated expressions and objects.
2.4 *(page 322)*	Make Changes through Expression View

2.5 (page 327)	Copy expressions to the Windows clipboard.

2.1. Open the Expression View Window

When the Expression View window opens, the information it displays depends on what you select before you open it. Once opened, you can expand or collapse your view at any time.

Tip: To select an object or group that is embedded in a group or frame container before Expression View is opened, open the group or frame container before you make your selection. When Expression View opens, its contents will reflect exactly what is selected. Once opened Expression View enables you to review the expression for any object in the group that has an expression.

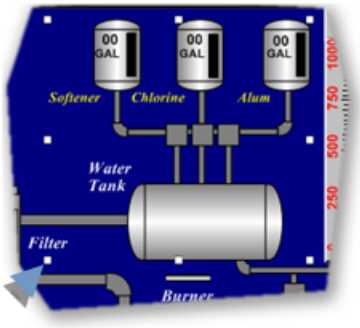
1 (page 313)	Select what will be reviewed.
2 (page 314)	Open Expression View for your selection.

Select what will be reviewed.

Do one of the following.

- Review the expressions associated with only one object or group.

Select an object or group on the CimEdit screen before you open the Expression View window.



This will bring you directly to the information you are looking for. You can expand your view when the Expression View window is open.

- Display the expressions for the entire CimEdit screen.

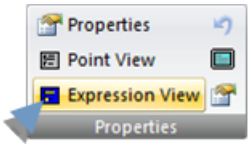
Place the cursor anywhere on the screen.

Open Expression View for your selection.

Ribbon bar

CimEdit only

Click the Expression View button in the Home>Properties group on the CimEdit Ribbon bar.

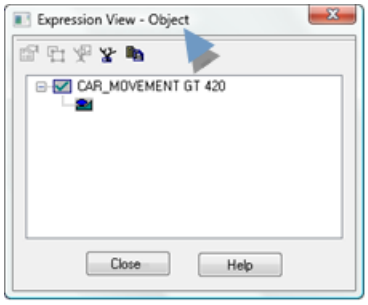
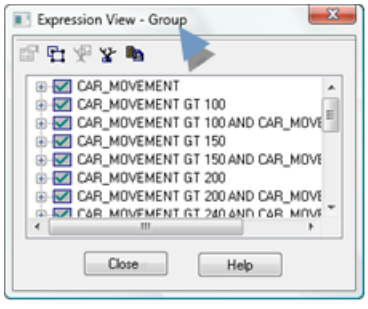


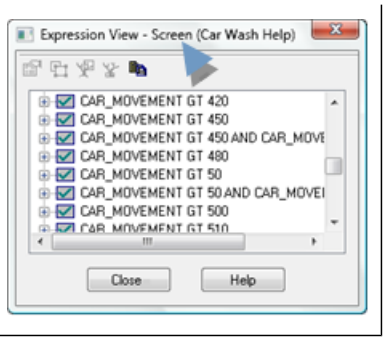
Popup menu

CimEdit and CimView

1. Click the right-mouse button.
2. Select Expression View on the Popup menu.

Result: When the Expression View window opens, the expressions you see are being used by the object, group or screen you selected.

Object	
Group	

Screen	

Note: The window is re-sizable. The Title bar always reflects the object type and name of the current base object.

2.2. Review the Expression View Tree

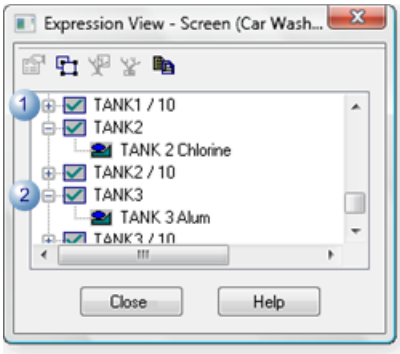
When you open *(page 313)* the Expression View window you see a list of all the expressions associated with the object you selected.

- Expand/collapse the tree.
- Icons in the Expression View tree.

Expand/Collapse the Tree

Choose one of the following.

Method 1

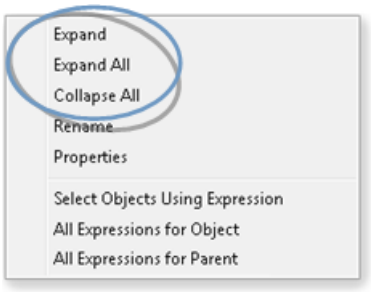


1	Click ⊞ to expand the view.
2	Click ⊟ to collapse the view.

Method 2

1. Select any item in the Expression View list.
2. Click the right-mouse button.

3. Select from the available expand/collapse options on the Expression View Popup menu.



The tree expands or collapses based on your selections.

Icons in the Expression View Tree: Expression View displays the following icons to represent objects on the CimEdit/CimView screen.

	Screen
	Group
	Line object
	Shape object
	Button
	Text

2.3. Display Associated Expressions and Objects

2.3. Display Associated Expressions and Objects

Expression View provides the tools to quickly find and review the:

2.3.1 *(page 316)*	Expressions associated with a selected object.
2.3.2 *(page 321)*	Objects associated with a selected expression.

2.3.1. Display Expressions Associated with an Object

2.3.1. Display Expressions Associated with an Object

When the Expression View window is open, you can display expressions for any object or group on the CimEdit/CimView screen.

- Display expressions for the object or parent: Overview

- Display expressions for the object or parent: Detailed

Display Expressions for the Object or Parent: Overview

- All expressions used by the highlighted object.
- All expressions used by the parent of the base object.

All expressions used by the highlighted object

Do one of the following to view all expressions that a highlighted object uses.

- Click the **All Expressions for object** button to display the selected item as the base object.
- Right-click the object; select **All Expressions for Object** on the Popup menu.

All expressions used by the parent of the base object.

Do one of the following.

- Click the **All Expressions for Parent** button to display the parent as the new base object.
- Right-click the base object; select **All Expressions for Parent** on the Popup menu.

Display Expressions for the Object or Parent: Detailed

2.3.1.1 *(page 318)*	Base object/parent levels defined.

2.3.1.2 *(page 320)*	All expressions by a highlighted object.

2.3.1.1. Base Object/Parent Levels Defined

By incorporating base object/parent functionality, in addition to its expand/collapse features, the Expression View window displays the information that you are looking for.

The two categories of objects are as follows.

Object	Description
Base	Object that is selected. The first object selected as the base can range from one object to the entire screen.
Parent	Item that is one level above the base object. The parent can range from a group of two items to the entire screen.

Select a parent of a base object as follows.

1 *(page 318)*	Select a single object as the base object.
2 *(page 319)*	Select the parent object of the base object.
3 *(page 319)*	Continue selecting "up" until the screen becomes the base object.

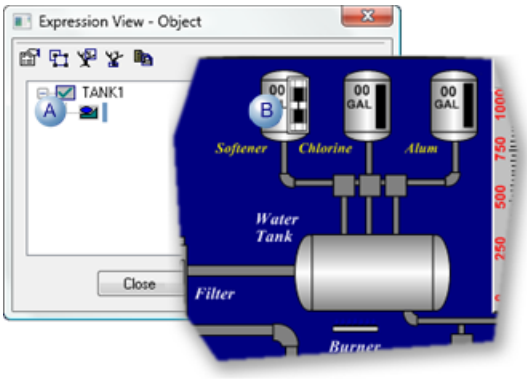
Select a single object as the base object.

1. Select an object.

Note: The object can be in an open group *(page 144)* .

1. Open the Expression View window.
2. Select the object in the window.

CimEdit does the following.



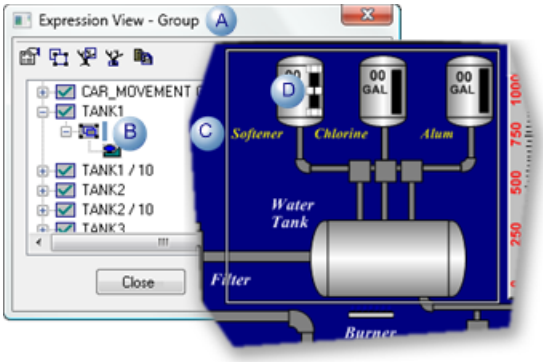
A	The Expression View window opens at the Object level.
B	A frame will surround the object if it is selected in the Expression View window.

Select the parent object of the base object.

1. Click to make the object's parent object the base object.

CimEdit does the following.

The next parent object up becomes the base object.



A	The Expression View window changes to the Group level.
B	Any object or group can be selected in the Expression View window.
C	A frame will surround the object or group that is selected in Expression View. **Note:** if the object extended to the group's borders the frame will be the same size as the frame surrounding the object.
D	The originally selected object may still be selected in addition to the selection through Expression View.

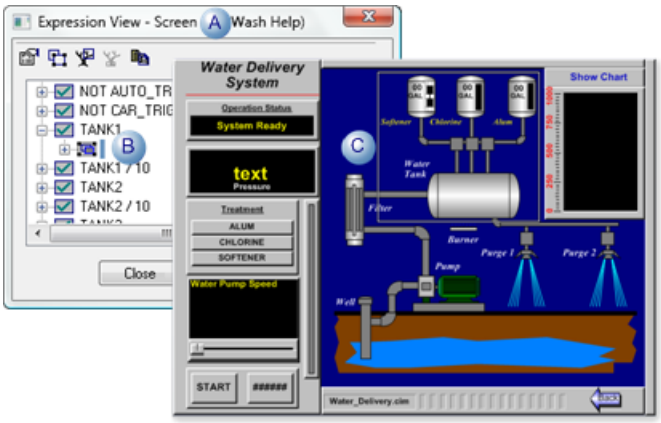
Continue selecting "up" until the screen becomes the base object.

1. Select the new base object.

2. Click if you want the next parent object to become the base object.

CimEdit continues to select the next parent object.

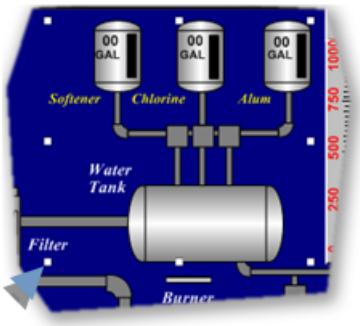
When is clicked and the screen is the parent of the base object CimEdit does the following.



A	The Expression View window changes to the Screen level.
B	Any object or group can be selected in the Expression View window.
C	A frame will surround any object or group on the CimEdit screen that is selected in the Expression View window.

2.3.1.2. All Expressions for a Highlighted Object

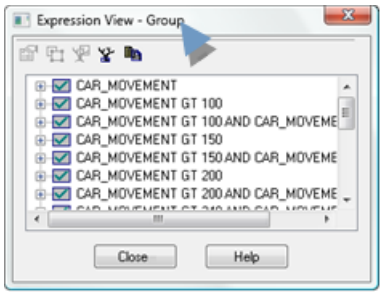
1. Select the group that includes the object to be reviewed.



Tip: Select the screen if you are not sure what group to select.

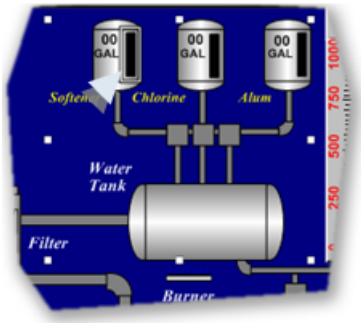
2. Open *(page 313)* the Expression View window.

Expression View displays the expressions for the selected object/group.

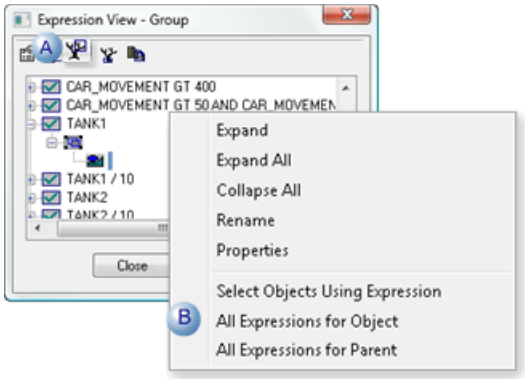


3. Expand expressions in the list to find and select the object or group to be reviewed.

   Note: Items that you select in the list are surrounded by a frame on the CimEdit screen.



4. Do one of the following.



A	Click the All Expressions for Object button on the Expression View toolbar.
B	a. Right-click the selected object. b. Select **All Expressions for Object** on the Popup menu.

5. Right-click the selected object.

6. Select **All Expressions for Object** on the Popup menu.

2.3.2. Display Objects used by a Selected Expression

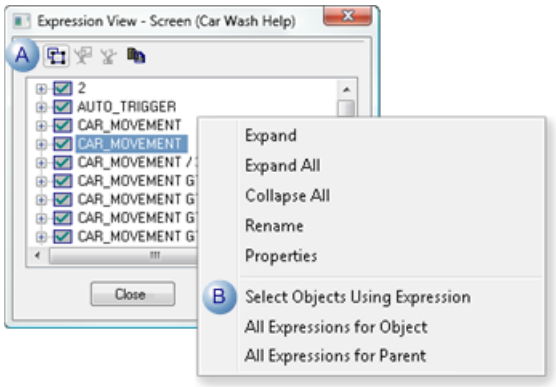
(CimEdit only) Expression View lists all the objects that use an expression and frames a selected object on the screen.

1. Open *(page 313)* the Expression View window for the screen or group you want to review.
2. Select the expression to be reviewed.
3. Do either of the following.

   Select all objects that are used by the expression.

   Select an expression.

   Do one of the following.



A	Click the Select Objects Using Expression button.
B	a. Right-click the expression. b. Choose Select Objects Using Expression on the Popup menu.

4. Right-click the expression.
5. Choose Select Objects Using Expression on the Popup menu.

2.4. Modify Expressions

2.4. Make Changes through Expression View

2.4.1 *(page 323)*	Open an object's Properties dialog box through Expression View.
2.4.2 *(page 324)*	Replace an expression through Expression View.

2.4.3 *(page 326)*	Rename an object through Expression View.

Note: In CimView, you can look at the item's properties, but you cannot edit them.

2.4.1. Open an Object's Properties Dialog Box through Expression View

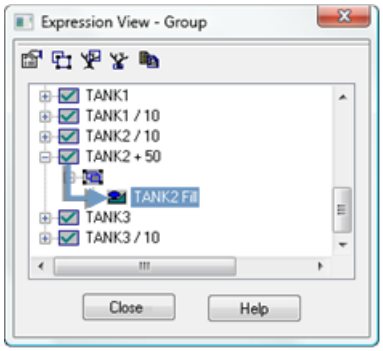
1. Select an object in the Expression View window.

   Note:

   - The object that is at the bottom of an expanded tree contains the selected expression. However, an object above the bottom may also include that expression. You can check for duplications and other possible inconsistencies by copying *(page 327)* the expression list to the clipboard and reviewing it in an application such as Excel.
   - You can open the Properties dialog box for any object in the tree.

   Example

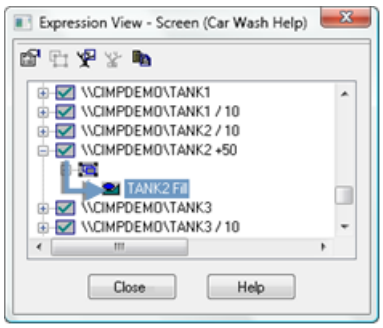
   A TANK2 fill object has an expression TANK2 + 50



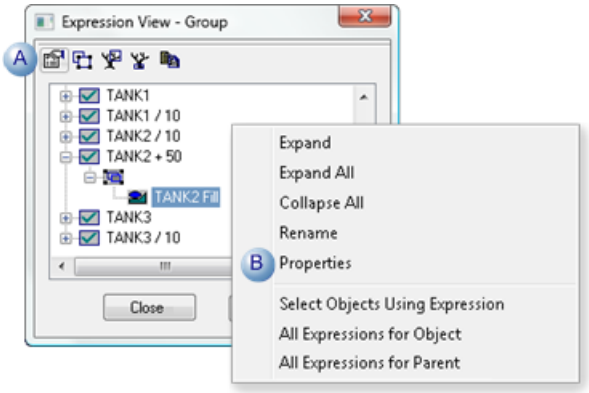
   Note: The expression in CimView would be `\\`<project name>`\`TANK2+50

   Where

   <project name> is the project that has the expression.

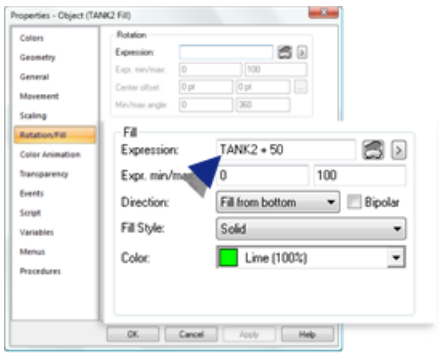
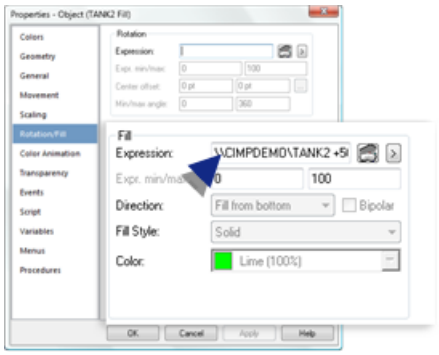


2. Do one of the following.



A	Click the Properties button on the Expression View menu bar.
B	a. Right-click the object. b. Select Properties on the Popup menu.

The Properties dialog box opens for the selected object.

CimEdit	Any object properties, including the expressions, can be configured. 
CimView	The expressions can be reviewed, but are read-only. 

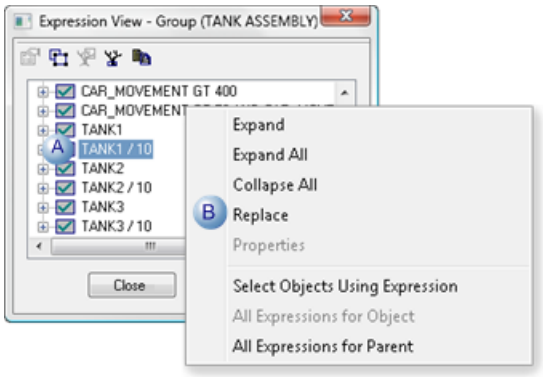
2.4.2. Replace an Expression through Expression View

(CimEdit only)

An expression can be changed directly in the Expression View window. The selected expression will change for all objects that are listed for that expression in the open Expression View window.

1. Select an expression in the Expression View window.

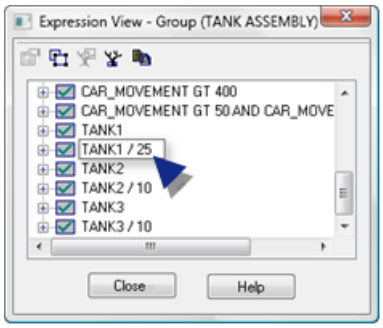
2. Do the following.



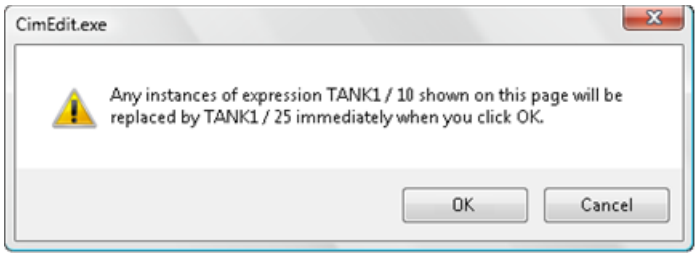
A	Right-click the expression in the Expression View window.
B	Select Replace on the Popup menu.

   The expression can now be replaced.

3. Enter the new or modified expression.



   A message opens cautioning you that the expression will change for all the objects listed in the Point View window.



4. Click OK.

Note: Clicking Cancel cancels the replacement.

The expression is changed for the objects listed in the Expression View window.

Note: To replace the expression for every associated object on the screen, simply open Expression View at the screen level.

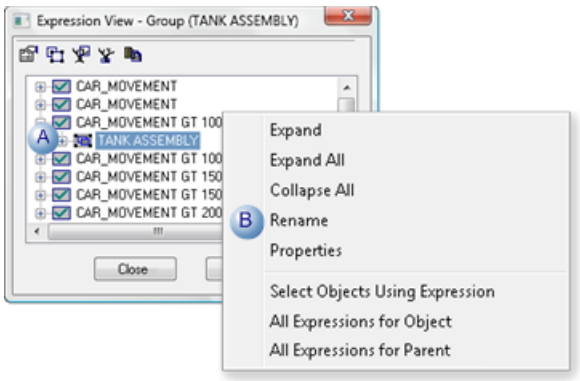
Important: If you replace an expression, it is highly recommended that you review all the expressions on the screen to insure its consistency.

2.4.3. Rename an Object through Expression View

(CimEdit only) An object can easily be named or renamed in the Expression View window.

This can be very useful when an expression has been assigned to an object within a group that might not have been named in its Properties dialog box. The name makes the object much easier to identify.

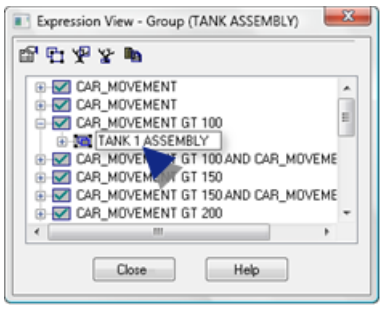
1. Do the following.



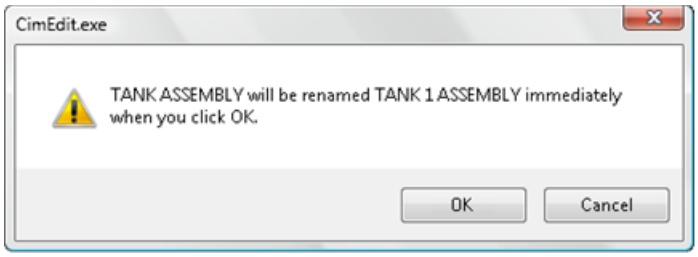
A	Right-click an object in the Expression View window.
B	Select Rename on the Popup menu.

A name can be entered or edited for the object.

2. Enter the new name.



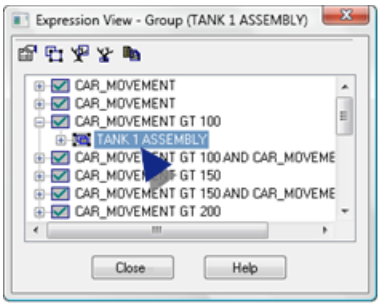
A message opens cautioning you that the name will change.



3. Click OK.

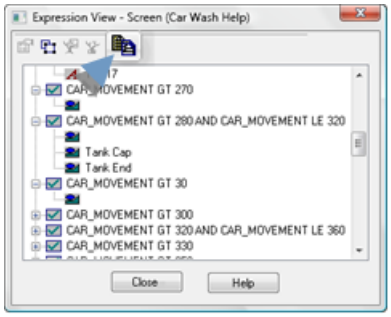
Note: Clicking Cancel cancels changing the name.

The new name is listed in the expression View window and on the General *(page 62)* tab in the object's Properties dialog box.

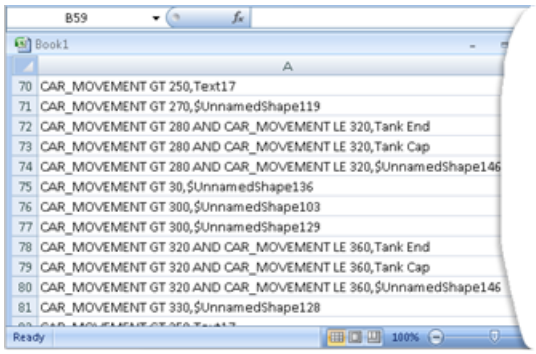


2.5. Copy Expressions to the Windows Clipboard

1. Open Expression View for a selected object, group or the screen.
2. Click the Paste to Clipboard button .



3. Paste the data into a text editing application, e.g. Excel.



Each item in the expression list provides the information displayed in Expression View.

Note: You can paste lists from several base objects into Excel; sort the list and review where expressions may be missing, duplicated or inconsistent.

Linked Objects

Linked Objects

Linked objects enable you to change a single linked source object that automatically updates every linked object in the project.

- Basic principles about linking objects.
- Linked object configuration.
- Find a linked object source.
- Protected link source objects.

Basic Principles about Linking Objects

Basic Principles about Linking Objects

Common uses for linked objects include

- Common navigation tools that appear on several screens e.g. title bars.
- SmartObjects.
- Common footers.
- When component designers need to create several objects that have the:
- Exact same features (e.g. a title bar) they create (or edit) one object with the intent that it will be the source object for several linked objects.

- Same internal functionality (for example, gauges), they can create a template that has the capability of being used as the source for a linked object.

Note: The template can use public variables whose values can be specific to each screen. This obviously provides the designer with extensive development flexibility.

- When screen designers use linked objects provided by the component designer, they can create and configure application screens much more quickly than they can without the linked object. Instead of having to regularly place updated copies of the same object on different screens, all they have to do is create as many links as they want. CIMPLICITY automatically updates all the links when any change is made to the linked source object.

It is recommended that you take time to plan

- Scripts and Procedures
- Need to be executed for all the linked objects, write them for the linked source object.
- Are unique to one or more linked containers, write them for the linked containers only.
- Variables

Public and private variables impact

Variable	Reason	Link Container Variable Values
Public	Different links need to have different values.	Read/write
Private	Links need to be identical to the value entered in the linked source object's variable value field.	Read-only

If you are a component designer creating a linked source object, you have the difficult job of determining how specific public and private Variable IDs, procedures and scripts operate between the source object and its links. Once you make your choices, making the object a source for linked objects is easy.

Guide: Sorting out the choices for linked objects:

The basic principles behind linked objects are straightforward.

- An object can be either:
- Copied to another location. In this case, the copy will not be linked to the original object

or

- Linked at another location.
- An object can have from one to several variables. Variables can be either:

- Public

or

- Private.
- Configuration written for linked source objects, including scripts or procedures, carry over to a linked object. However, the linked object's container can also have its own scripts or procedures.

A linked container is the shell of a linked object. You can configure properties for the container, such as movement and animation that, in addition to the linked properties, will affect the linked object's runtime behavior. This configuration has no affect on the linked source object.

Important:

- Because the linked container is a shell, it is a single entity. You cannot ungroup it, even if the source is actually a group of objects. Therefore, if any variables are public make sure they are at the highest group level. Once the linked container is placed on its target screen, you will not be able to change any public variable value that is lower than the top level, because you will not be able to access it.
- If a named object, including SmartObjects, is on a runtime-only screen (**.cimrt** *(page 56)*), you can only link it. You cannot copy it.

Linked vs Unlinked Object Copies Overview

- Linked objects.
- Unlinked copies.

Linked Objects

Linked object notes

- Components of objects that are linked to a source object include.
- A source link **container**.

- Can be configured with its own properties, e.g. color, movement, animation.
- Contains the path to the linked source object.

- A **dynamic copy of the source object**

Is updated when a component designer updates the source object (after the linked object screen is re-opened).

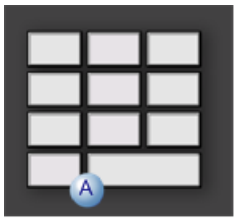
- When working with a link container, a screen designer:

- Can change the value of only public Variable IDs assigned to the source object.
- Cannot change any of the properties configured in the linked source object.

Linked object example overview

A screen and/or component designer:

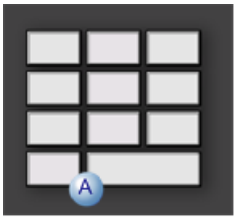
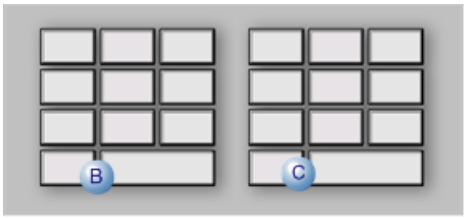
1. Selects **A** on the CimEdit SCREEN1.



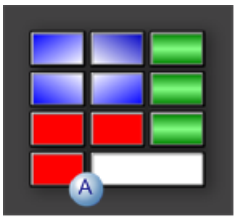
2. Names the object on the General tab of the Properties dialog box.

Note: Objects that will be linked must have a name.

3. Drags A while pressing Ctrl+Shift to create linked objects on SCREEN 2.

SCREEN 1	SCREEN 2
	

4. Changes A's colors. The linked objects' colors change automatically.

SCREEN 1	SCREEN 2
	

Unlinked Copies

Unlinked copies notes

An unlinked copy of an object initially contains all the properties of the original object. However, when the component designer updates the original object, the unlinked copy will not be updated.

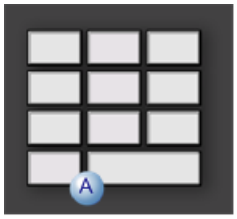
When working with an unlinked object, a screen designer can:

- Edit the value of any variable ID and scripts assigned to the unlinked object.
- Make changes to the unlinked object's properties; including those that were originally configured in the linked object.

Unlinked copies example overview

A screen and/or component designer:

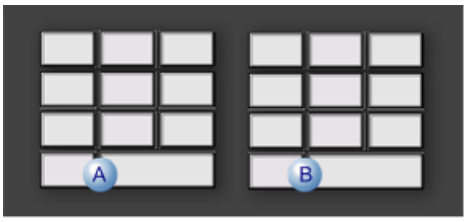
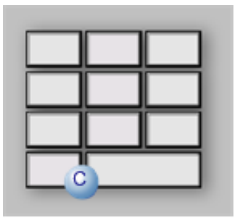
5. Selects object A on CimEdit SCREEN1.



6. Copies **A**.

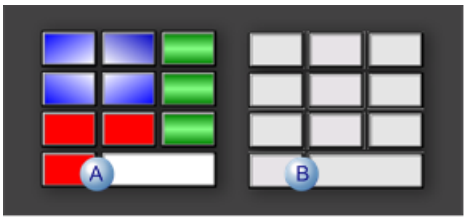
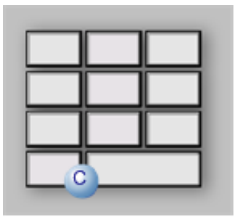
7. Pastes copies of A to:

B	Another location on SCREEN1.
C	SCREEN2.

SCREEN 1	**SCREEN 2**
	

8. Changes A's colors. Unlinked objects' colors

B	Do not change.
C	Do not change.

SCREEN 1	SCREEN 2
	

Public or Private Variables

A Variable ID can be a public or a private variable.

A component designer can make a Variable ID in a linked object either public or private. Using:

Variable	A Screen Designer:
Public variables	Can enter a value for the Variable ID in the link container, thus making some values unique to that container.
Private variables	Cannot change the value for the Variable ID in the link container.

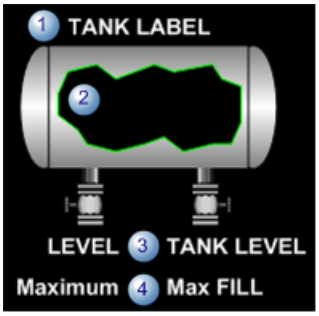
Example

The following example demonstrates configuration for a source/linked object.

- Source object configuration
- Linked container configuration
- Linked object runtime test
- Later changes

Source Object Configuration

A source object group includes the following objects.



1	Label that identifies the tank.

2	Tank that will fill based on an assigned variable value.
3	Text object that reports the runtime level.
4	Text object that reports the maximum allowed level.

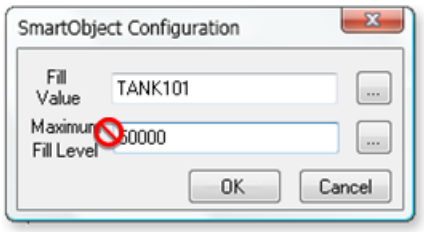
Following is the configuration for the source object group.

Item Type	Name	Level	Public/ Private	Value assigned at the Source	Comments
Name	TANKS	Top level group	NA	TANKS	TANKS is the source object name for a group that includes a tank and a label.
Variable	`LABEL`	TANKS	Public	No value	Text object that can be changed for the linked container.
Variable	`FILL_LEVEL`	TANKS	Public	No value	Value is a point that can be changed for the linked container. `FILL_LEVEL` is assigned to the: • Tank fill animation. • TANK LEVEL text.
Variable	`Maximum_Fill`	TANKS	Private	15000	Value cannot be changed for the linked container. Maximum_Fill is assigned to the: • Tank fill animation. • Max FILL text.

Linked Container Configuration

Because the tank is a SmartObject, when a screen designer creates a linked object, a SmartObject Configuration dialog box opens.

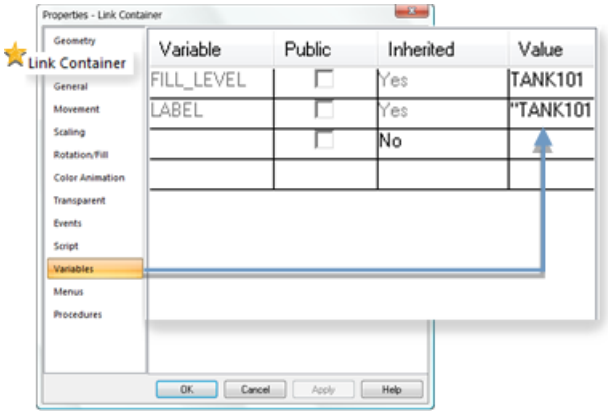
The screen designer enters the following values.



Field	Variable	Value	Description
Fill Value	Public	TANK101	A point that reports the actual level in TANK101.
Maximum Fill Level	Private	50000	Although the field appears to be read/write, 50000 will not be the value used. The source object's private variable had been assigned the value, 15,000, which will be the value used.

Once the object is placed on the CimEdit screen, the screen designer

1. Opens the object's Properties - Link container dialog box.



   The Link Container tab confirms that the object is a link container.

2. Selects the Variables tab.

3. Enters "TANK101" for the (public) Label variable.

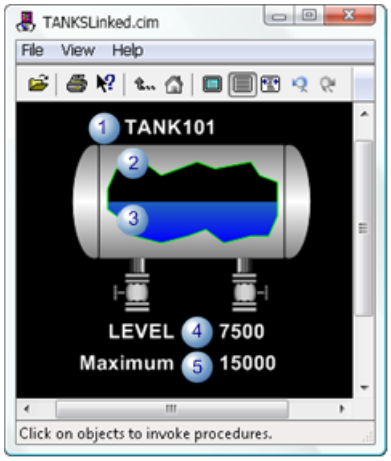
   Notes

   - TANK101 is already entered.
   - The variable Maximum_Fill_Level is not listed because it is a private variable and the value cannot be changed.

4. Clicks OK to close the dialog box.

   Linked Object Runtime Test

   The following occurs when the CimView screen with the linked object opens.



	Object	Value	Variable	Entered in the:
1	Label	"TANK101"	Public	Link container's Properties dialog box.
2	Maximum Fill	15000	Private	Source object's Properties dialog box.
3	Fill Value	TANK101	Public	Link container's SmartObject Configuration dialog box.
4	Fill Value	TANK101	Public	Link container's SmartObject Configuration dialog box.
5	Maximum Fill	15000	Private	Source object's Properties dialog box.

Later Changes

After the linked container was inserted on a CimEdit screen, the screen designer decided to change some source text to red.

The linked container screen is closed and re-opened.

The text that was changed in the source has changed to red in the link container.

Linked Object Configuration

Linked Object Configuration

If you are a component designer creating a linked source object, you have the difficult job of determining how specific public and private Variable IDs, procedures and scripts operate between the source object and its links. Once you make your choices, making the object a source for linked objects is easy.

guide: : When sorting out the choices for linked objects:

Variables	• Need to have different values for different links, make the variable public • Should be identical to the linked source object's variable, make the variable private
Scripts or procedures	• Need to be executed for all the linked objects, write them for the linked source object. • Are unique to one or more linked containers, write them for the linked containers only.

Steps to create linked objects are:

Step 1 *(page 337)*	Create linked objects.
Step 2 *(page 342)*	Create links to a source object.
Step 3 *(page 343)*	Change linked objects.

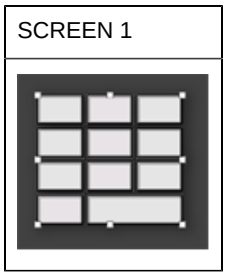
Step 1. Create Linked Objects

1 *(page 337)*	Create a source for linked objects.
2 *(page 339)*	Save the screen with the source in an easy to access location.
3 *(page 339)*	Create links to the source object.
	Create an Object that can be the Source for a Linked Object

1. Create the source for a linked object.

1. Open CimEdit.

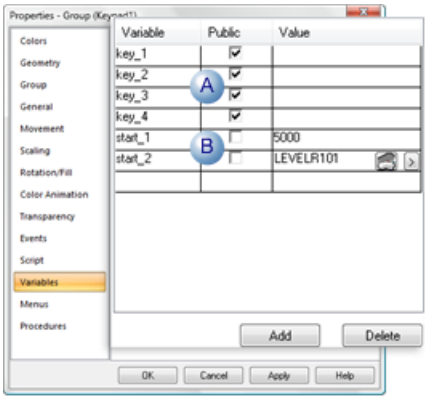
2. Create all the objects you want to include in the linked object.



Note: The object can be a single object or group. Most often it will be a group. the remainder of this procedure assumes that a group will be linked.

1. Open the Properties - Group dialog box.
2. Name the group.

1. Create public or private variables needed in the group as follows.



A	Checked	Public	Values can be entered for the variable ID in the link container.
			This makes some values unique to that container.
B	Clear	Private	A value cannot change for the variable ID in the link container.

1. Implement events, procedures and scripts as needed.
2. Save the CimEdit screen that now contains the source object.

Result: Now screen designers can create links to the source object on other screens and change the public variables to those needed for each new link.

Note: Screen designers can also do a normal copy of the source object to create an unlinked object on a new screen. However, when they do, any changes you make to the source object will not automatically be made to the copied object.

1. Save the screen with the source in an easy to access location.

The screen with the source objects can be in any location that is available to the screens that will contain the links.

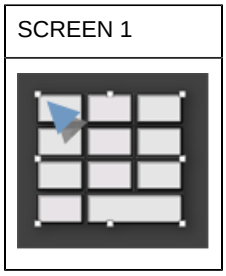
A convenient way to make the objects available is to add the screens to the Object Explorer.

1. Create links to the source object.

Create a link from either of the following.

- Source screen
- Object Explorer

1. Select a source object on the source screen, SCREEN1.

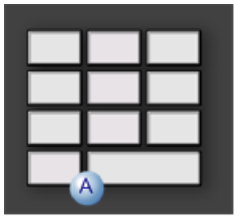
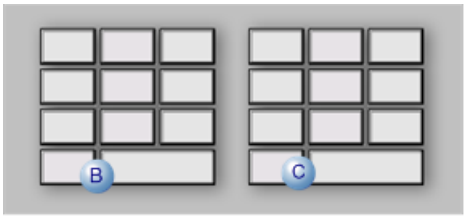


1. Hold down Ctrl+Shift keys and the left-mouse button.
2. Drag a link of the source object to another location button.

Note: Keep holding down the keys and button.

As you move the linked object to the destination screen, SCREEN2, the cursor looks like this.

1. Use the mouse or arrow keys to position the container where you want it on the destination screen, SCREEN2.

SCREEN 1	SCREEN 2
Source	Links
	

Result: The container is linked to the link source object.

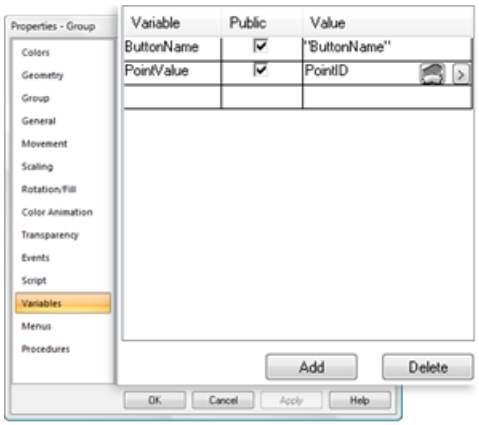
Example. Create an Object that can be the Source for a Linked Object

This example creates a button with a script that increments a point and text that displays the current value of the point.

1. Open the CimEdit screen where you want to save your source objects.
2. Place a text button object on the CimEdit screen.
3. Place a text object underneath the button.

4. Change the text for the button object to ButtonName; name the object SetpointButton.
5. Change the text for the text object to Setpoint; name the object SetpointValue.
6. Group the two objects.

7. Open the group's Properties dialog box.
8. Name the group IncrementSetpoint.
9. Create two public variables that look like this:



Variable	Public	Value
`ButtonName`	Checked	"ButtonName"
`PointValue`	Checked	PointID

10. Select the SetpointButton object in the group.
11. Open the SetpointButton object's Properties dialog box.
12. Do the following configuration in the Properties dialog box.

    Text Button.

    In the **Expression** field, enter the variable `{ButtonName}`.

    Event

    a. Select a MouseUp event.
    b. Create a new script in the **Action** field.

    The script will increment the value of the Point ID assigned to the variable `{PointValue}`, as follows.:

```
Sub OnMouseUp(x As Long, y As Long, flags As Long)
   Dim var As CimObjectVariable
   Set var = CimGetScriptOwner().GetVariable("PointValue")
   i = PointGet(var.Value)
   PointSet var.Value, i+1
End Sub
```

    Note: Even though the button owns the script, the variable will be resolved at the group level.

13. Select the Group tab.
14. Select the SetpointValue object in the group.

15. Open the SetpointValue object's Properties dialog box.
16. Do the following configuration in the Properties dialog box.

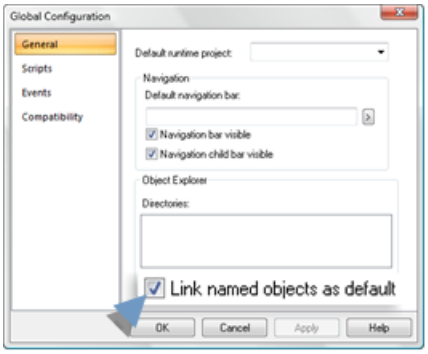
    Text

    In the **Expression** field, enter the variable `{PointValue}`.

17. Save the screen and close it.

Step 2. Create Links to a Source Object

Important: The procedure to create a link container vs. an unlinked duplicate depends on whether the Link named objects as default checkbox in the CimEdit Global Configuration dialog box *(page 558)* is clear or checked.



Link named objects checkbox is clear

1. Select the source object.
2. Hold down Ctrl+Shift keys while you drag the object to another screen.

   Result: The container is linked to the source object.

   Note: To simply duplicate the object, drag it to another screen.

   Link named objects checkbox is checked

3. Select the source object.
4. Drag the source object to another screen.

The container is linked to the source object.

Note: To simply duplicate the object, hold down the Ctrl+Shift keys while you drag it to another screen.

Step 3. Change Linked Objects

Important: If the link container was created from a read-only (.cimrt) screen, the obejct is also read-only. The following changes can only be applied to linked containers that were created from read/write (.cim) screens.

- Updated linked objects.
- Change the path type for all linked objects.
- Break the link to the source object.

Update Linked Objects

The component designer can update the source object's properties at any time. Changes to links appear when an individual viewing a screen with a linked container, opens the screen after the designer saves the screen containing the source object.

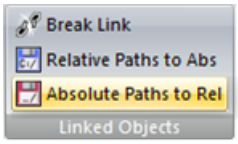
If you have a screen open that uses the link container *(page 337)* , when the screen containing the source object is saved, the link does not update immediately. You must close your screen and reopen it.

Note: Once you place a linked object, it remains in the location where it is placed. If you manipulate the source object, such as rotating it or changing its size, the linked object will change around its own center. If you move the source object, the link position will be unaffected.

Change the Path Type for all Linked Objects. You can change the path type for all linked objects.

Absolute Paths.

1. Select a linked container.
2. Click Absolute Paths to Rel in the Tools>Linked Objects Group on the CimEdit Ribbon bar.



Result: The path for all linked objects on the CimEdit screen are now absolute.

Static Paths

3. Select a linked container.
4. Click static Paths to Abs in the Tools>Linked Objects Group on the CimEdit Ribbon bar.

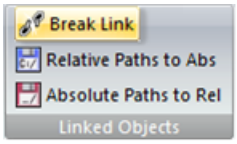
   Result: The path for all linked objects on the CimEdit screen are now static.

   Break the Link to the Source Object

   If you break the link to the source object, the link container will be a stand-alone object and can be configured as a stand-alone object.

   When the source is updated, the object with a broken link will no longer be updated.

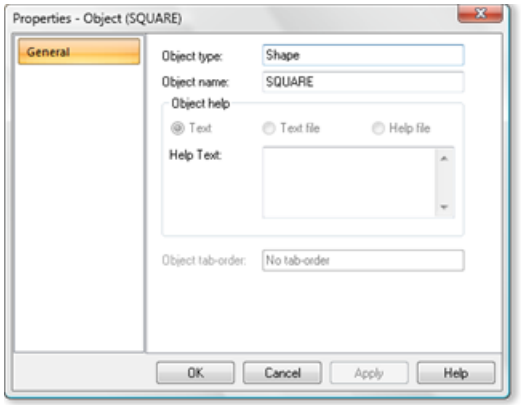
5. Select a linked container.
6. Click Break Link in the Tools>Linked Objects Group on the CimEdit Ribbon bar.



The linked container is now a stand-alone object. When you open the object's Properties dialog box, the Link Container is no longer listed.

Find a Linked Object Source

1. Open the linked object's Properties dialog box.
2. Select Linked Container.



The source file path, link source object and source description are displayed for the linked object.

Protected Link Source Objects

Protected Link Source Objects

CimEdit provides you with a powerful capability to create protected linked source objects. Protected links contain the properties of the source object.

However, a protected link user will not be able to

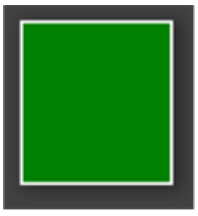
- View or decompose the source code or configuration.
- Break the link.
- Change the absolute/static path specification.

You create these links by saving them to a runtime-only screen (**.cimrt**) and distributing that screen for use instead of the editable (**.cim**) screen.

Example: Create a Link from a Protected Screen

Example: Create a Link from a Protected Screen

Creating a linked container from a protected source object uses the same procedure as this simple example to create a link from a rectangle object on a protected (*.cmrt *(page 55)*) screen.

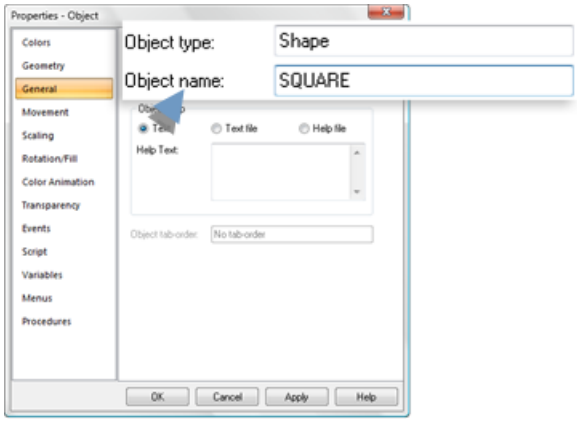


Step 1 *(page 346)*	Name the rectangle object.
Step 2 *(page 346)*	Configure properties for the object.
Step3 *(page 348)*	Create a runtime-only screen.
Step 4 *(page 349)*	Create a second .cim screen.

Step 5 *(page 349)*	Insert a link container on the new .cim screen.
Step 6 *(page 350)*	Enter a value for the link's public variable.
Step 7 *(page 351)*	Configure horizontal scaling for the link container.
Step 8 *(page 351)*	Test the screens in CimView.

Step 1. Name the Rectangle Object

1. Open the rectangle object's Properties dialog box.
2. Select **General**.
3. Enter **Square** in the **Object name** field.



The named object can now be linked.

Step 2. Configure Properties for the Object

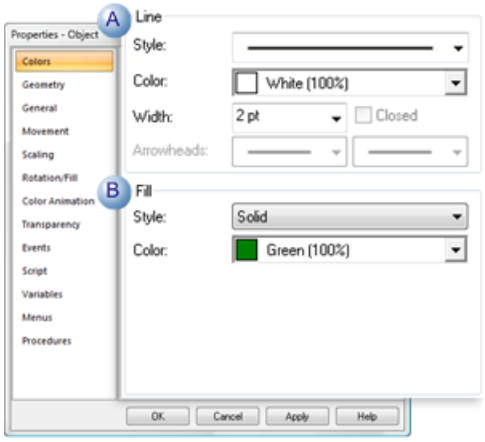
Configure the following features for the source rectangle object.

- Basic color.
- Public variable.
- Runtime fill.

Basic Color

1. Select **Colors**.

2. Do the following.

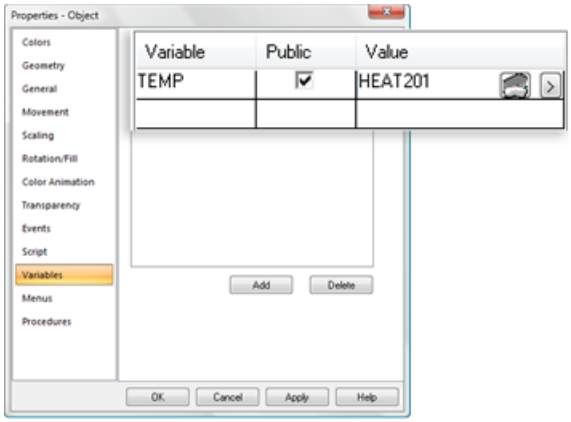


A	Add a White border
B	Select Green for the rectangle's default color.

Public Variable

3. Select **Variables**.

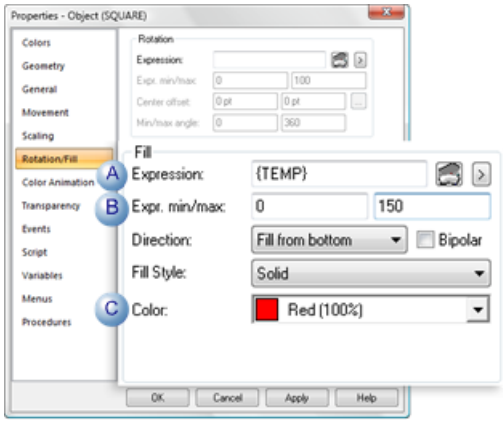
4. Create a variable, as follows.



Field	Value
Variable	TEMP
Public	Checked
Value	HEAT201 (an analog point used in this example). **Note:** Public variables can be blank for the source object if it is not intended for runtime use.

Animated Fill

5. Select **Rotation/Fill**.
6. Enter the following.



	Field	Value
A	Expression	{`TEMP`} **Note:** `TEMP` is the variable created on the Variables tab.
B	Exp. min/max	0/150
C	Color	Red

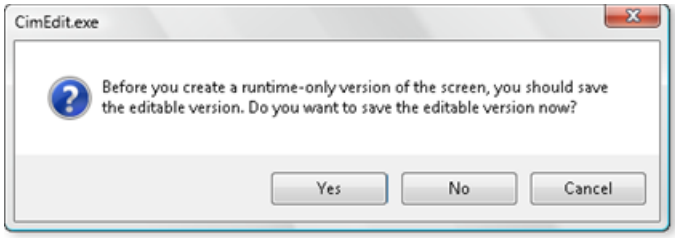
Step 3. Create a Runtime-only Screen

1. Click the CimEdit button .
2. Select **Create Runtime only**.

   A Message box opens asking you:

   Before you create a runtime-only version of the screen, you should save the editable version. Do you want to save the editable version now?

3. Click **Yes**.



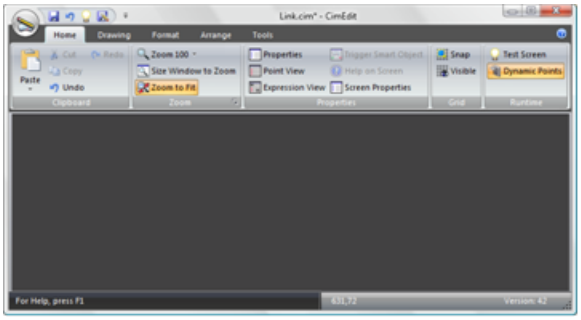
CimEdit saves the read/write and runtime only screens.

Note: In this example they are named LinkedSQUARE.cim and LinkedSQUARE.cimrt.

Step 4. Create a second .cim Screen

1. Right-click **Screens** in the Workbench left pane.
2. Select New on the Popup menu.
3. Save the .cim screen with the name Link.cim.

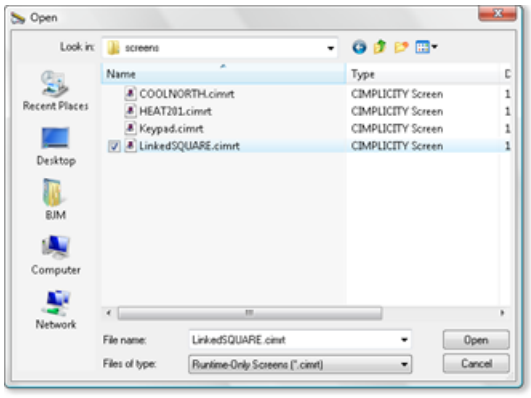
A new CimEdit screen is created.



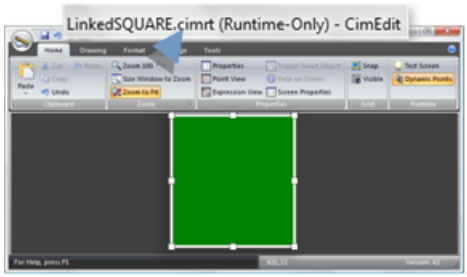
Step 5. Insert a Link Container on the new .cim Screen

1. Open the LinkedSQUARE.cimrt screen.

   Note: Make sure Runtime Only Screens (*.cimrt) is selected in the **Files of type** field in order to find the runtime-only screens.

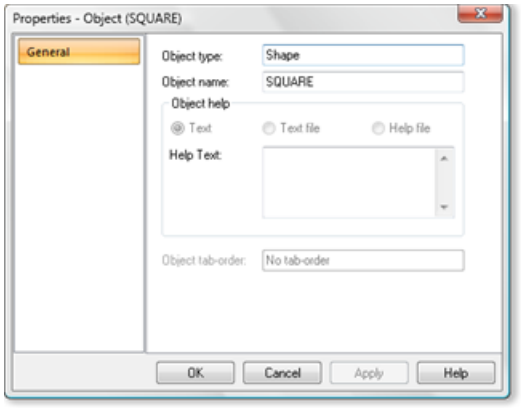


2. Select the object named **Square** on the runtime-only screen.



3. Right-click **Square** to review its properties.

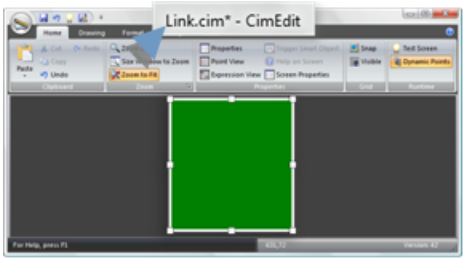
   Result: The configuration for the source object is completely protected. The Properties - Object tab. There are no other tabs.



4. Press Shift+Ctrl while dragging the object from Runtime_Link.cimrt to Link.cim.

   Note: If Link named objects as default is checked in the Global Configuration dialog box, simply drag the object from the source to target screen.

A link container is added to Link.cim.

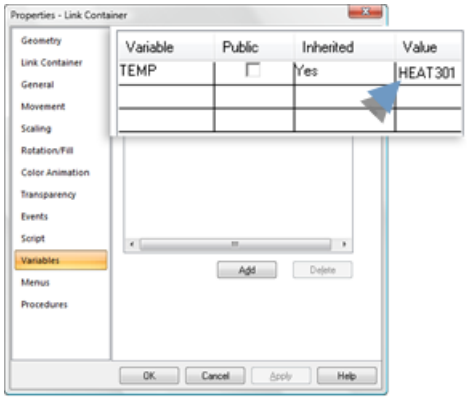


Step 6. Enter a Value for the Link's Public Variable

1. Open the Properties - Object dialog box for the link.

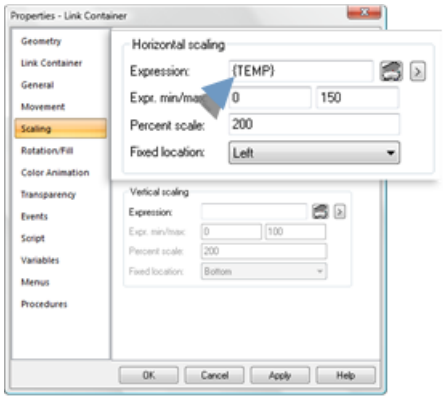
2. Select the Variables tab.
3. Enter a value in the **Value** field that is different from the value entered for the source object.

   Note: The example used an analog point named HEAT301.



Step 7. Configure Horizontal Scaling for the Link Container

1. Select the Scaling tab.
2. Enter the public variable {`TEMP`} in the Expression field.



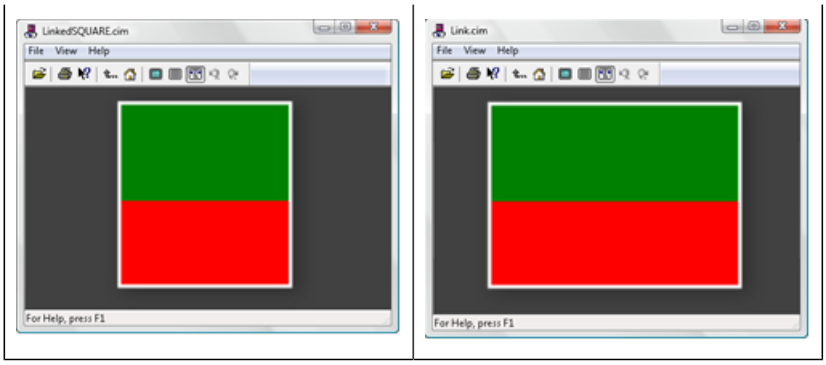
Step 8. Test the Screens in CimView

1. Double-click Runtime_Link.cimrt in the Workbench right pane.
2. Click Runtime in the Home>Runtime group on the Link.cim Ribbon bar.

   Results

   The rectangle on each screen displays the following.

Source Object	Link Container



A	The source object on the protected screen displays its configured runtime properties.
B	The link container: • Retains the properties of the source object, even though the source configuration did not display in the protected source's Properties - Object dialog boxes. • Displays the Scaling runtime property that was configured for the link container.

Runtime Movement and Animation

Runtime Movement and Animation

CimEdit provides several choices to create activity on your screens that makes it easy for a CimView user to quickly determine the status of a point or expression.

Step Number	Description
1 *(page 353)*	Object movement
2 *(page 359)*	Object scaling.
3 *(page 363)*	Object rotation.
4 *(page 368)*	Object fill.
5 *(page 373)*	Object transparency
6 *(page 376)*	Animated picture animation speed.
7 *(page 381)*	Attribute animation.
8 *(page 398)*	Text value animation
9 *(page 410)*	Frame animation
10 *(page 425)*	Copy/Paste Animation
11 *(page 427)*	Angle Animation

1. Object Movement

- Movement overview.
- Movement configuration.

Movement Overview

You can use movement on a CimEdit screen to:

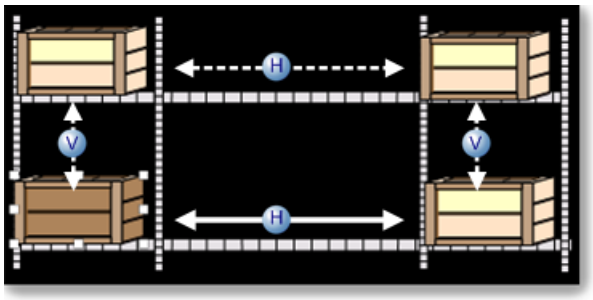
- Reflect the change in a value
- Set a value

Movement is an effective way to alert an operator that the value of a point or expression changed.

It is easy to make an object move, on the screen and it can be very effective. Because CimEdit provides you with the tools to precisely define your begin and end positions, you have complete control over how the object will move.

Movement involves the static distance between beginning and ending positions. Therefore, if you use exact location numbers you can still place the object anywhere on the screen and the movement will be the same.

1. Select the object you want to move.
2. Open the object's Properties dialog box.
3. Configure either or both of the following.

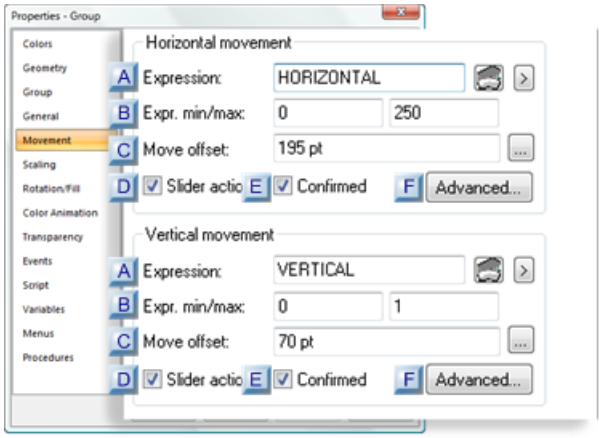


H	Horizontal movement.
V	Vertical movement.

Movement Configuration

Select Movement in the Properties dialog box.

Runtime horizontal and vertical movement specifications are as follows.



rect 61, 32, 88, 56 *(page 355)*
rect 61, 154, 88, 178 *(page 355)*
rect 59, 57, 86, 79 *(page 355)*
rect 60, 176, 87, 198 *(page 355)*
rect 236, 220, 263, 247 *(page 358)*
rect 238, 101, 265, 128 *(page 358)*
rect 144, 220, 171, 247 *(page 358)*
rect 144, 220, 171, 247 *(page 358)*
rect 143, 102, 170, 129 *(page 358)*
rect 60, 220, 87, 246 *(page 357)*
rect 60, 101, 87, 127 *(page 357)*
rect 60, 198, 87, 222 *(page 355)*
rect 60, 198, 87, 222 *(page 355)*
rect 59, 78, 86, 102 *(page 355)*

A *(page 355)*	Expression
B *(page 355)*	Expr min/max
C *(page 355)*	Move offset
D *(page 357)*	Slider action
E *(page 358)*	Confirmed

F *(page 358)*	Advanced: Execution Condition

A	Expression

A point ID or expression that will determine the object location between the START and END positions.

	Opens the Expression Editor.
	Opens a Popup menu.

B	Expr. min/max
	Min
	Max

C	Move offset

The Move offset defines the distance between the object's START position and END position on the CimEdit screen.

The direction can be:

Horizontal	Left to right or right to left.
Vertical	Bottom to top or top to bottom

During runtime, CimView calculates the object's location based on the:

- Expression minimum and maximum values,
- Object's START position and the offset specification.

Do one of the following to calculate the offset.

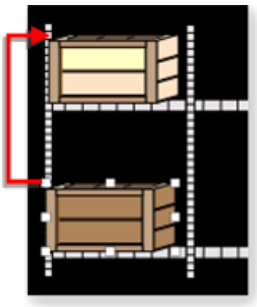
- Manual calculation
- Automatic calculation

Manual Calculation

When you calculate the offset manually, the object's start/end position is calculated from the following sides.

- Horizontal - Calculate from the object's left side.

- Vertical - Calculate from the object's top.

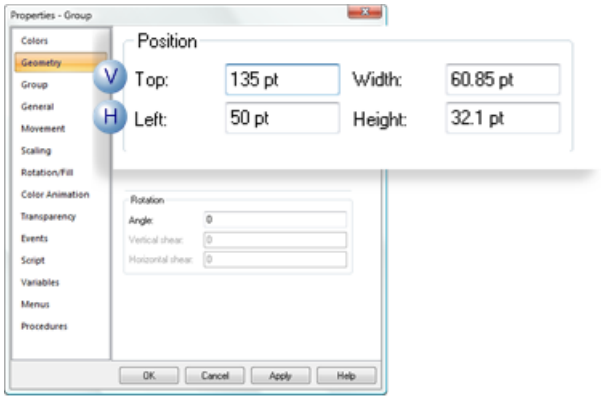


Status bar

- Place the object's top/left side in the START position.
- Place the cursor in the object's top/left corner.
- Note the Y-axis/X-axis positions in the CimEdit status bar.
- Do the same for the object's END positions.

Properties Dialog Box

- Place the object's top/left side in the START position.
- Open the Properties dialog box for the object.
- Select Geometry.
- Note the values in the Position Top and/or Left fields.



	Position	Field
H	Horizontal	Left
V	Vertical	Top

- Close the Properties dialog box.
- Do the same for the object's END position.
- Subtract the START from the END values.

The value is the distance the object will travel on the CimView screen.

Example: Horizontal

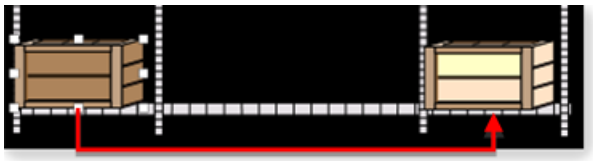
END	245

START	50
Offset value	195

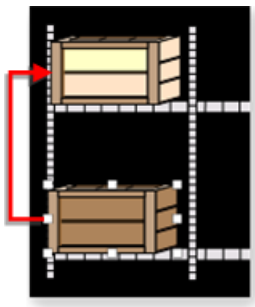
Automatic Calculation

When you calculate the offset automatically, the object's start/end position is calculated from the following sides.

- Horizontal - Calculate from the object's horizontal center.



- Vertical - Calculate from the object's vertical center.

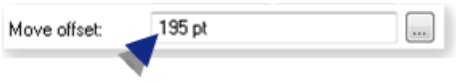


- Place the object's center in the START position.
- Open the object's Properties dialog box.
- Select Movement.
- Click the Offset button to the right of the Move offset field.

The Properties dialog box closes.

Click the END position on the CimEdit screen that will be the center of the object.

The Properties dialog box re-opens with the START to END value entered in the **Move offset** field.



D	Slider action

Check to enable runtime users to set an analog point by moving the object horizontally on the CimView screen.

Important: The slider action is enabled for analog or real points only.

During runtime when a slider object is released, its position between the beginning and end of the horizontal or vertical distance that you specify represents the value that will be written to the point or points.

You can use the slider action alone or combine it with other procedures.

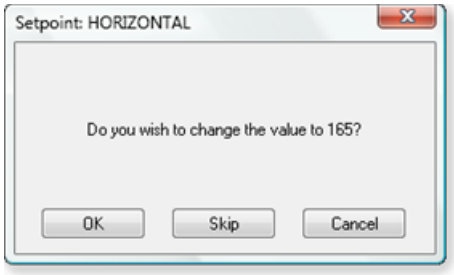
When you do, the order of execution is:

- The Mouse down procedure, if defined, runs.
- The Slider action runs.
- The Mouse up procedure, if defined, runs.

E	Confirmed

Check to have CimView display a message asking a user for confirmation to move the object.

Example



F	Advanced: Execution Conditon

Click the Advanced button to open an Execution Condition dialog box.

The execution condition directs CimView to disable the slider action if an entered expression is False.

Specifications are as follows.

Option	Description
Expression	One or more point ID's and mathematical operators that can be evaluated. The expression must evaluate to TRUE during runtime in order for a user to access the Setpoint action.

Option	Description
Display message when disabled	(Available if an expression is entered in the Expression field) Displays the entry in the message during runtime when a user attempts to perform the Setpoint action and the Execution Condition Expression evaluates to FALSE.
Message	String that displays in a Message box during runtime when a user attempts to perform the Setpoint action and the Execution Condition Expression evaluates to FALSE. **Tip:** Click the Text Box button to the right of the Message field to open a Text Box for entering a longer message.
Translate message	The message text string can be translated or revised for selected languages in the CIMPLICITY Language Mapper file. The translation or revised text display in CimView when the assigned language is selected. For more information on translation, see Translation *(page 66)*.

Execution Condition evaluated for a Slider action

When an Execution Condition is evaluated as:
True
False and No message is defined
False and A message is defined
If the user selects:
Yes
No

Examples of how an Execution Condition is Used

If a custom application provides security information as a CIMPLICITY point, this feature can be used to provide security control for procedure execution.

If an application runs in modes, such as Manual and Automatic, this feature can be used to disable certain procedures while in Automatic mode.

Note: The object will not begin to move until the minimum value has been exceeded.

2. Object Scaling

- Scaling overview
- Scaling configuration

Scaling Overview

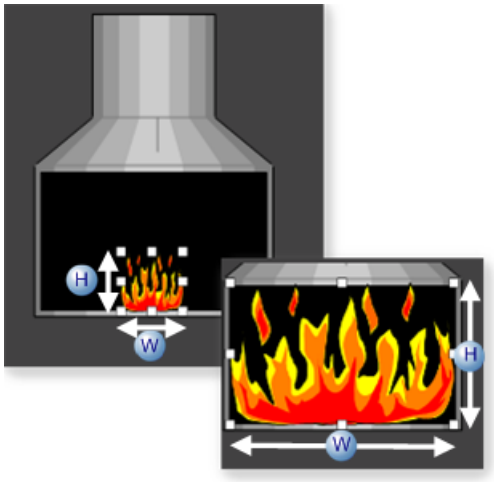
When you use CimEdit's scaling feature to configure an object, its horizontal and / or vertical dimensions will change during runtime based on the current value of one or two expressions. This is another way for a CimView user to be ably to glance quickly at a CimView screen and see when a change occurs.

If you define both horizontal and vertical scaling for an object, you may use different expressions, minimums, maximums, percent scales, and fixed locations for each scaling.

Note: In order to make sure text is legible:

- **Text objects** will only change size when scaled vertically.
- **Text Button objects** with Auto Size checked will not change size when scaled.

1. Select the object you want to scale.
2. Open the object's Properties dialog box.
3. Configure either or both of the following.

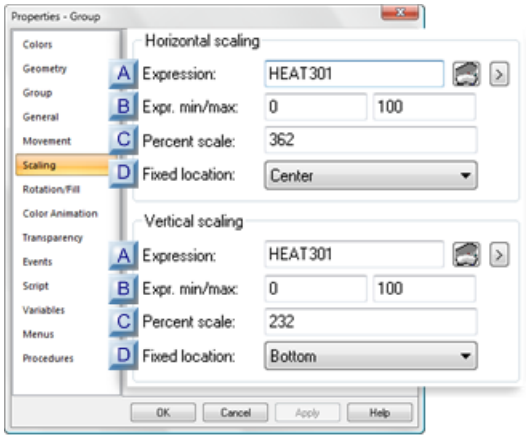


- W - Width scale.
- H - Height scale.

Scaling Configuration

Select Scaling in the Properties dialog box.

Runtime horizontal and vertical scaling specifications are as follows.



rect 64, 209, 84, 229 *(page 363)*
rect 66, 94, 86, 114 *(page 363)*
rect 64, 188, 84, 208 *(page 362)*
rect 65, 74, 85, 94 *(page 362)*
rect 65, 52, 85, 72 *(page 361)*
rect 65, 167, 85, 187 *(page 361)*
rect 65, 34, 85, 54 *(page 361)*
rect 64, 147, 84, 167 *(page 361)*

A *(page 361)*	Expression
B *(page 361)*	Expr min/max
C *(page 362)*	Percent scale
D *(page 363)*	Fixed location

A	Expression

A point ID or expression that will determine the object location between the START and END positions.

	Opens the Expression Editor.
	Opens a Popup menu.

B	Expr. min/max	
	Min	
	Max	

C	Percent scale

The percent scale is calculated by dividing the object's end size (largest width/height) by the object's start size (smallest width/height).

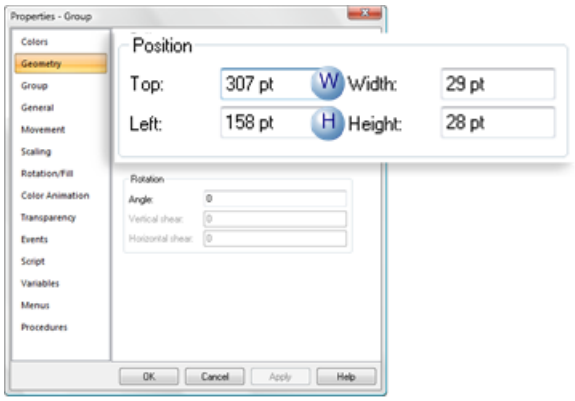
Status bar

4. Place the object's horizontal/vertical width in the START size.
5. Select the object.
6. Note the horizontal/vertical sizes in the CimEdit status bar.
7. Do the same for the object's END position.

   Properties Dialog Box

8. Place the object's horizontal/vertical width in the START size.
9. Open the Properties dialog box for the object.
10. Select Geometry.
11. Note the values in the **Position Width** and/or **Height** fields.

    Example



	Dimension	Field
W	Width	Width
H	Height	Height

12. Close the Properties dialog box.
13. Do the same for the object's END size.

14. Divide the END by START values.

    The value is the size the object will expand to on the CimView screen.

    Example: Horizontal

    ```
    Maximum Width/Minimum Width = Percent
    ```

    ```
    105/29 = 362(%)
    ```

    Example: Vertical

    ```
    Maximum Height/Minimum Height = Percent
    ```

    ```
    65/28 = 232(%)
    ```

D	Fixed Location

    The starting point for scaling is as follows.

Option	Side of object starts from:
Left	Left
Center	Center
Right	Right

3. Object Rotation

- Rotation overview
- Rotation configuration
- Runtime example

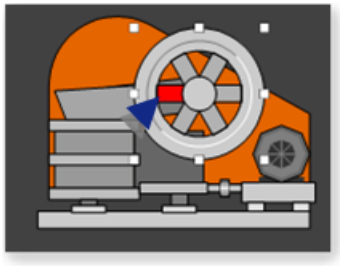
Rotation Overview

You can make the following objects rotate during runtime based on the value of one or two expressions.

- Lines
- Shapes
- Text
- Buttons
- Groups
- Frames

Note: One of the many useful applications for rotation is to place one or more gauges on the screen. The CimEdit Smart Object library provides you with gauges for which you only need to enter the expression they represent. During runtime, they will rotate to reflect the expression's value.

1. Select the object you want to rotate.



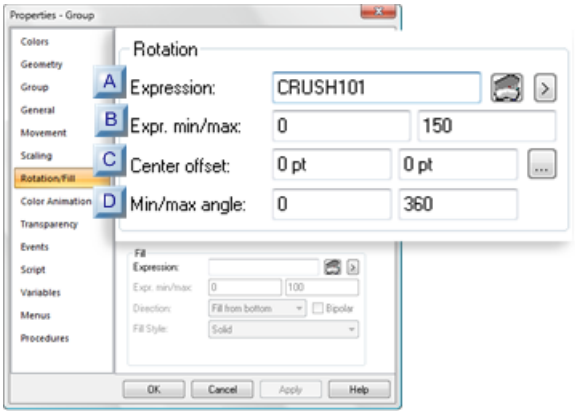
2. Open the object's Properties dialog box.

3. Configure the rotation.

   Rotation Configuration

   Select Rotation in the Properties dialog box.

   a. Rotation options are as follows.

   b. 

   rect 55, 111, 79, 134 *(page 367)*
   rect 55, 86, 79, 111 *(page 365)*
   rect 55, 60, 79, 85 *(page 365)*
   rect 55, 35, 79, 60 *(page 365)*

A *(page 365)*	Expression
B *(page 365)*	Expr. min/max

C *(page 365)*	Center offset
D *(page 367)*	Min/max angle

A	Expression

A point ID or expression that will determine the object location between the START and END positions.

	Opens the Expression Editor.
	Opens a Popup menu.

B	Expr. min/max	
	Min	
	Max	

Note: If the **Expression** field contains a single point ID with no operations performed on it and the point has Display Limits defined, you may leave the **Expr. Min/Max** fields blank.

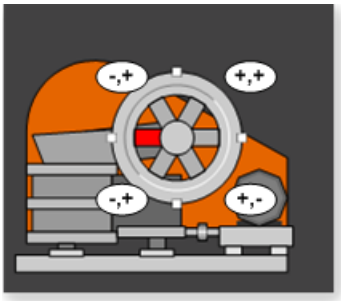
CimView will use the Display Limits configured for the point ID for the expression minimum and maximum values when the screen file opens.

C	Center offset

The **Center offset** specifies the x, y offset values where the center of the object will display during runtime.

guide: Offset guidelines

- Negative or positive numbers can be entered in either field.
- The higher the entry the further the object will display from the object center.
- 0,0 will cause the object to have no rotate in its default position.



Manual Offset

Enter the values that the object will be offset as follows.

X	Center offset from the expression value along the X axis.
Y	Center offset from the expression value along the Y axis.

Automatic Offset

a. Click the **Quick Offset** button ... to the right of the **Center Offset** field.

The dialog box closes temporarily.

a. Move the cursor on the CimEdit screen to the where you want the center of rotation
b. Hold down the left-mouse button.

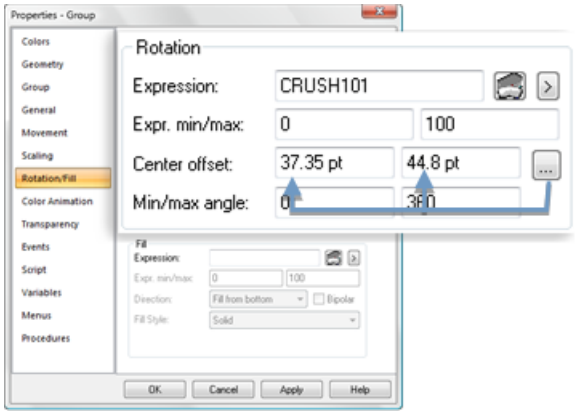
The cursor turns into a crosshair, indicating the object center location, surrounded by a circle.

The further the cursor moves from the original object center, the larger the circle becomes.

a. Release the left-mouse button.

The Properties dialog box re-opens.

The coordinates established when the mouse button was released, display in the **Center Offset** fields.



D	Min/Max angle

The **Min/Max angle** fields specify the object's minimum/maximum angles of rotation for the minimum/maximum expression values.

- Negative or positive numbers can be entered.
- The following values cause the to perform one full rotation around the center offset as the expression value increases from its minimum value to its maximum value.

Rotation	Min	Max
Clockwise	0	-360
Counter clockwise (default)	0	360

Runtime Example

A jaw crusher's wheel rotation has the following specifications

Exp. min/max	0, 100
Min/max angle	0, -360

- When the offset values are entered as 0, the wheel rests on the crusher with the red spoke to the left.
- A center offset other than 0, 0 would move the wheel around the crusher, based on the expression value, as follows

	Minimum/maximum angles			
Value	0/25			
25				
50				

75				
0 or 100				

4. Object Fill

- Fill overview
- Fill configuration
- Runtime example

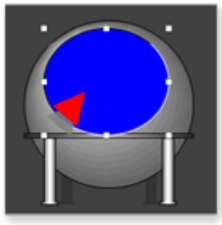
Fill Overview

Fills are a useful way to let a CimView user quickly see the level of a point or expression's value during runtime.

You can create fill animation for shapes that are single objects or included in a group.

Tip: The CimEdit Smart Object library provides a large assortment of fill Smart Objects, including tanks and a large number of fill shapes for you to use.

1. Select the object you want to fill.



2. Open the Properties dialog box.
3. Configure the Fill

   Fill Configuration

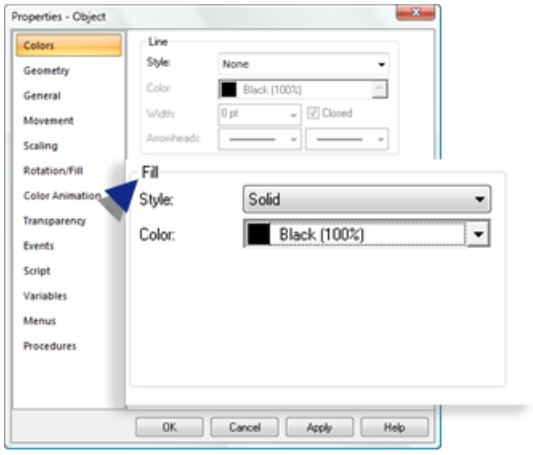
Fill configuration includes configuring the following.

- Default fill color.
- Animated fill.

Default fill color

4. Select Colors in the Properties dialog box.
5. Select a fill color or pattern in the Fill section.

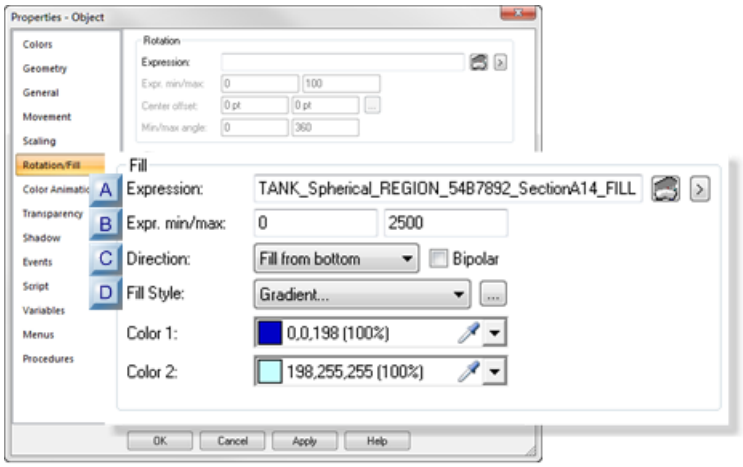
   This color will be the default color when the object is empty.



Animated Fill

Select Rotation/Fill in the Properties dialog box.

Configuration options are as follows.



rect 49, 103, 77, 125 *(page 370)*
rect 48, 123, 76, 145 *(page 370)*
rect 48, 144, 76, 166 *(page 371)*
rect 48, 166, 76, 188 *(page 372)*

A *(page 370)*	Expression
B *(page 370)*	Expr. min/max
C *(page 371)*	Direction
D *(page 372)*	Fill style

A	Expression

A point ID or expression that will determine how high the object will be filled.

	Opens the Edit Expression dialog box.
	Displays a Point Popup menu.

B	Expr. min/max

Min

Minimum value of the point ID or expression.

This value or lower will cause the object to have no fill; it will display only the default color.

Max

Maximum value of the Point ID or expression.

This value or higher will cause the object to be filled: it will display only the fill color.

Example

An object has the following specifications

Field	Value	Fill
Min	0	Black
Max	2500	Blue gradient

During runtime, the object displays as follows.

Min	Max

Note: If the **Expression** field contains a single point ID with no operations performed on it and the point has Display Limits defined, you may leave the **Expr. Min/Max** fields blank. CimView will use the Display Limits configured for the point ID during runtime.

C	Direction

The fill direction includes the following.

- Side of the object to start the fill.

Options are:

- Fill from bottom
- Fill from left
- Fill from right
- Fill from top
- Bipolar

Bipolar fill is based on the following.

Checked	Deviation from the mean of the difference between the maximum and minimum value of the point or expression.	
	Point/Expression value is:	

	Greater than the mean value	
	Equal to the mean value	
	Less than the mean value	
Clear	Expression value.	

D	Fill style

The fill style selection is the same procedure that is used for selecting color fills for other CimEdit procedures.

Runtime Example

- A tank fill expression specifications are:

Min	0
Max	2500

- The runtime value is 1700.

The fill display is based on the direction specifications.

Fill from:	Bi-Polar clear	Bi-Polar checked
Bottom		
Left		
Right		
Top		

5. Object Transparency

- Transparency overview
- Transparency configuration
- Transparency configuration and runtime example

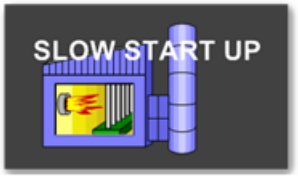
Transparency Overview

An object can become more or less transparent during runtime, based on a specified expression value.

1. Select the object whose transparency you want to change.

   Example

   A kiln will have text display SLOW START UP as it starts up and fade to invisible when the temperature reaches 45.

   The same kiln will have a DANGER sign slowly become opaque as the temperature goes from 130 through 150.

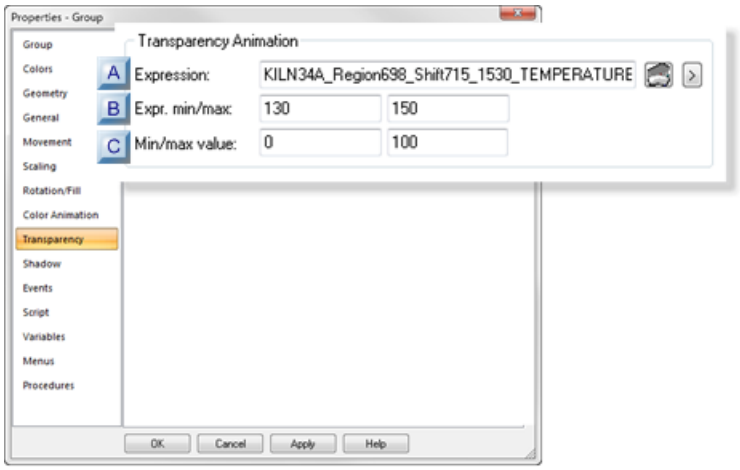
2. Open the objects Properties dialog box.

3. Configure the transparency.

   Transparency Configuration

   Select Transparency in the Properties dialog box.

   Transparency options are as follows.



rect 57, 29, 81, 52 *(page 374)*
rect 55, 52, 79, 75 *(page 374)*
rect 55, 77, 79, 100 *(page 374)*

A *(page 374)*	Expression
B *(page 374)*	Expr. min/max
C *(page 374)*	Min/max value

A	Expression

A point ID or expression that will determine the object transparency percent within the entered minimum and maximum values.

	Opens the Expression Editor.
	Opens a Popup menu.

B	Expr. min/max	
	Min	
	Max	

Note: If the **Expression** field contains a single point ID with no operations performed on it and the point has Display Limits defined, you may leave the **Expr. Min/Max** fields blank. CimView will use the Display Limits configured for the point ID during runtime.

C	Min/max value	
	Left field	

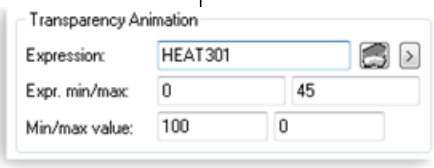
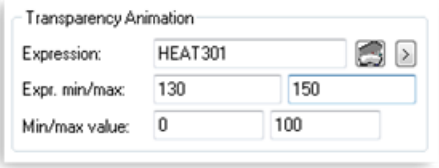
	Right field		

Transparency Configuration and Runtime Example

A CimView screen will report when a kiln is starting up and when it exceeds a minimum temperature.

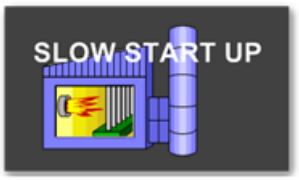
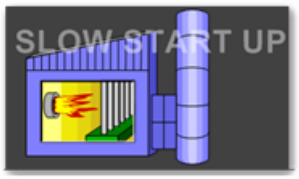
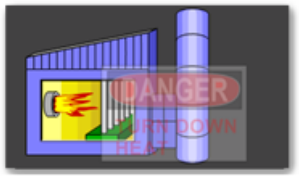
- Text reporting the startup will slowly fade off the screen.
- A danger sign will slowly become more visible.
- Both are based on values of the same point: HEAT301.

Configuration

A	The start up message HEAT301 transparency entries are as follows. 		
	Field		
	Expr. min/max		
	Min/max value		
B	The danger sign HEAT301 transparency entries are as follows. 		
	Field		
	Expr. min/max		
	Min/max value		

Runtime Performance

During runtime the startup text and danger sign display as follows.

Temp.	Startup Text	Danger Sign
0	100% Visible	Invisible
		
25	50% Visible	Invisible
		
100	Invisible	Invisible
140	Invisible	50% Visible
		
150	Invisible	100% Visible

6. Animated Picture Animation Speed

- Animated picture speed overview.
- Animated picture speed configuration.
- Runtime example.

Animation Picture Speed Overview

Animated picture files can combine more than one type of animation (e.g. movement and scaling and blinking, etc) to create a complex animation affect. By assigning value and percent ranges to one expression, you can cause the runtime animation to speed up or slow down based on changes in the expression value.

Currently animated files are saved as .gif files.

Animated Picture Speed Configuration

1 *(page 377)*	Place an animated graphic image on the screen.
2 *(page 377)*	Insure the .gif file transparency settings remain transparent.
3 *(page 378)*	Select animation speed settings.

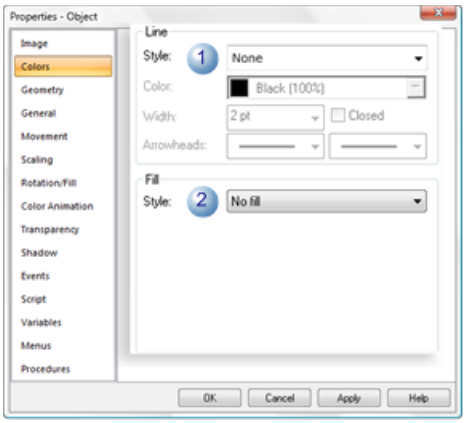
Place an animated graphic image on the screen.

Follow the steps to place a graphic picture *(page 115)* object on a CimEdit screen and select if the file should be embedded or linked *(page 118)* .

Insure the .gif file transparency settings remain transparent.

In order to maintain all of the transparent sections in the animated .gif, the object must have no CimEdit line or color fill.

1. Select Colors in the Properties - Object dialog box.
2. Select the following.



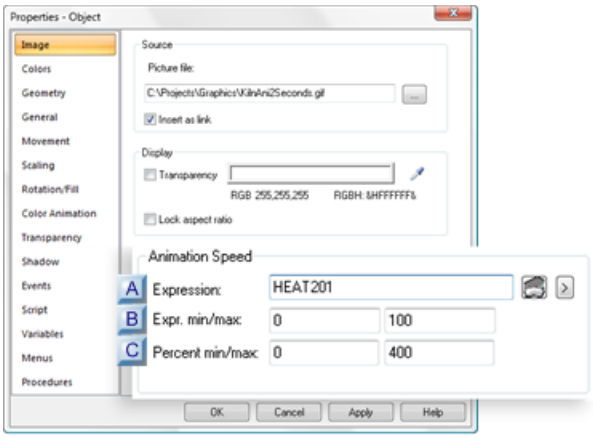
	Field	Select
1	Line Style	None (unless you want the image to have a border).
2	Fill Style	No fill

Important: The CimEdit display transparency feature in the Image section in the Properties - Object dialog box does not apply to animated .gif files. This feature is not supported.

Select animation speed settings.

The speed that the animated .gif displays in CimView is calculated against the actual animated .gif speed.

1. Select Image in the Properties - Object dialog box.
2. Select the following.



rect 69, 205, 90, 226 *(page 379)*
rect 68, 184, 89, 206 *(page 379)*
rect 68, 164, 89, 185 *(page 379)*

A *(page 379)*	Expression
B *(page 379)*	Expr. min/max
C *(page 379)*	Percent min/max

A	Expression

A point ID or expression that will determine the animated .gif speed.

	Opens the Edit Expression dialog box.
	Displays a Point Popup menu.

Note: When the **Expression** field is blank, the animated **.gif** runs at its original speed during runtime.

B	Expr. min/max

The min and max expression values define the lowest and highest value that the animation will respond to. Values between the min and max will cause the animation to speed up or slow down.

Min

Minimum value of the point ID or expression.

This value or lower will cause the object animation speed to be the lowest calculated speed.

Max

Maximum value of the Point ID or expression.

This value or higher will cause the object animation speed to be the highest calculated speed.

C	Percent min/max

The min and max percents define the lowest and highest animation speed as a percent of the original animated .gif speed.

The animation percent speeds between the min and max represent the expression value relative to the **Expr. min/max** values.

Min

Percent of the original animation speed that will display when the expression speed is equal to or lower than the expression minimum speed.

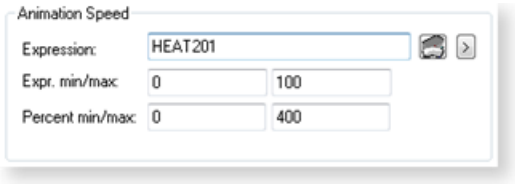
Max

Percent of the original animation speed that will display when the expression speed is equal to or higher than the expression maximum speed.

Runtime Example

A .gif file

- Displays a kiln with an animated fire
- Represents the kiln heat; the faster the animated fire speed, the higher the heat.
- Has 10 frames; each frame runs for 1 second; the 10 second speed will display when the expression value is 25.
- Is configured to use the default **Expr. min/max** and **Percent min/max** values.



Field	Min	Max
Expr. min/max	0	100
Percent min/max	0	400

During runtime, the animated .gif displays as follows.

Expr. Value	Percent Speed	Animation Speed*	Example
0	0%	0 sec.	

Expr. Value	Percent Speed	Animation Speed*	Example
12.5	50%	20 sec.	
25	100%	10 sec.	
50	200%	5 sec.	
100+	400%	2.5 sec.	

*****Animation Speed is seconds/10 frames.

7. Attribute Animation

7. Attribute Animation

There are several reasons to animate an object's color or text. Two very common applications are to:

- Show that a switch is on or off
- Provide clear warning if a value is in an alarm state

7.1 *(page 382)*	Animate the color/expression of an object.

7.2 (page 392)	Animate the visibility of an object.
7.3 (page 394)	Animate an object alarm state.
7.4 (page 395)	Color index for animation.
7.5 (page 396)	Shape Attribute Animation.

7.1. Animate the Color/Expression of an Object

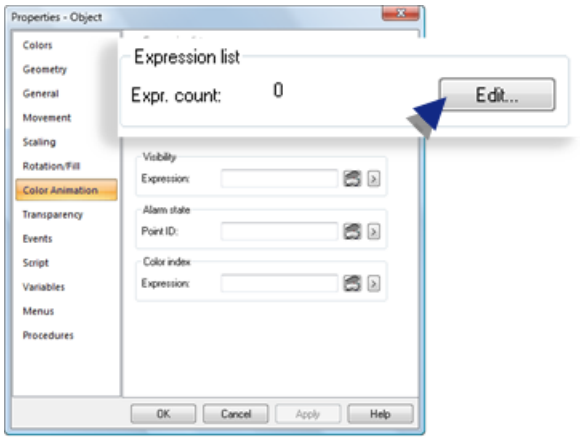
If you can fill an object with a color or pattern that changes during runtime based on a condition that is specified by an expression.

The number of expressions you can define for an object is limited only by the amount of system memory available.

- Open the Expression List Attribute Animation dialog box.
- Configure the animation list.
- Runtime example.

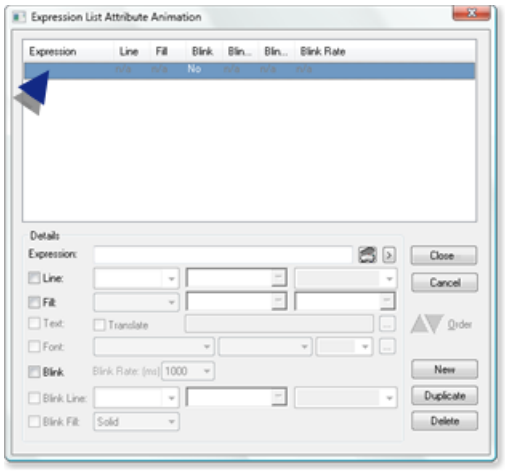
Open the Expression List Attribute Animation dialog box

1. Select the object that will be animated.
2. Open the object's Properties dialog box.
3. Select Color Animation.
4. Click Edit in the Expression list section.



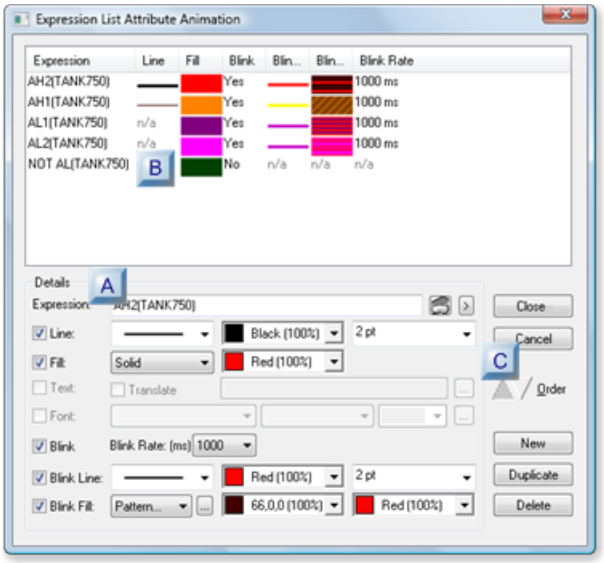
The Expression List Attribute Animation dialog box opens.

A first expression is available for configuration.



Configure the animation list

Use available options to configure the animation list.



rect 51, 164, 80, 193 *(page 384)*
rect 296, 207, 325, 236 *(page 388)*
rect 81, 88, 110, 117 *(page 387)*

A *(page 384)*	Details

B *(page 387)*	Expression list
C *(page 388)*	Expression list buttons

A	Details

Details that are available depend on the object selected.

Check a detail to enable its configuration selections.

Selections are:

Selection	Description
Expression	(Required) Condition required to display selected animation option(s); can include one or more point ID's and mathematical operators.
Line	Creates a border around the object when the expression evaluates to True. Line selections are: • Style • Color • Width The selected line displays when the expression evaluates to True.

Selection	Description
Fill	Changes the object fill to the selected fill when the expression evaluates to True. Fill selections are: • Style • Primary color • (Available when a pattern or gradient two colors is selected) Second color • (Available when gradient one color is selected) Brightness
	Style
	No fill
	Solid
	Pattern
	Gradient
Text	(Available when the object is a text object or text button.) Checked displays specified text string when the expression evaluates to True.
	Translate
	String
	...

Selection	Description
Font	Displays the selected font when the expression evaluates to True. Selections are. • Font • Type • Size **Note:** If you make the text available for translation make sure that the selected font accommodates the languages in which the string will display. Arial UNICODE MS can display all languages.
	...
Blink	The object alternates the blink configuration with the line/fill configuration to create a blink effect, when the expression evaluates to True.
	Blink Rate (ms)
Blink Line	(Enabled when **Blink** is checked) adds a border around the object when the blink configuration displays. This configuration alternates with the line/fill configuration to create a blink effect. Selections are line: • Style • Color • Width

Selection	Description
Blink Fill	(Enabled when **Blink** is checked) displays the selected configuration when the blink configuration displays. This configuration alternates with the line/fill configuration to create a blink effect. Fill selections are: • Style • Primary color • (Available when a pattern or gradient two colors is selected) Second color • (Available when gradient one color is selected) Brightness
	Style
	No fill
	Solid
	Pattern
	Gradient

B	Expression list

A list of animation expressions display in the Expression box.

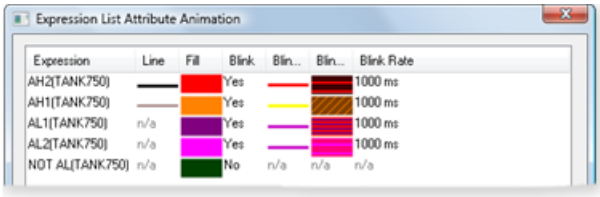
The item order in this list can be changed.

Important: Expressions in this list must be mutually exclusive. It is important to review it thoroughly to make sure there is no overlap or the animation will not work properly. The type of columns in the list depend on what object is selected.

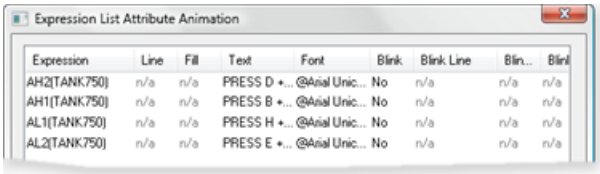
Example

An expression list for a:

- Rectangle contains line and fill specifications.



- Text object contains text specifications.



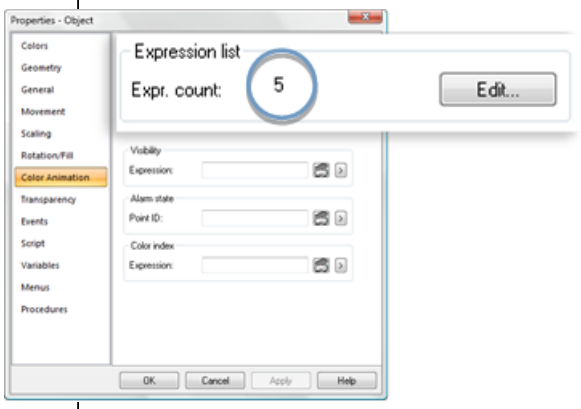
During runtime:

- The listed expressions are evaluated in the order they display in the list.
- The first condition that evaluates as TRUE or unavailable determines the object's color or text string. I
- If no conditions evaluate to TRUE the display reverts to values configured as placeholders in the Properties dialog box.

C	Expression list buttons

Buttons include:

Button	Description
Close	• Saves the expression animation list. • Closes the Expression List Attribute Animation dialog box.

Button	Description
	The **Expression Count** on the Color Animation tab will equal the number of expressions configured on the list. 
Cancel	Cancels expression item being configured.
Order	Moves the selected expression item in the list.
	Up
	B

Button	Description
	Displays a new line at the end of the list to configure a new expression item. Once the new line is created either the: • Expression must be configured or • Item must be deleted.

Button	Description
Duplicate	• Duplicates the selected expression. • Places the duplicate expression at the end of the list. **Important:** The duplicate expression must be edited in order to adhere to the requirement that expressions be mutually exclusive.
Delete	Deletes the selected expression item.

Runtime Example

A button is configured to change appearance based on its configured expression in the Expression List Attribute Animation dialog box.

The expressions use the following point.

Point ID	Class
TANK750	Integer

Expressions in the list are as follows.

- AH2(TANK750)
- AH1(TANK750)
- AL1(TANK750)

- AL2(TANK750)
- NOT AL(TANK750)

During runtime operators must press specific keys on a keypad based on a TANK750 alarm state.

The following objects contain expression animation to alert operators and insure that they press the correct keys.

Default Configuration	AH2 Configuration	AH2 Blink Configuration

Item	Description
A	Text (text object) reports what keys to press based on the alarm state expression that evaluates to True.
B	Text background (rectangle object) changes color and blinks based on the alarm state expression that evaluates to True.
C	Keys that must be pressed change color when their alarm state expression evaluates to True.
D	Text (text object) that reports the TANK750 point value changes color based on the alarm state expression that evaluates to True.

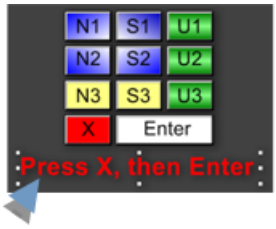
7.2. Animate the Visibility of an Object

- Configuration.
- Runtime example.

Configuration

Visibility animation lets you display or hide screen elements based on the evaluation of an expression.

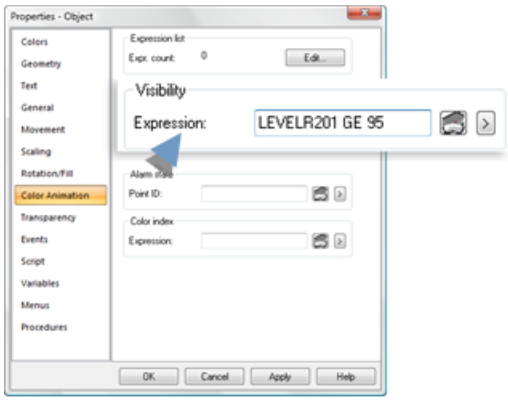
1. Select the object that will or will not display.



2. Open the object's Properties dialog box.

3. Select Color Animation>

4. Enter an expression in the **Expression** field in the **Visibility** box.



The expression can be one or more point ID's and mathematical operators that will be evaluated during runtime as follows.

Evaluation	The object is:
TRUE	Visible. Users can execute any procedures assigned to the object.
FALSE	Invisible Users cannot access procedures associated with the object.

5. Click Apply.

CimEdit will evaluate the expression during runtime.

Runtime Example

An arrow is configured to be visible or invisible base on the configured **Visibility** expression.

6. The expression uses the following point.

Point ID	Type	Class
N2B	Virtual	Boolean

7. Expressions in the list are as follows.

- N2B EQ 1

Note: The point N2B must equal 1 for the selected object to be visible.

Runtime Example

An operator must press X+Enter on an on screen keypad when a point (LEVELR201) value is equal to or greater than 95.

The expression was entered for instruction text in the Visibility>**Expression** field.

During runtime:

- The point value is equal to or greater than 95.

Instruction text displays on the screen.

- The point value is less than 95.

Instruction text does not display on the screen.

7.3. Animate an Object Alarm State

Alarm State animation provides shorthand for commonly used expression annunciation. When you specify a Boolean or analog point ID, the object will annunciate as follows.

Point Type	Color of the object will change when the value of the point ID:
Boolean	Changes from 0 to 1.

Analog	Is in one of the four alarm states: • Alarm High • Warning High • Warning Low • Alarm Low

Enter the Point ID to associate with the object in the **Alarm State Expression** field of the Color Animation tab. You can configure the colors and blink codes for Default Annunciation in the global parameters file.

Global parameters are:

- GSM_ANNUN_ALARM_H1
- GSM_ANNUN_ALARM_H2
- GSM_ANNUN_ALARM_L1
- GSM_ANNUN_ALARM_L2

If no global parameters are defined, the following defaults are used:

Boolean Points
Color
7
6
Analog Points
Color
15
14
10
13

Note: The colors used for Alarm State color animation are those defined by the RGB.dat file.

7.4. Color Index for Animation

Color Index animation lets you change the color of an object or group of objects to represent the current value of an expression under the following conditions

Expression Value	Color of the object
Ranges from 0 to 255	Will change
Is not in the range of 0 to 255	Will display in black as the unavailable color.

The colors displayed for the expression value correspond to the 256 colors in the RGB.dat file.

Enter an expression that can be evaluated in the **Color Index Expression** field of the Color Animation tab. The expression can consist of one or more point IDs along with mathematical operators.

7.5. Shape Attribute Animation

Shape attributes are common attributes for text and shape objects. Using shape attributes, you can create animations for objects at run time by configuring expression functions.

You can specify other expression functions as inputs for these functions. This gives you the flexibility to customize the attributes and create complex run time animations for objects.

Shape attribute	Description	Expression functions supported
Background fill	Changes the background fill of an object at run time by using the supported functions in an expression.	• CreateSolidFill • CreatePatternFill • CreateOneColorGradientFill • CreateTwoColorGradientFill
Foreground fill	Changes the foreground fill of an object at run time by using the supported functions in an expression. Note: This attribute is used by fill animation to fill an object with another color.	• CreateSolidFill • CreatePatternFill • CreateOneColorGradientFill • CreateTwoColorGradientFill
Fill mode	Changes the fill mode of an object at run time by using the CreateFillMode function in an expression. Note: To use this attribute for an object, configure fill animation.	CreateFillMode

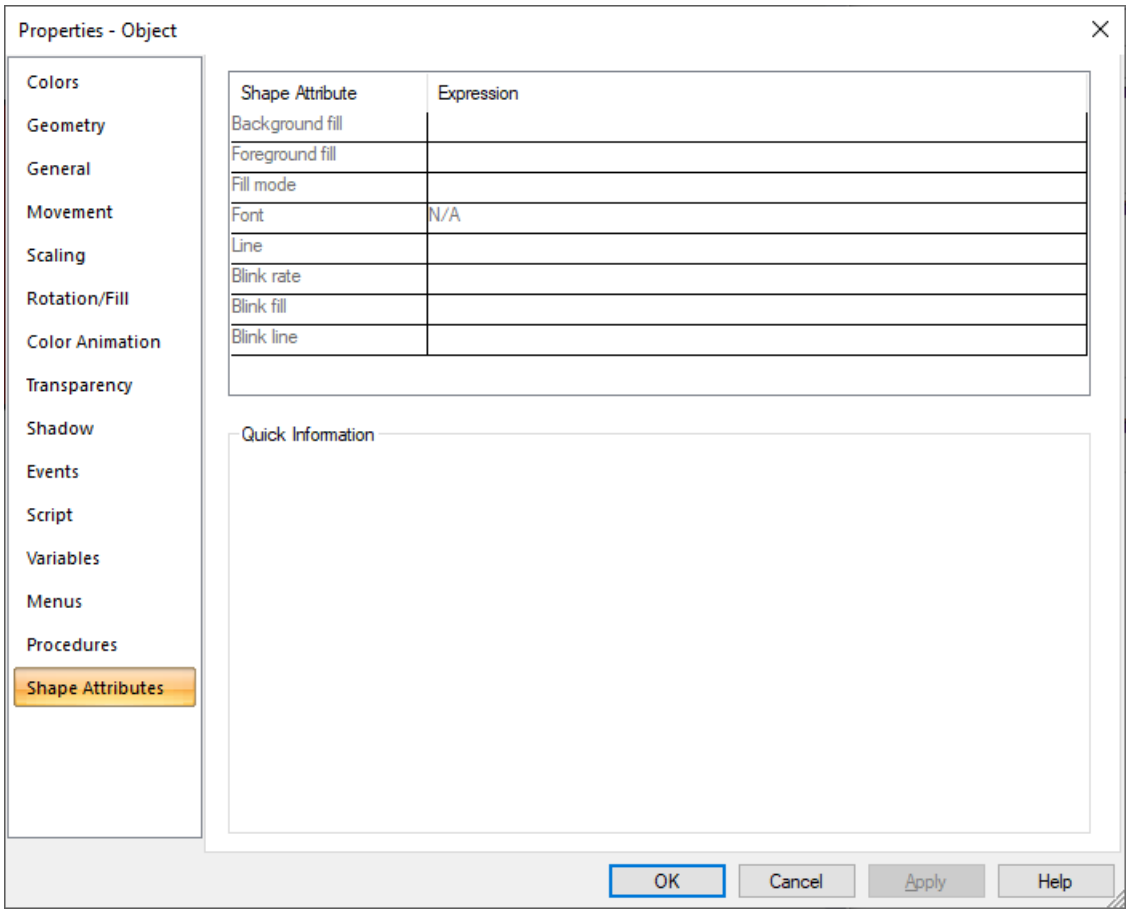
Shape attribute	Description	Expression functions supported
Font	Changes the font and its related properties for text and text button objects at run time by using the CreateFont function in an expression.	CreateFont
Line	Changes the line properties of an object at run time by using the CreateLine function in an expression.	CreateLine
Blink rate	Changes the blink animation rate of an object at run time by using the CreateBlinkRate function in an expression. Note: If an object does not have a blink animation, the CreateBlinkRate function creates a new blink animation.	CreateBlinkRate
Blink fill	Changes the blink fill animation attribute of an object at run time by using the supported functions in an expression. Note: If an object does not have a blink fill animation, the supported functions create a new blink fill animation.	• CreateSolidFill • CreatePatternFill • CreateOneColorGradientFill • CreateTwoColorGradientFill
Blink line	Changes the blink animation line attribute of an object at run time by using the CreateLine function in an expression.	CreateLine

Creating animations using Shape Attributes

To create animations for objects at run time using shape attributes:

1. Access CimEdit and create a shape or text object.
2. Right-click the object and select Properties. The Properties - Object window appears.
3. Select the Shape Attributes tab. The list of shape attributes appears.

Note: The shape attributes that are not applicable to the object are disabled.



4. For the shape attribute that you want to configure, select the Expression box, and then build the expression. For more information on building expressions, see 1. Build Expressions. *(page 271)*
5. Repeat this step for each shape attribute that you want to configure.
6. Select Apply, and then select OK. The animations are applied to the object in run time.

Fallback for Shape Attributes

There are two scenarios for a shape attribute where a fallback occurs:

1. Invalid expression: When an invalid expression is provided, the object is filled with black.
2. Valid expression with invalid parameters: When a valid expression contains parameters that are incorrect or beyond the specified range, the object reverts to the manually set parameters.

8. Text Value Animation

8. Text Value Animation

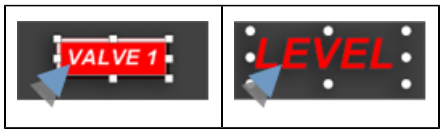
This task describes:

- Text value animation overview
- Text button text configuration
- Text object text configuration

Text Value Animation Overview

You can select text on a text button or a text object or text to:

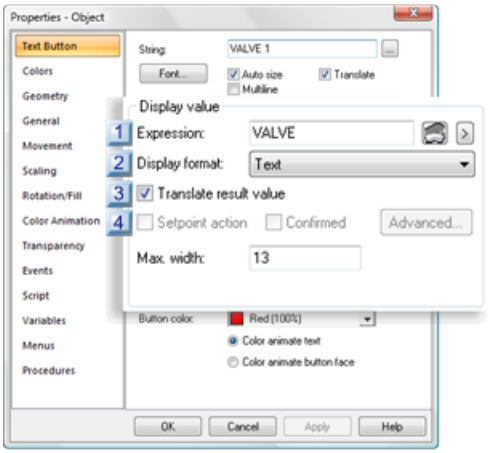
- Display current CIMPLICITY point, variable and expression values
- Set CIMPLICITY point values.



The entry options are the same for each.

Text Button Text Configuration

Select **Text Button** in a Properties dialog box. Entries in the **Display value** box direct the animation.

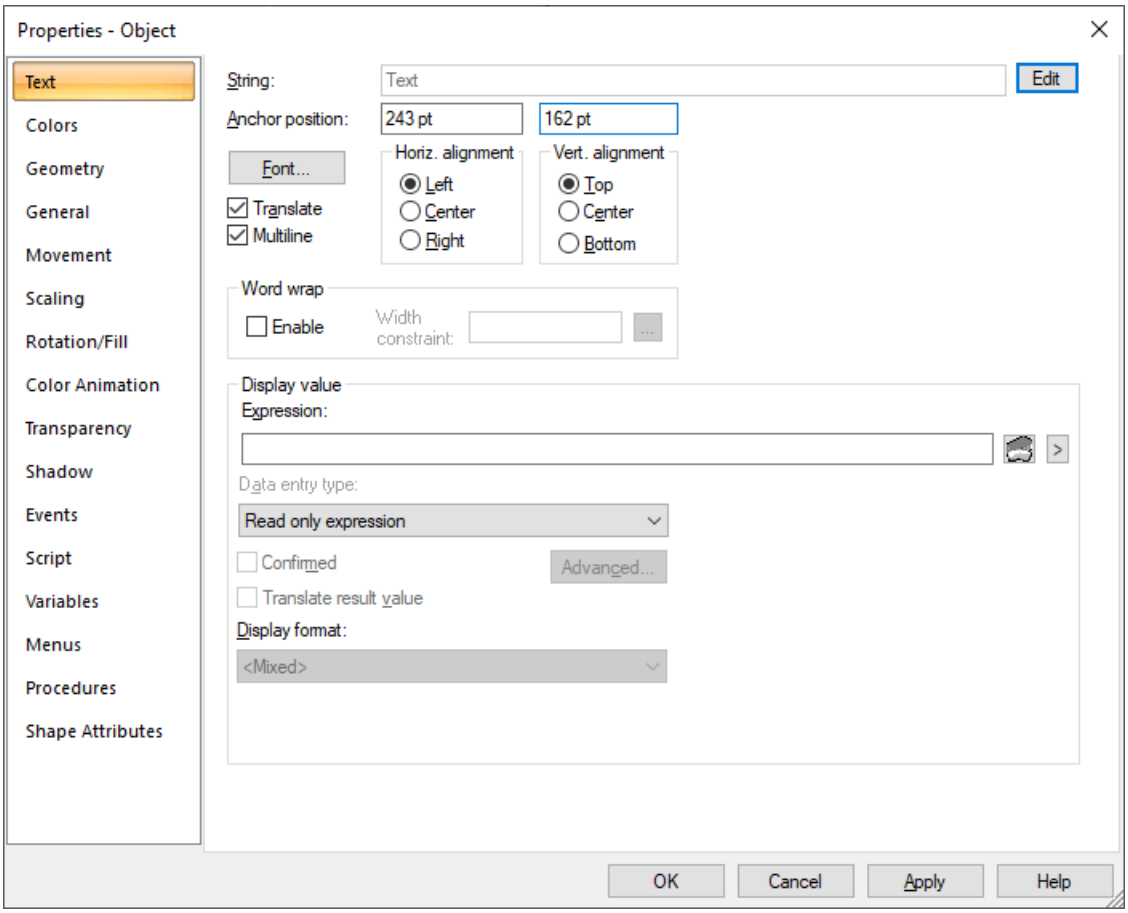


rect 63, 70, 85, 90 *(page 401)*
rect 59, 127, 85, 147 *(page)*
rect 59, 108, 87, 128 *(page)*
rect 61, 89, 85, 109 *(page 401)*

Text Object Text Configuration

Select **Text** in a Properties dialog box.

Entries in the **Display value** box direct the animation.



rect 63, 106, 85, 126 *(page 401)*
rect 62, 165, 88, 185 *(page)*
rect 62, 143, 90, 163 *(page)*
rect 63, 124, 87, 144 *(page 401)*

A *(page 401)*	Expression
B *(page 401)*	Data entry type
C *(page 401)*	Confirmed
D *(page 401)*	Translate result value
E *(page 401)*	Display Format

A	Expression

An expression *(page 272)* can include one or more CIMPLICITY Point IDs combined with logical or arithmetic operators.

B	Data entry type

The Data entry type check box contains options for displaying the expression, variable, or point in CimView.

The options are:

- **Read only expression:** Select this option to display an expression value in the read-only format during run time in CimView.
- **Read write point:** Select this option to display a point value and modify text inline during run time in CimView. After the inline text is modified, a setpoint operation is performed.
- **Read write variable:** Select this option to display a variable value and modify text inline during run time in CimView. After the inline text is modified, the variable is assigned a new value

C	Confirmed

If you select Confirmed when the Data entry type is set to Read write point or Read write variable, a confirmation message appears during run time before the accompanying action is performed. This allows you to cancel the setpoint or variable write action.

D	Translate result value

When you select the Translate result valuecheck box,known text values including literal and enumeration point values will be emitted to a selected CIMPLICITY Language Mapper file.

The value can be translated or revised for selected languages in the CIMPLICITY Language Mapper file. The translation or revised text is displayed in CimView when the assigned language is selected.

For more information on translation, see Translation *(page 66)*.

Note: The Translate result value check box is disabled when the Data entry type is set to Read write point or Read write variable.

E	Display format

Options for the format that the expression value displays in on the CimEdit screen are available in the **Display format** drop-down list.

- Configured
- General
- Custom
- Integer
- Real
- Text
- Time (Absolute, Seconds
- Time (Absolute, Sub Seconds)
- Time (Relative, Seconds)
- Time (Relative, Sub Seconds)

Configured

Settings that are entered on the General and View tabs in the Point Properties dialog box are used as the display format for point values in CimView, when you select **Configured** in CimEdit.

A width is specified in the Point Properties dialog box. This specification does not cause a value to be truncated. If the number of characters in the output value is greater than the specified width, or if a width is not given, all characters of the value are printed (subject to the precision specification).

If the number of characters in the output value is less than the specified width, blanks are added to the left or right of the values until the minimum width is reached. The blanks' position depends on whether the flag (for left alignment) is specified. If the width is prefixed with 0, zeros are added until the minimum width is reached.

If you configure a text object using this option's features, make sure that you make the text object long enough to accommodate the anticipated length of the value.

General

If the number of digits exceeds the **GSM_EXPONENT_PRECISION** global parameter, then the number is displayed in scientific notation. If **GSM_EXPONENT_PRECISION** is not defined, the default number of digits is 6. If the number is larger than the precision of the format, it will be rounded in the display. For example, with the default precision:

1234567 displays as 1.23457e+6

7654321 displays as 7.65432e+6

Custom

Text objects in CimView have a Custom Display Format that can be used (along with C style `'printf()'` specifiers) to change the behavior of the display.

This can be useful in modifying the format instead of the actual data to display it in a common format.

Example: Displaying a `%` sign preceding the actual value or converting the value such that it displays in a hexadecimal representation at runtime can be useful.

Format Specifiers List

Following is a simplified list of what format specifiers can be used.

Item	Type	Format
%d	Signed decimal integer	
%u	Unsigned decimal integer	
%f	Decimal floating point number	
%o	Octal representation	
%x	Unsigned hexadecimal integer	Lower Case
%X	Unsigned hexadecimal integer	Upper Case
%e	Scientific notation (mantissa/exponent)	Lower case
%E	Scientific notation (mantissa/exponent)	Upper case
%c	Character	Displays the equivalent ASCII character
%%	Displays a % sign in the field	Usually used as a suffix (for example, 25%)

Typically the format specifiers are used in the following notation:

```
%[flags][width][.precision]specifier
```

Custom Display Format Procedure

Do the following to use a Custom display format.

1. Add a new text object to a CimEdit Screen.
2. Right-click the text object
3. Select Properties on the Popup menu. A Properties - Object dialog box opens.
4. Select **Text**.
5. Change the **Display Value>Expression** to be the name of the point that contains the value to display.
6. Select Custom on the **Display Format** dropdown menu.

Result: This now allows for the user to enter information in the **Format Spec** field.

Custom Format Examples

Following are examples for the **Custom Format Specifier** field and their displayed values:

Item	Value
Point Value	123.45
Format Spec	`%d`
Displayed	123
Point Value	50
Format Spec	`%05d`
Displayed	00050
Point Value	123.45
Format Spec	`%3.1f` **Note** `3` represents the width `1` represents the precision (number of digits after the decimal place)
Displayed	123.4
Point Value	50
Format Spec	`%o`
Displayed	62
Point Value	123
Format Spec	`%x`
Displayed	7b
Point Value	123
Format Spec	`%X`
Displayed	7B
Point Value	123
Format Spec	`%#x` **Note:** `#` means precede the display with `0x`
Displayed	0x7B
Point Value	123
Format Spec	`%e`
Displayed	1.230000e+002
Point Value	123
Format Spec	`%E`
Displayed	1.230000E+002
Point Value	38

Item	Value
Format Spec	%c
Displayed	&
Point Value	50
Format Spec	%%%d
Displayed	50%

Integer

1. Enter, in the Width field, the number of characters needed to display the results of the expression evaluation.
2. Select the Zero filled check box to toggle the leading display of zeroes.
3. Select the Digit Grouping check box to add the thousands separator to the result of the expression evaluation.

Note: The Zero filled check box is disabled when you select the Digit Grouping check box.

Real

1. Enter, in the Precision field, the number of places to the right of the decimal place that you want displayed.
2. Enter, in the Width field, the number of characters needed to display the result of the expression evaluation.
3. Select the Zero filled check box to display leading zeroes.
4. Select the Scientific check box to use scientific notation for the value display.
5. Select the Digit Grouping check box to add the thousands separator to the result of the expression evaluation.

Note: The Zero filled and Scientific check boxes are disabled when you select the Digit Grouping check box.

Text

Enter, in the **Max width** field, the maximum number of characters needed to display the text string.

Time (Absolute, Seconds)

Interprets the expression as a UNIX absolute time. That is, the value represents the number of seconds since 00:00 GMT on January 1, 1970

Select the **Time format** you want to display.

Options are as follows.

- h:mm:tt
- HH:mm
- HH:mm:ss
- M/d/yy
- MM/dd/yy
- MM/dd/yyyy HH:mm:ss
- YYYY:MM:DD:HH:MM:SS
- yyyy:MM:dd:hh:mm:ss

You can choose to display time, date, or both.

If you choose the YYYY:MM:DD:HH:MM:SS format, you can only display values after 1995:01:01:00:00:00 (January 1, 1995 at 00:00 AM). Values before that date are displayed as ----:--:--:--:--:--.

Time (Absolute, Sub Seconds)

Interprets the expression as a UNIX absolute time. That is, the value represents the number of seconds since 00:00 GMT on January 1, 1970

Select the **Time format** you want to display.

Options are as follows.

- h:mm:tt
- HH:mm
- HH:mm:ss
- M/d/yy
- MM/dd/yy
- MM/dd/yyyy HH:mm:ss
- MM-dd-yyyy HH:mm:ss:TTT
- MM-dd-yyyy HH:mm:ss:TTTTTT
- YYYY:MM:DD:HH:MM:SS
- yyyy:MM:dd:hh:mm:ss

Note: TTT= milliseconds; TTTTTT=microseconds

Time (Relative, Seconds)

Interprets the expression value as an elapsed time value in seconds.

Select the Time format you want to display. You can choose to display time, date, or both.

Options are as follows.

- DD:HH:MM:SS
- HH.tenths
- HH:MM:SS
- MM.tenths
- MM:SS

Example

An expression has the value 2 000 000 000.

An `HH:MM:SS` format displays the entire string value as: `555555:33:20`.

Where

555555 = The entire `HH` value.

33 = `MM`

20 = `SS`

`(555555*60 + 33)*60+20` = 2 000 000 000 seconds

Time (Relative, Sub Seconds)

Interprets the expression value as an elapsed time value in subseconds (e.g. milliseconds or microseconds)

Select the Time format you want to display. You can choose to display time, date, or both.

Options are as follows

- DD:HH:MM:SS
- DD:HH:MM:SS:TTT
- DD:HH:MM:SS:TTTTTT
- HH.tenths
- HH:MM:SS
- MM.tenths
- MM:SS

Note: TTT= milliseconds; TTTTTT=microseconds

- If you use a point with Custom conversion, the EU value of the point is a real (floating point) number and is best displayed with the General or Real format. These formats round floating-point numbers, while the Integer format truncates them.

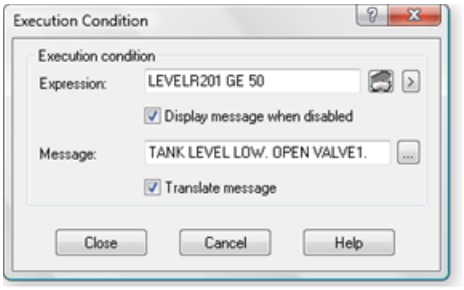
- For any of the Time display formats, if you wish to display text along with the expression, enclose each character of the text with single quotation marks. For example, to display the text as TTT ms, enter the text as TTT 'm"s'.

Execute Condition for a Text Setpoint

Click the Advanced button when a setpoint action is enabled for a text object's point ID expression.

Configure an Execution Condition if you want to control a user's access to a setpoint action.

Options are as follows.



Option	Description
Expression	One or more point ID's and mathematical operators that can be evaluated. The expression must evaluate to TRUE during runtime in order for a user to access the Setpoint action.

Option	Description
Display message when disabled	(Available if an expression is entered in the Expression field) Displays the entry in the message during runtime when a user attempts to perform the Setpoint action and the Execution Condition Expression evaluates to FALSE.
Message	String that displays in a Message box during runtime when a user attempts to perform the Setpoint action and the Execution Condition Expression evaluates to FALSE.

Option	Description
Translate message	The message text string can be translated or revised for selected languages in the CIMPLICITY Language Mapper file. The translation or revised text display in CimView when the assigned language is selected. For more information on translation, see Translation *(page 66)*.

9. Frame Animation

9. Frame Animation

Frames can be a time saving configuration choice if you have a section of a screen that will change considerably during runtime, based on a configured set of conditions.

The frame concept in CimEdit is based on frames that are created for a film. You create the frames in a frame container. While you work in an open Frame Container, you can select objects in the current frame, but you cannot select other objects on the screen.

As you configure frames, you can move forward and backward to view and edit the different frames. During runtime, at any given moment, the frame with the conditions that are TRUE will display.

You may create any object in a Frame Container that you can create on your screen.

Step 1 *(page 411)*	Open a frame container.
Step 2 *(page 414)*	Create new frames.
Step 3 *(page 415)*	Customize frame container names.
Step 4 *(page 420)*	Configure objects that are in frames.
Step 5 *(page 422)*	Configure frame animation.
Step 6 *(page 423)*	Configure frame container properties.

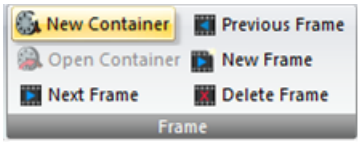
Step 1. Open a Frame Container

- Open a new frame container.
- Open an existing frame container.
- Frame container toolbar and menu items.

Open a new frame container

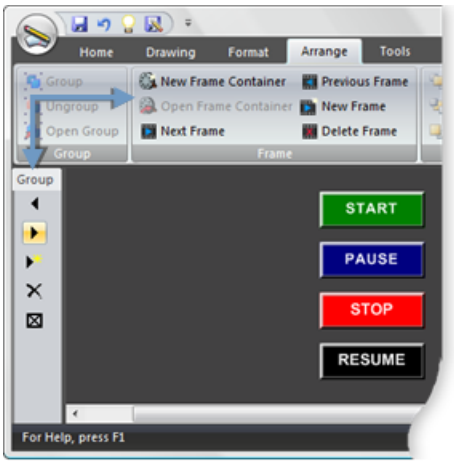
Choose one of the following methods.

Click New Container in the Arrange>Frame group on the CimEdit Ribbon bar.



Result: A new frame container opens displaying the Frame Container toolbar.

A Frame Container toolbar displays and corresponding Frame menu selections can be enabled.

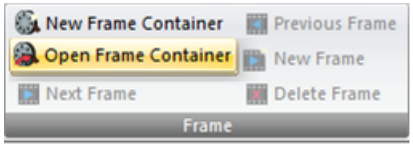


Open an existing frame

1. Select a frame container on the CimEdit screen.

   Note: The frame container size is determined by its widest frame. The handle's display that size.

2. Choose one of the following methods to open the frame container.
   - Click Open Frame Container in the Arrange>Frame group on the CimEdit Ribbon bar.



   - Right-click the container; select Open Frame Container on the Popup menu.

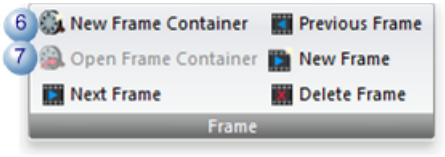
The selected frame container opens displaying the selected frame.

A Frame Container toolbar displays and corresponding Frame Ribbon bar selections can be enabled.

Frame container toolbar and menu items

A frame container toolbar displays when the frame container is open.

The toolbar and Frame menu selections enable the following actions during configuration.

	Frame Container Toolbar	Frame Menu
		
	Menu Item / Action	Comments
1	Previous Frame	Functions when there is more than one frame. The first frame goes to the last frame in the sequence.
2	Next Frame	Functions when there is more than one frame. The last frame goes to the first frame in the sequence.
3	New Frame	Creates a new frame in the container. The new frame is automatically placed at the end of the sequence.
4	Delete Frame	Deletes the active frame.
5	Close Frame Container	
Menu only		
6	New Frame Container	Enabled whether or not a frame container is open.

	Frame Container Toolbar	Frame Menu
7	Open Frame Container	Enabled when a frame container is available to be opened. Enabled with frame container actions when a frame container is open and there are more than one containers on the screen.

Step 2. Create New Frames

1. (For a new container) begin configuration as soon as you create the new container.

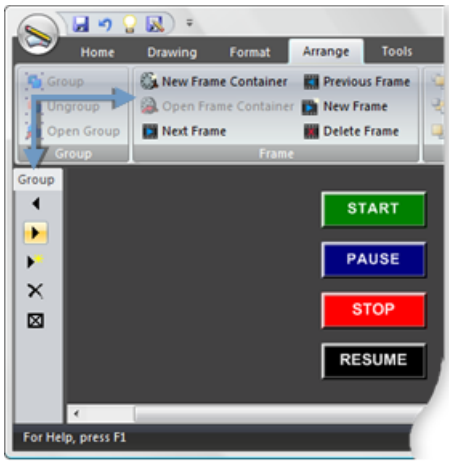
   The configuration is the first frame.

2. Do one of the following.

A	Click New Frame in the Arrange>New Frame group on the CimEdit Ribbon bar.
B	Click the New Frame button on the Frame Container toolbar.

An empty frame opens. The frame is automatically given an FAFrame<N>. This name can be changed.

Note: Objects created on the screen, not in other frames, continue to display.



3. Configure objects for the new frame.

   Note: The frame does not have to run in second place during runtime. The order can be changed at any time.

4. Continue configuring as many frames as you need.

   Tip: Use the Frame Container toolbar and Frame menu items to navigate through the frames.

Step 3. Customize Frame Container Names

Names are automatically assigned to the frame container and frames. It is recommended that you enter unique names to clearly identify each.

Note: While you are in an open frame you can name objects, groups and the open frame using the same procedures that you use when you are working on the CimEdit screen. However, you can also speed up your work and organize all of your frames through a single dialog box.

- Open the Properties - Frame Container dialog box.
- Rename the frame container.
- Rename frames.
- Rename objects.

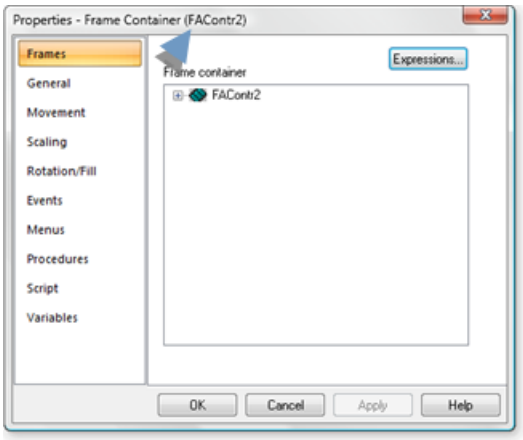
Open the Properties - Frame Container dialog box

Do one of the following.

- Double-click a closed frame container.
- Right-click a closed container; select Properties on the Popup menu.

- Right-click any location on the screen that is not in an open frame container; select Properties on the Popup menu.
- Click Properties in the Home>Properties group on the CimEdit Ribbon bar.

Result: A Properties dialog box for the frame container opens when you use any method.



Rename the Frame Container

1. Select the General tab in the Properties - Frame Container dialog box.

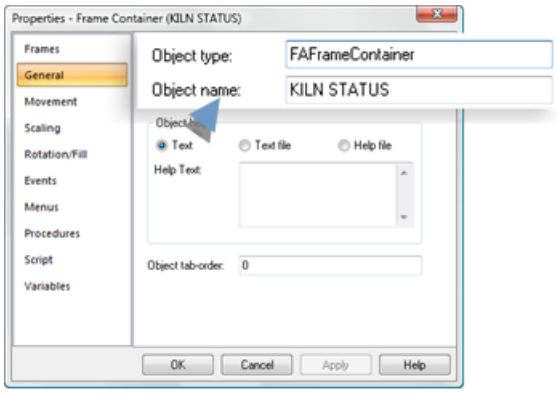
   The name that is initially in the is automatically generated, e.g. FAContr2.

2. An automatically generated name is in the **Object name** field.

3. Enter a unique name in the **Object name** field that will clearly identify the frame container.

4. Click Apply.

   Result: The frame container is renamed.

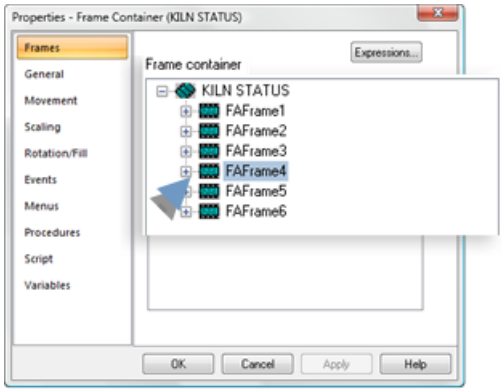


Rename Frames

Note: You can also use this method to rename the frame container.

5. Select the Frames tab in the Properties - Frame Container dialog box.

   The automatically generated names display in a list of frames.



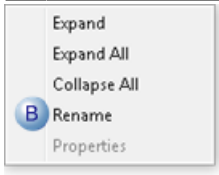
	Frame container	If a unique name has not been assigned on the General this procedure can be used to assign one.
	Frames	Automatically generated frame names.

6. Right-click a name in the list.

A	A box surrounds the frame that the name represents on the CimEdit screen.

Note: The box surrounds the frame as soon as it is selected.

B	A Popup menu opens in the frame's Properties dialog box.



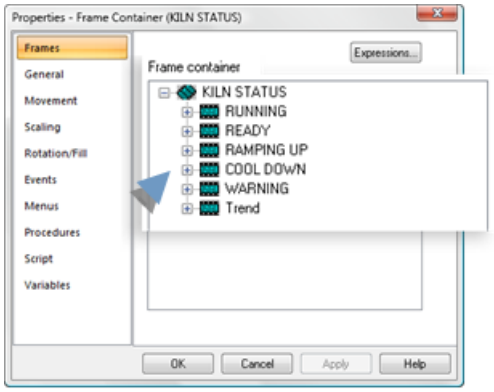
7. Select Rename on the Popup menu.

The **Name** field changes to read/write.

8. Enter a unique name that will clearly identify the selected frame.
9. Press Enter.

   A message displays asking you to confirm the name change.

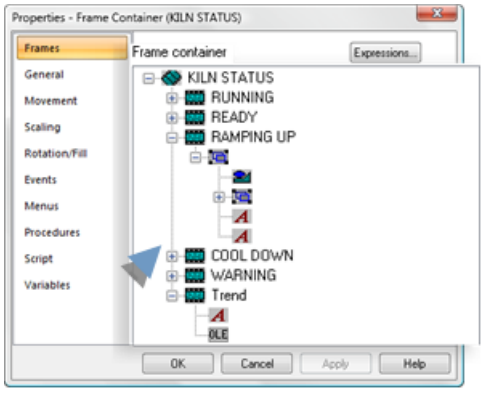
10. Click OK.
11. The frame is renamed.
12. Continue renaming any frames that should be renamed.



Rename objects

13. Expand the tree on the Frames tab.
14. Right-click one of the items in the tree.

Note: Item options include:



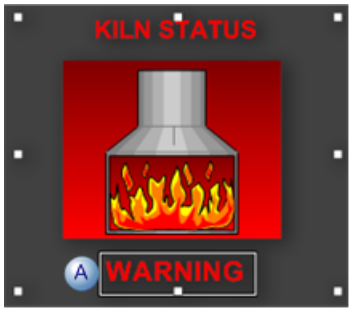
	Graphic object

	OLE
	Text
	Text button
	Group
	Line

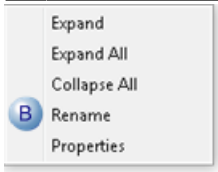
15. Right-click an object.

A	A box surrounds the object that the name represents on the CimEdit screen.

Note: The box surrounds the object as soon as it is selected.



B	A Popup menu opens in the frame's Properties dialog box.

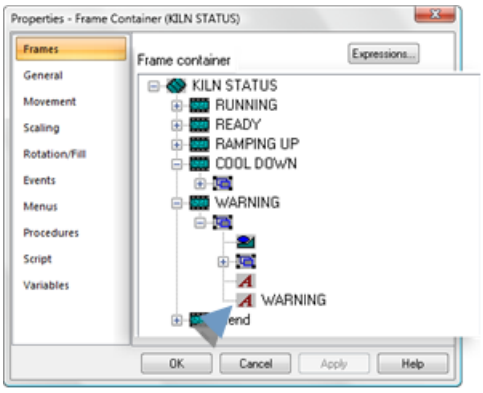


16. Select Rename on the Popup menu.
17. Rename the object.
18. Press Enter.

    A message displays asking you to confirm the change.

19. Click OK.

The item is renamed to the name you entered.



Step 4. Configure Objects that are in Frames

Configuration options are available for objects that have been placed in a frame.

Do either of the following to open the Properties - Object dialog box.

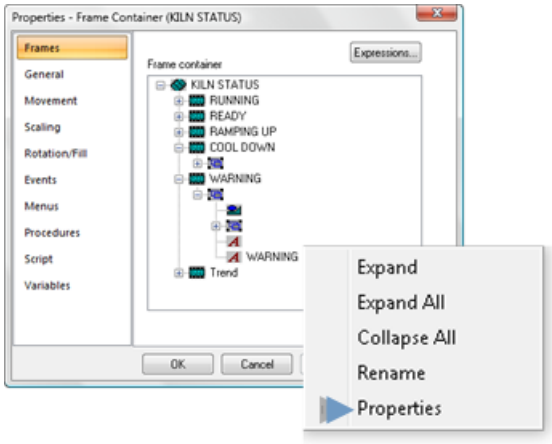
- Frame tab in the Properties - Frame Container dialog box.
- Standard.

Frame tab in the Properties - Frame Container dialog box

1. Open the Properties - Frame Container dialog box.
2. Select the Frames tab.
3. Expand the tree.
4. Right-click an item in the tree.

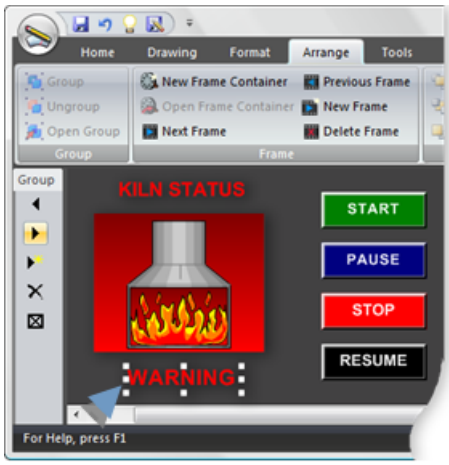
   Note: A box highlights the selected object on the CimEdit screen.

5. Select Properties on the Popup menu.



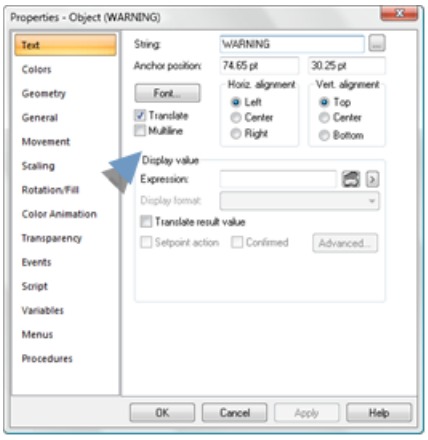
Standard

6. Open the frame that has the object.
7. Select the object as you would any object on a CimEdit screen.



8. Do either of the following.
   - Right-click the selected object; select Properties on the Popup menu.
   - Click Properties in the Home>Properties group on the CimEdit Ribbon bar.

The Properties - Object dialog box opens. Object configuration is available.



Step 5. Configure Frame Animation

1. Evaluation starts with the first frame/expression.
2. As soon as an expression evaluates to TRUE, the frame displays.
3. Evaluation begins again at the beginning of the list.
4. Expressions that use the following points are associated with frames in a KILN STATUS frame container.

Point ID	Type	Class
HEAT201	Device	Integer
TREND	Virtual	BOOLEAN
COOLDOWN	Virtual	BOOLEAN

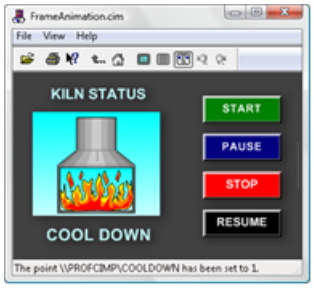
5. Expressions that are assigned to the six frames are as follows.

	Frame	Expression
1	COOL DOWN	COOLDOWN EQ 1
2	TREND	TREND EQ 1
3	READY	HEAT201 LE 25
4	RAMPING UP	HEAT201 GE 25 AND HEAT201LT 85
5	RUNNING	HEAT201 GE 85 AND HEAT201 LT 140
6	WARNING	HEAT201 GE 140

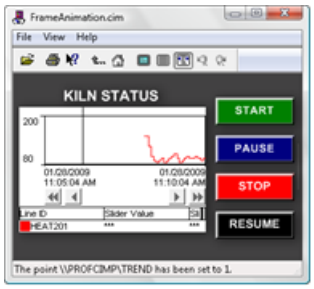
Result:

The expressions direct frames to display as follows.

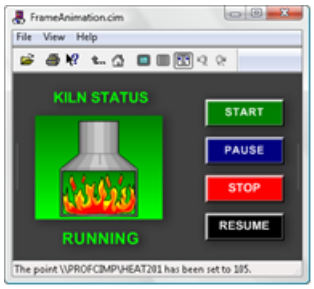
- COOL DOWN instantly displays when the point ID COOLDOWN equals 1.



- TREND displays when TREND equals 1 as long as COOLDOWN equals 0.



- The frame in the remaining four frames displays when its associated expression evaluates to True as long as COOLDOWN and TREND equal 0.



Step 6. Configure Frame Container Properties

- Appearance.
- Runtime behavior.

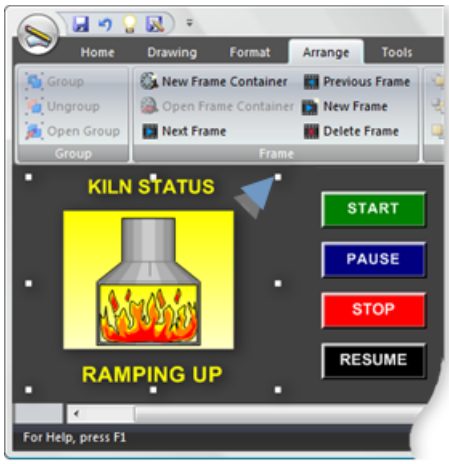
Appearance

1. Close the frame container.

   The container is now an object on the CimEdit screen.

2. Click any (sub)object in the frame container that displays when the container is closed.

   Handles outline the container object.



3. (Optional) use the toolbar buttons and menu selections to modify the object display the same as you do rectangle objects, including:

   Move

   Resize

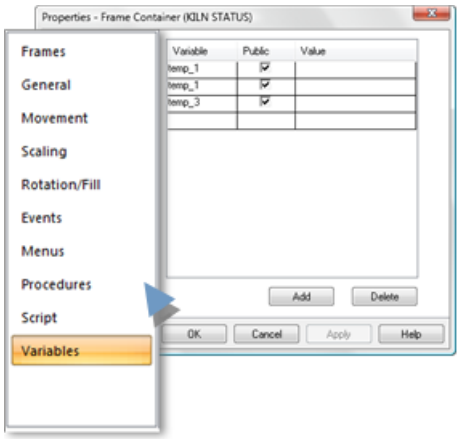
   Rotate

   Runtime Behavior

   The frame container has configuration options that are the same as options for other CimEdit objects.

   Open the Properties - Frame Container dialog box.

   Configuration that is available for the frame container includes the following.



   - Frames
   - General
   - Movement
   - Scaling

- Rotation/Fill
- Events
- Menus
- Procedures
- Script
- Variables

guide: Guidelines

- Configuration procedures are the same as they are for other objects.
- Where applicable, frame container behavior supersedes the behavior of objects in the frame.

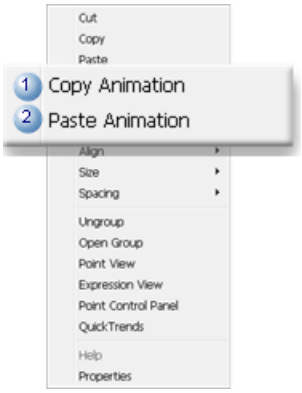
10. Copy/Paste Animation

Copy/paste animation enables you to configure one or more animations for any object, copy the animation configuration and paste it onto any other object on the same screen or a different screen.

- Copy/Paste Animation: Configuration
- Copy/Paste Animation: Notes.

Copy/Paste Animation: Configuration

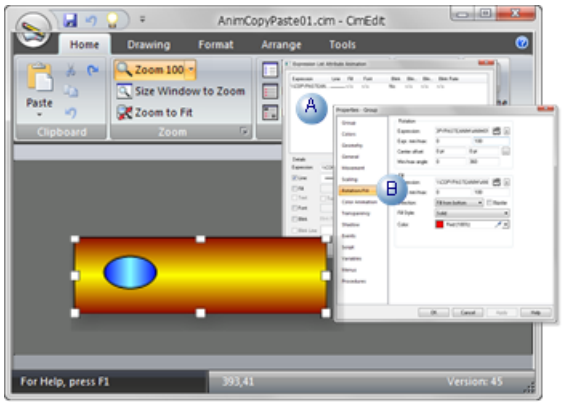
The CimEdit Popup menu includes two items, which are enabled as follows.



	Item	Enabled when:
A	Copy Animation.	A selected object is configured with one or more animations.
B	Paste Animation.	Copy Animation has been selected for an object on an open screen.

Example: Copy/Paste animation through the UI

1. A group object has the following animation configuration.

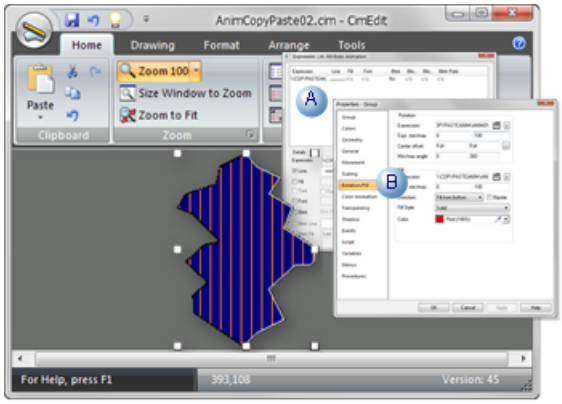


2. The object is right-clicked.
3. Copy Animation is selected on the Popup menu.

4. An object on a different screen is selected.

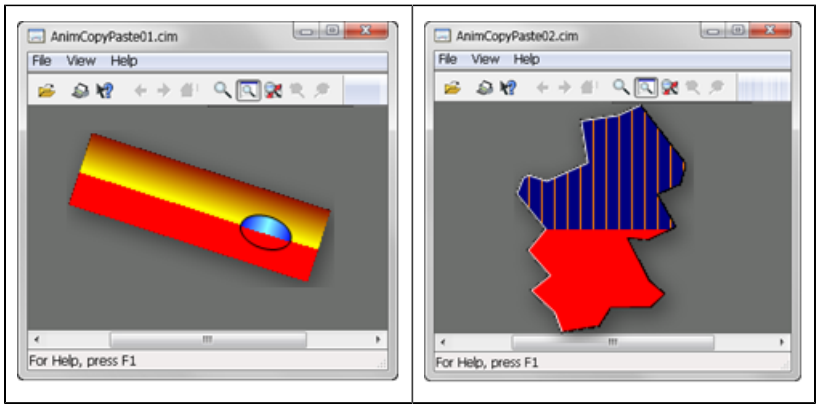
   Note: The object can also be on the same screen.
5. The target object is right-clicked.
6. Paste Animation is selected on the Popup menu.

7. The animations that were copied from the source object are now part of the target object's configuration.



A	List attribute animation.
B	Rotation and fill.

During runtime both the original and the target objects display the same runtime animation.



Copy/Paste Animation: Notes

- An object can be any object, including:
- Objects that are in a group.
- A group object.
- Objects that are different from the original object.
- Objects on the same or a different screen.
- When the screen with the source object is closed, copied animations can only be pasted into screens that were already open. If the source is closed and a new screen is then opened, Paste Animation will not be available for that screen.
- The Undo button can be clicked to remove the pasted animation. However, a single click may not remove the entire configuration that was pasted. Instead, **Undo** backtracks one step at a time, beginning with the most recent step in the configuration. The number of times the Undo button needs to be clicked in order to completely remove the pasted animation depends on what and how much was included.
- The following occurs if the target object has animations already configured (e.g. rotation, scaling, fill).

If the Target's Animation type is:	**The target object:**
Different from the copied animation types.	Retains its types of animation that are different from the copied animation types. Example If the target object has a rotation animation configured and a rotation has not been been copied, the target will retain its rotation animation.
The same as the copied animation types.	Animation types that are the same are overwritten. Example If the target has a fill animation configured and a fill has been copied, the target's fill configuration will be overwritten.

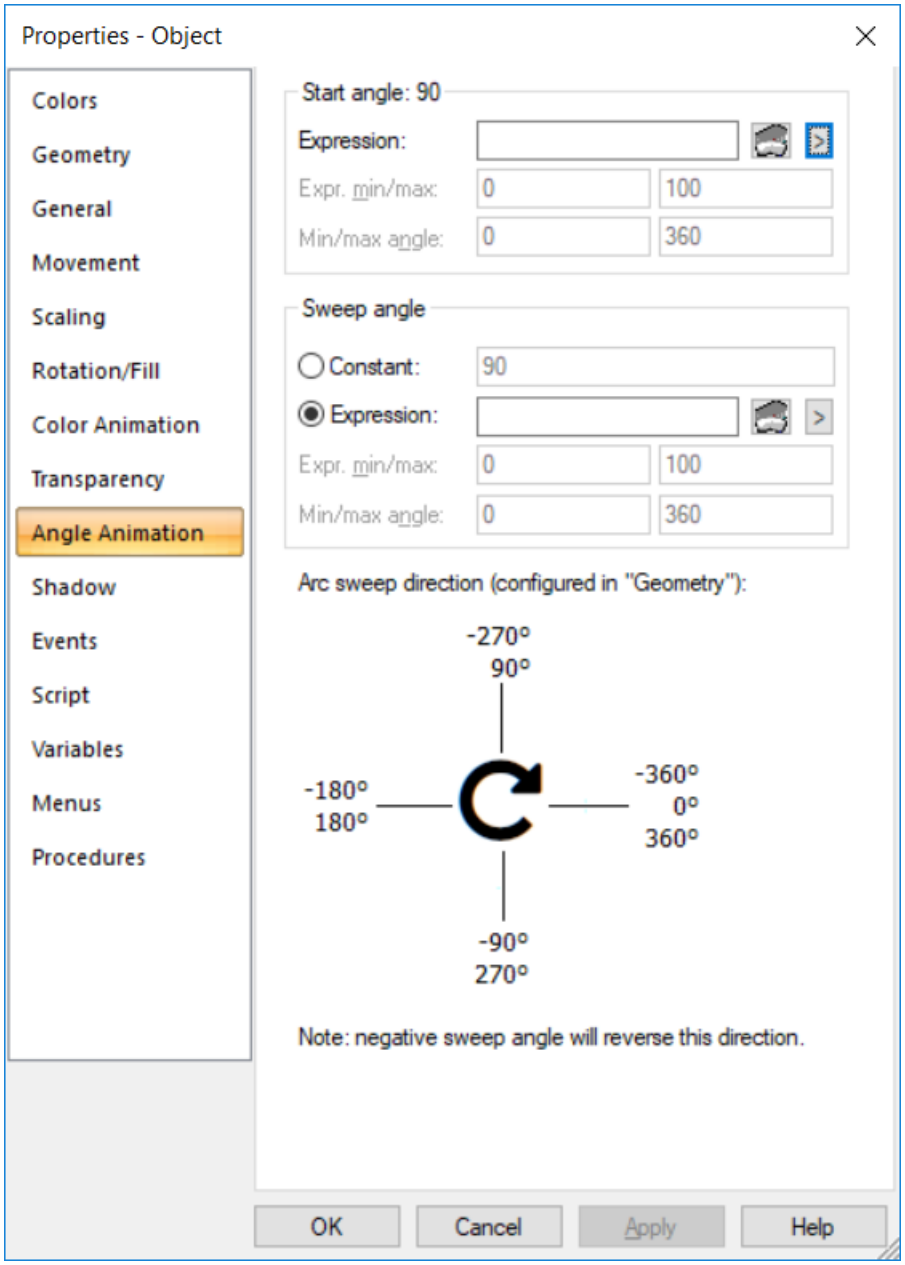
- `GefObject.CopyAnimations` (method) is an Object Model method to copy/paste animations.

11. Angle Animation

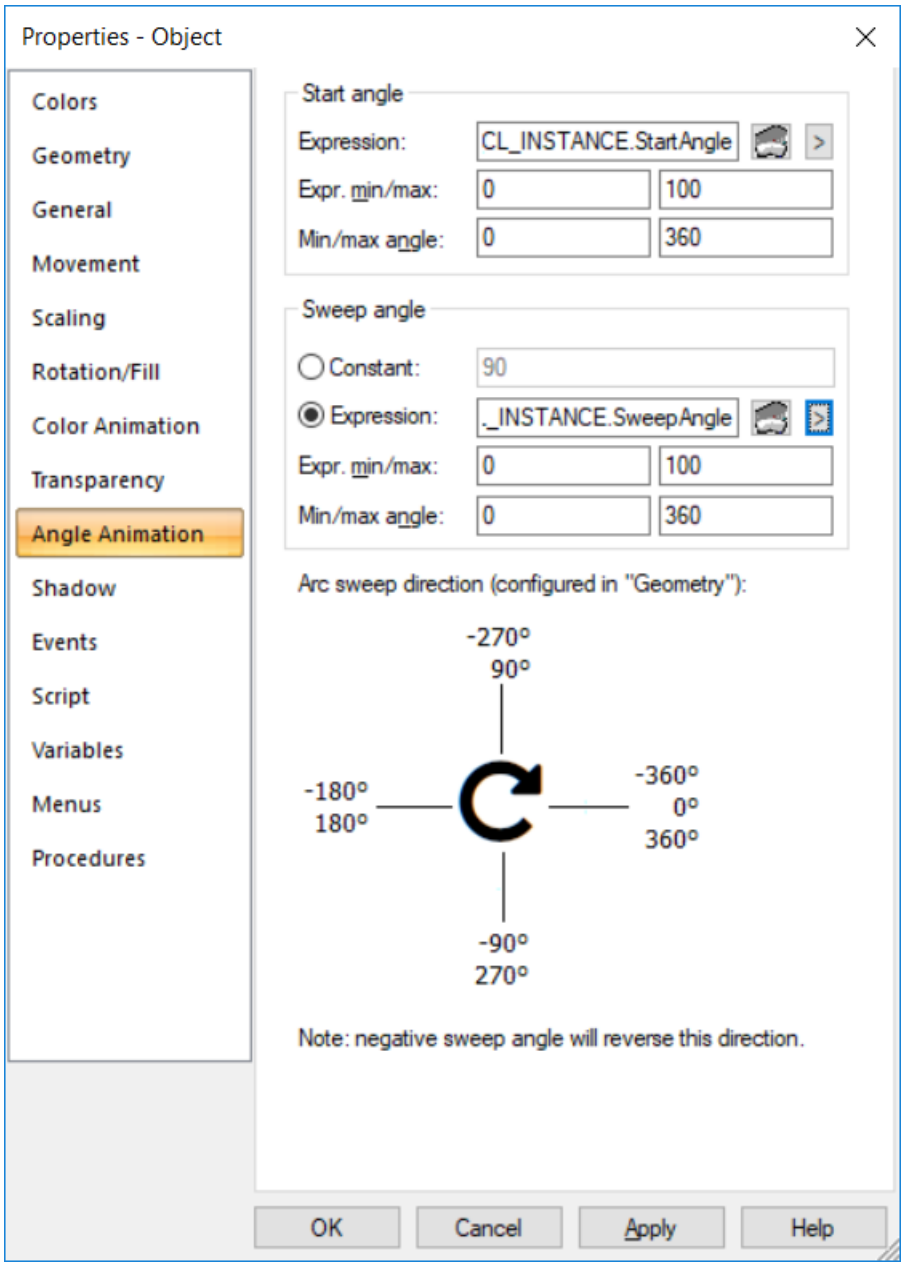
You can apply animation to angles of an arc, pie, or chord object. You can dynamically change the start and the sweep angles of an object based on the evaluation of expressions.

Animate Angles in an Object

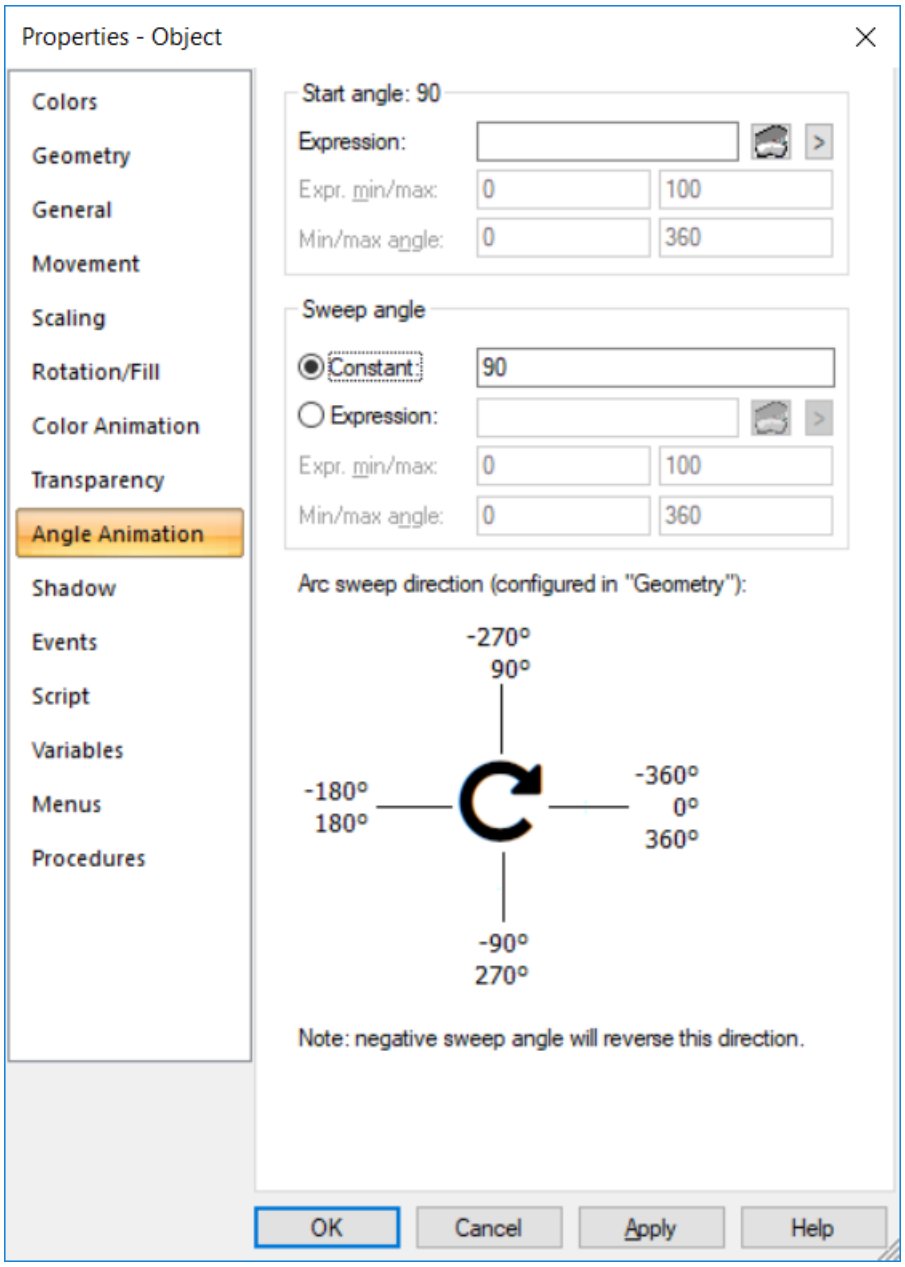
1. In the CimEdit window, select the object you want to animate, and then select Properties.
2. In the Properties window, select Angle Animation.



3. In the Expression box, enter an expression for the Start angle and/or Sweep angle. You can also modify the expression by selecting .



Note: Each expression determines the corresponding object angle locations between the Min and Max angle values. For Start angle, if you leave the expression blank, the start angle configured for the object (in the Geometry section) is used. For angle animation to occur, you must configure the sweep angle. If, however, you do not want to specify an expression for the Sweep angle, select the Constant option to enter a constant sweep angle.



4. In the Expr. min/max boxes, enter the minimum and maximum values for the expression.

   Note: You may leave the Expr. min/max boxes blank if the expression contains a single point ID with no operations performed on it and the point has Display Limits defined. CimView will use the Display Limits configured for the point ID for the Expr. min/max values when the screen file opens.

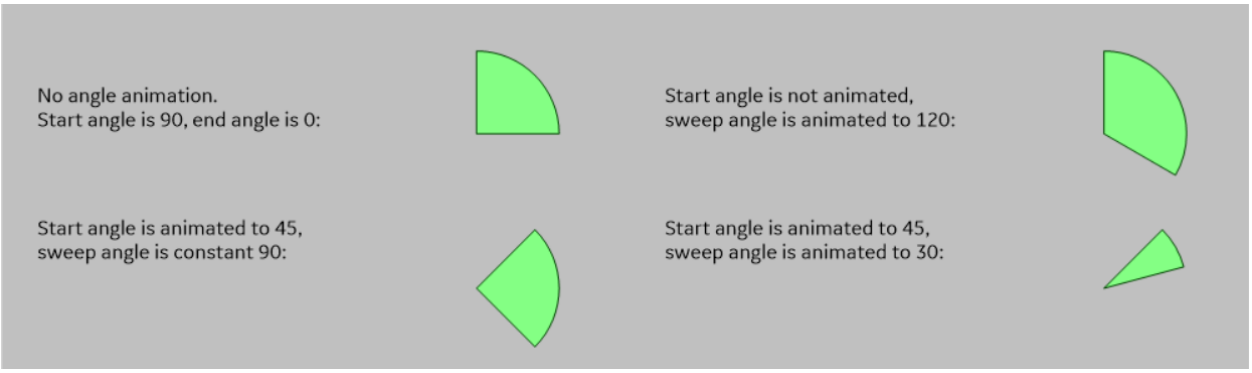
5. In the Min/max angle boxes, enter the minimum and maximum values for the angle. This will determine the start or sweep angle of the animation for the object.

When the expression value is equal to or less than the Expr. min value, the Min angle value will be used. When the expression value is equal to or greater than the Expr. max value, the Max angle value will be used.

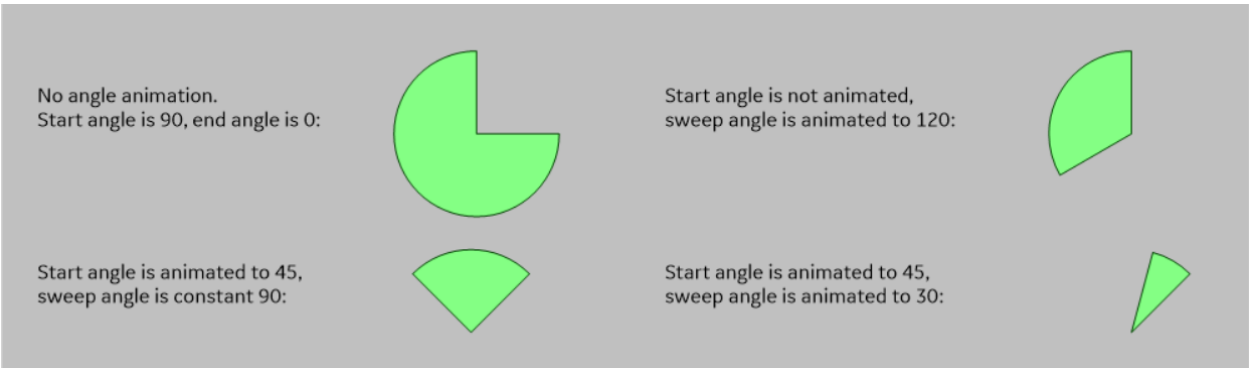
The direction of the underlying arc determines the direction of the animation. You can set the direction in the Geometry section of the Properties window.

Here are a few examples of the sweeps of an arc based on the direction, and the start and sweep angles:

Clockwise



Counter Clockwise



Events, Procedures and Scripts

Events, Procedures and Scripts

The triggers and actions that add the power to your monitoring and control system are:

- CimEdit events.
- CimEdit procedures.
- CimEdit scripts.

CimEdit Events

CimEdit Events

CIMPLICITY provides several tools to make things happen in CimEdit and in CimView. The tools can be categorized as triggers that are configured as events and results that can be configured through procedures or scripts.

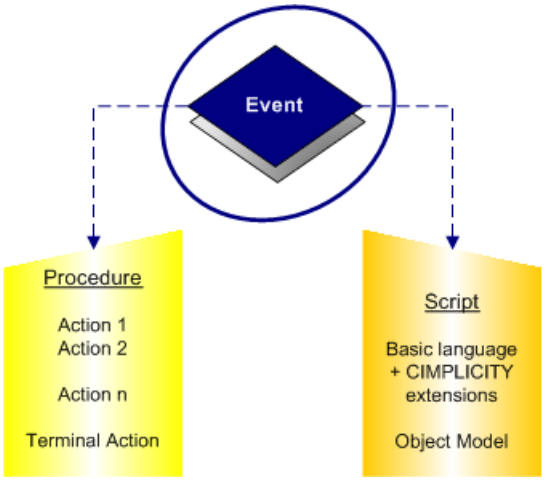
Trigger

An event triggers a procedure or calls a script. CimEdit provides a long list of events from which you can choose the best one for your requirements.

Result

A procedure *(page 482)* contains one or more actions that are triggered in the specified order when an event occurs and while the screen is displayed in CimView. CimEdit provides several actions from which a screen designer can easily compile a meaningful list.

A script *(page 530)* , which is usually written by a system administrator, uses the same Editor and Basic language as the Basic Control Engine. Anything you can do in a normal script, you can do in a CimEdit script. CimEdit provides additional extensions to give you a wider range of screen development choices. However, CimEdit scripts are only accessible from the screen in which you create them.



rect 7, 158, 107, 291 CimEdit Procedures *(page 477)*
rect 232, 156, 332, 289 CimEdit Scripts *(page 530)*

Event Configuration

Event Configuration

- Event configuration overview
- Event configuration
- Parameter field guideline

Event Configuration Overview

Configuring an event is straightforward. Because an event is the trigger that initiates a procedure (which includes one or more actions) or a script, configuration involves simply selecting the event and if there is an element involved with the event specifying that element. For example, for a **Key Up** event, the key needs to be selected.

When an event fires in CimView, its associated procedure (or script) is placed in the procedure queue. When the procedure (or script) that is at the head of the queue runs and finishes, the next procedure (or script) starts. The sequence continues until all of the procedures or scripts have run.

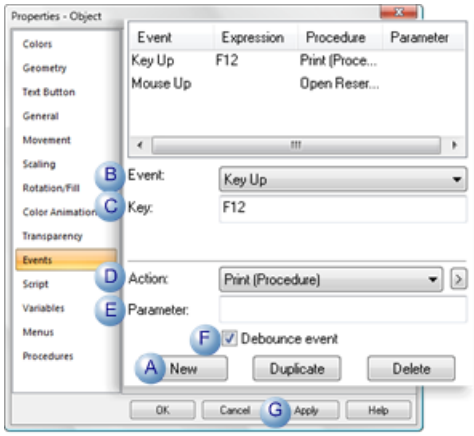
When you configure a procedure or script, you determine the order or the actions.

The number of procedures (or scripts) you can include in the queue is limited only by the amount of memory on the computer.

Event Configuration

You begin to configure an event by creating it on the Event tab of an object's Properties dialog box.

1. Open the Properties dialog box for a screen, group or object.
2. Select the Events section.
3. Do the following.



A	(If the event is not the first event in the list) click New to enable the **Event** field.
B	Select an **Event** in the drop-down list
C	Fill in fields that display based on the event selection.
D	Do one of the following to select an **Action**. • Select an action from the drop down list. • Create or edit one of the following. • Procedure *(page 482)* in the Procedures dialog box. • Script *(page 531)* in the Edit Script window.
R	(Optional) Enter a parameter.
F	(Check or clear) Debounce event.
G	Click Apply.

The event is applied to the CimEdit events list.

Parameter field Guideline

For any event you select, the Events tab will display a **Parameter** field.

If the procedure invokes a script *(page 530)* , you can use the **Parameter** field to pass a string to the script.

The script must use the `CimGetEventContext().UserParameter` property to accept the parameter.

Example: Events Triggering Actions

- You have:
- A CimEdit/CimView screen.
- One object.
- One group.
- You create a Screen Open procedure in each.
- You assign the same event to each. However, the event triggers actions that open different screens.

The type of event you assign affects when each action is triggered.

- Key Up event example
- Mouse Up event example

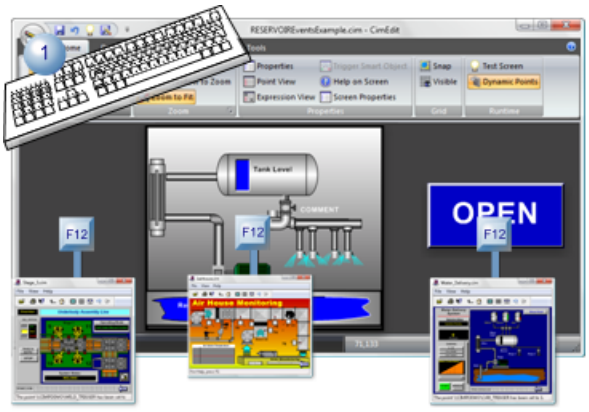
Key Up Event Example

1	CimEdit

You assign a Key Up event for key F12 to trigger an Open Screen action.

You attach the event to the screen, a group and an object.

The `Open Screen` action is set to a different screen for each, as follows.

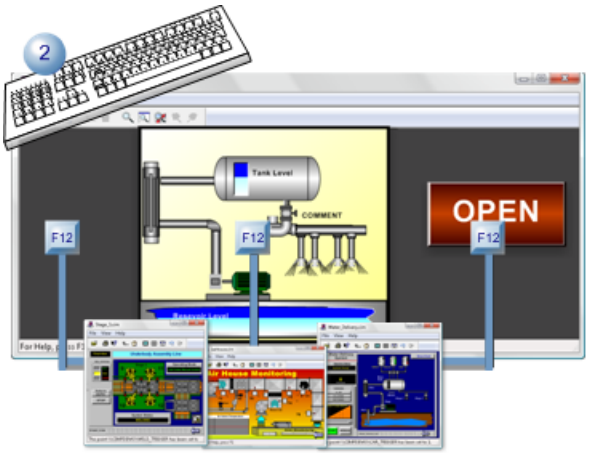


Selected	Screen Key Up F12 has the action:
Screen	Open Assembly Line
Group	Open Air House
Object	Open Water Delivery.

2	CimView

When a user presses then releases key F12 in CimView, all the screens open.

This is because the same key is selected for each event.



Selected	Key Up F12 opens
Screen	Assembly Line and Air House and Water Delivery
Group	Assembly Line and Air House and Water Delivery
Object	Assembly Line and Air House and Water Delivery

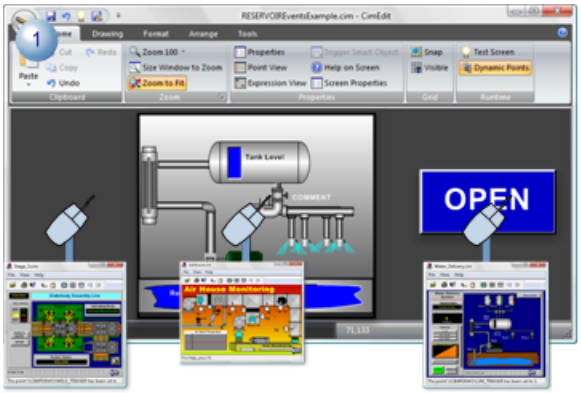
Mouse Up Event Example

1	CimEdit

You assign a Mouse Up event to trigger an Open Screen action.

You attach the event to the screen, a group and an object.

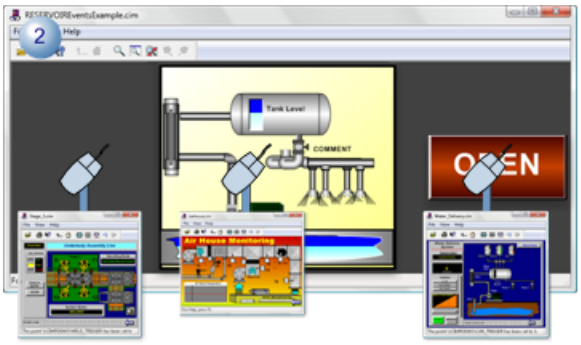
The `Open Screen` action is set to a different screen for each, as follows.



For	Screen Mouse Up is configured to trigger
Screen	Open Assembly Line
Group	Open Air House
Object	Open Water Delivery

2	CimView

When a user presses then releases the mouse in CimView, the screen that opens depends on what the mouse clicked.



For	Mouse Up opens
Screen	Assembly Line
Group	Air House
Object	Water Delivery

Available Events

Available Events

In CimEdit, you use an event to trigger a procedure (which contains an action or series of actions). If you associate a specific event with more than one item on the screen, how the event triggers actions depends on the event.

Note: The following events are triggered in CimView only, except for:

Event	Triggered In
Object Inserted	CimEdit only
Object Removed	CimEdit only
Smart Object	CimEdit and CimView

For	Event	Description	Screen*

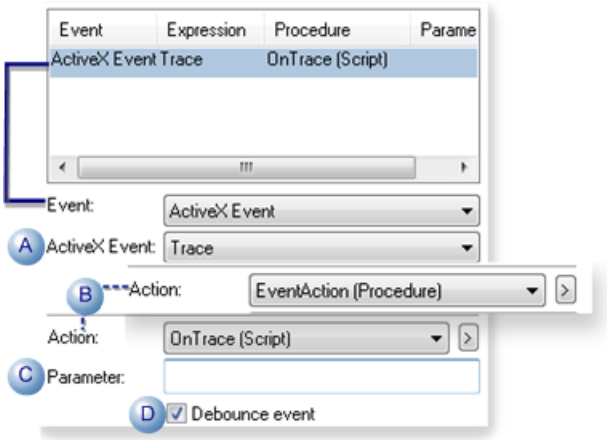
ActiveX Controls			
	ActiveX Event	See the ActiveX control documentation for specific information for an **ActiveX** event.	
Dialog Box			
	Dialog Close	A user closes a screen that was opened using the GefApplication. DialogPopup or GefScreens.DialogPopup method in the CimEdit/CimView Object Model.	Y
Expressions			
	Expression High	The criteria are met for an item with the expression event, no matter what object is highlighted.	Y
	Expression Update		
Events			
	Before Context Menu	Allows an event to be fired right before a CimView Popup (context) menu opens.	Y
	Before Editor Context Menu	Allows an event to be fired right before a CimEdit Popup (context) menu opens.	Y
Keystrokes			
	Key Down	(For all associated items) whenever a selected key is pressed (down or up), no matter what item is highlighted.	Y
	Key Up		
	While Key Down	For all associated objects while the key is pressed and at the specified intervals, no matter what item is highlighted.	Y
Mouse buttons			
	Highlight	When an object is highlighted or un-highlighted.	Y
	Unhighlight		
	Mouse Down	For the item the cursor is over when the left mouse button is pressed (down or up).	Y
	Mouse Up		
	While Mouse Down	(For selected item) while the left mouse button is down and at the specified intervals	Y
Periodic			
	Periodic	(For all associated items) whenever the specified interval elapses, no matter what item is highlighted.	Y
Objects			
	Object Inserted	(CimEdit only) for the associated SmartObject that is being inserted or removed.	

	Object Removed		
	SmartObject	(CimEdit and CimView) when the SmartObject is double-clicked, triggered from a CimEdit menu or from an associated popup menu. (A SmartObject event is configured the same way as any other event.	
		Note: The Object Explorer contains several SmartObjects. In addition, a developer can create new SmartObjects and use the Smart Object, Object Inserted, or Object Removed events to trigger procedures specific to the design project.	
Screens			
	Screen Close	When a user closes or opens the screen on which the event is configured.	Y
	Screen Open	For all associated objects when the screen is open or closed.	Y

*Screen: Y indicates that the event is available at the screen level configuration.

ActiveX Event

Note: Each ActiveX control has its own set of events. See the documentation for the ActiveX control you plan to use.



	Field	Description
A	ActiveX Event	The drop-down list contains events that are available in the ActiveX control.
B	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.
		Procedure
		Script
C	Parameter	Available parameters depend on the selected ActiveX event. Can be used if the selected ActiveX event invokes a script. (The event can invoke a script directly or through a procedure.

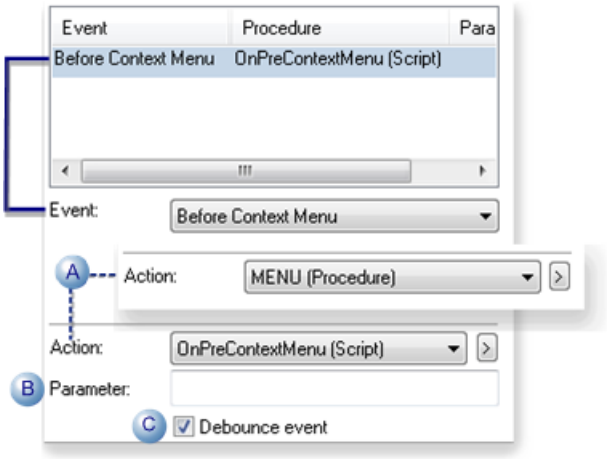
	Field	Description
D	Debounce event	Check
		Clear
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.

Before Context Menu

- Configuration description
- Pre-context menu object model objects
- Popup menu behavior in CimView

Configuration description

Before Context Menu allows an event to be fired right before a Popup menu opens in CimView.



	Field	Description
A	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.
		Procedure
		Script
B	Parameter	Can be used if the event invokes a script directly or through a procedure. The default code begins as follows. Sub OnPreContextMenu(oContextMenu As GefContextMenu) ' Where parameters are:
		oContextMenu

	Field	Description			
C	Debounce event	Check			
		Clear			
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.			

Pre-context menu object model objects

CIMPLICITY includes object model objects for manipulating the context menu that will be displayed.

- Objects include:

GefContextMenu (object)

GefContextMenuItem (object)

GefContextMenuItems (object)

- The GefScreen object also includes the following property.

GefScreen.ContextMenuEnabled (property) (disables the context menu for the screen)

Popup menu behavior in CimView

- Default

A user can click the right-mouse button on any object on a CimView screen to open a default Popup menu.

Exceptions include:

- If the `GefScreen (page 441)` object has been used in a script to disable the context menu, the Popup menu will not open; no further actions are taken.
- (In CIMPLICITY version 7.0.00 and higher) if the role of the logged in user.

- Does not allow a context menu the Popup menu will not open and no further actions are taken.

- Does not allow point targets, Point Control Panel and Quick Trends do not display on the menu.

  - `Before Context Menu` event and its configured script are used

A user can click the right-mouse button on any object to open a customized Popup menu that displays actionable items specified in your script configuration.

Exceptions include:

- If the `GefScreen` object has been used in a script to disable the context menu, the Popup menu will not open; no further actions are taken.
- If the `Before Context Menu` script sets the `GefContextMenu.Displayable (property)` to false, the Popup menu will not open based on that event call.
- (In CIMPLICITY version 7.0.00 and higher) if the user role

- Does not allow a context menu, the Popup menu will not open and no further actions are taken.

- Does not allow point targets, but the `Before Context Menu` script explicitly adds them, the Point Control Panel and/or Quick Trends will display; in the menu when that instance of the menu is called..

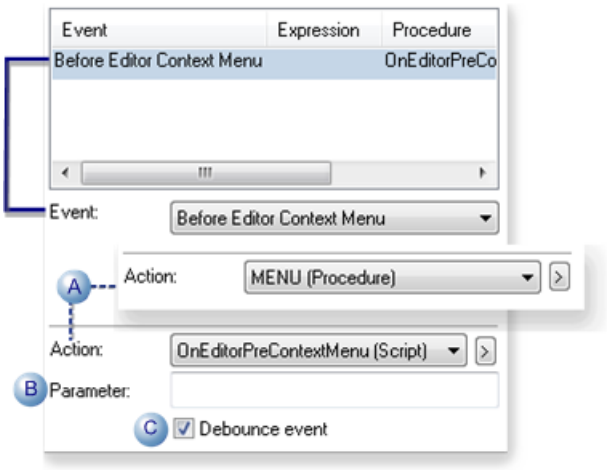
Before Editor Context Menu

- Configuration description
- Pre-context menu object model objects
- Popup menu behavior in CimView

Configuration description

Before Editor Context Menu allows an event to be fired right before a Popup menu opens in CimEdit.

Note: Global configuration allows the event to be handled by a global script on CimEdit screens that do not have the event already configured.



	Field	Description
A	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.
		Procedure
		Script
B	Parameter	Can be used if the event invokes a script directly or through a procedure. The default code begins as follows. Sub OnPreContextMenu(oContextMenu As GefContextMenu) ' Where parameters are:
		oContextMenu
C	Debounce event	Check
		Clear
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.

Pre-context menu object model objects

CIMPLICITY includes object model objects for manipulating the context menu that will be displayed.

- Objects include:

GefContextMenu (object)

GefContextMenuItem (object)

GefContextMenuItems (object)

- The GefScreen object also includes the following property.

GefScreen.ContextMenuEnabled (property) (disables the context menu for the screen)

Popup menu behavior in CimEdit

- Default

A screen designer can click the right-mouse button on any object to open a default Popup menu.

Exceptions include:

- If the `GefScreen` *(page 443)* object has been used in a script to disable the context menu, the Popup menu will not open; no further actions are taken.
- (In CIMPLICITY version 7.0.00 and higher) if the role of the logged in user.

- Does not allow a context menu the Popup menu will not open and no further actions are taken.
- Does not allow point targets, Point Control Panel and Quick Trends do not display on the menu.

- `Before Editor Context Menu` event and its configured script are used

A screen designer can click the right-mouse button on any object on a CimEdit screen to open a customized Popup menu that displays actionable items specified in your script configuration.

Exceptions include:

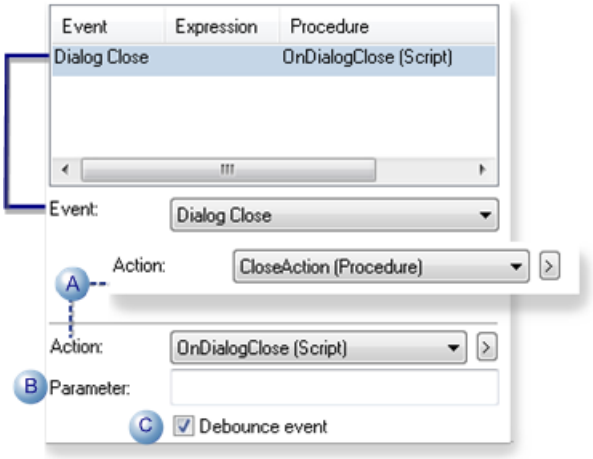
- If the `GefScreen` object has been used in a script to disable the context menu, the Popup menu will not open; no further actions are taken.
- If the `Before Editor Context Menu` script sets the `GefContextMenu.Displayable (property)` to false, the Popup menu will not open based on that event call.
- (In CIMPLICITY version 7.0.00 and higher) if the user role

- Does not allow a context menu, the Popup menu will not open and no further actions are taken.
- Does not allow point targets, but the `Before Editor Context Menu` script explicitly adds them, the Point Control Panel and/or Quick Trends will display; in the menu when that instance of the menu is called..

Dialog Close

Dialog Close triggers a procedure or calls a script when a user closes a screen that was opened using the **GefApplication.DialogPopup** or **GefScreens.DialogPopup** method in the CimEdit/CimView GefObject Model.

The event is triggered in the screen that was specified as the **parent screen** in the **DialogPopup** call.



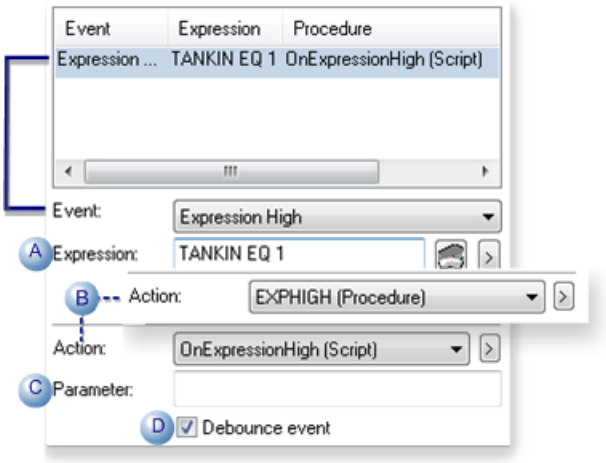
	Field	Description
A	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.
		Procedure
		Script
B	Parameter	Can be used if the event invokes a script directly or through a procedure. The default code begins as follows. Sub OnDialogClose(closeReason As Long) Where parameters are:
		CloseReason
C	Debounce event	Check
		Clear
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.

Note: The following methods and event allow you to use CimView screens the way you would use dialog boxes:

- **GefApplication.DialogPopup** method,
- **GefScreen.CloseEx** method,
- **GefScreen.ObjectToEdit** property and
- **Dialog Close** event.

Expression High Event

Expression High triggers a procedure or calls a script when the value of the expression goes from LOW (zero) to HIGH (non-zero) while the screen is being displayed in CimView.

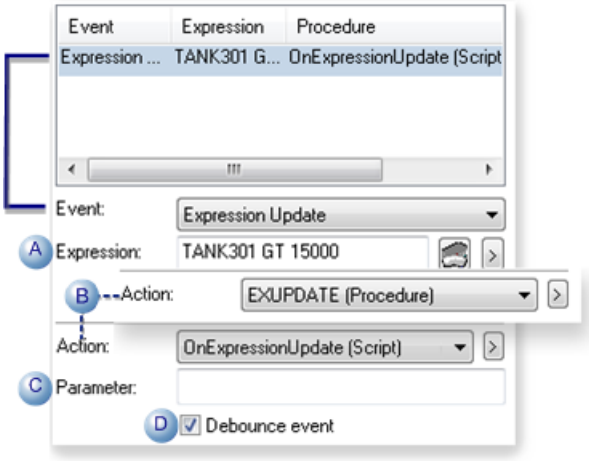


	Field	Description
A	Expression	One or more Point IDs and mathematical operators that can be evaluated.
B	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.
		Procedure
		Script
C	Parameter	Can be used if the event invokes a script directly or through a procedure. The default code begins as follows. Sub OnExpressionHigh(exprValue As Variant) Where parameters are:
		exprValue
D	Debounce event	Check
		Clear

	Field	Description
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.

Expression Update Event

Expression Update triggers a procedure or calls a script when the value of the expression changes.



	Field	Description
A	Expression	One or more Point IDs and mathematical operators that can be evaluated.
B	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.
		Procedure
		Script
C	Parameter	Can be used if the event invokes a script directly or through a procedure. The default code begins as follows. Sub OnExpressionUpdate(exprValue As Variant) Where parameters are:
		exprValue
D	Debounce event	Check
		Clear
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.

Key Events

Key Events Overview

- Key events
- Guidelines for assigning Key UP or Key Down keys
- Details for the Key parameter

Key events

- Key Down
- Key Up
- While Key Down

Guidelines for Assigning Key Up or Key Down Keys

- You may assign most keys as a **Key Down** or **Key Up** event. However, when you do, the triggered procedure overrides any other function the key may have performed.
- Do not use:
- **Ctrl**, **Shift**, or **Alt** alone
- **Esc**, **Tab**, **Enter**, or **Print Screen** either alone or in combination with other keys.
- Ctrl+Alt+Delete
- Ctrl+Alt+Shift+Delete
- Ctrl+Scroll Lock
- Ctrl+Shift+Scroll Lock
- Ctrl+Alt+Scroll Lock
- Pause
- **Ctrl+Pause** (regardless of other modifiers like **Shift** and **Alt**)
- You may assign a key that would normally invoke a menu or Help. If no procedure is assigned, the normal action is invoked. Keys that are affected are:

Key	Accelerator Action
Ctrl+O	File Open
Ctrl+P	File Print
F1	Help Contents
Shift+F1	Context Help
F10	Select Frame Menu

Key	Accelerator Action
Alt+F4	Close Window (**Note:** The window will close after the procedure is done.)

Details for the Key parameter

The low-order byte represents the Windows virtual keycode of the key that was pressed. For ASCII letters and numbers, this corresponds to the ASCII code for that character.

The high order byte has information about the state of the Alt, Control, and Shift keys. The other bits are currently reserved and should not be used. The values are:

GefModAlt	**&h01**
GefModControl	**&h02**
GefModShift	**&h04**

To extract just the high order byte containing the modifier states and the low order byte containing the key code, you can use the following sample code:

modifiers=key / 256

keycode=key mod 256

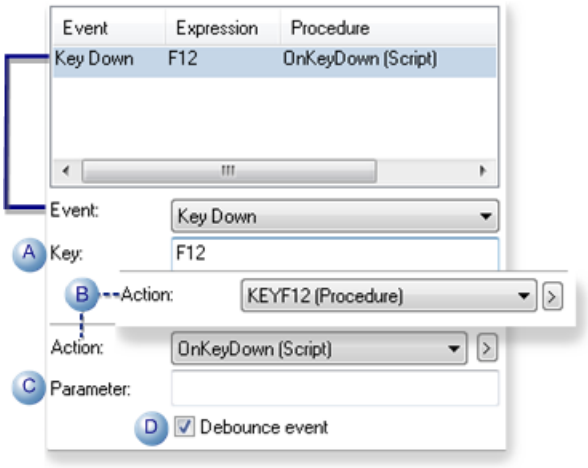
Since the modifiers are bit values, you must use the binary AND operator to test them.

Example

```
If modifiers And gefModAlt Then
  ' do something
End If
```

Key Down Event

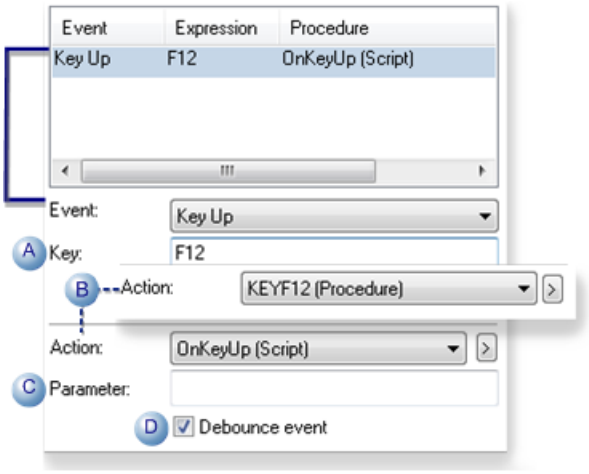
Key Down triggers a procedure or calls a script when a predefined key is pressed down.



	Field	Description
A	Key	Press one key or a combination of keys within guidelines *(page 448)*. The pressed key(s) will appear in the **Key** field.
B	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.
		Procedure
		Script
C	Parameter	Can be used if the event invokes a script directly or through a procedure. The default code begins as follows. Sub OnKeyDown(key As Integer) Where parameters are:
		key
D	Debounce event	Check
		Clear
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.

Key Up Event

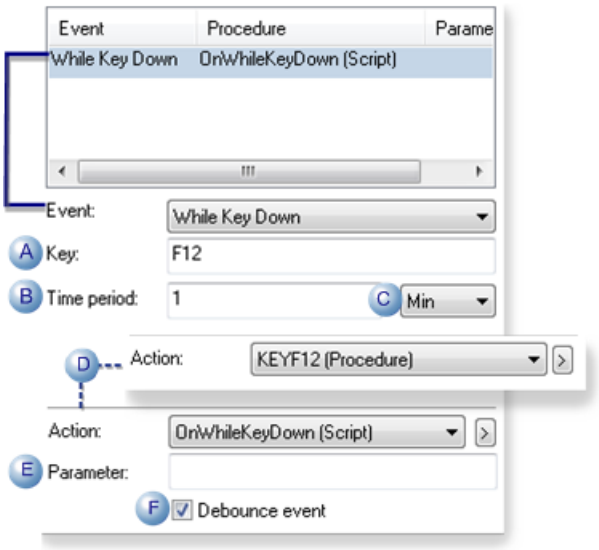
Key Up triggers a procedure or calls a script when a predefined key is pressed then released.



	Field	Description
A	Key	Press one key or a combination of keys within guidelines *(page 448)*. The pressed key(s) will appear in the **Key** field.
B	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.
		Procedure
		Script
C	Parameter	Can be used if the event invokes a script directly or through a procedure. The default code begins as follows. Sub OnKeyUp(key As Integer) Where parameters are:
		key
D	Debounce event	Check
		Clear
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.

While Key Down Event (Keystroke+Periodic)

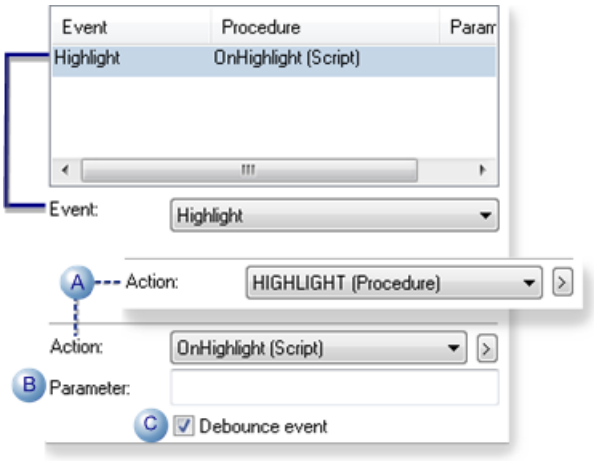
While Key Down triggers a procedure or calls a script when a user presses the selected key and holds it down. The action is not invoked when the user presses the key, but it is invoked every Time period thereafter until the user releases the key.



	Field	Description
A	Key	Press one key or a combination of keys within guidelines *(page 448)*. The pressed key(s) will appear in the **Key** field.
B	Time period	Number of milliseconds, seconds, minutes or hours between the time the event's procedure ends and the time it is re-triggered.
C	Time Type	Select one of the following. • Ms (milliseconds) • Sec (seconds) • Min (minutes) • Hour (hours)
D	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.
		Procedure
		Script
E	Parameter	Can be used if the event invokes a script directly or through a procedure. The default code begins as follows. Sub OnWhileKeyDown(key As Integer) Where parameters are:
		key
F	Debounce event	Check
		Clear
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.

Highlight Event

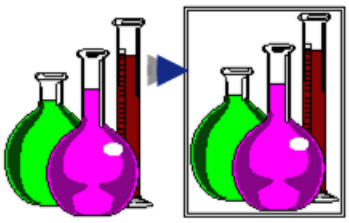
Highlight triggers a procedure or calls a script after an object in CimView changes from not highlighted to highlighted.



	Field	Description
A	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.
		Procedure
		Script
B	Parameter	Can be used if the event invokes a script directly or through a procedure. The default code begins as follows. Sub OnHighlight(highlightReason As Long) Where parameters are:
		highlightReason
C	Debounce event	Check
		Clear
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.

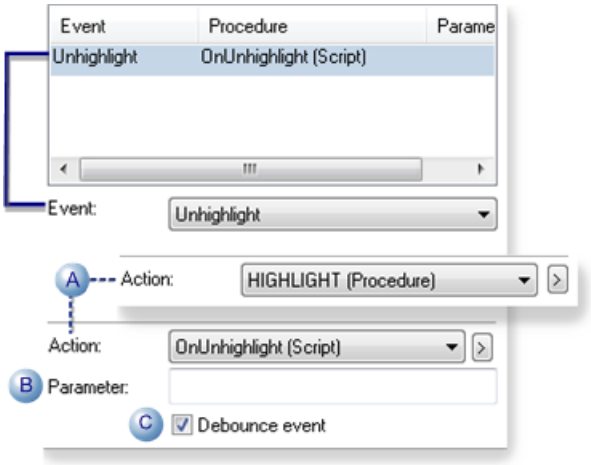
Note: When the `Highlight` event is configured for an object, a CimView user can highlight the object by:

- Placing the mouse cursor on it.
- Tabbing to it.
- Using the arrow keys to select it.



Unhighlight Event

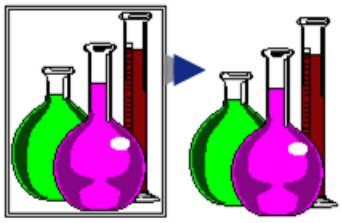
Unhighlight triggers a procedure or calls a script after an object in CimView changes from highlight to not highlighted.



	Field	Description
A	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.
		Procedure
		Script
B	Parameter	Can be used if the event invokes a script directly or through a procedure. The default code begins as follows. Sub OnUnhighlight(highlightReason As Long) Where parameters are:
		highlightReason
C	Debounce event	Check
		Clear
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.

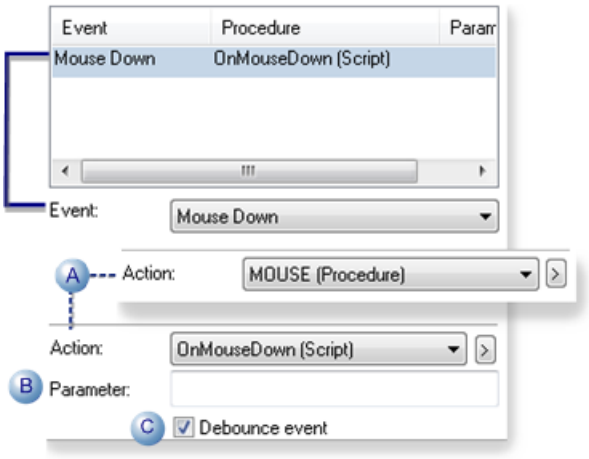
Note: When the `Unhighlight` event is configured for an object, if the object is highlighted, a user can remove the highlight (unhighlight) the object by:

- Removing the mouse cursor from it.
- Tabbing from it.
- Using arrow keys to leave it.



Mouse Down Event

Mouse Down triggers a procedure or calls a script when the left mouse button is clicked down.



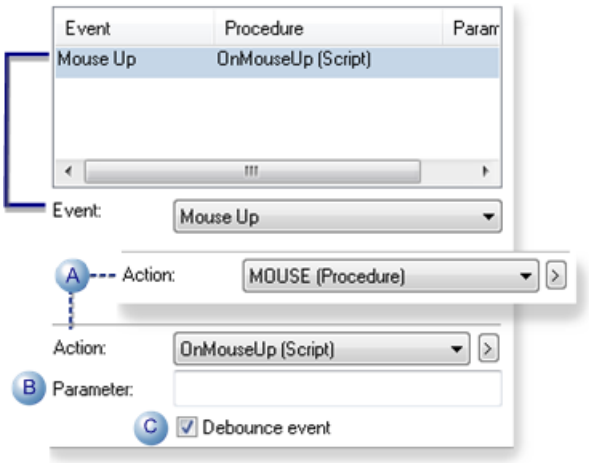
Note: This event also occurs when a user presses Enter on the keyboard.

	Field	Description
A	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.
		Procedure
		Script
B	Parameter	Can be used if the event invokes a script directly or through a procedure. The default code begins as follows. Sub OnMouseDown(x As Long, y As Long, flags As Long) Where parameters are:
		x
		y

	Field	Description
		flags
C	Debounce event	Check
		Clear
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.

Mouse Up Event

Mouse Up triggers a procedure or calls a script when the left mouse button is clicked then released.



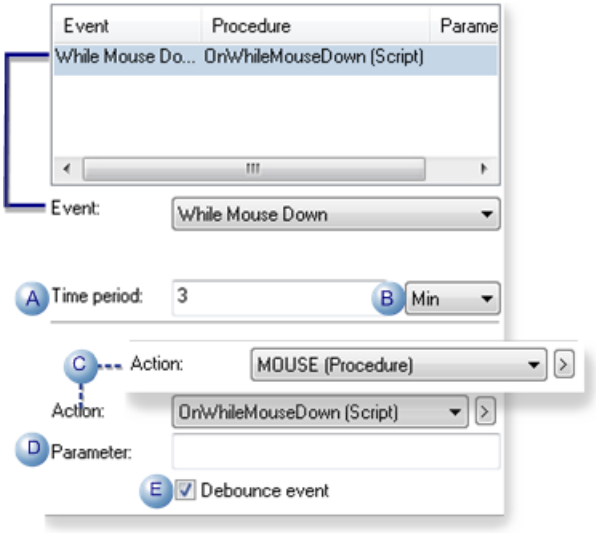
Note: This event also occurs when a user releases the Enter key on the keyboard.

	Field	Description
A	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.
		Procedure
		Script
B	Parameter	Can be used if the event invokes a script directly or through a procedure parameters are: The default code begins as follows. Sub OnMouseUp(x As Long, y As Long, flags As Long) Where parameters are

	Field	Description					
		x					
		y					
		flags					
C	Debounce event	Check					
		Clear					
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.					

While Mouse Down Event (Mouse Button + Periodic)

While Mouse Down triggers a procedure or calls a script when a user clicks the left mouse button and holds it down. The action is not invoked when the user clicks the button but is invoked every **Time period** thereafter until the user releases the key. :



Note: This event also occurs while a user holds **Enter** down on the keyboard.

	Field	Description
A	Time period	Number of milliseconds, seconds, minutes or hours between the time the event's procedure ends and the time it is re-triggered.
B	Time Type	Select one of the following. • Ms (milliseconds) • Sec (seconds) • Min (minutes) • Hour (hours)
C	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.
		Procedure
		Script
D	Parameter	Can be used if the event invokes a script directly or through a procedure. The default code begins as follows. Sub OnWhile Mouse Down(x As Long, y As Long, flags As Long) Where parameters are:
		x
		y
		flags
E	Debounce event	Check
		Clear
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.

Object Events

Object Events Overview

- Object Inserted, Object Removed events
- Object Inserted, Object Removed Overview
- Object Inserted and Smart Object example
- Trigger a Smart Object

Object Inserted, Object Removed Overview

Note: The object events are very useful if you want to trigger actions, that are as simple or as complex as you specify, by simply inserting or removing a SmartObject from a screen. The object can be a frame container, group or object.

All object events that are configured for objects in a frame container or group will trigger their accompanying procedures or scripts anytime the frame container or group is inserted (pasted, dragged).

You can make these objects trigger actions, for example, create a point that goes beyond the scope of the CimEdit screen. Therefore, make sure you anticipate what will happen if you insert an object more than once, on either the same or different screens and then remove one from a screen.

Example Object Inserted and Object Removed Application

An object can be configured so that when it is:

- Inserted on a screen, an **Object Inserted** event triggers a procedure or calls a script that creates several variables for your CimEdit screen.
- Removed from the screen an **Object Removed** event triggers a procedure or calls a script that deletes the variables from the screen.

Whenever the object is inserted, the variables are created. When the object is removed, the variables are deleted.

Tip: Configure several objects with the ObjectInserted and / or ObjectRemoved event on one or more CimEdit screens. Create your own object library *(page 91)* that will display in the Object Explorer. You can then easily find and insert them when you need them.

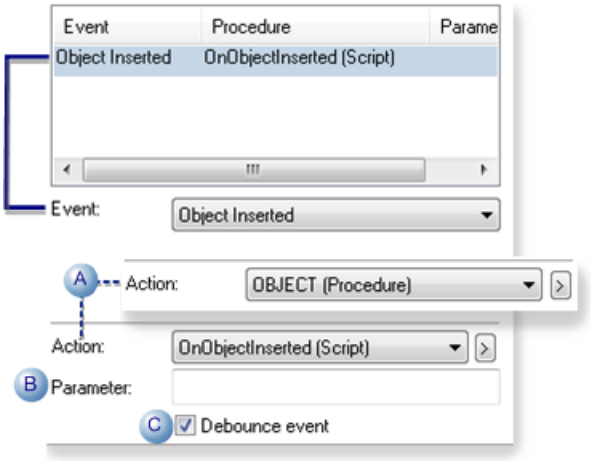
Configure a SmartObject event with the same script as the Object Inserted event. This creates Smart Objects that a designer can trigger both when the object is inserted on a CimEdit screen and during the design process when the screen is open.

To protect your named Smart Object from being decomposed, you can save the CimEdit screen as a runtime-only screen (.cimrt *(page 56)*). As a result, another designer will be able to link the protected SmartObject, but not copy it or view the code.

Object Inserted Event

Object Inserted events enable you to create objects whose actions are triggered on a CimEdit, as opposed to a CimView, screen.

Object Inserted triggers a procedure or calls a script when an object is pasted, dragged, and dropped, inserted from the CimEdit Symbols Library.



	Field	Description
A	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.
		Procedure
		Script
B	Parameter	These entry points have no parameters.
C	Debounce event	Check
		Clear
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/ action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.

Note: The Object Inserted event is not triggered if the object is:

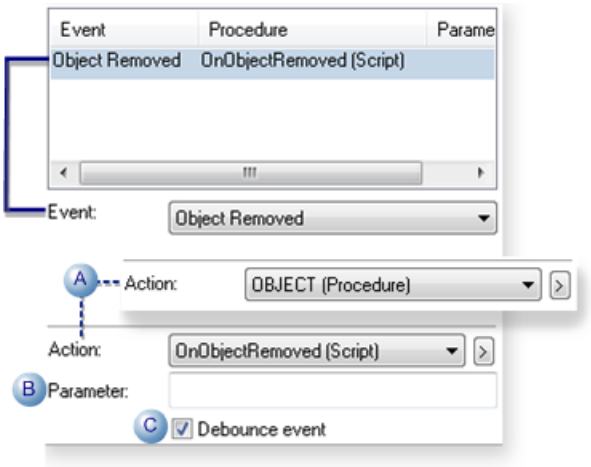
- Inserted using the Object Model,
- Inserted again by using the CimEdit Undo or Redo feature, e.g. after it has been cut or deleted from or previously inserted in the screen.

Note: Events based on inserting or removing objects are only available for the CimEdit screen. They do not apply to CimView or any other application.

Object Removed Event

Object Removed events enable you to create objects whose actions are triggered on a CimEdit, as opposed to a CimView, screen.

Object Removed triggers a procedure or calls a script when an object is deleted or cut from a CimEdit screen.



	Field	Description	
A	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.	
		Procedure	
		Script	
B	Parameter	These entry points have no parameters.	
C	Debounce event	Check	
		Clear	
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.	

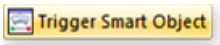
Note: The Object Removed event is not triggered if the object is:

- Deleted using the Object Model,
- Removed again by using the CimEdit Undo or Redo feature, e.g. after it has been inserted in or previously deleted from the screen.

Note: Events based on inserting or removing objects are only available for the CimEdit screen. They do not apply to CimView or any other application.

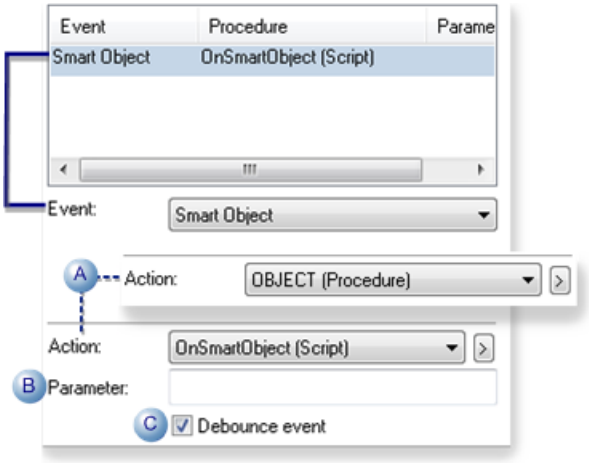
Smart Object Event

The primary purpose of a Smart Object event is to create a Smart Object that a user can double-click in a CimEdit screen and display a simple Smart Object Configuration dialog box instead of the detailed Properties dialog box. The user enters one or more values, as determined by the Smart Object event configuration. The Smart Object is ready for runtime use with no more configuration required.

Note: When the object is assigned a Smart Object event, a screen designer can select the object on the CimEdit screen and click Trigger Smart Object 

in the Home>Properties group on the CimEdit Ribbon bar to open the Smart Object Configuration dialog box.

You can protect your proprietary Smart Objects by preventing the interior logic from being examined. You do this by saving your CimEdit Screen as a runtime-only *(page 56)* screen.



	Field	Description	
A	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.	
		Procedure	
		Script	
B	Parameter	These entry points have no parameters.	
C	Debounce event	Check	
		Clear	
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.	

Tip: The Object Explorer library contains several pre-configured Smart Objects that you may want to review before you take the time to create a new Smart Object. You can also place any Smart Objects you create in the Object Explorer library *(page 91)* so they will be available for frequent use.

To protect your named Smart Object from being decomposed, you can save the CimEdit screen as a runtime-only screen (.cimrt *(page 56)*). As a result, another designer will be able to link the protected Smart Object, but not copy it or view the code.

Note: In order to provide developers with maximum flexibility when a developer configures Smart Object events, the CimView screen provides a Trigger Smart Object option on its popup menu when a Smart Object is right-clicked during runtime. As a result, a CimView operator might open and enter values in the:

- Same Smart Object Configuration dialog box that the CimEdit screen designer uses or,
- A dialog box that bypasses the CimEdit screen designer.

If you want to prohibit the CimView operator from opening the dialog box, you need to disable it through scripting.

Example: Object Inserted and Smart Object

Example: Object Inserted and Smart Object

Object Insert and **Smart Object** events can create greatly increase the ability for a screen designer to create powerful CimView screens with a minimum of effort. This is because actions, including opening dialog boxes that ask for values can now be triggered on the CimEdit screen and used for design purposes.

Although the Object Explorer library contains several Smart Objects, a designer may request Smart Objects that more specific to their design. Following is a simple example showing how **Smart Object** and **Object Inserted** events interact with other object configuration and how they can be easily scripted.

This example creates a rectangle that will display a fill during runtime. The rectangle becomes a Smart Object with a Smart Object event because the fill will reflect the value of a point that can be specified when the Smart Object is placed in CimEdit, during a CimEdit session and then, in CimView.

Tip: To protect your named Smart Object from being decomposed, you can save the CimEdit screen as a runtime-only screen (**. cimrt**). As a result, another designer will be able to link the protected Smart Object, but not copy it or view the code.

Procedures for the following steps will create the Smart Object with the **Smart Object** and **Object Inserted** events.

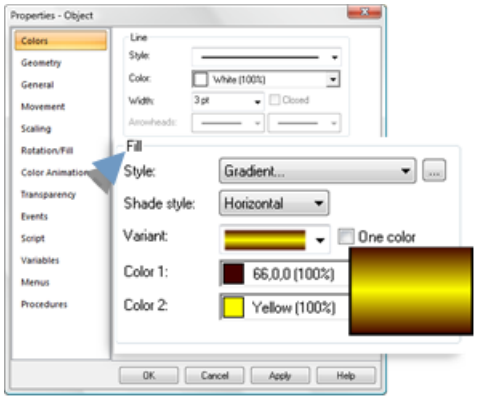
Step 1 *(page 464)*	Configure the basic object to fill during runtime.
Step 2 *(page 465)*	Turn the object into a Smart Object with a **Smart Object** event.
Step 3 *(page 467)*	Create an **Object Insert** event.

Step 4 *(page 468)*	Place the Smart Object in the Object Explorer Library.
Step 5 *(page 468)*	Test using the **Object Insert** event.
Step 6 *(page 470)*	Test using the Smart Object event in CimEdit.
Step 7 *(page 471)*	Test using the Smart Object event in CimView.

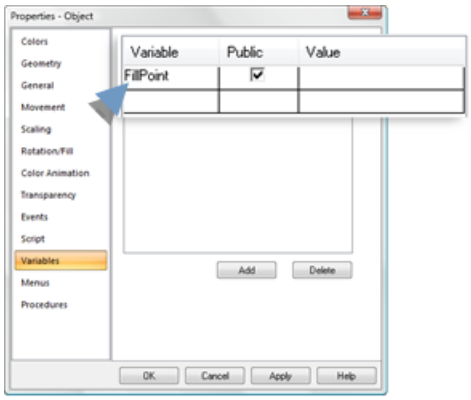
Step 1. Configure an Object to Fill

1. Place a rectangle on the CimEdit screen.

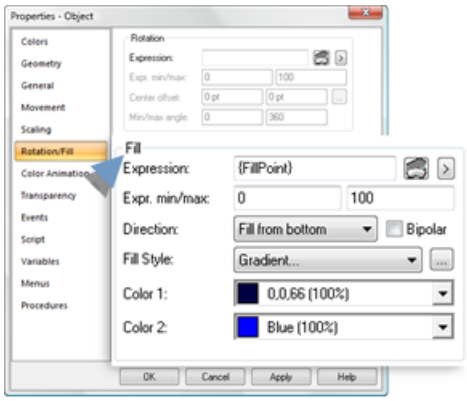
2. Open the rectangle's Properties dialog box.
3. Select the Colors tab.
4. Select a basic color or pattern in the Fill box.



5. Select the Variables tab.
6. Create a **FillPoint** variable. Leave the **Value** field blank.



7. Select the Rotation/Fill tab.
8. Enter **{FillPoint}** in the **Expression** field of the Fill box.
9. Configure the color you want to represent the fill.



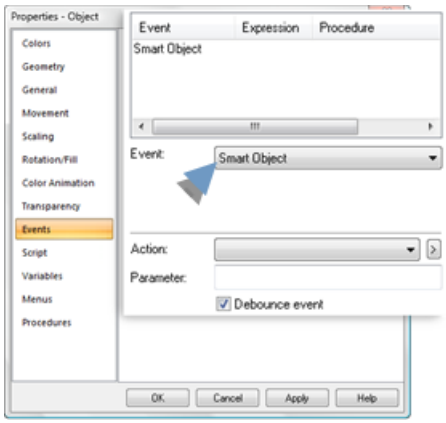
The object is formatted to fill during runtime, it now needs a value in order to work.

Step 2. Turn the Object into a Smart Object with a Smart Object Event

1. Select the Events tab.

   Note: You may see an error message telling you the Variable **variable name** is not defined. Click **OK**.

2. Select the **Smart Object** event.



3. Click the **Popup menu** button to the right of the Action field.

4. Select New Script from the Popup menu.

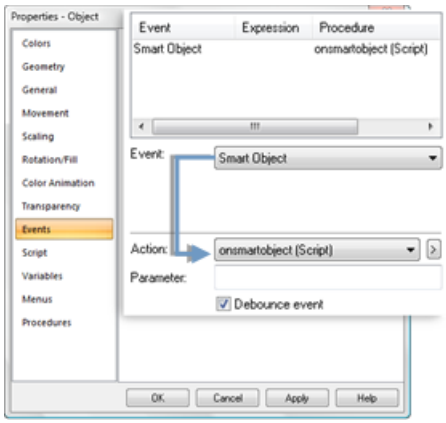
5. Create a script to display a dialog box with one field. The field requests a Point ID or expression for the **FillPoint** variable.

```
Begin Dialog UserDialog ,,194,84,"SmartObject Configuration",.DlgProc
      OKButton 100,66,40,14
      CancelButton 144,66,40,14
      TextBox 44,26,120,12,.PointBox
     Text 1,28,39,8,"Fill Point",.PointTxt
     PushButton 168,25,16,14,"...",.PointBrowse
End Dialog
Function DlgProc(ControlName$,Action%,SuppValue%)
      Dim browseObject As Object
      Dim entity As String
      If Action% = 2 And ControlName$ = "PointBrowse" Then
            Set browseObject = CreateObject("CimBrowse")
            browseObject.SetBrowserProperty 2
            browseObject.SetBrowserProperty 16
            browseObject.SetBrowseEntity "POINT"
            res = browseObject.BrowseEntity(entity)
            DlgText "PointBox",entity
            Set browseObject = Nothing
            DlgProc = 1
      End If
End Function
Sub onsmartobject()
Begin Dialog UserDialog ,,194,63,"SmartObject Configuration",.DlgProc
    OKButton 108,36,40,14
    CancelButton 152,36,40,14
    TextBox 48,16,120,12,.PointBox,&h0000
    Text 4,18,39,8,"Fill Point",.PointTxt
    PushButton 175,15,16,14,"...",.PointBrowse
End Dialog
  Dim PointVar As GefObjectVariable
  Dim obj As GefObject
```

```
  Dim configDlg As UserDialog
  Set obj = CimGetObject
  Set PointVar = obj.GetVariable("FillPoint")
  configDlg.PointBox = PointVar
    If Dialog(configDlg) Then
        PointVar = configDlg.PointBox
    End If
 End Sub
```

6. Close the Script Editor.

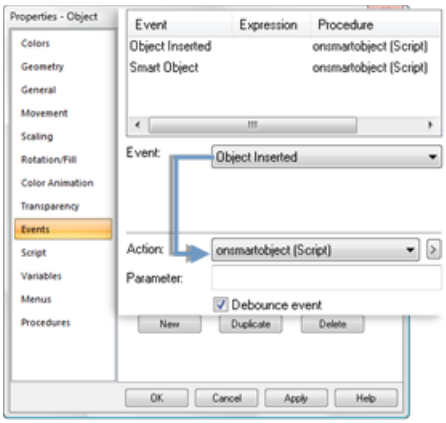
   The Events tab displays the **Smart Object** event and the name of its script.



Step 3. Create an Object Inserted Event

1. Click **New** on the Events tab.
2. Select **Object Inserted** in the **Event** field.
3. Click the down arrow in the **Action** field.
4. Select **onsmartobject (Script)** , which is the name given to the script written for the SmartObject event.

The Events tab displays the Smart Object and Insert Object events with the same script designated for the procedure.

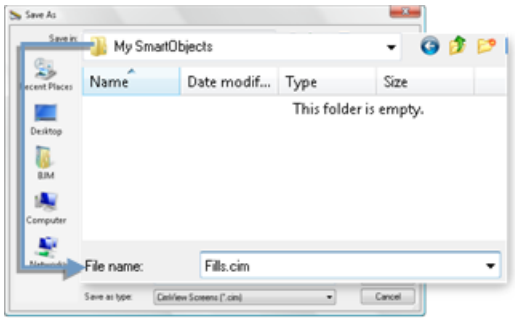


Step 4. Place the Smart Object in the Object Explorer Library

1. Click the File menu on the CimEdit menu bar.
2. Select Save As.

   The Save As dialog box opens.

3. Open the ...\Program Files\Proficy\Proficy CIMPLICITY\Symbols folder.
4. Create and open a folder called My SmartObjects.
5. Enter Fills in the **File name** field.



6. Click Save.
7. Close the Fills screen.

You are now ready to test the Smart Object.

Step 5. Test using the Object Inserted Event

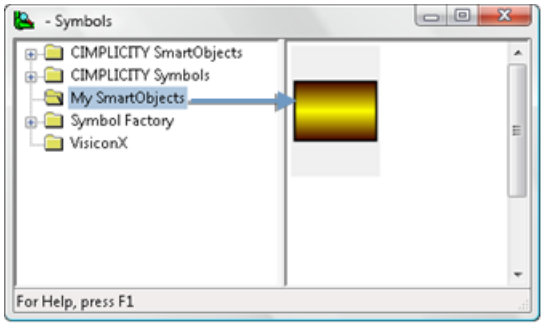
1. Start the CimEdit project if it is not already running.

2. Open a new CimEdit screen.

3. Click the Object Explorer button .

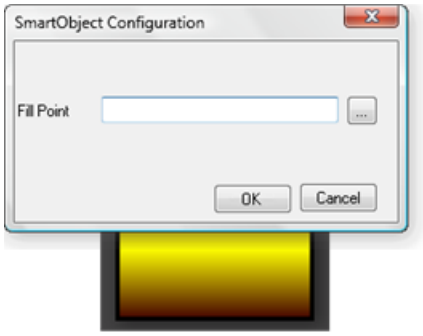
   The Object Explorer opens.

4. Select the My SmartObjects folder.



5. Drag the Fill object onto the CimEdit screen.

   The SmartObject Configuration dialog box that you created opens.

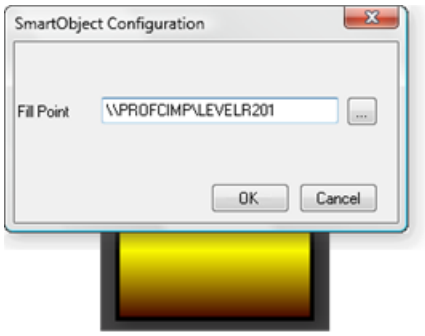


6. Click the **Browse** button to open the Select a Point browser.

7. Select a Point ID (e.g. an integer device point connected to a PLC) that will display values at runtime.

   You can enter any value that you can use for a variable. Therefore, you can also enter a number or expression (e.g. 25) as the value if you do not have an active Point ID.

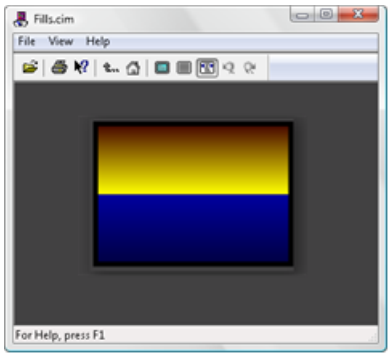
   The value you select displays in the **Fill Point** field.



8. Click **OK**.
9. Click the **Test Screen** button in the Home>Runtime group on the CimEdit Ribbon bar.

   CimView opens.

   The SmartObject displays the value of the selected Point ID (value).



10. Close CimView and return to the CimEdit screen.

Step 6. Test using the Smart Object Event in CimEdit

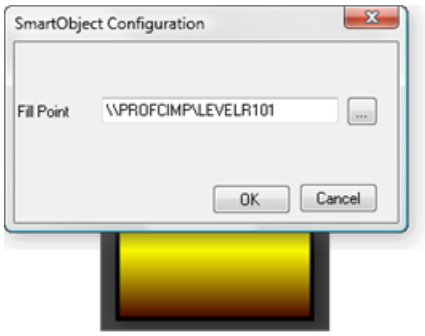
1. Double-click the Smart Object in CimEdit.

   The SmartObject Configuration dialog box appears displaying the last value that was entered. The line in the example script for displaying the last value is:

   configDlg.PointBox = myvar

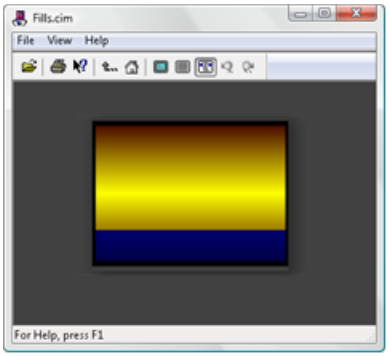
   Note: It the object was not a Smart Object, the Properties dialog box would appear.

2. Enter a new value in the **Fill Point** field.



3. Click the Test Screen button in the Home>Runtime group on the CimEdit Ribbon bar.

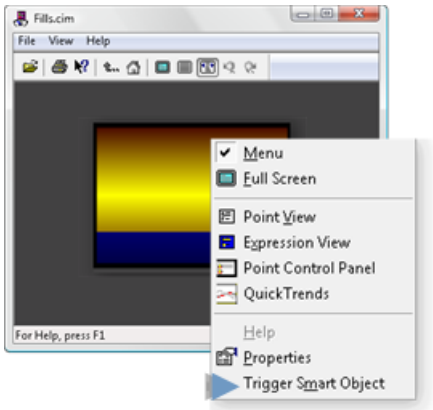
CimView opens. The Smart Object displays the last selected value.



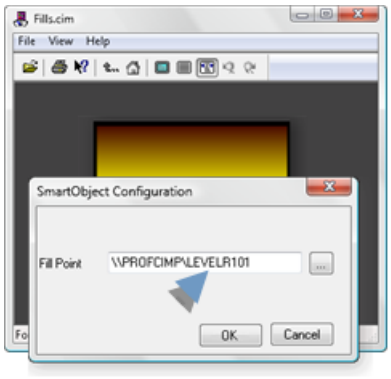
Step 7. Test using the Smart Object Event in CimView

Note: This functionality is part of the Smart Object event. You can suppress the functionality in your script if you do not want an operator to open the Smart Object Configuration dialog box.

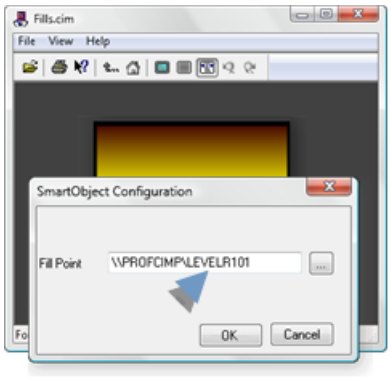
1. Right-click the Smart Object in runtime mode (on a CimView screen).
2. Select Trigger Smart Object on the Popup menu.



The same SmartObject Configuration dialog box that displayed in CimEdit opens in CimView.

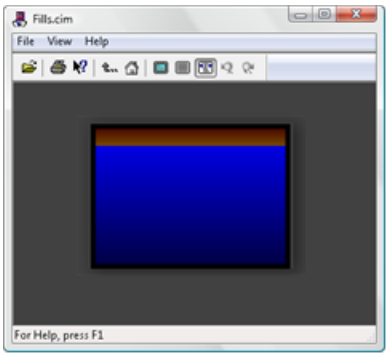


3. Enter another value in the **Fill Point** field.



4. Click OK.

The SmartObject displays the configuration for the new value.



Trigger a Smart Object

After a Smart Object is created it can be triggered in both CimEdit and CimView.

- CimEdit
- CimView

CimEdit

A screen designer can use any of the following procedures to trigger a Smart Object in CimEdit.

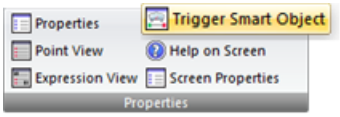
- Smart Object Double-click

Double-click the Smart Object.

If the object is not a Smart Object, the Properties dialog box will open. If the object is a SmartObject whatever actions were configured for the Smart Object event (usually a customized dialog box opening) will occur.

- CimEdit Ribbon Bar

Click Trigger Smart Object on the Home>Properties group on the CimEdit Ribbon bar.

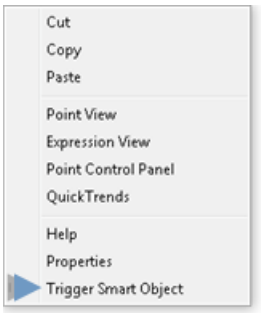


- CimEdit Popup Menu

1. Right-click the Smart Object on the CimEdit screen.

A Popup menu opens.

1. Select Trigger Smart Object.



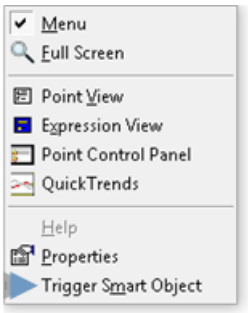
CimView

The Smart Object can also be triggered in CimView.

1. Right-click the Smart Object on the CimView screen.

A Popup menu opens.

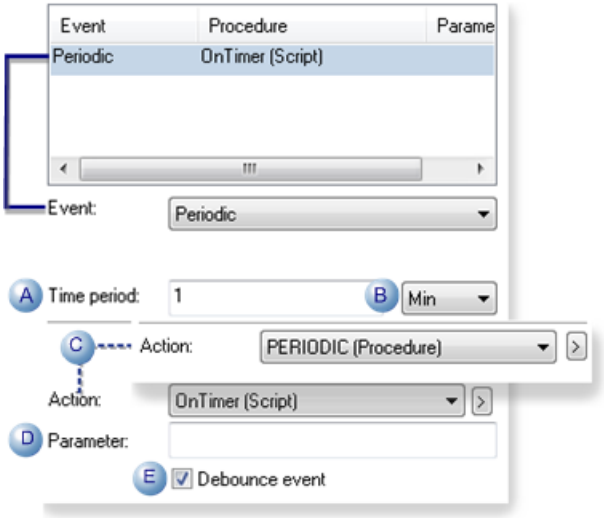
2. Select Trigger Smart Object.



Tip: Use the ObjectInserted and ObjectRemoved events to trigger a SmartObject when it is being placed or removed from the CimEdit screen.

Periodic Event

Periodic triggers a procedure or calls a script periodically at the rate you specify in the Time Period.

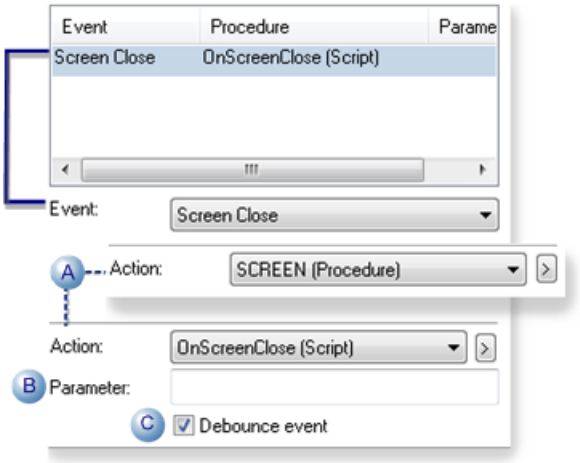


	Field	Description
A	Time period	Number of milliseconds, seconds, minutes or hours between the time the event's procedure ends and the time it is re-triggered.

	Field	Description
B	Time Type	Select one of the following. • Ms (milliseconds) • Sec (seconds) • Min (minutes) • Hour (hours)
C	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.
		Procedure
		Script
D	Parameter	These entry points have no parameters.
E	Debounce event	Check
		Clear
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.

Screen Close Event

Screen Close triggers a procedure or calls a script when a user closes the screen on which the event is configured.

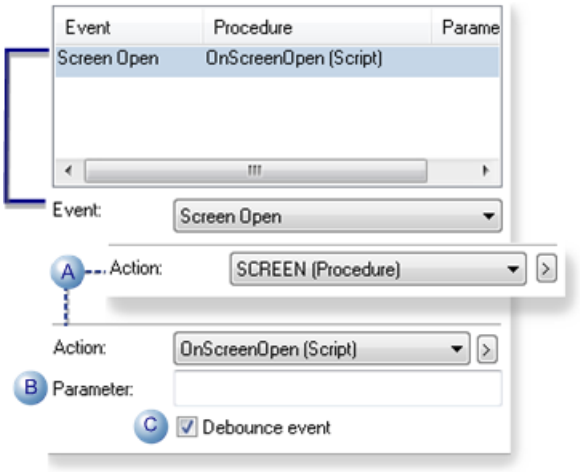


	Field	Description
A	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.
		Procedure
		Script

	Field	Description
B	Parameter	These entry points have no parameters.
C	Debounce event	Check
		Clear
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.

Screen Open Event

Screen Open triggers a procedure or calls a script when a user opens the screen on which the event is configured.



	Field	Description
A	Action	Contains a drop-down list of existing procedures and scripts. Use either of the following.
		Procedure
		Script
B	Parameter	These entry points have no parameters.
C	Debounce event	Check
		Clear

	Field	Description
		Note: When debouncing is enabled, the event/action queue is always processed one action at a time; the current action must be completed before the next action can start. This makes the queue's event/action sequencing predictable and avoids situations where many events could trigger actions and consume unacceptable levels of CPU and memory.

CimEdit Procedures

CimEdit Procedures

CIMPLICITY provides several tools to make things happen in CimEdit and in CimView. The tools can be categorized as triggers that are configured as events and results that can be configured through procedures or scripts.

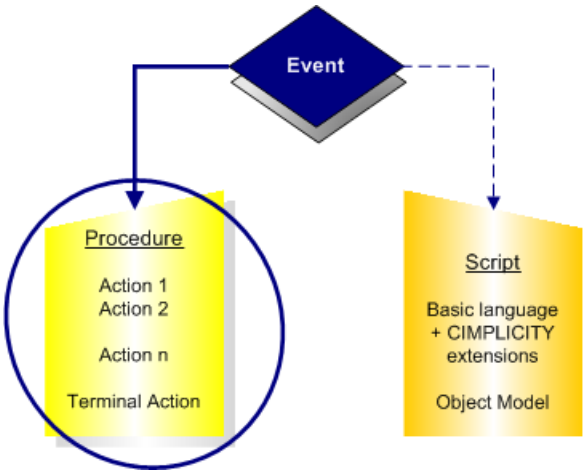
Trigger

An event *(page 432)* triggers a procedure or calls a script. CimEdit provides a long list of events from which you can choose the best one for your requirements.

Result

A procedure contains one or more actions that are triggered in the specified order when an event occurs and while the screen is displayed in CimView. CimEdit provides several actions from which a screen designer can easily compile a meaningful list.

A script *(page 530)* , which is usually written by a system administrator, uses the same Editor and Basic language as the Basic Control Engine. Anything you can do in a normal script, you can do in a CimEdit script. CimEdit provides additional extensions to give you a wider range of screen development choices. However, CimEdit scripts are only accessible from the screen in which you create them.



rect 254, 140, 364, 275 CimEdit Scripts *(page 530)*
rect 144, 3, 249, 85 CimEdit Events *(page 432)*

Actions Available to Build Procedures

CimEdit provides you with several choices of actions to build an appropriate procedure.

You can assign a procedure to be executed in response to events that you specify for a screen, frame container, group, or object. The actions that these events trigger affect different items in CimEdit, as follows:

Action Type	CimEdit Action
Command	Execute Command *(page 490)*
Screen	Close Screen (Terminal) *(page 492)*
	Home Screen (Terminal) *(page 493)*
	Open Faceplate *(page 495)*
	Next Screen *(page 494)*
	Open Screen *(page 502)*
	Overlay Screen (Terminal) *(page 508)*
	Previous Screen (Terminal) *(page 512)*
	Print Screen *(page 513)*
Method / Script	Invoke Method *(page 513)*

Action Type	CimEdit Action
	Invoke Script *(page 516)*
Setpoint	Absolute Setpoint *(page 517)*
	Expression Setpoint *(page 518)*
	Ramp Setpoint *(page 520)*
	Relative Setpoint *(page 522)*
	Toggle Setpoint *(page 523)*
	Variable Setpoint *(page 524)*
Variable	Variable Assign *(page 525)*
	Expression Set Variable *(page 528)*

Procedure Access Locations for a Selected Object

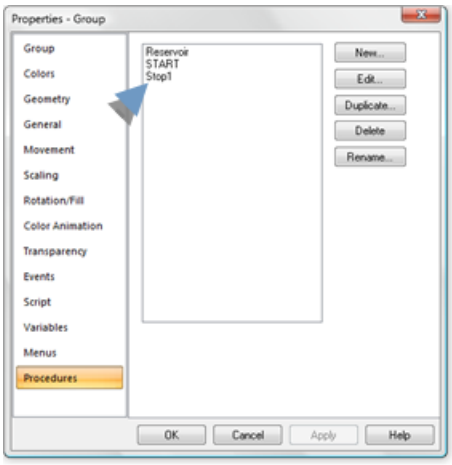
Procedures can be accessed in two locations in a selected object's dialog box.

- Procedures Created at the object level.
- Procedures Created at the screen level and below in an object's hierarchy.
- Procedure guidelines

Procedures Created at the Object Level

Select **Procedures** in an object's Properties dialog box.

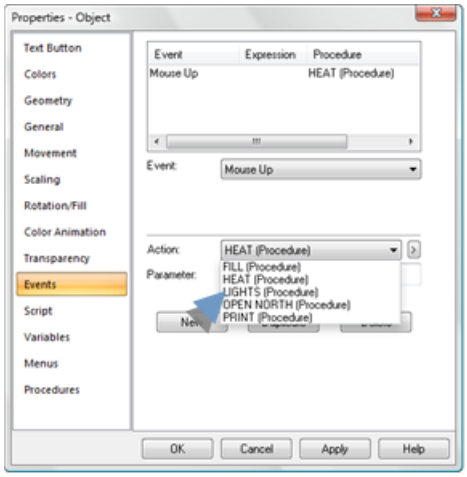
Result: Procedures that are owned by the object are listed.



Procedures Created at the Screen Level and Below in an Object Hierarchy

1. Select **Events** in the object's Properties dialog box.
2. Create or edit an event.
3. Expand the dropdown list in the Action field.

   Result: Procedures that have been created in the hierarchy above the object's listed, e.g. a group that the object is in or the screen.



guide: Procedure Guidelines

- Procedures that you configure at the screen level are available for any object on the screen when you configure an event for that object.

However, if you create a procedure for an object that has the same name as a procedure created for the screen, CimEdit uses the object's procedure for the object.

- All the procedures that have been configured for a screen, frame, or group to which the object belongs are available to the object
- Procedures that you configure for an object supersede any procedure with the same name that is further up in the hierarchy.

Example of Two Procedures with the Same Name

You have created a procedure called **Switch_On** at the screen level.

You insert a new object on the screen that contains a **Switch_On** procedure. CimView recognizes the object's **Switch_On** procedure, not the screen's, for that object.

When you configure procedures for an object, you can easily review what procedures they have been created previously for the object or higher in the object's tree. You do this through the Procedures tab of the Properties dialog box.

- New
- Edit
- Duplicate
- Delete
- Rename

New a new procedure–summary

4. Click **New**...

   The Procedure Information dialog box opens.

5. Enter information *(page 483)* to define the procedure.

   Note: A

   Edit a procedure - summary

6. Select a procedure from the list of procedures.
7. Click **Edit...** to change it.

   The Procedure Information dialog box opens with the current attributes for the procedure.

8. Continue with editing the procedure.

   Duplicate a procedure

9. Select a procedure from the list on the procedures.
10. Click **Duplicate...** to create another procedure with the same actions as the one you selected.

11. Enter the new procedure name when the Procedure Name dialog box opens.
12. Click **OK**.

    Delete a procedure:

13. Select a procedure from the list on the procedures.
14. Click **Delete** to remove the procedure from the list.

    A CimEdit dialog box opens, asking you to confirm your request. It also tells you the number of actions and objects associated with the procedure.

15. Click **OK** to confirm your request.

    Rename a procedure:

16. Select a procedure from the list of procedures.
17. Click **Rename...** to rename it.
18. Enter the new procedure name when the Procedure Name dialog box opens.
19. Click **OK**.

Procedure Configuration

Procedure Configuration

Procedures are very useful for anyone who configures a CimEdit screen. In fact, if you are a screen designer without a lot of programming experience you will find working with procedures can be a welcome alternative to writing scripts.

The core of your configuration is in the Procedure Information dialog box that you access through the selected object's Properties dialog box.

1 *(page 483)*	Create a new procedure.
2 *(page 484)*	Edit a procedure.
3 *(page 486)*	Delete an object's procedure.

4 *(page 486)*	Configure a procedure's advanced properties.

1. Create a New Procedure

1. Create a New Procedure

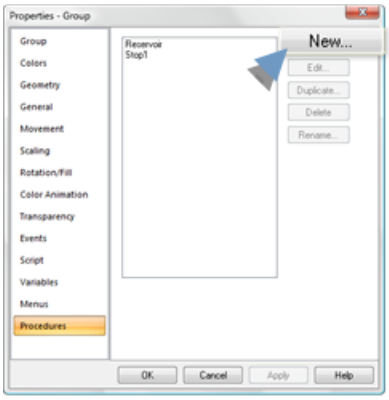
1.1 *(page 483)*	Open a New Procedure Information dialog box.
1.2 *(page 484)*	List actions for the procedure.

1.1. Open a New Procedure Information Dialog Box

Do one of the following in the Properties dialog box for a screen, a group or object.

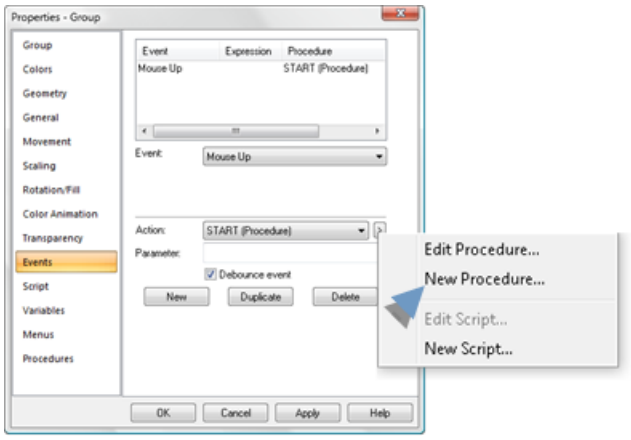
- Select **Procedures**.

Click the New button



- Select **Events**.

1. Click the Popup Menu button to the right of the **Action** field.
2. Select New Procedure on the Popup menu.



A Procedure Information dialog box opens when you use either method.

1.2. List Actions for the Procedure

1. Name the new procedure in the **Procedure name** field.
2. Configure as many actions as are needed to perform the procedure.

   Important: Make sure the actions are listed in the order that they should be executed.

3. Make sure the last action on the list is a terminal action.

   The following actions are terminal actions:

   - Close Screen
   - Home Screen
   - Overlay Screen
   - Previous Screen

4. Click **OK**.

2. Edit a Procedure

You can edit any existing procedure either through an object's:

- Procedures tab.
- Events tab.

Procedures Tab

1. Select **Procedure** in an object's Properties dialog box.

2. Select the procedure you want to edit.
3. Click Edit.

   The Procedure Information dialog box opens.

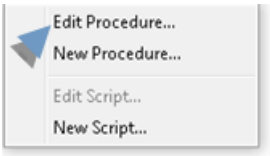
4. Select the Actions tab.

   The Actions tab displays the information for the selected procedure.

5. Make the required changes.
6. Click **OK**.

   Events Tab

7. Select **Events** in the object's Properties dialog box.
8. Open the Procedure Information dialog box using either method:
   - Select a procedure in the list of procedures displayed for the event; click the right mouse button.
   - Select a procedure from the list of procedures in the **Action** field; click the Popup Menu button to the right of the **Action** field.
9. Using either method, select Edit Procedure on the popup menu.



   The Procedure Information dialog box opens.

10. Select the Actions tab.

    The Actions tab displays the information for the selected procedure.

11. Make the required changes.
12. Click OK.

Note: If you edit a procedure, it will be associated with the event if it currently is not associated.

3. Duplicate or Delete an Object's Procedure

1. Click Procedure in an object's Properties dialog box.
2. Click Duplicate.

   Delete a Procedure

   You can delete the association between any procedure and an object. You will not delete the procedure's association with other objects or a group with which it is associated.

   Deleted from an Object

3. Display the Procedure tab of the object's Properties dialog box.
4. Click **Delete**.

   Deleted from an Event

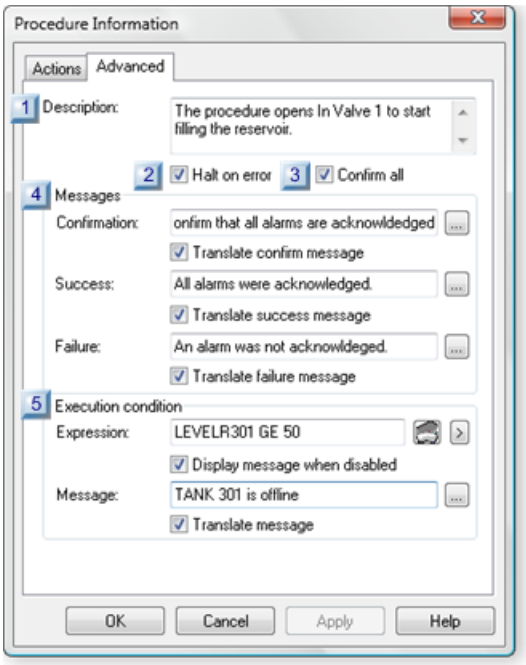
5. Select Events *(page 433)* in the object's Properties dialog box.
6. Select the procedure you want to delete in the list of procedures.
7. Click **Delete**.

4. Configure a Procedure's Advanced Properties

4. Configure a Procedure's Advanced Properties

When configuring a procedure, use the Advanced tab of the Procedure Information dialog box to:

- Enter a description for the procedure.
- Define procedure functions.
- Define procedure messages.
- Specify an **Execution condition** for the procedure.



rect 2, 50, 29, 78 *(page 487)*
rect 171, 93, 198, 117 *(page 488)*
rect 9, 104, 34, 135 *(page 488)*
rect 8, 237, 34, 263 *(page 488)*
rect 78, 96, 102, 120 *(page 487)*

1 *(page 487)*	Description
2 *(page 487)*	Halt on error
3 *(page 488)*	Confirm all
4 *(page 488)*	Messages
5 *(page 488)*	Execution condition

1	Description

(Optional) Procedure description.

2	Halt on error

Check to halt if an error occurs during an action in the procedure.

3	Confirm all

Check for the user to confirm each action in the procedure before it is executed.

4	Messages

Entries display for the specified condition, as follows.

Entry	Description
Confirmation	Entry displays when the procedure is activated. Supercedes other confirmation messages. Check Translate confirm message to make the confirmation message available for translation.
Success	Entry displays for success of the procedure. Check Translate success message to make the confirmation message available for translation.
Failure	Entry displays when an action fails. Check Translate failure message to make the confirmation message available for translation.

5	Execution condition

If an execution condition is:	The procedure is:
True	Available.
False	Not available.

An execution condition includes the following.

Item	Description
Expression	One or more point ID's and mathematical operators that can be evaluated.
Display message when disabled and Message	Check to display the Message field entry when the procedure is disabled.
Translate message	Check to make this string available for translation. For more information on translation, see Translation *(page 66)*.

Execution Condition for a Procedure

When an Execution Condition is evaluated as:

True	The procedure behaves normally
False and no message is defined	The system behaves as if the procedure does not exist. Example If the procedure is assigned to a key, nothing will happen when the user presses the key.
False and a message is defined	A user may think the procedure exists. However, when the user selects the procedure, the message displays and the procedure will not run.

If the expression contains unavailable points, a user may think a procedure exists. However, when the user selects the procedure, the following message displays.

The execution condition for this procedure contains unavailable points. Would you like to execute the procedure anyway?

If the user clicks:

Yes	The procedure can be performed
No	The procedure is not performed

Examples of how an Execution Condition is Used

If a custom application provides security information as a CIMPLICITY point, this feature can be used to provide security control for procedure execution.

If an application runs in modes, such as Manual and Automatic, this feature can be used to disable certain procedures while in Automatic mode.

Available Actions

Available Actions

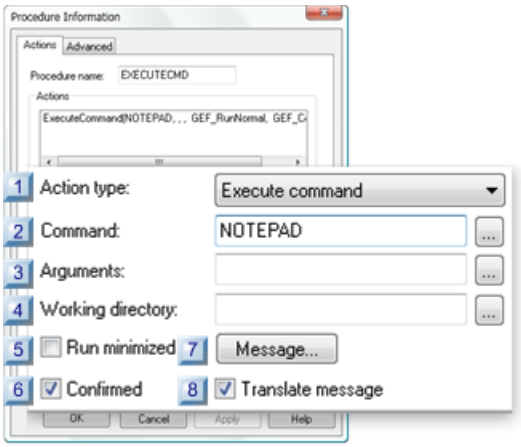
The specifications you enter to instruct CimView depend on what action you choose.

- Execute Command *(page 490)*
- Close Screen (Terminal) *(page 492)*
- Home Screen (Terminal) *(page 493)*
- Next Screen (Terminal) *(page 494)*
- Open Screen *(page 502)*
- Overlay Screen (Terminal) *(page 508)*
- Previous Screen (Terminal) *(page 512)*
- Print Screen *(page 513)*
- Invoke Method *(page 513)*
- Invoke Script *(page 516)*
- Absolute Setpoint *(page 517)*
- Expression setpoint *(page 518)*
- Ramp Setpoint *(page 520)*
- Relative setpoint *(page 522)*
- Toggle Setpoint *(page 523)*
- Variable Setpoint *(page 524)*
- Variable Assign *(page 525)*
- Expression Set Variable *(page 528)*

Execute Command Action

Execute Command executes a command in CimView

Configure the action as follows.



rect 105, 203, 132, 227 *(page 491)*
rect 105, 229, 132, 253 *(page 492)*
rect -3, 228, 24, 252 *(page 491)*
rect -3, 202, 24, 226 *(page 491)*
rect -5, 177, 22, 201 *(page 491)*
rect -5, 154, 22, 178 *(page 491)*
rect -5, 128, 22, 152 *(page 491)*
rect -5, 102, 22, 126 *(page 491)*

1 *(page 491)*	Action type
2 *(page 491)*	Command
3 *(page 491)*	Arguments
4 *(page 491)*	Working directory
5 *(page 491)*	Run minimized
6 *(page 491)*	Confirmed

7 *(page 491)*	Message
8 *(page 492)*	Translate message

1	Action type

Select Execute command.

2	Command

Enter one: Command path name, folder name, document name with a registered extension, web page reference. Click browse to open the Windows Select a Command dialog box.

Note: If the path name contains spaces, enclose it in double quotes.

3	Arguments

(Usually blank) Addition information used by the program

...	Opens a Text Box to enter arguments.

4	Working directory

Directory path to command if it is different from the screen file directory.

...	Opens a Browser for Folder browser.

5	Run minimized

When checked, minimizes window associated with the command.

6	Confirmed

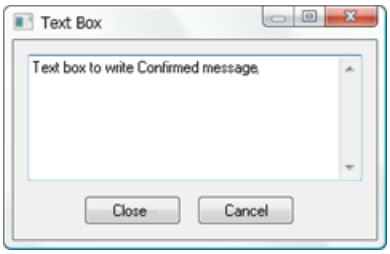
When checked, Confirmed requires confirmation at runtime.

A message box will display before the action occurs requesting confirmation to proceed.

Default: Clear. Confirmation is not required.

7	Message

Opens a Text Box to enter a customized confirmation message.



8	Translate message

Check to make the customized message text available to be emitted to the selected translation file.

For more information on translation, see Translation *(page 66)*.

Screen Actions

Screen Actions

Note: Both the **Home Screen** and **Previous Screen** features rely on actual files to work.

When you enter CimView by pressing the **Test Screen** button in CimEdit, CimView is not passed a file to work with. The screen is passed in memory. Therefore, because there are no available files for the previous or home screens, the **Previous Screen** and **Home Screen** buttons are disabled. (This is why you don't have to save the screen before you push the **Test Screen** button. Of course, it is recommended that you do.)

When you do an overlay from the test screen you have opened, the **Home Screen** button is enabled. The new screen file becomes the home screen–it is the first file opened. The **Previous Screen** button is disabled. There is next file for the previous screen.

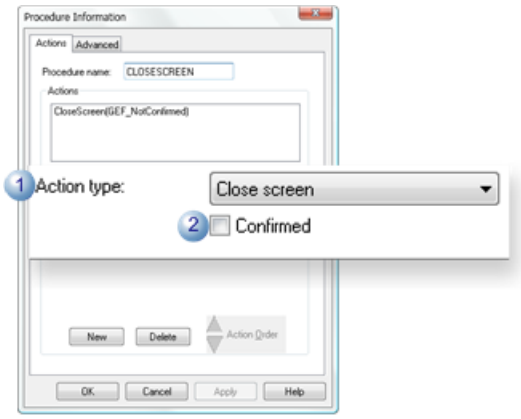
- Close Screen (Terminal) *(page 492)*
- Home Screen (Terminal) *(page 493)*
- Next Screen (Terminal) *(page 494)*
- Open Faceplate *(page 495)*
- Open Screen *(page 502)*
- Overlay Screen (Terminal) *(page 508)*
- Previous Screen (Terminal) *(page 512)*
- Print Screen *(page 513)*

Close Screen (Terminal)

Close Screen

- Closes the current screen.
- Is a terminal screen.

Configure the action as follows.



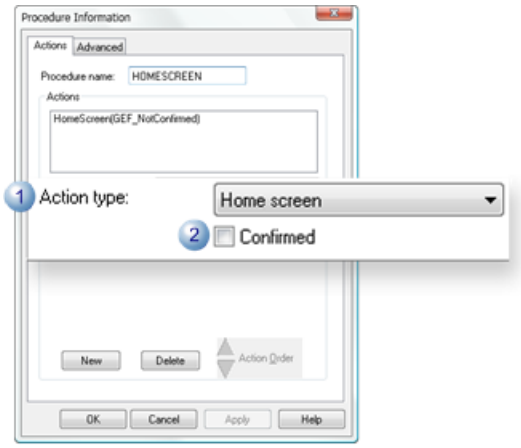
	Feature	Description	
1	Action type	Select Close screen.	
2	Confirmed	(Optional) Runtime confirmation	
		Checked	
		Clear	
		Default	

Home Screen (Terminal)

Home Screen

- Re-displays the first screen that was displayed when the user entered CimView.
- Is a terminal screen.

Configure the action as follows.



	Feature	Description	
1	Action type	Select Home screen.	
2	Confirmed	(Optional) Runtime confirmation	
		Checked	
		Clear	
		Default	

Next Screen

Next Screen

- Re-displays the screen that was overlaid directly after the current screen when moving forward and back among a series of screens.

The CimView stack records up to the last 100 screens that a user has displayed in a session.

- Is a terminal screen.

Configure the action as follows.

	Feature	Description	
1	Action type	Select Next screen.	
2	Confirmed	(Optional) Runtime confirmation	
		Checked	
		Clear	
		Default	

Open Faceplate

Open Faceplate enables you to specify precisely which screen should open, at what zoom static to the current screen, in which location; you can also define variables with initial values that the open faceplate will assume. However, your entries are optional.

Faceplate-specific screen variables

You can assign variables to a screen that are specific to faceplates.

$Pinned

You can assign a $Pinned variable to screens. When the screen is opened as a faceplate, the value of the $Pinned variable decides the behavior of the faceplate when another faceplate is opened.

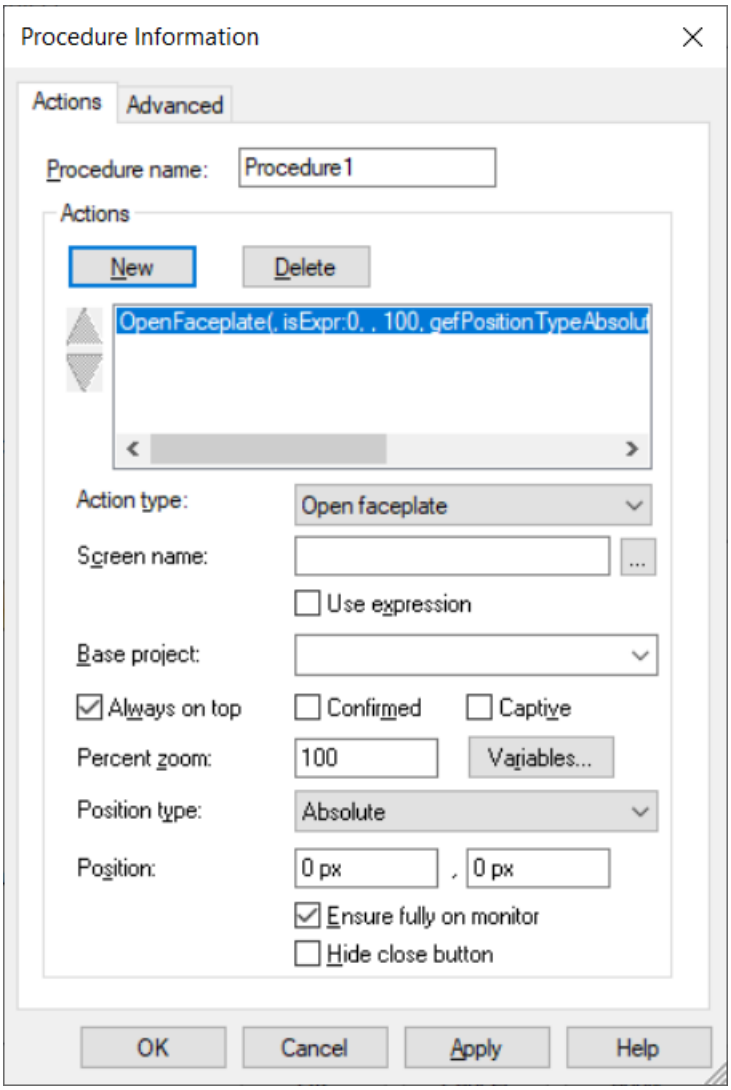
Value of $Pinned variable	Scenario	Result
0	Screen with $Pinned variable is opened as a faceplate, and then another faceplate is opened.	The first faceplate closes.
Non-zero value	Screen with $Pinned variable is opened as a faceplate, and then another faceplate is opened.	The first faceplate remains open.
$Pinned variable is not assigned.	Screen is opened as a faceplate, and then another faceplate is opened.	The first faceplate remains open.

Note: When you close a parent window, all of its child windows are also closed irrespective of whether they have the $Pinned variable assigned to them.

$Caption

You can assign a $Caption variable to screens. When the screen is opened as a faceplate, the title of the faceplate displays the value assigned to the $Caption variable.

Configure the action as follows.



1 *(page 497)*	Action type
2 *(page 497)*	Screen name
3 *(page 497)*	Use expression
4 *(page 497)*	Base project
5 *(page 498)*	Always on Top

6 *(page 498)*	Confirmed
7 *(page 498)*	Captive
8 *(page 498)*	Percent zoom
9 *(page 499)*	Variables
10 *(page 500)*	Position type
11 *(page 501)*	Position
12 *(page 501)*	Ensure fully on monitor
13 *(page 501)*	Hide close button

1	Action type

Select Open faceplate.

2	Screen name

Name of the CimView screen that will open when the action is triggered.

Default	Blank	CimView users specify the screen to open.

3	Use expression

When selected, Use expression displays options to assign the name of the screen through an expression.

Default	Clear	You can browse for the screen to be accessed.

4	Base project

A project name in the Base project field to qualify unqualified points.

Note: If the project is running during configuration all currently available projects are listed in the Base project drop down list. Any of the listed projects can be selected or another can be entered.

Default	Blank	Points are not associated with a project.

5	Always on Top

The window for the selected screen will open and stay in front of other windows as long as it is open.

This option insures that the selected screen will be visible (e.g if you have a toolbar, a main screen that is in full screen mode and the screen this will open is smaller, etc.).

Important: A user cannot turn off Always on Top during runtime.

6	Confirmed

When selected, Confirmed requires confirmation at runtime.

A message box is displayed before the action occurs requesting confirmation to proceed.

Default	Clear	Confirmation is not required.

7	Captive

When selected, Captive to prohibit a CimView user from performing functions in CimView until the current screen is closed.

Default	Clear	A CimView user can perform functions before the current screen is closed.

8	Percent zoom

Percent the screen will zoom to when it opens. The percent is based on the dimensions entered on the opening screen's Geometry tab.

Default	100	100%

Example

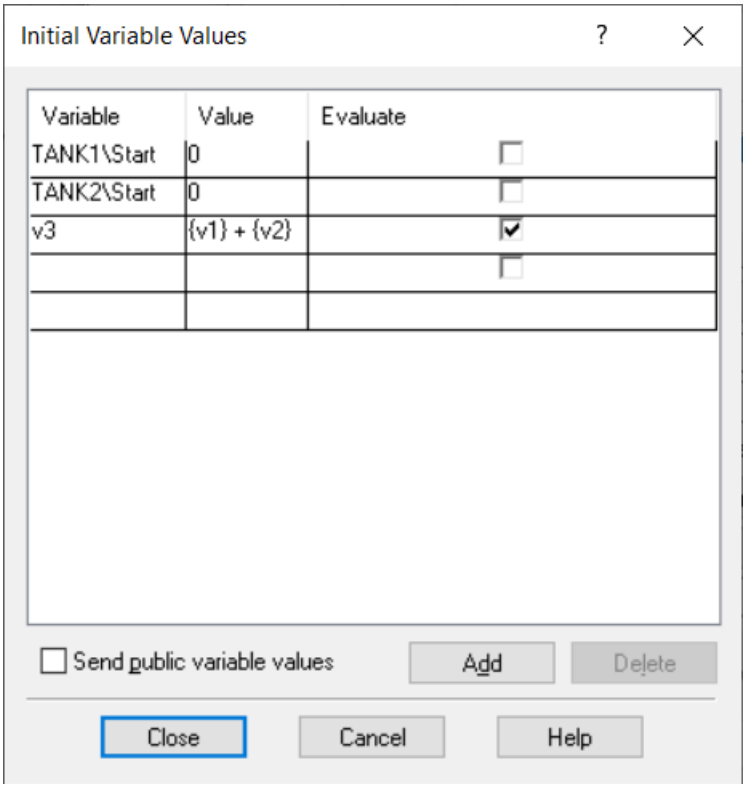
Screen Name	Open Screen

Open Screen	The dimensions on the Open Screen in the Properties dialog box Geometry tab are:	
	Width	
	Height	
Zoom	100	

The screen opens at 100%, a 300 x 300 dimension.

9	Variables

Opens an Initial Variable Values window. The open faceplate can take ownership of the variables listed in the window.



A	Variable

	Enter inputs that are: • Qualified variables with the objects that have the specified names. If there are no objects with names that match a qualified variable object name, that variable is disregarded. • Unqualified variables.
B	Value
	Enter a value or an expression to be assigned to a variable at run time.
C	Evaluate
	Select this check box to evaluate an expression and assign the value to a variable at run time.
D	Send public variable values
	Select this check box to set the values of screen level variables to variables with the same name in the target screen.

Example

TANK1\Start is a qualified variable

Where

TANK1 is the object name.

Start is the variable.

v3 is an unqualified variable.

10	Position type

Qualifying entries depend on the selected Position Type option.

For following selections specify the initial position X, Y for the top left corner of the open faceplate.

Type	Open Screen Top Left Corner	Positive values Move
Absolute	Top left corner (0, 0) of the monitor screen. (Default)	X
		Y
CimView screen relative	Bottom left corner (0, 0) of the current CimEdit screen.	X
		Y
Object relative	Top left corner (0, 0) of the "event trigger object."	X
		Y

Geometry String	Top left corner (0,0) of the default monitor. The Geometry String offers several options for specifying the open screen's: • Size. • Position. • Display surface. Specifications are made in the Geometry field that displays when Geometry String is selected.	

Note: The Event Trigger Object is the object on which the event is triggered. It is not necessarily the object through which you are configuring the procedure.

11	Position

Displays when the following position types are selected.

- Absolute
- CimView screen static
- Object static

X and Y positions.

- Absolute positions are in pixels.
- static positions are in points.

Default	0, 0	For the Absolute position type.

Tip: If you choose Object static, use the Geometry tab of the Event Trigger object to calculate your entries in the Position fields.

12	Ensure fully on monitor

When selected, Ensure fully on monitor ensures that the opened faceplate is completely visible on the monitor.

Default	Selected	The opened faceplate is completely visible on the monitor.

13	Hide close button

When selected, Hide close button removes the close button from the title bar of the opened faceplate.

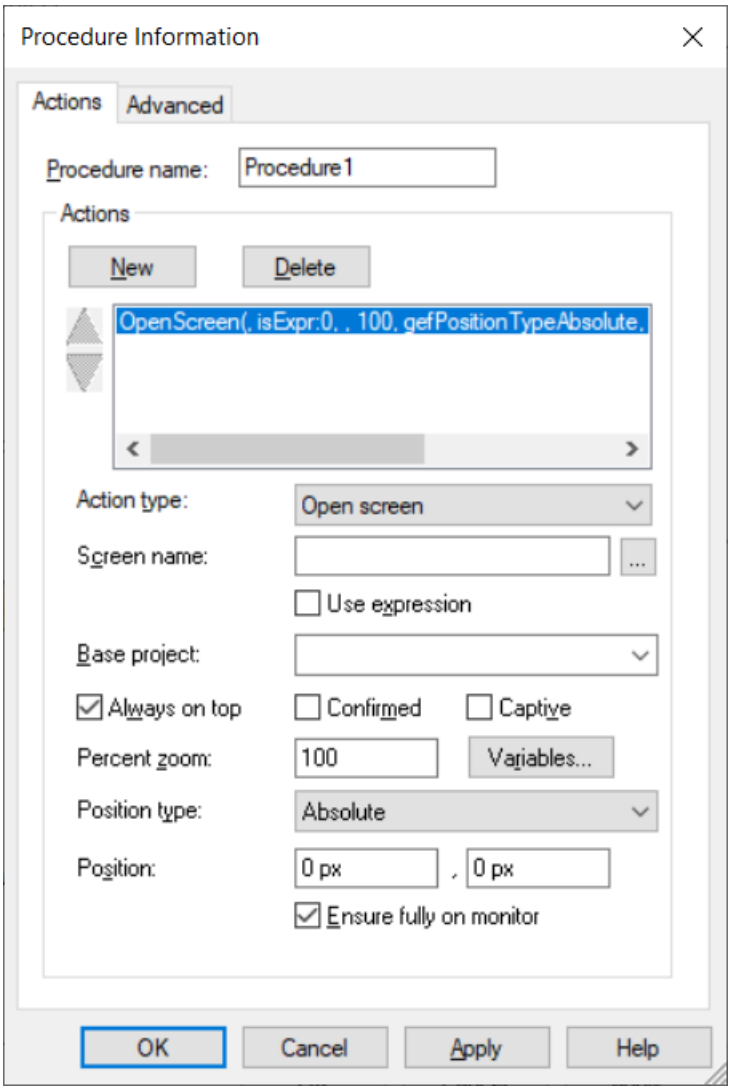
Default	Clear	The title bar of the opened faceplate contains the close button.

Open Screen

Open Screen displays a different CimView screen in another window.

Open Screen enables you to specify precisely what screen should open, at what zoom static to the current screen, in what location; you can also define variables with initial values that the open screen will assume. However, your entries are optional.

Configure the action as follows.



rect -2, 97, 24, 123 *(page 504)*
rect -4, 123, 22, 149 *(page 504)*
rect -4, 149, 22, 175 *(page 505)*
rect 85, 271, 111, 297 *(page)*
rect -2, 249, 24, 275 *(page)*
rect -2, 222, 24, 248 *(page)*
rect 195, 196, 221, 222 *(page 506)*
rect -2, 199, 24, 225 *(page)*
rect 194, 171, 220, 197 *(page)*
rect 108, 172, 134, 198 *(page 505)*
rect -1, 174, 25, 200 *(page 505)*

1 *(page 504)*	Action type

2 *(page 504)*	Screen name
3 *(page 504)*	Use expression
4 *(page 505)*	Base project
5 *(page 505)*	Always on Top
6 *(page 505)*	Confirmed
7 *(page 505)*	Captive
8 *(page 505)*	Percent zoom
9 *(page 506)*	Variables
10 *(page 507)*	Position type
11 *(page 508)*	Position
12 *(page 508)*	Ensure fully on monitor

1	Action type

Select Open screen.

2	Screen name

Name of the CimView screen that will open when the action is triggered.

Default	Blank	CimView users specify the screen to open.

3	Use Expression

When selected, Use expression displays options to assign the name of the screen through an expression.

Default	Clear	You can browse for the screen to be accessed.

4	Base project

A project name in the **Base project** field to qualify unqualified points.

Note: If the project is running during configuration all currently available projects are listed in the Base project drop down list. Any of the listed projects can be selected or another can be entered.

Default	Blank	Points are not associated with a project.

5	Always on Top

The window for the selected screen will open and stay in front of other windows as long as it is open.

This option insures that the selected screen will be visible (e.g if you have a toolbar, a main screen that is in full screen mode and the screen this will open is smaller, etc.).

Important: A user cannot turn off Always on Top during runtime.

6	Confirmed

When checked, Confirmed requires confirmation at runtime.

A message box will display before the action occurs requesting confirmation to proceed.

Default	Clear	Confirmation is not required.

7	Captive

When checked, Captive to prohibit a CimView user from performing functions in CimView until the current screen is closed.

Default	Clear	A CimView user can perform functions before the current screen is closed.

8	Percent zoom

Percent the screen will zoom to when it opens. The percent is based on the dimensions entered on the opening screen's Geometry tab.

Default	100	100%

Example

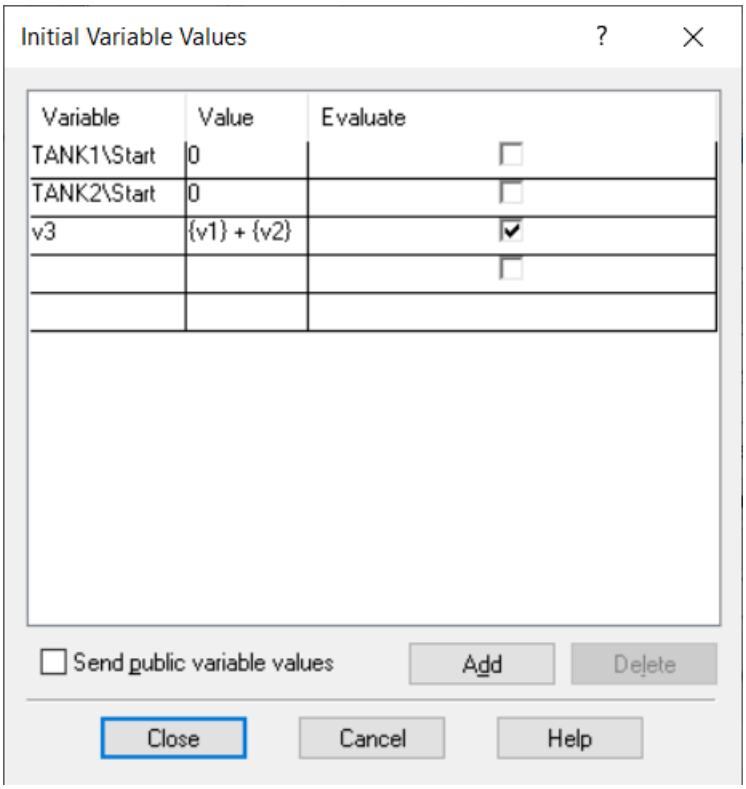
Screen Name	Open Screen
Open Screen	The dimensions on the Open Screen in the Properties dialog box Geometry tab are:
	Width

	Height
Zoom	100

The screen opens at 100%, a 300 x 300 dimension.

9	Variables

Opens an Initial Variable Values dialog box. The open screen can take ownership of the variables listed in the dialog box.



The screen takes ownership of:

A	Variable
	Enter inputs that are: • Qualified variables with the objects that have the specified names. If there are no objects with names that match a qualified variable object name, that variable is disregarded. • Unqualified variables.
B	Value
	Enter a value or an expression to be assigned to a variable at run time.

C	Evaluate
	Select this check box to evaluate an expression and assign the value to a variable at run time.
D	Send public variable values
	Select this check box to set the values of screen level variables to variables with the same name in the target screen.

Example

TANK1\Start is a qualified variable

Where

TANK1 is the object name.

Start is the variable.

v3 is an unqualified variable.

10	Position type

Qualifying entries depend on the selected **Position Type** option.

For following selections specify the initial position X, Y for the top left corner of the open screen.

Type	Open Screen Top Left Corner	Positive values Move
Absolute	Top left corner (0, 0) of the monitor screen. (Default**)**	**X**
		Y
CimView screen relative	Bottom left corner (0, 0) of the current CimEdit screen.	**X**
		Y
Object relative	Top left corner (0, 0) of the "event trigger object."	**X**
		Y
Geometry String	Top left corner (0,0) of the default monitor. The Geometry String offers several options for specifying the open screen's: • Size. • Position. • Display surface. Specifications are made in the **Geometry** field that displays when Geometry String is selected.	

Note: The Event Trigger Object is the object on which the event is triggered. It is not necessarily the object through which you are configuring the procedure.

11	Position

Displays when the following position types are selected.

- Absolute
- CimView screen static
- Object static

X and Y positions.

- Absolute positions are in pixels.
- static positions are in points.

Default	0, 0	For the Absolute position type.

Tip: If you choose Object static, use the Geometry tab of the Event Trigger object to calculate your entries in the Position fields.

12	Ensure fully on monitor

When selected, Ensure fully on monitor ensures that the opened screen is completely visible on the monitor.

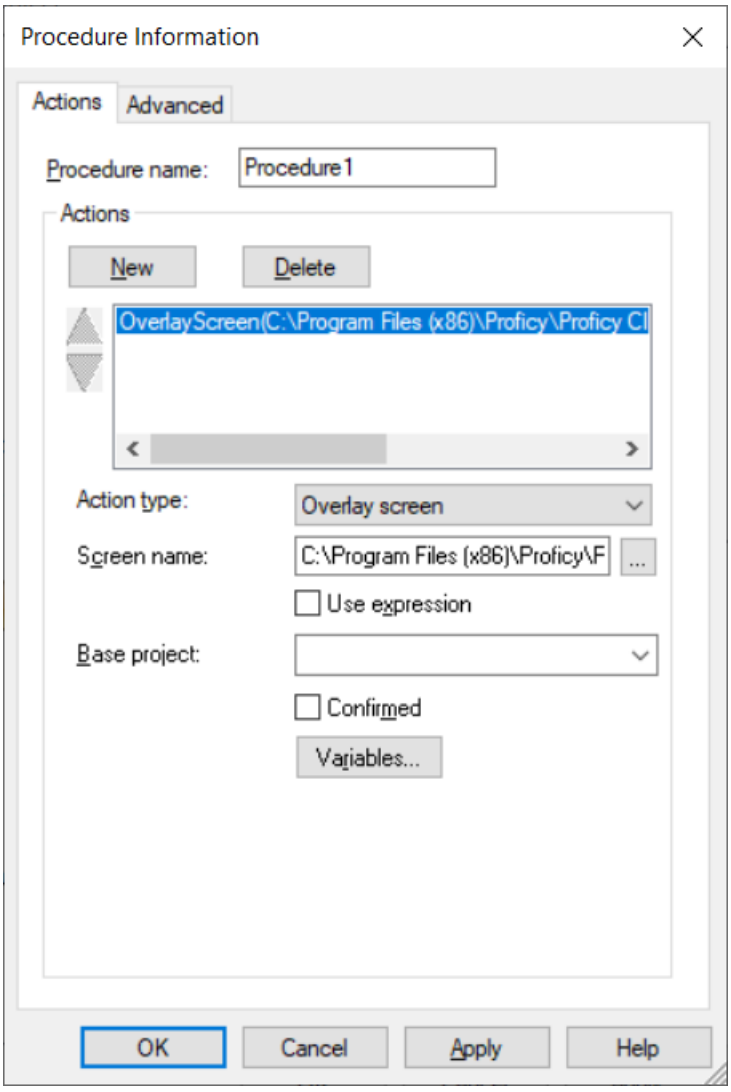
Default	Selected	The opened screen is completely visible on the monitor.

Overlay Screen (Terminal)

Overlay Screen

- Overlays the current screen with the designated screen.
- Is a terminal screen.

Configure the action as follows.



rect 107, 211, 130, 235 *(page 510)*
rect 106, 183, 129, 207 *(page 510)*
rect 0, 162, 23, 186 *(page 510)*
rect -1, 135, 22, 159 *(page 510)*
rect -1, 109, 22, 133 *(page 510)*

1 *(page 510)*	Action type
2 *(page 510)*	Screen name
3 *(page 510)*	Use Expression

4 *(page 510)*	Base project
5 *(page 510)*	Confirmed
6 *(page 510)*	Variables

1	Action type

Select Overlay screen.

2	Screen name

Name of the CimView screen that will overlay the current screen when the action is triggered.

Default	Blank	CimView users specify the screen that will overlay the current screen.

3	Use Expression

When selected, Use expression displays options to assign the name of the screen through an expression.

Default	Clear	You can browse for the screen to be accessed.

4	Base project

A project name in the **Base project** field to qualify unqualified points.

Note: If the project is running during configuration all currently available projects are listed in the Base project drop down list. Any of the listed projects can be selected or another can be entered.

Default	Blank	Points are not associated with a project.

5	Confirmed

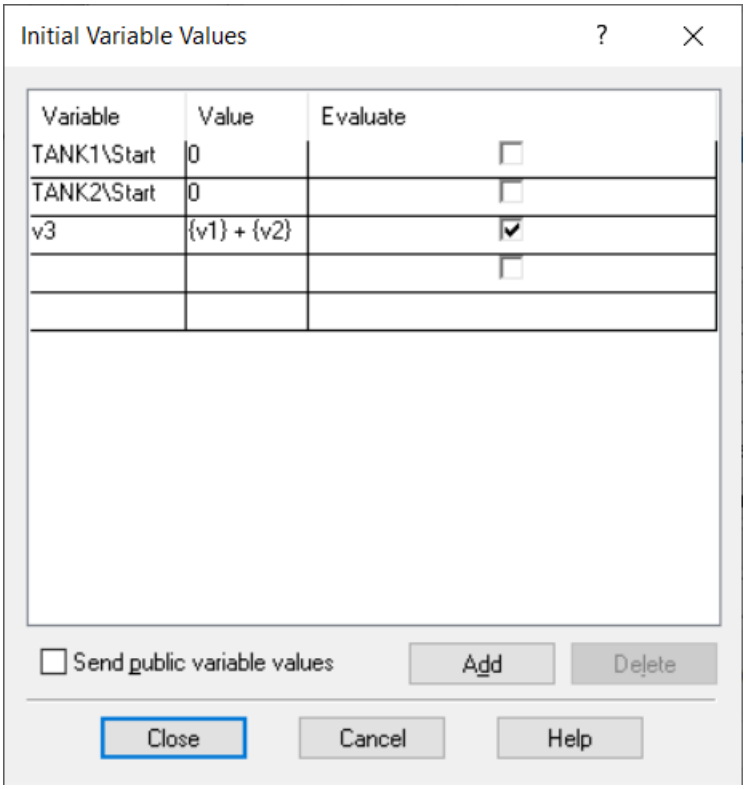
When checked, Confirmed requires confirmation at runtime.

A message box will display before the action occurs requesting confirmation to proceed.

Default	Clear	Confirmation is not required.

6	Variables

Opens an Initial Variable Values dialog box. The overlay screen can take ownership of the variables listed in the dialog box.



The screen takes ownership of:

A	Variable
	Enter inputs that are: • Qualified variables with the objects that have the specified names. If there are no objects with names that match a qualified variable object name, that variable is disregarded. • Unqualified variables.
B	Value
	Enter a value or an expression to be assigned to a variable at run time.
C	Evaluate
	Select this check box to evaluate an expression and assign the value to a variable at run time.
D	Send public variable values
	Select this check box to set the values of screen level variables to variables with the same name in the target screen.

Example

TANK1 \Start is a qualified variable

Where

TANK1 is the object name.

Start is the variable.

v3 is an unqualified variable.

Previous Screen (Terminal)

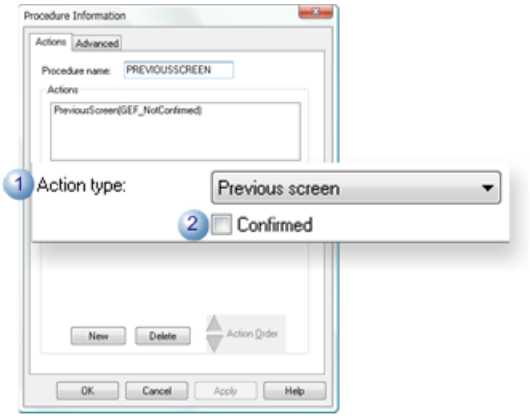
Previous Screen:

- Overlays the current screen with the previous screen that was displayed in the current window.

The CimView stack records up to the last 100 screens that a user has displayed in a session.

- Is a terminal screen.

Configure the action as follows.

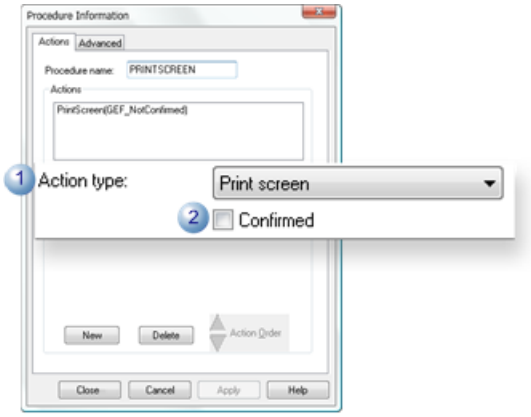


	Feature	Description	
1	Action type	Select Previous screen.	
2	Confirmed	(Optional) Runtime confirmation	
		Checked	
		Clear	
		Default	

Print Screen

Print Screen prints the current screen through the default printer.

Configure the action as follows.



	Feature	Description	
1	Action type	Select Print screen.	
2	Confirmed	(Optional) Runtime confirmation	
		Checked	
		Clear	
		Default	

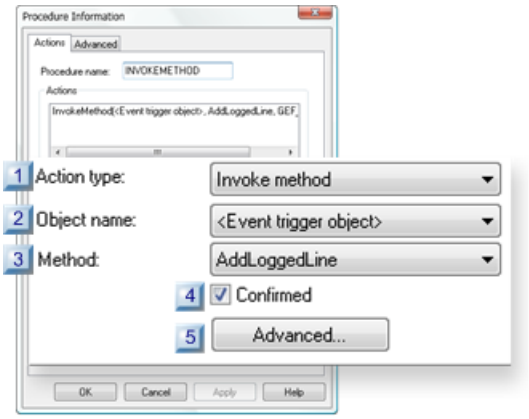
Invoke Method

Invoke Method invokes a designated method.

Currently, methods are supported for CIMPLICITY Trends, SPC Charts, Recipes, and Alarm Viewer controls.

Note: See the documentation for these controls for more information.

Configure the action as follows.



rect 104, 196, 129, 222 *(page 515)*
rect 104, 168, 129, 194 *(page 515)*
rect -6, 146, 19, 172 *(page 514)*
rect -4, 120, 21, 146 *(page 514)*
rect -4, 93, 21, 119 *(page 514)*

1 *(page 514)*	Action type
2 *(page 514)*	Object name
3 *(page 514)*	Method
4 *(page 515)*	Confirmed
5 *(page 515)*	Advanced

1	Action type

Select Invoke method.

2	Object name

Available control objects are listed.

If the object is named on the Properties dialog box General tab, the name is listed.

Default	<Event trigger object>

3	Method

Methods available for the control object being configured are listed.

4	Confirmed

When checked, Confirmed requires confirmation at runtime.

A message box will display before the action occurs requesting confirmation to proceed.

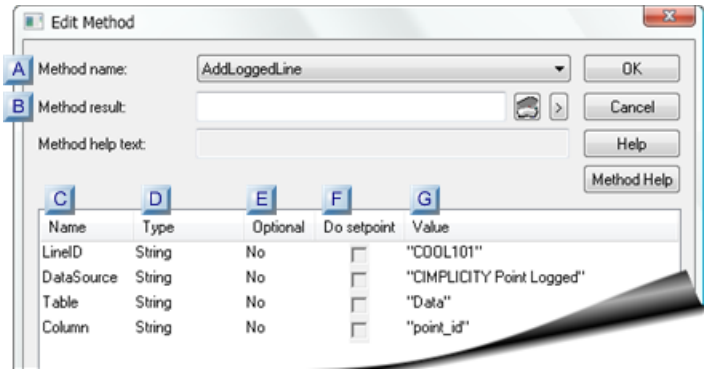
Default	Clear	Confirmation is not required.

5	Advanced

Advanced is enabled for methods that require further specifications.

The Edit Method dialog box for that method opens. The information and format that you enter in the Edit Method dialog depends on the method you select.

General configuration includes.



rect 253, 110, 275, 133 *(page 516)*
rect 196, 110, 218, 133 *(page 516)*
rect 150, 110, 172, 133 *(page 516)*
rect 84, 110, 106, 133 *(page 515)*
rect 24, 110, 46, 133 *(page 515)*
rect -2, 52, 20, 75 *(page 515)*
rect -2, 28, 20, 51 *(page 515)*

	Field	Description
A	Method name	Method selected on the Actions tab. Another method can be selected from the list of available methods.
B	Method result	(Optional) Enter a Setpoint in the **Method result** field if the function returns a value. You can use a device or global Point ID with the correct Point Type for the status value
C	Name	Parameters to be configured for the method.
D	Type	Format required for the corresponding entry in the **Value** field.

	Field	Description
		Integer
		String
E	Optional	Specifies if a **Value** entry is optional.
		No
		Yes
F	Do setpoint	Options are:
		Disabled
		Enabled
		If the argument can be used as an output argument, the **Do Setpoint** checkbox will be enabled. To assign the output value of the argument to a CIMPLICITY Point when the function exits, enter the Point ID in the **Expression** field.
G	Value	Expression that will pass as the argument for each argument in the **Name** column.

Invoke Script

Invoke Script execute the designated script in CimView.

When the script is invoked at runtime:

- If you select an object name in the **Object name** field, then the action looks first in the script for that object.
- If you select **<Event trigger object>** in the **Object name** field, then the action looks first in the script for the object whose event triggered the procedure.

In either case, if the entry point does not exist in that object's script, then the action looks for it in the script for that object's container, and so on, until it reaches the screen's script. If the entry point is not found, the action is skipped.

- Invoke Script (basic procedure)
- Invoke Script (advanced entries)

Setpoint Actions

Tip: CimEdit provides you with several ways to configure set points. They include:

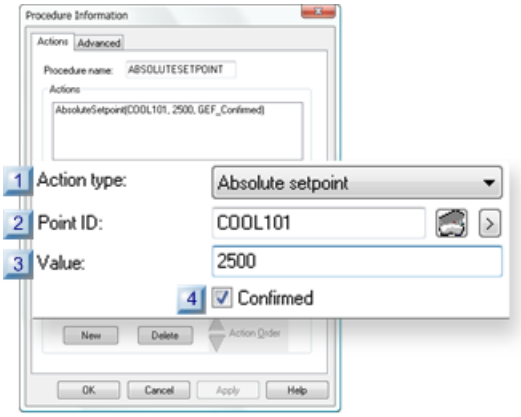
Setpoint actions described here.

- Slider action check boxes on the Movement tab in the Properties dialog box.
- Setpoint action check boxes on the Text tab in the Properties dialog box.

Absolute Setpoint

Absolute Setpoint identifies a Point ID and a value to associate with the object. When the procedure is invoked in CimView, the value will be downloaded to the Point ID.

Configure the action as follows.



rect 104, 174, 130, 196 *(page 518)*
rect -4, 152, 22, 174 *(page 518)*
rect -4, 126, 22, 148 *(page 517)*
rect -4, 100, 22, 122 *(page 517)*

1 *(page 517)*	Action type
2 *(page 517)*	Point ID
3 *(page 518)*	Value
4 *(page 518)*	Confirmed

1	Action type

Select Absolute setpoint.

2	Point ID

Point that will be set when the action is triggered.

The:

- Value you select must be compatible with the point's type.
- Point must be a Read/Write point.

	Opens the Select a Point browser.
	Opens a Points Popup menu.

3	Value

Value the point will be set to when the action is triggered.

4	Confirmed

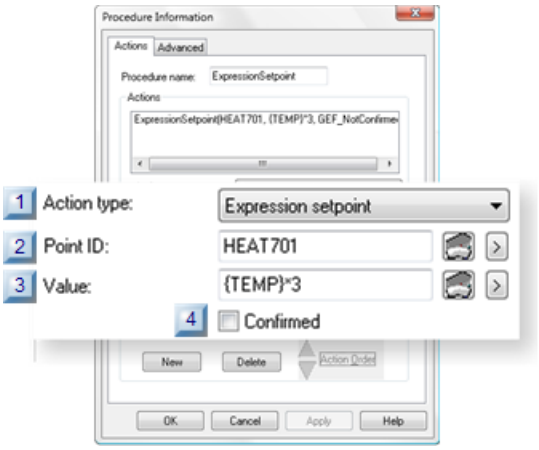
(Optional) Runtime confirmation.

Checked	Requires confirmation. A message box will display before the action occurs requesting confirmation to proceed.
Clear	Does not require confirmation.
Default	Clear

Expression Setpoint

Expression Setpoint identifies a Point ID and an expression. When the procedure is invoked in CimView, the expression is evaluated using the trigger object to qualify variables in the expression. After the successful evaluation of the expression the value is downloaded to the Point ID.

Configure the action as follows.



rect 107, 184, 129, 207 *(page 520)*
rect -1, 164, 21, 187 *(page 520)*
rect -2, 140, 20, 163 *(page 519)*
rect 0, 111, 22, 134 *(page 519)*

1 *(page 519)*	Action type
2 *(page 519)*	Point ID
3 *(page 520)*	Value
4 *(page 520)*	Confirmed

1	Action type

Select Expression setpoint.

2	Point ID

Point that will be set when the action is triggered.

The:

- Value you select must be compatible with the point's type.
- Point must be a Read/Write point.

	Opens the Select a Point browser.
	Opens a Points Popup menu.

3	Value

Value the point will be set to when the action is triggered.

Valid entries are the same as valid entries for any field that accepts expressions, including:

- Point ID's.
- Variables.
-
- Absolute values.
- Expression operations.

4	Confirmed

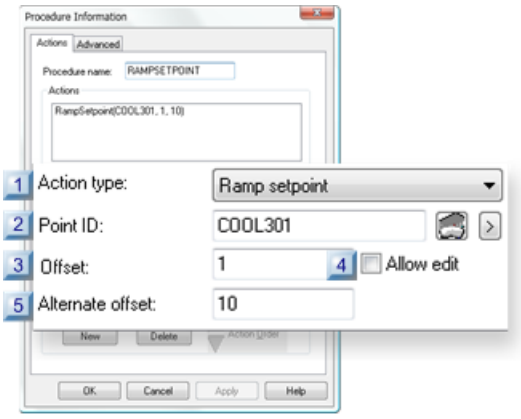
(Optional) Runtime confirmation.

Checked	Requires confirmation. A message box will display before the action occurs requesting confirmation to proceed.
Clear	Does not require confirmation.
Default	Clear

Ramp Setpoint

Ramp Setpoint enables users to add or subtract the standard and alternate values from the current value of a point, then perform a setpoint on the point using the modified value.

Configure the action as follows.



rect 198, 149, 222, 177 *(page 521)*
rect -4, 174, 20, 202 *(page 521)*
rect -4, 148, 20, 176 *(page 521)*
rect -4, 122, 20, 150 *(page 521)*
rect -4, 98, 20, 124 *(page 521)*

1 *(page 521)*	Action type
2 *(page 521)*	Point ID
3 *(page 521)*	Offset
4 *(page 521)*	Allow edit
5 *(page 521)*	Alternate offset

1	Action type

Select Ramp setpoint.

2	Point ID

Point on which a setpoint is performed.

- Offset values must be integers or real numbers.
- Point must be a Read/Write point.

	Opens the Select a Point browser.
	Opens a Points Popup menu.

3	Offset

Number added or subtracted from the point's value when a user

- Clicks the Up/Down arrows in the default box.
- Presses the Up/Down arrow keys.

4	Allow edit

Check to activate the runtime New Value edit field so a user can type a new offset value.

Checked or clear a user can use the Up/Down arrows in the Default and Alternate boxes to enter a new setpoint.

5	Alternate offset

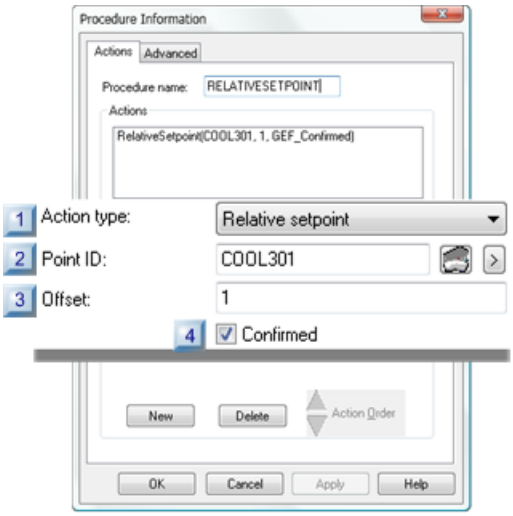
Number added or subtracted from the point's value when a user:

- Clicks the Up/Down arrows in the Alternate box.
- Presses the Left/Right arrow keys.

Relative Setpoint

Relative setpoint adds the offset value to the current value of the point. The result is downloaded to the point.

Configure the action as follows.



rect 107, 197, 129, 213 *(page 523)*
rect 1, 175, 23, 197 *(page 523)*
rect 0, 146, 22, 168 *(page 523)*
rect 0, 122, 22, 144 *(page 522)*

1 *(page 522)*	Action type
2 *(page 523)*	Point ID
3 *(page 523)*	Offset
4 *(page 523)*	Confirmed

1	Action type

Select Relative setpoint.

2	Point ID

Point on which a setpoint is performed.

- Offset values must be integers or real numbers.
- Point must be a Read/Write point.

	Opens the Select a Point browser.
	Opens a Points Popup menu.

3	Offset

Number added or subtracted from the point's value when a user

- Clicks the Up/Down arrows in the default box.
- Presses the Up/Down arrow keys.

4	Confirmed

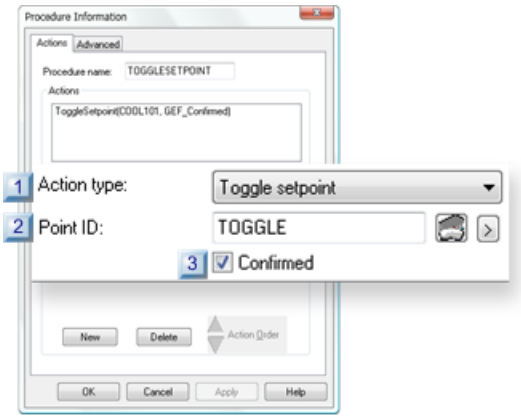
(Optional) Runtime confirmation.

Checked	Requires confirmation. A message box will display before the action occurs requesting confirmation to proceed.
Clear	Does not require confirmation.
Default	Clear

Toggle Setpoint

Toggle Setpoint toggles the current value of a Boolean point. The result is downloaded to the point.

Configure the action as follows.



rect 106, 149, 133, 176 *(page 524)*
rect -4, 125, 23, 152 *(page 524)*
rect -4, 98, 23, 125 *(page 524)*

1	Action type

Select Toggle setpoint.

2	Point ID

Point on which a setpoint is performed.

The point must be a:

- Boolean point.
- Read/Write point.

	Opens the Select a Point browser.
	Opens a Points Popup menu.

3	Confirmed

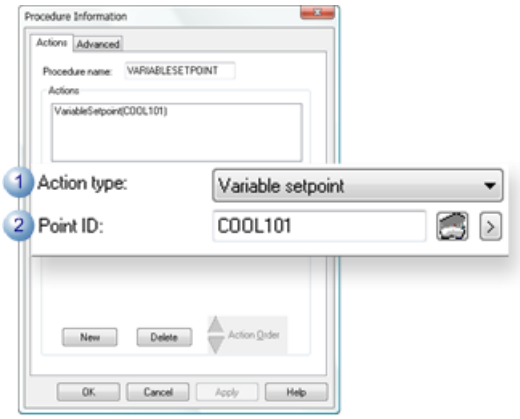
(Optional) Runtime confirmation.

Checked	Requires confirmation. A message box will display before the action occurs requesting confirmation to proceed.
Clear	Does not require confirmation.
Default	Clear

Variable Setpoint

Variable Setpoint displays a dialog box that prompts users for a value to download to the point.

Configure the action as follows.

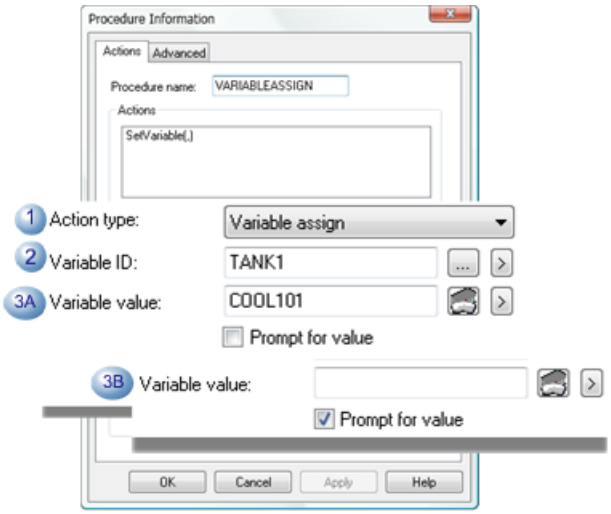


	Field	Description
1	Action type	Select Variable setpoint.
2	Point ID	Point on which the setpoint is performed.
		The point must be a Read/ Write point.

Variable Assign

Variable Assign assigns a Variable ID to a value or Point ID.

Configure the action as follows.



	Field	Description
1	Action type	Select Variable assign.
2	Variable ID	Variable on which the assignment is performed.
Specify either 3A or 3B.		

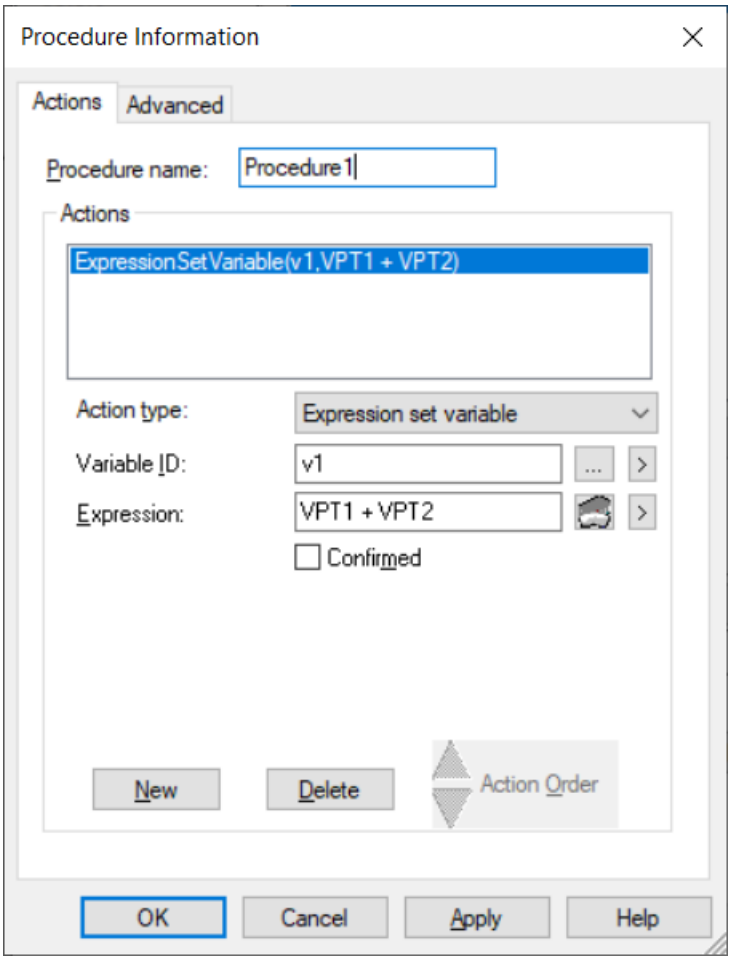
	Field	Description
3A	Variable value	Text value that is not evaluated but is literally substituted for the Variable ID. The value specification can include but is not limited to, a point ID, an expression that includes point ID's or absolute values. **Important:** If you enter a Variable value a user will not be able to change the value during runtime.

	Field	Description
3B	Prompt for value	(Optional) When checked and the action is triggered during runtime, a Variable Assignment *(page 789)* dialog box will open in which a user can enter a new value. **Important:** This feature will not work if the **Variable value** field has been filled in.

Expression Set Variable

Expression Set Variable evaluates an expression and assigns the evaluated value to a variable.

Configure the action as follows.



Action type

Select Expression set variable.

Variable ID

Specify a variable that will store the value of the evaluated expression.

Icon	Description
...	Opens the Select Variable window.
>	Opens a menu displaying the list of variables you can select.

Expression

Specify an expression that will be evaluated when the action is triggered.

Icon	Description
	Opens the Build Expression window.
	Opens a menu where you can browse for points and select expression functions.

Confirmed

(Optional) Select this check box to receive confirmation at run time.

Option	Description
Selected	Requires confirmation. A message box appears before the action occurs, requesting confirmation to proceed.
Cleared	Does not require confirmation.
Default	Cleared

CimEdit Scripts

CimEdit Scripts

CIMPLICITY provides several tools to make things happen in CimEdit and in CimView. The tools can be categorized as triggers that are configured as events and results that can be configured through procedures or scripts.

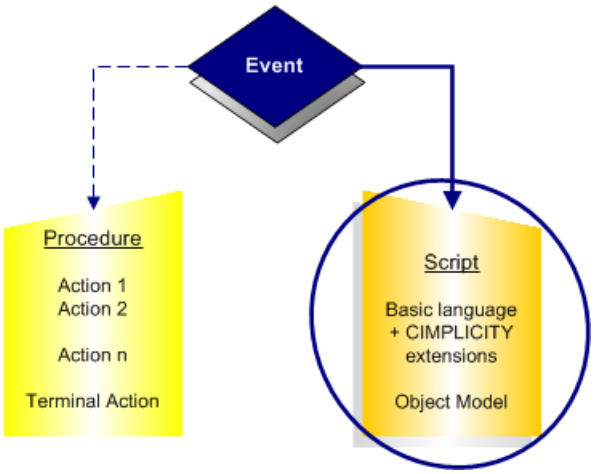
Trigger

An event *(page 432)* triggers a procedure or calls a script. CimEdit provides a long list of events from which you can choose the best one for your requirements.

Result

A procedure *(page 477)* contains one or more actions that are triggered in the specified order when an event occurs and while the screen is displayed in CimView. CimEdit provides several actions from which a screen designer can easily compile a meaningful list.

A script, which is usually written by more advanced users, uses the same Editor and Basic language as the Basic Control Engine. Anything you can do in a normal script, you can do in a **CimEdit** script. CimEdit provides additional extensions to give you a wider range of screen development choices. However, **CimEdit** scripts are only accessible from the screen in which you create them.



rect -2, 132, 117, 276 CimEdit Procedures *(page 477)*
rect 115, -3, 228, 84 CimEdit Events *(page 432)*

In addition, CimEdit scripting has implemented several automated objects. These automated objects can only be used by CimEdit scripts. You can create a script for any object, group, or frame container on your CimEdit screen. You can also create a script for the screen itself.

CimEdit scripting uses the same Editor and Basic language as the Basic Control Engine. Anything you can do in a normal script, you can do in a CimEdit script.

Tip: If you want to use objects with the same scripts on different screens, link them.

For unlinked objects, CimEdit scripts are only accessible from the screen in which you create them. If you have any questions on how to use the Editor or the Basic language, consult the following documents:

Script Configuration

Script Configuration

In CimEdit you can work with:

1 *(page 532)*	Work with CimEdit scripts.
2 *(page 540)*	Work with global scripts through CimEdit.
3 *(page 543)*	Test CimEdit scripts.

Note about Static Objects and Using the Object Model

Note: There is a notion in CimEdit of static objects. These are objects that have no animations, no actions, no help text and no name. The objects are compiled into very small representations that cannot be manipulated by the object model. In fact, they won't even show up in the GefObjects collection.

To make a static object accessible from the object model:

For a single object

1. Open the object's Properties dialog box.
2. Select the General tab.
3. Enter a name in the **Object name** field.

   For an object in a group

   If any object in a group has an:

   - Animation,
   - Action,
   - Help, or
   - Name,

   then every object in that group will be accessible from the object model.

1. Work with CimEdit Scripts

1. Work with CimEdit Scripts

CimEdit scripts exist in the context of a CimEdit object.

1.1 *(page 532)*	Open an Edit Script window.
1.2 *(page 534)*	Configure a CimEdit script.
1.3 *(page 537)*	Invoke the CimEdit script.

1.1. Open an Edit Script Window

CimEdit scripts can be created or edited for a selected object through its Properties dialog box.

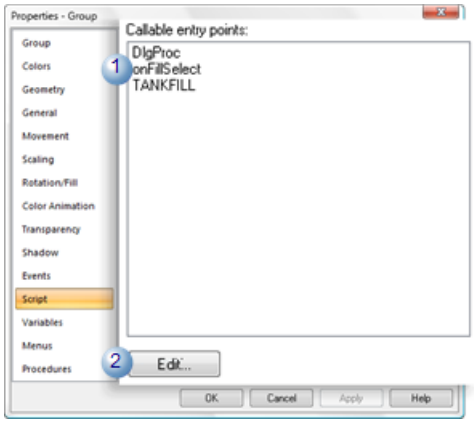
Note: Global scripts *(page 561)* can be assigned directly to a screen.

- Script
- Events

Script

Select **Script** in the Properties dialog box.

Scripts can be created or edited as follows.

1. 

1	Select an Entry point if there are any.
2	Click **Edit**.

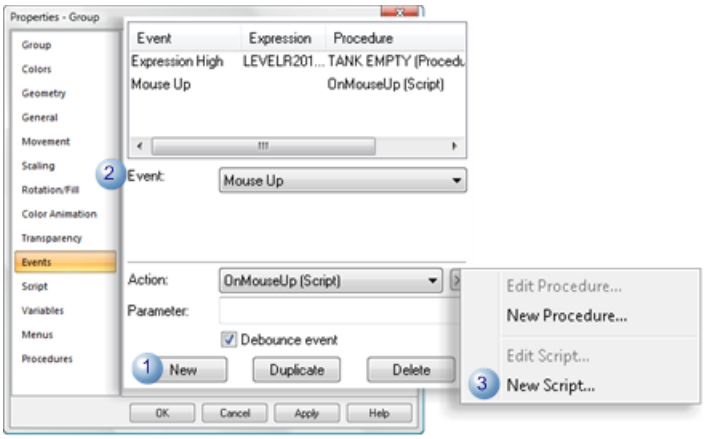
Result: The Edit Script window opens the script for the object. The script is scrolled to the selected entry point.

If no entry point is selected, the Edit Script window opens at the first line of the script.

Events

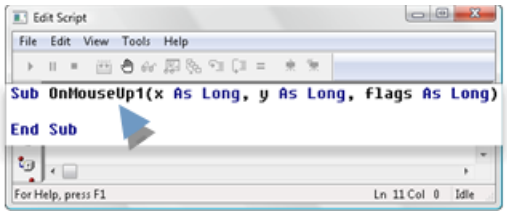
Select **Events**.

A script can be applied to a selected event.



	Field/Button	Description/Action
1	New button	(If an event is already listed) click to enable the **Event** field.
2	Event	Selection from the drop down menu triggers the action.
4	Popup menu	Select to open the Edit Script window. Note: Opens when the Popup menu button is clicked.

Result: An Edit Script window opens with an entry for the event's subroutine.



You can retain the default entry point name or change it.

Note: This window is similar to the window used by the Basic Control Engine Program Editor. Consult the Basic Control Engine documentation if you have any operational questions.

1.2. Configure a CimEdit Script

1.2. Configure a CimEdit Script

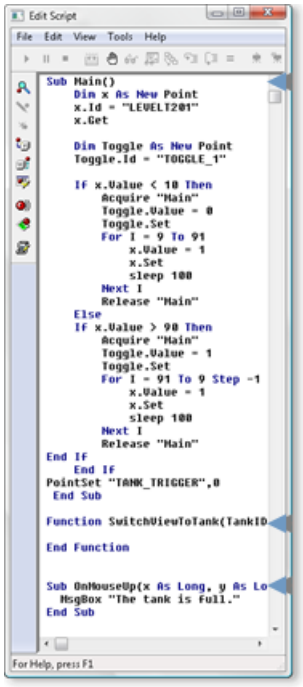
1. CIMPLICITY provides several references when you write the CimEdit script. They include:

CIMPLICITY Basic Control Engine Language Reference	Provides a dictionary of basic extension functions, operators and statements. These extensions can be used in scripts created in the CIMPLICITY script window..
Basic Extensions for CimEdit Scripts	Obsolete basic extensions and their replacements, in addition to describing current basic extensions. These extensions can be used in scripts created in the CimEdit script window.

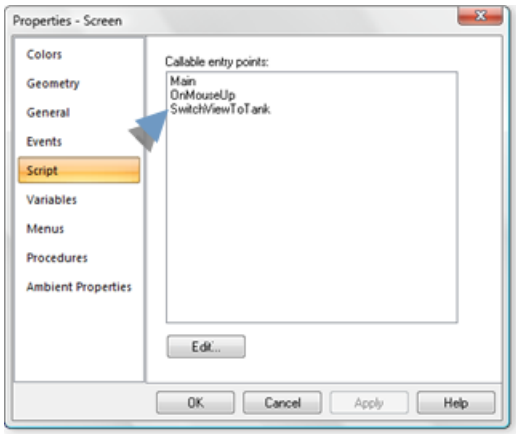
CimEdit/ CimView GefObject Model	Provides the CimEdit and CimView GefObject model, listing the properties, methods, objects and enums, which can be used in CIMPLICITY and other applications, such as C++, that have the capability to create OLE based objects. Note: Static objects and using the GefObject Model.
CIMPLICITY Configuration Object Model	CIMPLICITY server object model, listing objects and enums, which can be used in CIMPLICITY and other applications, such as C++, that have the capability to create OLE baed objects.

2. The script for a selected object can contain several entry points.

   Note: There are several ways that a script can access its entry points *(page 536)* .



3. The entry points are listed on the Scripts tab in the Properties dialog box.



Access Script Entry Points

There are several ways that a script can access its entry points.

An entry point can be a function or a subroutine.

Overall: A script can access entry points in its own script or any of its containers' scripts.

Example

You have an object with a script that is looking for entry point **xyz** .

The entry point is defined in the script for the group that contains the object.

Basic will execute the group's **xyz** entry point at runtime.

If	You define an entry point (in the scripts) for more than one of the object's containers.
Then	The entry point in the closest container to the object is executed.
Example	You have an object with a script that is looking for entry point **abc .** The entry point is defined in the script for the group that contains the object and in the script for the screen. Basic will execute the group's **abc** entry point at runtime.
If	You name an object that has a script.
Then	Entry points in that script can be accessed as follows: 1. Any object on the screen triggers an event. 2. The event invokes a procedure that has an Invoke Script action with the name of the object. 3. The script is accessed.
If	You configure a procedure to execute an **Invoke script** action for a particular entry point and use the <Event trigger object> as the object to look for entry points.
Then	Any object that implements the entry point in its script with the same parameters can use that procedure.
Example	Example You have two objects. Each object has its own script with an entry point **alpha . alpha** uses one input argument and returns a status Each **alpha** procedure does something different. If you create procedure **XYZ** to invoke **alpha** , you can assign it to events for both objects. When the event for the: 1. First object occurs, the **alpha** in the first object's script is executed. 2. Second object occurs, the **alpha** in the second object's script is executed.
If	Option 1 or Option 2
Then	A common script file can be shared among all screens loaded in CimView. Functions that are included in this common file will be loaded when CimView is first run and will be callable as if they were included in the Basic code for the local screen. If you utilize both options, CimView will load the contents of both files into memory. The functions specified in the `GSM_GLOBAL_SCRIPT` file will take precedence over those referenced by the `/LoadScript` option. If you specify the `/LoadScript` argument for a file after CimView is already running, its contents will be appended to the list of common functions with the lowest precedence.

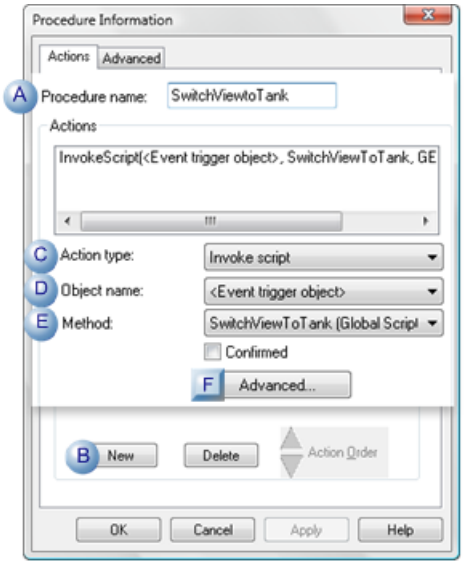
1.3. Invoke the CimEdit Script

1.3. Invoke the CimEdit Script

1.3.1 *(page 537)*	Invoke a script (basic procedure).
1.3.2 *(page 538)*	Invoke a CimEdit script (advanced entries).

1.3.1. Invoke a Script (Basic Procedure)

1. Select the object to which the script will be attached.
2. Open the object's Properties dialog box.
3. Select the Actions tab.
4. The fields are as follows.



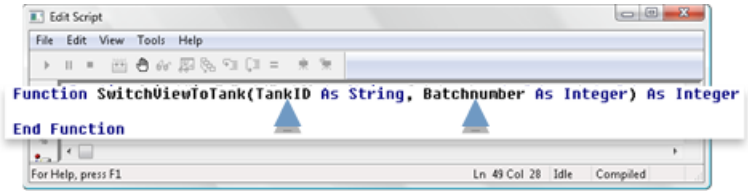
rect 118, 224, 140, 249 1.3.2. Invoke a CimEdit Script (Advanced Entries) *(page 538)*

	Field/ Button	Description/Action
A	Procedure name	Identifies the procedure being configured.
B	New button	(If an action already displays) click to activate the Action type field.

	Field/ Button	Description/Action
C	Invoke script	Action that invokes the selected CimEdit script.
D	Object Name	Name of object with which the script is associated. The object is named on the General tab of the Properties dialog box. **Default:** <Event Trigger object> When the script is invoked at run-time if you select: An object name or <Event Trigger object> . In either case, if the entry point does not exist in that object's script, then the action looks for it in the script of that object's container, and so on, until it reaches the screen's script. If the entry point is not found, the action is skipped.
E	Method	Script entry points available for the object are available in the drop down list.
F *(page 538)*	Advanced	Is enabled if the function returns a value that contains output arguments.

1.3.2. Invoke a CimEdit Script (Advanced Entries)

If you select a function to invoke a script and the function returns a value that contains arguments, you will have to do some easy, but advanced, configuration.

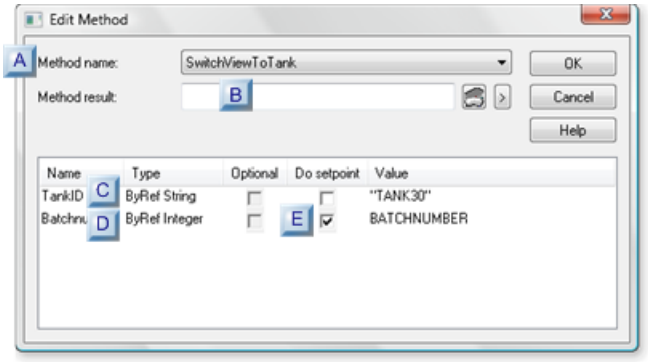


In these instances, the Advanced button on the Edit tab of the Properties dialog box will be enabled.

1. Click an enabled **Advanced… *(page 537)*** button in the Actions tab of the Properties dialog box.

   The Edit Method dialog box opens. The arguments correspond to arguments in the script function.

2. Enter values for the function's arguments.



rect 170, 122, 197, 147 *(page 539)*
rect 48, 125, 75, 150 *(page 539)*
rect 48, 102, 75, 127 *(page 539)*
rect 131, 44, 158, 69 *(page 539)*
rect -3, 23, 24, 48 *(page 539)*

	Field/ Type	Description
A	Method name	(Default) Method selected on the Action tab. You can select another method from the drop-down list. The listed arguments change or disappear to correspond with the newly selected method. Example: SwitchViewToTank
B	Method result	(Optional) A setpoint in the Method result field if the function returns a value can be a: • Device or global point ID with the correct Point Type for the status value. • Variable ID that evaluates to a point ID. If you use a variable ID that does not evaluate to a point ID, nothing happens.
C	ByRefString	Enter one: • An expression enclosed in double quotes. The results of the expression evaluation at runtime are passed to the method. • The point ID of a text point. The contents of the point ID are passed to the method. • A point ID for a non-text point enclosed in double quotes. The contents of the point ID are passed to the method.
D	ByRefInteger	Enter one: • An expression using the Edit Expression dialog box or the Popup menu. The results of the expression evaluation at runtime are passed to the method. • A Point ID. The contents of the point ID are passed to the method.
E	Do Setpoint	Enabled if the argument can be used as an output argument. The expression must consist of a single point ID for the setpoint to be performed.

3. Click **OK** to save your changes and return to the Actions tab.
4. Click **OK** to save the procedure information and return to the Events tab.
5. Click **OK** to save the event.

2. Work with Global Scripts through CimEdit

2. Work with Global Scripts through CimEdit

Global scripts are

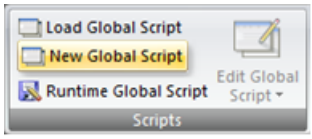
- Stored as external text (.cms, .bcl) files or .cmsrt *(page 543)* (compiled .cms).
- Accessible to all objects in all screens on a specific machine when they are loaded into CimEdit/ CimView.

- CimEdit global script *(page 561)* capability adds support to the: Command line options / LoadScript and /LoadScriptCfg, and Global parameters GSM_GLOBAL_SCRIPT and GSM_GLOBAL_SCRIPTCFG.

2.1 *(page 540)*	Create a global script.
2.2 *(page 541)*	Load a global script.
2.3 *(page 542)*	Edit a global script.
2.4 *(page 543)*	Create CimEdit runtime-only global scripts.

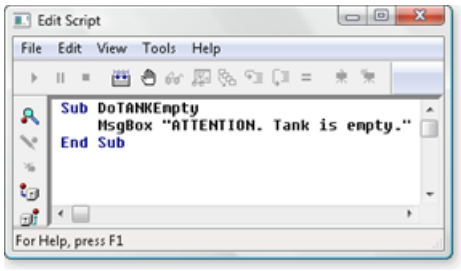
2.1. Create a Global Script

1. Click New Global Script in the Tools>Scripts group on the CimEdit Ribbon bar.



   A blank Edit Script window opens.

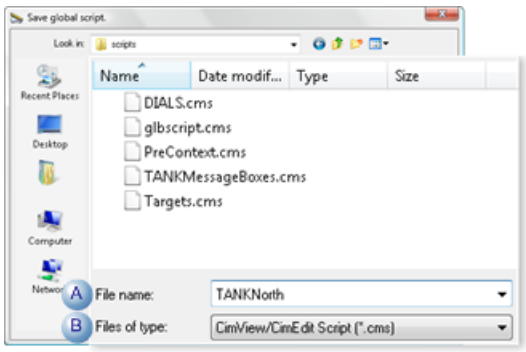
2. Write and compile the script.



3. Click File>Close on the Edit Script window menu bar.

   A Save global script. dialog box opens.

4. The selections are as follows.



	Field	Description
A	File name	Name for the saved file.
B	File of type	Option: .cms, .bcl, default

5. Click Save.

The script is saved as a global script.

2.2. Load a Global Script

1. Click Load Global Script in the Tools>Scripts group on the CimEdit Ribbon bar.

   The Select global script to load. dialog box opens.

   The selections are as follows.

Field	Description
File name	Selected file displays.

Field	Description
File or type	Filters files that display. Options are: • .bcl • .cms • .cmsrt • All files(*.*)

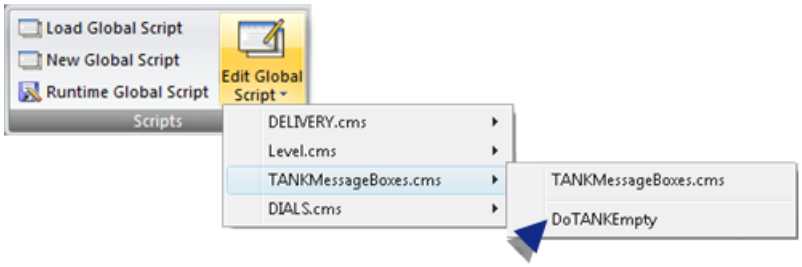
2. Click Open.

The selected file is loaded.

2.3. Edit a Global Script

1. Click Edit Global Script in the Tools>Scripts group on the CimEdit Ribbon bar.

   A dropdown list of loaded scripts displays.

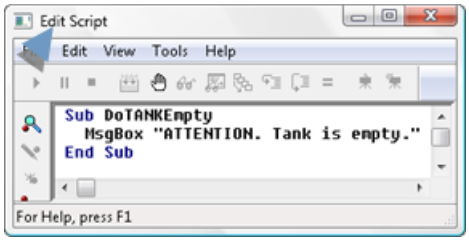


2. Expand the list.

   Note: Runtime-only script names display. However, the menu item is disabled.

3. Click a script>entry point to edit.

   The Edit script window opens with the selected script.



4. Edit and recompile the script.
5. Click File>Close on the Edit Script window.

The changes are saved and the Edit Script window closes.

2.4. Create CimEdit Runtime-only Global Scripts

You can protect your CimEdit scripts (intellectual property) by creating runtime-only scripts (.cmsrt).

Important: Runtime-only scripts cannot be opened. Therefore, save a read-write version of the script (.cms) when you create a runtime-only script so you can review and revise it, if necessary.

1. Click Runtime Global Script in the Tools>Scripts group on the CimEdit Ribbon bar.

   The Select source file for runtime only script. dialog box opens.

   The selections are as follows.

	Field	Description
A	File name	A runtime-only file will be created from the selected file.
B	Files of type	Filters files that display. Options are: • .bcl • .cms Files can be converted to runtime-only files. Both file types convert to .cmsrt. when you save them as runtime-only files in CimEdit.

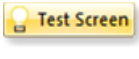
   Note: Runtime-only scripts that are created in the Program Editor (.bclrt) cannot be used in CimEdit.

2. Click Save.

The selected file is saved as a global runtime-only .cmsrt file.

3. Test CimEdit Scripts

1. Make sure you have done at least one of the following.
   - Set at least one breakpoint in the entry point you want to test.
   - Create a procedure that invokes the entry point you want to test.
   - Create an event that invokes the procedure.

2. Click **Test Screen** 

in the Home>Runtime group on the CimEdit Ribbon bar.

   The CimView runtime screen displays.

   Important: Global scripts that are loaded into CimEdit will not necessarily be loaded into CimView when you click the Test Screen button.

In order for a global script to be loaded into CimView when you click the Test Screen button one of the following has to be true.

- The script is named in the global parameter GSM_GLOBAL_SCRIPT or in the files specified by GSM_GLOBAL_SCRIPTCFG .
- CimView is already running and the script is loaded through command lines /loadscript *(page 634)* or /loadscriptcfg *(page 634)* .

3. Invoke the event that will start the script or procedure with the script that you want to test.

   When Basic comes to the first breakpoint in the script, it opens the Debug Script window.

   Note: You cannot debug a script during a regular CimView session.

4. At this point, you can:
   - Edit the code, change break points, recompile, restart, step into functions etc. in the Debug Script window. However, if you change the entry point that was called from the procedure, you will not be able to restart the script.
   - Open the Properties dialog box of an object and make changes to the script. However, you cannot execute the script from this editor.

5. Save the changed script in CimView to send the updated script back to CimEdit immediately.

   This does not save the script to the screen file.

6. Return to CimEdit.

7. Save the screen in order to save the screen file.

   Note: It is recommended that you close the Debug Script window before exiting Test Screen mode, or before going back to CimEdit and using the **Test Screen** button again.

Script Editor Reference Files

Basic Control Engine Provides a dictionary of basic extension functions, operators and statements

Basic Extensions for CimEdit Scripts Lists obsolete basic extensions and their replacements, in addition to describing current basic extensions.

CimEdit/CimView GefObject Model Provides the CimEdit and CimView GefObject model, listing the properties, methods, objects and enums. Static objects and using the GefObject Model.

CIMPLICITY Configuration Object Model Provides the CIMPLICITY server object model, listing objects and enums.

Basic Extensions for CimEdit Scripts

Basic Extensions for CimEdit Scripts

Following is a descriptive list of Basic Extensions that may appear in scripts that were written for existing CimEdit screens.

Some of the Basic Extensions, which are listed first, can be used in addition to the new powerful OLE automated extensions that should be used in current and future scripts.

Several of these Basic Extensions are obsolete. If they exist in previous scripts they can be left in them. It is not recommended that you use them for new development. The obsolete Basic Extensions are listed here, in case you find one in older scripts and need information about it.

Following is a list of the obsolete Basic Extensions and their replacements:

Obsolete Basic Extension	Replaced By
cimEvent (constant)	**CimGetEventContext** (function)
CimEvent (object)	**GefEventContext** (automation object)
CimGetEvent (function	**CimGetEventContext** (function)
CimGetRootObject (function)	**CimGetScreen** (function)
CimGetScriptOwner (function)	**CimGetObject** (function)
CimObject (object)	**GefObject** (automation object)
CimObjectVariable (object)	**GefObjectVariable** (automation object)
cimOwnerObject (constant)	**CimGetObject** (function)
CimAutoUpdateScreen (function)	**GefScreen.Refresh** (automation method)

Basic Extensions

Basic Extensions

Basic extensions include:

- CimCreateSafeArray (function)
- CimEdit (function)
- CimGetEventContext (function)
- CimGetObject (function)
- CimGetScreen (function)
- cimOleObj (constant)

- CimView (function)

CimCreateSafeArray (function)

Syntax	**CimSafeArray = CimCreateSafeArray()**
Description	Returns an **Automation object** that can create and manipulate an array that be used with Automation methods and properties that require arrays.
Status	The extension can be used.
Comments	Native arrays created in Basic cannot be passed to Automation methods that require array parameters. Similarly, Automation methods that return arrays, return Automation arrays – not Basic arrays. The **CimSafeArray** Automation object allows you to manipulate Automation arrays in Basic.

Example	`Dim vtx As CimSafeArrayLib.COCimSafeArray`
	`Dim obj As GefObjectModel.GefObject`
	`Dim bdr As GefObjectModel.GefObject`
	`Set vtx = CimCreateSafeArray()`
	`vtx.CreateVector cimVLong, 0, 8`
	`Set obj = CimGetObject()`
	`'Create a red line around the object`
	`vtx.SetVectorElement 0, obj.Left`
	`vtx.SetVectorElement 1, obj.Top`
	`vtx.SetVectorElement 2, obj.Left + obj.Width`
	`vtx.SetVectorElement 3, obj.Top`
	`vtx.SetVectorElement 4, obj.Left + obj.Width`
	`vtx.SetVectorElement 5, obj.Top - obj.Height`
	`vtx.SetVectorElement 6, obj.Left`
	`vtx.SetVectorElement 7, obj.Top - obj.Height`
	`Set bdr = obj.Parent.AddPolyline(vtx, True)`
	`bdr.Line.ForeColor.RGB = &hFF`
	`bdr.Fill.None`
	`CimGetScreen().Refresh False`

CimEdit (function)

Syntax	**GefApplication = CimEdit ()**
Description	Returns the GefApplication Automation object representing the CimEdit application.
Status	The extension can be used.

Example	`Dim EditApp As GefObjectModel.GefApplication`
	`Dim EditScreen As GefObjectModel.GefScreen`
	`Set EditApp = CimEdit()`
	Set EditScreen = ViewApp.Screens.Add(TRUE)

CimGetEventContext (function)

Syntax	**GefEventContext = CimGetEventContext ( )**
Description	Returns the GefEventContext Automation object representing the event that caused this script to run.
Status	The extension can be used.

Example	```Dim ev As GefObjectModel.GefEventContext
	Dim x As Long
	Dim y As Long
	Dim flags As Long
	Set ev = CimGetEventContext()
	If ev.GetMouseEvent(x, y, flags) Then
	'Center Object on mouse coordinates
	x = x - ev.TriggerObject.Width / 2
	y = y + ev.TriggerObject.Height / 2
	ev.TriggerObject.Left = x
	ev.TriggerObject.Top = y
	ev.Parent.Refresh False
	End If```

CimGetObject (function)

Syntax	**GefObject = CimGetObject ( )**
Description	Returns the GefObject Automation object representing the object owning this script.
Status	The extension can be used.
Example	```Dim led As GefObjectModel.GefObject Dim var As GefObjectModel.GefObjectVariable Set led = CimGetObject() Set var = led.GetVariable("on") var = Not var.GetValueAsNumber() CimGetScreen.Refresh False```

CimGetScreen (function)

Syntax	**GefScreen = CimGetScreen ( )**
Description	Returns the GefScreen Automation object representing the screen that the script is running in.
Status	The extension can be used.
Example	```CimGetScreen().Refresh True```

cimOleObj (constant)

Syntax	**cimOleObj**
Description	This is a private constant object that represents the OLE Automation interface to the ActiveX control that contains the script. Note a declaration will often be seen at the top of a script contained in an ActiveX control. It will appear as: **Private cimOleObj As Object** or as: **Private cimOleObj As ObjTypeLib.ObjType** Where **ObjTypeLib** is the name of the objects type library and **ObjectType** is the name of the object in the type library.
Status	The extension can be used.
Type	**Object**

Example	`cimOleObj.Value = 10`

CimView (function)

Syntax	**GefApplication = CimView ( )**
Description	Returns the GefApplication Automation object representing the CimView application.
Status	The extension can be used.

Example	`Dim ViewApp As GefObjectModel.GefApplication`
	`Dim currentScreen As GefObjectModel.GefScreen`
	`Set ViewApp = CimView()`
	`Set currentScreen = ViewApp.ActiveScreen`

Obsolete Basic Extensions

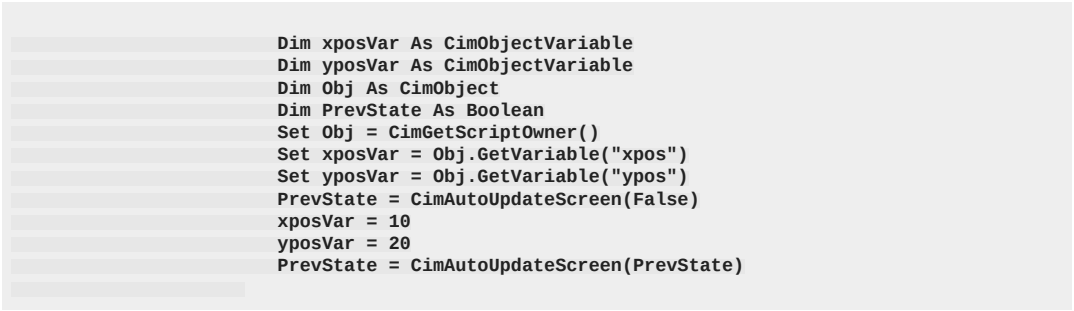
Obsolete Basic Extensions

Obsolete Basic extensions include:

- CimAutoUpdateScreen (function)
- CimEvent (object)
- CimEvent.GetMouseEvent (function)
- CimEvent.GetOleParmName (function)
- CimEvent.GetOleParm (function)
- CimEvent.OleParmsCount (read-only property)
- CimEvent.TriggerObject (read-only property
- CimEvent.Type (read-only property)
- CimEvent.UserParameter (read-only property)
- CimGetEvent (function)
- CimGetRootObject (function)
- CimGetScriptOwner (function)
- CimObject (object)
- CimObject.Container (read-only property)
- CimObject.GetObject (method)
- CimObject.GetVariable (method)
- CimObject.Name (read-only property)
- CimObject.ObjectCount (read-only property)
- CimObject.OleObject (read-only property)
- CimObjectVariable (object)
- CimObjectVariable.GetValueAsNumber (method)
- CimObjectVariable.GetValueAsString (method)
- CimObjectVariable.ID (read-only property)

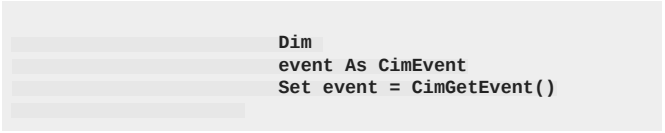
- CimObjectVariable.Value (read/write property)
- CimOwnerObj (constant)

CimAutoUpdateScreen (function)

Obsolete	Replaced by a GefScreen.Refresh
Syntax	**Boolean = CimAutoUpdateScreen (** AutoUpdate **)**
Description	Turns AutoUpdate on when set to TRUE and off when set to FALSE. Returns a **Boolean** representing the previous state of the Auto Update flag.
Comments	The AutoUpdate parameter is a Boolean specifying the new state of the Auto Update flag. AutoUpdate is set to TRUE by default when a script starts running. While Auto Update is **True**, if the value of a **CimObjectVariable** changes, the screen is synchronously updated, and any **CimObjects** that are animated by the variable are changed before the value set returns. If AutoUpdate is set to FALSE, the screen is not updated as **CimObjectVariables** change, but if the screen is updated for another reason, the correct value will be shown. This is useful for setting many variables that you want to appear to change at the same time. When AutoUpdate is set back to TRUE, the screen is immediately updated.
Example	

```
Dim xposVar As CimObjectVariable
Dim yposVar As CimObjectVariable
Dim Obj As CimObject
Dim PrevState As Boolean
Set Obj = CimGetScriptOwner()
Set xposVar = Obj.GetVariable("xpos")
Set yposVar = Obj.GetVariable("ypos")
PrevState = CimAutoUpdateScreen(False)
xposVar = 10
yposVar = 20
PrevState = CimAutoUpdateScreen(PrevState)
```

CimEvent (object)

Obsolete	Replaced by a GefEventContext
Overview	Represents the event in CimView that caused the script to be run.
Example	

```
Dim
event As CimEvent
Set event = CimGetEvent()
```

CimEvent.GetMouseEvent (function)

Obsolete	Replaced by a GefEventContext.
Syntax	**CimEvent** . **GetMouseEvent(** xPos, yPos, mouseFlags **)**
Description	Retrieves information about the mouse. Returns a **Boolean** indicating if the mouse parameters retrieved are valid. This will only return TRUE if the event type is **CIM_ET_MOUSE_DOWN** or **CIM_ET_MOUSE_UP** or **CIM_ET_WHILE_MOUSE_DOWN.**

Comments	The xPos parameter is a **Long** passed **ByRef** that receives the x position of the mouse. The yPos parameter is a **Long** passed **ByRef** that receives the y position of the mouse. The mouseFlags parameter is a **Long** passed **ByRef** that receives the flags indicating which mouse button are down and the states of the shift and control keys. These states are indicated by using the **And** operator and the following constants. For the left mouse button **CIM_MF_LBUTTON.** For the right mouse button **CIM_MF_RBUTTON.** For the shift key **CIM_MF_SHIFT** . For the control key **CIM_MF_CONTROL** . For the middle mouse button **CIM_MF_MBUTTON.** Note the units for the coordinates of the mouse are 1/20 of a point. These coordinates are scaled by the zoom factor of the view, and the origin of the coordinate system is the lower left corner of the CimView document. The effect of this coordinate mapping is to supply coordinates that are constant static to the objects in the document. For example, if there is a square in the document with its top left corner at xPos = 1000 and yPos = 3000 and the user clicks on the top left corner of the square the coordinates of the mouse will be reported as xPos = 1000 and yPos = 3000 regardless of the zoom factor and scrolled position of the view.

Example	```
	Dim
	xVar As CimObjectVariable
	Dim yVar As CimObjectVariable
	Set xVar = cimOwnerObj.GetVariable("xVar")
	Set yVar = cimOwnerObj.GetVariable("yVar")
	Dim pState As Boolean
	Dim x As Long
	Dim y As Long
	Dim f As Long
	If cimEvent.GetMouseEvent(x, y, f) Then
	  pState = CimAutoUpdateScreen(FALSE)
	  xVar = x
	  yVar = y
	pState = CimAutoUpdateScreen(pState)
	  If f And CIM_MF_CONTROL Then
	  'The control key is down
	  'So do some other stuff…
	  End If
	End If
	``

CimEvent.GetOleParmName (function)

Obsolete	Replaced by a GefEventContext
Syntax	**CimEvent.GetOleParmName(** parmIndex% **)**
Description	Returns a **String** representing the name of a parameter from an ActiveX control event. This is only valid if the event type is **CIM_ET_OLE_EVENT** .
Comments	The parmIndex% parameter is an Integer specifying the index of the parameter as it was passed from the ActiveX control event. For example, if the control fires the event **MouseDown(** xPos As Integer, yPos As Integer **)** then xPos can be retrieved using a parmIndex% with the value 0, and yPos with the value 1.

Example	

```
Dim
Ev As CimEvent
Dim Var As Variant
Set Ev = CimGetEvent()
If Ev.Type = CIM_ET_OLE_EVENT Then
  Count% = Ev.OleParmsCount
  i% = 0
  xPosIndex% = -1
  While i% < Count% And xPosIndex = -1
    If Ev.GetOleParmName(i%) = "xPos" Then
      XposIndex% = Index%
    End If.
    i% = i% + 1
  Wend
End If
```

CimEvent.GetOleParm (function)

Obsolete	Replaced by a GefEventContext
Syntax	**CimEvent.GetOleParm(** parmIndex% **)** **CimEvent.GetOleParm(** parmName$ **)**
Description	Returns a **Variant** representing the value of a parameter from an ActiveX control event. This is only valid if the event type is **CIM_ET_OLE_EVENT.**
Comments	The parmIndex% parameter is an **Integer** specifying the index of the parameter as it was passed from the ActiveX control event. The parmName$ parameter is a **String** specifying the name of the parameter as it was passed from the ActiveX control event. For example if the control fires the event **MouseDown(** xPos As Integer, yPos As Integer **)** then xPos will have the name **xPos**, and yPos will have the name **yPos**.
Example	

```
Dim Ev As CimEvent
Dim Var As Variant
Set Ev = CimGetEvent()
If Ev.Type = CIM_ET_OLE_EVENT Then
  Count% = Ev.OleParmsCount
  Index% = 0
  While Index% < Count%
    Var = Ev.GetOleParm(Index%)
    'Do Stuff with the parameter.
    Index% = Index% + 1
  Wend
  x% = Ev.GetOleParm("xPos")
  y% = Ev.GetOleParm("yPos")
End If
```

CimEvent.OleParmsCount (read-only property)

Obsolete	Replaced by a GefEventContext
Syntax	**CimEvent.OleParmsCount**
Description	Returns an **Integer** representing the number of parameters from an event fired from an ActiveX control. This is only valid if the event type is **CIM_ET_OLE_EVENT** .

Example	```
	Dim
	Ev As CimEvent
	Dim var As Variant
	Set Ev = CimGetEvent()
	If Ev.Type = CIM_ET_OLE_EVENT Then
	  Count% = Ev.OleParmsCount
	  Index% = 0
	  While Index% < Count%
	  var = Ev.GetOleParm(Index)
	  'Do Stuff with the parameter.
	  Wend
	End If
	``

CimEvent.TriggerObject (read-only property

Obsolete	Replaced by a GefEventContext
Syntax	**CimEvent.TriggerObject**
Description	Returns the **CimObject** object for which the event was configured.
Example	``` Dim Obj As CimObject Set Obj = CimGetEvent().TriggerObject ```

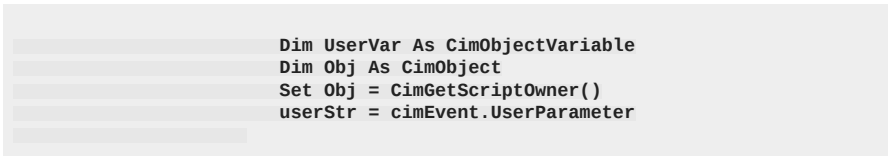
CimEvent.Type (read-only property)

Obsolete	Replaced by a GefEventContext
Syntax	**CimEvent** . **Type**
Description	Returns an **Integer** representing the type of event.
Comments	The type is one of the following values: **CIM_ET_NONE CIM_ET_MOUSE_DOWN CIM_ET_MOUSE_UP CIM_ET_KEY_DOWN CIM_ET_KEY_UP CIM_ET_WHILE_MOUSE_DOWN CIM_ET_WHILE_KEY_DOWN CIM_ET_TIMED CIM_ET_EXPRESSION_HIGH CIM_ET_EXPRESSION_UPDATE CIM_ET_SCREEN_OPEN CIM_ET_SCREEN_CLOSE CIM_ET_OLE_EVENT**

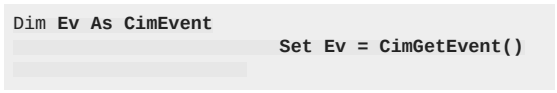
Example	```
	Dim
	Ev As CimEvent
	Set Ev = CimGetEvent()
	If Ev.Type = CIM_ET_SCREEN_OPEN Then
	``

CimEvent.UserParameter (read-only property)

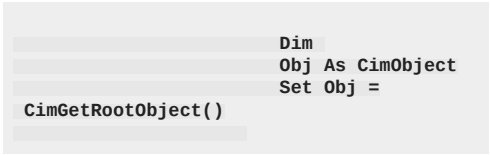
Obsolete	Replaced by a GefEventContext
Syntax	**CimEvent.UserParameter**)
Description	Returns a **String** representing the parameter specified in the event during configuration.

Example	 **Set UserVar = Obj.GetVariable(userStr)**

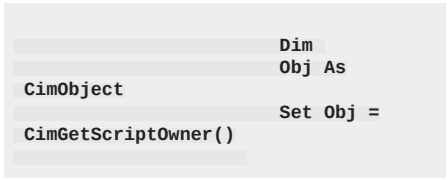
CimGetEvent (function)

Obsolete	Replaced by a GefEventContext
Syntax	**CimGetEvent()**
Description	Returns the **CimEvent** that caused the script to be run.
Example	

CimGetRootObject (function)

Obsolete	Replaced by CimGetScreen *(page 547)*
Syntax	**CimGetRootObject()**
Description	Returns the **CimObject** representing the screen.
Example	

CimGetScriptOwner (function)

Obsolete	Replaced by CimGetObject *(page 547)*
Syntax	**CimGetScriptOwner()**
Description	Returns the **CimObject** that owns the script.
Example	

CimObject (object)

Obsolete	Replaced by a GefObject
Description	Represents a visual object on the screen.
Example	`Dim Obj As CimObject` `Set obj = CimGetEvent().TriggerObject`

CimObject.Container (read-only property)

Obsolete	Replaced by a GefObject
Syntax	**CimObject.Container**
Description	Returns a **CimObject** representing the container of this object.

Example	`Sub ShowContainers`
	`  Dim obj As CimObject`
	`  Set obj = CimGetEvent().TriggerObject`
	`  mymsg$ = ""`
	`  While (Not obj Is Nothing)`
	`  mymsg$ = obj.Name + " " + mymsg$`
	`  Set obj = obj.Container`
	`  Wend`
	`  MsgBox mymsg$`
	`End Sub`

CimObject.GetObject (method)

Obsolete	Replaced by a GefObject
Syntax	**CimObject.GetObject(** index **)** **CimObject.GetObject(** name$ **)**
Description	Returns a **CimObject** object representing the child object with the specified index or name.
Comments	The index parameter is an Integer specifying the index of the child object. The name$ parameter is a String specifying the name of the child object.

Example	This example loops through all the children of an object.
	`Dim Obj As CimObject`
	`Dim Child As CimObject`
	`Set Obj = CimGetScriptOwner()`
	`Max% = Obj.ObjectCount - 1`
	`For Index% = 0 To Max%`
	`  Set Child = Obj.GetObject(Index%)`
	`Next Index`
	`This example gets the child object named "Tank10".`
	`Dim Obj As CimObject`
	`Dim Tank As CimObject`
	`Set Obj = cimEvent.TriggerObject`
	Set Tank = Obj.GetObject("Tank10")

CimObject.GetVariable (method)

Obsolete	Replaced by a GefObject
Syntax	**CimObject.GetVariable(** name$ **)**)
Description	Returns a **CimObjectVariable** representing the named variable in the object or one of the object's containers.
Comments	The name$ parameter is a String containing the name of the variable. You can then set the variable to display a value or the current value of a Point ID.

Example	`Dim xposVar As CimObjectVariable`
	`Dim Obj as CimObject`
	`Set Obj = CimGetScriptOwner`
	`Set xposVar = Obj.GetVariable ("xvar")`
	`xposVar = "TANK1"`

CimObject.Name (read-only property)

Obsolete	Replaced by a GefObject
Syntax	**CimObject.Name**)
Description	Returns a **String** representing the name of the object.

Example	`Dim Obj As CimObject`
	`Set Obj = CimGetScriptOwner()`
	`If Obj.Name = "Tank1" Then`
	`...`

CimObject.ObjectCount (read-only property)

Obsolete	Replaced by a GefObject

Syntax	**CimObject.ObjectCount**
Description	Returns an **Integer** representing the number of **CimObjects** this object contains.

Example	<pre>Dim Obj As CimObject
	Dim Child As CimObject
	Set Obj = CimGetScriptOwner()
	Dim Max As Integer
	Max = Obj.ObjectCount - 1
	Dim Index As Integer
	For Index = 0 To Max
	  Set Child = Obj.GetObject(Index)
	Next Index</pre>

CimObject.OleObject (read-only property)

Obsolete	Replaced by a GefObject
Syntax	**CimObject.OleObject**
Description	Returns an **Object** representing the OLE Automation interface to an ActiveX control.

Example	<pre>Dim Gauge As CimObject
	Dim RootObj As CimObject
	Set RootObj = CimGetRootObject()
	Set Gauge = RootObj.GetObject("FuelGauge")
	Gauge.OleObject.NeedleValue = 10</pre>

CimObjectVariable (object)

Obsolete	Replaced by a GefObjectVariable
Description	Represents an object variable on a CimView screen.

Example	<pre>Dim rotorVar As CimObjectVariable
	Dim Obj As CimObject
	Set Obj = CimGetScriptOwner()
	Set rotorVar = Obj.GetVariable("RotorNum")</pre>

CimObjectVariable.GetValueAsNumber (method)

Obsolete	Replaced by a GefObjectVariable
Syntax	**CimObjectVariable.GetValueAsNumber()**
Description	Returns a **Double** representing the value of the variable.

Comments	This method is more efficient than using the **Value** property, if you know the value is a number.

CimObjectVariable.GetValueAsString (method)

Obsolete	Replaced by a GefObjectVariable
Syntax	**CimObjectVariable.GetValueAsString()**
Description	Returns a **String** representing the value of the variable.

CimObjectVariable.ID (read-only property)

Obsolete	Replaced by a GefObjectVariable
Syntax	**CimObjectVariable.ID**
Description	Returns a **String** representing the name of the variable.
Example	`If xposVar.ID = "TankSelector" Then`

CimObjectVariable.Value (read/write property)

Obsolete	Replaced by a GefObjectVariable
Syntax	**CimObjectVariable.Value**
Description	Returns a **Variant** representing the value of the variable. However, the value is most commonly returned as a string.
Comments	This is the default property

Example	`Dim oldValue As String`
	`oldValue = xposVar.Value`
	`xposVar.Value = "TANK_LEVEL1"`

	Since this is the default property, the above can also be written as:

	`Dim oldValue As String`
	`oldValue = xposVar`
	`xposVar = "TANK_LEVEL1"`

CimOwnerObj (constant)

Obsolete	Replaced by CimGetObject *(page 547)*

Syntax	**cimOwnerObj**
Description	This a constant object that represents the same CimObject as the object returned by **CimGetScriptOwner()**
Type	`CimObject`
Example	`Dim var1 As CimObjectVariable` `Set var1 = cimOwnerObj.GetVariable("var1")`

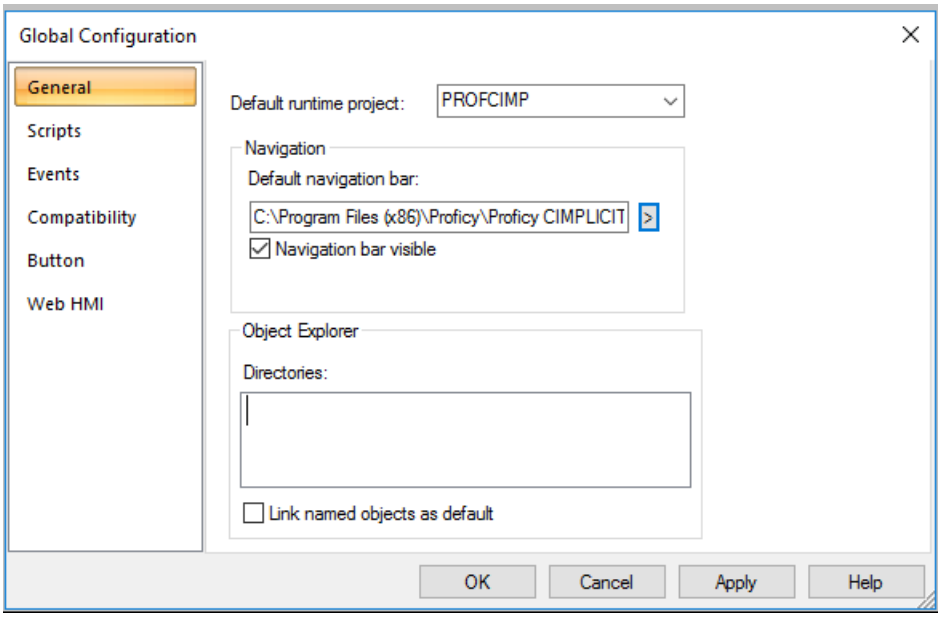
CimEdit Global Specifications

CimEdit Global Specifications

Several of CimEdit's features can now, in one location, be identified as global feature. The location is the Global Configuration dialog box.

1. Click the CimEdit button .
2. Click Global Configuration.

The Global Configuration dialog box opens.



rect -3, 110, 19, 131 5. Display Options *(page 574)*
rect -2, 82, 21, 103 4. Compatibility *(page 571)*
rect -2, 61, 21, 82 3. Before Context Menu/Before Editor Context Menu Global Events *(page 565)*
rect -2, 39, 21, 60 2. Global Scripts Load List *(page 561)*
rect -3, 18, 20, 39 1. General Features that can be Shared Globally *(page 560)*

Features that can be shared globally include the following.

Step Number	Description
1 *(page 560)*	General features that can be shared globally.
2 *(page 561)*	Global script load list.
3 *(page 565)*	Before Context Menu/Before Editor Context Menu global events.
4 *(page 571)*	Compatibility.
5 *(page 574)*	Global button selections.
6 *(page 575)*	Undo stack size for a CimEdit screen (CimEdit Options dialog box).
7 *(page 576)*	Global Web HMI Configuration details

Important: If CIMPLICITY has just been upgraded from CIMPLICITY v8.2 or lower to CIMPLICITY v9.0 copy a configured CimView.cfg file (saved from the previous version) so it is in the following location: ..\<CIMPLICITY installation>\Data\CimView.cfg

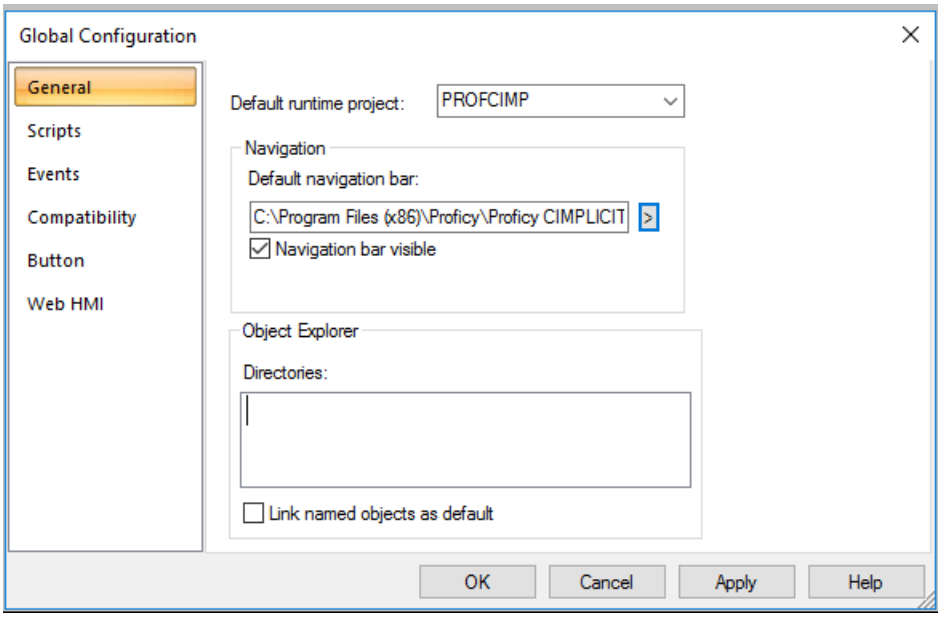
If this file has not been copied into the ..\Data directory, all of the global configuration settings will be the default settings, not settings that you might have configured in the earlier version.

If you do not have a configured CimView.cfg file, use the dialog boxes to enter and apply the settings that are appropriate for your CIMPLICITY v9.0 system.

1. General Features that can be Shared Globally

Select the General tab in the Global Configuration dialog box.

Features that can be selected to always be shared globally in CimEdit and/or CimView are as follows.



rect 81, 30, 104, 48 *(page 561)*
rect 80, 53, 103, 74 *(page 561)*
rect 80, 131, 103, 154 *(page 561)*
rect 81, 212, 104, 237 *(page 561)*

1 *(page 561)*	Default runtime project.
2 *(page 561)*	Navigation.
3 *(page 561)*	Object Explorer directories.
4 *(page 561)*	Link named objects as default.

1	Default runtime project

The selected project is the project that CIMPLICITY will use to request data for all unqualified points.

The **Default runtime project** field Is a convenient way to implement the `/project` command line option. However, if a project is also selected using the /project command, that project will override the project selected in the **Default runtime project** field.

If no project is selected using either method, the system uses present behavior.

2	Navigation

CimView can include a customized Navigation menu bar that guides users through the CimView screens that they to which they need access.

A screen designer can use the Navigation section to open the Navigation Configuration Editor to configure items on and computer access to the Navigation menu.

3	Object Explorer directories

Object Explorer folders contain *.cim screens that make up a library of objects, from basic shapes to complicated configurations, that can be used to rapidly design and configure CimEdit screens.

The Object explorer directories box provides a way to identify where the folders are located, if one or more locations are different from the default location.

4	Link named objects as default

CimEdit supports the ability to create one or more linked objects from a single source object.

Checking or clearing Link named objects as default determines how a screen designer will create links to a source object *(page 342)* .

2. Global Scripts Load List

2. Global Scripts Load List

You can easily create a prioritized list of global scripts that will be accessible to either:

- A selected screen
- All screens in a CimEdit/CimView instance.

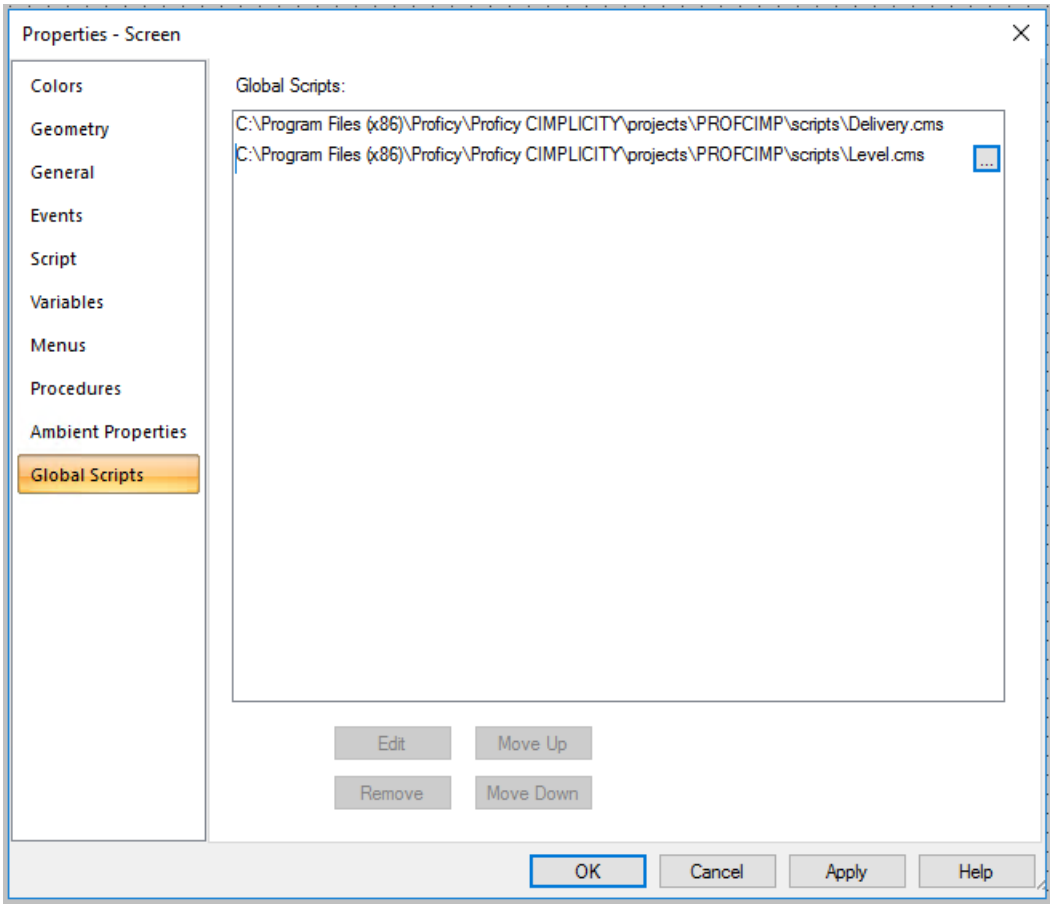
Either selection provides options for what scripts will be listed and in what priority.

- Script listing options.

A Selected Screen

Select **Global Scripts** in the Properties - Screen dialog box.

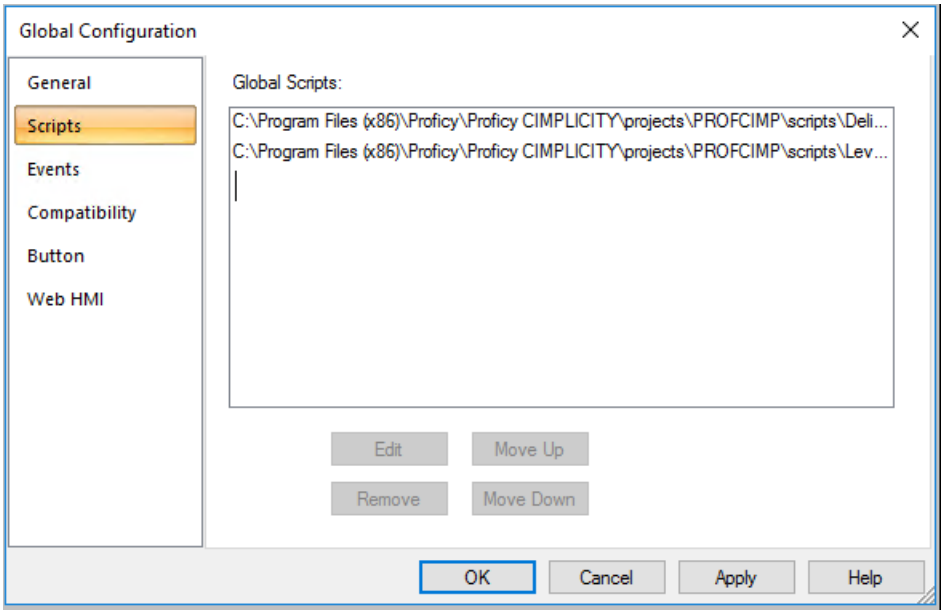
Global scripts loaded in the Properties - Screen dialog box are associated with the selected screen.



All Screens in a CimEdit/CimView Instance

Select **Scripts** in the Global Configuration dialog box.

Global scripts loaded in the Global Configuration dialog box are accessible to all objects on all screens that are loaded into a single instance of CimEdit/CimView.



Script List Options

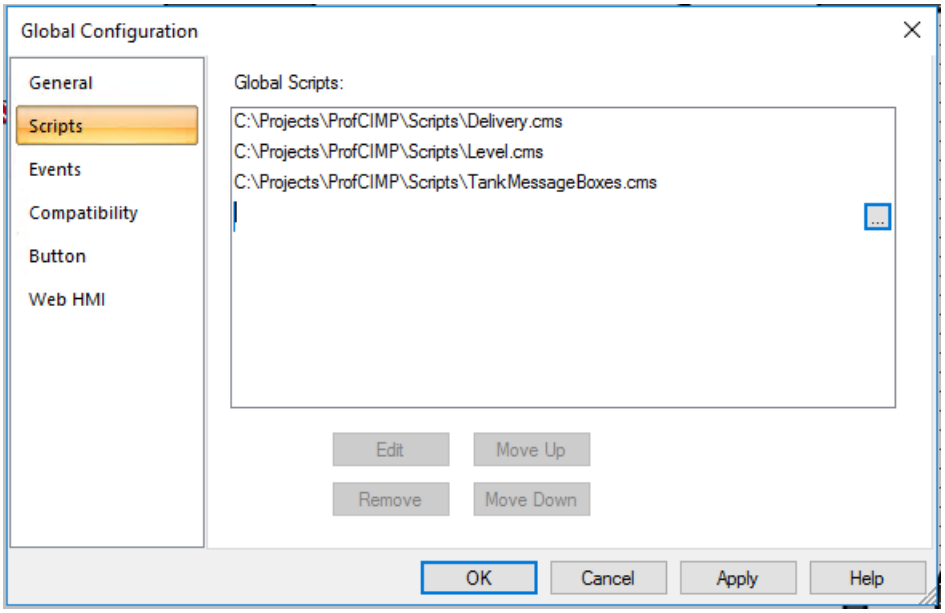
Script listing options are as follows.

2.1 *(page 563)*	Add scripts to the Global Scripts list.
2.2 *(page 564)*	Revise the Global Scripts list.
2.3 *(page 543)*	Create CimEdit runtime-only global scripts.

- The scripts can be anywhere on the network as long as the computer they are on is accessible to the computer that is running the instance of CimEdit/CimView.
- If you loaded scripts using the CimEdit menu *(page 541)* , those scripts will be listed when you select Scripts in the Global Configuration dialog box.

2.1. Add Scripts to the Global Scripts List

Do the following to lo load global scripts in the Global Configuration dialog box.

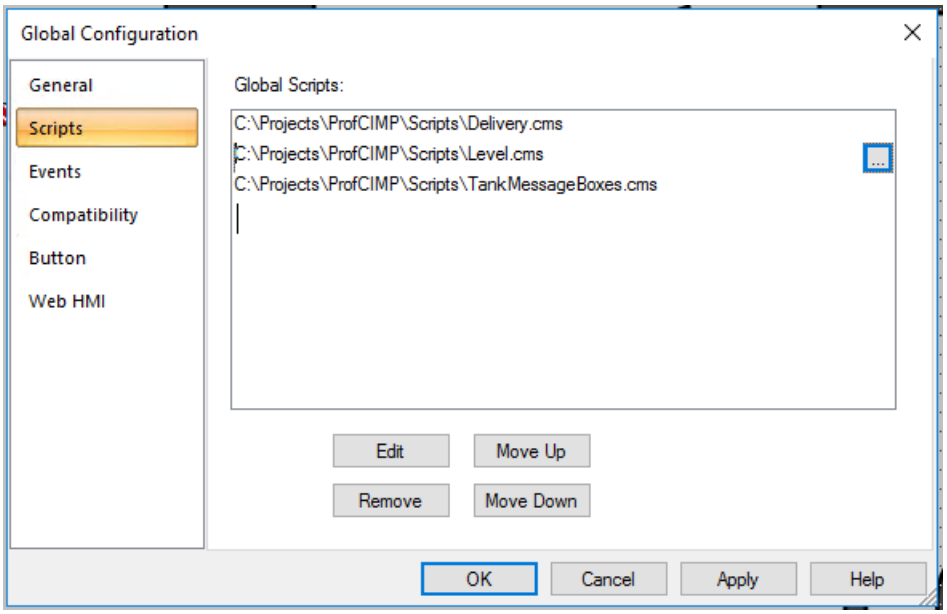


Step Number	Description
1	Click an empty line in the **Global scripts** box. A Browse button displays.
2	Click the Browse button>select a script in the Global script to load dialog box. Options are: • .bcl • .cms • .cmsrt • All files
3	Click Apply. **Important:** You must click Apply before going to the next line.

2.2. Revise the Global Scripts List

Scripts and script entry points are in the order they appear in the list; the script at the top of the list is accessed first.

Scripts can be removed and moved up or down in the list, as follows.



Step Number	Description
1	Select a script in the list. **Note:** The Browse button displays on the selected row.
2	Click any of the enabled buttons. Which buttons are enabled depends on the line's location in the list. Button, Edit, Remove, Move Up, Move Down.
3	Click Apply.

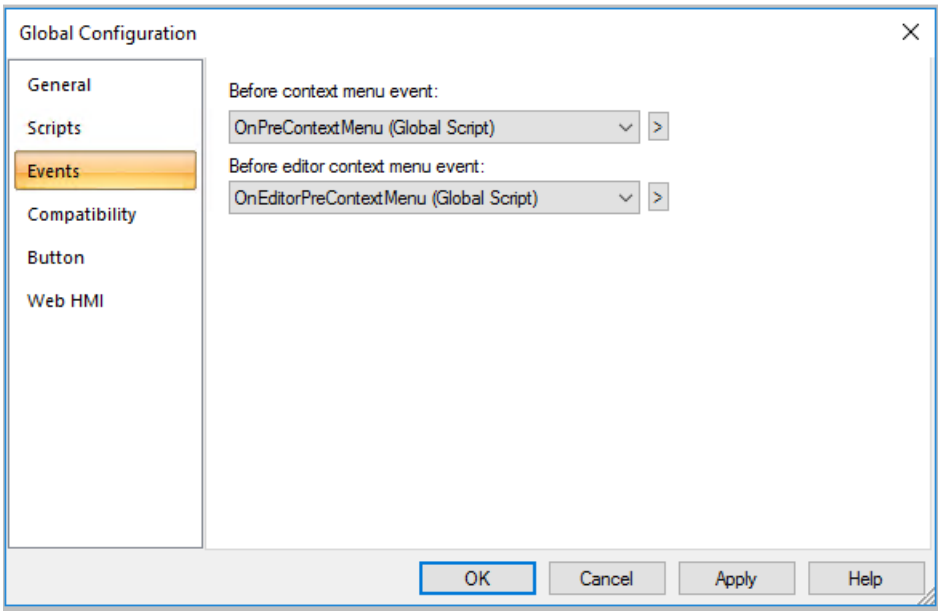
Result: You configuration is applied.

3. Before Context Menu/Before Editor Context Menu Global Events

3. Before Context Menu/Before Editor Context Menu Global Events

You can configure Before Context Menu and/or Before Editor Context Menu events that will be used for every screen that does not have those events already configured.

Select **Events** in the Global Configuration dialog box.



Configuring global context menus includes the following.

3.1 *(page 566)*	Create Scripts for Before Context Menu/Before Editor Context Menu events.
3.2 *(page 571)*	Edit Before Context Menu/Before Editor Context Menu scripts.
4 *(page 571)*	Compatibility.

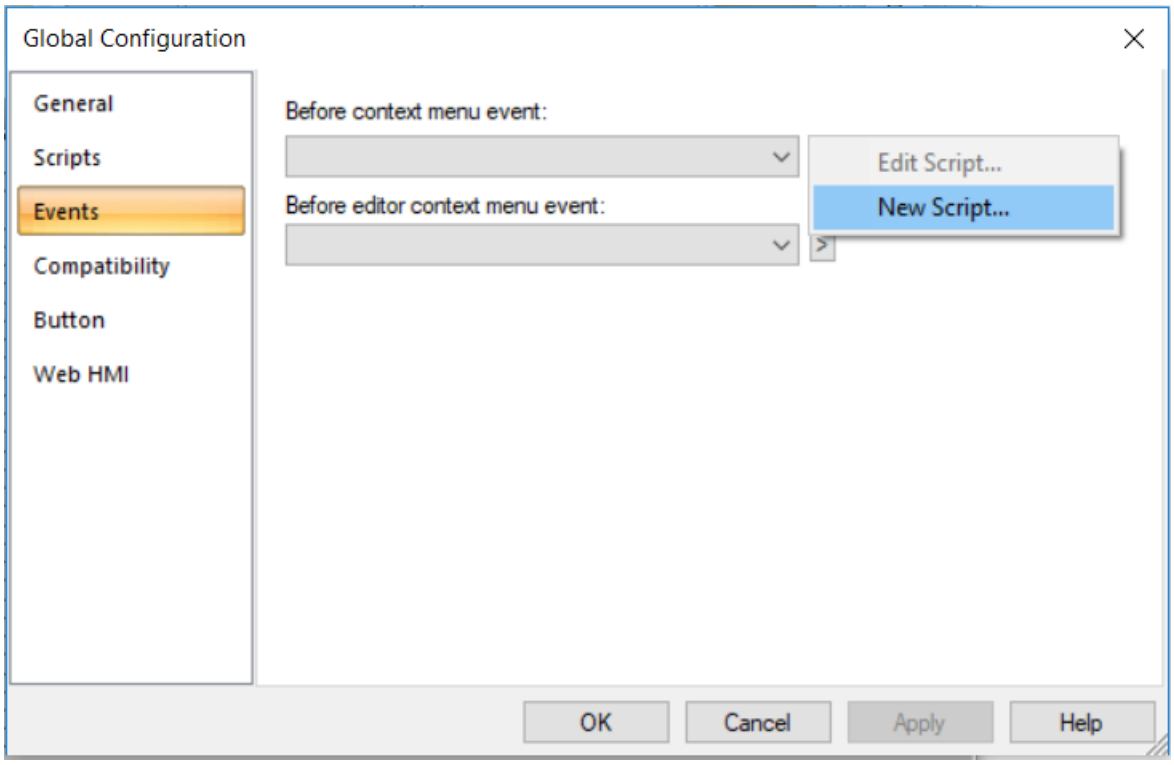
Tip: If you have a screen that should not use a Before Context Menu and/or Before Editor Context Menu event, create the event for that screen and assign an empty script.

3.1. Create Scripts for Before Context Menu/Before Editor Context Menu Events

This topic describes how to create scripts Before Context Menu event and Before Editor Context Menu Event.

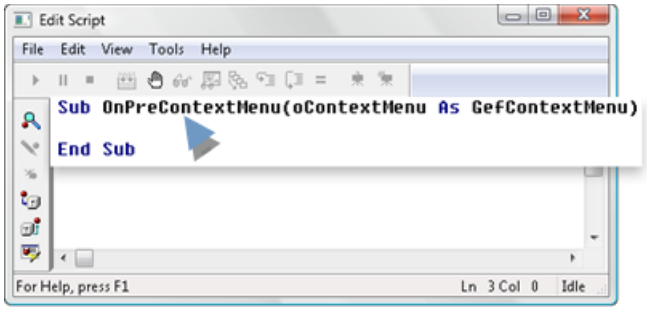
Before Context Menu Event

1. Click the Popup button to the right of the **Before context menu event** field.
2. Select New Script on the Popup menu.



An Edit Script window opens with the following default code.

```
Sub OnPreContextMenu(oContextMenu As GefContextMenu)
                    End Sub
```



3. Write and compile the script that is needed for the Before Context Menu event.
4. Close the Edit Script window.

   The Save global script browser opens.

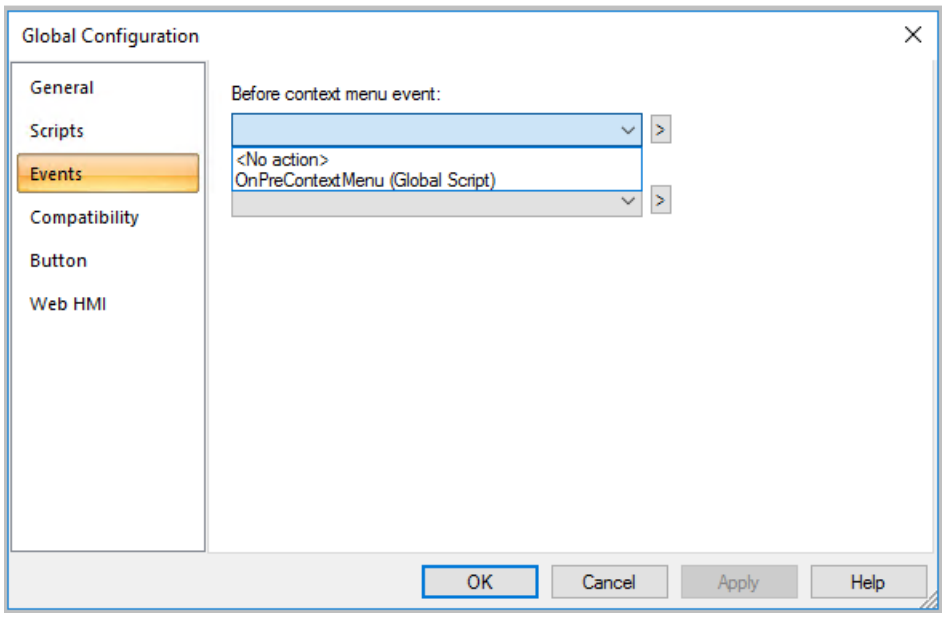
5. Do the following.

Item	Field	Action
A	File name	Enter a name for the script.
B	File of type	Select a script type. Options are: *.cms or *.bcl

6. Click Open.
   The script can be applied to the Before context menu event.

7. Click the Arrow button to the right of the Before context menu event.

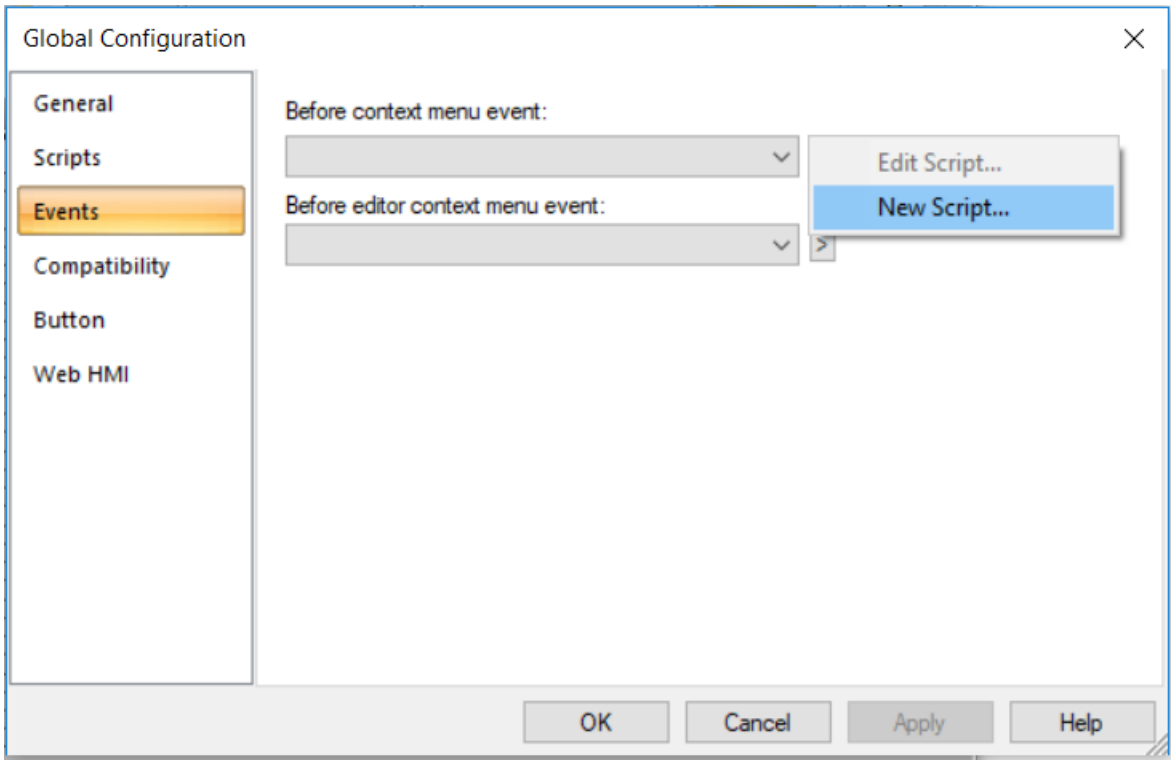
8. Select one of the following.



Option	Result
<No action>	The Before context menu event will not be used.
Script Entry point	Is triggered when a user opens a Popup (Context) menu on any CimView screen.

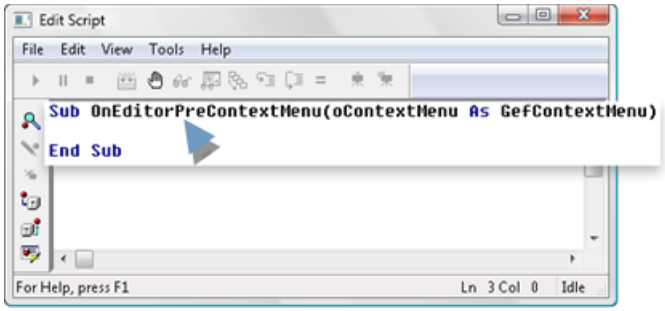
Before Editor Context Menu Event

1. Click the Popup button to the right of the **Before editor context menu event** field.
2. Select New Script on the Popup menu.



An Edit Script window opens with the following default code.

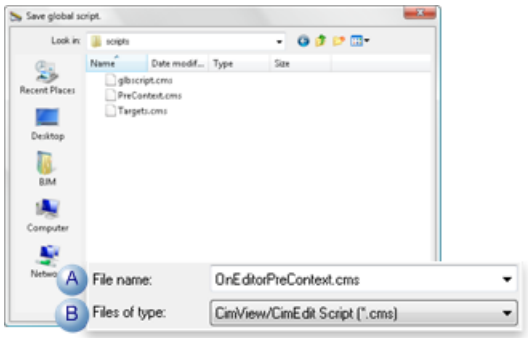
```
Sub OnEditorPreContextMenu(oContextMenu As GefContextMenu)
End Sub
```



3. Write and compile the script that is needed for the Before Editor Context Menu event.
4. Close the Edit Script window.

The Save global script browser opens.

5. Do the following.



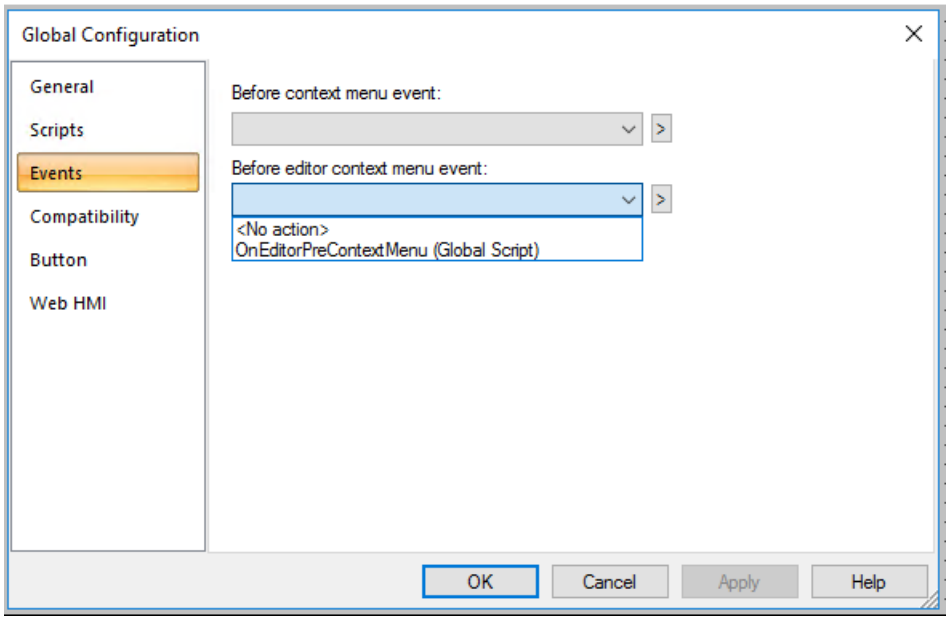
	Field	Action
A	File name	Enter a name for the script.
B	File of type	Select a script type. Options are: *.cms or *.bcl

6. Click Open.

Result: The script can be applied to the Before context menu event.

7. Click the Arrow button to the right of the Before context menu event.

8. Select one of the following.

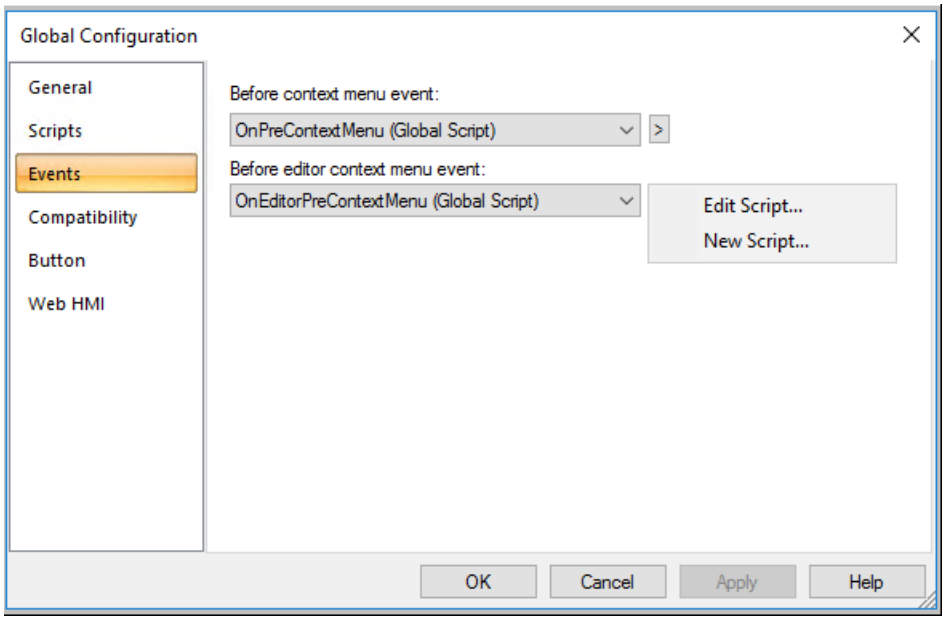


	Option	Result
A	<No action>	The Before context menu event will not be used.
B	Script Entry point	Is triggered when a user opens a Popup (Context) menu on any CimView screen.

3.2. Edit Before Context Menu/Before Editor Context Menu Scripts

The following procedure applies to either the **Before context menu event** or **Before editor context menu event** field.

1. Select the Script that you want to edit.
2. Click the Popup button to the right of the field with the selected script.
3. Select Edit Script on the Popup menu.



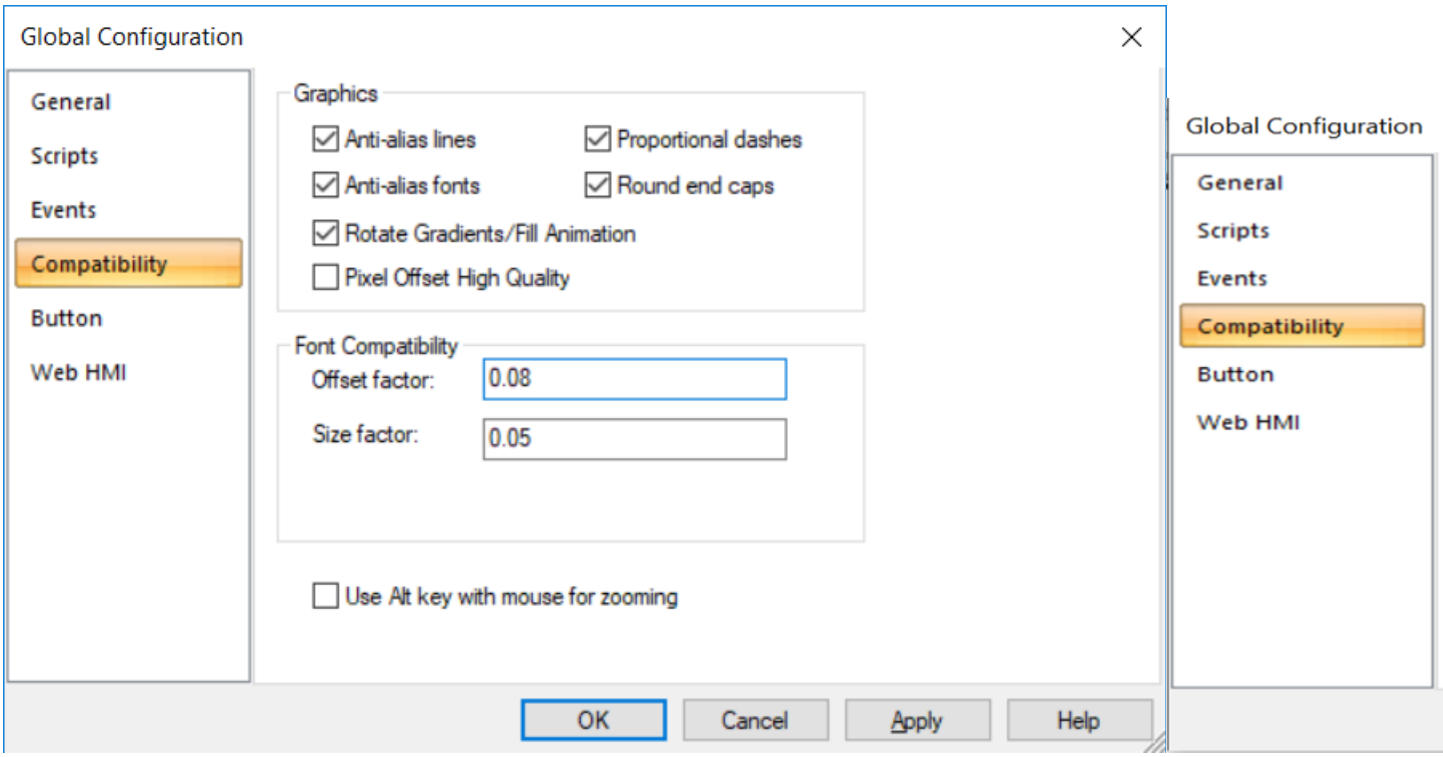
   The Edit Script window opens displaying the selected script.

4. Edit and recompile the script.
5. Close the Edit Script window.

The revised script will be used the next time it is called.

4. Compatibility

Use the following graphics settings to improve the display of graphics created in earlier versions. The default global values are intended to provide the highest quality display, but you can change them as needed for better clarity.



Under Graphics, select or clear check boxes as follows:

- The **Anti-alias lines** check box is selected by default. When the check box is are selected, the objects are blurred and appear to be smoother (without jagged edges) than lines that are not anti-alias. This is particularly visible with curved and diagonal lines.

- The **Rotate Gradients/Fill Animation** check box is selected by default. Beginning with CIMPLICITY v8.1, animated fills can rotate with the object but you can disable the feature to maintain backward compatibility when required.

- The **Pixel Offset High Quality** check box is cleared by default, since using it may slightly impact performance. However, selecting the check box improves the quality of your anti-aliased lines by making them appear sharper (since it helps the anti-aliased pixels better align with the center of the physical pixels on a monitor), and is the recommended setting.

- The **Proportional dashes** check box is selected by default. Dash size and spacing is adjusted for line thickness, as shown in the graphics below. The first is with the check box cleared and the second is with the check box selected.

- The **Round end caps** check box is selected by default. In earlier versions of CimEdit, ends were mitered (squared) but have been enhanced to appear rounded, as shown in the graphics below. The top graphic is with the check box cleared and the bottom graphic is with the check box selected.

Under Text rendering, select or clear radio buttons as follows:

- The **Non-anti-alias** radio button is cleared by default. When the radio button is selected, anti-aliasing is not applied to the alphanumeric objects.

- The **Anti-alias (speed optimized)** radio button is selected by default. When the radio button is selected, anti-aliasing is applied to the alphanumeric objects but it is speed optimized for faster performance.

- The **Anti-alias (appearance optimized) radio** button is cleared by default. When the radio button is selected, anti-aliasing is applied to the alphanumeric objects, and it is appearance optimized. Select this option for better font clarity than the Anti-alias (speed optimized) option. This option provides the highest possible font quality.

Under **Font Compatibility** you can enter offset and size factors, which are a percentage of the font height. Because CimEdit incorporated a new drawing engine in version 8.0, screens that were created in earlier CIMPLICITY versions show differences in the font spacing and alignment. For most fonts, this issue is corrected in CimEdit with the default factors but you can enter new values in the fields if necessary.

IMPORTANT: Do not change the defaults unless you are sure it is necessary. You should back up your screens before changing the default entries.

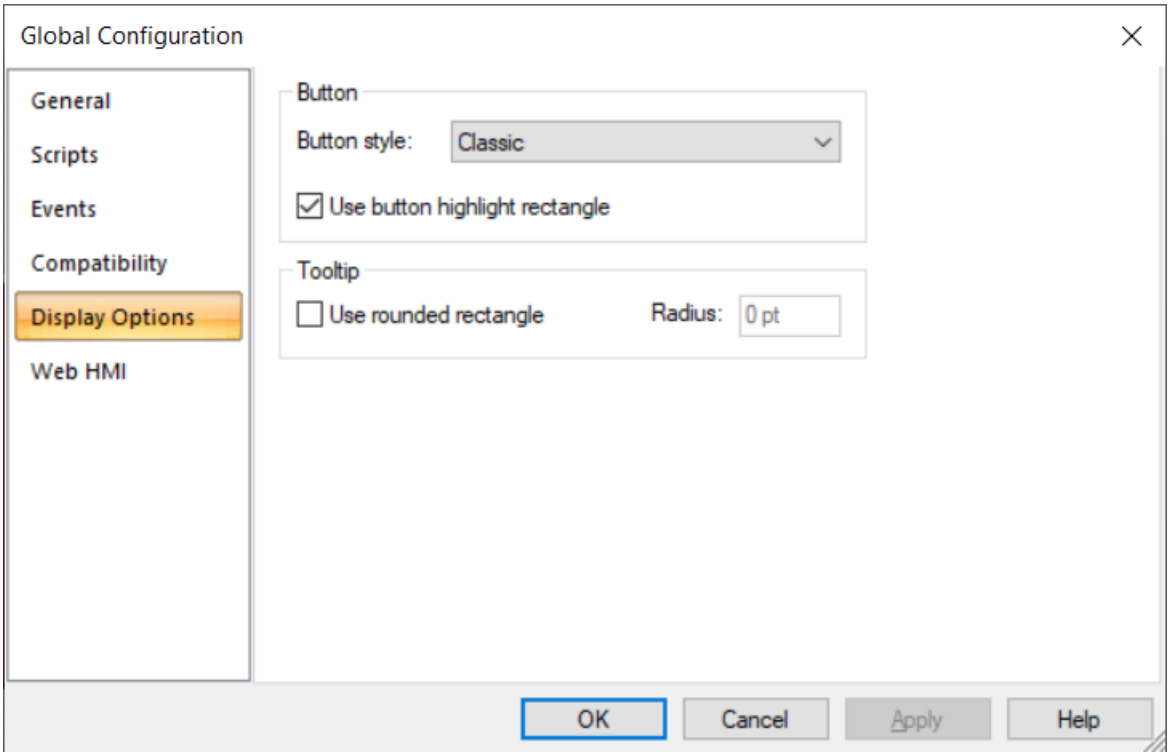
The fields calculate the percent to make the text do the following:

- For **Offset factor**, the text renders off from the anchor point (negative x direction).
- For **Size factor**, the text is reduced by that factor,

You can also select the check box to zoom when you simultaneously click the mouse and press the Alt key.

5. Display Options

Display options can help speed up button configuration, affect the runtime button display, and customize the tooltip display.



rect 74, 58, 95, 78 *(page)*
rect 74, 30, 95, 50 *(page)*

Button Style

The selected button style in the Global Configuration dialog box specifies which button style will display when a new button is placed on the CimEdit screen.

Button style options include:

Option	New button style will be the:
Auto	Last button style selected during configuration

Option	New button style will be the:
One of the Available Button Styles	Style selected in the Button Style field. • Classic • Standard • XP • Vista

Use Button Highlight Rectangle

Select whether or not buttons will display a highlighted rectangle when users click them during runtime.

Action	During Runtime
Check	Buttons will be surrounded by a highlighted box when they are clicked.
Clear	Buttons will not be surrounded by a highlighted box when they are clicked.

Use Rounded Rectangle

Select the Use rounded rectangle check box to display the tooltip with rounded edges. In the Radius box, enter the radius of the rounded edges of the tooltip.

Note: The Radius box is enabled only when the Use rounded rectangle check box is selected.

6. Undo Stack Size for a CimEdit Screen

You can specify how much information CimEdit should keep in an undo stack before the oldest information is discarded. This stack size determines how much you can undo in your configuration.

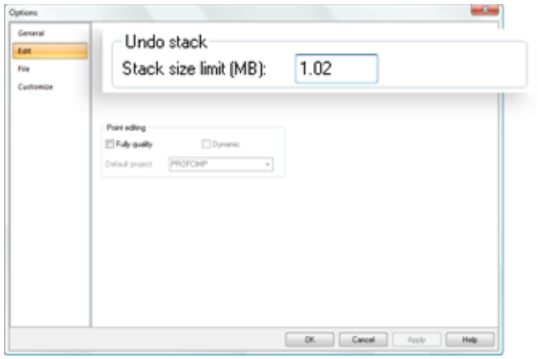
1. Click the CimEdit button .

   The CimEdit menu opens.

2. Click the Options button.

The Options dialog box opens.

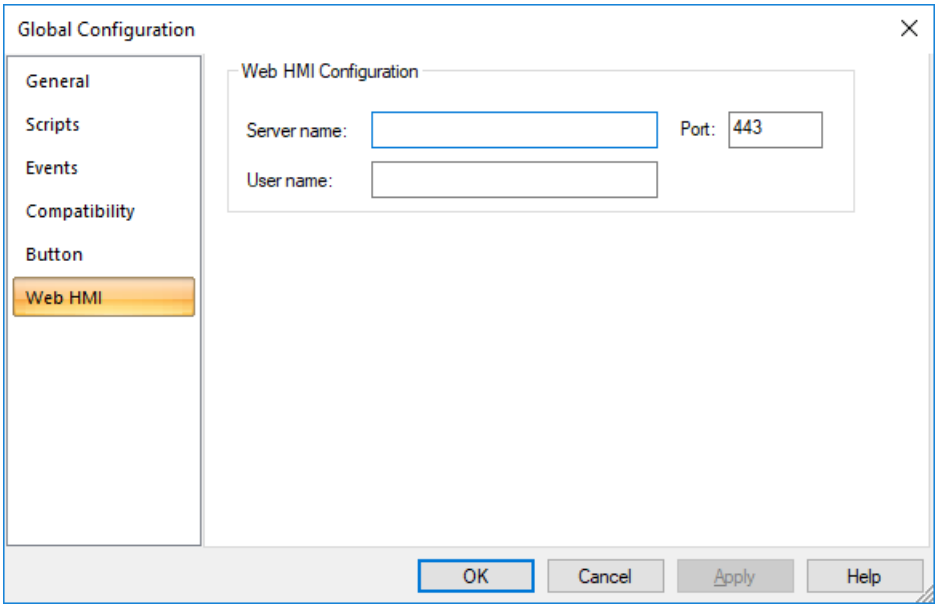
3. Select Edit.
4. Fill in the **Stack size limit (MB)**.



Field	Description
Stack size limit (MB)	Valid number (in megabytes) is one of the following. • A number between 0.001 and half the currently available page file memory. • Zero (0) to disable the undo feature.

7. Web HMI

You can configure Web HMI globally to avoid configuring it each time you publish a Mimic.



rect 74, 58, 95, 78 *(page)*
rect 74, 30, 95, 50 *(page)*

Enter the following details:

- **Server name:** Enter the server name on which Web HMI is running.
- **Port:** Enter the port number on which Web HMI is running.
- **User name:** Enter the Web HMI user name used to publish the Mimic.

ActiveX Controls

ActiveX Controls

The ActiveX control, originally developed by Microsoft to support the creation of Internet-enabled applications, provides interactive controls within an application, on or off the Internet. ActiveX controls are at the forefront of software technology. Many vendors develop these controls.

As a CIMPLICITY user, you now have access to this vast library of controls to use in CimView screens and to supplement the CIMPLICITY Symbol and SmartObject Library. When you incorporate ActiveX controls into a CimEdit screen, you greatly enhance the functionality and ease-of-use of your CimView screens.

ActiveX Controls allow different types of software objects to communicate, using standard interfaces. Developers create ActiveX controls, and you incorporate them into any software application that supports ActiveX controls. The software application acts as a container to "hold" an ActiveX control. It is up to you to configure the ActiveX control into the software application so that it provides the end-user with an easy, meaningful way to view or manipulate data.

ActiveX controls can be gauges, charts, displays, graphs, or any other object that allows a user to access the particular functionality of the object. There are literally hundreds, if not thousands, of ActiveX developers who have created thousands of different ActiveX controls. Some ActiveX controls are distributed at no cost, while others must be purchased. Once you acquire them, they are at your disposal when creating your CimEdit screens.

ActiveX Control Installation

Before you can use ActiveX controls in your CimEdit screen, they must be installed on your computer. Typically, when you download or purchase an ActiveX control from a vendor, you receive a setup or install file that must be run to install the ActiveX control. This executable program will take you through the steps necessary to successfully install the ActiveX control files on your hard drive and register the new control with your operating system.

As each vendor has its way of installing ActiveX controls, installation procedures will vary. Follow your vendor's installation instructions.

Given that there are thousands of ActiveX controls, there are several ways to use these controls to display and transmit CIMPLICITY project information and point data.

CIMPLICITY provides you with ActiveX controls that enable you to track and analyze detailed data pertaining to a project's points and alarms. CIMPLICITY ActiveX controls that may be included with your CIMPLICITY configuration include CIMPLICITY:

- AMV Control.
- HAD-Viewer Editor Control.
- Portal
- Recipe Control.
- SPC Control.
- Trend Control.
- XY Plot Control.

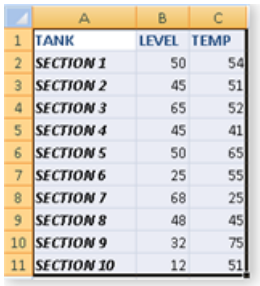
Special Pasting Objects onto a CimEdit Screen

CimEdit offers a Special Paste feature that enables you to Special Paste or Link an object onto the CimEdit screen. This method of placing an object provides you with the functionality of the application from which it was copied.

- Special paste an object on a CimEdit screen.
- Display the image in CimView.
- Work with the Image in CimEdit.

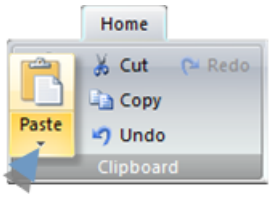
Special Paste an Object on a CimEdit Screen

1. Open the application that contains the object to be pasted.
2. Select the object.

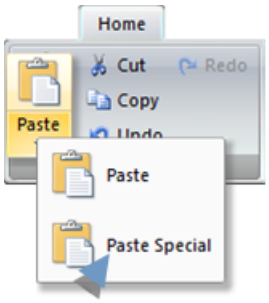


	A	B	C
1	TANK	LEVEL	TEMP
2	SECTION 1	50	54
3	SECTION 2	45	51
4	SECTION 3	65	52
5	SECTION 4	45	41
6	SECTION 5	50	65
7	SECTION 6	25	55
8	SECTION 7	68	25
9	SECTION 8	48	45
10	SECTION 9	32	75
11	SECTION 10	12	51

3. Copy your selection.
4. Click the Down arrow on the Paste button in the Home>Properties group on the CimEdit Ribbon bar.

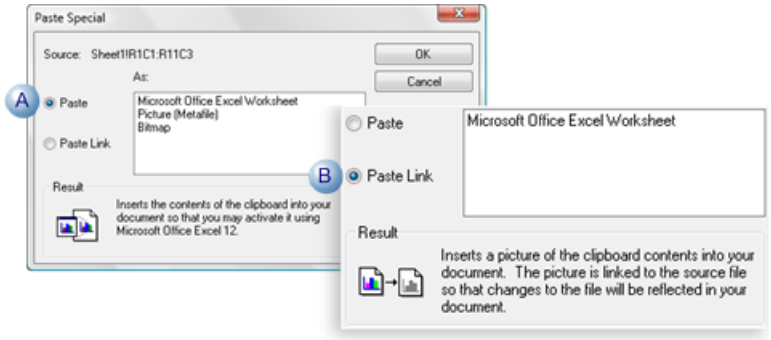


5. Select Paste Special on the Popup menu.



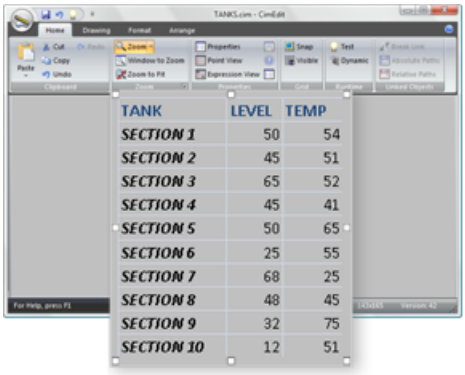
A Paste Special dialog box opens.

Check one of the following.



A	Paste	Inserts the clipboard contents; the image is static. It will not update if the copied object is updated. **Note:** You can activate the image to the application it was copied from.
B	Paste link	Inserts a picture of the clipboard contents. The image is linked to the source. As a result, it will reflect changes in the source. **Note:** You can activate the image to the application it was copied from.

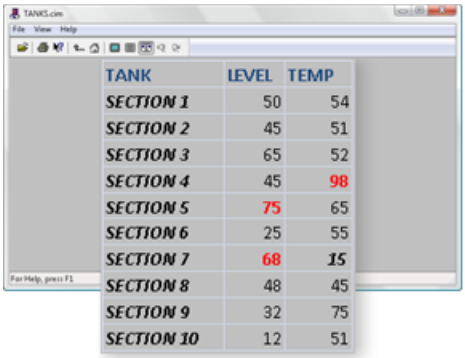
The copied image is inserted on the CimEdit screen.



You can do the following with the image.

Display the image in CimView.

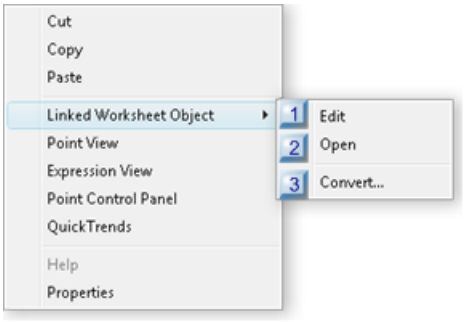
If the image was linked it will reflect an updated source.



Work with the Image in CimEdit

Right click the image; select the Popup listing for the object.

Options are:

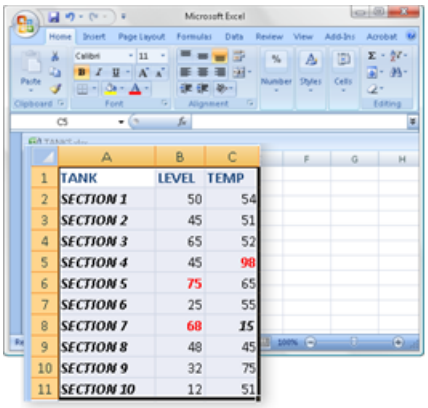


rect 171, 61, 195, 78 *(page 581)*
rect 171, 79, 189, 99 *(page 581)*
rect 171, 101, 189, 121 *(page 581)*

1	Edit
2	Open
3	Convert

1	Edit

When Edit is selected, the image opens in the application it was copied from.



2	Open

When open is selected, the image also opens in the application it was copied from.

3	Convert

Convert opens a Convert dialog box.

Selections depend on the selected special pasted object and whether or not it was linked.

Insert an ActiveX Control on a CimEdit Screen

You can easily insert an ActiveX Control anywhere on a CimEdit screen.

- Selected CIMPLICITY controls.
- CimEdit and other controls.

Selected CIMPLICITY controls

Click one of the following in the Drawing>Objects group on the CimEdit Ribbon bar.

- .NET (Components)
- Trend
- Statistical Process Control (SPC)
- Alarm Viewer
- Historical Alarm Viewer
- Portal
- Task List
- Task Indicator

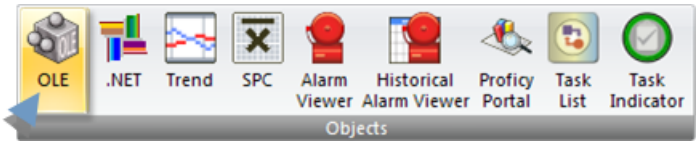
Result: The selected control is immediately placed on the screen.

CimEdit and other Controls

Several ActiveX controls that CIMPLICITY and CimEdit supports are listed in an Insert Object dialog box.

The procedure to place anyone of them on the screen is straightforward.

1. Click OLE in the Drawing>Objects group on the CimEdit Ribbon bar.

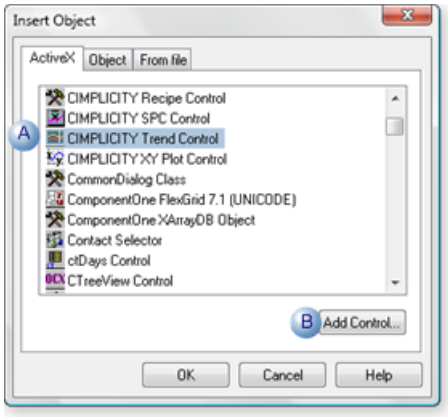
   The cursor changes into a bracket shape.

2. Move the cursor to the location for the top left corner of the object .

   An Insert Object dialog box opens.

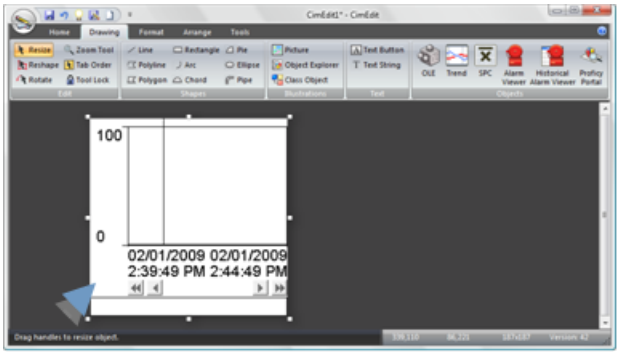
3. Select the ActiveX tab to see the list of ActiveX controls installed on your computer.

4. Do one of the following.



A	Double-click an object in the list.
B	Click Add Control; find and select an ActiveX control in the Select a Control dialog box that opens.

The selected ActiveX control is placed on your CimEdit screen.



ActiveX Control's Properties

- ActiveX control object properties.
- CIMPLICITY ActiveX control custom properties.

ActiveX Control Object Properties

Open a Properties dialog box for an ActiveX control the same way you open it for any CimEdit Smart Object.

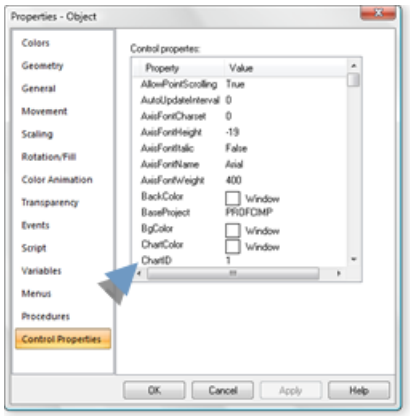
1. Select the object.

2. Do one of the following.

- Right-click the object; select properties on the Popup menu.
- Click Properties in the Home>Properties group on the CimEdit Ribbon bar.

A Properties dialog box that opens includes a Control Properties selection.

3. Select Control Properties to review and edit values for the control.



Note: Some controls provide access to dialog boxes that are generally used. However, there might be some limitations.

Example

When configuring fonts in the Font dialog box for .NET controls:

- The **Script** feature does not work for standard WinForm controls.
- Only the **Font** name works for .NET WPF controls

CIMPLICITY ActiveX Control Custom Properties

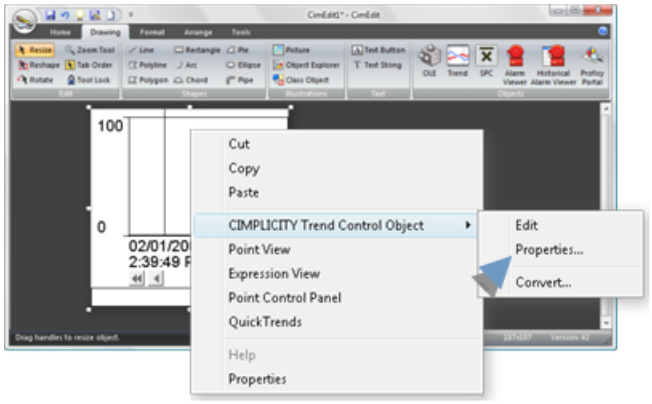
CIMPLICITY ActiveX control's have custom dialog boxes that enable you to configure the selected control in detail.

- .NET Control
- Alarm Viewer
- Document Delivery
- Historical Alarm Viewer
- Portal
- Recipes
- Statistical Process Control (SPC)
- Trending
- XY Charts

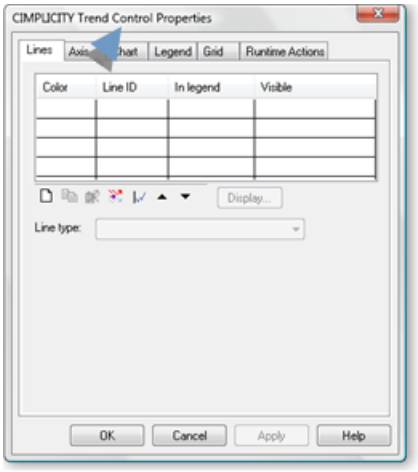
For each of the CIMPLICITY ActiveX controls:

4. Right-click the ActiveX Control.

5. Select the entry for the control object>Properties on the Popup menu.



A custom dialog box opens for the selected control.



Note: The procedure to open an ActiveX Properties dialog box may vary slightly based control that is being configured.

Other OLE Objects

Other OLE Objects

1 *(page 586)*	Insert an OLE object on a CimEdit screen.
2 *(page 587)*	Open or enable an OLE object on a CimEdit screen.
3 *(page 589)*	Convert an OLE object on a CimEdit screen.

Exactly how you manipulate OLE Objects that you place in CimEdit depends on the object.

You can see what options are available through both CimEdit's Edit menu and the object's popup menu.

An OLE object is information or an object created in other applications that is embedded or linked as an object in an application with OLE capability. You can insert OLE objects in CimEdit.

By linking or embedding, you can insert, for example:

- Clip Art pictures
- PowerPoint slides
- Microsoft Excel worksheets and charts
- Microsoft Word documents
- CIMPLICITY ActiveX controls for Trend charts, SPC charts, Recipes, and Alarm Viewer
- Third-party ActiveX controls

Linking and embedding determine where the object's data is stored. Basically:

An embedded object becomes part of the CimEdit screen.

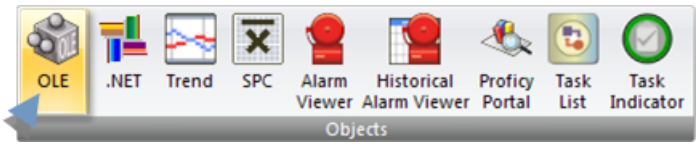
A linked object is stored in its source file, and CimEdit stores the location of the information.

You can assign object properties to OLE objects with some restrictions:

- For **Colors** properties, the object line and fill colors apply to the background of the OLE object. The line may only be partially visible. If the object does not have a transparent background, the fill color is invisible.
- For **Geometry** properties, you may define a position for the object, but not rotation or shear.
- For **Rotation/Fill** properties:
- You can rotate around any point on the screen. However, the center of the object rotates around the point and the orientation of the object remains fixed.
- Fill animation is applied to the background only. If the object does not have a transparent background, fill animation is invisible to the user.
- For **Color Animation** properties, animation is applied to the background only. If the object does not have a transparent background, color animation is invisible to the user.

1. Insert an OLE Object on a CimEdit Screen

1. Click OLE in the Drawing>Objects group on the CimEdit Ribbon bar.



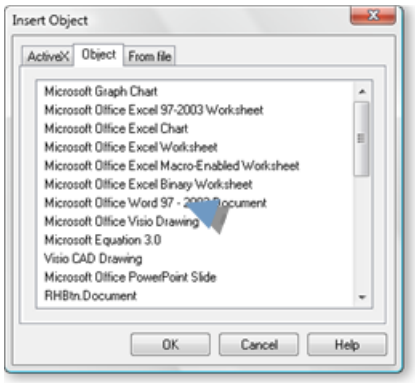
The cursor changes to into a bracket shape.

2. Move the cursor to the location for the top left corner of the object .

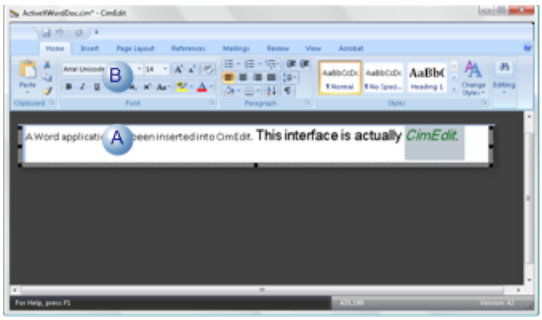
   An Insert Object dialog box opens.

3. Select the Object tab to see the list of OLE objects installed on your computer.

4. Double-click an object in the list.



The selected object and features are placed on your CimEdit screen.



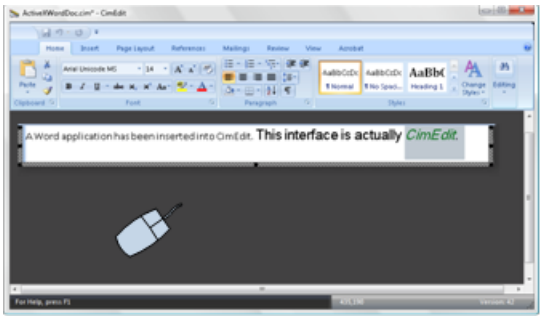
A	Object placed on the CimEdit screen.
B	OLE application features are available in the CimEdit window to modify the object.

2. Close, Edit or Open an OLE Object on a CimEdit Screen

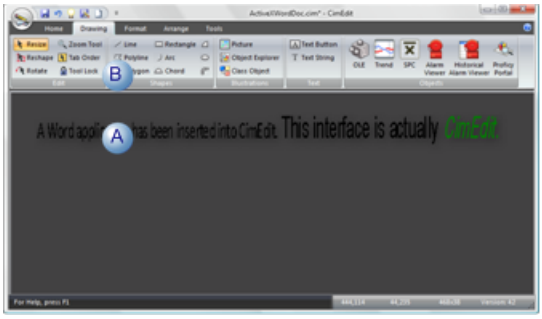
Some OLE objects can be edited in CimEdit and opened through CimEdit.

Close an OLE Application in CimEdit.

Click anywhere on the CimEdit screen outside of the inserted OLE object.



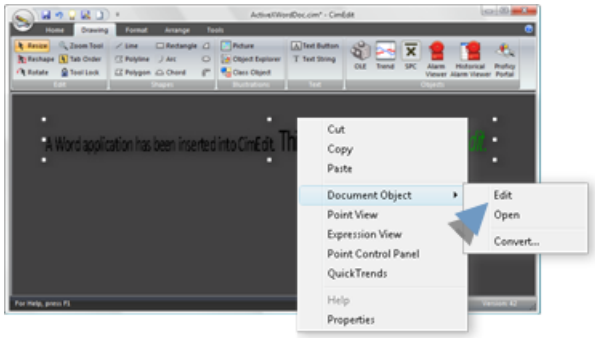
Result: The following happens.



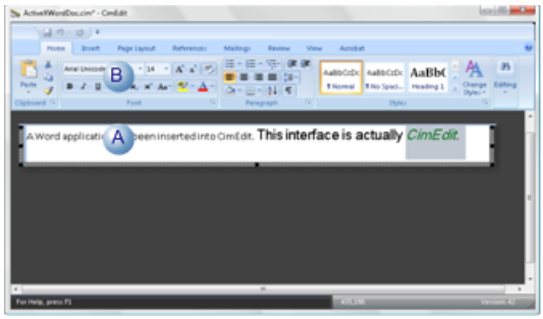
A	The object application closes.
B	The CimEdit Ribbon bar displays.

Edit an OLE Object

1. Right-click anywhere in the OLE object.
2. Select the <OLE object>Edit on the Popup menu.



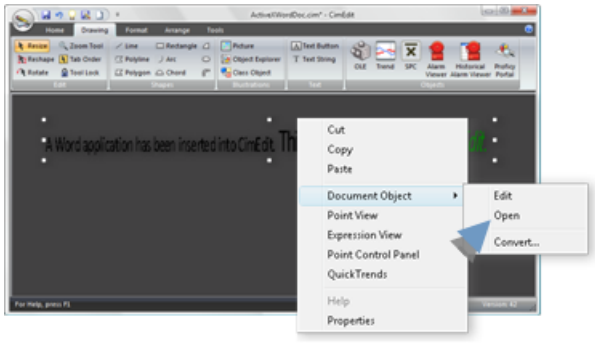
Result: The following happens.



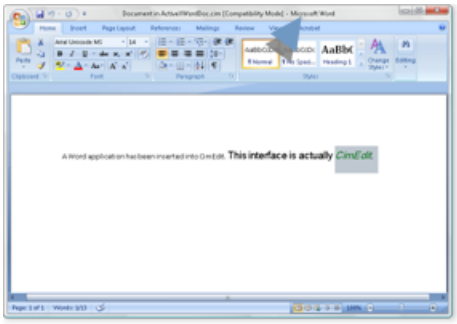
A	Object is edit-able.
B	OLE application features are available in the CimEdit window to modify the object.

Open an OLE Application

3. Right-click anywhere in the OLE object.
4. Select the <OLE object>Open on the Popup menu.

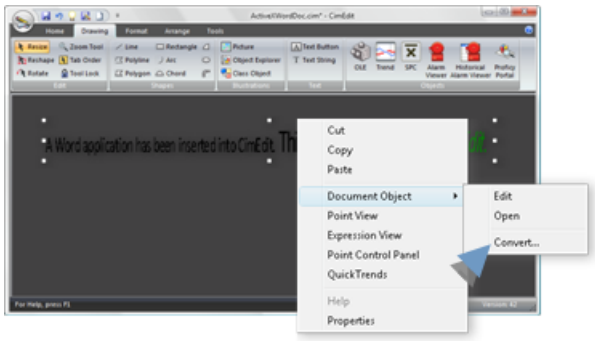


Selected application opens with the object content that was edited in CimEdit.



3. Convert an OLE Object on a CimEdit Screen

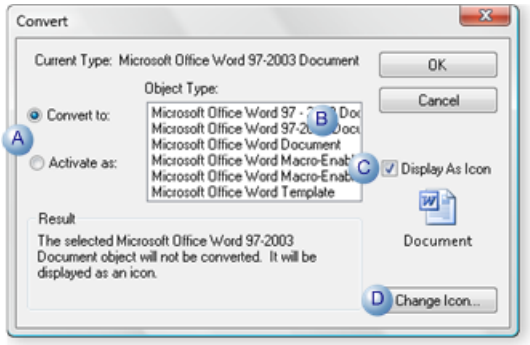
1. Right-click anywhere in the OLE object.
2. Select the <OLE object>Convert on the Popup menu.



The types that are available depend on the selected object.

A Convert dialog box opens.

3. Select the following.



A	Check one of the following: Convert to or Activate as
B	Select the type from the available options in the **Object Type** box. **Note:** The options depend on the selected object.
C	Do one of the following: Check or Clear
D	(Enabled if Display As Icon is checked) click to open a Change Icon dialog box and find and select another icon to display.

Note: CimEdit reports the result of your selections in the Result section.

4. Click OK.

The object is converted according to your specifications.

Metafile Objects to CimEdit Objects

1 *(page 591)*	Insert a Metafile object into a CimEdit screen.
2 *(page 591)*	Convert the Metafile object.

1. Insert a Metafile Object into a CimEdit Screen

1. Display the picture in the view you want in the other graphics program.
2. Select the portion of the picture you want to bring into CimEdit
3. Copy the selected portion to the clipboard.
4. Click Edit on the CimEdit menu bar.
5. Select Paste Special.
6. Select the Picture (Metafile) format from the Paste Special dialog box. This pastes the object into CimEdit as a (bitmap) picture.

1. Convert the Metafile Object

1. Select the Picture object.
2. Click Edit on the menu bar.
3. Select Convert Picture Object.

Result: The picture is converted into a group of CIMPLICITY objects.

Note: Not all of the drawing commands that are in a Picture can be accurately represented using CimEdit objects. If this is the case, your converted object will look different from your original Picture. If the results of the conversion are not acceptable, you can use the Undo command to restore the original picture object. When the original picture object is restored:

1. Click Format on the menu bar.

2. Select Ungroup.

   You can now select and manipulate the individual objects in the picture.

Text File Format Syntax

Text File Format Syntax

The text file you create with the **/converttoctx** command line option has a defined syntax. When the text file is created, the file begins with <TextFileFormat *(page 617)*>. Tokens, except for "**(**" and "**)**" should be delimited by whitespace.

Special notations used in the BNF Description include:

- <non-terminals> are displayed in italic type and enclosed with angle brackets
- **Terminals** are displayed in bold face type. Terminals are case insensitive. Terminals consisting of a single character are enclosed in quotation marks.
- <alternative> | <items> are separated by a vertical bar.
- [<optional-items>] are displayed in square brackets.
- <repetitive-items>}* are displayed in curly brackets. An asterisk (*) following the brackets indicates the item may appear 0 or more times. A plus sign (+) following the brackets indicates the item may appear 1 or more times.

BNF Syntax

BNF Syntax

A *(page 592)*	B *(page 592)*	E *(page 592)*	F *(page 592)*	Gmmi *(page 593)*	GR *(page 594)*	H *(page 594)*	K *(page 595)*	O *(page 595)*	P *(page 595)*	R *(page 595)*	S *(page 595)*	T *(page 595)*	W *(page 595)*

A

AnyAction	AnyPostiionAnim
AnyAttributeAnim	anySingleEvent
AnyEventList	AttributeAnimationTable
AnyObject	

B

	Bool		

E

- Empty
- EventOptionTable
- ExecCond
- ExprEventList

F

- FillMode

Gmmi

GmmiAbsoluteSetpointAction	GmmiKeyUpEvent
GmmiAction	GmmiLineObject
GmmiActionAnim	GmmiMethodParam
GmmiActionList	GmmiMouseDownEvent
GmmiAnnunOptionTable	GmmiMouseUpEvent
GmmiArcObject	GmmiMoveAnim
GmmiBackdrop	GmmiObject
GmmiBlink	GmmiOleEvent
GmmiBlinkAnnunAttr	GmmiOleObject
GmmiBorderAnnunAttr	GmmiParameterBlock
GmmiButtonObject	GmmiPoint
GmmiColorAnnunAttr	GmmiPointMap
GmmiContainer	GmmiPrevScreenAction
GmmiDefaultAnnun	GmmiPrintScreenAction
GmmiDocument	GmmiProcedure
GmmiEllipseShapeObject	GmmiProcedureMap
GmmiEvent	GmmiRampSetpointAction
GmmiExecAction	GmmiRectShapeObject
GmmiExpr	GmmistaticSetpointAction
GmmiExprAnim	GmmiRotateAnim
GmmiExprAnnun	GmmiScaleAnim
GmmiExprAnnunElement	GmmiScreenCloseEvent
GmmiExprEvent	GmmiScreenOpenEvent
GmmiExprHighEvent	GmmiScript
GmmiExprUpdateEvent	GmmiSetpointAction
GmmiExprValueAnnun	GmmiTabOrder
GmmiFAContainerObject	GmmiTextAnnunAttr
GmmiFAFrameObject	GmmiTextButtonObject
GmmiFileCloseAction	GmmiTextObject

GmmiFileOpenAction	GmmiTimedEvent
GmmiFillAnim	GmmiToggleSetpointAction
GmmiFontAnnunAttr	GmmiValueanim
GmmiGraphicObject	GmmiVariableAssignAction
GmmiGroupObject	GmmiVariableMap
GmmiHomeScreenAction	GmmiVariables
GmmiHorizMoveAnim	GmmiVariableSetpointAction
GmmiHorizScaleAnim	GmmiVertMoveAnim
GmmiInteriorAnnunAttr	GmmiVertScaleAnim
GmmiInvokeDispMethod Action	GmmiVisibilityAnim
GmmiInvokeScriptAction	GmmiWhileKeyDownEvent
GmmiKeyDownEvent	GmmiWhileMouseDownEvent
GMIKeyEvent	

GR

GRArc	GRObjAttr
GRArcDim	GRObjRes
GRAttrib	GROleClientItem
GRBitmapButton	GROleControl
GRBorderAttr	GROleGraphic
GRBorderRes	GROleGraphicData
GRBoundGraphic	GRPolyLine
GRButton	GRRectShapeDim
GRColorAttr	GRResource
GRColorObjAttr	GRShape
GRFontAttr	GRShapeResources
GRFontRes	GRSimpleRect
GRGrraphic	GRText
GRInteriorAttr	GRTextAttributes
GRInteriorRes	GRTextButton

H

- Help

- HexBlob

K

- KeyDownEventList
- KeyUpEventList

O

OleCurrency	OlePropBag
OleDate	OlePropStream
OleEventList	OleSafeArray
OleI64	OleU64
OleItemData	

P

Point	PositionAnimationTable
PointArray	PropElement
PointAttr	PropElementData
PointValue	

R

- Rect
- RootOptionTable

S

- Scaler
- Size

T

- TextFileformat
- TimedEventList
- ToplevelDocument
- ToplevelDocumentPO

W

- WhileKeyDownEventList

AnyAction

<AnyAction>	::=<GmmiFileOpenAction *(page 602)*> | <GmmiPrevScreenAction *(page 606)*> | <GmmiHomeScreenAction *(page 603)*> | <GmmiFileCloseAction *(page 602)*> | <GmmiExecAction *(page 600)*> | <GmmiAbsoluteSetpointAction *(page 598)*> | <GmmiInvokeDispMethodAction *(page 603)*> | <GmmistaticSetpointAction *(page 607)*> | <GmmiRampSetpointAction *(page 606)*> | <GmmiVariableSetpointAction *(page 609)*> | <GmmiToggleSetpointAction *(page 608)*> | <GmmiInvokeScriptAction *(page 603)*> | <GmmiVariableAssignAction *(page 609)*> | <GmmiPrintScreenAction *(page 606)*>

AnyAttributeAnim

<AnyAttributeAnim>	::=<GmmiFillAnim *(page 602)*> | <GmmiExprValueAnnun *(page 601)*> | <GmmiDefaultAnnun *(page 599)*> | <GmmiExprAnnun *(page 601)*> | <GmmiValueAnim *(page 608)*> | <GmmiVisibilityAnim *(page 610)*>

AnyEventList

<AnyEventList>	::=<KeyDownEventList *(page 614)*> | <KeyUpEventList *(page 614)*> | <OleEventList *(page 614)*> | <WhileKeyDownEventList *(page 617)*> | <TimedEventList *(page 617)*> | <ExprEventList *(page 597)*>

AnyObject

<AnyObject>	::=<GmmiLineObject *(page 604)*> | <GmmiRectShapeObject *(page 606)*> | <GmmiEllipseShapeObject *(page 600)*> | <GmmiTextObject *(page 608)*> | <GmmiGroupObject *(page 602)*> | <GmmiOleObject *(page 605)*> | <GmmiTextButtonObject *(page 608)*> | <GmmiArcObject *(page 598)*> | <GmmiFAContainerObject *(page 601)*> | <GmmiFAFrameObject *(page 601)*>

Note: a <GmmiFAFrameObject> can only appear in a <GmmiFAContainerObject>

AnyPositionAnim

<AnyPositionAnim>	::=<GmmiHorizMoveAnim *(page 603)*> | <GmmiVertMoveAnim *(page 609)*> | <GmmiRotateAnim *(page 607)*> | <GmmiHorizScaleAnim *(page 603)*> | <GmmiVertScaleAnim *(page 609)*>

AnySingleEvent

<AnySingleEvent>	::=<GmmiMouseUpEvent *(page 604)*> | <GmmiMouseDownEvent *(page 604)*> | <GmmiWhileMouseDownEvent *(page 610)*> | <GmmiScreenOpenEvent *(page 607)*> | <GmmiScreenCloseEvent *(page 607)*>

AttributeAnimationTable

<AttributeAnimationTable> ::=	**"(" GmmiOptionTable** {<AnyAttributeAnim *(page 596)*>}+ **")"**

Bool

<Bool> ::=	<number *(page 619)*>
.	Zero means false; non-zero means true

Empty

<Empty> **::=**

This non-terminal is used for emphasis to designate that nothing should appear.

EventOptionTable

<EventOptionTable> ::=	**"(" GmmiOptionTable** {<AnySingleEvent *(page 596)*>}* {<AnyEventList *(page 596)*>}* **")"**

ExecCond

<ExecCond> ::=	**"(" ExecCond** <expression.string *(page 620)*> <message.string *(page 620)*> **")"**

ExprEventList

<ExprEventList> ::=	**"(" GmmiOptionTable** {<GmmiExprHighEvent *(page 601)*> | <GmmiExprUpdateEvent *(page 601)*> }+ **")"**

FillMode

<FillMode> ::=	<mode.number *(page 619)*>

Where mode is one of:

0	NoFill
1	From Bottom
2	From Top
3	From Left
4	From Right

GmmiAbsoluteSetpointAction

<GmmiAbsoluteSetpointAction> ::=	**"(" GmmiAbsoluteSetpointAction** <GmmiSetpointAction *(page 607)*> <PointValue *(page 616)*> **")"**

GmmiAction

<GmmiAction> ::=	<flags.number *(page 619)*>
Valid flags are:	

0x01	Confirm

GmmiActionAnim

<GmmiActionAnim> ::=	<GmmiExprAnim *(page 600)*> <flags.number *(page 619)*> <ExecCond *(page 597)*>

Valid flags are:

0x02	Has Action (slider setpoint or in-place edit setpoint)
0x04	Confirmed

GmmiActionList

<GmmiActionList> ::=	**"(" GmmiActionList** {<AnyAction *(page 596)*>*} **")"**

GmmiAnnunOptionTable

<GmmiAnnunOptionTable> ::=	**"(" GmmiOptionTable** [<GmmiColorAnnunAttr *(page 599)*>] [<GmmiTextAnnunAttr *(page 608)*>] [<GmmiBlinkAnnunAttr *(page 599)*>] [<GmmiInteriorAnnunAttr *(page 603)*>] [<GmmiBorderAnnunAttr *(page 599)*>] [<GmmiFontAnnunAttr *(page 602)*>]")"

GmmiArcObject

<GmmiArcObject> ::=	**"(" GmmiArcObject** <GmmiGraphicObject *(page 602)*> <GRArc *(page 610)*> **")"**

GmmiBackdrop

<GmmiBackdrop> ::=	**"(" GmmiBackdrop** <border.GRBorderAttr *(page 610)*> <interior.GRInteriorAttr *(page 612)*> **")"**

GmmiBlink

<GmmiBlink> ::=	**"(" GmmiBlink** <rate.number *(page 619)*> <interior.GRInteriorAttr *(page 612)*> <fillInterior.GRInteriorAttr *(page 612)*> <border.GRBorderAttr *(page 610)*> **")"**

GmmiBlinkAnnunAttr

<GmmiBlinkAnnunAttr> ::=	**"(" GmmiBlinkAnnunAttr ")"**

GmmiBorderAnnunAttr

<GmmiBorderAnnunAttr> ::=	**"(" GmmiBorderAnnunAttr** <flags.number *(page 619)*> <line.GRBorderAttr *(page 610)*> **")"**

Valid flags are:	

0x0001	Ignore line style
0x0002	Ignore line color
0x0004	Ignore line width

GmmiButtonObject

<GmmiButtonObject> ::=	<GmmiGraphicObject *(page 602)*>

GmmiColorAnnunAttr

<GmmiColorAnnunAttr> ::=	**"(" GmmiColorAnnunAttr** <color.GRColorAttr *(page 611)*> **")"**

GmmiContainer

<GmmiContainer> ::=	**"(" GmmiContainerObject** <GmmiObject *(page 605)*> **"(" GmmiObjectPtrList** {<AnyObject *(page 596)*>}* **")"** <flags.number *(page 619)*> [<overrideInterior.GRInteriorAttr *(page 612)*>] [<overrideBorder.GRBorderAttr *(page 610)*>] [<overrideFont.GRFontAttr *(page 611)*>] <GmmiBackdrop *(page 598)*> **")"**

GmmiDefaultAnnun

<GmmiDefaultAnnun> ::=	**"(" GmmiDefaultAnnun** <pointID.string *(page 620)*> <element.number *(page 619)*> <type.number *(page 619)*> **")"**

Where type is 0 for Boolean points and 1 for numeric points.

GmmiDocument

<GmmiDocument> ::=	**"(" GmmiDocumentObject** <GmmiContainer *(page 599)*> **")"**

GmmiEllipseShapeObject

<GmmiEllipseShapeObject> ::=	**"(" GmmiEllipseShapeObject** <GmmiGraphicObject *(page 602)*> <GRSimpleRect *(page 613)*> **")"**

GmmiEvent

<GmmiEvent> ::=	**"(" GmmiEvent** <actionFlags.number *(page 619)*> <procedureName.string *(page 620)*> | <GmmiInvokeScriptAction *(page 603)*> <userParameter.string *(page 620)*> **")"**

Valid actionFlags are:

1	Call procedure name
2	Invoke script

GmmiExecAction

<GmmiExecAction> ::=	**"(" GmmiExecAction** <GmmiAction *(page 598)*> <flags.number *(page 619)*> <command.string *(page 620)*> <directory.string *(page 620)*> <arguments.string *(page 620)*> <confirmMsg.string *(page 620)*> **")"**

Valid flags are:

0x01	Minimize

GmmiExpr

<GmmiExpr> ::=	**"(" GmmiExpr** <expression.string *(page 620)*> **")"**

GmmiExprAnim

<GmmiExprAnim> ::=	**"(" GmmiExprAnim** <flags.number *(page 619)*> [<expr.string *(page 620)*>] [<minValue.number *(page 619)*> <maxValue.number *(page 619)*>] **")"**

Valid flags are:

0x01	Use configured limits (determines if limits appear)
0x02	Has expression (determines if expression appears)

GmmiExprAnnun

<GmmiExprAnnun> ::=	**"(" GmmiExprAnnun** <count.number *(page 619)*> {<GmmiExprAnnunElement *(page 601)*> }* **")"**

There must be exactly count <GmmiExprAnnunElement> entries.

GmmiExprAnnunElement

<GmmiExprAnnunElement> ::=	**"(" GmmiExprAnnunElement** <GmmiExprAnim *(page 600)*> <GmmiAnnunOptionTable *(page 598)*> **")"**

GmmiExprEvent

<GmmiExprEvent> ::=	**"(" GmmiExprEvent** <GmmiEvent *(page 600)*> <expr.GmmiExpr *(page 600)*> **")"**

GmmiExprHighEvent

<GmmiExprHighEvent> ::=	**"(" GmmiExprHightEvent** <GmmiExprEvent *(page 601)*> **")"**

GmmiExprUpdateEvent

<GmmiExprUpdateEvent> ::=	**"(" GmmiExprUpdateEvent** <GmmiExprEvent *(page 601)*> **")"**

GmmiExprValueAnnun

<GmmiExprValueAnnun> ::=	**"(" GmmiExprValueAnnun** <GmmiExprAnim *(page 600)*> **")"**

GmmiFAContainerObject

<GmmiFAContainerObject> ::=	**"(" GmmiFAContainerObject** <GmmiContainer *(page 599)*> <FillMode *(page 597)*> <fillInterior.GRInteriorAttr *(page 612)*> **")"**

Note: A GmmiFAContainerObject only contains GmmiFAFrameObjects

GmmiFAFrameObject

<GmmiFAFrameObject> ::=	**"(" GmmiFAFrameObject** <GmmiContainer *(page 599)*> <frameSelector.GmmiExpr *(page 600)*> <FillMode *(page 597)*> <fillInterior.GRInteriorAttr *(page 612)*> **")"**

GmmiFileCloseAction

<GmmiFileCloseAction> ::=	**"(" GmmiFileCloseAction** <GmmiAction *(page 598)*> **")"**

GmmiFileOpenAction

<GmmiFileOpenAction> ::=	**"(" GmmiFileOpenAction** <GmmiAction *(page 598)*> <pathname.string *(page 620)*> <flags.number *(page 619)*> <position.Point *(page 615)*> <zoom.number *(page 619)*> <project.string *(page 620)*> **")"**

Valid flags are

0x01	Open screen action (rather than overlay screen action)
0x02	Captive

GmmiFillAnim

<GmmiFillAnim> ::=	**"(" GmmiFillAnim** <GmmiExprAnim *(page 600)*> **")"**

GmmiFontAnnunAttr

<GmmiFontAnnunAttr> ::=	**"(" GmmiFontAnnunAttr**<flags.number *(page 619)*> <font.GRFontAttr *(page 611)*>**")"**

Valid flags are:

0x0001	Ignore font name
0x0002	Ignore font style
0x0004	Ignore font size
0x0008	Ignore font strikeout
0x0010	Ignore font underline
0x0020	Ignore font script

GmmiGraphicObject

<GmmiGraphicObject> ::=	<GmmiObject *(page 605)*>

GmmiGroupObject

<GmmiGroupObject> ::=	**"(" GmmiGroupObject** <GmmiContainer *(page 599)*> <FillMode *(page 597)*> <fillInterior.GRInteriorAttr *(page 612)*> **")"**

GmmiHomeScreenAction

<GmmiHomeScreenAction> ::=	**"(" GmmiHomeScreenAction** <GmmiAction *(page 598)*> **")"**

GmmiHorizMoveAnim

<GmmiHorizMoveAnim> ::=	**"(" GmmiHorizMoveAnim** <GmmiMoveAnim *(page 605)*> **")"**

GmmiHorizScaleAnim

<GmmiHorizScaleAnim> ::=	**"(" GmmiHorizScaleAnim** <GmmiScaleAnim *(page 607)*> **")"**

GmmiInteriorAnnunAttr

<GmmiInteriorAnnunAttr> ::=	**"(" GmmiInteriorAnnunAttr** <flags.number *(page 619)*> <fill.GRInteriorAttr *(page 612)*> **")"**

Valid flags are:

0x0001	Ignore fill style
0x0002	Ignore fill color1
0x0004	Ignore fill color2
0x0008	Ignore fill pattern
0x0010	Ignore whether one color or two color gradient shading
0x0020	Ignore number of gradient shades
0x0040	Ignore gradient shade style
0x0080	Ignore gradient shade variant

GmmiInvokeDispMethod Action

<GmmiInvokeDispMethodAction> ::=	**"(" GmmiInvokeDispMethodAction** <GmmiAction *(page 598)*> <objectName.string *(page 620)*> <methodName.string *(page 620)*> <GmmiParameterBlock *(page 605)*> **")"**

GmmiInvokeScriptAction

<GmmiInvokeScriptAction> ::=	**"(" GmmiInvokeScriptAction** <GmmiAction *(page 598)*> <objectName.string *(page 620)*> <scriptEntryPointName.string *(page 620)*> <GmmiParameterBlock *(page 605)*> **")"**

GmmiKeyDownEvent

<GmmiKeyDownEvent> ::=	**"(" GmmiKeyDownEvent** <GmmiKeyEvent *(page 604)*>**")"**

GmmiKeyEvent

<GmmiKeyEvent> ::=	**"(" GmmiKeyEvent** <GmmiEvent *(page 600)*> <keyCode.number *(page 619)*> **")"**

GmmiKeyUpEvent

<GmmiKeyUpEvent> ::=	**"(" GmmiKeyUpEvent** <GmmiKeyEvent *(page 604)*> **")"**

GmmiLineObject

<GmmiLineObject> ::=	**"(" GmmiLineObject** <GmmiGraphicObject *(page 602)*> <GRPolyLine *(page 613)*> **")"**

GmmiMethodParam

<GmmiMethodParam> ::=	<paramType.number *(page 619)*> <GmmiExpr *(page 600)*>

Where paramType is one of:

0	Empty parameter
1	Input parameter
2	Output parameter: A set-point will be performed on a point.
3	Method result: A set-point will be performed on a point.

GmmiMouseDownEvent

<GmmiMouseDownEvent> ::=	**"(" GmmiMouseDownEvent** <GmmiEvent *(page 600)*> **")"**

GmmiMouseUpEvent

<GmmiMouseUpEvent> ::=	**"(" GmmiMouseUpEvent** <GmmiEvent *(page 600)*> **")"**

GmmiMoveAnim

<GmmiMoveAnim> ::=	<GmmiActionAnim *(page 598)*> <offset.number *(page 619)*>

Valid flags are:

0x02	Slider Animation
0x04	Confirm Silder

GmmiObject

<GmmiObject> ::=	**"(" GmmiObject** <name.string *(page 620)*> <flags.number *(page 619)*> [<GmmiBlink *(page 599)*>] <Help *(page 614)*> [<ToplevelDocumentPO *(page 617)*>] <RootOptionTable *(page 616)*> <GmmiTabOrder *(page 608)*>**")"**

The ToplevelDocumentPO data must be present if and only if this object is a top level document.

GmmiOleEvent

<GmmiOleEvent> ::=	**"(" GmmiOleEvent** <GmmiEvent *(page 600)*> <eventName.string *(page 620)*> **")"**

GmmiOleObject

<GmmiOleObject> ::=	**"(" GmmiOleObject** <GmmiGraphicObject *(page 602)*> <GROleGraphic *(page 612)*> **")"**

GmmiParameterBlock

<GmmiParameterBlock>::=	**"(" GmmiParameterBlock** <countParams.number *(page 619)*> {<GmmiMethodParam *(page 604)*>}* **")"**

GmmiPoint

<GmmiPoint> ::=	**"(" GmmiPoint** <pointID.string *(page 620)*> <PointAttr *(page 615)*> **")"**

GmmiPointMap

<GmmiPointMap> ::=	**"(" GmmiPointMap** {<GmmiPoint *(page 605)*>}* **")"**

GmmiPrevScreenAction

<GmmiPrevScreenAction> ::=	**"(" GmmiPrevScreenAction** <GmmiAction *(page 598)*> **")"**

GmmiPrintScreenAction

<GmmiPrintScreenAction> ::=	**"(" GmmiPrintScreenAction** <GmmiAction *(page 598)*> **")"**

GmmiProcedure

<GmmiProcedure> ::=	**"(" GmmiProcedure** <name.string *(page 620)*> <description.string *(page 620)*> <procFlags.number *(page 619)*> <confMsg.string *(page 620)*> <successMsg.string *(page 620)*> <failureMsg.string *(page 620)*> <execFlags.number *(page 619)*> <GmmiActionList *(page 598)*> **")"**

Where procFlags is:

0x01	Halt On Error

Where execFlags is one of:

0x01	Always Confirm
0x02	Never Confirm

GmmiProcedureMap

<GmmiProcedureMap> ::=	**"(" GmmiProcedureMap** {<GmmiProcedure *(page 606)*>}* **")"**

GmmiRampSetpointAction

<GmmiRampSetpointAction> ::=	**"(" GmmiRampSetpointAction** <GmmistaticSetpointAction *(page 607)*> <altValue.number *(page 619)*> <flags.number *(page 619)*> **")"**

Valid flags are:

0x01	Allow direct edit of value

GmmiRectShapeObject

<GmmiRectShapeObject> ::=	**"(" GmmiRectShapeObject** <GmmiGraphicObject *(page 602)*> <GRSimpleRect *(page 613)*> **")"**

GmmistaticSetpointAction

<GmmistaticSetpointAction> ::=	**"(" GmmistaticSetpointAction** <GmmiSetpointAction *(page 607)*> <value.number *(page 619)*> **")"**

GmmiRotateAnim

<GmmiRotateAnim> ::=	**"(" GmmiRotateAnim** <GmmiExprAnim *(page 600)*> <centerOffset.Point *(page 615)*> <minAngle.number *(page 619)*> <maxAngle.number *(page 619)*> **")"**

GmmiScaleAnim

<GmmiScaleAnim> ::=	<GmmiExprAnim *(page 600)*> <scaleOrigin.number *(page 619)*> <scaleFactor.number *(page 619)*>

Where ScaleOrigin is one of:

0	bottom for vertical, left for horizontal
1	center
2	top for vertical, right for horizontal

GmmiScreenOpenEvent

<GmmiScreenOpenEvent> ::=	**"(" GmmiScreenOpenEvent** <GmmiEvent *(page 600)*> **")"**

GmmiScreenCloseEvent

<GmmiScreenCloseEvent> ::=	**"(" GmmiScreenCloseEvent** <GmmiEvent *(page 600)*> **")"**

GmmiScript

<GmmiScript> ::=	**"(" GmmiScript** <string *(page 620)*> **")"**

GmmiSetpointAction

<GmmiSetpointAction> ::=	**"(" GmmiSetpointAction** <GmmiAction *(page 598)*> <pointID.string *(page 620)*> <element.number *(page 619)*> <flags.number *(page 619)*> **")"**

Valid flags are:

0x01	Point not validated

Valid element numbers are:

-1	Non -array points and text strings
0 through n-1	Index of an array element to be set, where the array has n elements. 0 is the first element and n-1 is the last element of the array.

GmmiTabOrder

<GmmiTabOrder> ::=	**"(" GmmiTabOrder** <number *(page 619)*> **")"**

GmmiTextAnnunAttr

<GmmiTextAnnunAttr> ::=	**"(" GmmiTextAnnunAttr** <text.string *(page 620)*> **")"**

GmmiTextButtonObject

<GmmiTextButtonObject> ::=	**"(" GmmiTextButtonObject** <GmmiButtonObject *(page 599)*><GRTextButton *(page 614)*> **")"**

GmmiTextObject

<GmmiTextObject> ::=	**"(" GmmiTextObject** <GmmiGraphicObject *(page 602)*> <GRText *(page 613)*> **")"**

GmmiTimedEvent

<GmmiTimedEvent> ::=	**"(" GmmiTimedEvent** <GmmiEvent *(page 600)*> <eventFrequencyMS.number *(page 619)*> **")"**

GmmiToggleSetpointAction

<GmmiToggleSetpointAction> ::=	**"(" GmmiToggleSetpointAction** <GmmiSetpointAction *(page 607)*> **")"**

GmmiValueAnim

<GmmiValueAnim> ::=	**"(" GmmiValueAnim** <GmmiExprAnim *(page 600)*> <format.string *(page 620)*> <formatType.number *(page 619)*> **")"**

Where format is a **printf()** style format string and formatType is one of:

1	No format
2	Configured
3	Custom
4	General
5	Integer
6	Text
7	Real

GmmiVariableAssignAction

<GmmiVariableAssignAction> ::=	**"(" GmmiVariableAssignAction** <GmmiAction *(page 598)*> <variableID.string *(page 620)*> <value.string *(page 620)*> <flags.number *(page 619)*> **")"**

Valid flags are

0x01	Prompt for the variable value.

GmmiVariableMap

< GmmiVariableMap > ::=	**"(" GmmiVariables** {<GmmiVariables *(page 609)*>}* **")"**

GmmiVariables

<GmmiVariables> ::=	**"(" GmmiVariables** <variableID.string *(page 620)*> <value.string *(page 620)*> **")"**

GmmiVariableSetpointAction

<GmmiVariableSetpointAction> ::=	**"(" GmmiVariableSetpointAction** <GmmiSetpointAction *(page 607)*> **")"**

GmmiVertMoveAnim

<GmmiVertMoveAnim> ::=	**"(" GmmiVertMoveAnim** <GmmiMoveAnim *(page 605)*> **")"**

GmmiVertScaleAnim

<GmmiVertScaleAnim> ::=	**"(" GmmiVertScaleAnim** <GmmiScaleAnim *(page 607)*> **")"**

GmmiVisibilityAnim

<GmmiVisibilityAnim> ::=	**"(" GmmiVisibilityAnim** <GmmiExprAnim *(page 600)*> **")"**

GmmiWhileKeyDownEvent

<GmmiWhileKeyDownEvent> ::=	**"(" GmmiWhileKeyDownEvent** <GmmiKeyEvent *(page 604)*> <eventFrequencyMS.number *(page 619)*> **")"**

GmmiWhileMouseDownEvent

<GmmiWhileMouseDownEvent> ::=	**"(" GmmiWhileMouseDownEvent** <GmmiEvent *(page 600)*> <eventFrequencyMS.number *(page 619)*> **")"**

GRArc

<GRArc> ::=	<GRShape *(page 613)*> <GRArcDim *(page 610)*>

GRArcDim

<GRArcDim> ::=	<GRRectShapeDim *(page 613)*> <startAngle.number *(page 619)*> <endAngle.number *(page 619)*>

GRAttrib

<GRAttrib> ::=	<Empty *(page 597)*>

GRBitmapButton

<GRBitmapButton> ::=	<GRButton *(page 611)*> <bitmapButtonFlags.number *(page 619)*> <bitmapSize.size *(page 616)*>

GRBorderAttr

<GRBorderAttr> ::=	**"(" Border** <GRColorObjAttr *(page 611)*> <width.number *(page 619)*> <lineType.number *(page 619)*> <lineEndStyle.number *(page 619)*> **")"**

GRBorderRes

<GRBorderRes> ::=	<GRObjRes *(page 612)*> <GRBorderAttr *(page 610)*>

GRBoundGraphic

<GRBoundGraphic> ::=	<GRGraphic *(page 611)*>

GRButton

<GRButton> ::=	<GRBoundGraphic *(page 611)*> <face.GRInteriorRes *(page 612)*> <buttonFlags.number *(page 619)*>

GRColorAttr

<GRColorAttr> ::=	**"(" Color** "**0**" **")"**
	| **"(" Color** "**1**" <rgb.number *(page 619)*> **")"**
	| **"(" Color** "**2**" <index.number *(page 619)*> **")"**
	| **"(" Color** "**3**" <systemColor.number *(page 619)*> **")"**

GRColorObjAttr

<GRColorObjAttr> ::=	<GRObjAttr *(page 612)*> <GRColorAttr *(page 611)*>

GRFontAttr

<GRFontAttr> ::=	**"(" Font** <GRObjAttr *(page 612)*> <fontName.string *(page 620)*> <angle.number *(page 619)*> <height.number *(page 619)*> <weight.number *(page 619)*> <fontAttr.number *(page 619)*> **")"**

GRFontRes

<GRFontRes> ::=	<GRObjRes *(page 612)*> <GRFontAttr *(page 611)*>

GRGraphic

<GRGraphic> ::=	**"(" Extents** <extent.Rect *(page 616)*> <rawExtent.Rect *(page 616)*> **")"**

GRInteriorAttr

<GRInteriorAttr> ::=	**"(" Interior** <GRColorObjAttr *(page 611)*> <color2.GRColorAttr *(page 611)*> <fillStyle.number *(page 619)*> [<gradientColors.number *(page 619)*> <gradientShades.number *(page 619)*>] **")"**

The GRColorAttr contained in the GRColorObjAttr is used by the solid, patterned and gradient fill styles.

color2 is used only by the patterned and gradient fill styles.

GRInteriorRes

<GRInteriorRes> ::=	<GRObjRes *(page 612)*> <GRInteriorAttr *(page 612)*>

GRObjAttr

<GRObjAttr> ::=	<Empty *(page 597)*>

GRObjRes

<GRObjRes> ::=	<Empty *(page 597)*>

GROleClientItem

<GROleClientItem> ::=	**"("GROleClientItem** <itemNumber.number *(page 619)*> **")"**

GROleControl

<GROleControl> ::=	**"("GROleControl** <OlePropBag *(page 615)*> **")"**

GROleGraphic

<GROleGraphic> ::=	<GRShape *(page 613)*> <itemScale.Scaler *(page 616)*> <Rect *(page 616)*> **"("GROleClientItem** <itemNumber.number *(page 619)*> **")"**

GROleGraphicData

<GROleGraphicData> ::=	<GROleControl *(page 612)*> | <GROleClientItem *(page 612)*>

GRPolyLine

<GRPolyLine> ::=	<GRShape *(page 613)*> <PointArray *(page 615)*> <lineFlags.number *(page 619)*> **"("Arrow** <arrowWidth0.number *(page 619)*> <arrowLength0.number *(page 619)*> <arrowStyle0.number *(page 619)*> **")" "("Arrow** <arrowWidth1.number *(page 619)*> <arrowLength1.number *(page 619)*> <arrowStyle1.number *(page 619)*> **")"**

GRRectShapeDim

<GRRectShapeDim> ::=	**"(" Size** <aDim.number *(page 619)*> <bDim.number *(page 619)*> **")"** <center.Point *(page 615)*> <angle.number *(page 619)*> <xShear.number *(page 619)*> <yShear.number *(page 619)*>

GRResource

<GRResource> ::=	<Empty *(page 597)*>

GRShape

<GRShape> ::=	<GRBoundGraphic *(page 611)*> <GRShapeResources *(page 613)*>

GRShapeResources

<GRShapeResources> ::=	<GRBorderRes *(page 610)*> <interior.GRInteriorRes *(page 612)*> <fillInterior.GRInteriorRes *(page 612)*> <fillMode.number *(page 619)*> <percentFill.number *(page 619)*>

GRSimpleRect

<GRSimpeRect> ::=	<GRShape *(page 613)*> <GRRectShapeDim *(page 613)*>

GRText

<GRText> ::=	<GRGraphic *(page 611)*> <GRTextAttributes *(page 613)*>

GRTextAttributes

<GRTextAttributes> ::=	<GRFontAttr *(page 611)*> <GRInteriorRes *(page 612)*> <GRBorderRes *(page 610)*> <textAlign.number *(page 619)*> <textAnchor.Point *(page 615)*> <text.string *(page 620)*>

GRTextButton

<GRTextButton> ::=	<GRButton *(page 611)*> <GRText *(page 613)*>

Help

<Help> ::=	**"(" Help** <contextIDExpr.string *(page 620)*> <text.string *(page 620)*> <file.string *(page 620)*> **")"**

HexBlob

<HexBlob> ::=	**"(" HexBlob** { <string *(page 620)*> }* **")"**

KeyDownEventList

<KeyDownEventList> ::=	**"(" GmmiOptionArray** { <GmmiKeyDownEvent *(page 604)*> }+ **")"**

KeyUpEventList

<KeyUpEventList> ::=	**"(" GmmiOptionArray** { <GmmiKeyUpEvent *(page 604)*> }+ **")"**

OleCurrency

<OleCurrency> ::=	**"(" OleCurrency** <currencyValue.string *(page 620)*> **")"**

OleDate

<OleDate> ::=	**"(" OleDate** <dateValue.string *(page 620)*> **")"**

OleEventList

<OleEventList> ::=	**"(" GmmiOptionArray** { <GmmiOleEvent *(page 605)*> }+ **")"**

OleI64

<OleI64> ::=	**"(" OleI64** <I64Value.string *(page 620)*> **")"**

OleItemData

<OleItemData> ::=	**"(" GROleClientItem** <itemNumber.number *(page 619)*> <HexBlob *(page 614)*> **")"**

OlePropBag

<OlePropBag> ::=	**"(" OlePropBag** <objectCLSID.guid *(page 618)*> {<PropElement *(page 616)*>}* **")"**

OlePropStream

<OlePropStream> ::=	**"(" OlePropStream** <objectCLSID.guid *(page 618)*> <HexBlob *(page 614)*> **")"**

OleSafeArray

<OleSafeAray> ::=	**"(" OleSafeArray** <elementType.number *(page 619)*> <dimensionCount.number *(page 619)*> {**"(" OleArrayBound** <lowerBound.number *(page 619)*> <elementCount.number *(page 619)*> **")"** }+ { <PropElementData *(page 616)*> }+ **")"**

OleU64

<OleU64> ::=	**"(" OleU64** <U64Value.string *(page 620)*> **")"**

Point

<Point> ::=	**"(" Point** <x.number *(page 619)*> <y.number *(page 619)*> **")"**

PointArray

<PointArray> ::=	**"(" PointArray** {<Point *(page 615)*>}* **")"**

PointAttr

<PointAttr> ::=	**255**
	| <type.number *(page 619)*> <length.number *(page 619)*><elements.number *(page 619)*>

PointValue

<PointValue> ::=	<value.number *(page 619)*> | <value.string *(page 620)*>

PositionAnimationTable

<PositionAnimationTable> ::=	**"(" GmmiOptionTable** {<AnyPositionAnim *(page 596)*>}+ **")"**

PropElement

<PropElement> ::=	<propName.string *(page 620)*> <PropElementData *(page 616)*>

PropElementData

<PropElementData> ::=	<numValue.number *(page 619)*> | <strValue.string *(page 620)*> | <OlePropBag *(page 615)*> |<OlePropStream *(page 615)*> |<OleCurrency *(page 614)*> |<OleDate *(page 614)*> |<OleI64 *(page 615)*> |<OleU64 *(page 615)*> |<OleSafeArray *(page 615)*>

Rect

<Rect> ::=	**"(" Rect** <left.number *(page 619)*> <top.number *(page 619)*> <right.number *(page 619)*> <bottom.number *(page 619)*> **")"**

RootOptionTable

<RootOptionTabet> ::=	**"("Empty")"** | **"(" GmmiOptionTable** [<PositionAnimationTable *(page 616)*>] [<AttributeAnimationTable *(page 596)*>] [<EventOptionTable *(page 597)*>] [<GmmiScript *(page 607)*>] [<GmmiVariableMap *(page 609)*>] **")"**

Scaler

<Scaler> ::=	<xNum.number *(page 619)*> <yNum.number *(page 619)*> <xDen.number *(page 619)*> <yDen.number *(page 619)*>

Size

<Size> ::=	**"(" Size** <width.number *(page 619)*> <height.number *(page 619)*> **")"**

TextFileFormat

<TextFileFormat> ::=	**"(" Version** <number *(page 619)*> **")" "(" DocumentSummary ")"** <ToplevelDocument *(page 617)*> "**{Newline}(**" **OleItems** { <OleItemData *(page 615)*> **")"**

TimedEventList

<TimedEventList> ::=	**"(" GmmiOptionTable** {<GmmiTimedEvent *(page 608)*>}+ **")"**

ToplevelDocument

<ToplevelDocument> ::=	**"(" GmmiToplevelDocument** <GmmiDocument *(page 600)*> <initialPosition.Point *(page 615)*> <Size *(page 616)*> <zoom.number *(page 619)*> **")"**

ToplevelDocumentPO

<ToplevelDocumentPO> ::=	<GmmiPointMap *(page 605)*> <ambientForeground.GRColorAttr *(page 611)*> <ambientBackground.GRColorAttr *(page 611)*> <ambientFont.GRFontAttr *(page 611)*> <GmmiProcedureMap *(page 606)*>

WhileKeyDownEventList

<WhileKeyDownEventList> ::=	**"(" GmmiOptionArray** { <GmmiWhileKeyDownEvent *(page 610)*> }+ **")"**

Token Syntax

Token Syntax

Here is the modified BNF syntax for tokens. Tokens (except for <string>) may not contain embedded whitespace.

- alpha-char
- digit
- exponent-char
- guid
- hex-digit
- mantissa
- named-char
- non-negative-integer

- number
- octal-digit
- sign
- string
- string-char
- symbol
- symbol-char
- token

alpha-char

<alpha-char> ::=	"**A**" through "**Z**" and "**a**" through "**z**"

digit

<digit> ::=	"**0**" | "**1**" | "**2**" | "**3**" | "**4**" | "**5**" | "**6**" | "**7**" | "**8**" | "**9**"

exponent-char

<exponent-char> ::=	"**e**" | "**E**" | "**d**" | "**D**"

guid

<guid> ::=	"**{**"xxxxxxxx"**-**"xxxx"**-**"xxxx"**-**"xxxx"**-**" xxxxxxxxxxxx"**}**"
	where x == <hex-digit *(page 618)*>

hex-digit

<hex-digit> ::=	<digit *(page 618)*> | "**A**" | "**B**" | "**C**" | "**D**" | "**E**" | "**F**" | "**a**" | "**b**" | "**c**" | "**d**" | "**e**" | "**f**"

mantissa

<mantissa> ::=	{<digit *(page 618)*> }+ ["**.**" {<digit *(page 618)*> }*] | {<digit *(page 618)*> }* "**.**" {<digit *(page 618)*> }+

named-char

<named-char> ::=	One of the following characters:
	Null (&h00)
	Bell (&h07)
	Backspace (&h08)

	CharacterTabulation or **Tab** (&h09)
	LineFeed or **NewLine** (&h0A)
	LineTabulation (&h0B)
	FormFeed (&h0C)
	CarriageReturn or **Return** (&h0D)
	Escape (&h1B)
	QuotationMark (&h3F)
	LeftCurlyBracket or **LeftBrace** (&h7B**)**
	RightCurlyBracket or **RightBrace** (&h7D**)**
	Delete (&h7F)

Named characters use the ISO 1064 naming convention for characters. Convenience names for some characters are also defined.

non-negative-integer

<non-negative-integer> ::=	{<digit *(page 618)*> }+
	| "**&H**" {<hex-digit *(page 618)*>}+
	| "**&h**" {<hex-digit *(page 618)*>}+
	| "**&O**" {<octal-digit *(page 619)*>}+
	| "**&o**" {<octal-digit *(page 619)*>}+

number

<number> ::=	[<sign *(page 619)*>] <non-negative-integer *(page 619)*> | [<sign *(page 619)*>] <mantissa *(page 618)*> [<exponent-char *(page 618)*> [<sign *(page 619)*>] {<digit *(page 618)*> }+]

octal-digit

<octal-digit> ::=	"**0**" | "**1**" | "**2**" | "**3**" | "**4**" | "**5**" | "**6**" | "**7**"

sign

<sign> ::=	"+" | "**-**"

string

<string> ::=	""" {<string-char *(page 620)*>}* """
	| <string> {"**&**" <string>}*

string-char

<string-char> ::=	any character except LineFeed (&h0A), "**{**", or """
	| "**{**" <non-negative-integer *(page 619)*> "**}**"
	| "**{**" <named-char *(page 618)*> "**}**"

symbol

<symbol> ::=	<alpha-char *(page 618)*> {<symbol-char *(page 620)*>}*

symbol-char

<symbol-char> ::=	<alpha-char *(page 618)*> <digit *(page 618)*> "_"

token

<token> ::=	**"("**
	| **")"**
	| <guid *(page 618)*>
	| <number *(page 619)*>
	| <symbol *(page 620)*>
	| <string *(page 620)*>

CimEdit/CimView Screen Management

Screen Performance Issues

If your runtime requirements are such that your users need to be able to quickly access screens in CimView, consider the following.

- Size memory.
- Size the screen cache.
- Preload the screen cache.
- Prevent swaps to the pagefile.
- Adjust the touch interval.

Tip: : You can use the Windows Performance Monitor to review CimView performance counters to monitor the performance of CimView during any single session.

Size Memory

For best performance on screen transfers, make sure that you have at least 60 MB of RAM on the computer that is displaying CimView screens.

Size the Screen Cache

The default cache size for CimView is eight (8) screens. You may use the GSM_CACHE_SIZE global parameter to change the default.

Note: All screens in the cache, whether being displayed or not, continue to have their points updated from the Point Manager. If you have many screens in the cache with rapidly changing points, this may affect your system performance.

Screens are put in the cache as they are displayed by the users. If space is needed for a new screen and the cache is full, the oldest screen in the cache is removed to make space for the new screen.

Preload the Screen Cache

For faster screen access, you can install your initial CimView screen with the **/loadcache** <cachefile> argument. The screens that you specify in the <cachefile> file will be pre-loaded into the screen cache. In addition, you can specify which screens you want to lock into the cache.

Screens that are locked into the cache are never removed, no matter how old they are.

The default screen cache size is eight (8). You can use the **GSM_CACHE_SIZE** global parameter to change this default.

The screen cache size used by CimView is the larger of the default screen cache size or the number of files you specify in the <cachefile> file. For example, if **GSM_CACHE_SIZE** is set to 10 and you list three screens in the <cachefile> file, the screen cache size is set to 10. However, if you list 12 screens in the <cachefile> file, the screen cache size is set to 12.

The <cachefile> file is a text file. Each line of the text file contains the name of a screen to be preloaded in the cache when the primary CimView screen opens. Add a "Lock" parameter after those screens that you want CimView to keep in the cache. Screens that do not have a "Lock" parameter will be removed from the cache to make room for a more recently used screens.

An example of a <cachefile> file is:

```
"screen1.cim" Lock
"screen2.cim"
"screen3.cim" Lock
"screen4.cim"
"screen5.cim" Lock
"screen6.cim" Lock
"screen7.cim" Lock
"screen8.cim" Lock
```

If the <cachefile> name is not fully qualified, its location depends on how users start CimView:

- When starting CimView from a shortcut, the working directory for the <cachefile> is determined by the **Start in** field of the Properties dialog.
- When starting CimView from the command prompt, the working directory for the <cachefile> is static to where the user is in the Command Prompt.

Prevent Swaps in the Screen File

When a user attempts to transfer from one screen to another in the cache after a long period of inactivity, the transfer may be slow because the cache has been swapped out to the pagefile. In addition, if the current screen has no animation, it can also be swapped out to the pagefile.

If you need an immediate response, no matter how long the screen has been displayed, install your initial CimView screen with the **/TouchActive** , **/TouchDyn** , and **/TouchStat** arguments. These options keep the screens from being swapped out.

The first time a user invokes a CimView screen with one of the **/Touch…** arguments, the **Touch Documents Interval** is created in the Registry under the following.

32-bit machines

```
HKEY_CURRENT_USER\Software\GE Fanuc\CIMPLICITY\HMI\<version>\CimView\Touch
Documents Interval
```

64-bit machines

```
HKEY_CURRENT_USER\Software\Wow6432Node\GE Fanuc\CIMPLICITY\HMI\<version>
\CimView\Touch Documents Interval
```

Its value is in milliseconds, and defaults to 420000 (7 minutes).

Adjust the Touch Interval

If you are having still having problems with cached screens transferring slowly after a long period of inactivity, you can adjust the rate that CimView touches the screens to a value below 60000 (1 minute).

- Run regedit.exe.
- Enter a new value in the following.

32-bit machines

```
HKEY_CURRENT_USER\Software\GE Fanuc\CIMPLICITY\HMI\<version>\CimView\Touch
Documents Interval
```

64-bit machines

```
HKEY_CURRENT_USER\Software\WOW6432Node\GE Fanuc\CIMPLICITY\HMI\<version>
\CimView\Touch Documents Interval
```

⚠ **Important:** Adjusting this value too low will slow animation on the screens.

Improve the Screen Caching Algorithm

You can use the global parameter GSM_CACHE_USE_VARS to improve the screen caching algorithm. When the value of GSM_CACHE_USE_VARS is set to Y, the cache logic searches for a screen with the same screen name, project, and initial variables. The default value of the parameter is N.

When the value of the parameter is N or if the parameter does not exist, the cache logic searches for a screen with the same screen name and project

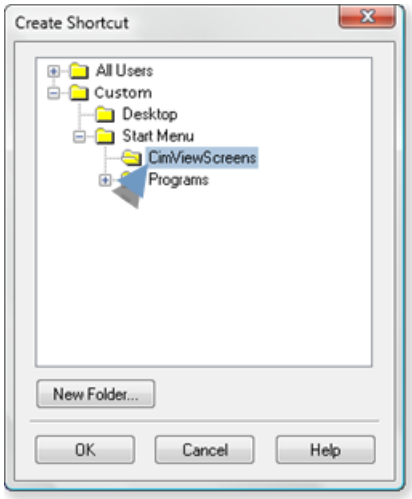
Create a CimView Screen Start Menu Item or Shortcut

- CimView screen Start menu.
- CimView screen shortcut.

CimView Screen Start Menu

1. Click the CimEdit button .
2. Select Install to Program Manager on the CimEdit menu.

   A Create Shortcut dialog opens.



   Menus in the Start menu are represented as folders.

3. Select the folder where you want to place the shortcut.

   Note: Click New Folder to create a custom folder for CimView shortcuts.

4. Click **OK**.

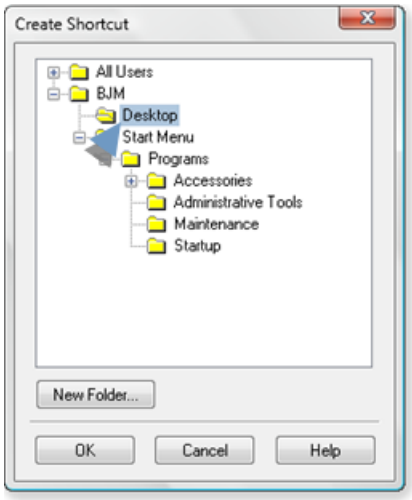
   Result: The shortcut is placed on the Start Menu in the selected folder

   CimView Screen Shortcut

5. Click the CimEdit button .
6. Select Install to Program Manager on the CimEdit menu.

   A Create Shortcut dialog opens.

7. Select Desktop.



8. Click **OK**.

A shortcut is placed on the desktop.

Add Arguments to the Command Line for a CimView Shortcut

You can create a Windows shortcut on the desktop, in folders, or in the Start menu as a convenient way to run CimView. Command line arguments can be added after creating the shortcut to configure how Cimview runs.

- Create a CimView shortcut.
- Add arguments to the command line for a CimView shortcut.

Create a CimView shortcut in the Windows Start menu

1. Right-click the Windows Start menu. Select Explore All Users.
2. An Explorer window opens. Expand the Start Menu folder under All Users.
3. Expand the Programs folder. Double-click the folder for the program group from which you want to launch Cimview.

   Note: The example below shows how to add a Cimview shortcut to the Startup program group, but you can add a Cimview shortcut to any program group or create a folder for a new program group that will appear whenever you use the Start menu.

   CAUTION: Adding a CimView shortcut to the Startup group causes CimView to run automatically whenever you start your computer. Configure System Boot Options correctly when launching CimView at system startup.

4. In the right pane of the Explorer window you will see the startup programs, if any. Right-click in that pane and select New > Shortcut from the pop-up menu.

5. A Browse dialog box appears. Navigate to the folder where your CimView screen is located and expand the that folder.

6. Select a CimView file and click OK.

7. The Create Shortcut dialog box appears. Click Next to name the shortcut and then click Finish.

   Result: The new shortcut appears in the selected folder.

   Once you have created a shortcut for a CimView screen, you can add arguments to its command line to control user access.

   To add arguments to the command line for a CimView shortcut

8. Select Properties... from the shortcut's menu.

   The Shortcut to <screen_name> Properties dialog opens.

9. At the start of the **Target** field, before the target pathname, insert **C imView.exe** followed by the option (or options) you want.

   Example

   If you have created a primary screen for a project (myproj.gef), your initial screen is **ini.cim** , and you do not want users to exit that screen, you can change the command line for the screen's CimView icon:

   From

   C:\cimplicity\myproj\ini.cim

   to

   cimview.exe /waitforproject myproj /noexit C:\cimplicity\myproj\ini.cim

   Note: The `/waiforproject` argument has been added to insure that the project being viewed has completed its startup. This assumes that CIMPLICITY System Boot Options have been configured to automatically load the relevant project when the system boots up. When boot options have been set to run as a viewer, The `/wait` argument should be used instead. The `/waitforproject` or `/wait` argument can be omitted if you are added the shortcut to a program group other than Startup and you are sure the project associated with this screen is running already.

10. Click **OK** to save your changes and exit the dialog.

You can also use the command line to start multiple CimView screens from a single shortcut. When you do this and use a command line option that affects the primary window, the first file in the list is designated as the primary window. For example, if you use **/alwaysmaximized,** the first screen in the list is maximized, and all other screens are displayed in their normal windows.

Command Line Options

Command Line Options

There are currently three types of command line options available for CimEdit and CimView screens:

They are:

- Command line arguments
- CimEdit shortcut options
- Interactive execution options

Command Line Arguments

Command Line Arguments

Command line arguments include:

/alwaysmaximized
/autostart
/captive
/geometry [<size>][<position>][<displaysurface>]
/HideNavBar <name>
/HideNavChildBar <name>
/keypad
/loadcache <cachefile>
/LoadNav <name>
/loadpassword <password configuration file name>
/LoadScript
/LoadScriptCfg
/maximized
/MaxUpdate <milliseconds>

/minimized
/MinUpdate <milliseconds>
/NavNode <name><nodename>
/noexit
/nomenutitle
/noopen
/noPointTargets
/noresize
/noSetpoints
/project <name>
/setvar
/TouchActive
/TouchDyn
/TouchStat
/Viewer
/wait [<time>]
/waitforproject <name>
/WaitUpdate <milliseconds>
/zoomtobestfit
Options for CimEdit shortcuts.
Options that can be executed interactively.

/alwaysmaximized

For: CimView (and WebSpace) only

Displays the primary CimView window in a maximized state. The user will not be allowed to resize the primary window. The window will not rise to the top when the user clicks on it (this prevents it from obscuring other windows on the user's terminal screen

This option is only applied if it is used with the first CimView window opened. This window is known as the primary window. If used with subsequent open commands, it is equivalent to **/maximize** .

Example

cimview.exe /alwaysmaximized C:\myproj\scr.cim

❗ **Important:** The first opened screen is always the mainframe when you use `/alwaysmaximized`. If you close this screen; all other open screens will close also.

/autostart

For: CimView (and WebSpace) only

Automatically starts the project associated with a CimView screen before launching this screen.

Example

cimview.exe /autostart C:\myproj\scr.cim

❗ **Important:** Use **/autostart** on a local computer only. Only use this argument when you are certain that the associated project is not already running or in the process of starting up.

/captive

For: CimView only

Displays the primary CimView window in captive state. When the user opens this window, the Explorer shuts down. Other screens are displayed on top of the primary window, and the user will not be able to go below (see anything below) the primary window.

Example for a shortcut

cimview.exe /captive /noexit C:\myproj\scr.cim

Result: When the CimView screen is open:

- The user can use Alt+Tab to navigate through all opened CimView Windows.
- The Alt+Esc and Ctrl+Esc sequences are disabled.
- The Ctrl+Alt+Esc sequence works as normal

You can use any other command line options in conjunction with **/captive** . In particular, note the following:

- Use `/alwaysmaximized` if you want to display the initial window in the maximized state.
- Use `/noexit` if you do not want users to exit from CimView.

If you do not use `/noexit`, when a user exits CimView, the Explorer restarts, and any processes in the Startup program group are restarted.

❗ **Important:**

- On some Windows 8 systems, the Charm bar might display when the `/captive` command line argument is used. If this occurs a `CimDesktop` utility can be used as the command and, in most instances, will cause the Charm bar to remain hidden.

The syntax is:

CimDesktop CimView /captive c:\myproj\scr.cim

- `/keypad (page 632)` and `/nomenutitle (page 636)` are also currently supported when using this tool.

/geometry [<size>][<position>][<displaysurface>]

For: CimView (and WebSpace) only

Sizes and positions the CimView screen in one or across two monitors when it opens.

The arguments for this command line option are:

[<size>][<position>][<displaysurface>]

Where

Argument	Specifies the:
<size>	is either a percent of the selected display area or an integer that specifies a fixed value in pixels.
<position>	is distance from the left or right and top or bottom edges of the screen.
<displaysurface>	can cover one or more monitors.

Entries correspond to parameters available for the `GefScreen.PositionWithGeometry` method in the CimEdit/CimView object model.

Example

cimview.exe /geometry 50%x60%-0-20%@2W c:\projects\ecimp\screens\airhouse.cim

Opens the airhouse.cim screen in an ecimp project and:

- Sizes the screen to be:
- 50% of the width of monitor 2's working area.
- 60% of the of the width of monitor 2's working area.
- Positions the screen so the:
- Right edge of the screen is at the right edge of the monitor's working area

- Bottom of the screen is offset from the bottom of the working area of the monitor by 20% of the working height of the monitor.

/HideNavBar<name>

For CimView

Opens an indicated CimView screen with no Navigation bar.

`/HideNavBar` takes priority over settings in the CimView Options *(page 763)* dialog box to hide the navigation bar.

If a CimView screen, except the indicated screen, is already open, the screen indicated in the command line will open in a new window; it will not display a Navigation bar.

Example

```
cimview.exe /HideNavBar c:\myproj\scr.cim
```

guide: Guidelines

- Make sure the path is included with the CimView filename.
- If there are spaces in the <path\screen name>, surround it with quotes.
- If the screen indicated in the command line is already open, a new window will not open.

/HideNavChildBar

For CimView

Opens an indicated CimView screen that displays the navigation bar without the Child *(page 701)* menu bar.

`/HideNavChildBar` takes priority over settings in the CimView Options *(page 763)* dialog box to hide the child navigation bar.

If a CimView screen, except the indicated screen, is already open, the screen indicated in the command line will open in a new window; it will display a navigation bar without the Child menu bar.

Example

```
cimview.exe /HideNavChildBar c:\myproj\scr.cim
```

guide: Guidelines

- Make sure the path is included with the CimView filename.
- If there are spaces in the <path\screen name>, surround it with quotes.

- If the screen indicated in the command line is already open, a new window will not open.

/keypad

For: CimView (and WebSpace) only

Displays a keypad window on the user's screen whenever the user has to perform a Variable Setpoint action in a procedure. To enter a new setpoint value, the user clicks on the appropriate keys in the keypad window rather than entering them from a keyboard.

This option is only applied if it is used with the first CimView window opened. This window is known as the primary window. If used with subsequent open commands, it will be ignored.

Example

cimview /keypad C:\myproj\scr.cim

Important

- On some Windows 8 systems, keypad may not work well if the standard `/keypad` command line argument (above) is used. If this occurs a `CimDesktop` utility can be used as the command and, in most instances, will display the keypad correctly.

The syntax is:

CimDesktop CimView /keypad c:\myproj\scr.cim

- `/captive` *(page 629)* and `/nomenutitle` *(page 636)* are also currently supported when using this tool.

/loadcache <cachefile>

For: CimView only

Pre-loads the CimView screen cache with the files contained in <cachefile>. If the number of files in the <cachefile> is greater than the default cache size, this number overrides the default.

/LoadNav <name>

For CimView

Opens a blank CimView screen that displays the indicated navigation bar.

`/LoadNav<name>` takes priority over settings in the CimView Options *(page 763)* dialog box to load a selected Navigation bar.

Example

cimview.exe /LoadNav c:\myproj\navfile.navbar

guide: Guideline

- Make sure the path is included with the Navigation bar filename.
- If there are spaces in the <path\screen name>, surround it with quotes.
- The blank CimView screen with the Navigation bar will open regardless of whether or not another CimView screen is open.

/loadpassword <password configuration file name>

For: CimView (and WebSpace) only

Alerts CimView to a text file that contains a list of projects, user IDs and passwords that CimView will use to grant or deny access when a user attempts to view a screen.

If CIMPLICITY is configured to prompt a user for login, CimView looks in this file for the correct user ID-password combination when the user attempts to login.

If the user is already logged in, CimView ignores values in the file.

The file format is:

PROJECT NAME|USER ID|PASSWORD

The entries must be in upper case letters.

If the project name is omitted the user ID-password combination is used for all CIMPLICITY projects.

Important: The password configuration file must be located in the data subdirectory of the CIMPLICITY installation directory (e.g. C:\Program Files\Proficy\Proficy CIMPLICITY\Data).

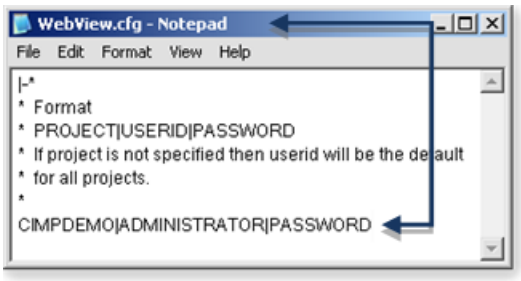
Example

Your CIMPLICITY Web site server uses the **WebView.cfg** file.

/ loadpassword WebView.cfg

Where

WebView.cfg is the specification for a file that requires user passwords to open CimView/WebView screens either through your CIMPLICITY Web site or at a specific URL for a single WebView screen.



/LoadScript<file to open>

For: CimEdit and CimView (and WebSpace)

Enables a common script file to be shared among all screens loaded in CimView. Functions that are included in this common file will be loaded when CimView is first run, and will be callable as if they were included in the Basic code for the local screen.

If the /LoadScript *(page 536)* argument is specified for a file after CimView is already running, its contents will be appended to the list of common functions with the lowest precedence.

Example

/LoadScript scriptFileName

Where

scriptFileName is a file specification to specify a common script file.

Important: The **/LoadScript** command line option, which is a counterpart to the global parameter **GSM_GLOBAL_SCRIPT** , is an essential command line option for CimEdit if you are going to use it for CimView. This is because you can only configure a call directly in the global scripts if the call is also loaded into CimEdit and, as a result, CimEdit knows what script entry points are available.

/LoadScriptCfg<file to open>

For: CimEdit and CimView

Opens a text file that contains a list of global scripts that will be loaded.

- The file names must be enclosed in "" and may contain logicals such as `SITE_ROOT`, `BSM_ROOT` etc.
- The command line option itself can contain logicals.
- Multiple files may appear on the same line.

Example

File name	GlobalScripts.txt
File contents	`"SITE_ROOT:screens\tuning.cmsrt"` `"c:\testscripts\mytest.cms"` `"BSM_ROOT:scripts\ClassScripts.bcl"`

Note: If a script is not found the file name with different extensions will be tried, e.g. .cmsrt will be substituted for .cms and .cms will be substituted for .cmsrt.

- `/LoadScriptCfg` is supported by CimEdit Global Script features *(page 540)* .
- The global parameter GSM_GLOBAL_SCRIPTCFG can also be used to specify the global configuration file.

/maximized

For: CimEdit and CimView (and WebSpace)

Opens the screen at its maximum size. The screen takes up the entire terminal screen.

The user will be able to resize the window after it opens.

Example

cimview.exe /maximized C:\myproj\scr.cim

/MaxUpdate <milliseconds>

For: CimView running as a WebSpace client only

Time between updates while a WebSpace Client user is using the mouse and/or keyboard.

/minimized

For: CimEdit and CimView (and WebSpace)

Opens the screen at its minimum size. The screen appears on the Taskbar only.

The user can click on the screen in the Taskbar to open it.

Example

cimview.exe /minimized C:\myproj\scr.cim

/MinUpdate <milliseconds>

For: CimView running as a WebSpace client only

Number of milliseconds between updates when the WebSpace client user is not using the keyboard and/or mouse.

Note: /MinUpdate <milliseconds> is normally used after /WaitUpdate <milliseconds> *(page 642)* .

/NavNode <name><nodename>

For CimView

Opens the CimView screen that is associated with the indicated node name.

`/LoadNav<name><nodename>` takes priority over settings in the CimView Options *(page 763)* dialog box to load an indicated CimView screen (associated with the node name) that displays the selected Navigation bar.

Example

cimview.exe /NavNode c:\myproj\navfile.navbar NODE3

guide: Guideline

- Make sure the path is included with the Navigation bar filename.
- If there are spaces in the <path\screen name>, surround it with quotes.
- If the node indicated in the command line is already open, a new window will not open.

/noexit

For: CimView (and WebSpace) only)

Will not let the user exit the primary CimView window.

The Exit menu item is removed from the File menu, and the Close menu item and its Alt+F4 shortcut key are removed from the Control menu. **Close Screen** actions are ignored in the primary window.

This option is only applied if it is used with the first CimView window opened. This window is known as the primary window. If used with subsequent open commands, it will be ignored.

Example

cimview.exe /noexit C:\myproj\scr.cim

/nomenutitle

For: CimView (and WebSpace) only)

Removes the menu bar and title bar from the primary CimView window and all subsequent windows that you open.

This option is applied to the new windows and to any windows that are opened from within CimView. If a later CimView command is executed without the option, the option is disabled for those windows and any new windows opened from within CimView.

Exarmple

cimview.exe /nomenutitle C:\myproj\scr.cim

Important:

- A button or other object that enables a user to close CimView is necessary. If one is not present, the machine will have to be rebooted to re-enable the standard desktop.
- An issue could occur when `/nomenutitle` was used that would cause a screen to lose focus and remain out of focus.

This issue occurred, for example, if a user:

1. Opened a CimView screen.
2. Opened a second screen using an `Open Screen` procedure.
3. Opened a dialog box on the second screen.
4. Clicked the original screen to put it back in focus.
5. Clicked the dialog box on the second screen.

The dialog box would stay in focus only briefly. Instead focus would flip back and forth between the dialog box and the original screen. The second screen would remain out of focus.

A utility, `CimDesktop`, can work around this issue.

The syntax is:

CimDesktop CimView /nomenutitle c:\myproj\scr.cim

Result:

- A selected CimView screen with no menu or title bar takes over the desktop.
- Screens, Popup menus and other features that are associated with the selected screen perform successfully.
- All other applications are disabled and hidden.
- The standard desktop is re-enabled when CimView is closed.
- `/captive (page 629)` and `/keypad (page 632)` are also currently supported when using this tool.

/noopen

For: CimView (and WebSpace) only)

Lets a user open only the CimView screens that are explicitly mentioned in Open Screen and Overlay Screen actions. Open Screen and Overlay Screen actions that do not specify a screen are ignored.

In addition, the Open and Open Window menu items and the file list are removed from the File menu, and the File Open Toolbar button is disabled.

This option is applied to the new windows and to any windows that are opened from within CimView. If a later CimView command is executed without the option, the option is disabled for those windows and any new windows opened from within CimView.

Example

cimview.exe /noopen C:\myproj\scr.cim

/noPointTargets

For: CimView (and WebSpace) only

Prevents the point targets, e.g. Point Control Panel and quick trends, from being available from Point View or the right mouse menu.

/noresize

For: CimView (and WebSpace) only)

Prevents a user from resizing any CimView windows that the user displays.

This option is applied to the new windows and to any windows that are opened from within CimView. If a later CimView command is executed without the option, the option is disabled for those windows and any new windows opened from within CimView.

Example

cimview.exe /noresize C:\myproj\scr.cim

/noSetpoints

For: CimView (and WebSpace) only

Disables setpoints; all setpoint requests will fail.

/project <name>

For: CimEdit and CimView (and WebSpace)

Sets the base project to <name>.

The data for all unqualified points on the CimView screens will be requested from the named project.

This option is applied to the new windows and to any windows that are opened from within CimView. If a later CimView command is executed without the option, the previous value for the default project continues to be used.

Example

cimview.exe /project myproj C:\myproj\scr.cim

/setvar<name><value>

For: CimView (and WebSpace) only)

(Configured on the current screen), sets variables and their initial values that a CimView screen uses when it opens from or overlays the current screen.

The arguments for this command line option are:

Argument	Specifies the:
<name>	Variable name, which can be qualified
<value>	Initial value of the variable the open screen uses.

Example

/setvar "\Button1\Tank" "0" /setvar "\Button2\Tank" "0" /setvar "Value" ""

sets the qualified variable \ **Button1\Tank** to the initial value **0**; the qualified variable \ **Button2\Tank** to the initial value **0** and the unqualified variable, **Value** to no specified initial value.

Use double quotes as follows:

1. If the name contains non-alphanumeric characters, e.g. spaces, hyphens, surround it with double quotes.
2. If the value has embedded double quotes in it, double the embedded double-quotes.
3. If the value contains non-alphanumeric characters, e.g. spaces, hyphens, embedded double-quotes, surround the entire value with double quotes.

   Example

	Name	Value	Syntax
No spaces	**var1**	**value1**	**/setvar var1 value1**

Space in name	**my var**	**tank1**	**/setvar "my var" tank1**
Space in name and Embedded quotes in value	**my var**	**"tank1"**	**/setvar "my var" """tank1"""**
Embedded quotes in value	**var1**	**"hello, world!"**	**/setvar var1 """hello, world!"""**
Embedded quotes in value	**var1**	**a"b**	**/setvar var1 "a""b"**

/TouchActive

For: CimView only

Keeps the documents currently being displayed swapped into memory. Without this flag, only documents in the cache are touched periodically to keep them swapped into memory.

Example

```
cimview.exe /TouchActive myproj C:\myproj\scr.cim
```

- Review screen performance issues.
- Command Line Arguments list.

/TouchDyn

For: CimView only

Keeps documents with many dynamic objects swapped into memory.

Example

cimview.exe /TouchDyn myproj C:\myproj\scr.cim

/TouchStat

For: CimView only

Keeps documents with many static objects swapped into memory.

This is faster than **/TouchDyn** , but dynamic parts of a screen may be swapped out.

Example

```
cimview.exe /TouchStat myproj C:\myproj\scr.cim
```

/viewer

Automatically (and quietly) starts the Router, if it has not started already.

Example

A shortcut was created to start CimView.

```
Cimview.exe /Viewer ..\myproj\scr.cim
```

/wait [<time>]

For: CimView (and WebSpace) only)

Makes CimView wait for the Router to start before opening the screen. Note that CimView does not try to start the Router. The Router must be started by another method. Typically, this option is used when configuring CimView to start at boot or when the user logs in to Windows.

You can specify a time, in seconds, to wait.

If the router has not started and a:

• Wait time is specified:	The screen opens at the end of the specified time whether or not the router has started.
• Wait time is not specified:	CimView waits forever for the router to start.

If the router is already started:

- CimView starts immediately

Example

```
cimview.exe /wait 60 C:\myproj\scr.cim
```

/waitforproject <name>

For: CimView (and WebSpace) only)

Makes CimView wait for the project to start before opening the window.

The project is considered started after CIMPLICITY Program Control terminates normally. Note that individual programs in the project may be in the Running or Failed state.

Example

```
cimview.exe /waitforproject myproj C:\myproj\scr.cim
```

Important: Use `/waitforproject` on a local computer only.

/WaitUpdate <milliseconds>

For: CimView running as a WebSpace client only

Number of milliseconds that WebSpace will continue to send updates to the WebSpace client at the normal rate after the user stops using the keyboard and mouse.

/zoomtobestfit

For: CimEdit and CimView (and WebSpace)

Initially places the primary CimView window in **Zoom To Best Fit** mode. All subsequent windows are also displayed in this mode.

This option is applied to the new windows and to any windows that are opened from within CimView.

Example

```
cimview.exe /zoomtobestfit C:\myproj\scr.cim
```

Options for CimEdit Shortcuts

CimEdit Shortcut Options

You cannot modify the command line for the **CimEdit** icon in a project Workbench. You can only add options to a shortcut.

The format for entering an option to the command line is:

<pathname> <option>

For example, if you want to force CimEdit to work offline, even if a CIMPLICITY project is running, the command line would be:

c:\cimplicity\exe\CimEdit.exe /offline

Command line arguments include:

- /offline
- /project <name>

/offline

For: CimEdit and CimView

Forces CimEdit to work offline, even if a CIMPLICITY project is running.

Example

cimedit.exe /offline

/project <name>

For: CimEdit and CimView

Sets the base project to <name>.

The base project is the project that will be used to fully qualify any unqualified Point IDs. This does not affect the Point IDs stored in the file. Point IDs that are entered as unqualified will be stored unqualified.

The option is only recognized when it is used with the first CimEdit window, and is applied to all subsequent windows until they are closed.

Example

cimedit.exe /project myproj

Interactive Execution Options

Interactive Execution Options

- Interactive execution options overview.
- Command line syntax.
- Command line arguments.

Interactive Execution Options Overview

Currently, there are three utilities available in CimEdit that you can access from the command prompt launched from your Configuration Cabinet. These utilities let you:

- Convert .asc files to .cim or .cimrt files, or upgrade .cim files from an earlier release.
- Display a dialog listing the Point IDs used in the screen file.
- Convert .cim files to text file format.

To access any of these utilities on an MMI or Server:

1. Make sure that CimEdit is not running on your terminal.

2. Open the project in the Configuration cabinet.
3. Click Tools on the menu bar.
4. Select Command Prompt. This ensures that all project pointers are correct.
5. Enter the command that will start the utility.

   If you are on a Viewer, launch the Command Prompt from the Start menu.

   Command Line Syntax

   You can execute each utility for one file, or a list of files. The command line syntax is:

   `cimedit.exe <option> <filename> [<filename> ...]`

   Where

   <option> is the command line argument.

   <filename> includes path and filename

   Command Line Arguments

   Command line arguments include:

   - **/convert**
   - **/converttoctx**
   - **/converttocimrt**
   - **/dumppoints**

/convert

For: CimEdit only

Reads each of the files on the command line and writes them out again, changing the extension to **.cim** .

This utility can be used to

- Convert **.asc** or **.ctx** files to **.cim** files, or
- Upgrade **.cim** files from an older release.

Note:

- When upgrading files from an older release, save a copy of the .cim files in another directory before executing this utility since this option will overwrite the original files.

- Make sure to include in the command line the complete path to the file.
- No CimEdit window is displayed, and CimEdit exits when done. When the conversion is complete, check the Status Log for error messages. Messages longer than 79 characters are in multiple log entries, and the continuation lines begin with "… ".

Example

```
cimedit.exe /convert C:\MyProj\scr1.asc C:\MyProj\scr2.ctx
```

/converttoctx

For: CimEdit only

Reads each of the files on the command line and writes them out again in text format, changing the extension to **.ctx** .

You can use the text format to make modifications to the screen file with a text editor.

Note:

- Do not use text files as the primary file format in a production environment, as they take longer to read and process.
- Make sure to include in the command line the complete path to the file.
- (Required) Update any older version .cim screen to the current version before you convert it to a .ctx file.

This is required in order for the conversion to be completely successful.

Example

```
cimedit.exe /converttoctx C:\Proj\scr1.cim C:\Proj\scr2.cim
```

/converttocimrt

For: CimEdit only

Reads each of the files on the command line and writes them out again as a runtime-only screen file, changing the extension to **.cimrt**

Note: Make sure to include in the command line the complete path to the file.

Example

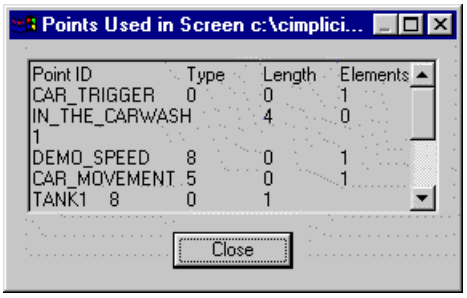
```
cimedit.exe /converttocimrt C:\Proj\scr1.cim C:\Proj\scr2.cim
```

/dumppoints

For: CimEdit only

Loads each file and displays a dialog listing the points used in the file.

The dialog looks like this:



No CimEdit window is displayed, and CimEdit exits when done.

You can also use the Point View tab in the Screen Properties dialog to display the points currently used on a screen. In addition to displaying the list of points, the Point View tab will show you where they are used, and let you edit the Point IDs.

Note: Make sure to include in the command line the complete path to the file.

Example

```
cimedit.exe /dumppoints C:\myproj\myscreen.cim
```

CimEdit/CimView GMMI_SCREENS Libraries

CimEdit/CimView GMMI_SCREENS Libraries

1. First static to the directory of the current screen,
2. Then static to the project's **Screens** directory.

   You cannot change the first search location (static to the directory of the current screen) and, in most cases, you will want to leave the default second location (static to the project's **Screens** directory) intact. However, if you need to change the second location you can by editing the **GMMI_SCREENS** logical name.

Edit GMMI_SCREENS Logical Name

Edit GMMI_SCREENS Logical Name

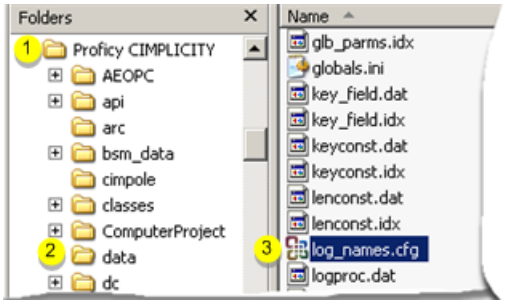
To edit the GMMI_SCREENS Logical name:

Step 1 *(page 647)*	Find the log_names.cfg file.
Step 2 *(page 648)*	Edit the GMMI_SCREENS lines.

Step 1. Edit the GMMI_SCREENS Logical Name

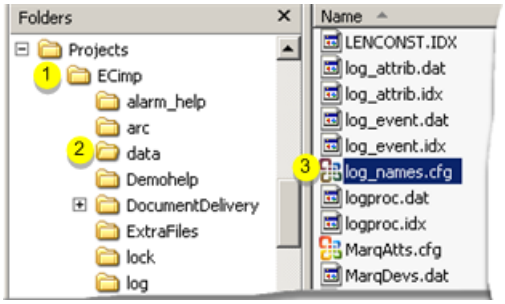
Logical names are configured in the Log_Names.cfg files located in the:

- BSM_Root:\Data



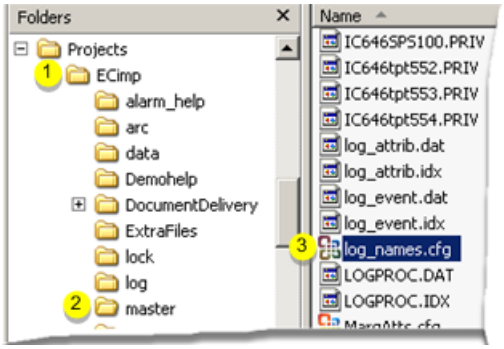
1	CIMPLICITY application root folder.
2	Data subfolder.
3	log_names.cfg file.

- Site_Root:\Data



1	Project (site) root folder
2	Data subfolder.
3	log_names.cfg file.

- Site_Root:\Master



1	Project (site) root folder
2	Master subfolder.
3	log_names.cfg file.

Note: Because a Viewer does not have directories for projects, the only log_names.cfg file is in the Bsm_Root:\Data directory.

Step 2. Edit the GMMI_SCREENS Logical Name

CAUTION: It is recommended that you keep the **GMMI_SCREENS** entries in all the **Log_Names.cfg** files are in sync with each other. Care should be used when editing these files.

When you open either **Log_Names.cfg** file, you will find two lines that refer to the screen location. These entries should be identical.

GMMI_SCREENS|S|default|30|SITE_ROOT:\screens

GMMI_SCREENS|P|default|30|SITE_ROOT:\screens

Where

GMMI_SCREENS is the CimEdit and CimView screens logical name.

SITE_ROOT:\SCREENS is the default directory for those files. This directory can be changed. In fact, the logical path can actually be a list of directories to be searched, separated by semicolons.

| 30 | is the maximum length of the logical value. You need to change this if you change the value of the logical.

See the example below.

The default **GMMI_SCREENS** entry ensures that CimEdit and CimView will always look in the project screens directory.

On a Viewer, **Site_Root** defaults to **Bsm_Root**, so if you create a **Screens** directory in the installation directory of Viewers, they will look there.

Important: Changes will take affect only after all CIMPLICITY processes on the computer you are configuring are entirely shut down and restarted. The best way to ensure this is to reboot.

Example of an Edited GMMI_SCREENS Logical Name

The following example entry for **GMMI_SCREENS** will make CimEdit and CimView look

1. First static to the current directory,
2. Then in the project screens directory (or installation screens directory on viewers),
3. Then in **C:\Screens**, and
4. Finally in **D:\Screens**.

   Note that the length was increased to 60 to account for the extra space.

   GMMI_SCREENS|S|default|60|SITE_ROOT:\screens;C:\SCREENS;D:\SCREENS

   GMMI_SCREENS|P|default|60|SITE_ROOT:\screens;C:\SCREENS;D:\SCREENS

Chapter 2. CimLayout

About CimLayout

CimLayout: provides the means to organize more than one CimView screen into a container that opens all of the screens at the same time; the container is organized according to your specifications.

- CimLayout containers overview.
- CimLayout configuration.

CimLayout Containers Overview

Important: CimLayout provides the capability to open and display several CimView screens at the same time. However, keep in mind your system's capabilities and resources when configuring the layout.

CimLayout provides the following layout selections.

- Screen container.
- Splitter container.
- Tab container.

Screen Container

The screen container:

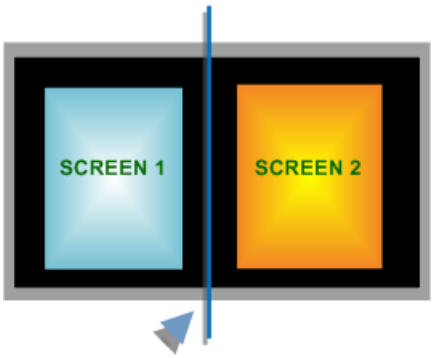
- Consists of one screen only.
- Is normally the last level in a layout section.
- Screen container level selections determine the CimView:
- Screen that will display
- Features, e.g. status bar, Navigation bar and/or toolbar, which will display for the selected screen.

Splitter Container

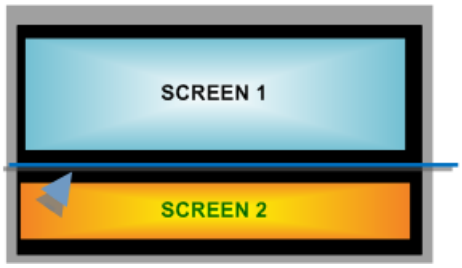
The splitter container:

- Splits the container or selected container section into two or more sections.
- Splitter container level selections determine the
- Location and direction of the splitter bar.
- Sizing of the splitter section.
- Is defined by the splitter bar direction, as follows.

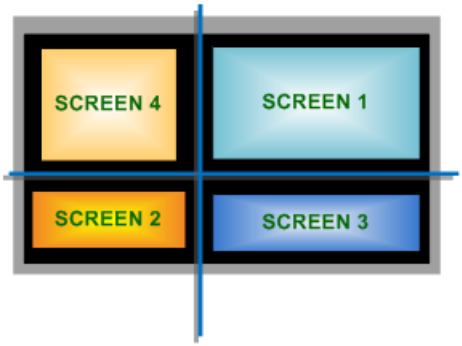
Splitter bar: Vertical



Splitter bar: Horizontal

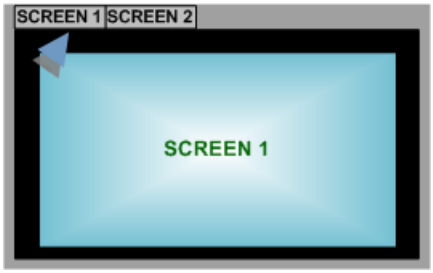


Splitter bar: Vertical and Horizontal



Tab Container

A tab container provides tabs; each tab can display a screen container.



CimLayout Configuration

1 *(page 652)*	Open the CimLayout Configuration Editor
2 *(page 654)*	CimLayout Configuration Editor: components.
3 *(page 661)*	CimLayout: monitor selections.
4 *(page 661)*	CimLayout: container type configuration.
5 *(page 681)*	CimLayout: variables.

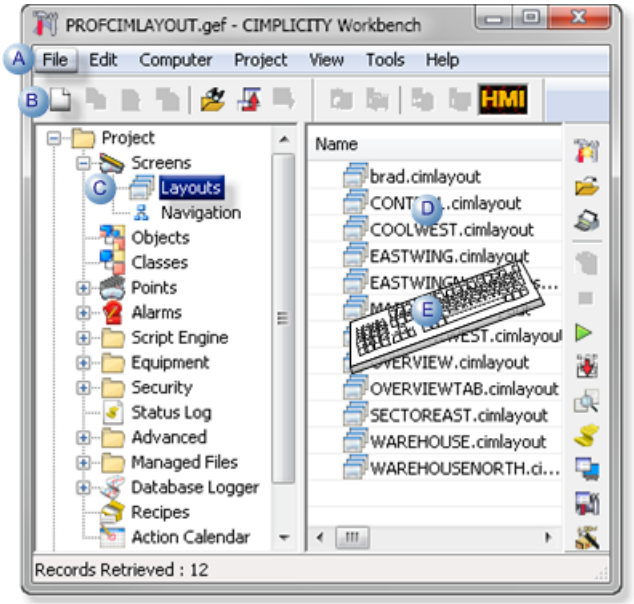
1. Open the CimLayout Configuration Editor

1. Open the CimLayout Configuration Editor

1.1 *(page 652)*	Open a new CimLayout Configuration Editor.
1.2 *(page 653)*	Open an existing CimLayout configuration.

1.1. Open a New CimLayout Configuration Window

1. Select Project>Screens>Layouts in the Workbench left-pane.
2. Do one of the following.

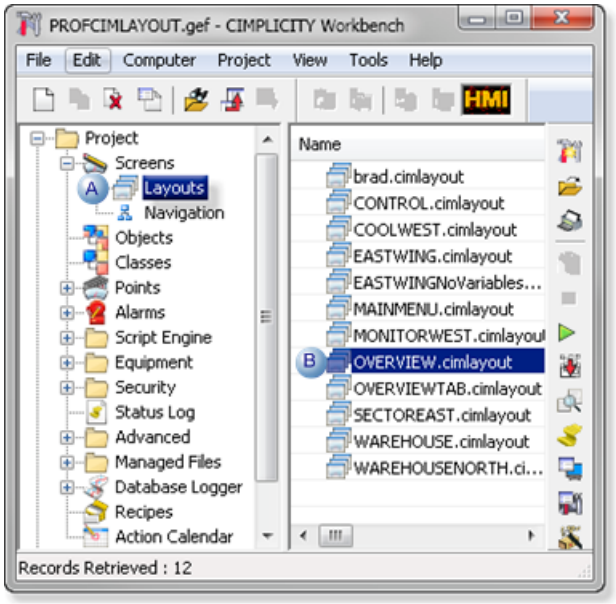


A	Click File>New>Object on the Workbench menu bar.	
B	Click the New Object button on the Workbench toolbar.	
C	In the Workbench left pane:	
	Either	Or
	Double click **Layouts**.	a. Right-click **Layouts**. b. Select New on the Popup menu.
D	a. In the Workbench right pane. a. Right-click anywhere. b. Select New on the Popup menu.	
E	Press Ctrl+N on the keyboard.	

3. Right-click **Layouts**.
4. Select New on the Popup menu.
5. Right-click anywhere.
6. Select New on the Popup menu.

1.2. Open an Existing CimLayout Configuration

1. Select Project>Screens>Layouts in the Workbench left-pane.
2. Do one of the following.



A	In the Workbench left pane.
	a. Make sure an existing CimLayout configuration file is selected in the Workbench right-pane. b. Right-click **Layouts** (Workbench left-pane). c. Select Edit on the Popup menu.
B	a. In the Workbench right pane.
	a. Right-click an existing CimLayout configuration file. b. Select Edit on the Popup menu.

3. Make sure an existing CimLayout configuration file is selected in the Workbench right-pane.
4. Right-click **Layouts** (Workbench left-pane).
5. Select Edit on the Popup menu.
6. Right-click an existing CimLayout configuration file.
7. Select Edit on the Popup menu.

2. CimLayout Configuration Editor: Components

2. CimLayout Configuration Editor: Components

The following tools are available in the CIMPLICITY (Navigation) Configuration Editor.

2.1 *(page 655)*	CimLayout: Toolbar
2.2 *(page 655)*	CimLayout: Menu
2.3 *(page 659)*	CimLayout: Left pane.
2.4 *(page 660)*	CimLayout: Right pane.

2.1 CimLayout Configuration Editor: Toolbar

	Description	When Enabled
1	Create a new configuration file.	Always.
2	Open an existing configuration file.	
3	Reload the configuration file from the disk.	After the configuration file is saved for the first time.
4	Save the configuration file.	Always
5	Cut the selected entity	When a configuration object is selected.
6	Copy a selected entity or value.	When a configuration object is selected.
7	Paste a cut/copied entity or value.	When a layout section, e.g. Left/Top, that does not have a configuration object is selected.
8	Undo Last Action	When an action has just been taken, e.g. a screen was added to the layout configuration; undo will revert to the configuration just before the screen was added.
9	Open CimLayout documentation.	Always.

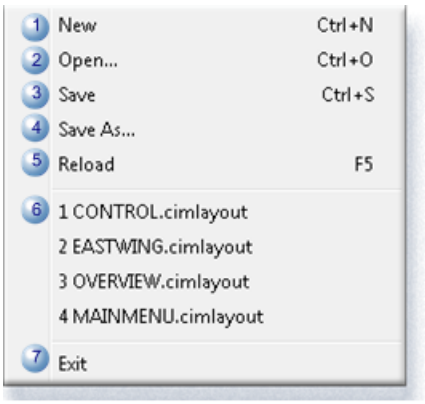
2.2 CimLayout Configuration Editor: Menus

The CIMPLICITY CimLayout Configuration Editor menu bar includes the following features.

rect 6, 20, 27, 43 *(page 656)*
rect 39, 20, 60, 43 *(page 657)*
rect 98, 20, 119, 43 *(page 657)*
rect 163, 20, 184, 43 *(page 658)*
rect 206, 20, 227, 43 *(page 658)*

A *(page 656)*	File
B *(page 657)*	Edit
C *(page 657)*	Configuration
D *(page 658)*	View
E *(page 658)*	Help

1. File menu

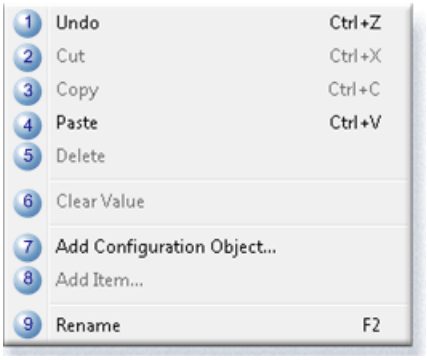


	Item	Description
1	New	Creates a new configuration file.
2	Open	Opens a Windows Open browser to find and open an existing configuration file.
3	Save	Saves the current configuration file

4	Save As	Save the current configuration file with the name and location entered in the Windows Save As browser.
5	Reload	Refreshes the CIMPLICITY Configuration Editor display.
6	Recent files	Opens a selected configuration file that was recently opened in the CIMPLICITY (CimLayout) Configuration Editor.
7	Exit	Closes the CIMPLICITY Configuration Editor.

1. Edit/Popup menus

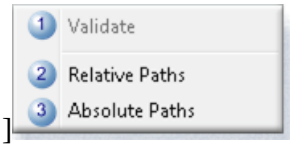
A Popup menu can be opened (right-click) that is the same as the Edit menu when an entity is selected but is not in edit mode.



	Item	Description
1	Undo	When an action has just been taken, e.g. a screen was added to the CimLayout configuration; undo will revert to the configuration just before the screen was added.
2	Cut	Cuts a selected entity, e.g. configuration object; the entity can be pasted in a selected location.
3	Copy	Copies a selected entity, e.g. a configuration object; the entity can be pasted in a selected location
4	Paste	Pastes a cut or copied entity in a selected location. A number is appended to the name.
5	Delete	Deletes the selected entity.
6	Clear Value	Reverts the current selection to a non-assigned state (**Not set**).5
7	Add Configuration Object	Opens an Add Configuration Object dialog box to add a new entity, e.g. a split screen, to the tree.
8	Add Item	Opens an Add dialog box to add a new, such as a CimLayout variable.
9	Rename	Makes a selected entity available for editing.

1. Configuration menu

Features can be accessed when Navigation Configuration is selected in the CIMPLICITY (Navigation) Configuration Editor left pane.

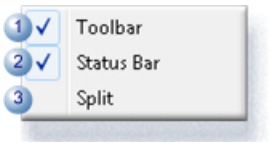


]

	Item	Description
1	Validate	Check for errors in the CimLayout configuration.
2	Relative Paths	Assigns screen paths as relative paths, which are dependent on the file, not on the project.
3	Absolute Paths	Assigns screen paths as absolute paths. If the menu is used on other computers the paths to the screens must be exactly the same as specified in the configuration file.
4	Emit Translation	Emits words to a selected translation file.
5	Edit Translation Text	Opens the selected translation file.

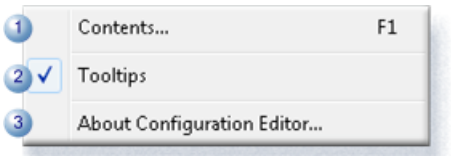
Note: The relative Paths and Absolute Paths menu items are enabled after the first time the Navigation Configuration file is saved.

1. View menu



	Item	Description	
1	Toolbar	Select to:	
		Check	Display the toolbar.
		Clear	Hide the toolbar.
2	Status Bar	Select to:	
		Check	Display the status bar.
		Clear	Hide the status bar.
3	Split	Enables resizing the panes using the arrow keys on the keyboard.	

1. Help menu



	Item	Description	
1	Contents	Opens the CimLayout configuration topic in the CIMPLICITY documentation.	
2	Tool tips	Select to:	
		Check	Display tool tips
		Clear	Hide tool tips.
3	About Configuration Editor.	Opens details about the Proficy CIMPLICITY version and build.	

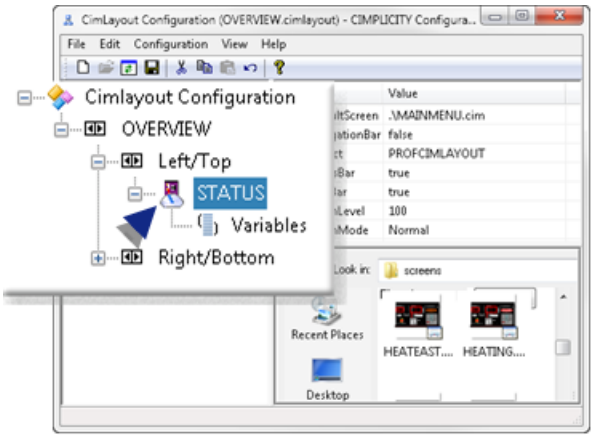
2.3. CimLayout Configuration Editor: Left Pane

The tree in the CimLayout Configuration Editor left pane lists the Container layout sections. The tree components that display depend on the selected container type.

- Left Pane: Screen container
- Left Pane: Splitter container`
- Left Pane: Tab container

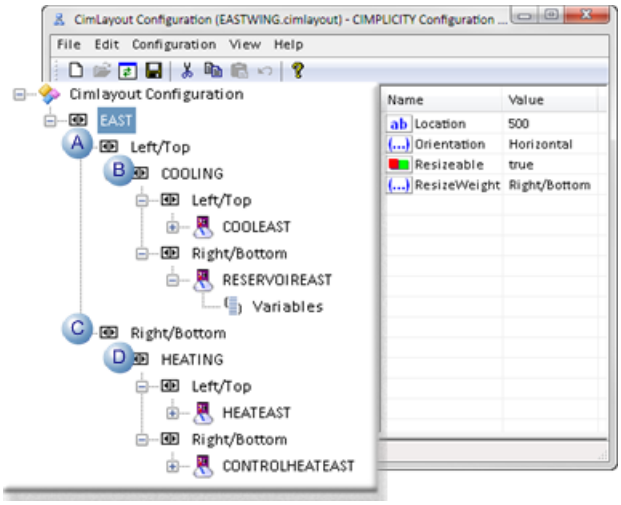
Left Pane: Screen Container

A screen container identifying label and Variables node display in the left-pane tree.; the screen container is the last container entry in a splitter container tree section.



Left Pane: Splitter Container

A splitter container displays a tree of splits (Left/Top and Right/Bottom) in the left pane.

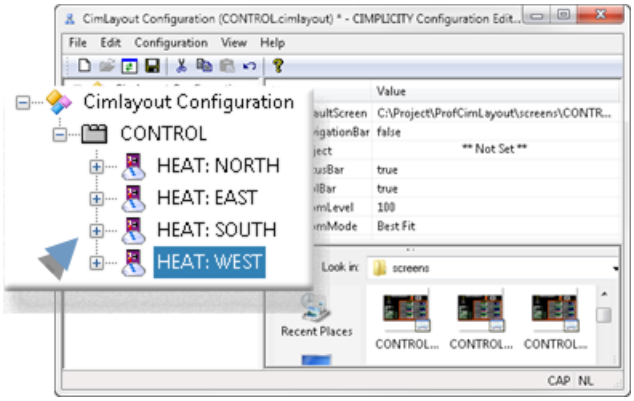


A	Left/Top
B	Left/Top identifier.
C	Right/Bottom
D	Right/Bottom identifier.

Note: A screen container is the last entry in each tree section.

Left Pane: Tab Container

A tab container left-pane lists the tabs that will be included in the CimLayout container.

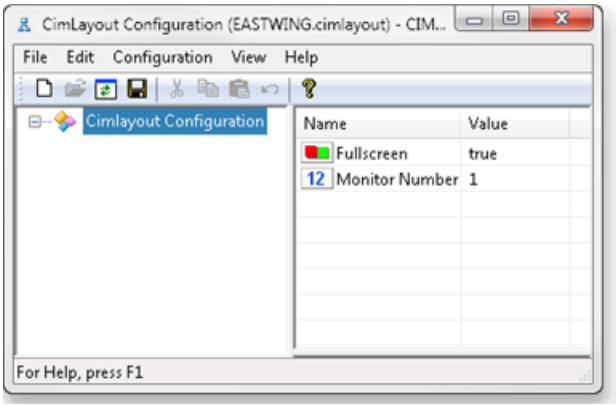


2.4. CimLayout Configuration Editor: Right Pane

The CIMPLICITY CimLayout Configuration Editor right pane provides the tools to define properties for the container layout *(page 661)* and section details.

3. CimLayout: Monitor Selections

A CimLayout container can be positioned



Field	Description	
Fullscreen	Select true or false to display the CimLayout container as a full or partial screen application.	
	True	Full screen application.
	False	Partial screen application.
Monitor Number	The number that the monitor is assigned in your graphic card or Windows Setup dialog box.	

4. CimLayout: Container Type Configuration

4. CimLayout: Container Type Configuration

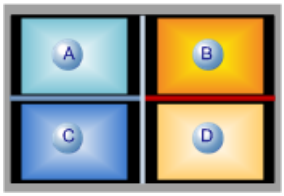
4.1 (page 661)	CimLayout: Screen container.
4.2 (page 665)	CimLayout: Splitter container.
4.3 (page 678)	CimLayout: Tab container.

4.1. CimLayout: Screen Container Type

A CimLayout screen container type consists of one screen only. It is used as the last container for each CimLayout section.

Example: Screen Containers

A CimLayout container is divided into four sections; each section holds a screen container.



A	B	C	D	Screen container.

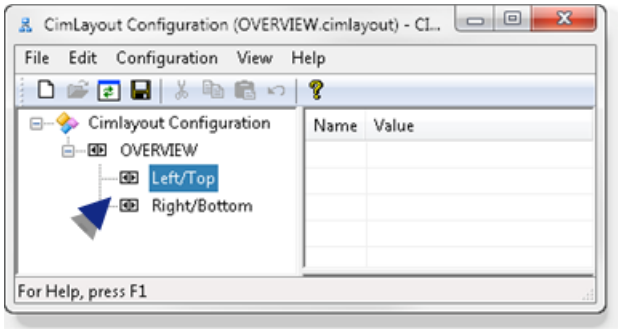
1. Right-click a location node that does not include a screen in the CimLayout Configuration Editor left-pane.

   Location nodes are:

Location	Description
Left/Top	Left or top side of the CimLayout section.
Right/Bottom	Right of bottom side of the CimLayout section.

   Note:

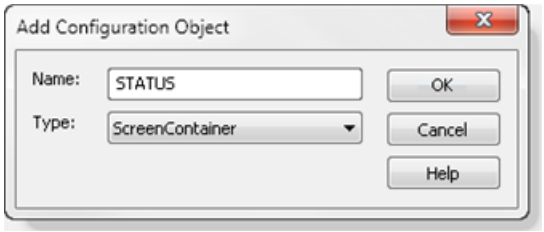
   - The actual screen location, e.g. left or top, is determined by the selected orientation.
   - There is no reason to use a ScreenContainer type at the top level; it almost always will be used at the last level.



2. Do either of the following.
   - Click Edit>Add Configuration Object on the CimLayout Configuration Editor menu bar.
   - Right-click the selection; select Add Configuration Object on the Popup menu.

   An Add Configuration Object dialog box opens.

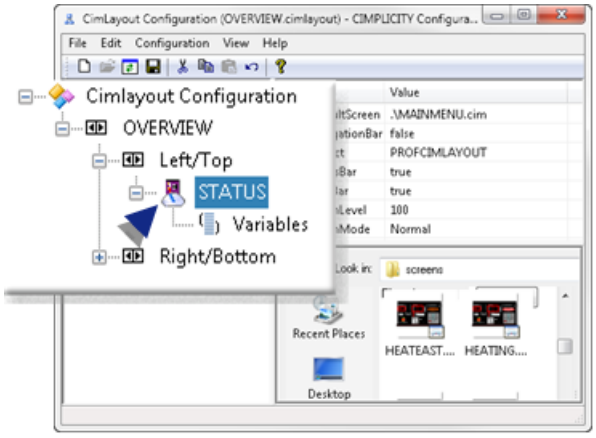
3. Enter and select the following.



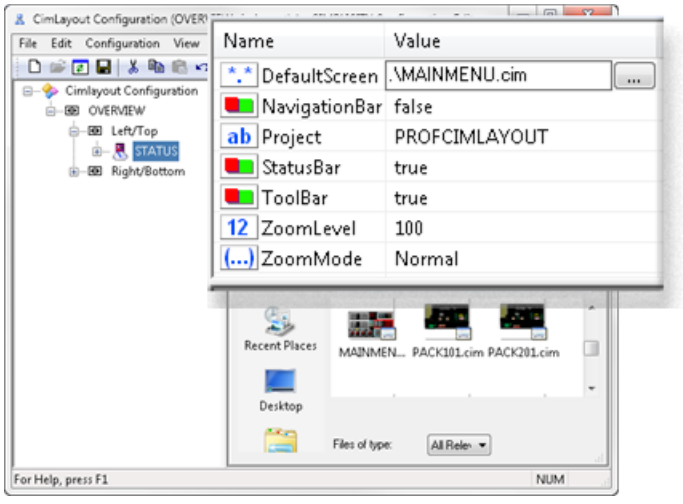
Field	Description
Name	Identifies the selected container.
Type	ScreenContainer

4. Click OK.

   The named configuration object is added to the CimLayout container; no additional objects can be added to this screen container object's section of the layout.



5. Configure the object in the CimLayout Configuration Editor right-pane, as follows.



Field	Description	
DefaultScreen	Screen that will display in the selected location. Note: Do any of the following to select the screen. • Enter a name and absolute or relative path in the DefaultScreen field. • Click the button to the right of the DefaultScreen field; find and select the screen that will be used. • Drag a screen from the CimLayout Configuration Editor lower right-pane either to the: • Configuration object in the left-pane. • DefaultScreen field in the configuration object's right-pane.	
NavigationBar	Choose either of the following to display or hide the project's selected Navigation bar for the selected screen.	
	true	Display the Navigation bar.
	false	Hide the Navigation bar.
Project	Project name to fully qualify unqualified *(page 220)* point ID's for the selected screen.	
Status bar	Choose either of the following to display or hide the selected screen's status bar.	
	true	Display the status bar.
	false	Hide the status bar.
ToolBar	Choose either of the following to display or hide the selected screen's toolbar.	
	true	Display the toolbar.
	false	Hide the toolbar.
ZoomLevel	Percent of the actual screen size that the screen will display when the CimLayout container opens. **Note:** The zoom level is particularly helpful if screens that are different sizes will be included in the layout. Example: Two Screens/One Zoomed Two screens, an Overview screen and an Alarm Viewer, are included in a CimLayout configuration. The Overview screen is 80% as wide as the Alarm Viewer. The Alarm Viewer is set to 80, with a normal ZoomMode. When the CimLayout configuration is opened, the two screens are the exact same width. **Note:** The screen size display if the container size changes depends on other selections for the container.	
ZoomMode	Select one of the following	
	BestFit	Screen is automatically resized to fit the section of the container allocated to the selected screen.

Field	Description	
	Normal	Screen displays in its real selected zoom level size.

4.2. CimLayout: Splitter Container

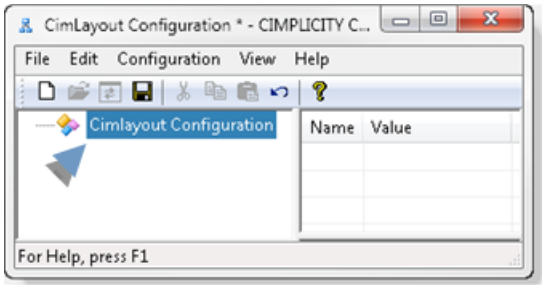
The CimLayout Splitter Container is the container that enables you to divide the CimLayout configuration into parts. Selecting the splitter container begins at the top level.

- Splitter Container: Top Level Split
- Splitter Container: Multiple Splits

Splitter Container: Top Level Split

When the splitter container is selected at the top level, the CimLayout configuration will be split into two parts. You define details, such as the position, size and orientation of these parts.

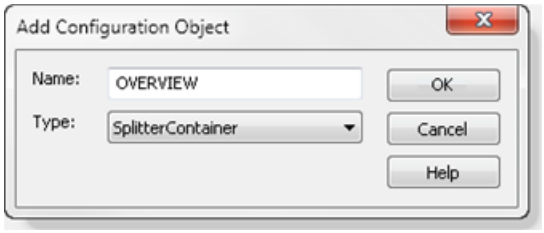
1. Select CimLayout Configuration in the CimLayout Configuration Editor.



2. Do either of the following.
   - Click Edit>Add Configuration Object on the CimLayout Configuration Editor menu bar.
   - Right-click the selection; select Add Configuration Object on the Popup menu.

   An Add Configuration Object dialog box opens.

3. Enter and select the following.



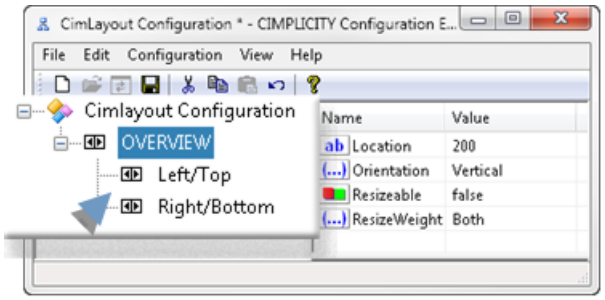
Field	Description
Name	Identifies the selected container.

Field	Description
Type	SplitterContainer

4. Click OK.

   Two location nodes are attached to the layout section; either node can also be split or can be configured as a screen container..

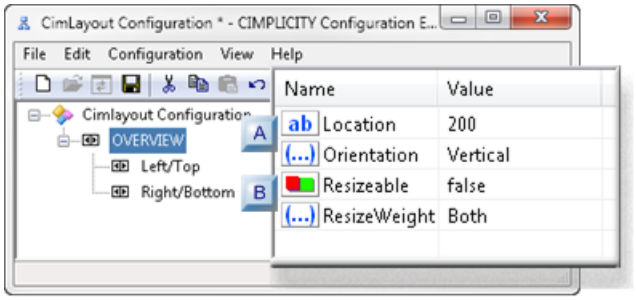
   Location options are as follows.



Location	Description
Left/Top	Left or top side of the CimLayout section.
Right/Bottom	Right of bottom side of the CimLayout section.

Note: The actual screen location (e.g. left or top) is determined by the selected orientation.

5. Configure the object in the CimLayout Configuration Editor right-pane, as follows.



rect 147, 106, 171, 130 *(page 668)*
rect 147, 66, 171, 90 *(page 667)*

A *(page 667)*	Location/Orientation
B *(page 668)*	Resizable/ResizeWeight

	Splitter bar container: Examples for two sections.

A	Location/Orientation

The Location and Orientation refer to the splitter bar that divides the CimLayout container.

Location

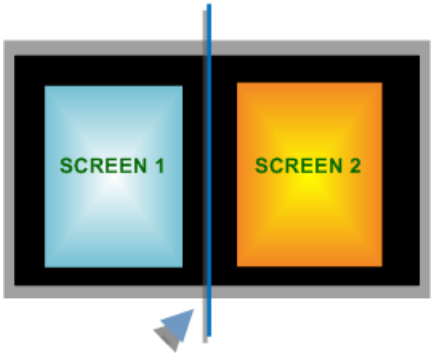
The value entered in the **Location** field specifies the distance the splitter bar will be from the screen edge.

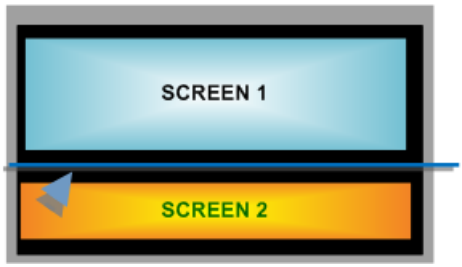
Important: The side the distance is measured from depends on the splitter bar's selected orientation and resize weight.

Bar Orientation	Resize/Weight	Bar Distance is from:
Vertical	Left/Top	Right
	Right/Bottom	Left
	Both	Left and Right
Horizontal	Left/Top	Bottom
	Right/Bottom	Top
	Both	Top and Bottom

Orientation

The splitter bar orientation is vertical or horizontal.

Vertical	The CimLayout configuration is split vertically. 

Horizontal	The CimLayout configuration is split horizontally. 

B	Resizable/ResizeWeight

One or both sides of the CimLayout container can be resized, as follows.

Resizable

The screen, in the selected **ResizeWeight** location is resized or not, as follows.

true	Is resized when the CimLayout container is resized; it will automatically zoom up or down, as required.
false	Is not resized when the CimLayout container is resized; scroll bars will display if the allocated size in the container is smaller than the screen.

Note: The container side that is resizable is affected by the splitter bar orientation.

ResizeWeight

If resizable is set to True, the container side that users can resize during runtime is affected by the splitter orientation.

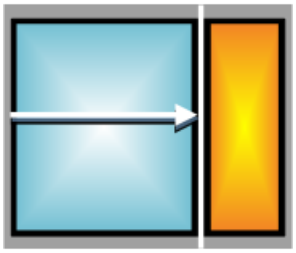
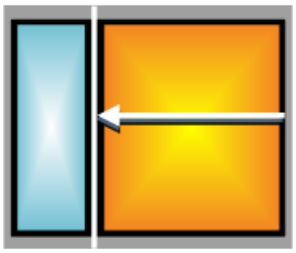
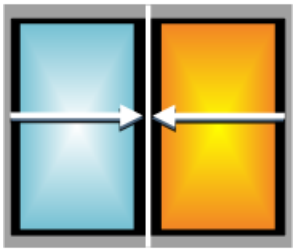
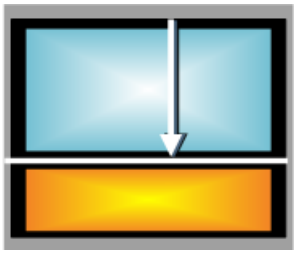
Section Location	Splitter Bar Orientation	Resizable Side
Left/Top	Vertical	Left
	Horizontal	Top
Right/Bottom	Vertical	Right
	Horizontal	Bottom
Both	Vertical	Left and Right
	Horizontal	Top and Bottom

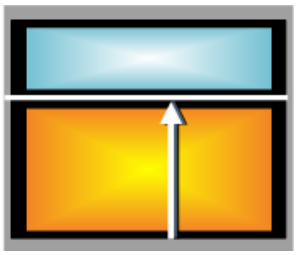
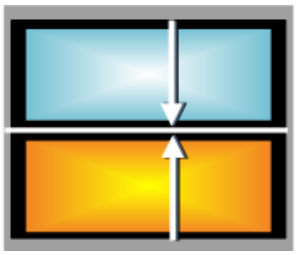
Result: CimLayout is split in two based on your specifications.

Splitter Container: Examples for Two Sections

- A CimLayout container includes one Splitter container that holds with two screen containers.
- The splitter bar distance from the side is 175.
- Where the distance is measured is based:

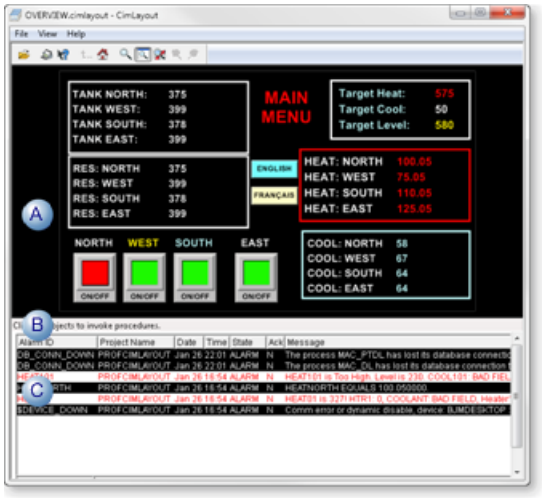
- On the splitter bar orientation
- If one or both screen containers can be resized during runtime.

Vertical Splitter Bar Orientation	
Resize/Weight is:	Screen Containers
Left/Top	
Right/Bottom	
Both	
Horizontal Splitter Bar Orientation	
Resize/Weight is:	Screen Containers
Left/Top	

Vertical Splitter Bar Orientation	
Right/Bottom	
Both	

Example: Runtime

During runtime a CimLayout container displays the two screens based on the Splitter container configuration.

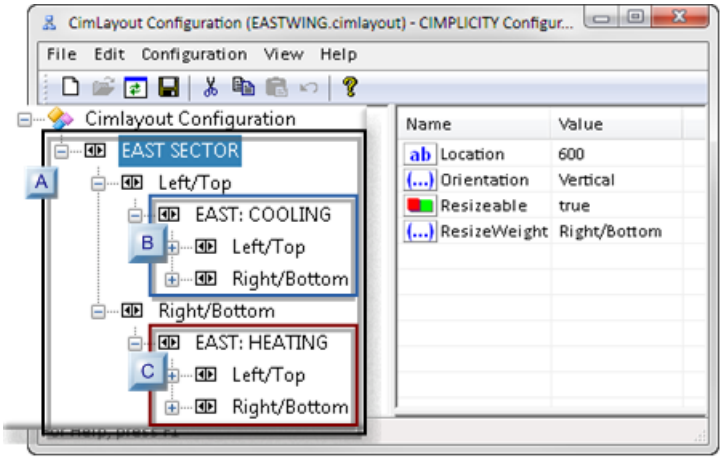


	Item	Description
A	Screen container	Left/Top
B	Splitter bar	Horizontal
C	Screen container	Right/Bottom

Splitter Container: Multiple Splits

A CimLayout container can be divided into more than two sections by adding a splitter container object to a splitter container object. Each new split is configured within the boundaries set by its parent split.

CAUTION: Make sure that your system resources can support the number of splits and the screens that are selected for the CimLayout container.



rect 79, 217, 103, 242 *(page 674)*
rect 77, 138, 101, 163 *(page 671)*
rect 11, 99, 35, 124 *(page 671)*

	Split Level	Contains Splits	Example
A *(page 671)*	Top level	Left/Top and Right/Bottom	EAST: SECTOR
B *(page 671)*	Left/Top	Left/Top and Right/Bottom	EAST: COOLING
C *(page 674)*	Right/Bottom	Left/Top and Right/Bottom	EAST: HEATING
	Example	Runtime four screen containers.	

A	Top level split: Left/Top and Right/Bottom

Because the top level split divides the screen into the two basic sections, the values entered at the top level define the sections in which additional splitting will be performed.

B	Left/Top split: Left/Top and Right/Bottom

One or both splits in the CimLayout container can be split again.

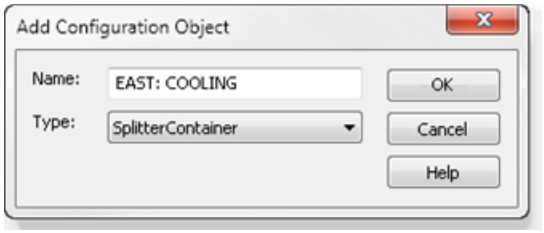
Note: This procedure description begins with splitting left/top; either can actually be split first.

6. Right-click **Left/top** in the CimLayout Configuration Editor left-pane.

7. Do either of the following.
   - Click Edit>Add Configuration Object on the CimLayout Configuration Editor menu bar.
   - Right-click the selection; select Add Configuration Object on the Popup menu.

   An Add Configuration Object dialog box opens.

8. Enter the following.



Field	Description
Name	Identifies the selected container.
Type	SplitterContainer

   A new (Left/Top) <NAME> splitter container now has its own Left/Top and Right/Bottom split.

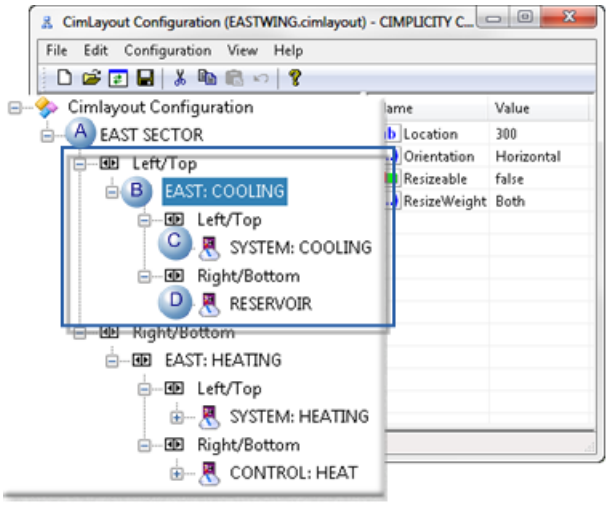
9. Configure the (Left/Top) <NAME> container layout in context of where it is located and the allocated space that it has been assigned by the parent level split.

guide: Guidelines

- The child section's splitter bar location must be within the allocated location/orientation.

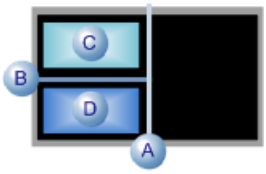
Note: The exact orientation and location will depend on how you want the split within the split to display.

- The child split that is being configured cannot be resized if either of the following has been configured as the parent level.
- Resizable is disabled for the CimLayout container.
- Resize/Weight is assigned to the other child split (e.g. Right/Bottom).

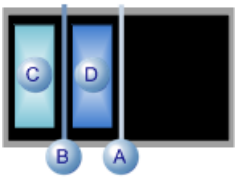


Examples: Left/Top (EAST: COOLING) Split Section in a CimLayout Container

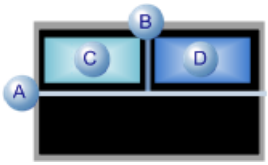
Some examples of screen container positions in a (Left/Top) <NAME> split container, before the (Right/Bottom) <NAME> split is configured, are as follows.



	Item	Example	Orientation/Position	Location/Height
A	Parent Splitter bar	EAST: SECTOR	Vertical	600
B	Left/Top splitter bar	EAST: COOLING	Horizontal	300
C	Screen container	SYSTEM: COOLING	Top	150
D	Screen container	RESERVOIR	Bottom	150



	Item	Example	Orientation/Position	Location/Height
A	Parent Splitter bar	EAST: SECTOR	Vertical	600
B	Left/Top splitter bar	EAST: COOLING	Vertical	300
C	Screen container	SYSTEM: COOLING	Left	150
D	Screen container	RESERVOIR	Right	150



	Item	Example	Orientation/Position	Location/Height
A	Parent Splitter bar	EAST: SECTOR	Horizontal	600
B	Left/Top splitter bar	EAST: COOLING	Vertical	300
C	Screen container	SYSTEM: COOLING	Left	150
D	Screen container	RESERVOIR	Right	150

	Item	Example	Orientation/Position	Location/Height
A	Parent Splitter bar	EAST: SECTOR	Horizontal	600
B	Left/Top splitter bar	EAST: COOLING	Horizontal	300
C	Screen container	SYSTEM: COOLING	Top	150
D	Screen container	RESERVOIR	Bottom	150

C	Right/Bottom split: Left/Top and Right/Bottom

Creating another child split in the CimLayout container so the container will have four sections is performed using the same procedure as creating the first child split.

10. Right-click Right/Bottom.
11. Select CimLayout Configuration in the CimLayout Configuration Editor.

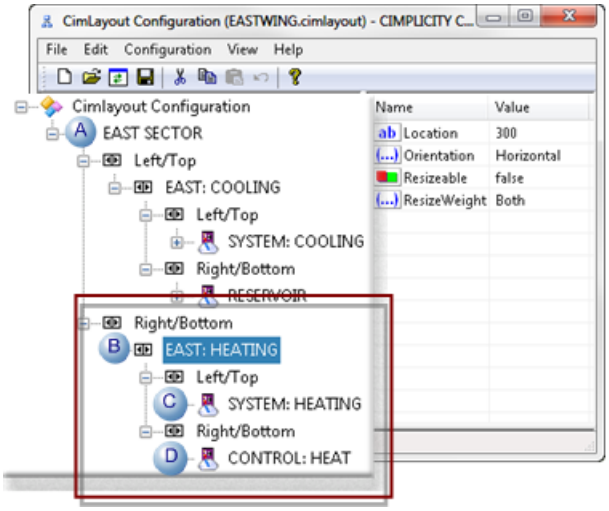
    An Add Configuration Object dialog box opens.

12. Enter the following.

Field	Description
Name	Identifies the selected container.
Type	SplitterContainer

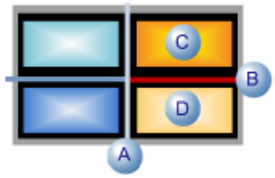
13. Configure the (Left/Top) <NAME> container layout in context of where it is located and the allocated space that it has been assigned by the parent level split.

The Bottom/Right section location is configured within the section of the CimLayout area that is identified by its parent split levels location and orientation.

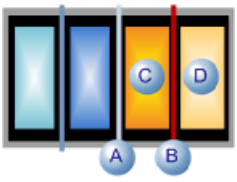


Examples: Right/Bottom (EAST: HEATING) Split Section in a CimLayout Container

Some examples of screen container positions in a (Right/Bottom) <NAME> split container, after the (Left/Top) <NAME> split is configured, are as follows.

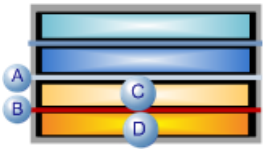


	Item	Example	Orientation/Position	Location/Height
A	Parent Splitter bar	EAST: SECTOR	Vertical	600
B	Right/Bottom splitter bar	EAST: HEATING	Horizontal	300
C	Screen container	SYSTEM: HEATING	Top	150
D	Screen container	CONTROL: HEAT	Bottom	150

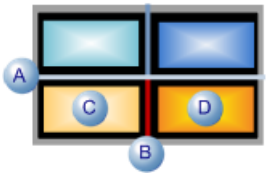


	Item	Example	Orientation/Position	Location/Height
A	Parent Splitter bar	EAST: SECTOR	Vertical	600
B	Right/Bottom splitter bar	EAST: HEATING	Vertical	300

C	Screen container	SYSTEM: HEATING	Left	150
D	Screen container	CONTROL: HEAT	Right	150



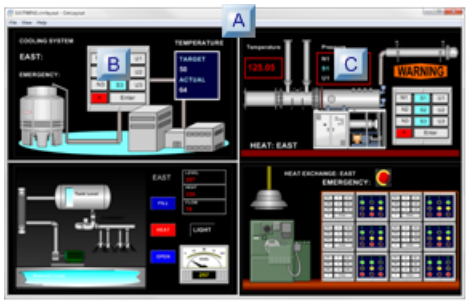
	Item	Example	Orientation/Position	Location/Height
A	Parent Splitter bar	EAST: SECTOR	Horizontal	600
B	Right/Bottom splitter bar	EAST: HEATING	Horizontal	300
C	Screen container	SYSTEM: HEATING	Top	150
D	Screen container	CONTROL: HEAT	Bottom	150



	Item	Example	Orientation/Position	Location/Height
A	Parent Splitter bar	EAST: SECTOR	Horizontal	600
B	Right/Bottom splitter bar	EAST: HEATING	Vertical	300
C	Screen container	SYSTEM: HEATING	Left	150
D	Screen container	CONTROL: HEAT	Right	150

Example: Runtime Four Screen Containers

During runtime a CimLayout container displays the four screens based on the Splitter containers' configuration.



A *(page 677)*	Top Parent container: EAST: SECTOR.

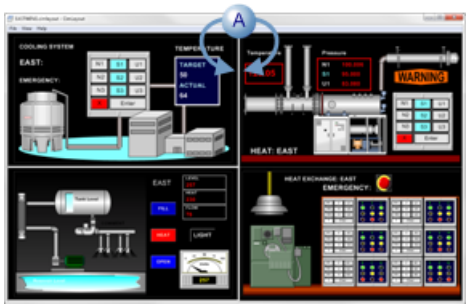
B *(page 677)*	Left/Top splitter container: EAST: COOLING.
C *(page 678)*	Right/Bottom splitter container: EAST: HEATING.

A	Top Parent Container: EAST: SECTOR

The top level parent CimLayout container (EAST: SECTOR) includes child splitter containers

The CimLayout container's splitter:

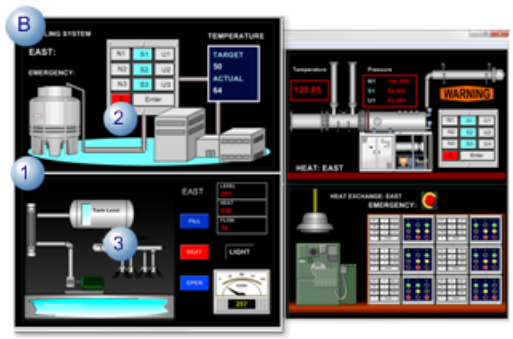
- Bar is vertical.
- Containers' positions are Left and Right.



B	Left/Top Splitter Container: EAST: COOLING

The parent CimLayout (EAST: SECTOR) container's Left/Top container includes its own child splitter container.

EAST: COOLING displays the following in the left side of the EAST: SECTOR container.

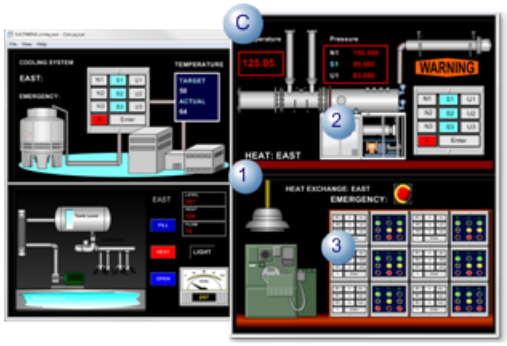


	Item	Example	Orientation/Position	Location/Height
1	Splitter bar	EAST: COOLING	Horizontal	300
2	Screen container	SYSTEM: COOLING	Top	150
3	Screen container	RESERVOIR	Bottom	150

C	Right/Bottom Splitter Container: EAST: HEATING

The Parent CimLayout (EAST: SECTOR) container's Right/Bottom container includes its own child splitter container.

EAST: HEATING displays the following in the right side of the EAST: SECTOR container.



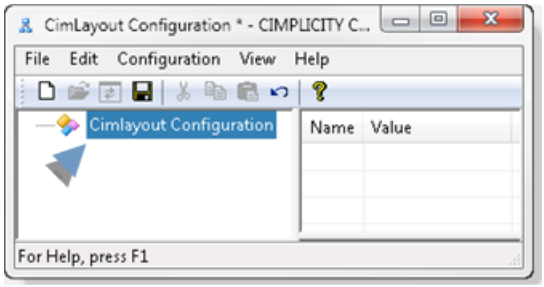
	Item	Example	Orientation/Position	Location/Height
1	Splitter bar	EAST: HEATING	Horizontal	300
2	Screen container	SYSTEM: HEATING	Top	150
3	Screen container	CONTROL: HEAT	Bottom	150

4.3. CimLayout: Tab Container

The CimLayout tab container creates tabs, each of which can be configured the same way a CimLayout configuration with one tab only is configured.

CAUTION: Make sure that your system resources can support the number of tabs, splits and the screens that are selected for the CimLayout container.

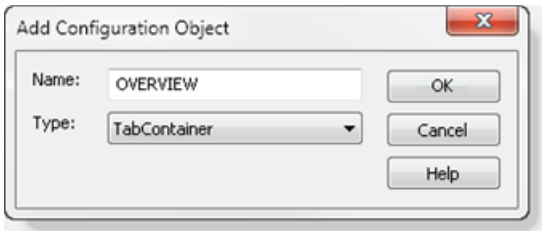
1. Right-click CimLayout Configuration in the CimLayout Configuration Editor.



2. Do either of the following.
   - Click Edit>Add Configuration Object on the CimLayout Configuration Editor menu bar.
   - Right-click the selection; select Add Configuration Object on the Popup menu.

An Add Configuration Object dialog box opens.

3. Enter and select the following.

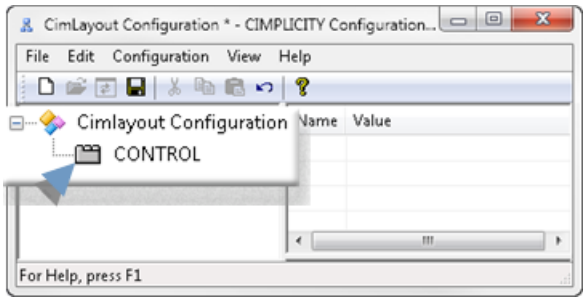


Field	Description
Name	Identifies the selected container.
Type	TabContainer

4. Click OK.

A tab configuration object is added to the CimLayout Configuration Editor left-pane.

Note: Each Configuration object that is added to the tab container object will display on its own tab in the CimLayout container.



5. Select the tab container object.

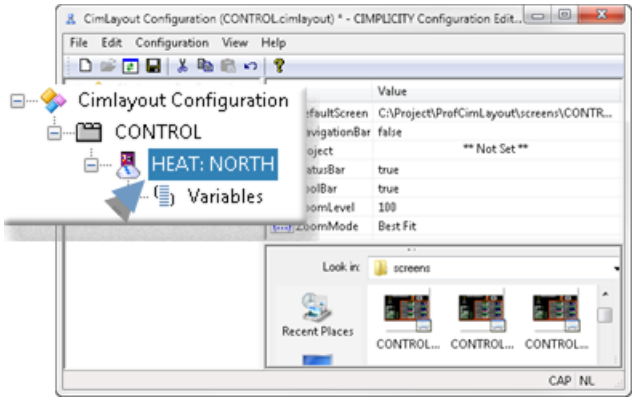
6. Do either of the following.
   - Click Edit>Add Configuration Object on the CimLayout Configuration Editor menu bar.
   - Right-click the selection; select Add Configuration Object on the Popup menu.

An Add Configuration Object dialog box opens.

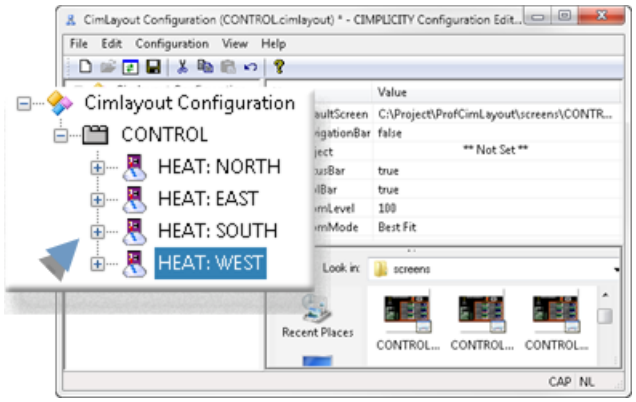
7. Add and configure the object type that will define the first tab.

Note: Any one of the containers, Screen *(page 661)* , Splitter *(page 665)* or Tab container can be added.

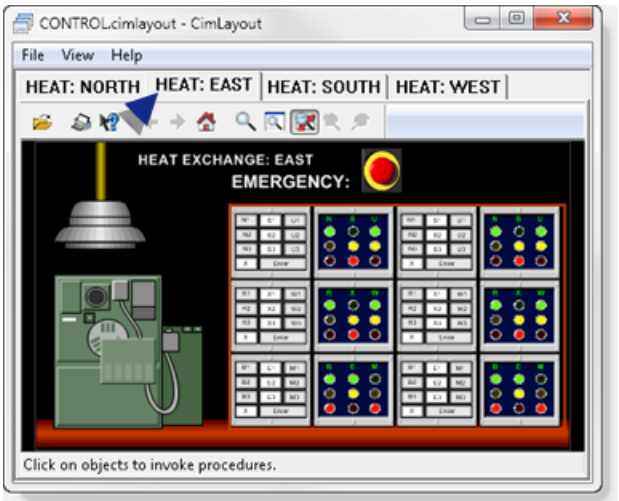
CAUTION: Make sure that your system resources can support the number of containers and the screens that are selected for the CimLayout container.



8. Select the tab container object again to add and configure one or more additional tabs.



During runtime the tabs are available for selection in the CimLayout container.

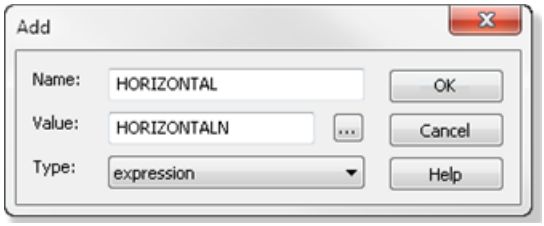


5. CimLayout: Variables

1. Select Variables in the left pane for the selected screen container.
2. Do one of the following.
   - Click Edit>Add Item on the CimLayout Configuration Editor menu bar.
   - Right-click Variables; select Add Item on the Popup menu.
   - Right-click the right-pane; select Add Item on the Popup menu.

   An Add dialog box opens.

   a. Enter the following.



Field	Description	
Name	Unique name.	
Value	A variable value. **Note:** Click the Expression Editor button to enter the value in the Expression Editor.	
Type	expression	A variable is an expression type.

Note: :

- If you have assigned a value to a selected screen and that value is different from the value in the CimLayout Configuration Editor, the following will occur based on your configuration.

The Screen	The value assigned at the:
Does not inherit values	Screen level will display.
Does inherit values	Navigation Configuration Editor level will display.

- If a child screen can inherit variables from the parent, all of the variable values that are listed in the Navigation Configuration Editor will be applied to all of the same variables on the screen. It is important to make sure that variables that should not be inherited are not listed with variables that should be inherited.

Note: If the value on the screen must be unique, it is safer to make the variable name unique to that screen. That way there will never be a chance that the value will be inherited.

- Duplicate variable names are not allowed.

Chapter 3. Navigation Configuration

About Navigation Configuration

- Navigation bar: Overview.
- Navigation bar: Configuration and use.

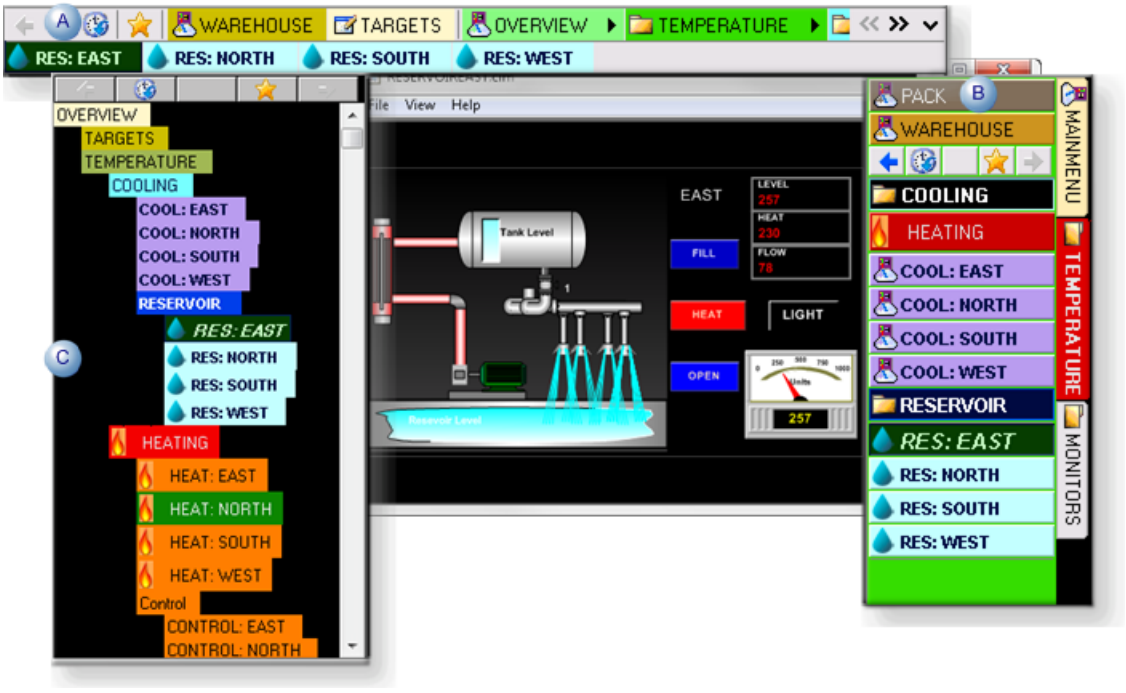
Navigation Bar: Overview

The Navigation Configuration Editor provides the tools to create CimView menus that runtime operators can use to easily find, select and open any screen or run any script to which they have access.

When selected to display, a configured Navigation file displays the Navigation bar components for CimView screens that are assigned to the navigation configuration.

A Navigation bar can display any of three styles, in different locations *(page 701)* .

Available Navigation bar styles are:



A	Standard style
B	Cascade style

C	Tree style

Navigation Bar: Configuration and Use

1 *(page 684)*	Open the CIMPLICITY Navigation Configuration Editor.
2 *(page 690)*	Navigation Configuration Editor: Components.
3 *(page 698)*	Navigation: Global properties.
4 *(page 717)*	Add: Configuration objects.
5 *(page 728)*	Navigation: Child Properties and Variables
6 *(page 732)*	Configuration object: Properties.
7 *(page 758)*	Navigation: Runtime tools.

1. Open the CIMPLICITY Navigation Configuration Editor

1. Open the CIMPLICITY Navigation Configuration Editor

The (*.navbar) file that is created using the CIMPLICITY Navigation Configuration Editor contains the configuration for the Navigation bar.

When a configuration file is entered in the **Default navigation bar** field its configuration will be available to display for the screens that are available in the configuration during a runtime session.

Do one of the following on the General tab in the Global Configuration dialog box.

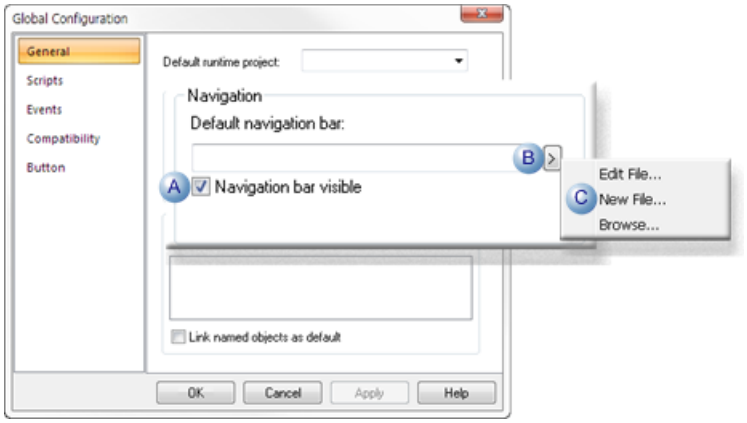
1.1 *(page 685)*	Create a new Navigation Configuration file.
1.2 *(page 688)*	Open an existing Navigation Configuration file.

1.1. Create a New Navigation Configuration File

- New Navigation file: Global Configuration dialog box.
- New Navigation file: Workbench.

New Navigation File: Global Configuration dialog box.

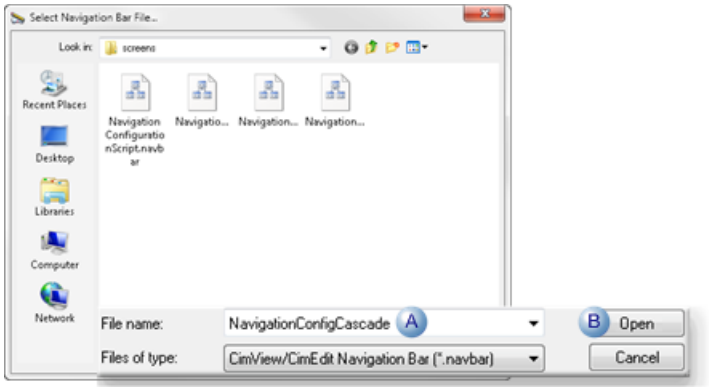
1. Select **General** in the Global Configuration dialog box
2. Do the following.



A	(Optional) Check Navigation bar visible to cause the Navigation bar to display during runtime.
B	Click the Popup menu button to the right of the **Default navigation bar** field.
C	Select New File on the Popup menu.

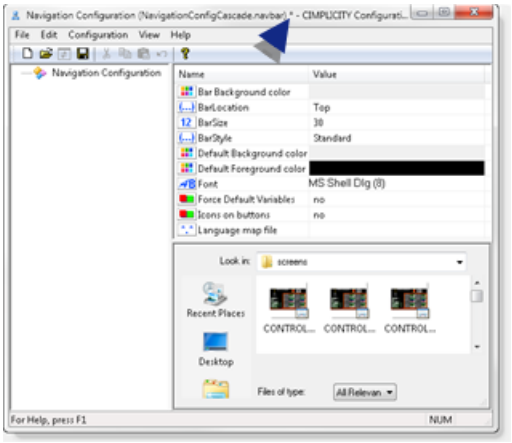
A Select Navigation Bar File browser opens.

3. Find the folder where the new navigation file will be located.
4. Do the following.



A	Enter a name for the file in the **File name** field. **Note:** The configuration file extension is .navbar.
B	Click Open.

Result: The Navigation Configuration window opens, displaying a navigation file and its assigned name.

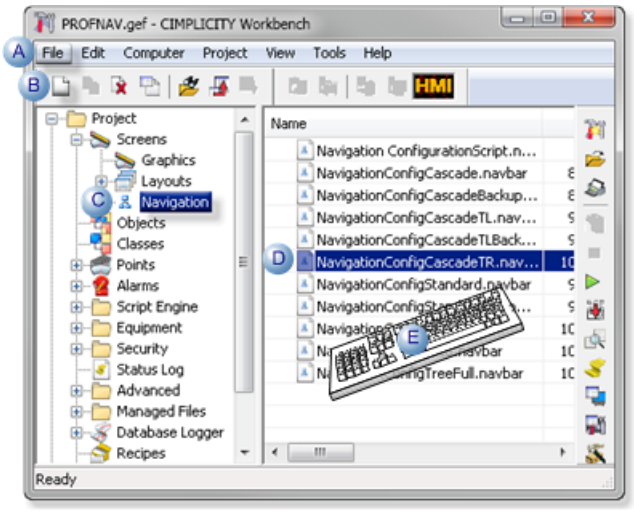


5. Do any of the following to save the file.
   - Click File>Save on the Navigation Configuration window Menu bar.
   - Click Save on the Navigation Configuration window toolbar.
   - Press Ctrl+S on the keyboard.

   New Navigation File: Workbench

6. Expand Project>Screens in the Workbench left-pane.

7. Do one of the following.



A	Click File>New>Object on the Workbench menu bar.
B	Click the New Object button on the Workbench toolbar.

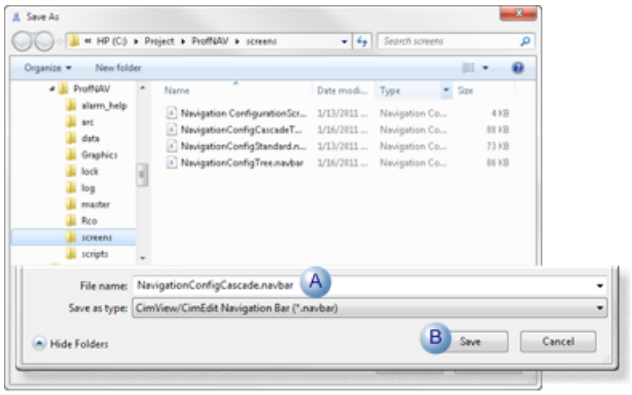
C	In the Workbench left pane:	
	Either	Or
	Double click **Navigation**.	a. Right-click **Navigation**. b. Select New on the Popup menu.
D	a. In the Workbench right pane. a. Right-click any *.navbar file. b. Select New on the Popup menu.	
E	Press Ctrl+N on the keyboard.	

Result: A Navigation Configuration window opens; no file has been created yet.

8. Right-click **Navigation**.
9. Select New on the Popup menu.
10. Right-click any *.navbar file.
11. Select New on the Popup menu.
12. Do any of the following to save a file.
    - Click File>Save on the Navigation Configuration window Menu bar.
    - Click Save on the Navigation Configuration window toolbar.
    - Press Ctrl+S on the keyboard.

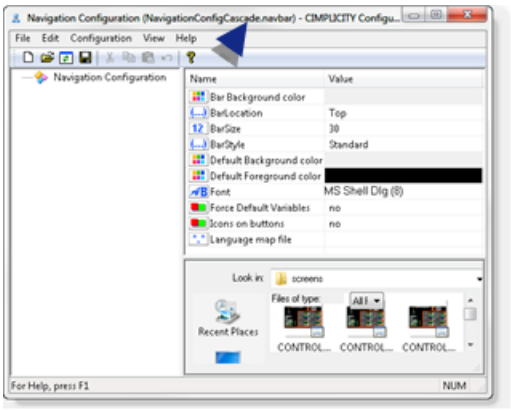
    A Save As browser opens.

13. Do the following.



A	Enter a name for the file in the **File name** field. **Note:** The configuration file extension is .navbar.
B	Click Save.

The Navigation Configuration window displays the newly saved file.



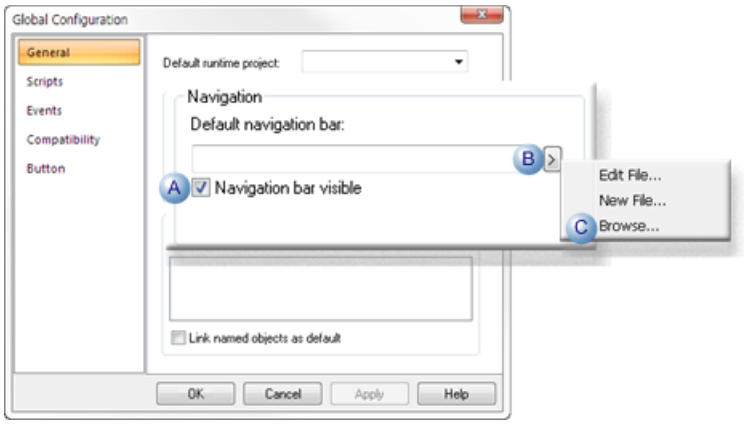
1.2. Open an Existing Navigation Configuration File

When a navigation file already exists, it can be opened from several locations, as follows.

- Global Configuration dialog box.
- Workbench.

Global Configuration Dialog Box

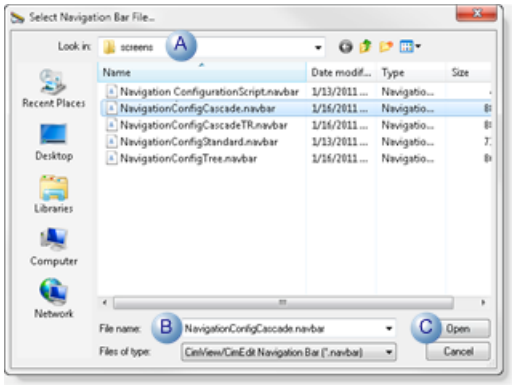
1. Select **General** in the Global Configuration dialog box
2. Do the following.



A	(Optional) Check Navigation bar visible to cause the Navigation bar to display during runtime
B	Click the Popup Menu button to the right of the **Default navigation bar** field.
C	Select Browse on the Popup menu.

The Select Navigation Bar File browser opens.

3. Do the following.



A	Open the folder that contains the configuration *.navbar file.
B	Select the file to open.
B	Click Open.

The path and filename are entered in the **Default navigation bar** field in the Global Configuration dialog box.

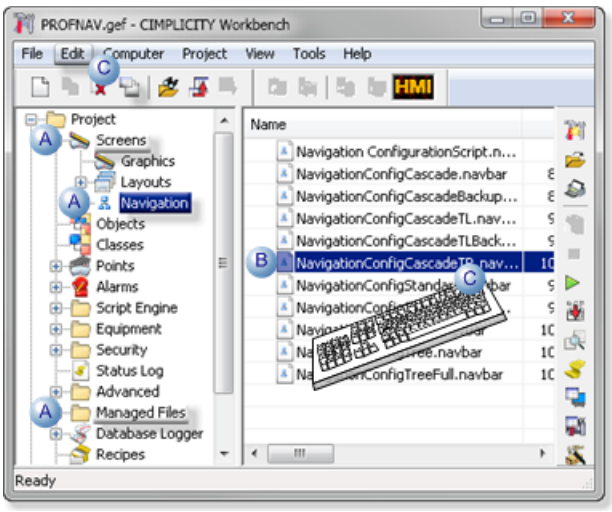
4. Click the Popup Menu button to the right of the **Default navigation bar** field.
5. Select Edit on the Popup menu.

The selected file opens in the Navigation CIMPLICITY Configuration Editor.

Workbench

A Navigation Configuration file can be opened through several Workbench locations.

Do the following.



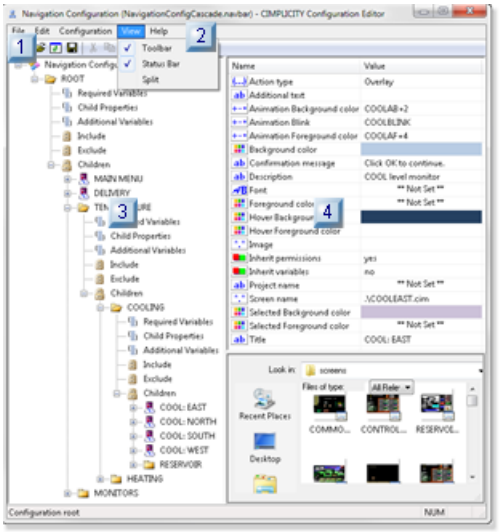
A	Select one of the following in the Workbench left-pane. • Screens • Navigation • Managed Files
B	Select a *.navbar file in the Workbench right-pane.
C	Do any of the following. • Select Edit>Properties on the Workbench menu bar. • Click the Properties button on the Workbench toolbar. • Right-click the navigation configuration file (<name>.navbar); select Open on the Popup menu. • Double-click the navigation configuration file (<name>.navbar). • Press Alt+Enter on the keyboard.

The selected file opens in the Navigation CIMPLICITY Configuration Editor.

2. Navigation Configuration Editor: Components

2. Navigation Configuration Editor: Components

The following tools are available in the CIMPLICITY (Navigation) Configuration Editor.



rect 112, 10, 132, 30 2.2. Navigation Configuration Editor: Menus *(page 691)*
rect 1, 17, 21, 37 2.1. Navigation Configuration Editor: Toolbar *(page 691)*
rect 65, 120, 85, 140 2.3. Navigation Configuration Editor: Left Pane *(page 695)*
rect 193, 119, 213, 139 2.4. Navigation Configuration Editor: Right Pane *(page 697)*

2.1 *(page 691)*	Navigation Configuration Editor: Toolbar
2.2 *(page 691)*	Navigation Configuration Editor: Menu
2.3 *(page 695)*	Navigation Configuration Editor: Left pane.
2.4 *(page 697)*	Navigation Configuration Editor: Right pane.

2.1. Navigation Configuration Editor: Toolbar

	Description	When Enabled
1	Create a new configuration file.	Always.
2	Open an existing configuration file.	Always.
3	Reload the configuration file from the disk.	After the configuration file is saved for the first time.
4	Save the configuration file.	Always
5	Cut the selected entity	When an entity, e.g. a folder or viewer, is selected.
6	Copy a selected entity or value.	When an entity, e.g. a folder or viewer, is selected.
7	Paste a cut/copied entity or value.	When an entity, e.g. a folder or viewer, is selected.
8	Undo Last Action	When an action has just been taken, e.g. a screen was added to the navigation configuration; undo will revert to the configuration just before the screen was added.
9	Open navigation documentation.	Always.

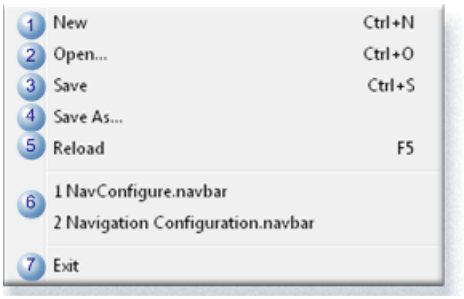
2.2. Navigation Configuration Editor: Menus

The CIMPLICITY Navigation Configuration Editor menu bar includes the following features.

rect 163, 20, 184, 43 *(page 694)*
rect 163, 20, 184, 43 *(page 694)*
rect 206, 20, 227, 43 *(page 695)*
rect 98, 20, 119, 43 *(page 694)*
rect 98, 20, 119, 43 *(page 694)*
rect 39, 20, 60, 43 *(page 693)*
rect 6, 20, 27, 43 *(page 692)*

A *(page 692)*	File
B *(page 693)*	Edit
C *(page 694)*	Configuration
D *(page 694)*	View
E *(page 695)*	Help

1. File menu

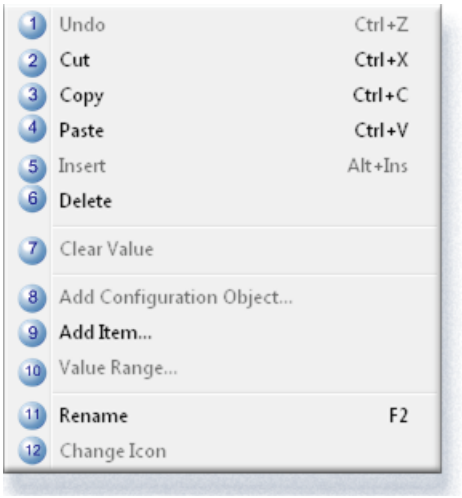


	Item	Description
1	New	Creates a new configuration file.
2	Open	Opens a Windows Open browser to find and open an existing configuration file.
3	Save	Saves the current configuration file
4	Save As	Save the current configuration file with the name and location entered in the Windows Save As browser.

	Item	Description
5	Reload	Refreshes the CIMPLICITY Configuration Editor display.
6	Recent files	Opens a selected configuration file that was recently opened in the CIMPLICITY (Navigation) Configuration Editor.
7	Exit	Closes the CIMPLICITY Configuration Editor.

1. Edit/Popup menus

A Popup menu can be opened (right-click) that is the same as the Edit menu when an entity is selected but is not in edit mode.

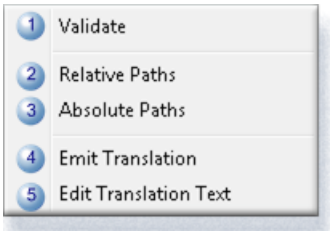


	Item	Description
1	Undo	When an action has just been taken, e.g. a screen was added to the navigation configuration; undo will revert to the configuration just before the screen was added.
2	Cut	Cuts a selected entity, e.g. a folder; the entity can be pasted in a selection location.
3	Copy	Copies a selected entity, e.g. a folder; the entity can be pasted in a selected location
4	Paste	Pastes a cut or copied entity in a selected location. A number is appended to the name.
5	Insert	Inserts a new item before a selected item in an ordered list, e.g. arguments list.
6	Delete	Deletes the selected entity.
7	Clear Value	Reverts the current selection to a non-assigned state (**Not set**). The color or property will be inherited from the parent.
8	Add Configuration Object	Opens an Add Configuration Object dialog box to add a new entity, e.g. a viewer or folder to the tree.
9	Add Item	Opens an Add dialog box to add a new, such as Navigation Configuration comment, variable, argument

	Item	Description
10	Value Range	
11	Rename	Makes a selected entity available for editing.
12	Change Icon	Turns the icon to the left of the selected configuration object into a drop-down list field, from which another icon can be selected, e.g. .

1. Configuration menu

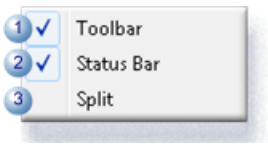
Features can be accessed when Navigation Configuration is selected in the CIMPLICITY (Navigation) Configuration Editor left pane.



	Item	Description
1	Validate	Check for errors in the navigation configuration.
2	Relative Paths	Assigns screen paths as relative paths, which are dependent on the file, not on the project.
3	Absolute Paths	Assigns screen paths as absolute paths. If the menu is used on other computers the paths to the screens must be exactly the same as specified in the configuration file.
4	Emit Translation	Emits words to a selected translation file.
5	Edit Translation Text	Opens the selected translation file.

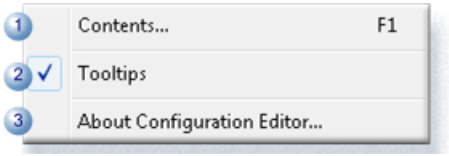
Note: The static Paths and Absolute Paths menu items are enabled after the first time the Navigation Configuration file is saved.

1. View menu



	Item	Description	
1	Toolbar	Select to:	
		Check	Display the toolbar.
		Clear	Hide the toolbar.
2	Status Bar	Select to:	
		Check	Display the status bar.
		Clear	Hide the status bar.
3	Split	Enables resizing the panes using the arrow keys on the keyboard.	

1. Help menu



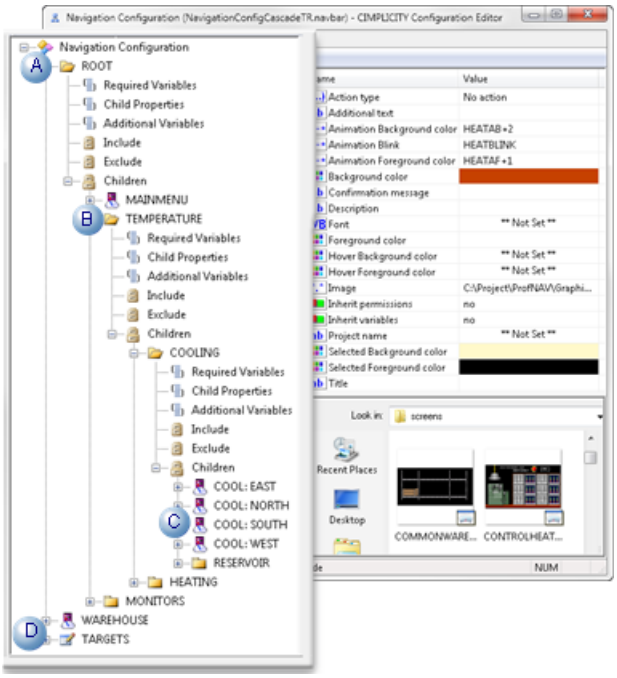
	Item	Description	
1	Contents	Opens the Navigation configuration topic in the CIMPLICITY documentation.	
2	Tool tips	Select to:	
		Check	Display tool tips
		Clear	Hide tool tips.
3	About Configuration Editor.	Opens details about the Proficy CIMPLICITY version and build.	

2.3. Navigation Configuration Editor: Left Pane

- Overview: Left pane.
- Runtime example: Based on left pane configuration.

Overview: Left-Pane

The tree in the CIMPLICITY Navigation Configuration Editor left pane lists the extended menu components, as follows.

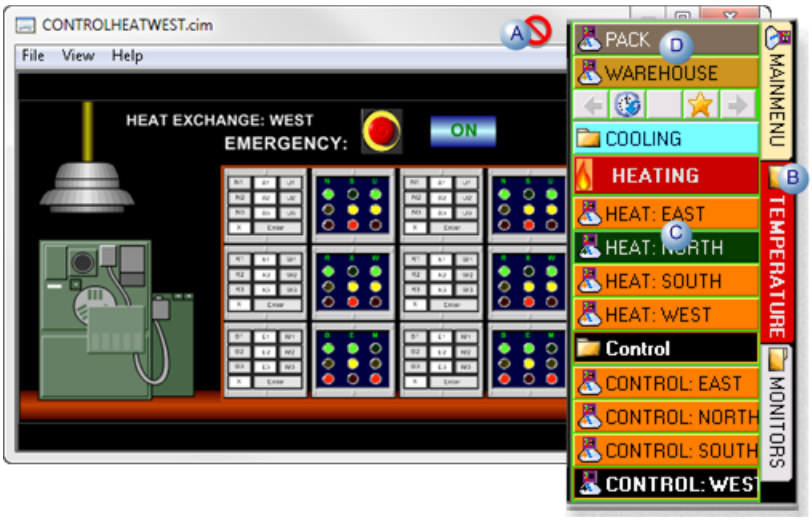


	Menu Item	Example
A	Root node of extended menu	ROOT **Important:** The root node displays for the following bar styles *(page 701)*: • Standard • Tree The node does not display for Cascade bar styles.
B	Child of root node	TEMPERATURE **Note:** The root node can have several children.
C	Children of child node	COOL: EAST COOL: NORTH COOL: SOUTH COOL: WEST **Note:** Each of these children can have children.
D	Quick menu items	WAREHOUSE TARGETS

Runtime Example: Based on Left Pane Configuration

During runtime, the menu items display in the hierarchy as it is displayed in the Navigation Configuration Editor left pane.

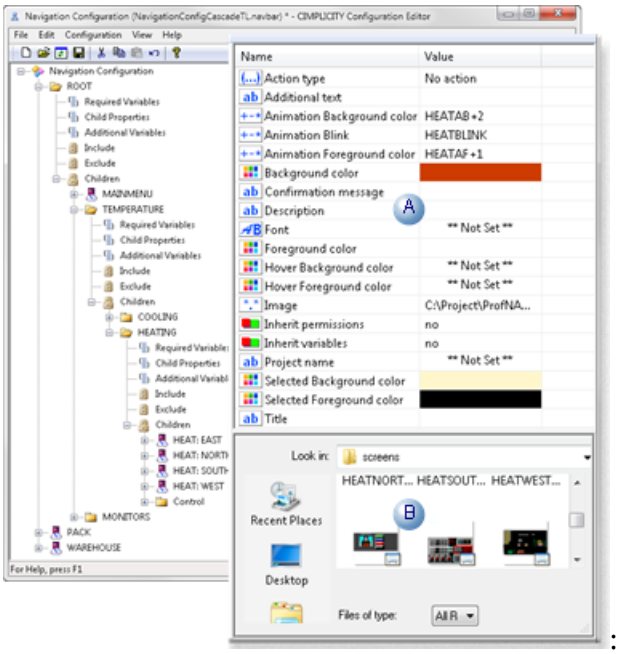
A Cascade bar style displays as follows.



	Menu Item	Example
A	Root node of extended menu	ROOT The node does not display for Cascade bar styles.
B	Child of root node	TEMPERATURE
C	Children oF child node	HEAT: EAST HEAT: NORTH HEAT: SOUTH HEAT: WEST
D	Quick menu items	PACK WAREHOUSE

2.4. Navigation Configuration Editor: Right Pane

The CIMPLICITY Navigation Configuration Editor right pane provides the tools to do the following.


:

1	Define and configure *(page 732)* the menu item nodes.
2	Quickly find the screens that will be used for the menus.

3. Navigation: Global Properties

3. Navigation: Global Properties

General navigation specifications can apply to the entire navigation menu.

Note: Some values that can be customized at child levels will override the entries at the selected level.

Several entries may be modified as configuration continues. They include the following.

3.1 *(page 699)*	Navigation: Default color properties.
3.2 *(page 701)*	Navigation: Bar display properties.
3.3 *(page 704)*	Navigation: Text properties.

3.4 *(page 706)*	Navigation: Icons on buttons.
3.5 *(page 709)*	Navigation: Force default variables.
3.6 *(page 713)*	Navigation: Menu translation.

3.1. Navigation: Default Color Properties

The following colors can be selected at the Navigation Configuration level in the Navigation Configuration Editor.

- Configuration: Stationary color properties.
- Runtime Example: Default colors

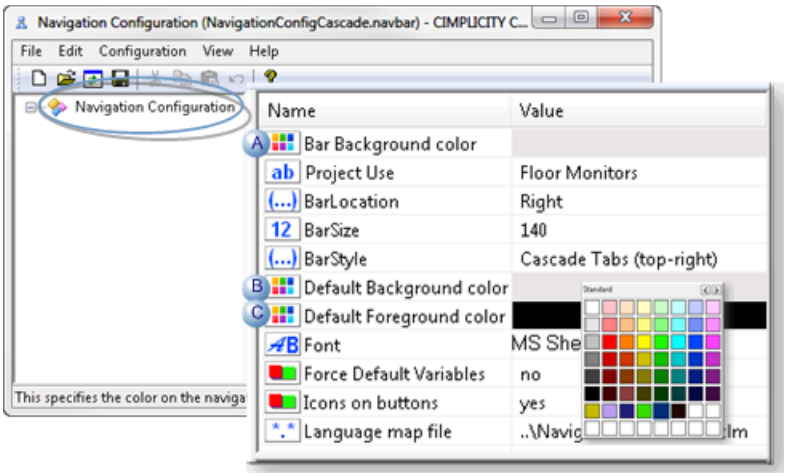
Configuration: Stationary Color Properties

1. Double-click a color property field.

A color palette opens.

1. Select the color that will display for the selected property.

Global Navigation Configuration color properties are as follows.



	Field	Selected Color colors the:
A	Bar Background color	Navigation toolbar color.

	Field	Selected Color colors the:
B	Default Background color	Background of the menu item when the node is not selected.
C	Default Foreground color	Foreground (font) of the menu item when the node is not selected.

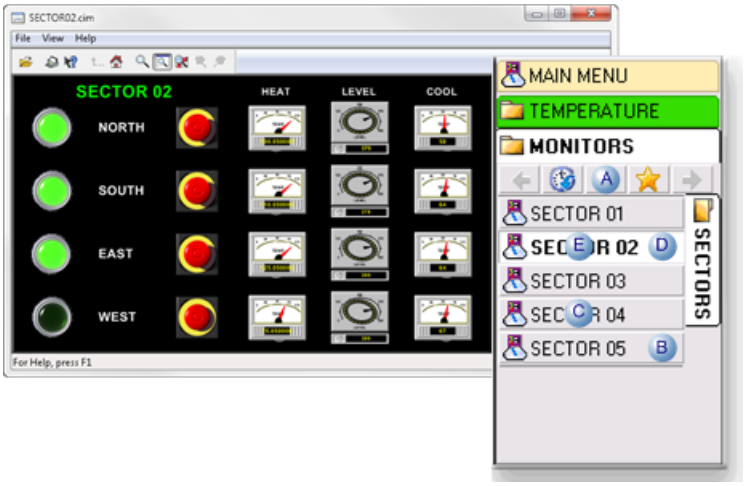
Note:

- Your computer's ambient colors display in the stationary color fields before other colors are selected.
- Each child in the Navigation hierarchy can have its own customized *(page 733)* background and/or foreground color. If the colors are not set for a child, the colors selected at the Navigation Configuration level display.

Runtime Example: Default Colors

A configuration object group on the navigation menu does not set *(page 729)* custom properties for the background and foreground colors.

The colors that display during runtime are the Navigation Configuration default colors.



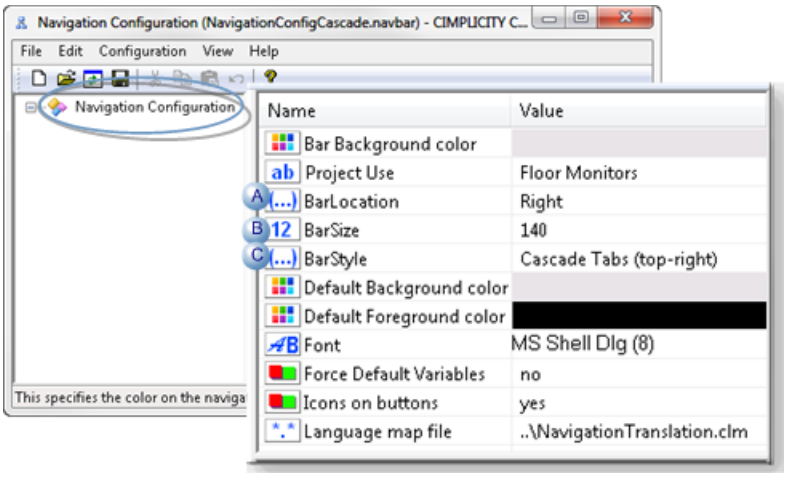
	Color	Selected Color colors the:
A	Bar Background	Color
B	Default Background	Background of the menu item when the node is not selected.
C	Default Foreground	Foreground (font) of the menu item when the node is not selected.
D	Default Selected Background	Color that is, by default, a lighter shade than the default background color.
E	Default Selected Foreground	Color that is, by default, darker than the default foreground color.

3.2. Navigation: Bar Display Properties

	Configuration: Navigation bar display.
	Runtime examples: Navigation bar display.

Configuration: Navigation Bar Display

Three fields work together to define how the navigation bar will display during runtime, as follows.



	Field	Description	
A	BarLocation	Choices in the dropdown menu enable you to display the navigation bar at any side of the CimView screen. Choices are:: • Top • Bottom • Left • Right	
B	BarSize	Factors that affect the bar size value (pixels) include: • Bar location. • Amount of detail included in the bar. • Depth and complexity of configuration object lists • Selected style. How the size displays is controlled by the bar location, as follows.	
		Bar Location	Value controls the:
		Top/ Bottom	Height
		Left/ Right	Width

C	BarStyle	Several styles are available. The navigation bar location partly determines the style that will be best suited for the selected navigation bar. However, any pairing of location and style is available. Styles are as follows.	
		Bar Style	Recommended Bar Location
		Standard	Top/Bottom
		Standard w/Child	Top/Bottom **Notes** • During runtime, a Child navigation bar displays the children of a selected node as a horizontal row of buttons. The buttons behave like quick launch buttons. • Select this option if the Navigation bar is being upgraded from a Navigation bar and child bar configured in an earlier CIMPLICITY version.
		Cascade Tabs (top-right)	Right
		Cascade Tabs (top-left)	Left
		Cascade Tabs (right)	Right
		Cascade Tabs (left)	Left

Important: An operator cannot change any of these specifications during runtime.

Runtime Examples: Navigation Bar

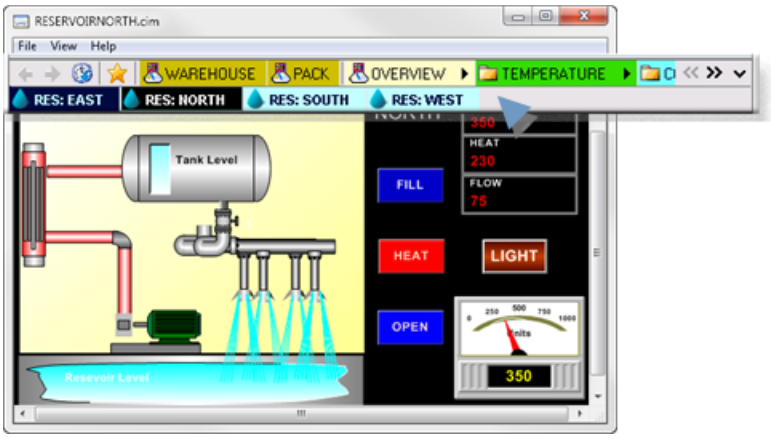
The bar location/size/style interaction influences values for a navigation bar that includes almost the same entries.

Note: Whether or not the root node displays depends on the selected bar style.

- Standard style.
- Cascade style.
- Tree style.

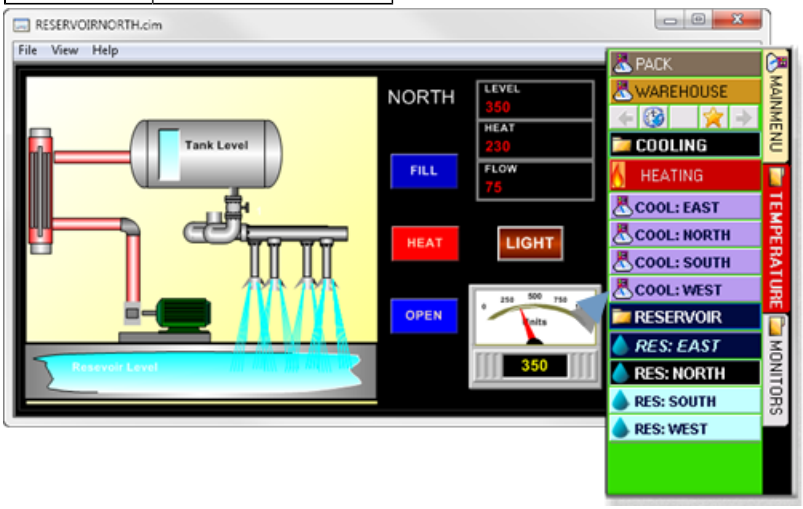
Standard Style

Field	Value
BarLocation	Top
BarSize	40
BarStyle	Standard/child



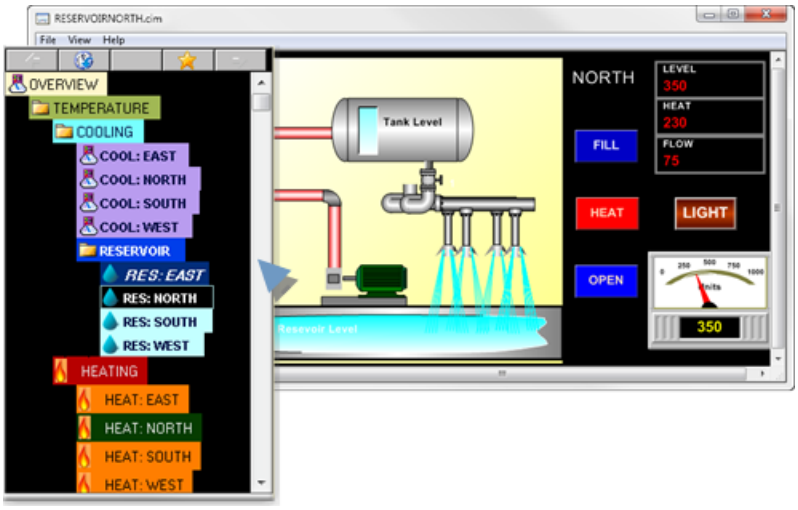
Cascade Style

Field	Value
BarLocation	Right
BarSize	140
BarStyle	Cascade Tabs (right)



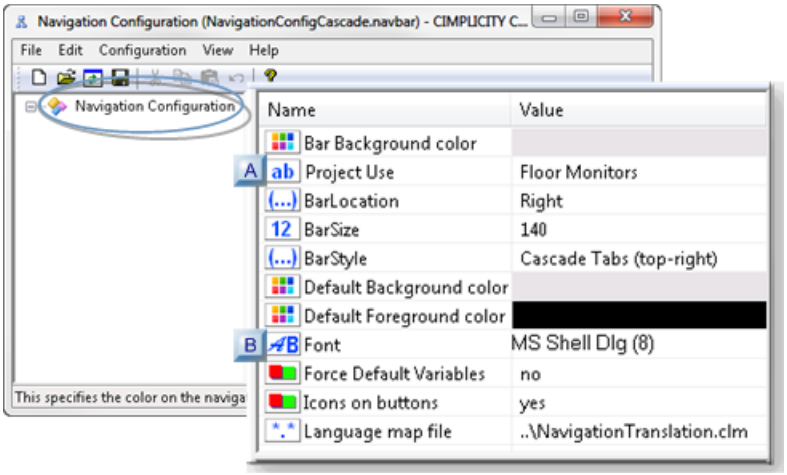
Tree Style

Field	Value
BarLocation	Left
BarSize	225
BarStyle	Tree



3.3. Navigation: Text Properties

Text properties at the Navigation Configuration level are as follows.



A *(page 704)*	Configuration comments.
B *(page 705)*	Default font.

A	Configuration Comments

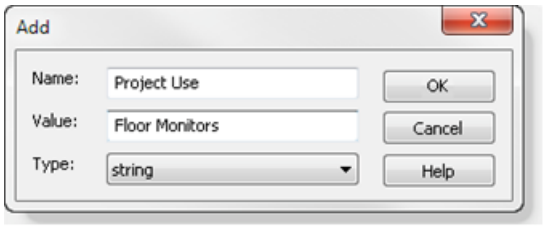
If there are any comments that should be retained as notes for the future or other applications engineers they can be entered in the Navigation Configuration section of the Navigation Configuration Editor.

These comments will be not be used in any way during runtime.

1. Do any of the following.
   - Right-click Navigation Configuration in the Configuration Editor left pane; select Add Item on the Popup menu.
   - Right-click the grid in the Configuration Editor right pane; Select Add Item on the Popup menu.
   - Select Navigation Configuration in the Configuration Editor left pane; click Edit>Add Item on the menu bar.

   An Add Item dialog box opens when you use any method.

2. Enter the following.



Name	A name that will help to quickly identify the comment
Value	Item that might be a useful reference for other users or in the future.
Type	`string` is the only type available.

3. Click OK.

   Result: The dialog box closes; the name and value are entered in the Navigation Configuration right pane.

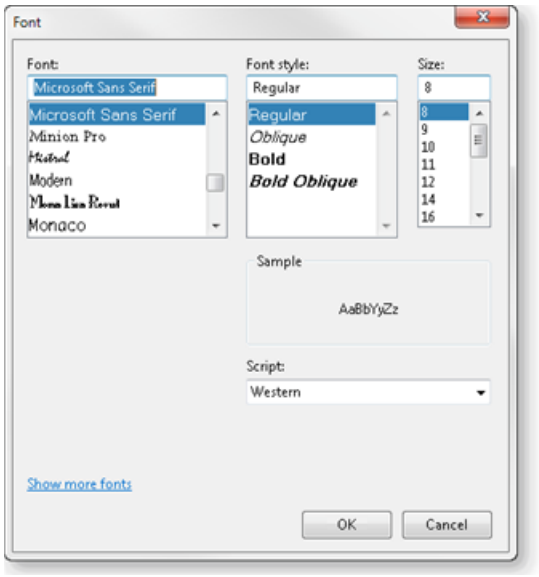
B	Default font

4. Double-click the **Font** field in the Navigation Configuration right pane.

   A Font dialog box opens.

5. Select the font, size and style you want to display on the Navigation buttons and menus.

   Note: Fonts can also be selected for a group of children and for Individual configuration objects; each selection will override this value for the group or object.



Note: If you select a font that cannot be used in a language you select, the words will not display correctly. Any UNICODE font, e.g. Arial UNICODE MS, can be used for all languages.

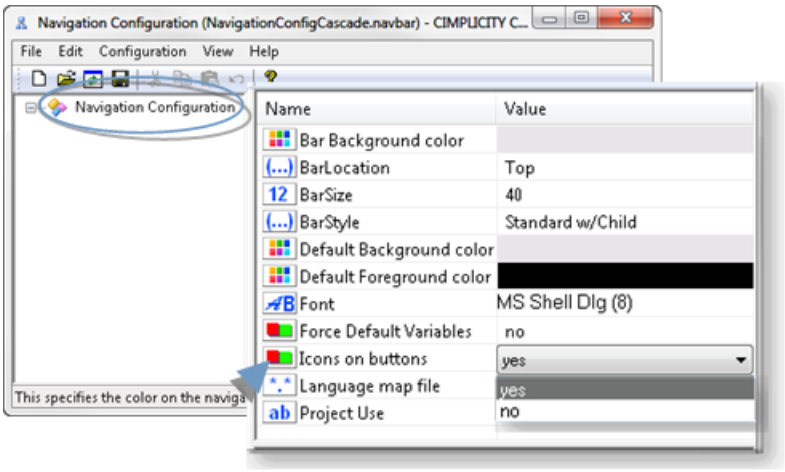
3.4. Navigation: Icons on Buttons

The CIMPLICITY Navigation Configuration Editor enables you to select to display or not display icons for each CimView Navigation button.

- Configuration: Icons on buttons.
- Runtime examples: Icons on buttons.

Configuration: Icons on Buttons

Select one of the following



yes

The icons that display in the Navigation Configuration Editor left pane will display on the CimView menu items.

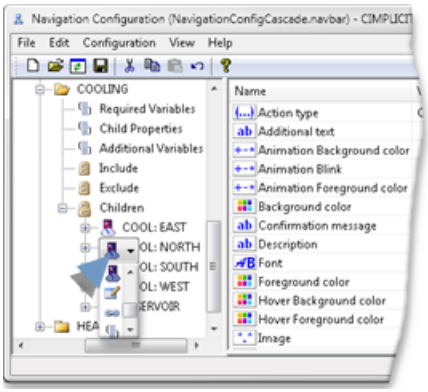
Select the icons that should display, as follows.

1. Select the menu configuration object.
2. Do one of the following.

- Click Edit>Change Icon on the Navigation Configuration Editor menu bar.
- Right-click the object; select Change Icon on the Popup menu.

An Icon drop down list is enabled.

1. Select the icon that should display on the menu for the selected object, if it should be different from the default.



Tip: An image *(page 741)* can also be selected for the configuration object, if it is, that image will display instead of the selected icon.

no

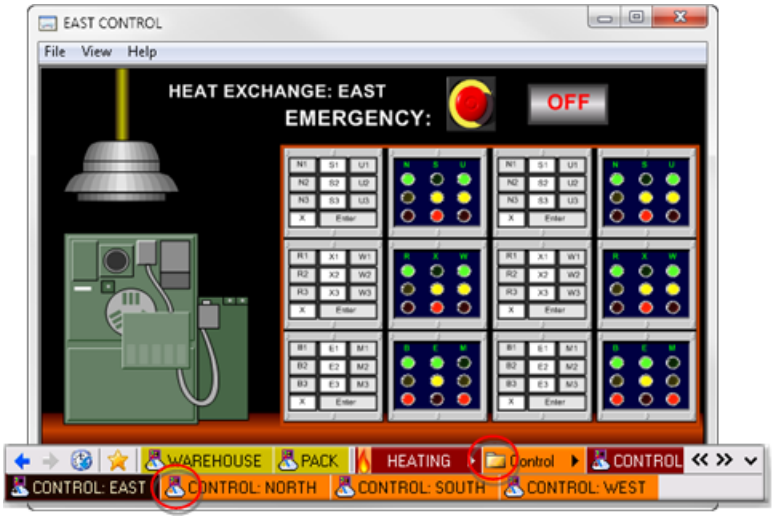
The node will not display an icon.

Runtime Example: Icons on Buttons

Icons display or are hidden, on a navigation menu, as follows.

Icons Display

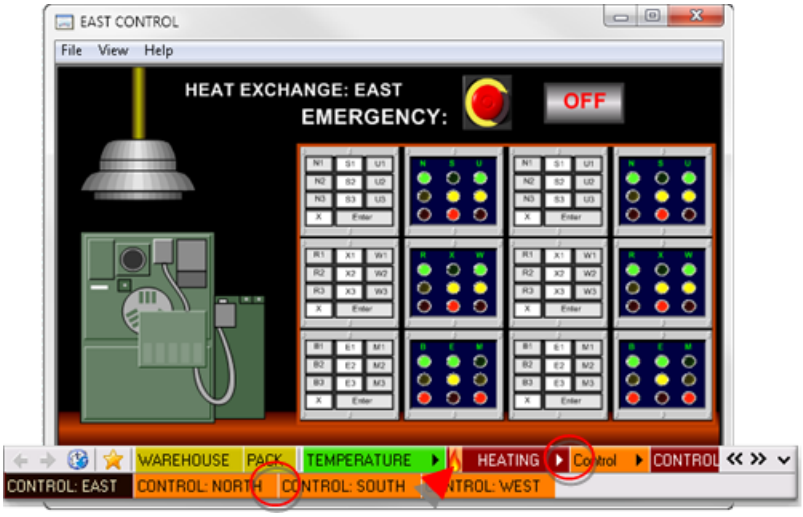
When **Icons on Buttons** is set to Yes, icons that were selected in the Navigation Configuration left-pane display on the menu nodes.



Icons Do Not Display

When **Icons on Buttons** is set to No, icons that were selected in the Navigation Configuration left-pane do not display on the menu nodes.

Note: Selected images *(page 741)* do display.



3.5. Navigation: Force Default Variables

Force Default Variables both:

- Causes variables' values that are assigned to a node to be passed to the CimView screen.
- Deals with navigation that is performed outside of the Navigation bar.

Examples

A screen can be opened outside of the Navigation bar when a user:

- Clicks a button on an open screen to open or overlay another screen; the Navigation bar is not used.
- Double-clicks a .cim file in Windows Explorer.

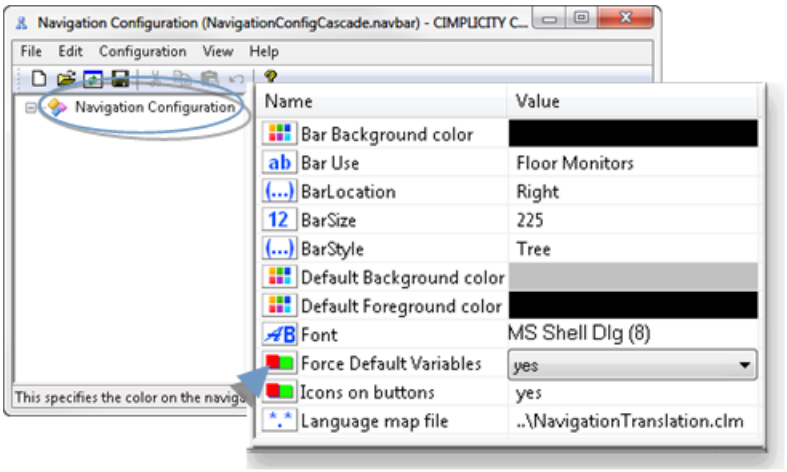
Because the screen is opened outside of the Navigation bar, there are no variables from the Navigation bar to pass. Force Default Variables enables the screen to use its own variables.

Note: How Navigation deals with what the Navigation bar display and how variables function when a screen is opened outside of the Navigation bar during runtime depend both on the:

- Configuration: Force Default Variables.
- Required Variables.
- Runtime Examples: Force Default Variables

Configuration: Force Default Variables

Selecting yes or no in the Force Default Variables field does the following when a screen is opened outside of the Navigation bar.



Value	Description
yes	The value of required variables that are defined for a node and that match variables on the screen are passed to the screen. **Note:** The screen's button (or tree entry) for the selected node will display on the Navigation bar.
no	Screen variable values are used either: • For required variables that are defined for a node and that match variables on the screen. • When there are no required variables. **Note:** The screen's button (or tree entry) for the selected node will not display on the Navigation bar.

Runtime: Force Default Variables

- Configuration: Force Default Variables example.
- Runtime: Force Default Variables example.

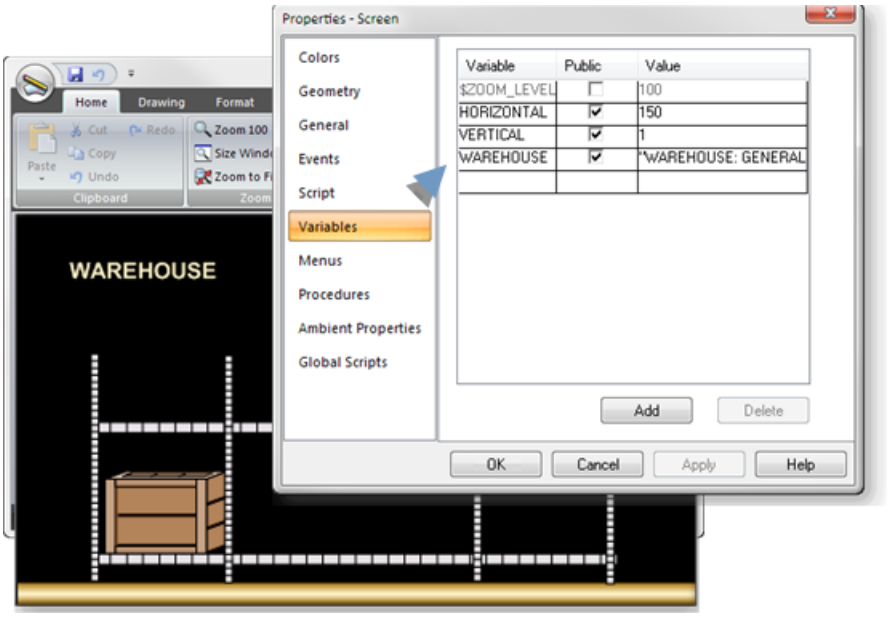
Configuration: Force Default Variables Example

A project includes one screen, Warehouse.cim, to monitor Warehouse activity.

CimEdit Screen Variables

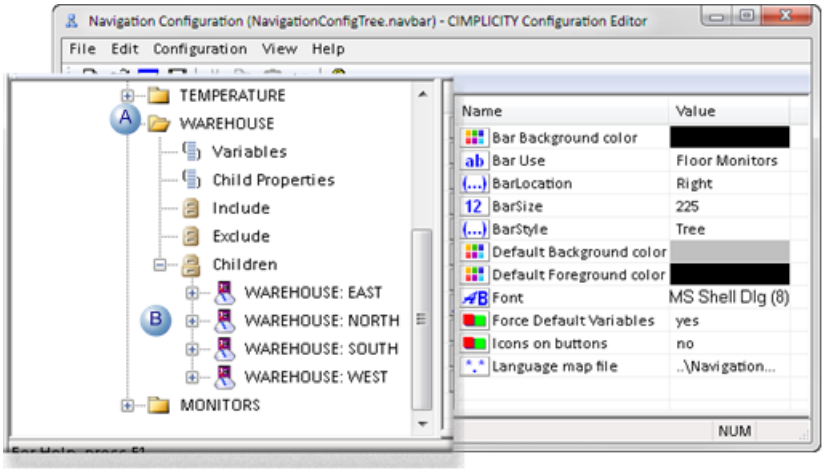
The CimEdit screen has three variables.

- HORIZONTAL
- VERTICAL
- WAREHOUSE



Navigation Configuration Editor Node Variables

The Navigation tree has the following WAREHOUSE nodes.



	Action	Nodes
A	No action	WAREHOUSE
B	Overlay	• WAREHOUSE: EAST • WAREHOUSE: NORTH • WAREHOUSE: SOUTH • WAREHOUSE: WEST Each Overlay node uses the Warehouse.cim screen.

Runtime: Force Default Variables Example

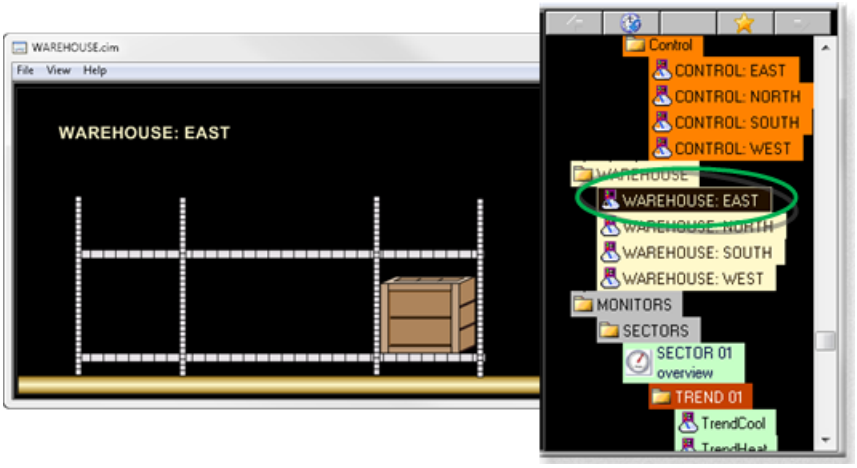
- Force default variables:
- Yes
- Force default variables:
- No

Force default variables: Yes

If the screen (e.g. Warehouse.cim) is opened outside of the Navigation bar for a CimView session, the variable values from the first node that the Navigation bar finds are passed to the screen.

Example

WAREHOUSE: EAST, which is the first node in the tree that uses the Warehouse.cim screen, is selected.

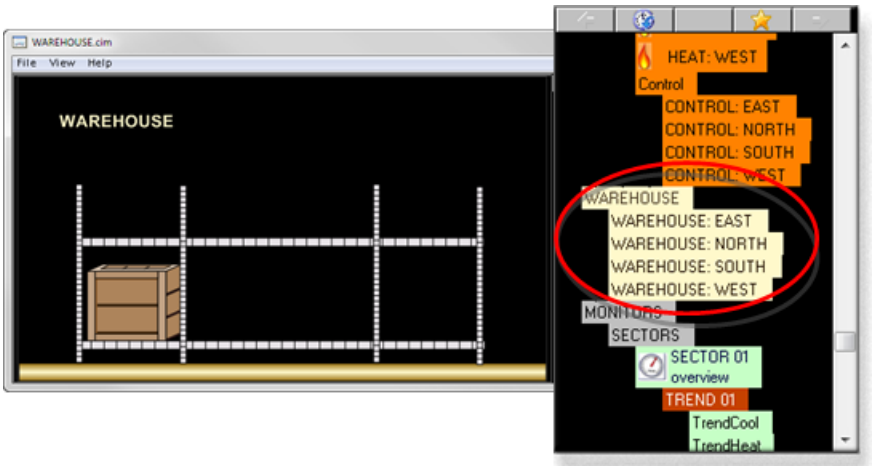


Force default variables: No

When the screen (e.g. Warehouse.cim) is opened outside of the Navigation bar for a CimView session, no variables are passed in, Navigation cannot find the screen; a node is not selected on the Navigation bar.

Example

WAREHOUSE: EAST is not selected.



Note: The basic Navigation bar display depends on the selected Navigation Configuration>Bar Style *(page 701)* in the Navigation Configuration window.

3.6. Navigation: Menu Translation

Text that is entered for the menus can be translated using CIMPLICITY Language Mapping .

- Configuration: Menu translation.
- Runtime examples: Menu translation.

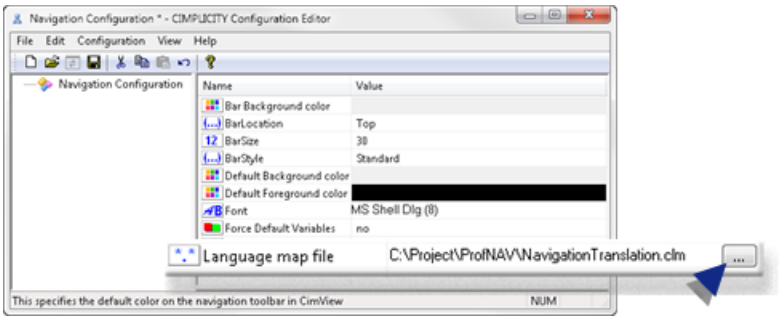
Configuration: Menu Translation

A *(page 713)*	Select the Language map file.
B *(page 714)*	Enter translation keys and text.

1. Select the Language Map file

Do one of the following in the Navigation Configuration Editor right pane.

- Enter the path to selected translation (*.clm) file in the **Language map file** field.
- Use the Browse button at the right of the field to find and select the translation (*.clm) file.

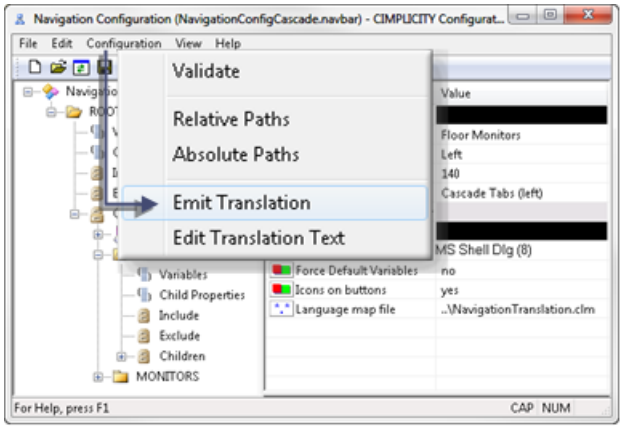


Note: After the Navigation Configuration Editor is saved for the first time the path can be relative or absolute. The default path is absolute.

1. Enter translation keys and text

You can emit text (and continue to emit text) to the translation file and edit the translation file anytime during the navigation configuration.

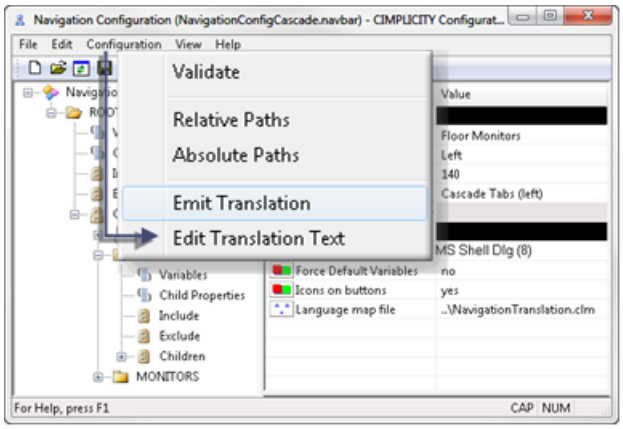
1. Click Configuration>Emit Translation on the Navigation Configuration Editor menu bar when you are ready to emit text.



   The following text is emitted to the selected translation file.

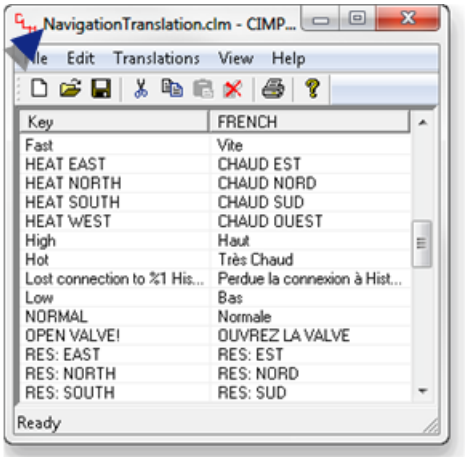
   - Button text
   - Title
   - Confirmation message
   - Description
   - Additional text

2. Click Configuration>Edit Translation Text on the Navigation Configuration Editor menu bar.



The selected translation file opens.

3. Edit the translation keys the same way you do with translation for other text.



Result: The menu text items will be translated according to your specifications.

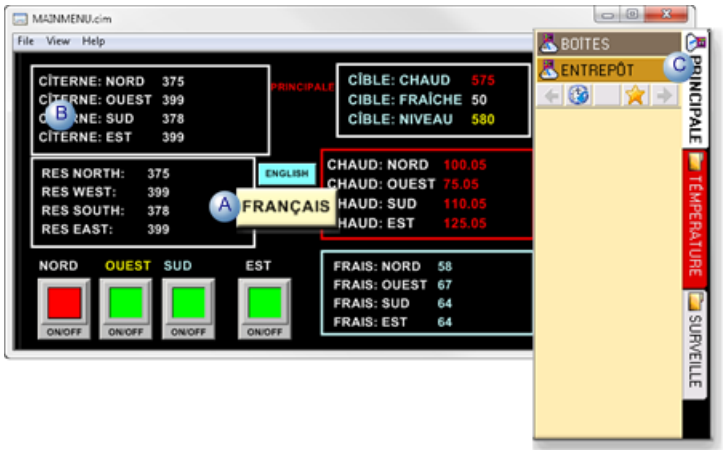
Runtime Example: Menu Translation

A project's Navigation bar can be viewed in English or French.

Buttons to select English or French display on an Overview (Principale) screen.

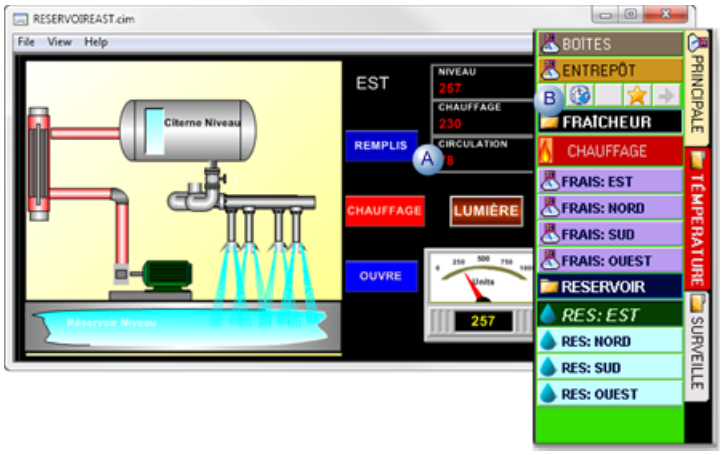
The language that displays on the Navigation menu is as follows.

4. French is selected for the Navigation menu.

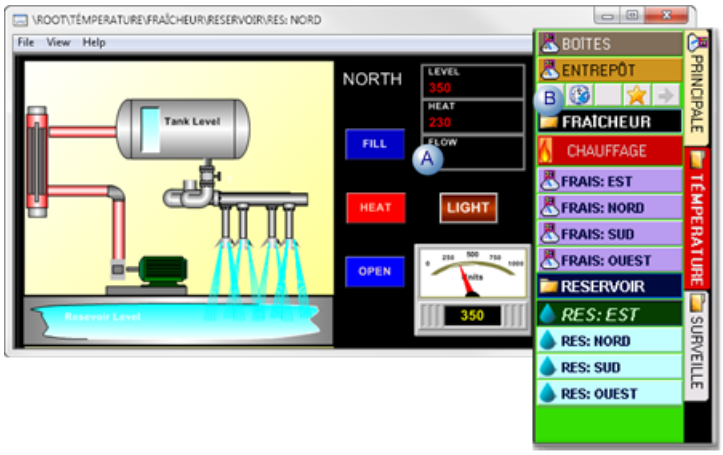


A	An operator clicks a French button to display the screen in French.
B	The screen displays in French.
C	The Navigation menu displays in French.

5. The Navigation menu displays in French, as follows.

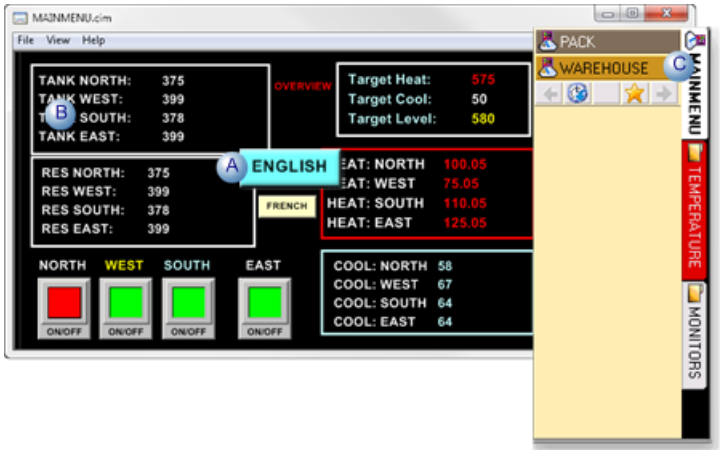


A	An operator selects a CimView screen that displays in French.
B	The Navigation menu displays in French.



A	An operator selects a CimView screen that displays in English only.
B	The Navigation menu continues to display in French.

6. English is selected for the Navigation menu.



A	An operator clicks an English button to display the screen in English.
B	The screen language changes to English.
C	The Navigation menu changes to English. The Navigation menu will continue to display in English until the French language is selected again.

4. Add: Configuration Objects

4. Add: Configuration Objects

A Navigation bar includes the objects that are added to the tree in the Navigation Configuration Editor's left-pane.

4.1 *(page 718)*	Add: Root object
4.2 *(page 721)*	Add: Children objects.
4.3 *(page 724)*	Add: Quick Launch objects.

4.1. Add: Root Object

Important:

- The root configuration object can be edited after it is added to the navigation tree. However, it cannot be deleted; children can be deleted.
- The root configuration object does not display during runtime when a cascade menu style is used.

Do any of the following to add the root configuration object.

- Configuration: Add a root node.
- Runtime examples: Navigation node.

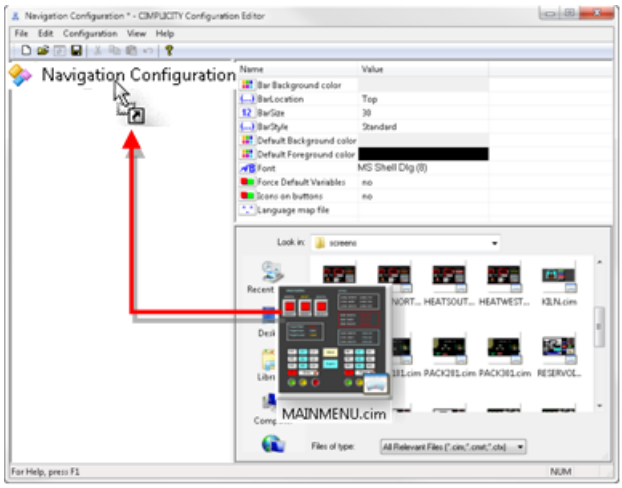
Configuration: Add a Root Node

Drag and Drop a Screen

Note: Dragging and dropping a screen speeds up configuration when the node will use an `Overlay` action type, which is one of three available action types.

1. Select a screen in the Navigation Configuration window lower right pane.
2. Drag the screen to the Navigation Configuration Editor left-pane.

   Important: This method is useful for the Standard and Tree bar styles. When a Cascade style is used, the Root node does not display on a Cascade style Navigation bar, so a screen that is selected for an Overlay action cannot be accessed.



The screen file name displays as the node name.

3. (Optional) Rename the node name.
   a. Select the root node.
   b. Do one of the following.
   - Right-click the name; select Rename on the Popup menu.
   - Select Edit>Rename on the Configuration Editor menu bar.
   - Press F2 on the keyboard.

   The name can now be edited.

   a. Type in the name that should display on the menu bar.

4. (Optional) Change the icon *(page 706)* .

   Result: The node will now display on the Navigation menu bar; the screen that will display is automatically entered in the Overlay parameters Screen name field.

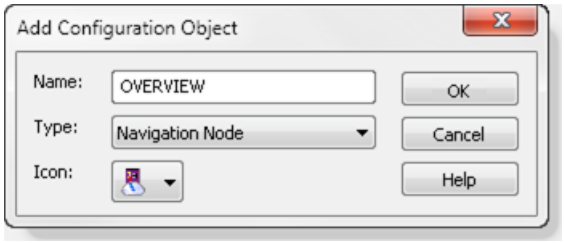
   Use Menu Items

5. Select Navigation Configuration in the Navigation Configuration Editor left-pane.

6. Do one of the following.
   - Right-click the selected **Navigation Configuration** or **Children** icon ; select **Add Configuration Object** on the Popup menu.
   - Click Edit>Add Configuration object on the Navigation Configuration Editor menu bar.

   An Add Configuration Object dialog box opens.

7. Enter the following.



Field	Description	
Name	Name that will display on the Navigation Menu bar.	
Type	The Navigation Node is the object that a user will click on to perform the selected root node action.	
Icon	An icon can be selected from the icon drop down list to represent the node. The icon selection will depend on your system's specifications and whether or not you use custom icons.	
	Default	**Notes** • The default icon for a node may be different, based on your system's icon list.. • The Navigation bar configuration can specify whether or not an icon *(page 706)* will display or if an image *(page 741)* will display.

8. Click OK.

The Add Configuration Object dialog box closes; The name that was entered displays as the root node name in the Navigation Configuration window left pane.

Note: When the root is added using the Add Configuration Object dialog box, an action type and specifications required for that action type will have to be entered before the navigation configuration can be saved with no errors.

Runtime Examples: Navigation Node

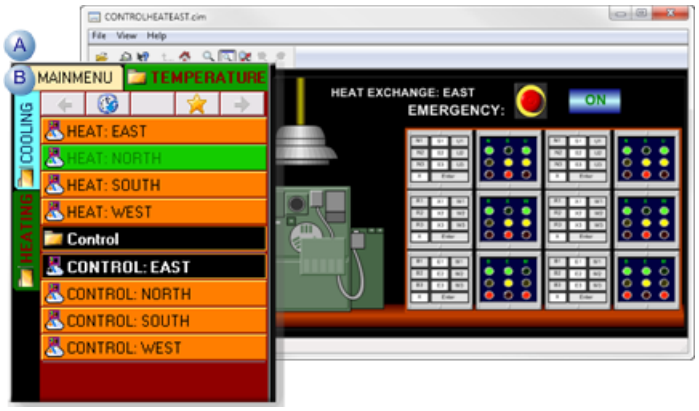
The bar style determines whether or not the Navigation object will display during runtime.

- Cascade bar styles.
- Tree bar style.

Cascade Bar Styles

A navigation bar specifies the following.

A	BarStyle	Cascade Tabs (top-left)
B	Root Node Name	ROOT

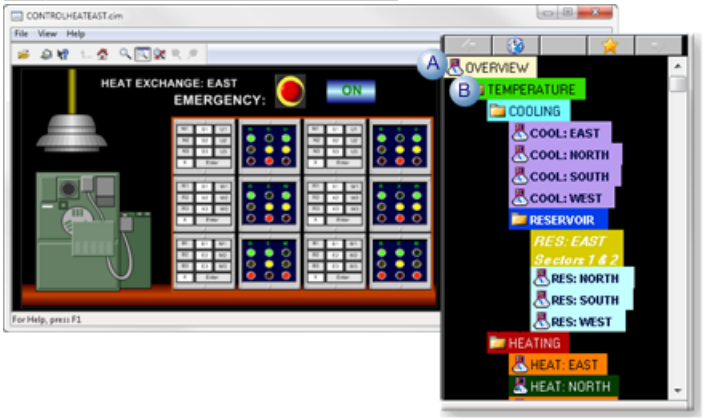


A	ROOT does not display during runtime.
B	First level child nodes display as the first menu items. **Note:** If the Navigation bar includes Quick Launch objects, they might display before the child nodes.

Tree Bar Style

A Tree bar contains a root object, named OVERVIEW.

A	BarStyle	Tree
B	Root Node Name	OVERVIEW



A	OVERVIEW displays as the tree's first node.
B	Objects on the tree display in the order they are listed in the Navigation Configuration Editor.

4.2. Add: Children Objects

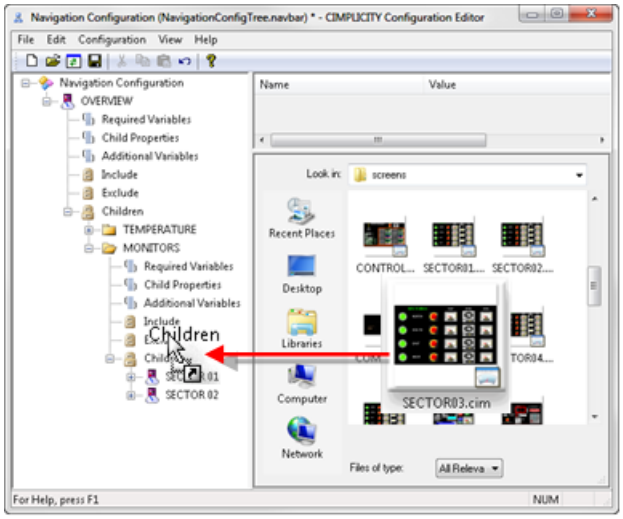
- Configuration: Add a child node.
- Runtime example: Child node.

Configuration: Add a Child Node

Drag and Drop a Screen

Note: Dragging and dropping a screen speeds up configuration when the node will use an `Overlay` action type, which is one of three available action types.

1. Select a screen in the Navigation Configuration window lower right pane.
2. Drag the screen to a selected **Children** folder in the Navigation Configuration Editor left-pane.



The screen file name displays as the node name.

3. (Optional) Rename the node name.
   a. Select the root node.
   b. Do one of the following.
   - Right-click the name; select Rename on the Popup menu.
   - Select Edit>Rename on the Configuration Editor menu bar.
   - Press F2 on the keyboard.

   The name can now be edited.

   a. Type in the name that should display on the menu bar.

4. (Optional) Change the icon *(page 706)* .

   Result: The node will now display on the Navigation menu bar; the screen that will display is automatically entered in the Overlay parameters Screen name field.

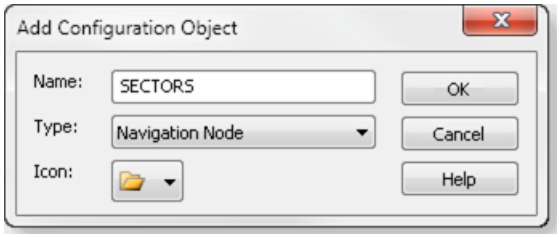
   Use Menu Items

5. Select Navigation Configuration in the Navigation Configuration Editor left-pane.
6. Do one of the following.

- Right-click the selected **Navigation Configuration** or **Children** icon; select **Add Configuration Object** on the Popup menu.
- Click Edit>Add Configuration object on the Navigation Configuration Editor menu bar.

An Add Configuration Object dialog box opens.

7. Enter the following.



Field	Description	
Name	Name that will display on the Navigation Menu bar.	
Type	The Navigation Node is the object that a user will click on to perform the selected root node action.	
Icon	An icon can be selected from the icon drop down list to represent the node. The icon selection will depend on your system's specifications and whether or not you use custom icons.	
	Default	**Notes** • The default icon for a node may be different, based on your system's icon list. • The Navigation bar configuration can specify whether or not an icon *(page 706)* will display or if an image *(page 741)* will display.

8. Click OK.

The Add Configuration Object dialog box closes; The name that was entered displays as the root node name in the Navigation Configuration window left pane.

Note: When a child is added using the Add Configuration Object dialog box, an action type and specifications required for that action type will have to be entered before the navigation configuration can be saved with no errors.

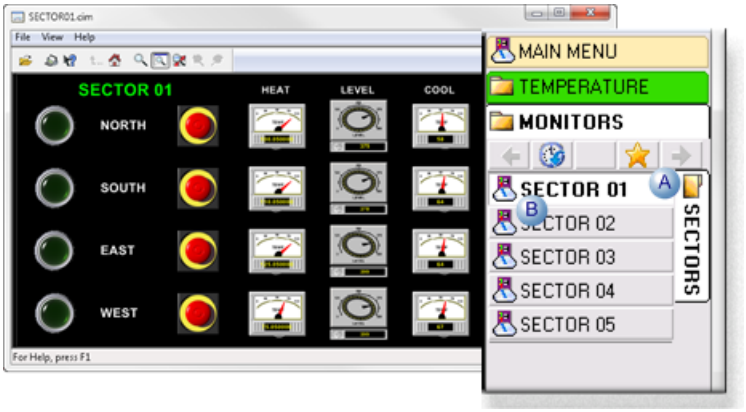
Runtime Example: Child Nodes

A Child group has been added to the Navigation bar, as follows.

A	SECTORS	A • No Action child object contains children that have Overlay actions. • **Folder** icon has been selected.

B	SECTOR 01 - SECTOR 05	• The SECTORS child contains five children, each of which opens an assigned CimView screen. • Each child is assigned the default icon.

A cascade menu displays the children relative to their position in the configuration tree.



A	SECTORS	Displays with the assigned folder as a child container.
B	SECTOR 01 - SECTOR 05	Display in a list that expands when SECTORS is selected.

4.3. Add: Quick Launch Objects

A Quick Launch button provides users with quick access to a selected CimView screen or script.

Important: A Quick Launch button:

- Cannot have children.
- Cannot use the No action action.
- Do not support::
- Inherit permissions.
- Inherit variables.
- Does not display on a Tree Navigation bar.

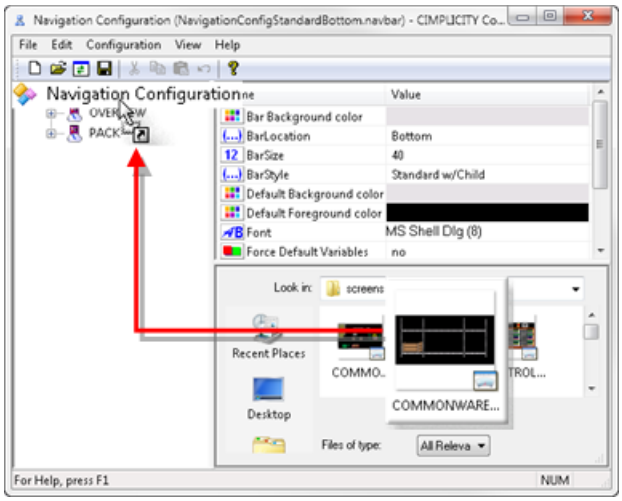
- Configuration: Add a Quick Launch node.
- Runtime example: Quick launch node.

Configuration: Add a Quick Launch Node

Drag and Drop a Screen

Note: Dragging and dropping a screen speeds up configuration when the node will use an `Overlay` action type, which is one of three available action types.

1. Select a screen in the Navigation Configuration window lower right pane.
2. Drag the screen to the Navigation Configuration Editor left-pane.



The screen file name displays as the node name.

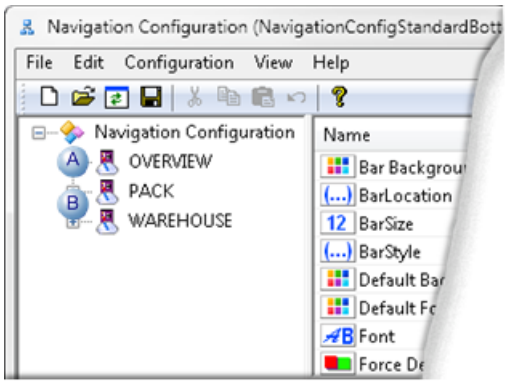
3. (Optional) Rename the node name.
   a. Select the root node.
   b. Do one of the following.
   - Right-click the name; select Rename on the Popup menu.
   - Select Edit>Rename on the Configuration Editor menu bar.
   - Press F2 on the keyboard.

   The name can now be edited.

   a. Type in the name that should display on the menu bar.

4. (Optional) Change the icon *(page 706)* .

Result: The Quick Launch nodes are listed on the same level as the Root node.



A	Root node
B	Quick Launch nodes

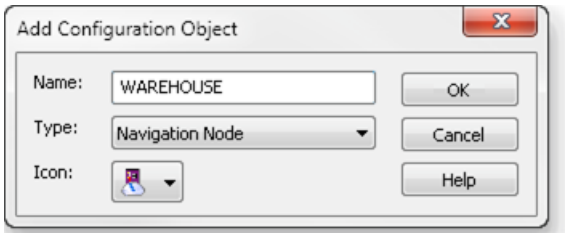
Navigation Configuration Menu Items

5. Select the Navigation Configuration object in the Navigation Configuration editor left pane.

6. Do one of the following.
   - Right-click **Navigation Configuration**; select Add Configuration Object on the Popup menu.
   - Click Edit>Add Configuration object on the Navigation Configuration Editor menu bar.

   An Add Configuration Object dialog box opens.

7. Enter the following.



Field	Description
Name	Name that will display on the Menu bar.
Type	The Navigation Node is the object that a user will click on to open a screen or run a script.
Icon	An icon can be selected from the icon drop down list to represent the node. The icon selection will depend on your system's specifications and whether or not you use custom icons.

Field	Description	
	Default	 **Notes** • The default icon for a node may be different, based on your system's icon list. • The Navigation bar configuration can specify whether or not an icon *(page 706)* will display or if an image *(page 741)* will display.

8. Click OK.

The Add Configuration Object dialog box closes; The name that was entered displays as the root node name in the Navigation Configuration window left pane.

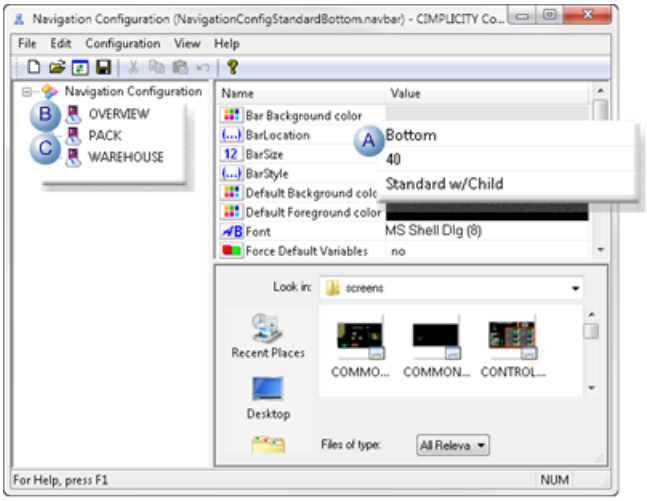
Important: When a Quick Launch node is added using the Add Configuration Object dialog box, an action type and specifications required for that action type will have to be entered before the navigation configuration can be saved with no errors.

Runtime Example: Quick Launch Node

- Configuration: Quick launch example.
- Runtime: Quick launch example.

Configuration: Quick Launch Example

A Navigation bar is configured with Quick Launch nodes as follows.

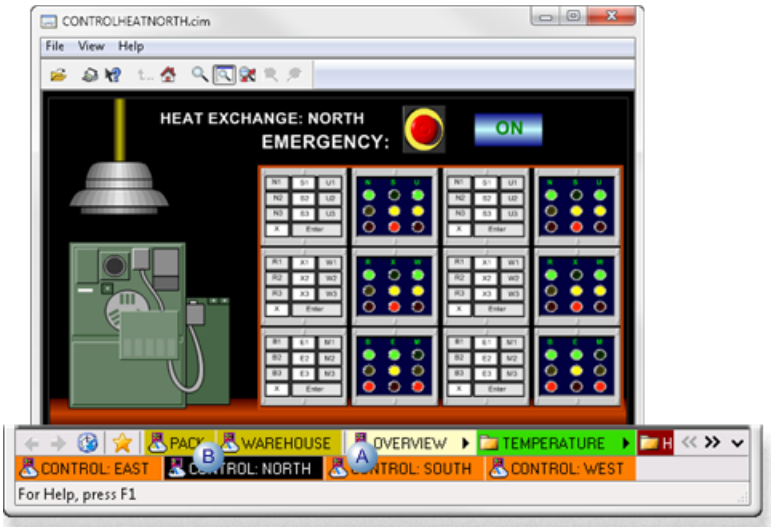


A	Selected Bar	Bar specifications are:	
		BarLocation	Bottom
		BarSize	40
		BarStyle	Standard w/Child

B	OVERVIEW	Root node.
C	PACK/WAREHOUSE	Quick Launch nodes with an Overlay action.

Runtime: Quick Launch Example

The Quick Launch menu items display next to the root node. Their exact position (above, to the left, etc.) depends on the bar location and style.



A	OVERVIEW	Root node
B	PACK/WAREHOUSE	Quick Launch nodes with an Overlay action.

Add Configuration Object Dialog Box Options

An Add Configuration Object dialog box is available for the following objects.

- Add root object.
- Add children objects.
- Add quick launch objects.
- Configuration object: Inherit permissions.

5. Navigation: Child Properties and Variables

5. Navigation: Child Properties and Variables

Recurring additional features in the Navigation Configuration window left-pane include the following.

5.1 *(page 729)*	Required Variables.
5.2 *(page 729)*	Child properties.
5.3 *(page 732)*	Additional variables.

5.1. Navigation: Required Variables

1. Lists required variables for a parent.
2. Sets Inherit Variables to Yes.

5.2. Navigation: Child Properties

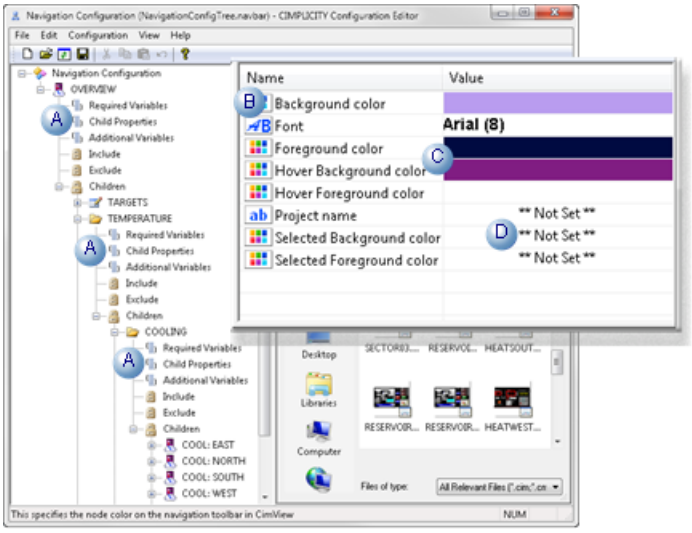
The Child Properties feature can eliminate duplication in configuration by automatically applying selected property values to objects that:

- Should display the same values during runtime.
- Are lower on the configuration tree than the selected Child Properties level.

- Configuration: Child properties.
- Runtime example: Child properties.

Configuration: Child Properties

Select the Child Properties level in which selected properties will be set to be inherited by configuration objects.



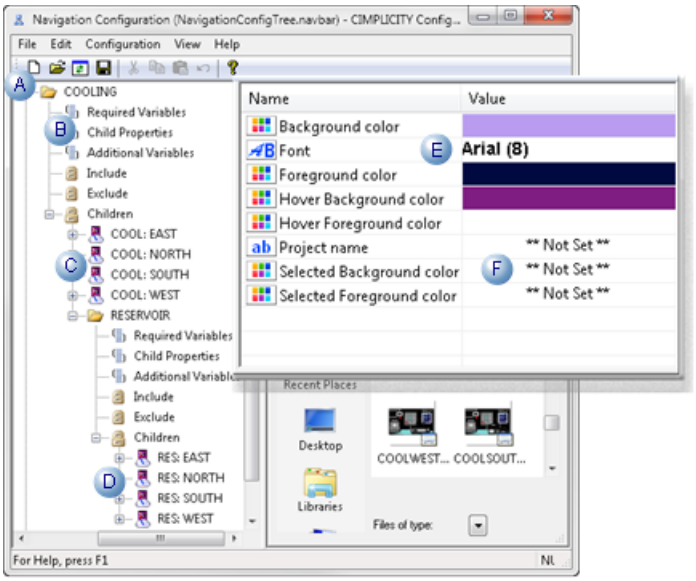
A	The Child Properties configuration pane is listed in the Navigation tree for each configuration object.
B	Properties that are available to be inherited by configuration objects that are lower on the Navigation tree. • Background color • Font • Foreground color • Hover Background color • Hover Foreground color • Project name • Selected Background color • Selected Foreground color
C	Any object below the selected level that does not have a value entered in its property pane will inherit the Child Properties value.
D	If a property is "Not Set" the Navigation bar continues going up to the next Child properties level in the tree. The first value that is set going up the tree will be used.

Runtime Example: Child Properties Set and Not Set

- Configuration: Child properties example.
- Runtime: Child properties example.

Configuration: Child Properties Example

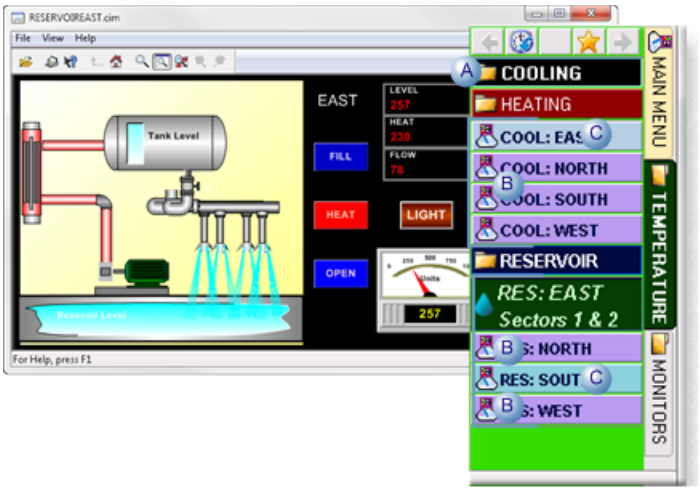
Some child properties have been set, as follows.



A	Selected Child (COOLING)	A configuration object is expanded; the object will have configured child properties.
B	Child Properties	The folder's child properties are selected.
C	Children (COOL:*)	The first children level below the configured child properties can inherit the configured child properties. Properties that the children have set will not be inherited.
D	Children (RES:*)	The second children level below the configured child properties can inherit the configured child properties. Properties that the children have set will not be inherited.
E	Properties Set	Values are entered in the following fields. • Background color • Font • Foreground color • Hover Background color • Hover Foreground color
F	Properties "Not Set"	Values have not been entered for the following properties. The values used will come from the first child properties pane going up the tree that has them set.

Runtime: Child Properties Example

A cascade Navigation bar displays inherited and custom properties, based on the Navigation bar configuration.



A	Selected Child (COOLING)	The selected configuration object displays colors that are inherited from a higher level.
B	Properties inherited	Children that did "Not Set" values for the configured properties inherit the properties set in the Child Properties pane.
C	Properties Custom	Children that set their own values for the configured properties display those values.

5.3. Navigation: Additional Variables

Additional variables:

- Are always used whenever a node that has additonal variables is selected and the CimView screen opened through through the Navigation bar.

The variables are not required to match any screen variable.

- Do nothing if the CimView screen is opened outside of the Navigation bar.

6. Configuration Object: Properties

6. Configuration Object: Properties

Action type Properties

The Navigation bar includes three action type properties.

Action	When a user clicks the node:

No action	No action is taken.
Overlay	The screen selected for the node overlays the screen that is currently displaying.
Script	A script that is selected for the node is triggered.

Other Object Properties

Most object properties apply to any action type.

6.1 *(page 733)*	Configuration Object: Color properties.
6.2 *(page 739)*	Configuration Object: Text properties.
6.3 *(page 741)*	Configuration Object: Image property.
6.4 *(page 743)*	Configuration Object: Inherit permissions.
6.5 *(page 748)*	Configuration Object: Inherit variables.
6.6 *(page 753)*	Configuration Object: Project name to qualify points.
6.7 *(page 753)*	Configuration Object: Screen name. (Overlay action only)
6.8 *(page 755)*	Configuration Object: Script entry point. (Script action only)

6.1. Configuration Object: Color Properties

6.1. Configuration Object: Color Properties

(All action types)

6.1.1 *(page 734)*	Configuration Object Stationary Colors
6.1.2 *(page 735)*	Configuration Object: Animated Colors

6.1.1. Configuration Object Stationary Colors

The following stationary colors can be configured for each configuration object.

- Configuration: Stationary color properties.
- Runtime example: Stationary colors.

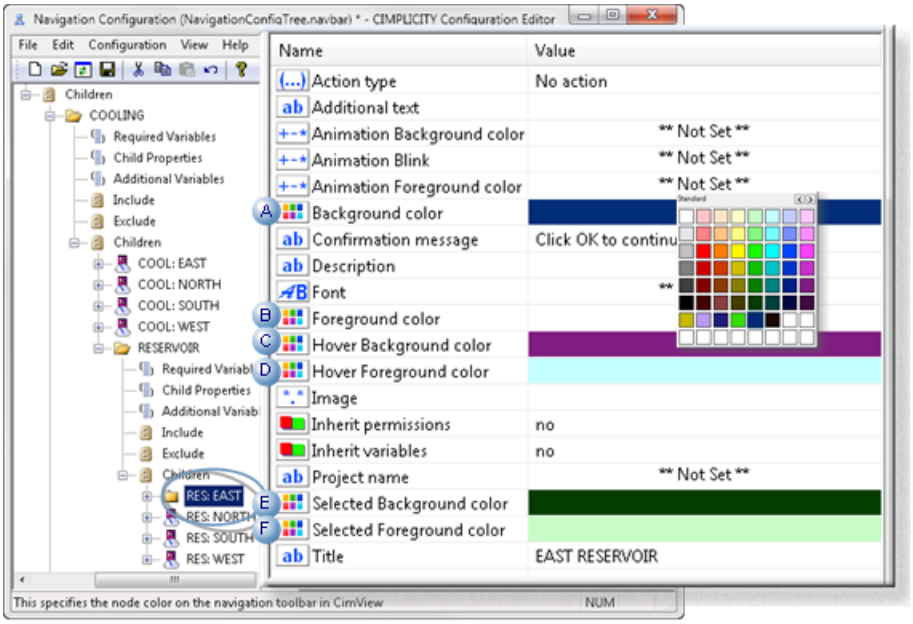
Configuration: Stationary Color Properties

1. Double-click a stationary color property field.

A color palette opens.

1. Select the color that will display for the selected property.

Stationary color properties are as follows.

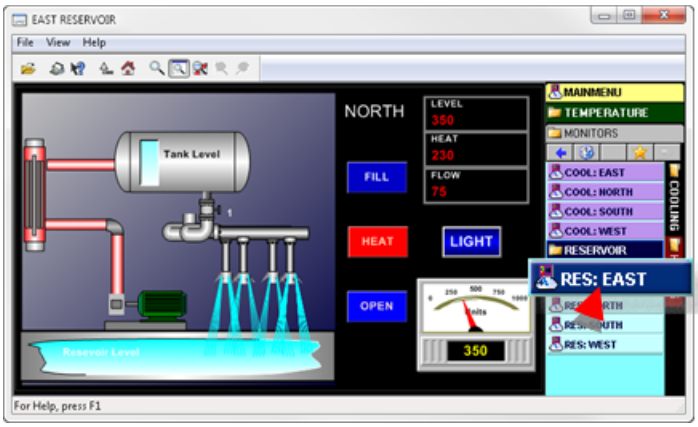


	Field	Selected Color colors the:
A	Background color	Background of the menu item when the node is not selected.
B	Foreground color	Foreground (font) of the menu item when the node is not selected.
C	Hover Background color	Background of the menu item when the mouse hovers over it.
D	Hover Foreground color	Foreground (font) on the menu item when the mouse hovers over it.
E	Selected Background color	Background of the menu item when the node is selected.

F	Selected Foreground color	Foreground (font) of on the menu item when the node is not selected.

Runtime Example: Stationary Colors

Colors were selected for the menu item status, RES: EAST.



During runtime, a menu item, RES: EAST, changes color based on the colors selected during configuration and a user's actions, follows.

	Menu Item is:	Field Defining Color	Menu Item Color
A	Is not selected	Background color	
B		Foreground color	
C	Hovered over	Hover Background color	
D		Hover Foreground color	
E	Selected	Selected Background color	
F		Selected Foreground color	

6.1.2. Configuration Object: Animated Colors

Animated colors provide a flexible information tool that can change Navigation node colors during runtime to help alert operators that there is some change or issue they need to be aware of.

	Configuration: Add a root node.
	Runtime examples: Navigation node.

Configuration: Animated Colors

Animation color is an integer expression that is not inherited. The expression can be an absolute number, a point (normally a virtual point) or a combination of point and integer.

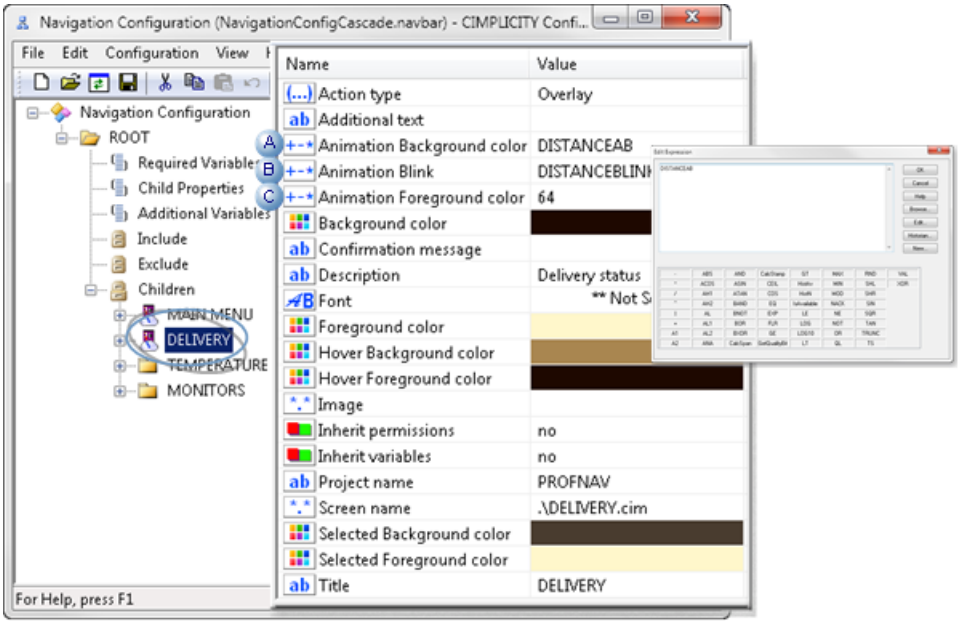
The purpose of the expression is to evaluate to a selected RGB *(page 178)* color value.

1. Double-click an **Animation** color field.

An Edit Expression dialog box opens.

1. Enter an expression that will evaluate to the RGB color number.

Animation color fields are as follows.



Field		Description
Animation Background color		Background color that either: • Contrasts with the stationary background color *(page 734)*, when the menu object is blinking. • Displays when the menu object is not blinking.
Animation Blink		Controls how rapidly the menu object will blink. Note; the rate corresponds to the CimView blink rate.
	Value	Number of milliseconds between blinks
	***	(Point not set) The animated background and foreground colors displays without blinking. **Note:** The stationary background and foreground colors (not selected or selected) do not display.
	0	The animated background and foreground colors displays without blinking. **Note:** The stationary background and foreground colors (not selected or selected) do not display.
	1	200

	2	400
	3	600
	4	800
	5+	1000
Animation foreground color	Foreground (font) color that either: • Contrasts with the stationary background color *(page 734)*, when the menu object is blinking. • Displays when the menu object is not blinking.	

Runtime Example: Animated Colors

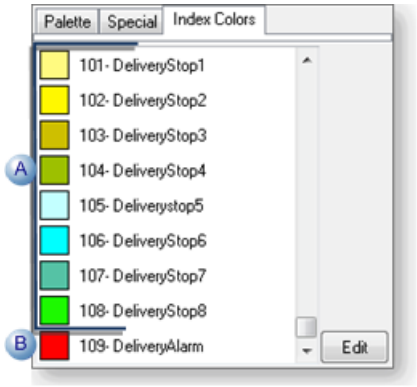
A node on the Navigation bar opens a screen that tracks the status of a delivery truck to and from its destination.

The node changes color to provide location status that can be quickly noted.

- Configuration: Color animation example.
- Runtime: Color animation example.

Configuration: Color Animation Example

1. Colors are assigned custom RGB (index) numbers, as follows.



	Index	Represents
A	101-108	Eight locations on the To and Back Delivery route.
B	109	An alarm state when the delivery driver needs assistance.

   a. Animated colors for the node are configured, as follows.

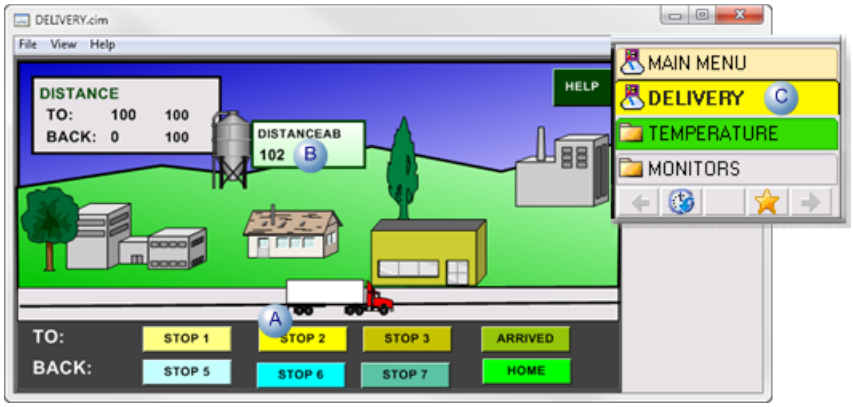
Field	Expression
Animation Background color	DISTANCEAB
Animation Blink	DISTANCEBLINK

Animation Foreground color	RGB #64 (Dark gray)

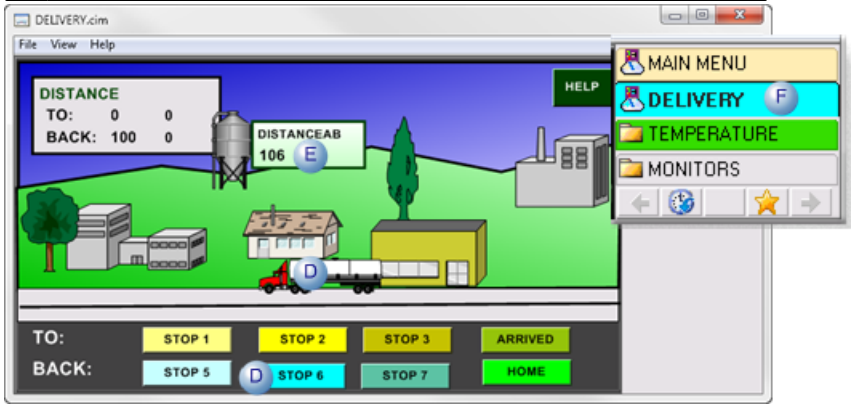
Runtime: Color Animation Example

Some DELIVERY node color changes during runtime are as follows.

2. The Navigation DELIVERY node displays the color that depicts the delivery status.

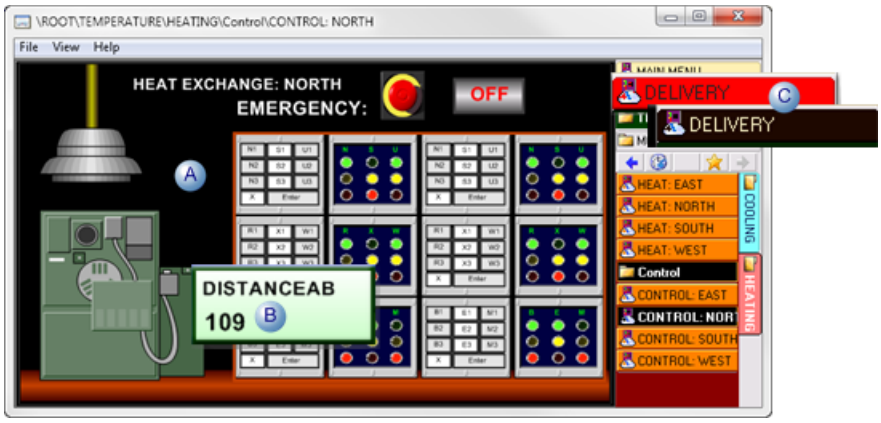


A	The delivery status is at STOP 2, headed to the delivery destination.
B	DISTANCEAB value is the custom RGB number 102.
C	The Navigation DELIVERY node displays the index color, 102.



C	The delivery status is at STOP 6, headed back to the factory.
D	DISTANCEAB value is the custom RGB number 106
E	The Navigation DELIVERY node displays the index color, 106.

3. An alarm situation occurs while delivery is in progress.



A	A screen that is not the DELIVERY screen is the current screen.
B	Because of the DELIVERY alarm state, DISTANCEAB value has changed to the custom RGB number 109, which triggers an event that sets the DISTANCEBLINK point value to 3.
C	The Navigation DELIVERY node blinks between the alarm Index color 109 and the stationary background/ foreground colors.

6.2. Configuration Object: Text Properties

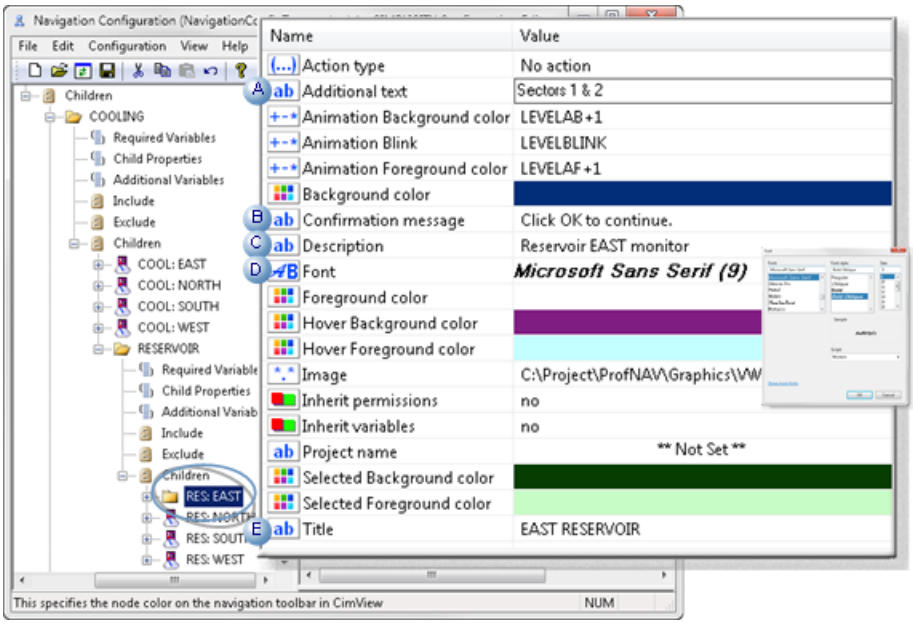
- Configuration: Text properties.
- Runtime example: Text properties.

Configuration: Text Properties

1. Double-click a text property field.
2. Enter the required text.

Note: When the Font field is double-clicked a Font dialog box opens.

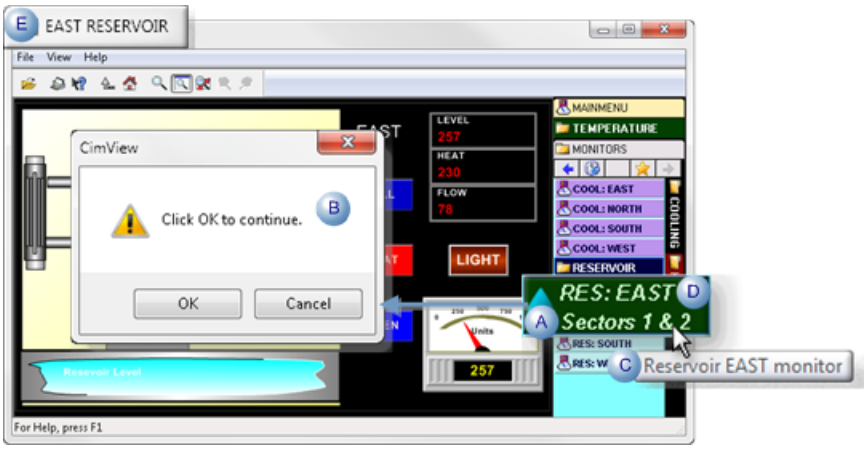
Text Property fields are as follows.



	Field	Description
A	Additional text	Displays as a second line of text on the node's menu item.
B	Confirmation message	Opens a confirmation message box that displays the entered text when a user clicks the selected node. The user must close the message box to proceed with the button's selected action.
C	Description	Displays as a tool tip in CimView, when a user selects the node's Navigation menu item. **Note:** If a description is not entered, the name entered as the node name, in the Navigation Configuration Editor left pane, displays as a tool tip.
D	Font	Opens a Font dialog box; the node text displays in the selected font/style/size.
E	Title	Displays on the title bar for the CimView screen that is opened when the selected configuration item is clicked.

Runtime Example: Text Properties

During runtime, the Navigation menu and CimView screen display the text properties selected for a node, as follows.



	Field	Entered Value
A	Additional text	Sections 1 & 2
B	Confirmation message	Click OK to continue.
C	Description	Reservoir EAST monitor
D	Font	Microsoft Sans Serif/Bold Oblique/9
E	Title	EAST RESERVOIR

6.3. Configuration Object: Image Property

An image file can display on a menu node instead of an icon.

The image can:

- Include transparency, which is supported.
- Be a maximum of 48x48 pixels.

The actual image size does not have to be an exact square, e.g. 16x16 or 32x32; the space assigned on the node will be the square that accommodates the largest side.

- Be any of the popular file types, including:
- .png
- .gif
- .jpg

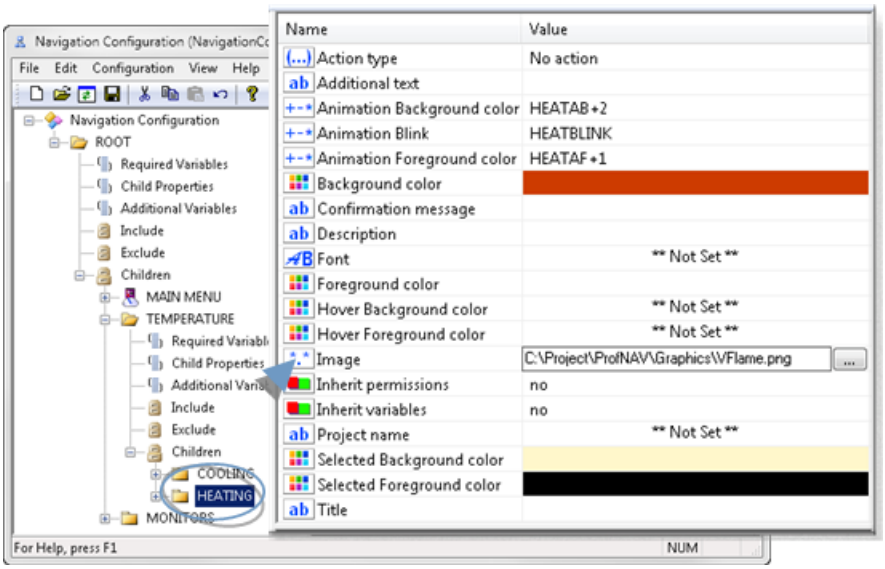
- Runtime example: Image on menu items.
- Configuration: Image property.

Configuration: Image Property

1. Click the button to the right of the Image field.

An Open dialog box opens.

1. Select any of the popular file types.



Result: The selected image will display on the node instead of an icon.

Important: If the graphic file is in the Screens folder or subfolder, a relative path can be entered in the Image field. for any other location, an absolute path must be used.

Example

The correct Image field entry for a Waterdrop.png image depends on its location, as follows.

File location	Correct Image Field Entry
C:\Project\ProfNav\Graphics\Waterdrop.png	C:\Project\ProfNav\Graphics\Waterdrop.png
C:\Project\ProfNav\Screens\Waterdrop.png	.\Waterdrop.png
C:\Project\ProfNav\Screens\Graphics\Waterdrop.png	.\Graphics\Waterdrop.png

Two .png files are selected to display on menu items.

The files and their sizes are:

File	Size (Pixels)
VFlame.png	10x20
Waterdrop.png	14x16

Runtime Example: Image on Menu Items

A Navigation menu in CimView displays the images in a square space that accommodates the largest image side, as follows.

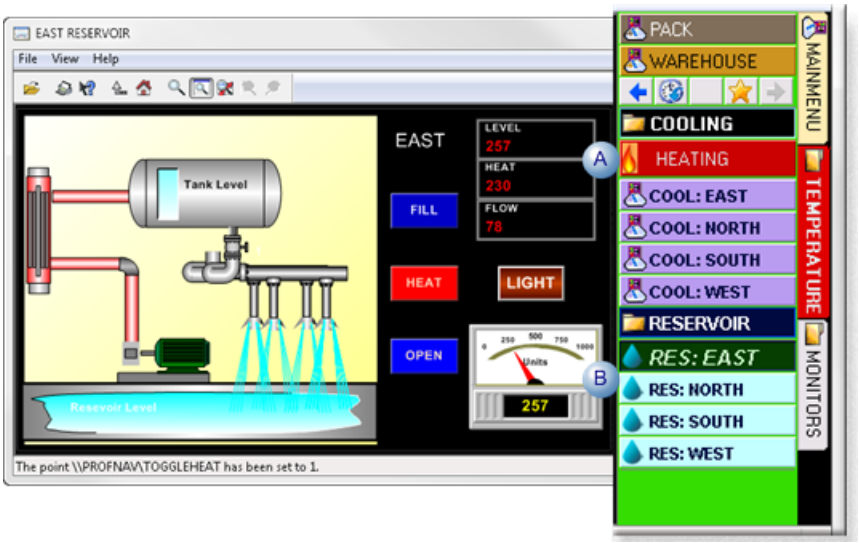
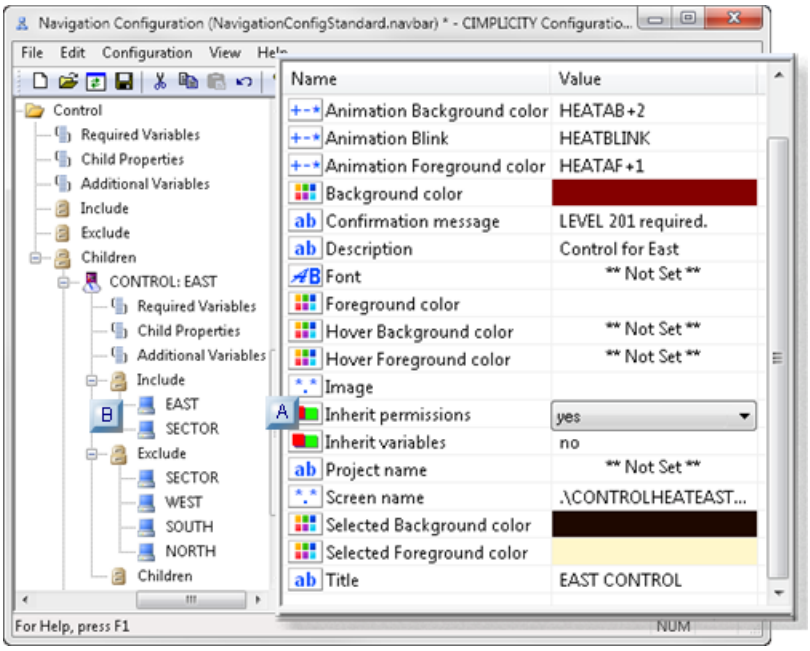


	Image	Display Space (Pixels)
A	Flame image	32x32
B	Water drop	16x16

6.4. Configuration Object: Inherit Permissions

(All action types)

Computers can be given permission to be included in performing the navigation action of a selected node or can be excluded. The configuration determines whether or not to grant permissions for the node and identifies the computers that will be included or excluded.



rect 161, 242, 189, 265 *(page 744)*
rect 54, 243, 73, 266 *(page 744)*

A *(page 744)*	Inherit Permissions.
B *(page 744)*	Include/exclude.

1. Inherit Permissions

A selected node can inherit the specifications to include or exclude computers from performing the navigation action.

Select one of the following in the **Inherit permissions** field in the Configuration Editor right pane for the selected node.

Yes	The node will inherit computer permissions from any higher node in the tree path, up to a node that does not inherit permissions.
No	The node will not inherit permissions from its parent. **Note:** Since the root node has no parent, select No for that node. (Yes is meaningless.)
Default	No

1. Include/Exclude

Computers can be included or excluded from the root node menu screen.

For example, a broad range of computers can be identified to be excluded from the screen. However within that range a few specific computers may need access. They can be selected to be included.

The opposite can also be configured. A broad range of computers can be identified to be included to view the screen. However within that range a few specific computers may need to be excluded. They can be identified to be excluded.

- Guidelines: Include/exclude.
- Computers to be excluded.
- Computers to be included.

guide: Guidelines: Include/Exclude

Keep the following in mind when you are determining what computer or range of computers you plan to include or exclude from the root node screen.

- If no computers are added to the Include list, all computers in the network will have access to the root node screen, except any computers or range of computers that are in the Exclude list.
- If one or more computers are added to the Include list, they will be the only computers that have access to the root node screen.

Computers to be Excluded

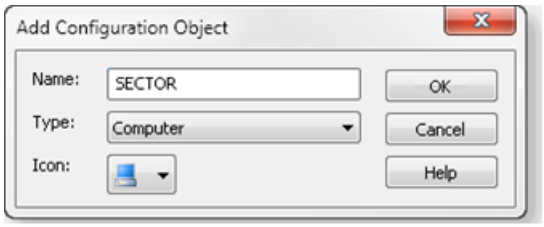
Computers that meet the criteria listed in the Exclude objects do not have permission to access the root node screen.

1. Select the Root Node>Exclude in the Navigation Configuration Editor left pane.

2. Do one of the following.
   - Right-click Include; select Add Configuration Object on the Popup menu.
   - Click Edit>Add Configuration object on the Navigation Configuration Editor menu bar.

   An Add Configuration Object dialog box opens.

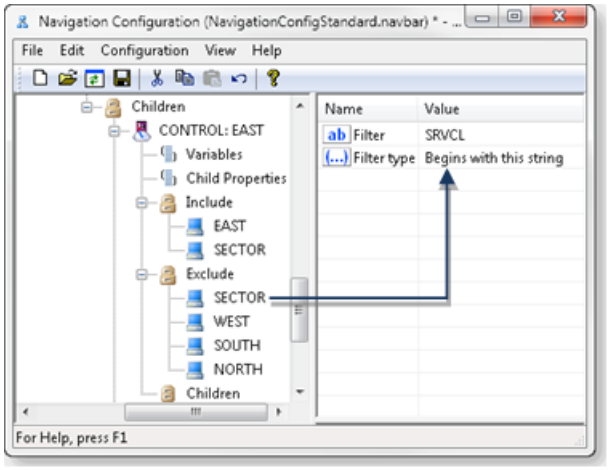
3. Enter a name in the **Name** field that will identify a computer or group of computers that will be excluded from the root node.



4. Click OK.

5. Select the object name in the Navigation Configuration Editor.

6. Enter the following.



Field	Description		
Filter	The value can be either of the following: • A single computer name • Characters that may be found in one or more computer names		
Filter type	The selected **Filter type** determines how Navigation will evaluate the entries.		
	No filter	The **Filter** value will be ignored until it is assigned a different **Filter type**.	
	Filter	Filter options are:	
		Option	Excludes computers whose names:
		Matches this string	Exactly matches the **Filter** value.
		Begins with this string	Begin with the string.
		Ends with this string	End with the string.
		Contains this string	Contain the string.
		Regular expression	Fulfill the regular expression criteria.

Computers to be Included

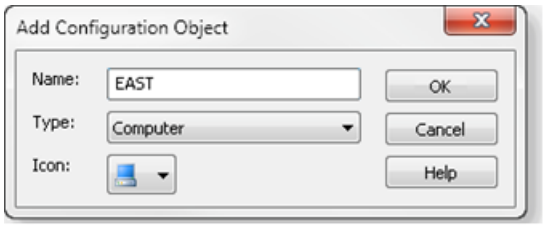
Computers that meet the criteria listed in the Include objects are the only computers that have permission to access the root node screen.

7. Select the Root Node>Include in the Navigation Configuration Editor left pane.

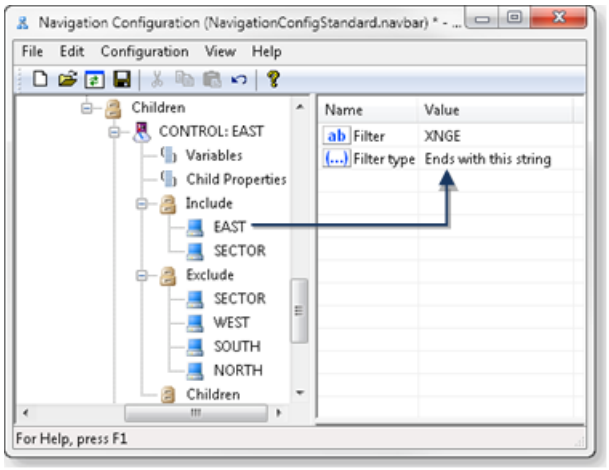
8. Do one of the following.
   - Right-click Include; select Add Configuration Object on the Popup menu.
   - Click Edit>Add Configuration object on the Navigation Configuration Editor menu bar.

An Add configuration Object dialog box opens.

9. Enter a name that will identify the computer or group of computers that will be included.



10. Click OK.
11. Select the object name in the Navigation Configuration Editor left-pane.
12. Enter the following in the right-pane.



Field	Description		
Filter	The value can be either of the following: • A single computer name • Characters that may be found in one or more computer names		
Filter type	The selected **Filter type** determines how Navigation will evaluate the entries.		
	No filter	The **Filter** value will be ignored until it is assigned a different **Filter type**.	
	Filter	Filter options are:	
		Option	Includes computers whose names:
		Matches this string	Exactly matches the **Filter** value.
		Begins with this string	Begin with the string.
		Ends with this string	End with the string.
		Contains this string	Contain the string.

Field	Description		
		Regular expression	Fulfill the regular expression criteria.

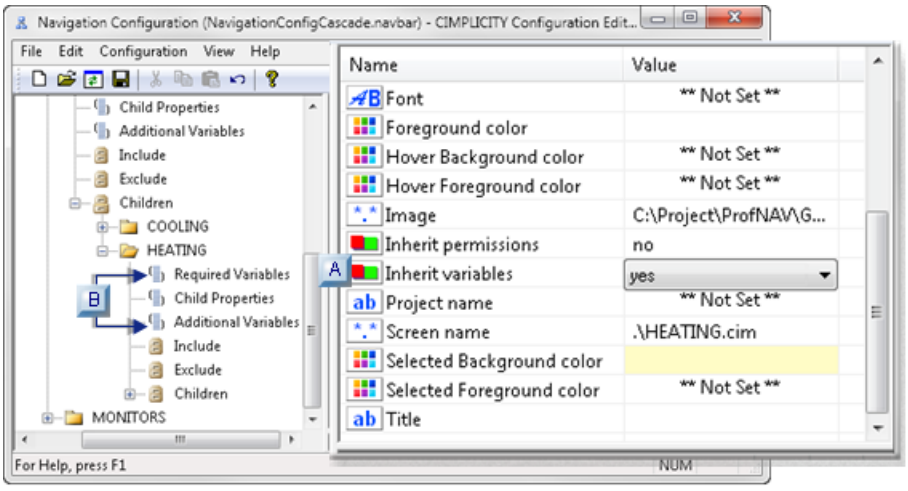
6.5. Configuration Object: Inherit Variables

(All action types)

- Configuration: Variables.
- Runtime examples: Variables.

Configuration: Variables

Variables can be entered at any level in the Navigation hierarchy and a selected node can inherit variables from parent nodes.



rect 44, 171, 68, 197 *(page 749)*
rect 198, 151, 219, 179 *(page 748)*

A *(page 748)*	Inherit Variables
B *(page 749)*	Variables

1. Inherit Variables

A selected node can inherit the variables entered for parent nodes in the navigation hierarchy.

Select one of the following in the **Inherit variables** field in the Configuration Editor right pane for the selected node.

Yes	Inherits variables from parent nodes in the navigation path up through the node that does not allow variables to be inherited.
No	The selected node will not inherit variables from parent nodes. **Note:** Since the root node has no parents, select No for that node. (Yes is meaningless.)
Default	No

1. Variables

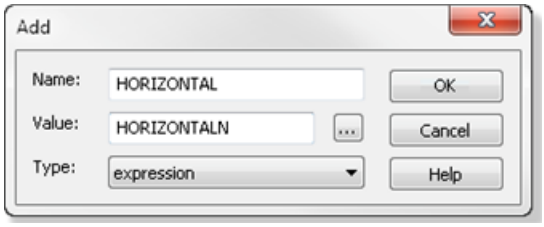
A node can include variables whose values can be used for that node and inherited by child nodes.

1. Select Variables in the left pane for the selected node.

2. Do one of the following.
   - Click Edit>Add on the Navigation Configuration Editor menu bar.
   - Right-click Variables; select Add on the Popup menu.
   - Right-click the right pane; select Add or Insert on the Popup menu.

   An Add dialog box opens.

3. Enter the following.



Field	Description
Name	Unique name
Value	A variable value.
Type	A variable is a string type.

Note: :

- If you have assigned a value to a selected screen and that value is different from the value in the Navigation Editor, the following will occur based on your configuration.

The Screen	The value assigned at the:
Does not inherit values	Screen level will display.

The Screen	The value assigned at the:
Does inherit values	Navigation Configuration Editor level will display.

- If a child screen can inherit variables from the parent, all of the variable values that are listed in the Navigation Configuration Editor will be applied to all of the same variables on the screen. It is important to make sure that variables that should not be inherited are not listed with variables that should be inherited.

Tip: If the value on the screen must be unique, it is safer to make the variable name unique to that screen. That way there will never be a chance that the value will be inherited.

Runtime: Variables

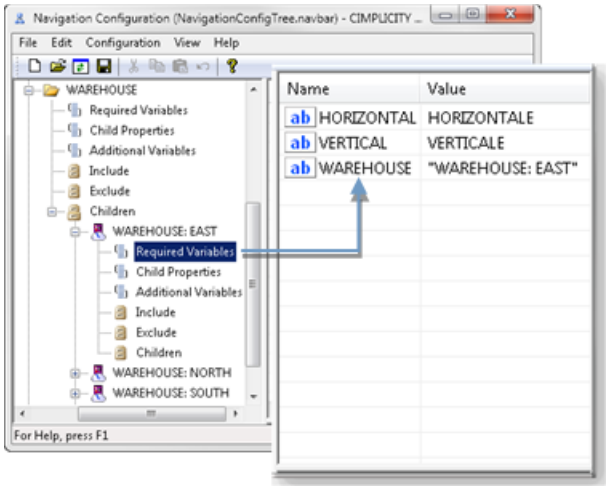
- Configuration: Variables example
- Runtime: Variables example

Configuration: Required Variables Example

A project includes one screen to monitor Warehouse activity.

Four Navigation menu nodes will use the same Warehouse.cim screen in an `Overlay` action.

A WAREHOUSE: EAST node includes the following string variables.



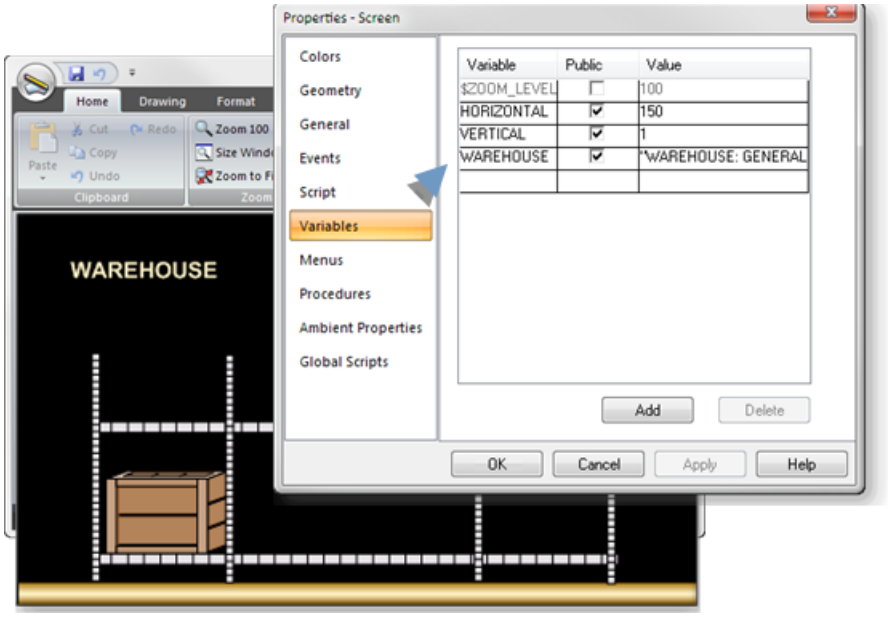
Variable	Value	Type
HORIZONTAL	HORIZONTALE	Point
VERTICAL	VERTICALE	Point
WAREHOUSE	"WAREHOUSE: EAST"	String

Important: Point values are treated as strings.

Three additional nodes include the same variables; the values are different, as follows.

Node	HORIZONTAL	VERTICAL	WAREHOUSE
WAREHOUSE: NORTH	HORIZONTALN	VERTICALN	"WAREHOUSE: NORTH"
WAREHOUSE: SOUTH	HORIZONTALS	VERTICALS	"WAREHOUSE: SOUTH"
WAREHOUSE: WEST	HORIZONTALW	VERTICALW	"WAREHOUSE: WEST"

The WAREHOUSE CimEdit screen includes the same three variables.

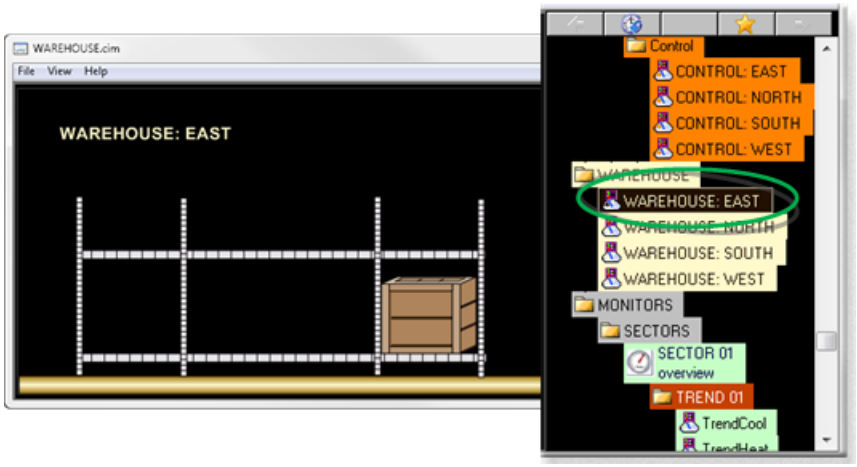


Runtime: Variables Example

As each node is selected during runtime, the variable values for that node are passed to the one CimView screen that is used for each node (Warehouse.cim).

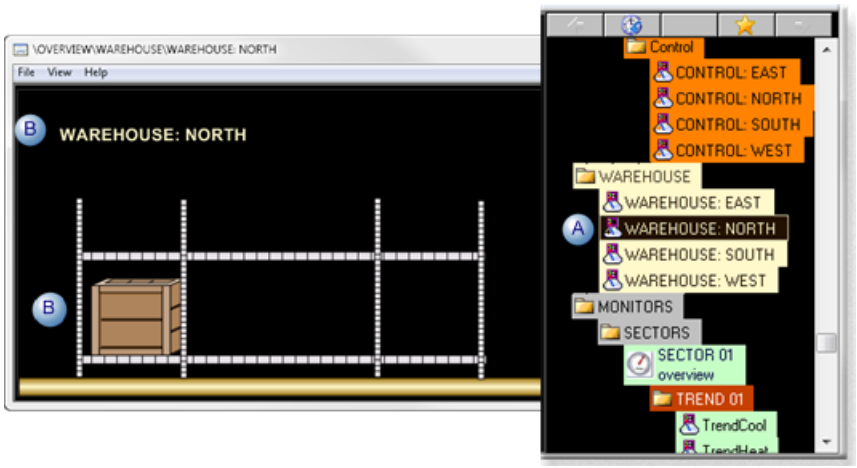
Values are passed to the screen for two of the nodes, as follows.

WAREHOUSE: EAST



A	An operator selects WAREHOUSE: EAST on the Navigation tree.	
B	The variable values, for the WAREHOUSE: EAST node, are passed to the Warehouse.cim screen.	
	Variable	Runtime Value
	WAREHOUSE	WAREHOUSE: EAST
	VERTICALE	0
	HORIZONTALE	238

WAREHOUSE: NORTH



A	An operator selects WAREHOUSE: NORTH on the Navigation tree.	
B	The variable values, for the WAREHOUSE: NORTH node, are passed to the Warehouse.cim screen.	
	Variable	Runtime Value
	WAREHOUSE	WAREHOUSE: NORTH
	VERTICALN	0

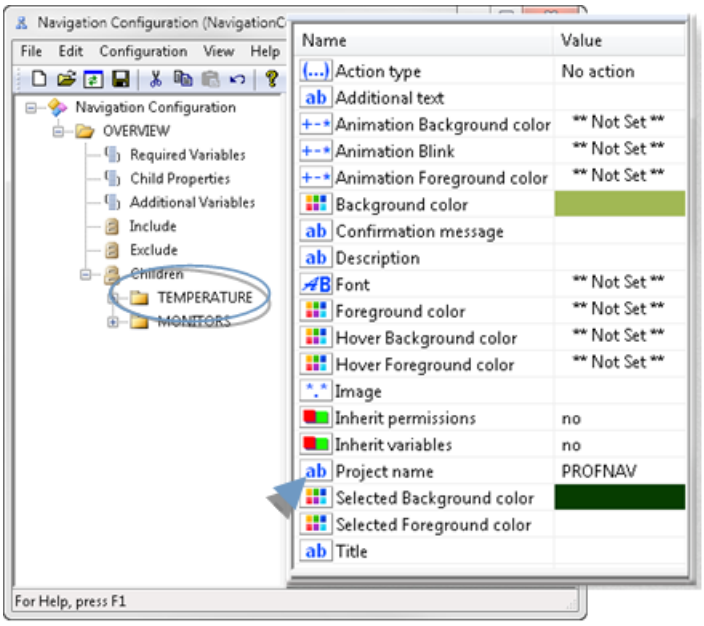
	HORIZONTALN	0

6.6. Configuration Object: Project Name to Qualify Points

(All action types)

You can specify a project to fully qualify unqualified *(page 220)* point ID's for the selected node.

Enter a project name in the **Project Name** field.



If a point on the CimEdit screen is qualified on the screen and the specified project is different from the project name entered in the Project name field, the project name in the CimEdit field overrides the **Project name** entry.

6.7. Configuration Object: Screen Name

(Overlay action only)

A screen name is required when an Overlay action is selected.

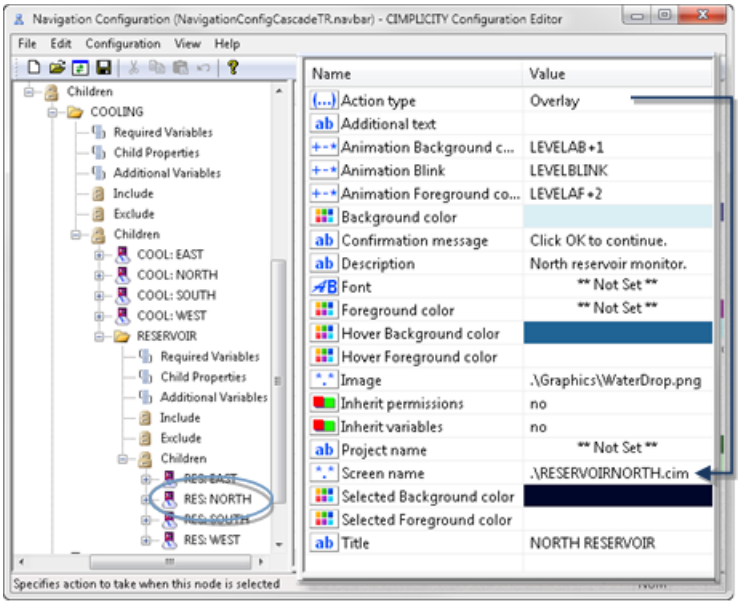
Note:

- If you dragged a screen to create the node or selected the Overlay action, the screen name is already entered.

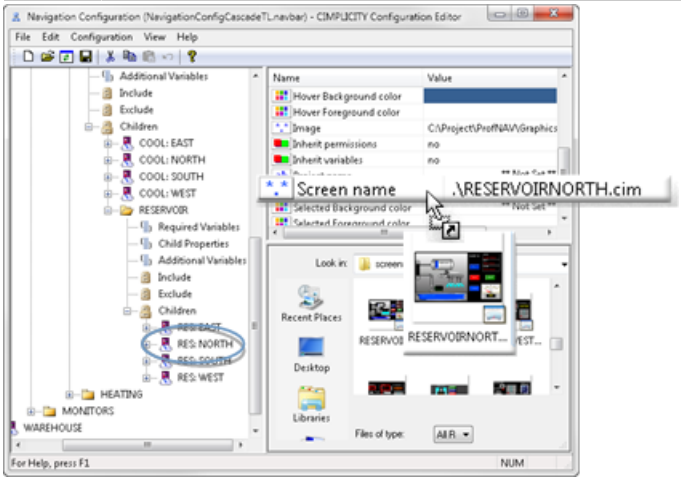
If you had selected No action or Script as the action type and change it to an Overlay action, a screen must be selected.

- If a Script entry point was entered for a Script action, the Navigation Configuration Editor hides it, but does not lose it. If you select a Script action again, the hidden Script entry point will display.

Methods to assign a screen to an Overlay node include the following.



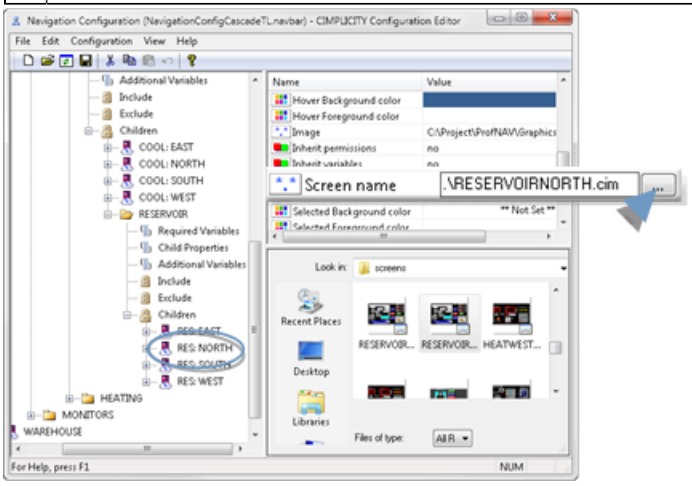
A *(page 717)*	Drag a screen icon to the Navigation Configuration Editor left pane to create a new node.
B	Drag a screen icon in the Navigation Editor lower right-pane to the configuration object's **Screen name** field.



Result: The Navigation Configuration Editor:

- Enters the selected screen filename in the Screen name field.
- Automatically enters the path type, absolute or relative, based on what has been selected.

C	Click the button to the right of the **Screen name** field; select a screen in the Open dialog box.



Result: The Navigation Configuration Editor:

- Enters the selected screen filename in the Screen name field.
- Automatically enters the path type, absolute or relative, based on what has been selected.

D	Enter a Screen name and path in the **Screen name** field.

1. Double-click the **Screen name** field.

   The field is enabled for entry.

2. Enter the screen name with its static or absolute path.

   Note: The Navigation Configuration Editor will do the following with the path type that has been selected if it is different from the type you entered.

Path type Entered	Path Type Selected	Configuration Editor Action
Relative	Absolute	Displays an error message that the: "...\<screen name> not found for overlay action in node "<node name>."
Absolute	Relative	Changes the absolute path to a relative path.

6.8. Configuration Object: Script Entry Point

(Script action only)

Important:

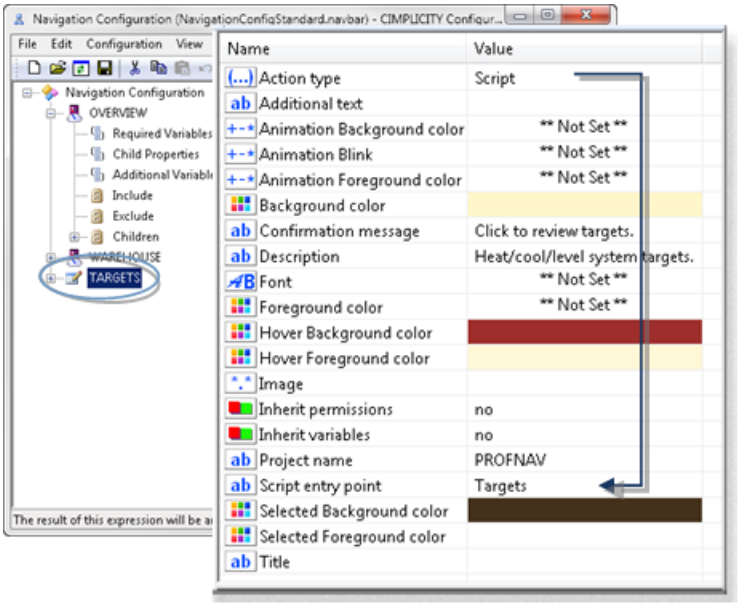
- Nodes that have a script action cannot have children.
- The script that is assigned to the node will be triggered when a user clicks the Navigation button or menu item during runtime.
- If the bar style will be a Tree style, script node cannot be a Quick Launch button; it must be located in the tree.

Enter the script entry point in the **Script entry point** field.

- Configuration: Script entry point.

- Runtime examples: Script entry point.

Configuration: Script Entry Point



Runtime Examples: Script Entry Point

A Quick Launch button has been assigned a script action when the button is clicked it will trigger the script procedure that displays message boxes with selected system targets.

A script item has the following values.

Configuration	Values
Button/Menu name	TARGETS

Configuration	Values
Confirmation message	Click to review targets
Script entry point	Targets

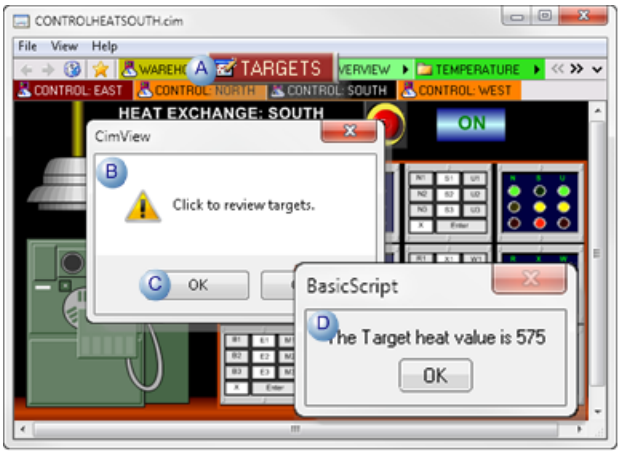
The action is included in two different bar styles, as follows.

- Standard bar style
- Tree bar style

Standard Bar Style

A TARGETS Quick Launch button is configured for a Standard bar style.

During runtime, the following occurs.

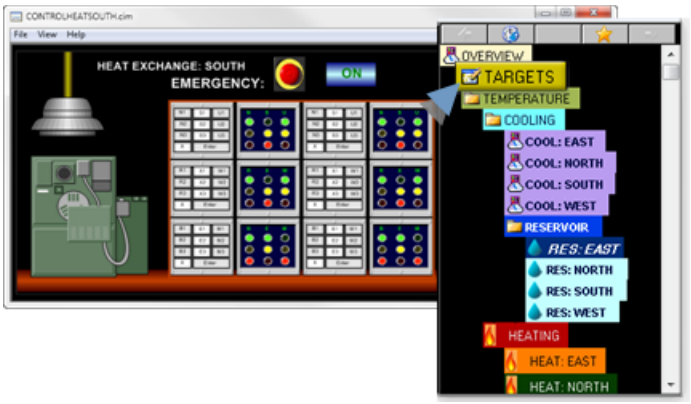


A	An operator clicks the button, TARGETS.
B	A message box that displays the confirmation message opens.
C	The operator clicks OK.
D	The script procedure with the Targets entry point is triggered.

Tree Bar Style

The TARGETS menu item with the Targets script action is added as a child in a Tree Navigation bar.

During runtime the item displays in the configured position; the action is the same as it is for a Navigation button.



7. Navigation: Runtime Tools

7. Navigation: Runtime Tools

7.1 (page 758)	Navigation runtime: History.
7.2 (page 759)	Navigation runtime: Tree view.
7.3 (page 760)	Navigation runtime: Favorites.
7.4 (page 763)	Navigation Runtime: Previous, Forward and Shift buttons.
7.5 (page 763)	Navigation runtime: Display/hide the Navigation bar.

7.1. Navigation Runtime: History

CIMPLICITY Navigation keeps track of the last screens that were open, in the order they were open.

Click the History button on the Navigation bar.

A History list opens.

Click any screen in the list to immediately overlay the current screen.

guide: Guidelines

- The History list:
- Is started when the CimView session starts.
- Can contain a maximum of 100 entries

When 101 is reached, that screen is added to the end of the list and the first screen is removed from the list.

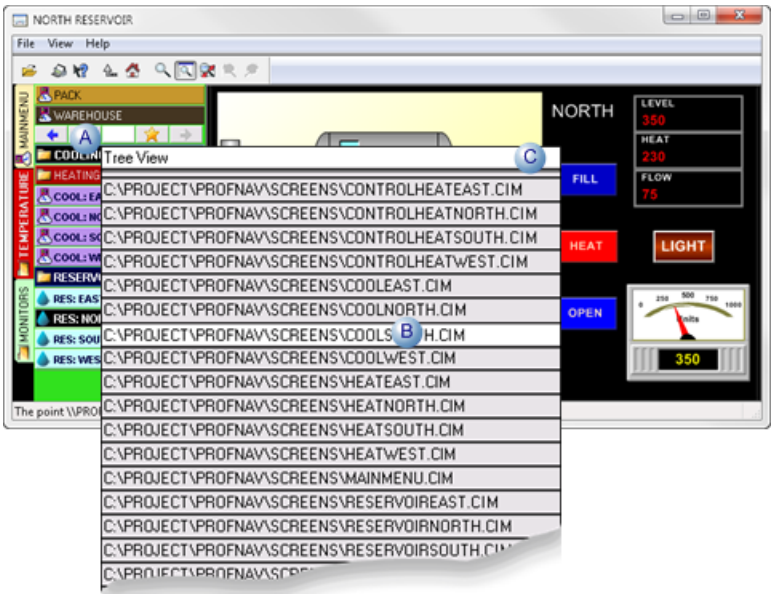
- A screen is added to the list each time it is opened; there can be duplicate entries on the list.
- Buttons with no action or script action are not added to the list, even if they are clicked during the session.

7.2. Navigation Runtime: Tree View

(Cascade and Tree styles only)

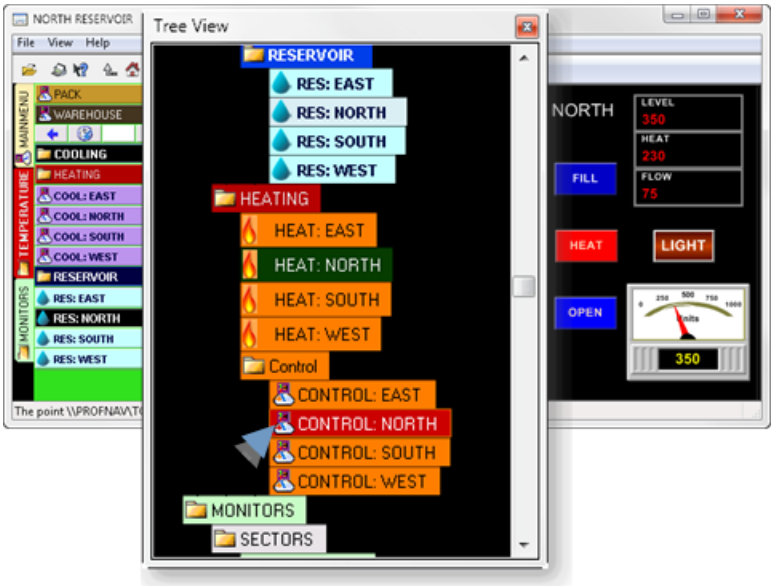
The Navigation bar can display a Tree View window.

Do the following.



A	Click the Tree View button on the Navigation bar. A list of the CimView screens that are included in the Navigation bar opens.
B	(Optional) Click any screen in the list to overlay the current screen with the selected screen.
C	Click the Tree View title bar on the CimView screen list.

Result: A Tree View window opens that displays the entire Navigation Tree.



Note: Click any node to overlay the current screen with the clicked screen.

7.3. Navigation Runtime: Favorites

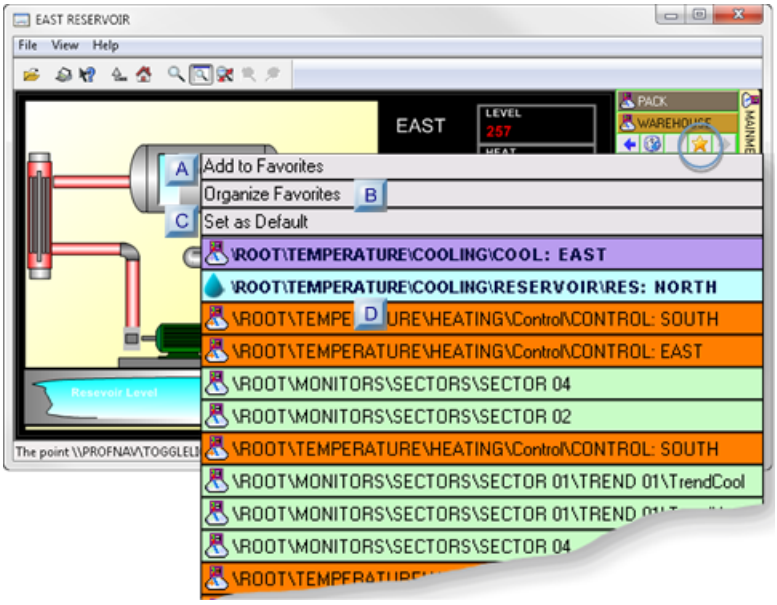
In addition to the Quick Launch buttons, the Favorites tool enables a runtime user to quickly access frequently used screens. This can be particularly valuable in systems with extensive menus.

A major difference between Quick Launch and the Favorites menu is that an applications engineer configures the Quick Launch buttons when you design the Navigation bar. A runtime user adds or deletes screens to or from the Favorites menu during runtime.

Click the Favorites button on the CimView Navigation bar.

The Favorites menu opens.

Favorites options are as follows.



A *(page 761)*	Add to Favorites
B *(page 762)*	Organize Favorites
C *(page 762)*	Set as Default
D *(page 763)*	Favorites list

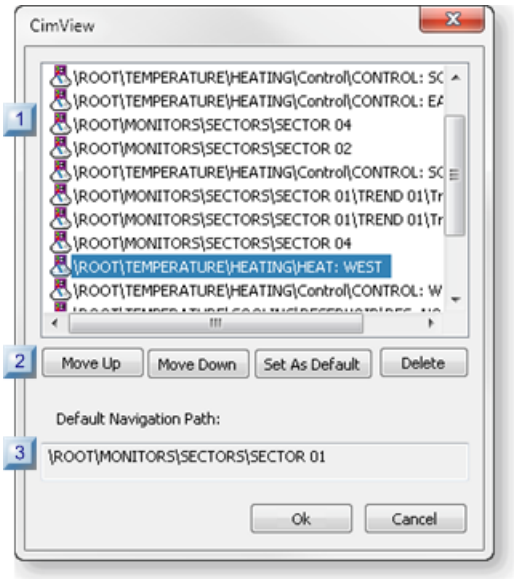
1. Add to Favorites

1. Overlay a screen to which you want quick access.
2. Click the Favorites button on the CimView Navigation bar.
3. Select Add to Favorites on the Favorites menu.

   Result: The selected screen is added to the Favorites list.

   a. Organize Favorites
4. Select Organize Favorites on the Favorites menu.

   A CimView (Organize Favorites) dialog box opens.
5. Organize favorites as follows.



1	Select a screen in the Favorites list.		
2	Click one of the following buttons to perform the required action.		
	Move Up	Move Up	Move the selected screen up one level in the list.
	Move Down	Move Down	Move the selected screen down one level in the list.
	Set As Default	Set as default	Make the selected screen the default.
	Delete	Delete	Remove the selected screen from the Favorites list.
3	(Read-only) When a screen is set as the default, its path displays in the **Default Navigation Path** field.		

   a. Set as default

6. Overlay the screen that should be the default screen.

7. Select Set as default on the Favorites menu.

   Note: The path for the new default screen will be entered in or replace a previous default path in the CimView (Favorites) dialog box **Default Navigation Path** field.

   a. Favorites List

   The screens that have been selected as favorites are listed on the Favorites menu.

   Click any screen in the list to quickly overlay it.

   guide: Guidelines

   - There can be a maximum of 100 favorites in the list.
   - If the open screen is in the Favorites list, the Favorites menu will display a checkmark at its left.

7.4. Navigation Runtime: Previous, Forward and Shift Buttons

1. Click the Previous button to overlay the previous screen that was open.

2. Continue to click the button to go back through all the screens that were previously open until you find the screen you want.
   Note: The button stops at the first screen that was open.

3. Click the Forward button after using the Previous button to move forward through the previous screens.

4. Continue to click the button until you find the screen you want.
   Note: The button stops when the screen that was the starting point is open again.

7.5. Navigation Runtime: Display/Hide the Navigation Bar

Note: You, an application engineer, can configure the Navigation bar(s) to display or be hidden, by default, during runtime.

- CimView Options dialog box.
- Command line arguments.

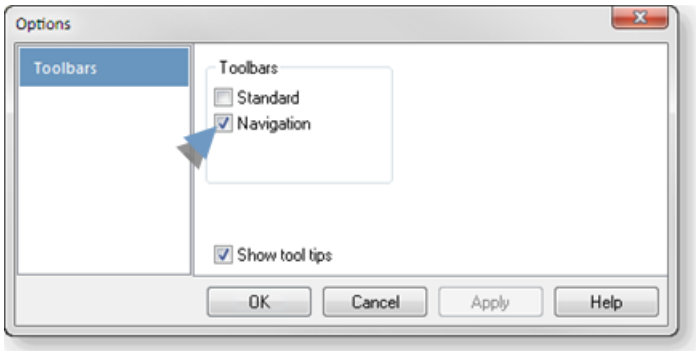
CimView Options Dialog Box

A runtime user can also choose to display or hide the Navigation bar(s) for screens that are included in a selected navigation configuration *(page 684)* , as follows.

1. Click View>Toolbars on the CimView screen menu bar.

   An Options dialog box opens.

2. Do one of the following.



Check	The Navigation Bar:
No	Does not display in CimView.
Yes	Displays in CimView

Command Line Arguments

The following command line arguments will open a new CimView window whose display overrides selections in the Options dialog box.

- /HideNavBar <name>
- /HideNavChildBar <name>
- /LoadNav <name>
- /NavNode <name><nodename>

Technical Reference

CimView Screen Navigation Error Messages

Following are error messages that will display when the error is detected during screen navigation configuration validation.

Error: node <node name> **overlay action has no screen associated with it.**	
Problem	An Overlay action type was selected, which requires a screen file name.

Solution	Select a .cim file in the **Screen Name** field. **Note:** The path, which is also required, will be absolute or static, depending on which you selected on the Configuration menu.
Error: node <node name> **script action does not have an entry point.**	
Problem	A Script action type was selected, which requires an entry point..
Solution	Enter the entry point in the **Script entry point** field, which displays when Script is selected as the action type.
Error: Cannot complete validation.	
Problem	This is a catch all error. If it appears, it means that the configuration file is corrupt.
Solution	The safest approach is to create a new navigation file.
Error: root node <node name> **has children, but only the first node at the root level can have children.**	
Problem	You have attempted to add a child to a Quick launch node, which cannot have children. Quick launch nodes are available to launch a selected screen quickly. That is their only purpose.
Solution	Add children to the primary node and its descendents to create the actual navigation menu.
Error: root node <node name> **has no action.**	
Problem	A No action action type was selected for either the primary root node (first node in the navigation menu tree) or one of the quick launch nodes. These nodes must have an action.
Solution	Select either Overlay or Script in the **Action type** field for each of these nodes.
Warning: cannot construct static path from configuration file <static path><Navigation file name> **to screen**<static path><screen file name> **in node** <node name>.	
Problem	The static path type requires the configuration (.navbar) file and screen (.cim) files to be on the same drive. If they are not a warning displays that the static path cannot be constructed.
Solution	Make sure the configuration and screen files are on the same drive before finalizing the navigation configuration.
Warning: screen <file absolute or static path><file name> **not found for overlay action in node** <node name>.	
Problem	The .cim file path and name that were entered for an Overlay action type could not be found during validation. Navigation issues a warning, not an error, because the navigation structure may be created before the screens are created.
Solution	Make sure the screens are located in the specified location before finalizing the navigation configuration.
Info: Validation succeeded.	
Description	This message displays only when the validation is successfully started and completed by clicking Configuration>Validate on the CIMPLICITY Configuration Editor menu bar. **Note:** Other instances when validation is performed are when the file is being: • Opened • Saved In those cases only errors and warnings are displayed.

Chapter 4. CimView

About CimView

CimView is a runtime, interactive graphical user interface through which you can monitor and control your facility. CimView displays screens that were created in CimEdit for specific applications.

As a result, the appearance, purpose and interactive capability of the CimView screen being viewed is the result of a system administrator, applications engineer and/or screen designer's planning and configuration. (This document references these individuals as system administrators.)

The system administrator also has control over whether or not an operator can access several of CimView's available functions. Therefore, the features that are described in this document may not be available on your CimView screens.

Example

CimView has a feature that, when functioning, enables a user to find and open a selected screen. An applications engineer may design a CimView screen that runs when a viewer boots up. The CimView screen contains buttons that enable an operator to open pre-determined screens. However, the system administrator disables the CimView File>Open capability. As a result an operator cannot open any other screens.

Note: CIMPLICITY no longer supports WebView, but it still supports broadcast sessions. The server, and service name have been changed to CIMBroadcastServer.exe and CIMPLICITY Broadcast Service (CIMBroadcastSerivce.exe). These are now located in copyfiles.bat but the installer needs to register the service as a service.

CimView features include the following:

- Typical Windows application tasks.
- CIMPLICITY runtime features in CimView

CimView Opened

CimView Opened

- Open CimView on a configuration server or viewer.

- Open CimView on a runtime server or viewer.

Note: If you have not logged into CIMPLICITY a CIMPLICITY® Login dialog box opens when you try to open a CimView screen that contains points. A screen will open. However, it will be not display point values, e.g. objects that include fill animations based on point values will be black. When you enter the correct User ID and password you will be able to use the runtime features that can be accessed by the role assigned to your user definition.

Open CimView on a Configuration Server or Viewer

1. Select **Project>Screens** in the Workbench left pane.
2. Select a screen in the Workbench right pane.
3. Do one of the following.

A	Click Edit>Properties on the Workbench menu bar.	
B	Click the Properties button on the Workbench toolbar.	
C	In the Workbench left pane: a. Right-click **Screens**. b. Select Properties on the Popup menu.	
D	In the Workbench right pane:	
	Either	Or
	Double click a screen.	a. Right-click a screen. b. Select Properties on the Popup menu.
E	Press Alt+Enter on the keyboard.	

Result: If the CIMPLICITY project is running, CimView displays the runtime version of the selected screen.

If the CIMPLICITY project is not running, a Select CIMPLICITY® Project dialog box opens. CimView will display the runtime version of the selected screen after you start the project for which the screen is configured.

Note: You can also open CimView through the Windows Start menu.

4. Right-click **Screens**.
5. Select Properties on the Popup menu.
6. Right-click a screen.
7. Select Properties on the Popup menu.
8. Click **Start** on the Windows task bar.
9. Select (All) Programs>Proficy HMI SCADA - CIMPLICITY version>CimView.
10. Select the **.cim** screen from the Open dialog box.

Open CimView on a Runtime Server or Viewer

The method an operator uses to open a CimView screen on a runtime server or viewer depends on the system administrator's configuration. The system administrator has full control over what CimView opens and how it opens.

The most common methods are:

- A pre-determined CimView screen starts when the runtime server or viewer is booted.
- An operator clicks a shortcut icon on the Windows desktop to open the screen for which the shortcut is configured.

Typical Windows Application Tasks Available in CimView

Typical Windows Application Tasks Available in CimView

The following typical Windows application tasks are available in CimView.

1 *(page 769)*	Open other CimView screens through an open CimView screen.

2 *(page 770)*	Flush cached screens.
3 *(page 770)*	Distribute copies of the screen or screen file.
4 *(page 771)*	Select whether or not to display the toolbar, menu and/or status bar.
5 *(page 774)*	Go back to the previous screen or an Overview (home) screen, if there is one.
6 *(page 774)*	Change the size of (zoom) the CimView screen and window that you are viewing.

Note: The features and behavior described are the default settings. You system may vary due to customized configuration.

1. Open Other CimView Screens

The method you use to open another CimView screen determines whether it will overlay the current screen or open in a new window.

Overlay the current screen

1. Do one of the following:
   - Click File>Open on the CimView menu bar.
   - Press **Ctrl+O** on the keyboard.

   An Open dialog box opens.

2. Select a CimView screen.

   Result: The selected screen overlays the current CimView screen in the open window.

   Open a CimView screen in a new window

   Do any of the following.

   Use the Open dialog box.

3. Do either of the following to open the Open dialog box.
   - Click the **Open** button on the CimView toolbar, or
   - Click File>Open Window on the CimView menu bar.

An Open dialog box opens;

4. Select a CimView screen.

   Select a recently used screen.

5. Click File on the CimView menu bar.

6. Select a screen from the list or recently opened screens.

The selected screen opens in a new window. The current CimView screen remains open.

Note: The CimView screen may also have objects designed to open other screens. Whether the screen overlays the current screen or appears in a separate window depends on the configuration that was done in CimEdit.

2. Flush Cached Screens

The screen cache lets CimView retrieve the most recently replaced screens from memory rather than a disk file. Flushing the cache requires CimView to retrieve these screens from their disk files rather than from memory. This is useful if a screen has been updated recently and you want to make sure you are looking at the most recent version.

Click File>Flush Cached Screens on the CimView menu bar.

3. Distribute Copies of CimView Screens and Files

You can distribute copies of CimView screens and files by:

- Print paper copies.
- Send a copy of the screen file to someone via email.

Print paper copies

Do any of the following to print a copy of a running CimView screen.

- Click the Print button on the CimView toolbar.
- Click File>Print on the CimView menu bar.
- Press Ctrl+P on the keyboard.

Result: The printed copy reflects the CimView display as it appeared when the print command was given.

Note: Select Print Setup on the File menu to configure the printer.

Email CimView screens

1. Click File>Send on the CimView menu bar.

   If an email application is open and accessible, an email window from your email application opens with a copy of the open CimView screen attached.

   If no email application is open or accessible, a Choose Profile dialog box opens. When you select the correct application, an email window opens with a copy of the open CimView screen attached.

2. Enter the email address and message to the recipient as you would when sending any other email.

   Result: A copy of the file that displays is sent to the recipient. The file may be a **CimView.cim** or **CimView.cimrt** (runtime only) file.

   Tip: You can email a captured picture of the current screen contents by using the Windows Print Screen tools.

3. Press Alt+Print Screen on the keyboard while the CimView screen is the active window.

4. Display the email message in which you want to paste the picture of the screen.

5. Press Ctrl+V (or use one of the other paste methods for your email application).

A copy of the display at the time you pressed Alt+Print Screen appears on the email message ready to be sent.

4. Display the Toolbar, Menu and/or Status Bar

If your CimView screen is designed so that you can control the display of the CimView toolbar, menu and/or status bar, CimView provides you with a few methods for doing this.

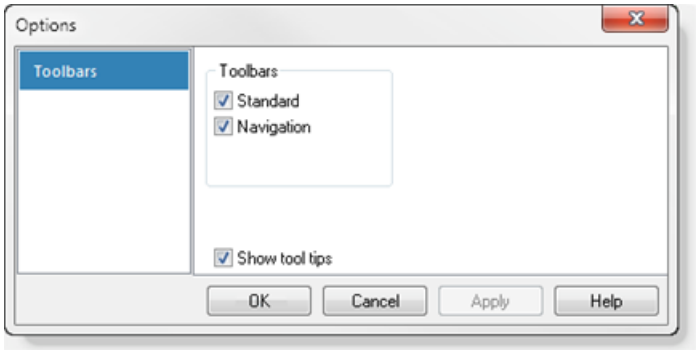
- Toolbar
- Status bar
- Menu/Title Bar
- Example

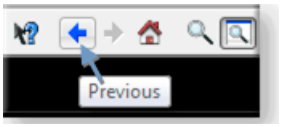
Toolbar

1. Click View on the CimView menu bar.
2. Select Toolbars.

   An Options dialog box opens.

3. Select your preferences as follows.



Checkbox	Description	
Standard	Check	Standard toolbar displays.
	Clear	No toolbar displays for the CimView screen.
Navigation *(page 758)*	Check	Navigation bar displays in CimView
	Clear	The Navigation bar does not display in CimView.
	Note: If your system does not have a Navigation toolbar, this option is disabled.	
Show tool tips	Check	Tool tips display when the mouse hovers over a toolbar button. 

   Status Bar

4. Click View on the CimView menu bar.
5. Check Status bar.

   Menu/Title Bar

   Do one of the following.

   - Click View on the CimView menu bar; check Menu/Title Bar.
   - Right-click the CimView screen; check Menu/Title Bar on the Popup menu.

Result: All of the features that you checked display in CimView. The features you cleared do not display.

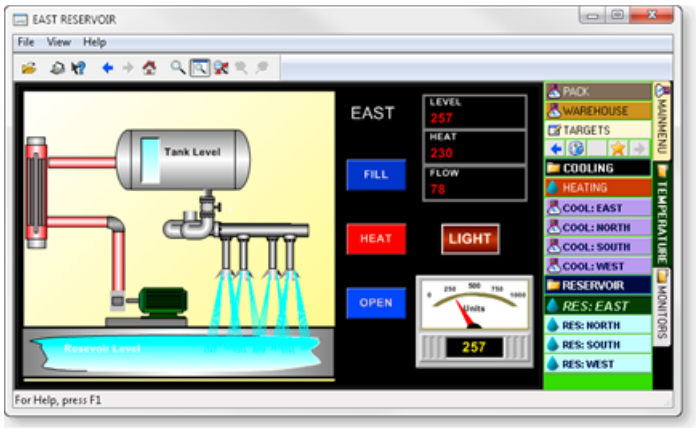
Tip: You can position the toolbar anywhere you want on the CimView screen.

6. Click and hold down the right-mouse button.
7. Grab the toolbar along the edge.
8. Drag it to another location.

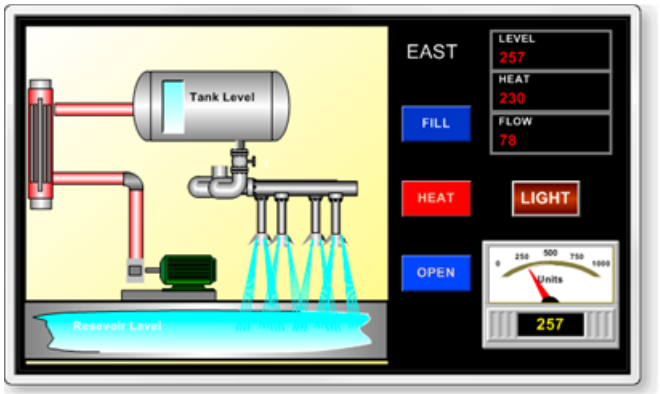
Note: The Status bar provides information about objects and values. The Title bar displays the screen file name.

Example: CimView with and Without Toolbars, Menu and Status Bar

Following is a CimView screen with a title bar, menu, toolbar, status bar and Navigation bar.



Following is a CimView screen with no title bar, menu, toolbar or status bar.



5. Go Back to Other CimView Screens

If your CimView session includes overlaying CimView screens you can use CimView tools to do the following.

CimView Toolbar

Click one of the following

Selection	Goes to the
A	Previous screen, the last screen that was overlaid before the current screen.
B	Next screen, the screen that was overlaid directly after the current screen when navigating back and forward in the screen series.
C	First screen that was opened.

Result: The screen you selected displays.

CimView Menu Bar

1. Click View on the CimView menu bar.
2. Select one of the following:

Selection	Goes to the
Previous Screen	Last screen that was opened before the current screen.
Next Screen	Next screen, the screen that was overlaid directly after the current screen when navigating back and forward in the screen series.
Home Screen	First screen that was opened.

The screen you selected displays.

6. Zoom and Fit a CimView Screen or Window

If this feature is available on your CimView screen, you can change the size of the screen and window to fit your viewing needs.

CimView offers you several methods to magnify or reduce your CimView display size during configuration.

- Zoom a CimView screen.
- Fit the CimView screen to the window size.

Note:

- These methods are for display in CimView. To change the display of the CimView screen, enter the screen dimensions on the Geometry tab of the Properties dialog box.
- When the CimView screen display is magnified hold down the mouse middle button (wheel) and move the mouse in any direction to scroll (pan) the screen in that direction.

Zoom a CimView Screen

1. Do one of the following to change a CimView screen display size during configuration.

   Method 1. Use CimView toolbar buttons

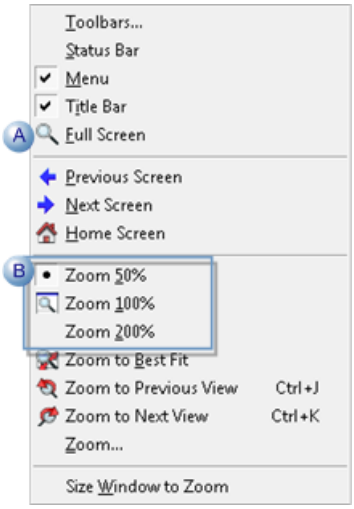
   CimView provides the following screen size buttons.

A	Full screen	Uses the entire monitor screen to display the CimView screen. Press the Esc key to return to a Window view.
B	Zoom 100	Zooms the screen to the size it in which it was saved in CimEdit.
C	Zoom to fit	Expands or contracts the screen expands to fit the workspace.
D	Previous Zoom	Returns the screen to the previous zoom size. **Note:** At least one zoom is required to enable the Previous Zoom button.
E	Next Zoom	Returns to screen to the zoom size before Previous Zoom was clicked. **Note:** The screen has to be in at least one previous zoom size to enable the Next Zoom button.

   CimView View Menu: Quick Zoom

2. Click View on the CimView menu bar.

3. Select any of the following.



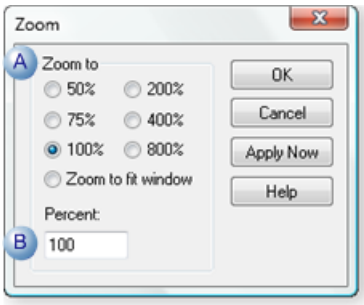
A	Full Screen uses the entire monitor screen for display.
B	Choices for the percent you want the display to zoom to.

CimView View Menu: Precise Zoom

4. Click View on the menu bar.
5. Select Zoom.

   The Zoom dialog box opens.

6. Select any of the following.



1	Zoom to	Choices for the percent you want the display to zoom to.
2	Zoom to fit window	The screen expands or contracts to fit the workspace.
3	Percent	Precise percent the display will zoom to.

Additional Methods

- Press the Ctrl key on the keyboard and do any of the following.
- Click mouse-buttons, as follows.

Button	Result
Right	Enlarges the image by 5% increments.
Left	Decreases the image by 5% increments.

- Use the mouse scroll wheel to zoom in and out.
- Select an area to zoom, as follows.

a. Hold down the left-mouse button to drag the cursor around an area

A box surrounds the selected area.

a. Release the mouse button.

The screen zooms to maximize the selected area display size.

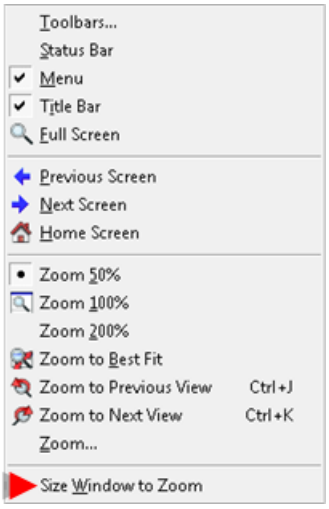
- Use keyboard keys only, as follows.

Note: Keyboard zooms include both location and percent.

Keys	Result
Ctrl+K	Zooms to the next zoom in the zoom stack. Ctrl+K functionality is only available if you have already used Ctrl+J or one of the buttons on the CimView toolbar.
Ctrl+J	Zooms to the previous zoom.

Size Window to Zoom

Click View>Size Window to Zoom on the CimView menu bar.



The CimView window will expand or contract to fit the workspace display size.

Fit the CimView screen to the window size

If you change the CimView window size, e.g. make the entire window smaller or larger, you can increase or decrease the size of the CimView screen (in proportion) to display a best fit in the window.

In a best fit, the screen maintains the width and height proportion it was assigned during configuration.

Do any of the following to fit a CimView screen to the window size.

- Click the Zoom to fit button on the toolbar
- Check the Zoom to fit window in the Zoom *(page 776)* dialog box.
- Click View>Zoom to best fit on the CimEdit menu bar.

The CimView screen will expand or contract in proportion to display the best fit in the window.

CIMPLICITY Runtime Features in CimView

CIMPLICITY Runtime Features in CimView

You can review exact details about points that display through an open CimView screen or a trend for one or more of the included points. You do this by opening through CimView CIMPLICITY's.

- Runtime features.
- ActiveX objects embedded in CimView

Runtime Features

- Open the Point Control Panel through CimView.
- Open Quick Trends through CimView.
- Use Expression View in CimView
- Use Point View in CimView.
- Review screen object properties during runtime.

Note: The role assigned to your user ID must have authorization in order for you to

- Display the Point Control Panel's right-click Popup menu.

- Open the Point Control Panel or Quick Trends .

ActiveX Objects embedded in CimView

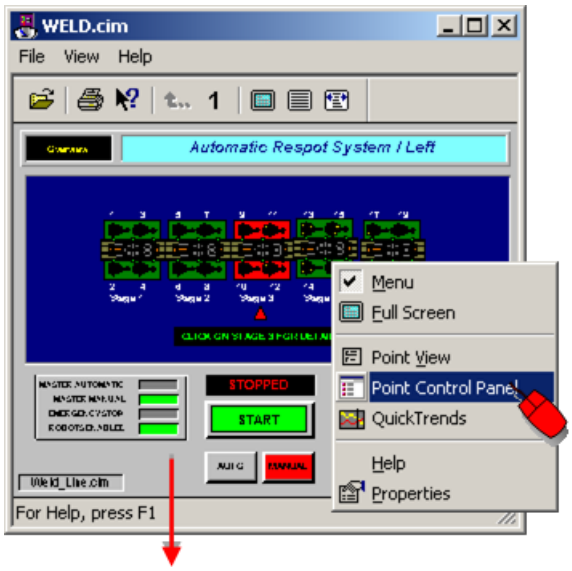
- CIMPLICITY AMV Control
- CIMPLICITY Historical Alarm Viewer
- CIMPLICITY Recipe Control
- CIMPLICITY SPC Control
- CIMPLICITY Trend Control
- CIMPLICITY XY Plot Control
- Proficy Portal Control

Open the Point Control Panel through CimView

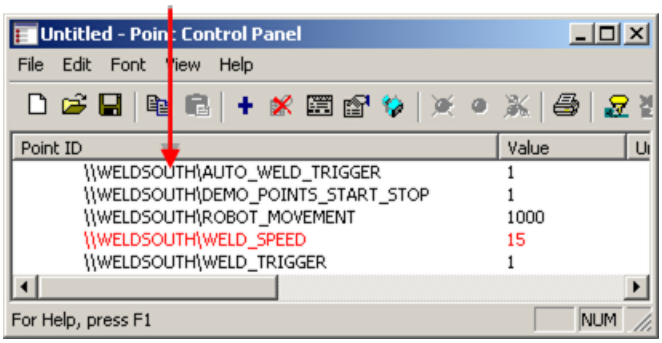
Note: Viewing the Point Control Panel through CimView is available if the role assigned to your user ID has authorization .

The Point Control Panel provides you with a forum in which you can easily review and change point values and status during runtime.

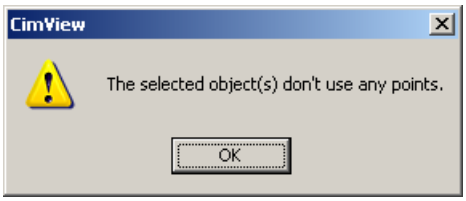
1. Right-click the CimView screen or an object.
2. Select Point Control Panel on the Popup menu.



The Point Control Panel opens displaying the points that are configured for the CimView screen or the selected object.



Note: If there are no points associated with the selected object, a message informs you. The Point Control Panel does not open.

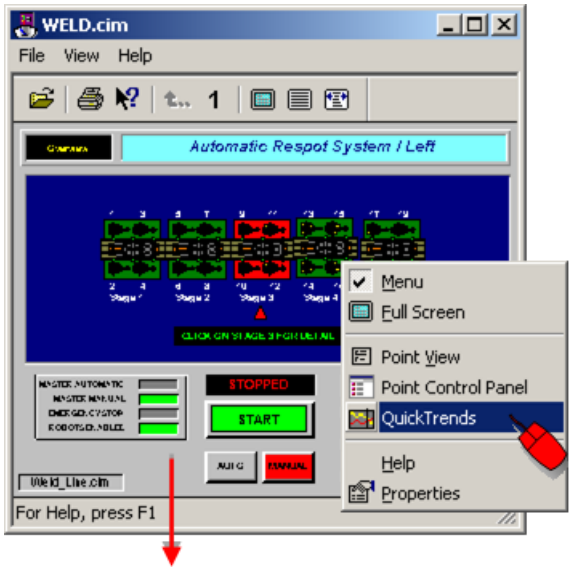


Open Quick Trends through CimView

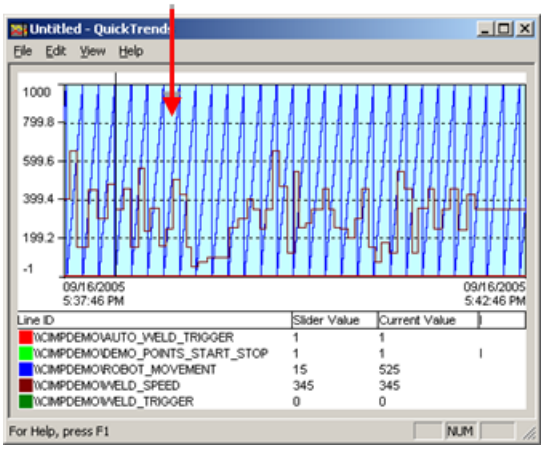
Note: Viewing Quick Trends through CimView is available if the role assigned to your user ID has authorization .

Quick Trends provides you with the ability to instantly review a selected point's trend.

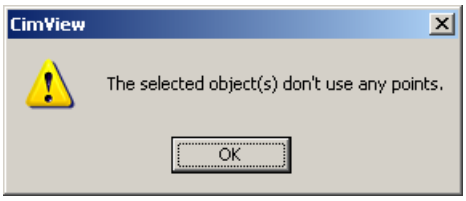
1. Right-click the CimView screen or an object on the screen.
2. Select Quick Trends on the Popup menu.



A trend chart opens. A trend for the points that are configured for the current CimView screen or selected object begins immediately.



Note: If there are no points associated with the selected object, a message informs you. The Quick Trend chart does not open.



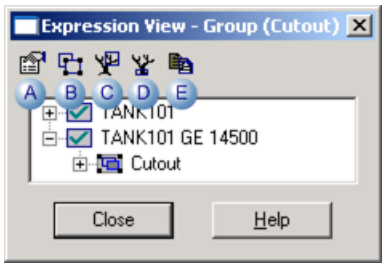
Use Expression View in CimView

Note: Opening Expression View through CimView is available if the role assigned to your user ID has authorization .

CimView users can view what expressions are associated with CimView screen objects through the Expression View window.

1. Right-click any object, group or the CimView screen
2. Select Expression View.

   The Expression View window lists expressions associated with your selection.



A	Display selected expression properties.
B	Select all objects using the selected expression (only available in CimEdit).
C	Show all expressions used by the highlighted object.4
D	Show all expressions used by the parent of the base object.
E	Copy an expression to the clipboard.

Note:

- For details about using Expression View, review Expression View *(page 312)* in CimEdit.
- Configuration capability is available only in CimEdit.

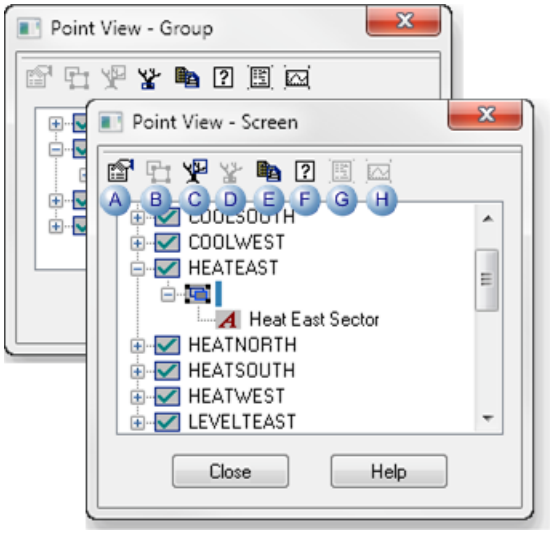
Use Point View in CimView

Note: Opening Point View through CimView is available if the role assigned to your user ID has authorization .

CimView users can view what points are associated with CimView screen objects through the Point View window.

1. Right-click any object, group or the CimView screen
2. Select Point View.

   The Point View window lists points and objects associated with your selection.



A	Display selected point properties.
B	Select all objects using the selected point (only available in CimEdit).
C	Show all points used by the highlighted object.4
D	Show all points used by the parent of the base object.
E	Display the point ID description.
F	Open the Point Control Panel for a selected point.
G	Open a Trend chart for a selected point.

Note:

- For details about using Point View, review Point View in CimEdit *(page 229)* .
- Configuration capability is available only in CimEdit.

Review Screen Object Properties during Runtime

Review Screen Object Properties during Runtime

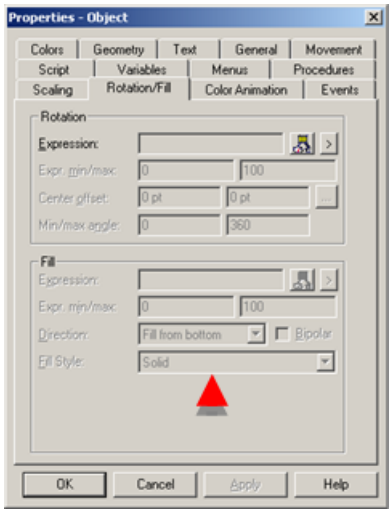
The Object Properties dialog boxes can be available to a CimView user for reviewing the configured properties for a selected CimView object, including the screen object.

Note:

- ActiveX objects may not display the CimEdit Properties dialog box for that object in CimView. The screen or other objects' Properties dialog boxes will be available.
- The Properties dialog boxes in CimView are read-only. Any modifications to the screen must be made in CimEdit.

1. Right-click the selected object on the CimView screen.
2. Select Properties on the Popup menu.

   The Properties dialog box related to your selection displays in read-only format.



3. Select the tab that contains the configuration you want to view.

   Note: You can open some dialog boxes through Properties dialog box, e.g. if a procedure is configured for an object, you can open a Procedure Information dialog box through the Procedures tab. Tabs in these dialog boxes are also in read-only format.

CimView Properties Window

When you display an object's Properties dialog box in CimView you can view the object's configured properties on the read-only tabs.

Detailed information about configuring these tabs is in the CimEdit documentation.

Tabs include:

Tab	Displays Configuration Specifications for or lists of:
Ambient Properties	Screen foreground and background colors.
Angle Animation	Start and sweep angle.
Attribute Animation	Expression animation. Visibility expression. Alarm animation. Color index.
Class Object	Graphic name.
Color Animation	Expression List Attribute Animation: Visibility Animation, Alarm State Animation, and Color Index Animation.
Colors	Static colors, patterns, and style.
Control Properties	Properties of an ActiveX Control
Events	Configured events.
Geometry	Size, Rotation angle, Position, Arc details.
General	Name and description of a screen or object type. Help defined specifically for the screen or object. Object's tab order.
Group Scripts	Global scripts for a screen.
Group	Objects in group object.
Image	Display and animation speed of a graphic object.
Link Container	Linked objects.
Menus	Popup menu items.
Movement	Horizontal and vertical runtime movement.
Pipe Properties	Width and color of the pipe, and shading properties of the elbow and pipe.
Procedures	Configured procedures.
Rotation/Fill	Rotation and fill animation.
Scaling	Horizontal and vertical scaling.
Script	Callable entry points.
Shadow	Style, Distance, Angle, Color.
Text	String/Multiline, Anchor position, Font, Translate, Horizontal alignment, Vertical alignment, Wrap text

Tab	Displays Configuration Specifications for or lists of:
Text Button	Displays when a text button is configured.
Transparency	Expression, Expr. min/max, Min/max value.
Variables	Variables.

Set Points or Variables

Set Points or Variables

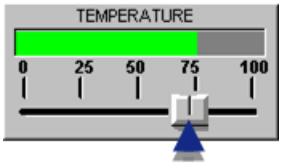
Your CimView screen may provide you with the capability to set point values. If it does following are some guidelines for using the method that is provided on the screen.

Methods include:

- Execute slider setpoints.
- Enter a value for a setpoint.
- Enter a Variable assign value.

Execute Slider Setpoints

A slider setpoint literally enables you to slide an object or dial to the setpoint value.



Use either of the following to select the object.

- Mouse.
- Keyboard.

Note: When you use a Slider setpoint, you may be setting different points in the horizontal and vertical directions.

Setpoint object selected with the mouse

- Perform a slider setpoint with the mouse

1. Move the mouse cursor over the object.
2. Hold down the left mouse button and move the object in the direction (vertical and/or horizontal) you want for the setpoint value.

   As you move the object, the proposed setpoint value will be displayed in the CimView Status Bar message area.

3. Release the left mouse button.

   Result: The new value for the point displays in the Status Bar.

   - Cancel the Slider setpoint action:

4. Move the mouse cursor off the object
5. Do one of the following:
   a. Release the left mouse button, or
   b. Press the **Esc** key and release the button.

   Result: The original value for the point display in the Status Bar.

   Note: If the object is also configured with action when the mouse is clicked down or up, the actions execute as configured.

   Setpoint object selected with the keyboard

   Once you have highlighted an object that has a slider setpoint action, you will use the following arrow keys to perform the action:

Arrow Keys	Action
Up and Down	Vertical
Left and Right	Horizontal

   - Perform a slider setpoint with the keyboard:

6. Press **Enter** on the keyboard to activate the arrow keys for the Slider setpoint action,

   Note: Once the arrow keys are activated for the Slider setpoint action, you will no longer be able to move the object with the mouse or use the left mouse button to select other objects.

7. (Move the object by one pixel.) Press an arrow key. The direction of the key defines the direction of the movement.

8. (Move the object by 10 pixels.) Hold down the **Ctrl** key, then press an arrow key.
9. Press **Enter** to accept the slider action, and perform the setpoint.

The new value for the point displays in the Status Bar.

- Cancel the Slider setpoint action:

Press **Esc** on the keyboard.

The original value for the point display in the Status Bar.

Enter a Value for a Setpoint

A value setpoint enables you to enter a value for a point on the CimView screen.

- Setpoint: Enter Value.
- Setpoint: Cancel Value Entry.
- Example: Setpoint in CimView.

Setpoint: Enter Value

1. Select the object.
2. Do one of the following to activate the object for editing:
   - Press **Enter** on the keyboard or
   - Click the object with the left mouse button.
3. Enter the new value for the setpoint.

   You can use **Ctrl+Z** to undo a single level of editing.

4. Press **Enter** or **Tab** to perform the setpoint.

   Note: If you press **Tab**, the next action object is automatically highlighted.

The new value for the point displays in the Setpoint object.

Setpoint: Cancel Value Entry

Do one of the following.

- Move the cursor away from the object; click the left mouse button.

- Press Esc on the keyboard.

Example: Setpoint in Cinview

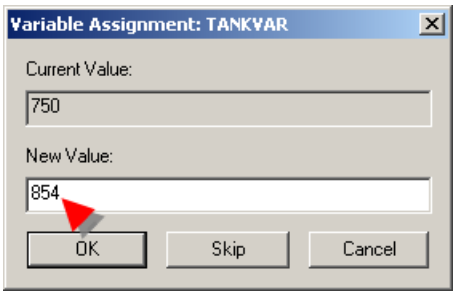
1	Current point value
2	Setpoint being changed.
3	New point value displays

Enter a Variable Assign Value

CimView can include objects with an assigned variable whose value can be changed by an authorized user.

The event that triggers the action to change the variable value depends on the customized configuration, e.g. a Mouse Up *(page 456)* event can trigger a Variable assign *(page 525)* action.

When the Variable assign action is triggered, a Variable Assignment dialog box opens.



Enter a new value in the **New Value** field if the current value needs to be changed.

Help for a Selected Object

1. Click the **Help** button on the CimView toolbar.
2. Click the object for which you want help.

   Result: If the system administrator has created help for that object, it will display.

   CimView Menu Bar

3. Select the object.
4. Click the right-mouse button.
5. Select Help on the Popup menu.

If the system administrator has created help for that object, it will display.

Tip: Selecting Help on the CimView toolbar or pressing F1 displays the CimView help file.

Standard Windows Features

1. Click File>Setup on the CimView menu bar.

   A Print Setup dialog box opens.

2. Enter your printer specifications.

   Note: If you are connected to printers on a network, you can access those printers by clicking the Network button and opening a Connect to Printer dialog box.

3. Click OK to close the Print Setup dialog box and save your specifications.

   Print Screen

4. Do one of the following.
   - Click File>Print on the CimView menu bar.
   - (If the toolbar is visible) click the Print button.

   A Print dialog box opens.

5. (Optional) change printer specifications that were entered in the Print Setup dialog box.
6. Print one or more copies of the screen.

Exit CimView

Do either of the following.

- Click File>Exit on the CimView menu bar.
- Click the Windows **Exit** button .

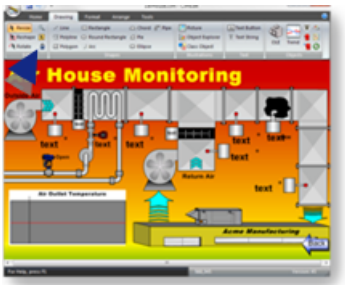
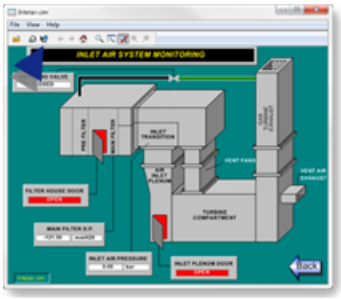
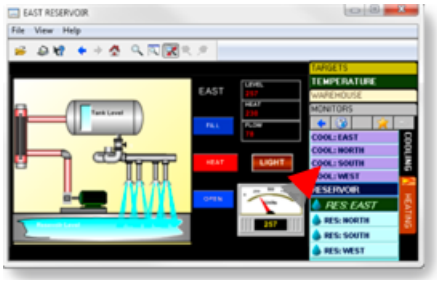
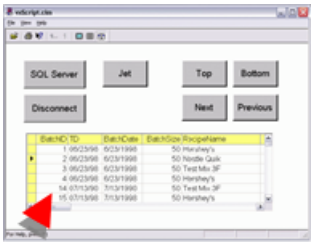
Other Windows Buttons

	Maximize the window. The screen remains the same size, unless the **Zoom to best fit** button is depressed.
	Make the window resizable. The screen remains the same size, unless the **Zoom to best fit** button is depressed.
	Minimize the window.

Chapter 5. About CIMPLICITY Screens

CIMPLICITY provides powerful tools to create runtime user interfaces that are customized for your systems requirements.

The tools include:

Tool	Description
CimEdit	CimEdit, an object oriented graphics editing tool, is configuration half of the CimEdit/CimView duo is easy to learn and easy to use configuration tool that enables you to blend industry standards with advanced interface designs. 
CimView	CimView, an interactive graphical user interface through which you can monitor and control your facility, is the runtime half of the CimEdit/CimView duo. **Note:** If a Navigation bar is configured and selected to display, it displays in all CimView sessions running on the Server. 
Screen Navigation Bar	CimView Navigation includes CimView menus that runtime operators can use to easily find, select and open any screen to which they have access. 

Tool	Description
CimLayout	CimLayout: provides the means to organize more than one CimView screen into a container that opens all of the screens at the same time; the container is organized according to your specifications
Proficy Portal	When the Proficy Real-Time Information Portal control is configured on a CimEdit screen, CimView users can browse for RTIP screens in the Proficy database and view Proficy Portal screens.
VisiconX	VisiconX™ is a GE Intelligent Platforms Plug and Solve® solution for the Proficy iFIX® and CIMPLICITY® products. It is an easy-to-use, object-oriented development tool for accessing data in any type of database. Its simple point-and-click interface provides the means to configure access to any database and then create an SQL query as simple or complex as necessary to retrieve the information you need. The CIMPLICITY Object Explorer provides VisiconX control objects that can be configured on a CimEdit screen and run in CimView.