CIMPLICITY 11

Getting Started

Proprietary Notice

Trademark Notices

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

Getting Started

Chapter 1. HMI/SCADA CIMPLICITY Introduction

About HMI/SCADA CIMPLICITY

This section contains information on customer and technical support, the basic CIMPLICITY architecture, a tutorial describing CIMPLICITY applications, and some information on optimal usage of the CIMPLICITY online help system.

Customer and Technical Support Contact Information

For information on how to contact our Technical Support team, see digitalsupport.ge.com.

System Architecture Overview

System Architecture Overview

CIMPLICITY software is scalable from a Human Machine Interface to a fully networked Supervisory Control and Data Acquisition (SCADA) system. The networking capabilities inherent at all levels within the product line let you achieve levels of integration that virtually eliminate redundant configuration within a network.

rect 3, 2, 273, 88 *(page 4)*
rect 6, 90, 273, 182 *(page 4)*
rect 4, 183, 274, 269 *(page 5)*

V	Viewer	Connects to Server
		Status monitoring and control
		Viewer options available
		Development configuration
		Graphics configuration
S	Server	Connects to Viewer
		Status monitoring and control
		Development configuration
		Graphics configuration
		Data collection

		Server options available
C	Industrial controllers	N/A

CIMPLICITY is based on a client–server architecture consisting of Servers and Viewers. Servers are responsible for the collection and distribution of data. Viewers connect into Servers and have full access to the collected data for viewing and control actions.

Servers and Viewers can be easily networked together to seamlessly share data without the need to replicate your point database from node to node. For example, points are configured once and only once on a server. Screens can be developed and stored in a single location on the network and accessed by any other CIMPLICITY display on the network.

CIMPLICITY provides the flexibility to build a larger system through multiple smaller nodes without forcing you to purchase large and expensive server hardware to service multiple users.

CIMPLICITY Server and Viewer Defined

HMI/SCADA CIMPLICITY provides the following three options. The server or viewer's license determines which option it will use.

Options are:

1	CIMPLICITY Server	Receives data from the PLC.
		Stores data.
		Provides CIMPLICITY configuration tools.
		Performs calculations.
		Displays data through Viewers
		Displays data
2	Viewer	Enables configuration on the server from a separate computer.
		Displays data from the server.
		Displays data from the server.
3	Web Client	N/A

You have to install at least one CIMPLICITY server. The total number of servers and viewers you can install depends on your licensing agreement.

Tip: CIMPLICITY also provides numerous options *(page 11)* for remotely interacting with your CIMPLICITY projects.

Contact your sales representative *(page 3)* with questions about purchase options.

Chapter 2. CIMPLICITY Applications Tour

CIMPLICITY Applications Tour

CIMPLICITY provides an extraordinary selection of features that enable you to configure comprehensive and robust projects.

Once you have installed CIMPLICITY this quick tour will guide you through the order for configuring a basic project.

This tour provides links to the related subject in the documentation. Once you think you understand the basic concepts about the subject, you can come back to the tour at any time.

The tour is divided into five parts that provide links to documentation that describes:

CIMPLICITY APPLICATIONS TOUR	
Part 1 *(page 6)*	How to set up the foundation for your system goals.
Part 2 *(page 7)*	How to set up points and alarms.
Part 3 *(page 8)*	How to create powerful applications that can graphically deal with system data for whoever has access privileges.
Part 4 *(page 9)*	Many other powerful tools.
Part 5 *(page 11)*	CIMPLICITY options.

Part 1. CIMPLICITY Tour

Part 1 of the CIMPLICITY tour provides links to documentation that describes how to set up the foundation for your system goals.

Part 1 of CIMPLICITY Tour	
Step 1.	Open the CIMPLICITY Workbench. The Workbench is at the center of your CIMPLICITY project.
Step 2.	Create a new project. A project contains the configuration that defines what CIMPLICITY will do for your system and how it will work.
Step 3.	Look over the Workbench. The Workbench provides the power you need to view, configure, organize, and manage every component of your project through one easy to use window.

Part 1 of CIMPLICITY Tour	
Step 4.	Configure a device including: • A port, which is a communication socket, connects one or more factory devices such as PLC's to the computer • A device is anything that can communicate point data to CIMPLICITY software. CIMPLICITY software can read data from and write data to devices. Examples of devices are programmable controllers such as the Series 90. The *Quick Device Setup* in the documentation gives you quick start for both.
Step 5.	Define security and routing including: • Resources are the physical or conceptual units that comprise your facility. • A user is an individual person working with a CIMPLICITY project. • A role specifies what privileges its users have when they work in CIMPLICITY.

Part 2. CIMPLICITY Tour

Part 2 of the CIMPLICITY tour provides links to documentation that describes how to set up points and alarms.

Note: CIMPLICITY collects or calculates point data that it distributes to:

- CimView screens
- Alarm Viewer screens
- Alarm printers
- Logging tables
- Other CIMPLICITY software options

The collection and distribution of point data is handled by the Point Management subsystem.

Part 2 of CIMPLICITY Tour	
Step 6.	Create points including: • A device point communicates back and forth with a device that is attached to the server for monitoring and control purposes. • A virtual point provides you with the ability to calculate and report data that is independent of any one device.

Part 2 of CIMPLICITY Tour	
Step 7.	Configure alarms. • Point alarms alert users when points are in a defined alarm states. You create and modify point alarms in the Point Properties dialog box or the Alarm Definition dialog box through the Alarms folder. • System event alarms alert users for alarm states such as device failures, program terminations, system startups, and system shutdowns. You create and modify system event alarms in the Alarm Definition dialog box through the Alarms folder.
Step 8.	Test your configuration in the Point Control Panel. • The Point Control Panel provides you with a forum in which you can easily review and change point values and status during runtime.

Part 3. CIMPLICITY Tour

Part 3 of the CIMPLICITY tour provides links to documentation that describes how to create powerful applications that graphically deal with system data for whoever has access privileges.

Part 3 of CIMPLICITY Tour	
Step 9.	• Configure a CimEdit screen. • 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. • CimEdit Screens provide you with several diverse features and capabilities that you can use at any time during your screen design session. Some, but not all of the capabilities include: ◦ Preliminary Layout CimEdit offers you a wide assortment of objects and object types to place on your CimEdit screen. Consequently, 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. ◦ Inanimate Visual Features enable you to modify the appearance of an object. They range from modifying its size so it will fit where you want it go, to displaying a several similar objects that represent similar but independent functions. ◦ Runtime movement and Animation 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. ◦ Points report specific conditions in the system. Points are the result of detailed configuration, which is done in the Point Properties dialog box. As with other CIMPLICITY applications, when you are in CimEdit, you can find and use any point that is already in any broadcasting project on your network. In addition, you can create new points by opening the Point Properties dialog box through CimEdit. ◦ Variables can be used in an expression to represent different types of values ◦ Events trigger a procedure or call a script. CimEdit provides a long list of events from which you can choose the best one for your requirements. ◦ Procedures contain 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.
Step 10.	Test your configuration through 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.

Part 4. CIMPLICITY More Features

CIMPLICITY is so powerful that you will constantly discover new possible solutions as you continue to use it.

Part 4 provides links to documentation for CIMPLICITY's many other powerful tools. Which tools you use depend on your system needs. (Some of the tools are options that you can purchase through your CIMPLICITY representative.)

Feature	Description
Alarm Management	• Alarm Classes are groups of Alarms with similar characteristics. • Alarm Strings name alarm states. An alarm displays the string for its alarm state when **%State** is included in the alarm message. • The Stand-alone Alarm Viewer, AMV, is useful for a user to quickly monitor and responds to alarms anywhere in the system. • The Alarm Viewer Control is an ActiveX object that you embed in a CimEdit screen. The AMV Control provides a powerful tool for you to fully integrate the Alarm Viewer capability with your other CimEdit screens. • Database Logging provides you with a seamless way to analyze your system processes and equipment performance by logging data to and reporting data from a wide variety of ODBC (Open Database Connectivity)-compliant databases. • Trend Control is an ActiveX Control that enables you to review, evaluate and log point values over time. • Historical Alarm Viewer Control is an ActiveX control through which you can easily review logged alarm data through CimView in an easy-to-read table format and print one or more pages of the display at any time during a session.
Basic Control Engine	The Basic Control Engine option consists of three main components: • Program Editor provides a set of sophisticated development tools that let you create programs with a Visual Basic compliant programming language. These programs can then be executed as actions in response to events. The programming language has a rich set of nearly 500 standard Basic functions, and also provides an object interface to CIMPLICITY points, alarms and the Status Logger, further enriching the language. • Event Editor enables you to define actions to take in response to events that occur in a process. An event can be defined as a changing point, alarm state, or even a particular time of day. One event may invoke multiple actions, or one action may be invoked by many events. • Basic Control Engine monitors for events and executes the configured actions. The Basic Control Engine is based on a multi-threaded design that allows the system to invoke and execute multiple Visual Basic programs concurrently. • Classes enable you to do the basic configuration once and use it over and over instead of repeating configuration, which may include creating complex CimEdit/CimView screens, for several objects that have similar requirements. • Class Objects provide an easy way to do complex configuration for one or more objects that are similar. Class objects, which are based on a Class template, can include pre-configured attributes, points, events, actions and scripts. • Dynamic Graphic Replay is a powerful tool to help you troubleshoot problems that have occurred in your processes. • XY Plot provides you with the ability to visually represent values in relation to each other. For example, you can plot real data vs. calculated date, or elements such pressure vs. temperature. • Remote Projects need to be defined when a project starts, the Point Bridge or Point Data Logger need to get points from projects on other computers running CIMPLICITY projects. • Recipes enable you to create and manage recipe data for your production processes. The Recipes interface consists of a spreadsheet format in which you enter the configuration data for each of your recipes. This format allows you to group similar products together.

Feature	Description
Object Model	Interfaces into components (e.g. objects, services, CimEdit screens both configuration and runtime, Project configuration Trend control, XY Plot control) that enable a developer to manipulate the components from a programming or scripting language, such as CIMPILICITY Basic, VB, C++, VBA, VBScript. • CIMPLICITY Configuration Object Model • CimEdit/CimView Object Model • CIMPLICITY XY Plot Object Model • CIMPLICITY Safe Array Object Model • CIMPLICITY Historical Data Connector Object Model

Part 5. CIMPLICITY Options

Option	Description
Options	• CIMPLICITY OPC Server provides a standards-based interface to some form of run-time data. The data may come from a specific physical device (e.g. a PLC) or from a Distributed Control System. The OPC Server conforms to the OLE for Process Control (OPC) 2.0 Data Access standards, a technology standard initially developed by a group of automation industry companies and now managed by the not-for-profit organization called the OPC Foundation. • Server Redundancy in automated systems, provides for switchover of functionality to a backup component in case of failure of a primary component. The switchover is considered automatic if no operator intervention is required. Redundancy applies to both hardware and software, and implies minimal loss of continuity during the transfer of control between primary (active) and redundant (backup) components. • Statistical Process Control enhances your ability to manage a quality control program by addressing the four major phases of quality control: measurement, analysis, improvement, control.
Tracker	There are two distinct, yet interrelated pieces to Tracker: Production Tracking (PRT) and Routing Control Objects (RCO). • Production Tracking module monitors the progress of items through the production process. • Routing Control Objects performs routing decisions for enhanced production flow.

Option	Description
Order Execution Management	Order Execution Management provides a comprehensive addition to Tracker that enables you to track, store, categorize and sequence your customers' orders based on your configured criteria. Order Execution Management includes: • XMLT tools take raw data orders, translates them into an .xml format and enters valid data into PRT and TADB. • Directory Watcher confirms that order files have completed downloading to XMLT output directory and moves files to the POMS input directory. • Product Order Management System (POMS) can be the hub of your Order Execution Management order management system. POMS is essentially a project that contains the basic configuration on which you can build your customized system. • CimView Order Entry provides order entry screens if you find that you have to manually edit an order item. • Tracker Attribute Database (TADB) stores comprehensive data about items, including orders and product components. • Range Source Architecture (RSA) enhances the traditional RCO concept of a Tracker source (source region). • Tracker Query Engine is a powerful high level query engine that has its own syntax for forming queries. It pulls data from both the Tracker Attribute Database and the Order Execution Management runtime memory map. Queries may be named and stored for future use, or for subdividing and abbreviating complicated queries. • Order Execution Management Broadcast is the delivery of a configurable list of product related information (including at least build options, location information, other/supporting data and subsets of the unit bill of material) to plant floor devices and to suppliers. • Alarm Cast messaging engine is a standardized interface between personal communication devices and applications sending messages through either an internal paging service and/or external service providers. • Marquee Manager product family monitors manufacturing environments and sends real time, automated messages to visual and/or audible devices.

Chapter 3. Common Licensing

About Common Licensing

The family of products provides you with hardware keys that are programmed with licenses for your selected products and options.

You simply:

1. Buy *(page 15)* your products and options.

   Visit the GE Customer Center web site at http://support.ge-ip.com to obtain information about the latest GE product offerings.

   Note: If you visit this web site, click a topic on the HMI/SCADA CIMPLICITY documentation Contents tab to return to the documentation.

   When you receive your DVD you will receive as many hardware keys as you need.

   If you have any questions about licensing call the GE Intelligent Platforms support line *(page 3)* . A GE representative will direct your call to the correct resource.

2. Install your products.

3. Plug one of the keys into each Server that requires one or more licenses.

   You are ready to go.

   You can choose either of two key types.

   - One key can be plugged into a Server's parallel port.
   - The other key can be plugged into a Server's USB port.

   Note: If you need to reboot a computer that requires a license, make sure the key is inserted in the computer before you reboot so any services that start up can read the key.

Installed License Issues

- Hardware key removal.
- License upgrades.

- License expiration.

Hardware key removal

When the hardware key is removed from a Server:

- An Alert Message will display on the first Server, informing you that the hardware key has been removed; please plug it in.

When you click OK to close the message, it will display again in one minute.

The message will continue until the key is re-attached to the Server.

- If a project is running, it will continue running. However, if it is restarted before the key is re-attached, it will revert to demonstration mode.

License upgrades

If you buy additional options after you receive your hardware key(s) you will receive an email with an update .plic license file.

❗ **Important:** Before upgrading your license, shut down all of the applications on the computer you are working on. When you upgrade your licensing, the Licensing Service is automatically stopped and then restarted when the upgrade successfully completes.

1. Save the .plic file to a directory of your choice.
2. Double-click the file.
   An update utility opens.
3. Click Update Now.

The utility validates your current license. If the validation is successful, the utility updates the license; if the validation is not successful, the upgrade is halted.

📝 **Note:** Contact your sales representative for information about obtaining the license file.

License expiration

When you purchase a product, it is licensed without an expiration date.

If a demonstration license is provided to you for you to try the software, it will be provided with an expiration date that will allow the software to run up until that date. You will receive notices on your screen alerting you to when the licenses will expire.

New License Steps

1. Obtain a hardware key for products and options.
   Specify the following that you are ordering:
   - Part number for each product and feature.
   - Quantity of each product and feature.
   - If the hardware should be parallel or USB port.
     - There can be only one hardware key per computer.
     - You can attach a parallel port printer cable to a parallel port hardware key.
   - List of what products/options will be installed on each Server.

   The default hardware key configuration is that each hardware key will contain the set of licenses for every product and option purchased.

   You will receive a hardware key for each set of licenses.

   If you order 4 CIMPLICITY licenses and 3 Historian licenses, you will receive the following:

No. of Keys	Key Contains Licensing for
3	CIMPLICITY and Historian
1	CIMPLICITY

   Each key will contain that products/options according to your specifications.

   If you order 4 CIMPLICITY licenses and 3 Historian licenses and plan to run each on a different server, you will receive the following:

No. of Keys	Key Contains Licensing for
4	CIMPLICITY
3	Historian

   ! **Important:** You can attach only one key to a Server.

   When your order is fulfilled you will receive one or more CDs and hardware keys.

2. Install your products and options.
   Follow the installation directions for each of your products.

3. Attach the hardware key to the server.

Attach one hardware key to either the parallel or USB port, depending on what you ordered.

Type of Key	Attach to a
Parallel port	Parallel port on the Server. You can then attach a parallel printer cable to the hardware key.
USB	USB port on the Server.

Important: Attach the hardware key to the Server only after you install all of the licensed products.

4. Review the license report.

   a. Click Start on the Windows task bar.

   b. Select (All) Programs>Common>License Viewer on the Start menu.
   The M4 License Viewer opens.
   rect 0, 9, 23, 30 *(page 16)*
   rect 19, 78, 40, 99 *(page 16)*
   rect 170, 119, 191, 140 *(page 16)*

License report tools are as follows.

Tools	Description
1	The right pane in the M4 License Viewer displays the: • License key details. • Customer name • Serial number • Key expiration • License creation • License key version License type • License details for the selected product(s), including: • License expiration • Number of licensed points • Number of <option> nodes • Number of <option> connections • Enabled options.
2	The left pane displays icons for each of your licensed products. Click an icon to select it and the details display in the right pane that correspond to the selected product. **Note:** You can click the Show All button on the M4 License Viewer toolbar to display details about all the licensed products on the same page
3	Click any of the buttons on the M4 License Viewer toolbar to do the following: • Show All - Displays all of the details about all of the licensed products on the same page. • Save - Saves. • Prints - Prints the report that displays.

5. Run your project(s) on the licensed server. Start any project on the licensed server.

   Important: If you start a project on a Server that does not have a license a message may warn you that no license is present, and the project will run in demonstration mode. Demonstration mode provides you with limited functionality, including no networking. Demonstration mode runs for a maximum of two hours. The actual time depends on which product is unlicensed.

Update Licensed Options without Rebooting

1. Make sure the hard key is inserted in the computer on which the utility will be run.
2. Click Start on the Windows task bar.
3. Select Run on the Start menu.
4. Enter `ShowLicense` in the Run dialog box.
   A Show License utility opens.
5. Select the following options on the **Show License** screen (shown below):

Option	Description
A	Check **Direct** to direct the report to what is on the new key.
B	Click **Refresh** to update the screen with updated licensing information when the licensing service reads the key. The key is read every 30 seconds.
C	Click **Update cache**.
D	Click **OK** when finished.

CIMPLICITY now recognizes the new option(s) as being licensed.

Chapter 4. Integrating CIMPLICITY with Web HMI

Setup Prerequisites

By integrating Web HMI and CIMPLICITY, you can view CIMPLICITY data in Web HMI using screens called mimics that were created in CimEdit. Complete the following tasks before you begin the integration configuration tasks.

1. Install the latest versions of CIMPLICITY and Web HMI.
2. Create an administrative user that you can use for both CIMPLICITY and Web HMI. The user IDs must match in order to connect and manage communications and operations between the two systems, but the passwords must be different.
3. Using the Web HMI model editor, create the Web HMI model. For information, refer to the Web HMI user documentation.

CIMPLICITY Configuration

Configuring OPC UA

Enabling and Configuring the OPC UA Server

After you have set up your CIMPLICITY project, you can open the Project Properties dialog to configure the OPC UA server and secure communications between CIMPLICITY and Web HMI.

! **Important:** The CIMPLICITY OPC UA Server does not support enterprise points for use with Web HMI.

1. In Workbench, select Project from the menu bar, and then click Properties.

2. Select the OPC UA Server tab and click Enable Server.

3. Enter the applicable data in the Port, Network Address, and Logical Host Name fields. You can view how this data affects the entries in the Endpoint URL, Server URI, and Server Name fields.

Note: If you do not change the default value of the Logical Host Name field, it will be replaced with your computer's host name.

4. Click **Security Configuration** and choose the security levels from which you can select when you complete the steps to secure your endpoints.

   Note: The higher security levels are selected by default and it is recommended that you keep both Sign and Encrypt check boxes selected. You can also select None to facilitate your initial setup but you should always set up strong security for all non-test environments. The recommended setting is Basic256Sha256 Sign and Encrypt.

5. Click OK.

   Using a secure configuration requires using security certificates. You can either manage these certificates manually or you can use the Global Discovery Server to manage them.

   For information, continue to Securing CIMPLICITY OPC UA Endpoints *(page 19)*.

Securing CIMPLICITY OPC UA Endpoints

Certificates are strongly recommended for production environments. You can configure certificates manually or you can set up automatic management by enabling the Global Discovery Server for your CIMPLICITY project.

Note: This section provides basic information for setting up your endpoint security. For comprehensive information on the OPC UA Server security options, navigate to **Communications Equipment > CIMPLICITY OPC UA Security Configuration > OPC UA Server Certificate Configuration**.

1. In Workbench, from the tree on the left, select **Project > Security > OPCUA Security Configuration**.
2. In the CIMPLICITY OPC UA Certificate Configuration dialog: if you are configuring GDS as part of the OPC UA endpoint security, ensure the **Use GDS** check box is selected. If you are managing certificates manually, clear the check box.
3. Click **Enable Security**. One of the following occurs:

   If you are using a self-signed certificate (manual certificate management), the shorter validation sequence appears immediately in the Certificate Configuration dialog.

If you are using GDS, a login screen appears for the GDS endpoint, and you must enter the user name and password for the Global Discovery Server.

4. Click **OK** to continue. The validation sequence for accessing the GDS server begins.
5. When all operations show as complete in the **Status** column (green check mark), click **Close**.
6. Start or restart your CIMPLICITY project.

Exporting CIMPLICITY Model Data to Web HMI

1. Select **Project** from the menu bar.
2. Click **Export to Web HMI** to export the CSV file to the location specified when you set up the OPC UA server information.
3. Click **Save**.

   For a successful export, the following message appears:

   ```
   Successfully exported Web HMI tag groups.  Do you want to review the
    resulting file?
   ```

4. Click **Yes** or **No**.

Publishing CimEdit Screens

Referencing Data in CimEdit Screen Animations

You can use the following two methods to reference data in CimEdit screens that will work in Web HMI mimics.

❗ **Important:** The CIMPLICITY OPC UA Server does not support enterprise points; therefore, they are not supported in exports for Web HMI. If you use enterprise points in your CIMPLICITY screens, they will not be available in the imported Web HMI mimic.

Method	Description
Method 1	Continue to directly reference your CIMPLICITY point database in your animations – This method retrieves data directly from CIMPLICITY and shows the values of the points in the database. This is often useful when referencing data that is global to or outside of the model context you build in Web HMI. For example, OverallPartCount.

Method	Description
Method 2	Reference model context-based data that resolves in accordance with the model context you are in when you display the mimic in Web HMI. This is the most powerful way to use Web HMI mimics because it allows you to create screens as templates using CIMPLICITY’s existing local variable mechanisms that resolve at Web HMI runtime. For example, if you have created a Web HMI object type to represent a pump and created 2 pumps objects from that type, you can build one screen in Web HMI to represent the pump object type, and then have the data resolve and appear in that screen/mimic based on the Pump instance that is selected.

You can reference both direct CIMPLICITY database points and context references in the same screen.

Use the $OBJECT local variable to reference context-based data. You must define this variable at the screen level and leave the value blank when you export the screen as a mimic for Web HMI. You can use this local variable to represent the model context with which this screen is displayed. For example, a pump screen is shown when in the Pump01 or Pump02 context.

Note: You should build the model in Web HMI before you build Web HMI mimics in CimEdit. If you reference model object variables using the same names, Web HMI can automatically map your model context references in your animations to the model when you import the mimic into Web HMI.

Example 1: Referencing Contained/Child Model Object Variables

When in the Inlet pump context, the animation shown below displays the value of Inlet Pump’s flow. If on the Outflow Pump context, this same screen will show Outflow Pump’s flow.

$Object.Flow

Example 2: Referencing Contained/Child Model Object Variables

You can also reference contained object properties in your Web HMI mimics that will be exported from CimEdit. For example, if I want to build an overview screen to show two pumps in my pump station, then I can build a CimEdit screen where the $Object represents PumpStation01 and I reference my animations to show the individual pump values as follows:

{$Object}.InletPump.Flow

{$Object}.OutflowPump.Flow

Example 3: Referencing CIMPLICITY Point Attributes in Web HMI Example

You can continue to reference CIMPLICITY attributes like engineering units, description, alarm limits, and so forth, in Web HMI screens on direct point references and model context references as follows:

{$Object}.Flow.Description or

{$Object}.InletPump.Flow.EU_LABEL

Using Linked Objects to Generate Web HMI Screens

In order to build Web HMI screens using linked objects, any Linked Object local variables that are used to reference Web HMI Object variables must eventually resolve back to the screen $Object variable in which they are placed. When you create the source of a linked object, you create a public "$Object" local variable in your source group object. Use this $Object local variable to reference Web HMI model variables.

For example:

{$Object}.Flow

When you create a linked object from this source group object, you set the value of the inherited $Object local variable to point to the screen $Object defined in the screen in which you created the linked object using local variable syntax like {..\$Object} or {..\$Object}.InletTank"

What is Supported in Web HMI

The information below describes the items that you can add to a CimEdit screen, and which are viewable in Web HMI.

Objects

Object	Supported	Exceptions
Line/Polyline/Polygon	Yes	• Arrowheads are not supported in Web HMI. • The "winding rules" are currently different in Web HMI and CimEdit/CimView. If a filled polyline crosses over itself, in CimEdit/CimView you may see "holes" in the filled object. However in Web HMI there will be no holes. A classic example is a 5-pointed star: in CimEdit the center will be empty while in Web HMI it will be filled.
Rectangle	Yes	
Rounded Rectangle	Yes	
Arc	Yes	
Chord	Yes	
Pie	Yes	
Ellipse	Yes	

Object	Supported	Exceptions
Pipe	No	
Picture	Yes	• Insert as link: If an image is inserted as a link, the image file will be copied and bundled as part of the mimic. Later changes made to the linked image file will not be reflected in Web HMI unless the screen is re-exported as a mimic. • Display Transparency: Image transparency that is part of the image file is fully supported (e.g., transparency in the PNG file). The "Transparency" property in the image property page that lets you specify an additional color to be made transparent in the image is not supported in Web HMI. • Lock Aspect Ratio is not supported in Web HMI. The resized image may still have different aspect ratio when rendered in Web HMI. • Animation Speed: Using an expression to control the frame rate of a GIF image is not supported in Web HMI. • Line and Fill: In CimEdit the line and fill on the image object provide a frame and backdrop for the image. These are not rendered by Web HMI. Only the image itself will be displayed. These attributes are exported, but Web HMI does not render them. In addition having solid fill behind the image will result in just the filled rectangle displayed in Web HMI, as if the image is behind the fill, and not vice versa.
Text	Yes	• Font strikeout and underline are not supported. Only normal, bold, and italic font styles are supported. There are some restrictions on the font names that can be used. • Text translation (the Translate checkbox) is not supported in Web HMI, neither as part of the static text nor as part of Display Value Animation. • Word wrap is not supported in Web HMI. • Display Value Animation: A text object can show the value of a point or expression. There are some differences between the way Web HMI formats expression values and the way CimView does. These may be addressed in future releases. In particular the “%s” format (used by Configured, General, and Text) will format numbers differently in Web HMI and CimView. When displaying numbers, it is best to use one of the numeric format types. The Configured display format will always use the “%s” format; it will not get the display format from the point configuration. The Time display formats are not supported.

Object	Supported	Exceptions
Text Button	No	
Groups (nested and animations)	Yes	
Smart Objects	Partial	• Smart Objects are just groups that can be added to a CimEdit screen either by copying or linking. All of the restrictions that apply to other objects apply to Smart Objects as well. • Smart Objects that have “Smart Object” events tied to a script can be used because those scripts run in CimEdit. • Smart Objects that have runtime scripts (that execute in CimView) will not have those scripts run in Web HMI.
Linked objects	Partial	• When exported into a mimic, linked objects are copied into the mimics as groups. Later changes to the link source objects will not affect the mimic unless the screen is re-exported as a mimic.
OLE and .NET objects	No	• Including Trend, SPC, Alarm Viewer, Historical Alarm Viewer, Proficy Portal, Task List, and Task Indicator.

Object Properties

Object Property	Supported	Exceptions
Colors	Partial	• Line: Only the “none” and “solid” line styles are supported. Dashed and dotted lines are not currently supported. • Fill: Only the “no fill” and “solid” fill styles are supported. Patterns and gradients are not currently supported. • Special colors (e.g., “Active title bar” color) and index colors are exported as their ARGB values at the time of export.
Geometry	Partial	• Static rotation and shear as configured in the Geometry tab are not supported. (Rotation animation is supported.)

Object Property	Supported	Exceptions
General	Partial	• Object help, tab order, and tool tips are not supported.
Movement Animation	No	
Scaling Animation	No	
Rotation/Fill > Rotation Animation	Partial	• Currently, Web HMI only supports rotation for rectanglse, polylines, and polygons. Other shapes "revolve," in that they move around the center of rotation but do not change orientation. If the center of rotation is the same as the center of the shape, "revolution" has no effect. Text and child groups currently do not rotate or revolve – they stay in place. • Fill animation doesn't work properly with rotation. • Shape rotation does not add to group rotation. • Static rotation (configured on the Geometry tab) is not supported, so it has no cumulative effect when combined with rotation animation. • Click zones do not rotate properly with shapes.
Rotation/Fill > Fill Animation	Yes	• Bipolar fill animation is not supported. • As with the Colors tab, only “no fill” and “solid” fill styles are supported and only ARGB colors are supported.
Color Animation > Expression List Animation	Partial	• In Web HMI, expression list animation can only update the line color, the fill color, and/or the text value. You cannot change the line or fill style, font attributes, or blink.
Color Animation > Visibility	Yes	
Color Animation > Alarm State	No	
Color Animation > Color Index	No	
Transparency Animation	No	
Angle Animation	Yes	Fill animation does not work properly with angle animation.
Shadow	No	

Object Property	Supported	Exceptions
Events	Partial	• Only mouse up/down events on group objects are supported. They are converted into Web HMI click zones. These may be configured in Web HMI to do several kinds of actions. • Smart Object events that run in CimEdit are, of course, still supported in CimEdit.
Script	No	
Variables	Yes	• Variables that have non-empty values in CimEdit will be substituted into the expressions as they are parsed. • Variables that have empty values in CimEdit will be turned into model references.
Events	Partial	
Script	No	
Menus	No	
Procedures	No	

Expression and Data Sources

Item	Supported	Details
Expressions	Partial	The following operations are supported: • Arithmetic functions: + - * / ^ (pow) ABS CEIL FLR RND EXP LOG LOG10 MOD SQR TRUNC • Bitwise functions: BAND BNOT BOR BXOR SHL SHR • Logical functions: AND NOT OR XOR • Relational functions: EQ GE GT LE LT NE • Statistics functions: MAX MIN • String functions: + (concat) VAL CHR$ FindToken FindTokenCS GetToken InStr InStrCS Left Mid Right StrLen ToLower ToUpper Trim • Trigonometry functions: ACOS ASIN ATAN COS SIN TAN • IF/THEN/ELSE (?:) • Time functions: CalcSpan CalcStamp The following operations/functions are not supported: • Hard-coded string expressions. • Advanced string functions (CHR$ FindToken FindTokenCS GetToken InStr InStrCS Left Mid Right StrLen ToLower ToUpper Trim). • Alarm and point functions (A1 A2 AH1 AH2 AL AL1 AL2 ANA CalcSpan CalcStamp EU_CONV NACK GetQualityBit IsAvailable IsConfigured QL TS). • Historian functions (GetHistTagID HistAv HistCom HistCount HistDesc HistEU HistHighEU HistI HistLowEU HistMax HistMin HistN HistRawAv HistRawStandDev HistRawTot HistS HistStandDev HistTot). Note the following differences/restrictions: • / (division) - In CIMPLICITY, integer division yields an integer result. This behavior has been reproduced in the exported expressions for Web HMI, so that will work in Web HMI. • float to integer conversion - In CIMPLICITY if a floating point number is converted to an integer and the value exceeds the min or max value of a 64-bit integer, it evaluates to an error. In Web HMI, all numbers are represented as floating point and the conversion does not evaluate to an error. • integer precision - In Web HMI, all numbers are represented as 64-bit floating point numbers. Therefore integer values have only about 53-bit (15 digits) of precision.

Item	Supported	Details
Data Sources	Yes	• CimEdit supports exporting mimics with both "model references" (export as tags like @Temperature@ which get bound to model properties at run time) and "hard coded" point IDs (which get exported as explicit OPC UA node IDs). • Model references are created in a CimEdit screen by using either $OBJECT notation (like {$OBJECT}.Temperature) or a simple variable reference (like {Temperature}, where the Temperature variable is defined in the screen but has an empty value). • Hard coded point references are created in a CimEdit screen by using a point ID directly. • In Web HMI, model references can be bound to real-time (OPC UA) or historical (Historian) data sources. • Hard-coded point references always refer to a real-time data source. A direct reference to a historian tag (e.g., obtained through the Browse > Browse Historian Tags) is not supported. • The $LOCAL system points are not supported in Web HMI. • The system points $USER, $ROLE and $ROLE.LEVEL will reflect the user configured with the Web HMI Server Configuration Manager OPC UA Endpoint Server Credentials.

Point Attributes

Point attributes (e.g., MyPoint.Description) are not currently available in Web HMI for historical data sources.

Supported Attributes	Unsupported Attributes
ADDR_OFFSET	$RAW_VALUE
ALARM_HIGH	ACCESS_FLAG/ACCESS_LEVEL
ALARM_HIGH_N	ACK_TOUT (Obsolete)
ALARM_LOW	ADDR
ALARM_LOW_N	ALARM_CRITERIA
ANALOG_DEADBAND	ALARM_DELAY (Obsolete)
DEADBAND	ALARM_STATE
DEADBAND_N	ANALOG_DEADBAND_N
DESCRIPTION	CALCULATION_TYPE
DEVIATION_PTID	CLR_TOUT (Obsolete)
DEVICE_ID	CONV_TYPE

Supported Attributes	Unsupported Attributes
DISPLAY_LIM_HIGH	DEL_OPT (Obsolete)
DISPLAY_LIM_HIGH_N	DP_FLAG
DISPLAY_LIM_LOW	EU_EXPRESSION
DISPLAY_LIM_LOW_N	EU_LABEL
EXTRA	EU_REV_EXP
LEVEL	EXTENDED_USER_FLAGS_HIGH
POINT_STATE	EXTENDED_USER_FLAGS_LOW
QUALITY	FLAGS
QUALITY_ALARMED	FORMAT_PREC
QUALITY_ALARMS_ENABLED	FORMAT_WID
QUALITY_DISABLE_WRITE	FR_ID
QUALITY.IS_AVAILABLE	HI_ACK_TOUT
QUALITY.IS_IN_RANGE	HI_ALARM_DELAY
QUALITY.LAST_UPD_MAN	HI_ALARM_OFF_DELAY
QUALITY.MANUAL_MODE	HI_CLR_TOUT
QUALITY.STALE_DATA	HI_DEL_OPT
RANGE_HIGH	HI_REP_TOUT
RANGE_HIGH_N	HIHI_ACK_TOUT
RANGE_LOW	HIHI_ALARM_DELAY
RANGE_LOW_N	HIHI_ALARM_OFF_DELAY
SETPOINT_HIGH	HIHI_CLR_TOUT
SETPOINT_HIGH_N	HIHI_DEL_OPT
SETPOINT_LOW	HIHI_REP_TOUT
SETPOINT_LOW_N	INIT_VALUE
SETPT_CHECK_PTID	LO_ACK_TOUT
USER_FLAGS	LO_ALARM_DELAY
WARNING_HIGH	LO_ALARM_OFF_DELAY
WARNING_HIGH_N	LO_CLR_TOUT
WARNING_LOW	LO_DEL_OPT
WARNING_LOW_N	LO_REP_TOUT
	LOCAL

Supported Attributes	Unsupported Attributes
	LOLO_ACK_TOUT
	LOLO_ALARM_DELAY
	LOLO_ALARM_OFF_DELAY
	LOLO_CLR_TOUT
	LOLO_DEL_OPT
	LOLO_REP_TOUT
	MEASUREMENT_UNIT_ID
	POINT_ID
	POINT_SET_INTERVAL
	POINT_SET_TIME
	PROCESS_ID
	PTMGMT_PROCESS_ID
	RATE_TIME_INTERVAL
	REP_TOUT (Obsolete)
	RESET_POINT_ID
	ROLLOVER_VALUE
	SCAN_POINT
	SCAN_RATE
	TIMESTAMP
	TRIGGER_POINT
	TRIGGER_TYPE
	TRIGGER_VALUE
	VARIANCE_VALUE

Publish Mimics to Web HMI directly from CIMPLICITY

You can publish a Mimic to Web HMI directly from CIMPLICITY. This eliminates the two-step process of publishing a Mimic by exporting a screen as a Mimic zip file from the CIMPLICITY server and then importing the Mimic zip file to Web HMI. Publishing a Mimic to Web HMI directly from CIMPLICITY is especially useful when you want to publish multiple CimEdit screens simultaneously. You can configure Web HMI at the project level and globally to avoid configuring it while publishing Mimics. This task describes how to configure Web HMI at the project level. (Previously, to configure Web HMI, we selected Web HMI Configuration in the OPC UA Server tab in the Project Properties window. This has now been updated as mentioned in the procedure below.)

1. In the **CIMPLICITY Workbench** window, select **Project**, and then select **Properties**.

2. In the Project Properties window, select the Web HMI tab and enter the following details:
    - **Server Name:** The machine name on which Web HMI is running or the URL of the Web HMI Server
    - **Port:** The port on which Web HMI is running
    - **User Name:** The Web HMI user name used to publish the Mimic
    - **Require Trusted Connection:** Select this check box to use a trusted connection to publish Mimics. You can also select View Certificate to view the certificate used for the connection.

3. Select OK.

4. To configure Web HMI globally,cClose the CIMPLICITY Workbench window (if it is open), and launch CimEdit from the Start menu.

5. In the CimEdit window, select , and then select Global Configuration.

6. In the Global Configuration window, select the Web HMI tab and enter the following details:
    - **Server Name:** The server name on which Web HMI is running
    - **Port:** The port on which Web HMI is running
    - **User Name:** The Web HMI user name used to publish the Mimic

7. Select Apply, and then select OK.
   Web HMI is now configured for Mimics to be published in it. You can publish a Mimic from the following applications: Workbench , CimEdit.

8. To publish Mimics to Web HMI directly from Workbench, In the **CIMPLICITY Workbench** window, select the Mimics that you want to publish to Web HMI, right-click the Mimics, and then select **Publish to Web HMI**.

   Note: In the Publish window, the Server Name, Port, and User Name are automatically populated if Web HMI has already been configured at the project level. If the fields are blank, enter the details before proceeding to the next step.

9. In the Publish window, enter the following details:
    - **Save To:** The location where the Mimics will be saved by default. This field is disabled.
    - **Publish to Web HMI:** Select this check box to publish the Mimics to Web HMI.
    - **Overwrite:** Select this check box to overwrite Mimics with the same name.
    - **Password:** The password to publish the Mimics.
    - **Require Trusted Connection:** Select this check box to publish Mimics only when a trusted connection is used. You can also select View Certificate to view the certificate used for the connection.

Note: These details are saved after you publish the Mimics. You only need to enter the password if you close and reopen the CIMPLICITY Workbench window.

10. Select Publish.
    The selected Mimics are published to Web HMI and appear in the Mimic Management workspace.
11. To publish a Mimic to Web HMI directly from CimEdit, go to the location where the Mimic file you want to publish is saved, right-click the file, and select **Edit**.

    Note: You can publish only one Mimic at a time to Web HMI from CimEdit.

12. In the CimEdit window, select , and then select Publish to Web HMI.

    Note: In the Publish window, the Server Name, Port, and User Name are automatically populated if Web HMI has already been configured (depending on the project context) at the project level or globally. If the fields are empty, enter the details as described in step 2 of the Configure Web HMI section before proceeding to the next step.

13. In the **Publish** window, enter the following details:

    Note: If you want to save the Mimic to a shared location on a remote server, the network drive should be mapped to the shared location.

    - **Save To**: The location where the Mimic will be saved locally. You can choose this location since you are publishing only one Mimic at a time.
    - **Publish to Web HMI:** Select this check box to publish the Mimic to Web HMI.
    - **Overwrite:** Select this check box to overwrite any Mimic with the same name.
    - **Password:** The password to publish the Mimic.
    - **Require Trusted Connection:** Select this check box to publish Mimics only when a trusted connection is used. You can also select View Certificate to view the certificate used for the connection.

    Note: These details are saved after you publish the Mimic. You only need to enter the password if you close and reopen the CimEdit window

14. Select **Publish**.
    The Mimic is published to Web HMI and appears in the Mimic Management workspace.

Publish Model to Web HMI directly from CIMPLICITY

You can publish a Model to Web HMI directly from CIMPLICITY. This eliminates the two-step process of publishing a Model by first exporting the model data as a .csv file from the CIMPLICITY server and then importing the .csv file to Web HMI. You can publish a Model to Web HMI directly from CIMPLICITY only on versions of Web HMI 2.2 and later. The Workbench application may become unresponsive during the process of publishing a Model to Web HMI. Model data consists of classes, objects, points, and details of OPC UA and Historian servers. Model data is also project-specific, where the details configured will apply to the last updated project.

1. In the **CIMPLICITY Workbench** window, select **Project**, and then select **Publish Model** to Web HMI.
2. In the Model Publish window, enter the following details:
   - **Server name:** The machine name on which Web HMI is running or the URL of the Web HMI Server.
   - **Port:** The port on which Web HMI is running.
   - **User name:** The Web HMI user name used to publish the Model data.
   - **Password:** The password used to publish the Model data.
   - **Include Historian source:** Select this check box to include points configured to the Historian server in the Model data published to Web HMI.
   - **Include non-class/object points:** Select this check box to include points that are not related to classes or objects in the Model data published to Web HMI.
   - **Require trusted connection:** Select this check box to publish Model data only when a trusted connection is used. You can also select View Certificate to view the certificate used for the connection.

   Be aware that:

   - The Server name, Port, and User name fields are automatically populated if Web HMI has already been configured at the project level.
   - All the details in the Model Publish window are saved for a project after you publish the Model data. You only need to enter the password if you close and reopen the CIMPLICITY Workbench window.
   - The connection is not trusted when the server has a blank or invalid name, or has an invalid certificate.
3. Select **Publish**.
   The Model data is published to Web HMI and appears in the Assets workspace with classes mapped to asset types and objects mapped to assets.

Web HMI Configuration

Configuring the OPC UA Endpoint in Web HMI

Certificates are strongly recommended for production environments. You can configure certificates manually, as described below, or you can set up automatic management by enabling the Global Discovery Server. For information, see Configuring Web HMI for GDS *(page 34)*.

To complete your secure endpoint setup in the Web HMI Server Configuration Manager, do the following:

1. Ensure your COMPLICITY project is running. If necessary, redisplay Workbench, then click the green arrow in the toolbar to start the project. A pop-up showing the start sequence appears. When the sequence is complete the green arrow changes to a red square and the pop-up closes automatically.
2. Click the **Web HMI Server Configuration Manager** icon on the desktop, then select the **OPC UA Endpoints** tab and click Add to add the CIMPLICITY endpoint.,
3. In the **Discovery URL** field, enter the registered CIMPLICITY endpoint URI, then select **Direct** from the drop-down list on the right and click **Discover** to populate the area below with the URI and associated endpoints.
4. Click the + symbol to expand the URL and display the endpoint, then click the endpoint to display the security settings you chose when you enabled the OPC UA Server in the CIMPLICITY configuration.
5. Select a security setting and click **Server Certificate.**
6. Click **Trust > Close**.
7. With the security setting still selected, click **Server Credentials**, then enter the user name and password for your CIMPLICITY project and click **Test**. The initial test fails and a certificate appears in your project's rejected folder.
8. Navigate to the **PKI** subfolder of your project folder and open the **rejected** folder.
9. Copy the certificate in the **rejected** folder to the **pki\trusted\certs**. Create the **trusted** and **certs** folders if necessary.
10. Verify the connection again.
    The Log will now display a success message.
11. Click **OK** to close the Server Credentials dialog, then click **OK** to close the Configure Endpoint dialog.
12. Click **Save** to save the configuration and exit. You may receive a prompt to have the tool restart key services so changes made in this tool are picked up.

Configuring Web HMI for GDS

1. Open the Server Configuration Manager and select the **OPC UA Client** tab.

2. Select the **Use GDS** check box, then click **Configure GDS**.
3. In the GDS dialog, enter the URL for the Global Discovery Server, then enter the corresponding user name and password.
4. Click **Test** to verify the endpoint connection and display the log screen.

   If the connection was successful, the log screen shows green status lines for **Server Status (Connected)** and **Successfully connected to endpoint with URL <url identifer>**..

   If you receive a **BadCertificateUntrusted** error in the log results, a separate Server Certificate dialog opens, click **Trust** and then run the test again.
5. When the test results are successful, click **Enable Security**.

   The following log window appears.
6. Click **Save**.

Defining the CIMPLICITY Project Connection in Web HMI

1. In Web HMI, select **Administration** from the drop-down menu in the upper-right corner.
2. From the left panel, select **Set Up > Server** to open the Server Details screen.
3. Click the + sign to add a row.
4. Open the project information CSV file that you exported from CIMPLICITY.
5. Enter the information as described below.

Field	Description
Server Alias	Add a server alias name. You reference the CIMPLICITY project connection using this value when setting up the object data sources. In general, use the Project name as the alias. This is required if you are referencing CIMPLICITY point values directly in your screen.
Server Type	Select OPCUA for the server type. CIMPLICITY communicates to GE Web HMI through its OPC UA server.
Server Name	Enter your CIMPLICITY Uniform Resource Name (URN) for the server name. You can retrieve the URN from the CIMPLICITY Project Properties screen Server Name or from the exported CSV file, as shown in the above sample.

6. Click the **Save** button in the upper right corner.

Creating an Object and its Defining Data Sources

In the Web HMI model, configuring real time data sources for object variables allows Web HMI to retrieve data from applicable CIMPLICITY points. The object variables must reference the CIMPLICITY points using the OPC UA node format. This topic assumes you have created object variables as part of your model, which you created as part of your setup prerequisites.

1. Select the object from the list to display the details on the right.
2. Open the project information CSV file that you exported from CIMPLICITY (per the Exporting CIMPLICITY Model Data to Web HMI *(page 20)* topic).
3. Locate the data in the file and then enter it in the applicable data row under **Data Variables**. As shown above, the #TagGroupHeader row matches the headings and the TagGroup row shows the information you should enter in the data row.
4. Click the **Save** button in the upper right corner.

Importing and Binding the Mimics

Import the Mimics

1. Open Web HMI and select Administration from the drop-down list in the upper-right corner.
2. Click **Visualizations** on the left, then select **Mimic Management**.
3. Click **Import.**
4. In the Import Mimics pop-up, click **Browse**.
5. In the file selection dialog, select the mimics to be imported and click **Open**. The file selection dialog closes.
6. Click **Import** and wait for the mimics to be imported, then click **Close.**

Bind the Mimics

1. Under **Visualizations**, click **Designer**, as shown below.
2. Under **Object Types**, select an object.
3. On the right, select the corresponding mimic from the **Mimic** drop-down list.

4. Click **Save**.

Checking Mimics in Runtime Environment

After you bind your imported mimics to Object Types, you can select Runtime from the drop-down list in the upper-right corner to check them in runtime, as shown below.