Proficy HMI/SCADA - CIMPLICITY 2022

Advanced Features

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

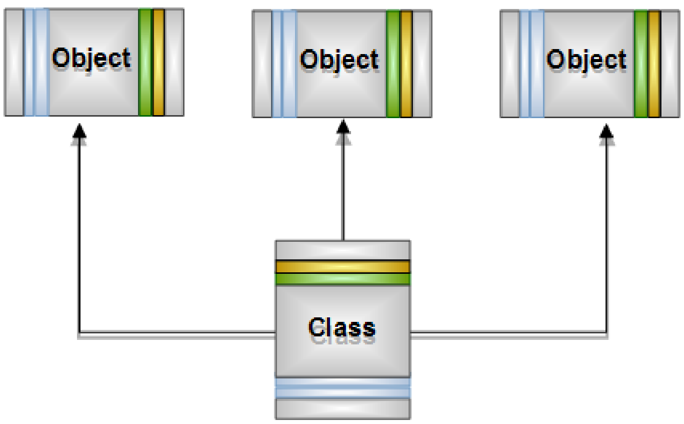
doc@ge.com

Chapter 1. Classes and Objects

About Classes and Class Objects

Instead of repeating the same configuration for components that have similar requirements, you can create the basic configuration for those components as a template to use as a configuration starting point for each.

The CIMPLICITY tools that provide this capability are Classes and Objects.



Item	Description
Classes	A Class is a template that: • Provides a flexible structure to quickly configure objects with similar requirements. • Includes attributes, points (data items), actions, events, scripts, and CimEdit screens. A class can be exported in a single .soc file. A custom help file can be created with third party tools and also associated with the class. Beginning in CIMPLICITY 9.5, there are Composite Classes and Objects. Classes can reference other classes. You can modify a class dynamically and you can also delete a class dynamically if there are no references to the class or no object instances of that class with composite references to the class. Read more about Composite Classes here *(page 114)*.
Objects	As many objects as needed can be created based on a single class template. An Object is one instance that is created using a class as a template. An object can be whatever you define it to be. Object configuration takes advantage of the work that went into creating the class. This can greatly reduce repetition that would otherwise be required if configuration for each object was done independently. If a value that was entered as a default value for a class is changed for a selected object in that class, the object's value will not be overwritten if the default class value is changed.

Important: It is extremely important for you to understand CIMPLICITY project configuration before you begin to configure classes and objects. If you have, you will find that class/object configuration is straightforward. If you have not, you will most likely run into problems.

Overview: Class Instantiation Process

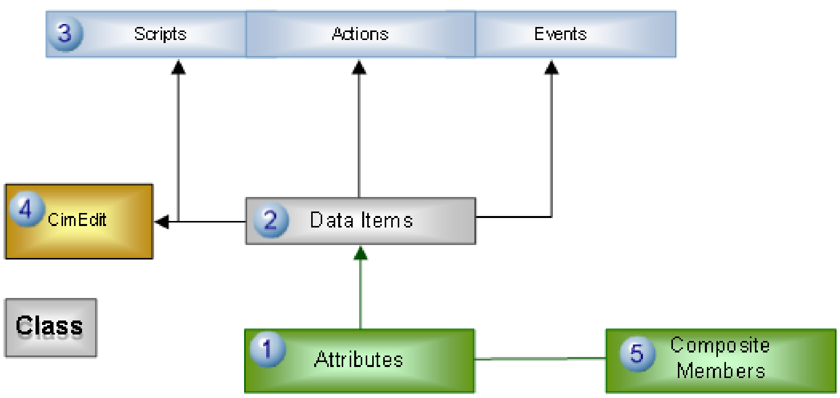
Class: Class Configuration

Configuring features for a class is similar to configuring features in a project. The class includes some additional functionality to enable a class developer to create a flexible template that can serve as the basis for many objects.

- Features included in a class template.
- Class dialog box.
- Classes listed.

Features Included in a Class Template

The following features can be included in class template.



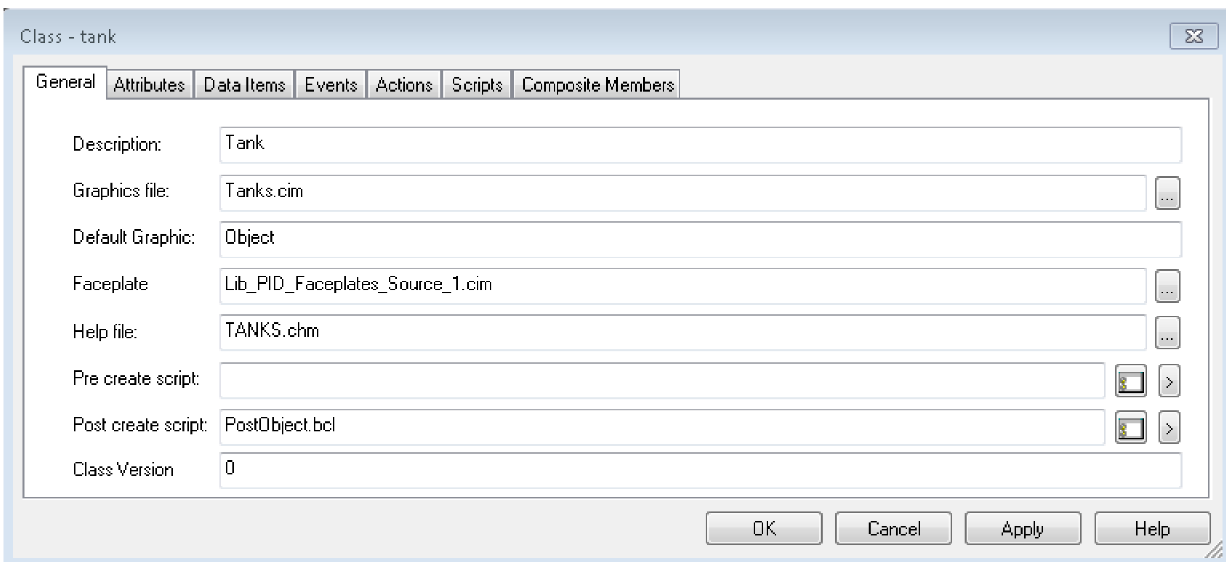
Tool		Use
1 (page 16)	Attributes (variables)	Numeric or string placeholders to which values will be assigned when instantiating the object. Provide the flexibility required to insure that the class has the range of possible values among objects that are created from the class. **Note:** When you configure the class, you specify how the attribute can and will be dealt with by an object designer.
2 (page 29)	Data items	Data items make use of the attributes' flexibility by Incorporating attributes in their definitions. There are the templates for points that will be instantiated when objects are created.
3 (page 75)	Scripts/ Actions/ Events	Scripts, actions and events can include data items in their configuration.

		Scripts	Apply to all objects in the class..
		Actions	Apply to all objects in the class
		Events	Apply to all objects in the class for triggering actions.
4 (page 104)	CimEdit Graphic File		A CimEdit screen that includes one or more class objects can be associated with a Class. Any of the CimEdit graphic class objects can be used on an object screen; one can be selected as the default. Any of the CimEdit graphic class objects can also incorporate data items that hold the place for the class objects' instantiated points.
5 (page 114)	Composite Members		Composite members are references to other classes. They can have their own data items, events, and actions.

Note: A Help file, which is a custom file designed to provide help during object creation.

Class Dialog Box

Class entries are made in a Class dialog box.



Field	Use
Faceplate	The faceplate screen is a screen you can add to the class. When the class has a faceplate associated with it, anytime you left click on a class object in a screen its does a popup of the faceplate screen, and assigns the $OBJECT variable on the faceplate with the $OBJECT variable value of the class object that was clicked on. The faceplate screen is automatically shown with a dialog border, and basically does the same thing that the PPS global faceplate script did.
Class Version	The class version is automatically incremented when there is a structural change to the class. This include adding or removing data items, and composite members. The class editor can also manually assign to version also. When doing class imports, if the version of the class is lower than the current class version, there will be a warning that you are importing a lower version number over a higher version number.

Classes Listed

Classes that have been created are listed in the Workbench right pane. These classes can be exported from projects or imported into projects.

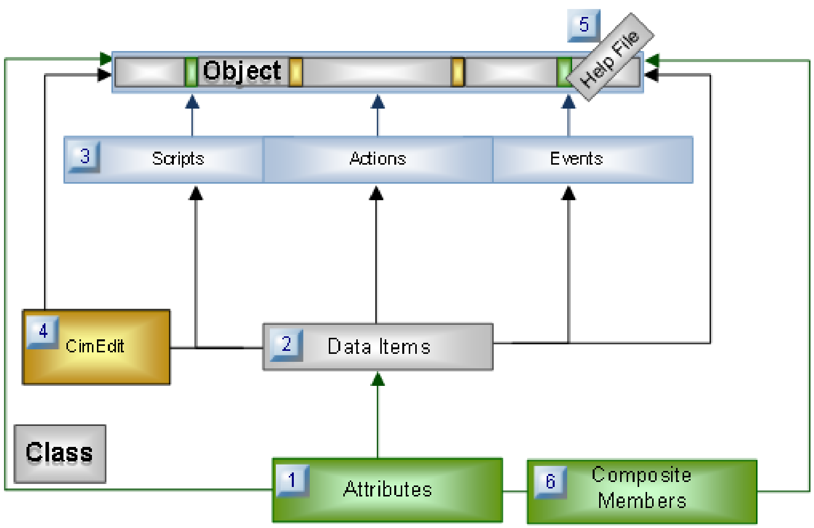
Object: Object Configuration

A class object is a quickly created set of project features that are included in its class template.

- Object based on class.
- Object dialog box.
- Objects listed.

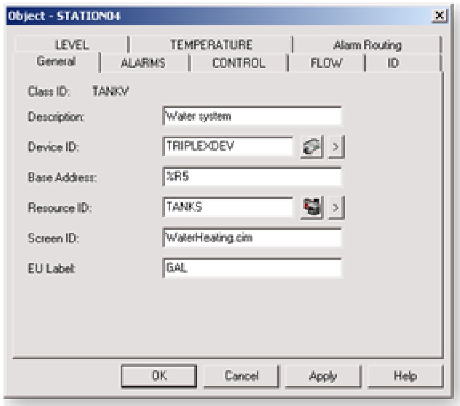
Object Based on Class

Objects can be created using any class as its template.



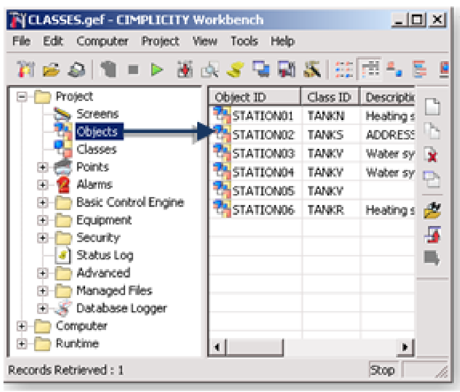
Object Dialog Box

Entries to define a class object are made in an Object dialog box. Fields and tabs that are included in the dialog box are based on the selected Class configuration.



Objects Listed

Objects that are created are listed in the Workbench right pane.



Instantiation: Features Enabled

Each object includes a full set of features instantiated from the class on which it was created.

- Attribute placeholders are replaced by values that are associated with class objects.
- Data items are replaced by points whose ID's include an object ID prefix.
- Scripts whose ID's include a class ID prefix are applied to the object.
- Actions whose ID's include an object prefix are listed in the Event Manager.
- Events whose ID's include an object prefix are listed in the Event Manager.
- CimEdit/CimView screens that incorporate selected class containers are available.

Checklist: Planning Classes

Because a point class is a template for objects that may be created in remote as well as local projects, it is essential that you plan the class to anticipate all possible object requirements.

A brief checklist includes the following setup considerations. Beginning with CIMPLICITY 9.5, Composite Classes are supported.

Determine if the:

	Application in the production facility, if more than one object (process) appears to have similar requirements.
	PLC layout is consistent among the objects (where applicable).
	Data to be collected is identical. **Note:** The values do not have to be identical.
	Layout of how memory is stored in the devices from which data is collected.
	Calculations needed for the data are identical.
	CIMPLICITY features, e.g. points, scripts, actions and events, required to collect and calculate data.

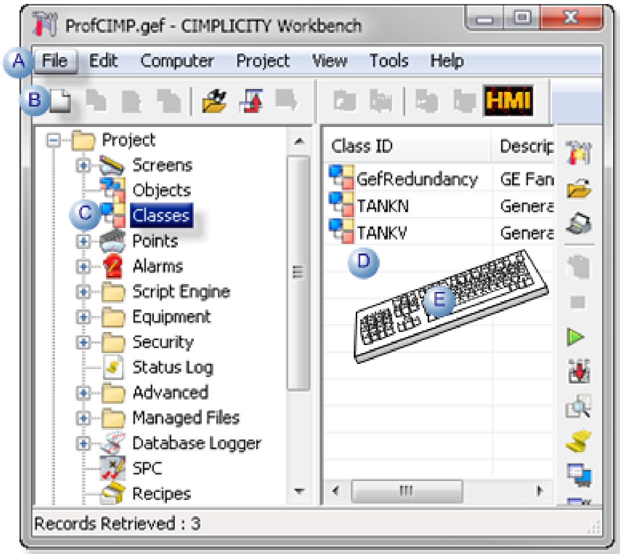
Open a Class Dialog Box

Open a Class Dialog Box

Option 2.1 *(page 11)*	Create a new class
Option 2.2 *(page 13)*	Open an existing Class dialog box.

Option 1. Create a New Class

1. Select **Project>Classes** in the Workbench left pane.
2. Do 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 **Classes**.	a. Right-click **Classes**. 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.	

The New Class dialog box opens using any of the above methods.

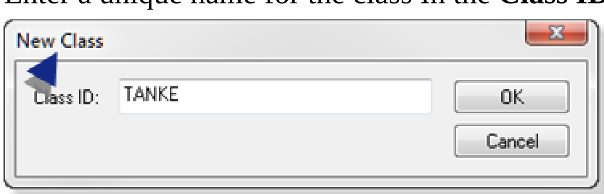
3. Right-click **Classes**.

4. Select New on the Popup menu.

5. Right-click anywhere.

6. Select New on the Popup menu.

7. Enter a unique name for the class In the **Class ID** field.



The Class ID

- Is limited to 32 characters.
- Can be composed of:
- Alphanumeric characters.
- Underscores.
- Must begin with an alphabetic character.
- Cannot have spaces.

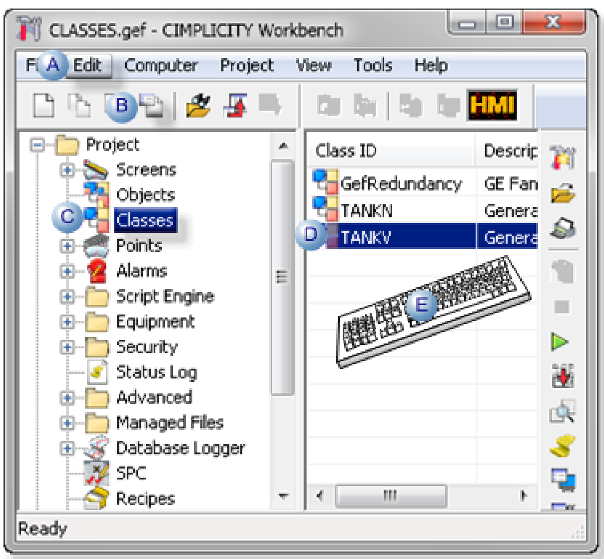
8. Click OK.

Option 2. Open an Existing Class Dialog Box

1. Select **Project>Classes** in the Workbench left pane.

2. Select a class in the Workbench right pane.

3. Do 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 **Classes**. b. Select Properties on the Popup menu.
D	In the Workbench right pane, double click a class.
E	Press Alt+Enter on the keyboard.

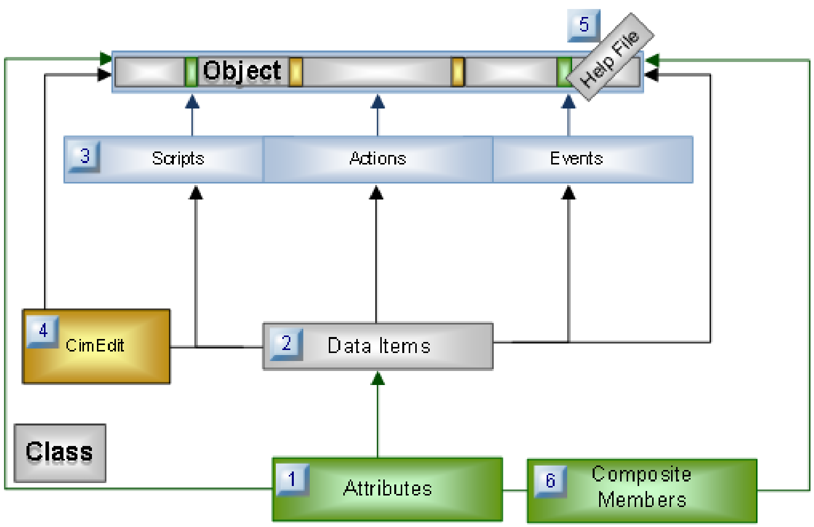
4. Right-click **Classes**.

5. Select Properties on the Popup menu.

Class Configuration

Class Configuration

You can configure a class and create objects from the class using the Class and Object dialog boxes. You can also use CIMEdit.



Before you begin, you should be familiar with your object requirements. Determine how the class will be applied and the possible range of values for the objects created from the class. You can also use Composite Classes to facilitate the creation of other classes and objects.

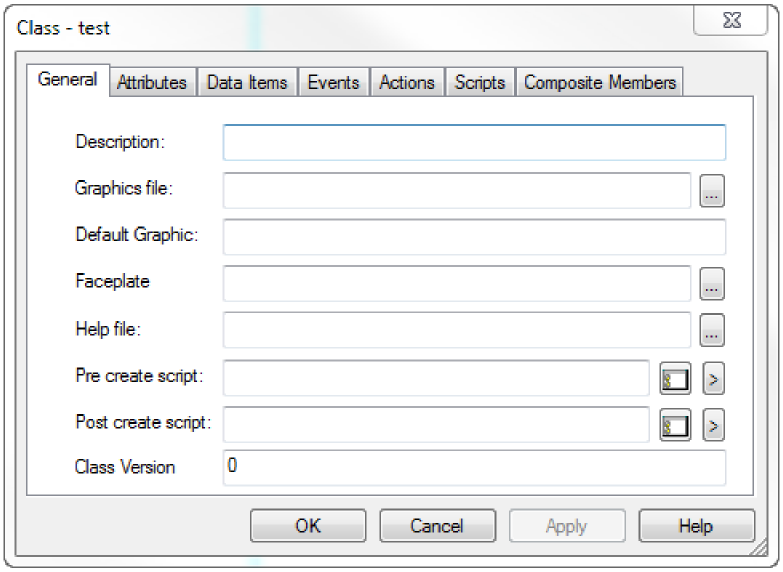
Important: You can modify a class dynamically and you can also delete a class dynamically if there are no references to the class or no object instances of that class with composite references to the class.

What you configure for the class depends on the purpose for the class. In general, once you have mapped out a plan you can configure the options for the class in any order.

Before you configure a class, you should be familiar with how to configure each of the features in a CIMPLICITY application.

General Tab in the Class Dialog Box

When you open the Class dialog, it defaults to the **General** tab, as shown below.



This field...	Contains this...
Description	Brief description of the class you are creating.
Graphics file	CimEdit screen name that will be associated with the class. The selected CimEdit screen contains the source objects that are available to object designers.
Default Graphic	Name of the CimEdit group object that displays as the default when a class object is dragged to a new CimEdit screen. Result: When an object that is created from the class is dragged from the Workbench into a new CimEdit screen, a class object graphic, which is linked to the class source graphic display
Faceplate	Specifies the CimEdit screen name. if the screen object used for a class has a mouse up event, this entry overrides the mouse up action that would have been created for the class faceplate behavior. There continues to be a faceplate menu item in the right mouse click menu for the screen object at runtime. The faceplate field can be left blank. It is not a required field.
Help file:	Custom help (.hlp) file associated with a class. The help file assists bject designers when they are configuring point objects based on the point class. In order to make the help file specific to the class you are creating, it is recommended that you write the help file after you configure the other class components. Result: When an object designer presses F1 or clicks Help in the Object dialog box, the table of contents for the specified help file displays.
Pre create script	See 3.1.3, Pre Create Script *(page 81)*.
Post create script	See 3.1.4, Post Create Script *(page 83)*.
Class Version	The default entry is **0**. Enter **1** to prompt the system to increment the class version each time data members, or a composite member, is added.

1. Type or select entries for each of the fields, as follows:
2. Click **Apply**.
3. Select the other tabs and enter or select data as required. Click the buttons below for information.

1. Class Attributes

1. Class Attributes

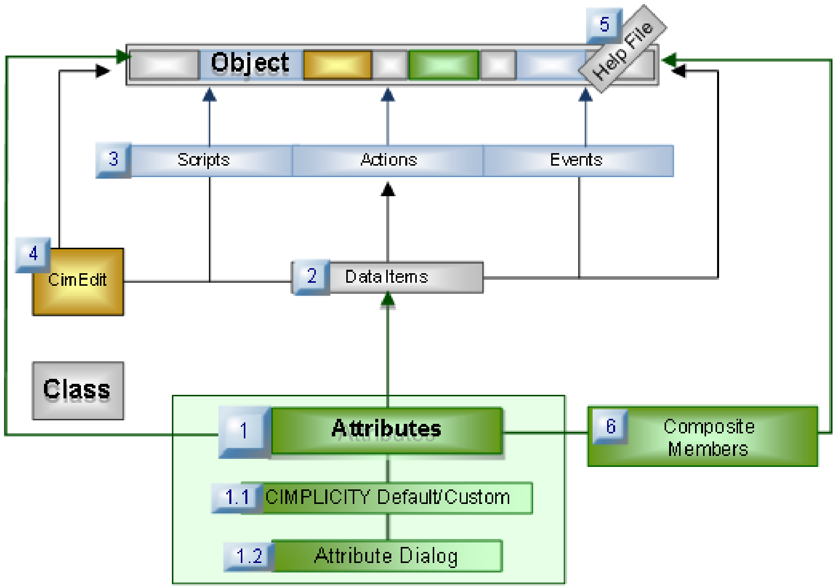
- Overview: Class attributes.
- Overview: Attribute included in an instantiated point.
- Configuration: Class attributes.

Overview: Class Attributes

Class attributes:

- Can be used to drive the way that objects are created and store information.
- Provide the tools that enable an object designer to quickly enter custom specifications when creating objects from a selected class.

One or more attributes will be used throughout the class configuration.



1.1 (page 20)	CIMPLICITY class attributes.
1.2 (page 22)	Class attribute configuration.

Overview: Attribute Included in an Instantiated Point

Attributes provide

A (page 17)	Class: Attribute configuration.
B (page 19)	Object: Object configured.
C (page 20)	Instantiated: Attributes in Point Properties dialog boxes.

1. Class: Attribute Configuration

Attributes that are CIMPLICITY pre-defined or custom attributes are configured for the class and assigned to data items. If a class references another class (Composite Classes), the class that does the references will have the attributes of the class that is being referred to.

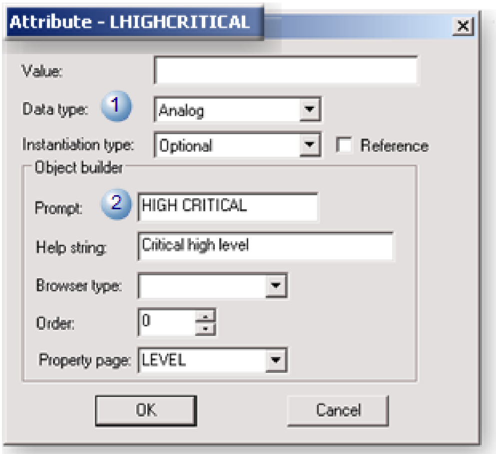
Attribute Configuration

Attributes are created and configured in an Attribute dialog box. Values can be assigned at this level or at the object level.

Example

An analog attribute, named `LHIGHCRITICAL` is created in the TANKV class.

Two attribute features are as follows.



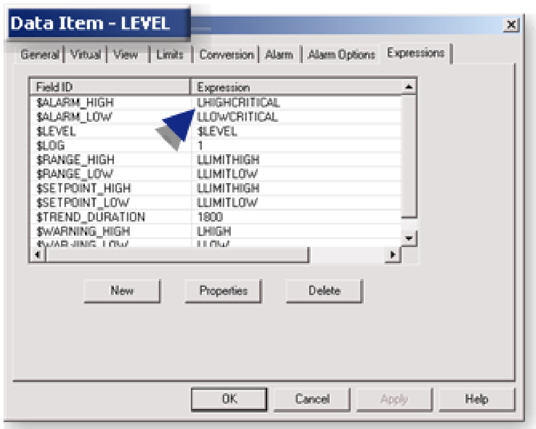
1	Data type	Analog
2	Prompt	HIGH CRITICAL Note: The prompt will be the field label in the Object dialog box.

Attribute Assignment

Attributes are assigned to data items.

Example

The attribute `LHIGHCRITICAL` is assigned as an expression to the `$Alarm_High` field for a data item LEVEL.



1. Object: Object Configured

Once an object is created attributes can be assigned values in its Object dialog box.

Object Created

A class is selected to create an object.

Example

A class object created from the class TANKV is named: STATION04.

Attribute Value Assigned

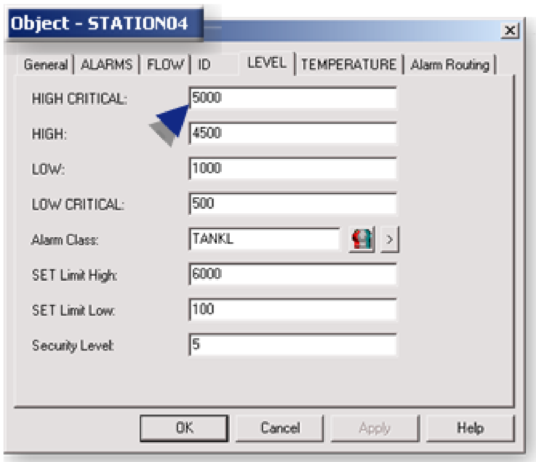
Fields that represent the configured attributes display in the Object dialog box that opens when the object is created.

Values that are entered in the Object dialog box fields will be assigned to instantiated points.

Note: Data item configuration will determine which points are assigned which attributes.

Example

5000 is entered in the **HIGH CRITICAL** field for the STATION04 object.

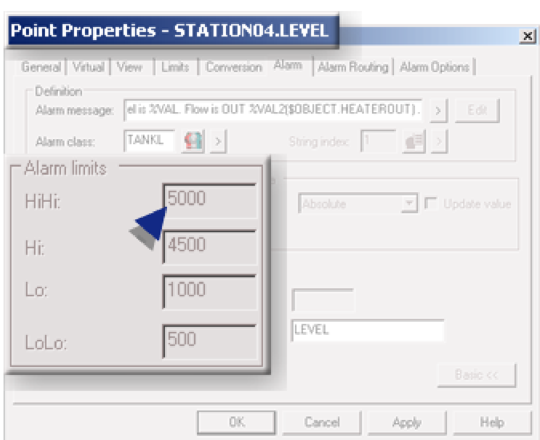


1. Instantiated: Attributes in Point Properties Dialog Boxes

Values that are entered in the Object dialog box display as read-only values for assigned instantiated points.

Example

The Alarm High field for the instantiated point LEVEL displays the read-only 5000 value.



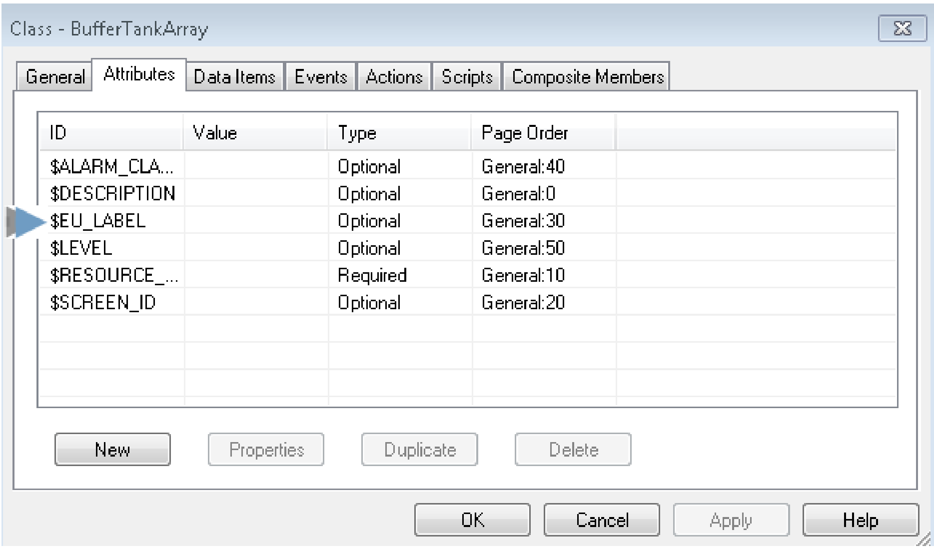
Configuration: Class Attributes

1.1 (page 20)	CIMPLICITY class attributes.
1.2 (page 22)	Class attribute configuration.

1.1. CIMPLICITY Class Attributes

CIMPLICITY classes include default CIMPLICITY attributes, which are listed on the Attributes tab in the Class dialog box.

Important: Default CIMPLICITY attributes begin with $.



CIMPLICITY class attributes are:

Class Attribute	Provides the:
$ADDRESS	Base address for data items. **Important:** $ADDRESS can also be used to support address adjustment expressions.
$ALARM_CLASS	Default alarm class for data items with alarms. **If not specified:** Alarms will not be created for the data items (object points). **Note:** The class developer can change $ALARM_CLASS from **Optional** to **Required** to override this default behavior.
$DESCRIPTION	Default description for all data items.
$DEVICE_ID	Default Device ID for all data items. **Important:** Each class object supports one device ID only.
$EU_LABEL	Default engineering units label for all data items.
$LEVEL	Default security level for all data items.
$RESOURCE_ID	Default Resource ID for all data items.
$SCREEN_ID	Default Screen ID for all data items.
$TYPE	Displays static objects that cannot be modified on the object configuration screens. The objects display as disabled.

1.2. Class Attribute Configuration

1.2. Class Attribute Configuration

Class attributes are created through the Attributes tab in the Class dialog box and defined in the Attribute dialog box.

1.2.1 *(page 22)*	Open an Attribute dialog box.
1.2.2 *(page 23)*	Define a class attribute.

1.2.1. Open an Attribute Dialog Box

Buttons on the Attributes tab enable you to open an Attribute dialog box for a new or existing attribute.

- New attribute.
- Existing attribute.
- Duplicate attribute.

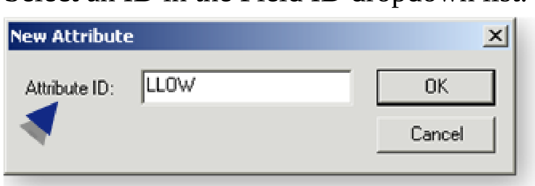
Note: You can delete any attribute by selecting an ID and clicking Delete.

New Attribute

1. Click **New** on the Attributes tab in the Class dialog box.

   The New Attribute dialog box opens.

2. Select an ID in the Field ID dropdown list.

   Note: If a Field ID is already listed on the Expressions tab, it will not be included in the dropdown list.

3. Click **OK**.

   Result: An Expression dialog box for the new field expression opens.

Existing Attribute Properties

4. Select an existing attribute.
5. Click Properties.

Result: The Attribute dialog box with specifications for the existing attribute opens.

Duplicate Attribute

6. Select an existing attribute.
7. Click Duplicate.

A Duplicate Attribute dialog box opens.

The selected source data item displays.

8. Enter a unique name for the **Destination attribute.**

9. Click OK.

The Attribute dialog box with specifications for the source attribute opens.

1.2.2. Class Attribute Definition

A (page 23)	Configuration: Attribute.
B (page 28)	Result: Object dialog box.

1. Configuration: Attribute

The attribute dialog box provides the tools to define the following.

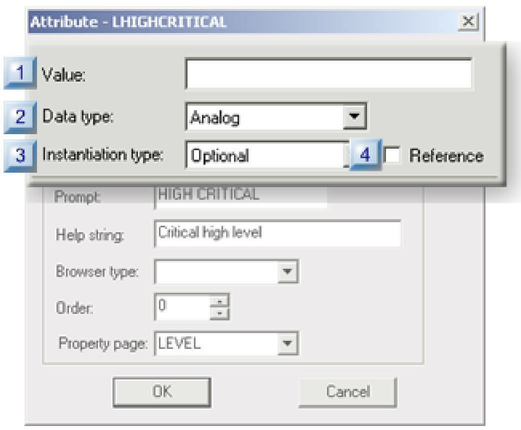
- Definition: Attribute
- Definition: Attribute fields for the Object dialog box.

Definition: Attribute

The top part of the dialog box provides fields for you to define the attribute. Your definition can provide default values and determines the options available to an object designer when creating an object from the class.

Open either a new or existing Attribute dialog box in which you will fill in or modify the top part.

Specifications are as follows.



rect 0, 31, 20, 51 (page 24)
rect 0, 59, 20, 79 (page 25)
rect 1, 83, 21, 103 (page 25)
rect 216, 82, 236, 102 (page 25)

1 (page 24)	Value
2 (page 25)	Data type
3 (page 25)	Instantiation type
4 (page 25)	Reference

1	Value

(Optional) Enter the default value for the attribute.

Acceptable values:

- Must be within the range for the attribute type

- Are limited to 80 characters.
- Cannot contain the vertical bar '|' character.

2	Data type

Select the data type of the value from the drop-down list:

Data types are:

- Analog
- Boolean
- String

3	Instantiation type

Select the option from the drop-down list that will control how a user will deal with the attribute when an object is being created:

Option	When an object is created, a user:
Hidden	Will be in the class object; however, it will not be listed in the Object dialog box.
Optional	Can override a class default value, if there is one. The attribute does not have to be specified to create the object.
Readonly	Will be listed in the Object dialog box; however, the field will be disabled, preventing users from editing it.
Required	Must provide a value for this attribute. Failure to provide a value when creating an object will generate an error.
Static	• Cannot change the value. • All objects of this class will have the same value for this attribute. • The value does not display in the Object dialog box.
Static Display	• Cannot change the value. • All objects of this class will have the same value for this attribute. • The value displays as a read-only field in the Object dialog box.

4	Reference

Check **Reference** if you want to identify the value of this attribute as the value of another attribute.

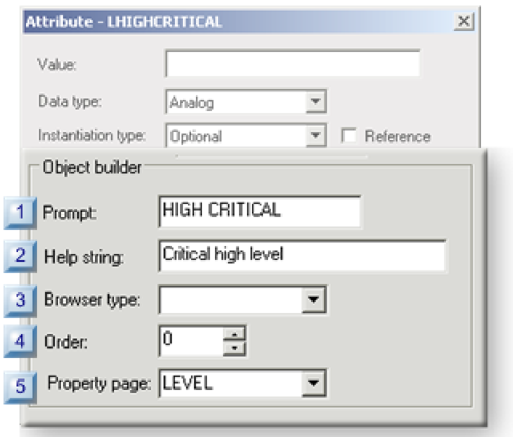
Note: The **Value** field in the Attribute dialog box must contain the attribute ID of the value to be referenced.

Definition: Attribute Fields for the Object Dialog Box

An object designer configures an object based on the class you are creating in an Object dialog box. The specifications you make in the Object Builder Group of the class attribute's Attribute dialog box determine where and how the attribute appears in the Object dialog box.

1. Open the appropriate Attribute dialog box (if it is not open), in which you will fill in or modify the fields in the Object Builder box.

2. Fill in the fields as follows.



rect -5, 117, 21, 143 (page 26)
rect -5, 141, 21, 167 (page 27)
rect -4, 168, 22, 194 (page 27)
rect -4, 197, 22, 223 (page 27)
rect -4, 225, 22, 251 (page 27)

1 (page 26)	Prompt
2 (page 27)	Help string
3 (page 27)	Browser type
4 (page 27)	Order
5 (page 27)	Property page

1	Prompt

(Optional) A meaningful description that is up to 20 characters.

Your entry appears as the field name for the class attribute in the Object dialog box.

The class attribute ID appears as the field name if there is no entry.

2	Help string

(Optional) A tip up to 80 characters.

Your entry appears as a tool tip in the Object dialog box when the user places the cursor over the field.

3	Browser type

(Optional) A user has access to the selected browser in the Object dialog box.

The Browser's button will appear to the right of the field in the Object dialog box.

Browser Options	Button that will display
Alarm Class	
Resource	
Device	
Role	
Point	
User	
Port	

4	Order

The order in which the field will appear on the Object dialog box tab in the field.

- Lower numbers appear before higher numbers.
- Class attributes with the same order number (for the same tab) are sorted alphabetically by Prompt (field name).

5	Property page

Selects the tab on which the attribute will display in the Object dialog box as follows.

Do one of the following.

- Select the General which is the default tab.

Note: The General tab, has a limit of 8 attributes (fields).

CIMPLICITY attributes can be moved to another tab make room on the General tab for Custom attributes.

- Enter a new name. The name you enter will appear as a tab in the Object dialog box.

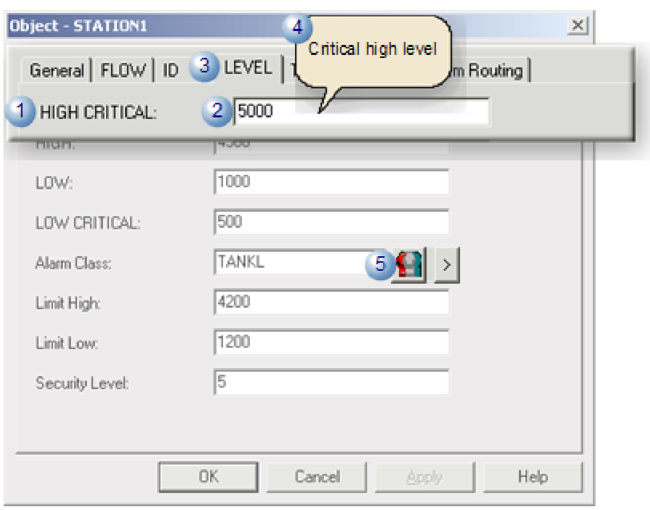
Note: Additional tabs have a limit of 9 attributes each.

3. Click OK or Cancel.

OK	A new class attribute is added to the list of attributes and existing attributes are modified. In addition: • Existing objects based on this class are reevaluated to reflect the new class structure. • You will be prompted to configure new required values for each existing object. • Expressions that contain an existing attribute are re-evaluated. • Points are modified when a configuration update is performed on the project.
Cancel	Cancels the procedure.

4. Continue creating the attributes that should be included in the class.
   a. Result: Object Dialog Box

When an Object dialog box is opened for a class object, the attribute's location and specifications reflect entries in the Attribute dialog box.



	Field in:	Purpose in:
	Class Attribute Dialog Box	Object Dialog Box
1	Both the: • Prompt • Order	Specifies the: • Field Label • Order in which the field displays on the Attribute dialog box tab.
2	Both the: • Data Type • Instantiation Type	Control the: • Data type allowed for the field value • If a field entry is optional or required

3	Property Page	Tab that the attribute is located on.
4	Help String	Tool tip that displays then the mouse hovers over the attribute field.

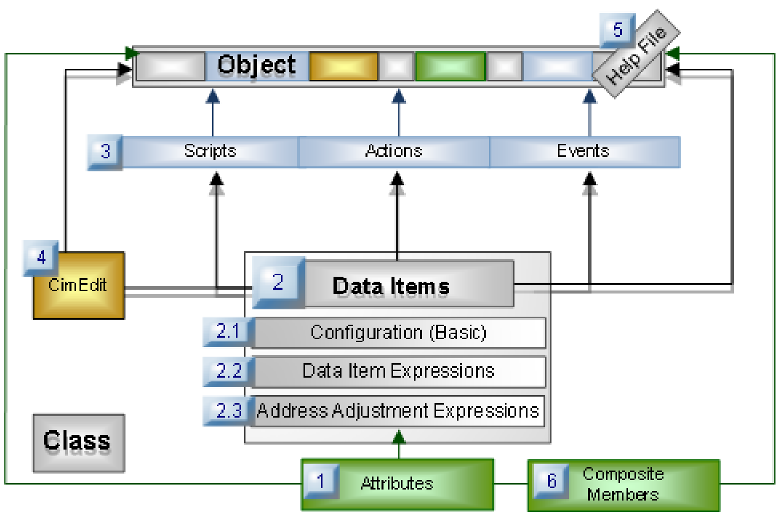
2. Data Items

2. Data Items

- Overview: Data item configuration.
- Overview: Data item instantiated into a point.
- Configuration: Data items.

Overview: Data item Configuration

A data item is a definition that becomes a CIMPLICITY point when an object is created from the class.



2.1 (page 32)	Data item configuration (basic).
2.2 (page 52)	Data item expressions.
2.3 (page 65)	Address adjustment expressions.

A data item:

- Can be used in any application that supports points, for example:
- Database Logger

- Event Editor
- BCE
- Point Control Panel
- CimView
- Other CIMPLICITY Software Options
- Can include a subset of expressions of type: string or numeric. Note that a limited number of string substitutions and numeric expressions are supported.
- Displays in the Workbench right pane as a point with its associated object.

Overview: Data Item Instantiated into a Point

Data items are instantiated into CIMPLICITY points, as follows.

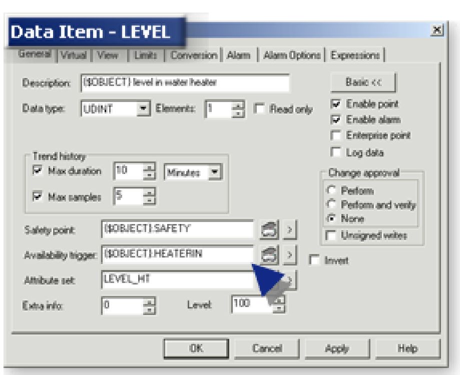
A *(page 30)*	Class: Data item configuration.
B *(page 30)*	Object: Object created.
C *(page 31)*	Instantiation: Data item to point.

1. Class: Data Item Configuration

Data items are configured in Data Item dialog boxes that are opened through the Class dialog box.

Example

An analog data item named `LEVEL` is created.



Note: Data item ID's *(page 33)* are listed on the Data Items tab in the Class dialog box.

1. Object: Object Created

A class is selected to create an object.

Example

A class object created from the class TANKS is named: STATION01.

1. Instantiation: Data Item to Point

Data items are instantiated into points for the new object.

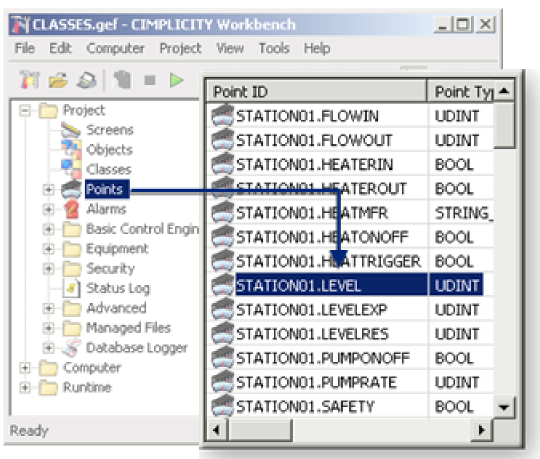
Workbench

The instantiated points are listed in the Workbench right pane.

Example

STATION01 instantiated points are listed in the Workbench.

The data item LEVEL instantiated into a class object point is named: STATION01.LEVEL.



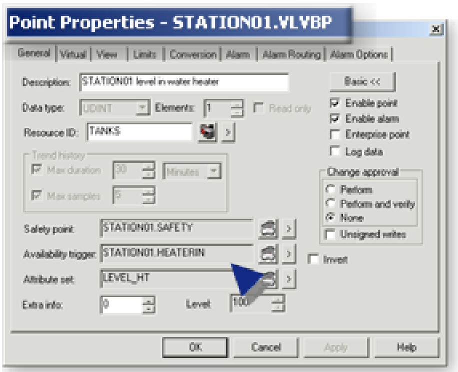
Point Properties Dialog Box

An object's Point Properties dialog box can include several fields that class/object configuration have caused to be read-only.

Example

Several data items have been entered in the LEVEL Data Item dialog box.

Those fields in the STATION01.LEVEL Point Properties dialog box are read-only.



Configuration: Data items

2.1. Data Item Configuration (Basic)

2.1. Data Item Configuration (Basic)

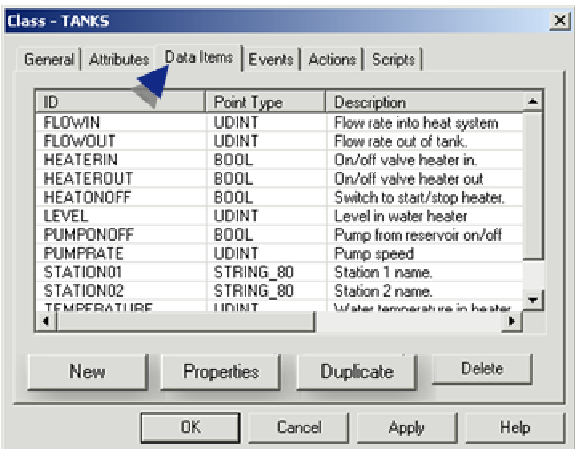
The basic configuration for data items is very similar to point configuration.

The core issue in the basic configuration is to be aware of what happens when the data item is instantiated into a point.

2.1.6 (page 47)	Device data item memory usage options.

2.1.1. Open a Data Item Dialog Box

Buttons on the Data Items tab enable you to open a Data Item dialog box for a new or existing data item.



rect 15, 217, 92, 239 (page 33)
rect 99, 218, 176, 240 (page 34)
rect 184, 218, 261, 240 (page 34)

- New data item.
- Existing data item.
- Duplicate data item.

New Data Item

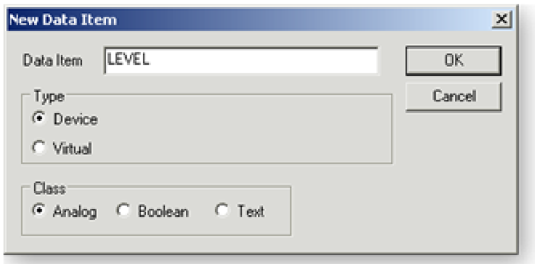
Each new data item represents a CIMPLICITY point definition. Points are created when an object is created. The points are associated with the object.

1. Select the Data Items tab in the Class dialog box.

2. Click **New**.

   A New Data Item dialog box opens.

3. Identify the new data item as follows.



Field / Radio Button	Description
Data Item	A unique name for the data item. The data item template will become a point for a class object. The Data Item ID: • Is limited to 16 characters. • Can be composed of • Uppercase alphabetic characters, • Numeric characters and • Underscores. • Must begin with an alphabetic character. • Cannot contain spaces.
Device/Virtual	Selected device or virtual data item definition will become a device or virtual point.
Analog/Boolean/ Text	Selection for Data item class will be the point's class for an object.

4. Click **OK**.

   Result: The Data Item dialog box opens. The available tabs and options in the dialog box depend on your Type and Class selections.

   Existing Data Item Properties

5. Select an existing data item.
6. Click Properties.

   Result: The Data Item dialog box with specifications for the existing data item opens.

   Duplicate Data Item

7. Select an existing data item.
8. Click Duplicate.

   A Duplicate Data Item dialog box opens.

   The selected source data item displays.

9. Enter a unique name for the **Destination data** that adheres to the data item ID requirements *(page 34)*.

10. Click OK.

The Data Item dialog box with specifications for the source data item opens.

2.1.2. Resource Configuration

A Data Item dialog box does not include a **Resource ID** field.

The resource is assigned to the object in the Object dialog box.

The resource is included, as follows, for instantiation.

A *(page 35)*	Class: Data item configuration
B *(page 36)*	Object: Object defined.
C *(page 37)*	Instantiation: Point properties result.

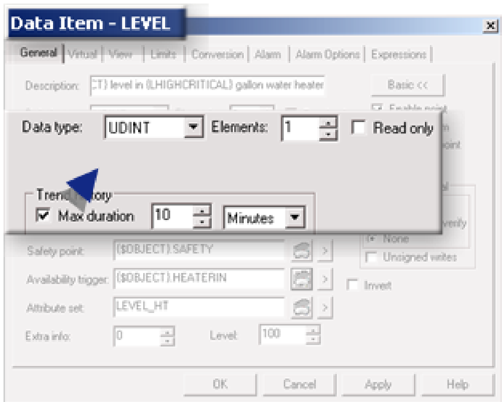
1. Class: Data Item Configuration

A resource cannot be entered for a data item.

There is no **Resource ID** field in the Data Item dialog box.

Example

A Data Item dialog box named LEVEL has no **Resource ID** field.



1. Object: Object Defined

A Resource ID is entered at the object level.

Object Created

Class objects are created from the class.

Example

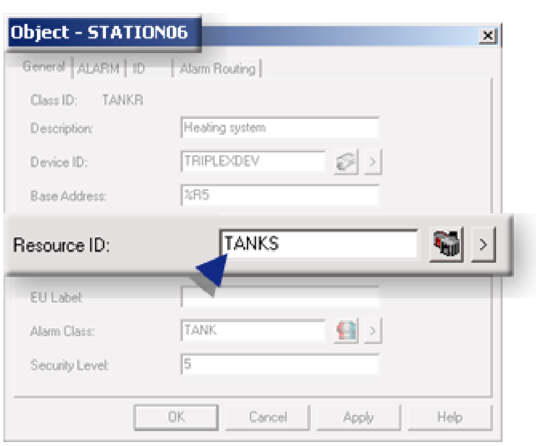
A class object created from the class TANKR is named: STATION06.

Attribute Value Assigned

A Resource ID must be entered in the Object dialog box.

Example

The STATION06 Object dialog box has TANKS entered in the **Resource ID** field.

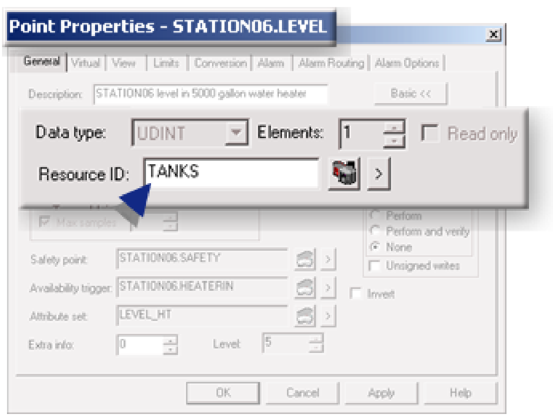


1. Instantiation: Point Properties Result

A **Resource ID** field in the Point Properties dialog box, displays the resource ID that is assigned to the object.

Example

The **Resource ID** field in the STATION06.LEVEL Point Properties dialog box displays the TANKS resource.



2.1.3. Data Item Fields in Data Item Dialog Boxes

Data Item Fields

Fields that take points in a Point Properties dialog box, take data items in a Data Item dialog box.

The following occurs when a data item is selected for a Data Item field.

A *(page 37)*	Class: Fields for data items.
B *(page 38)*	Object: Object created.
C *(page 38)*	Instantiation: Point properties result.

1. Class: Data Item Configuration

Data items can be entered in data item fields.

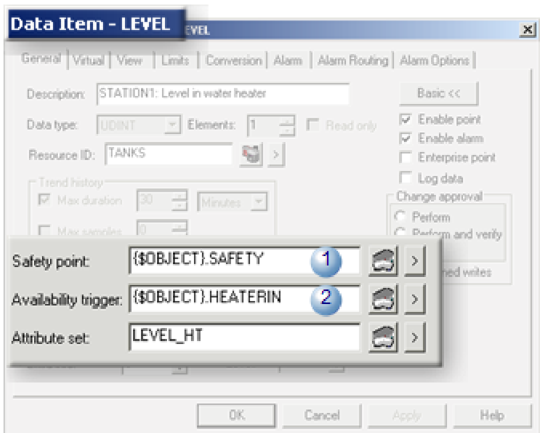
CIMPLICITY automatically inserts `{$OBJECT}`. before the data item ID.

`{$OBJECT}` will become the object name when the data item is instantiated into a point.

Example

In a LEVEL data Item dialog box , data items are selected for the **Safety point** and **Availability trigger** fields.

The field entries with CIMPLICITY's `{$OBJECT}` additions are as follows.



	Field	Data Item Entry
1	Safety point	{$OBJECT}.SAFETY
2	Availability trigger	{$OBJECT}.HEATERIN

1. Object: Object Created

Class objects are created from the class.

Example

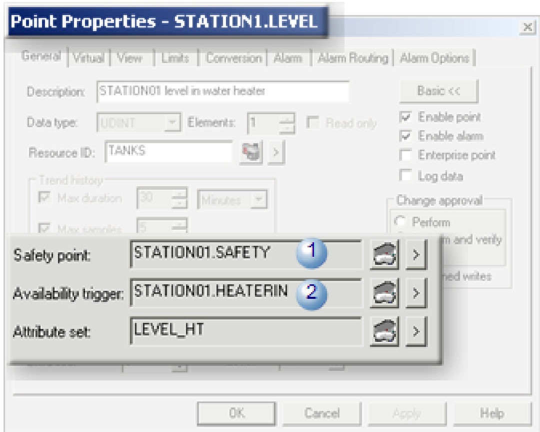
A class object created from the class TANKS is named: STATION01.

1. Instantiation: Point Properties Result

All point fields in the object's Point Properties dialog boxes display read-only instantiated points.

Example

In the Point Properties dialog box, the **Safety point** and **Availability trigger** fields for the instantiated STATION01.LEVEL point are as follows.



	Field	Instantiated Point
1	Safety point	STATION01.SAFETY
2	Availability trigger	STATION01.HEATERIN

2.1.4. String Substitutions

- Overview: String substitutions.
- Configuration: String substitution.

Overview: String Substitutions

String substitutions provide the capability to customize text values at the object level.

String Substitution Sources

Sources for string substitutions are available for the following.

- All class attributes.

Note: If an attribute is not initially a text type, it is treated as a text type when it is used as a string substitution.

- CIMPLICITY predefined variables, which are:

Variable	Description	Example (When Instantiated)
$OBJECT	Object ID	STATION01
$ID	ID of the data item	LEVEL.
$CLASS	Class ID	TANKS

Limitations: String substitution

Nested substitution is not supported.

Example

You have created the following variables:

VAR1="{**VAR2**}"

VAR2="**HELLO WORLD.**"

You enter the string substitution **{VAR1}**

The expression after substitution will be the string **"{VAR2}".**

The expression will not be reevaluated to yield the result "**HELLO WORLD.**"

Configuration: String Substitution:

A *(page 40)*	Class: Data item configuration
B *(page 41)*	Object: Object defined.
C *(page 42)*	Instantiation: Point properties result.

1. Class: Data Item Configuration

Enter string substitutions in any of the following fields in a Data Item dialog box.

Use brackets **{}** to enclose substitution entries.

Data Item Type	Fields Supporting String Substitution
All	Description Safety Point Screen Availability Trigger Measurement Units Label
All Analog and Boolean	Point Enumeration
Derived	Expression Reset Point
Device	Address Device ID
Point Alarm	Alarm Message Alarm Class Deviation Point
Text	Initialization value

Example

The description for a LEVEL data item that is entered in the Description field on the Data Item dialog box>General tab includes two string substitutions.

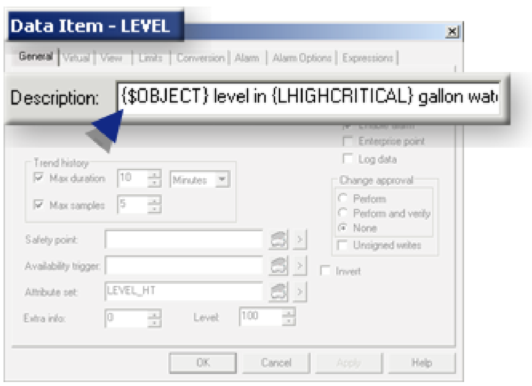
The description is as follows.

{`$OBJECT`} level in {`LHIGHCRITICAL`} gallon water heater.

Where

{`$OBJECT`} is the CIMPLICITY predefined variable.

{`LHIGHCRITICAL`} is a custom analog attribute for the Alarm High value.



1. Object: Object Defined

String values can be entered in the Object dialog box.

Object Created

Class objects are created from the class.

Example

A class object created from the class TANKV is named: STATION04.

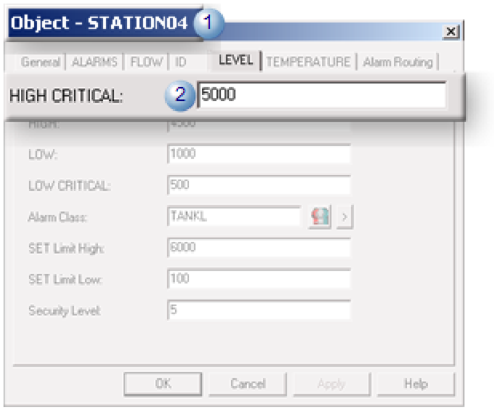
Attribute Value Assigned

Values that are entered in the fields created from the class attributes will replace the string substitution attributes entered in the Data Item dialog boxes.

Example

A class object created from the class TANKV is named: STATION04

Values will replace the string substitutions that were entered in the LEVEL Data Item dialog box>**Description** field are the following.



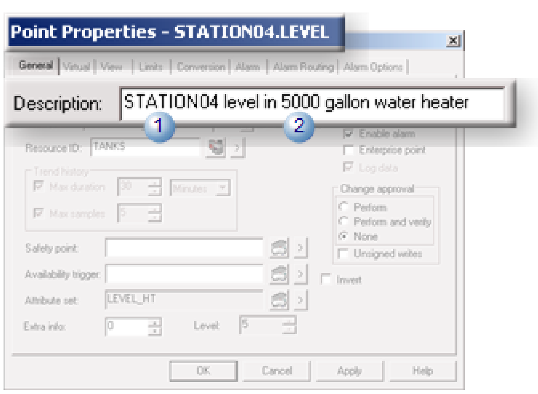
	String Substitution	Value
1	($OBJECT)	STATION04
2	(LHIGHCRITICAL)	5000

1. Instantiation: Point Properties Result

Fields in instantiated Point Properties dialog boxes display instantiated values for string substitutions. Fields that display some substitutions continue to be enabled so they can be edited at the instantiated point level.

Example

The **Description** field in the instantiated STATION04.LEVEL Point Properties dialog box displays the substituted values. The field is read-write so these values can be changed.



	String Substitution	Value
1	($OBJECT)	STATION04

2	(LHIGHCRITICAL)	5000

2.1.5. Device Data Item Device ID and Address

The Data Item dialog box Device tab provides the flexibility to use the class template for different object requirements.

- Device ID field.
- Addressing.

Device ID Field

A Data Item dialog box for a device does not include a **Device ID** field.

The device is assigned to the object in the Object dialog box.

Important: Each object can support only one device.

The device is included, as follows, for instantiation.

A *(page 43)*	Class: Data item configuration
B *(page 44)*	Object: Object defined.
C *(page 45)*	Instantiation: Point properties result.

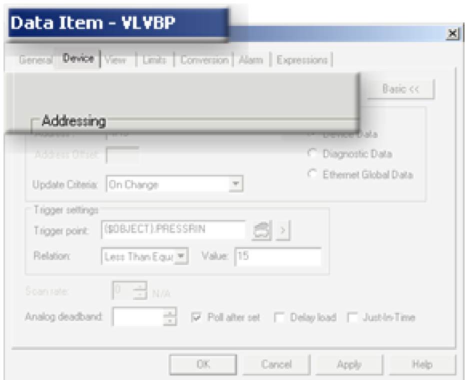
1. Class: Data Item Configuration

A device cannot be entered for a data item.

There is no **Device ID** field in the Data Item dialog box.

Example

A device Data Item dialog box named VLVBP has no **Device ID** field.



1. Object: Object Defined

A Device ID is entered at the object level.

Object Created

Class objects are created from the class.

Example

A class object created from the class TANKS is named: STATION01.

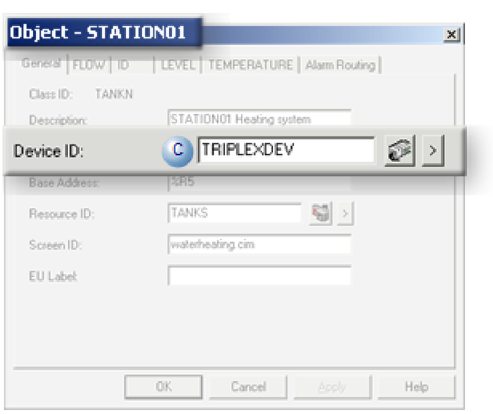
Attribute Value Assigned

A Device ID must be entered in the Object dialog box.

Note: If you plan to create device data items and objects already exist, enter a device ID in each Object dialog box before you create the device data items. If you do not, you will not be able to apply the changes to the class. You can change the device at anytime.

Example

The STATION01 Object dialog box has TRIPLEXDEV entered in the **Device ID** field.

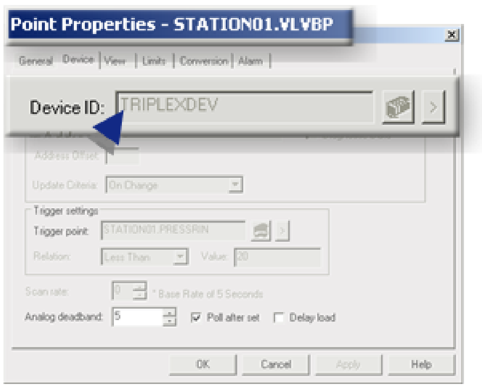


1. Instantiation: Point Properties Result

A **Device ID** field, which is read-only in the Point Properties dialog box, displays the device ID that is assigned to the object.

Example

The Device ID read-only field in the STATION01.VLVBP Point Properties dialog box displays the TRIPLEXDEV device.



Address field

An address does not have to be entered in a Data Item dialog box for a device data item.

An address is instantiated with an object as follows.

A *(page 45)*	Class: Data item configuration
B *(page 46)*	Object: Object created.
C *(page 47)*	Instantiation: Point properties result.

1. Class: Data Item Configuration

An address does not have to be entered in the **Address** field for a device data item but if an address is entered:

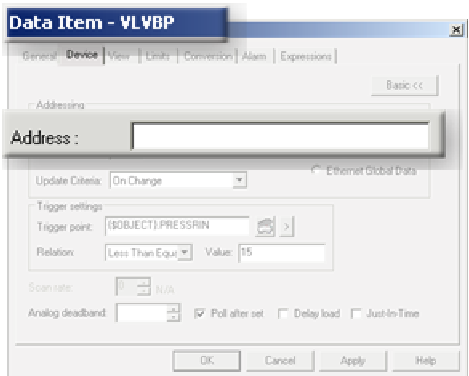
* It overrides an address entry in the Object dialog box.

* It is applied to the instantiated point for every object `{$OBJECT}`.<point ID> in the class.

* Do not configure the $ADDRESS_ADJ data item. A Device Base address should only be specified at Class level attribute or Object level attribute when instantiated or modified at Object properties.

Example

The **Address** field in a device Data Item dialog box for a device data item named VLVBP has no entered address.



1. Object: Object Configuration

A different base address can be entered for each object in the class.

Object Created

Class objects are created from the class.

Example

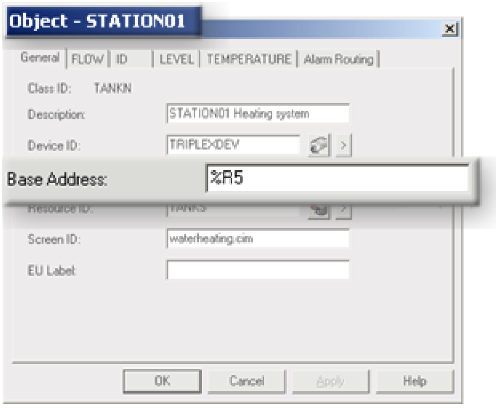
A class object created from the class TANKS is named: STATION01.

Attribute Value Assigned

An address is entered in the Address field in the Object dialog box.

Example

STATION01 is assigned the %5R address.



1. Instantiation: Point Properties Result

The read-only Address field in the instantiated point's Point Properties dialog box displays, in the following priority, the address entered in the:

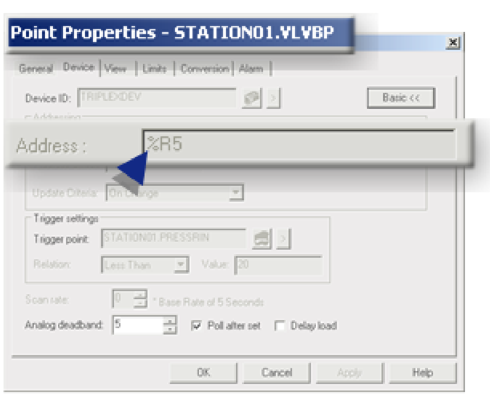
1. Data Item dialog box.
2. Object dialog box.

   Example

   The read-only **Address** field in the STATION01.VLVBP Point Properties dialog box displays the %5R address entered in the Object dialog box.

   Note: If a different address, e.g. %8R was entered in the STATION01.VLVBP Data Item dialog box, that address would display in the Point Properties dialog box.

   The **Address** field for other STATION01 object points display %5R or addresses entered in their Data item dialog box.



2.1.6. Device Data Item: Memory Usage Options

The Device tab in the Data Item dialog box provides Just-In-Time as an additional memory usage option.

Memory options for instantiated points perform as follows

- Delay load
- Just-In-Time
- Standard Memory Load

Delay Load

Delay loadis available in point configuration to help conserve memory.

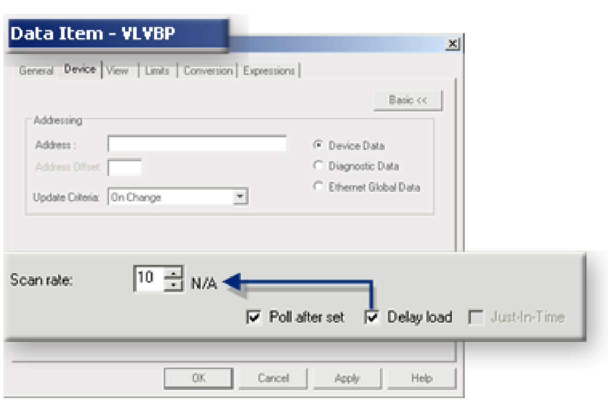
The point is not loaded into memory in the Point Manager or Devcom at project startup. Rather, the point is loaded into memory only when it is demanded by an application. When the point ceases to be demanded it will be unloaded from memory in the Point Manager and Devcom.

Class: Data item configuration
Instantiation: Point properties result.
Benefits and disadvantages: Delay Load

1. Class: Data item Configuration

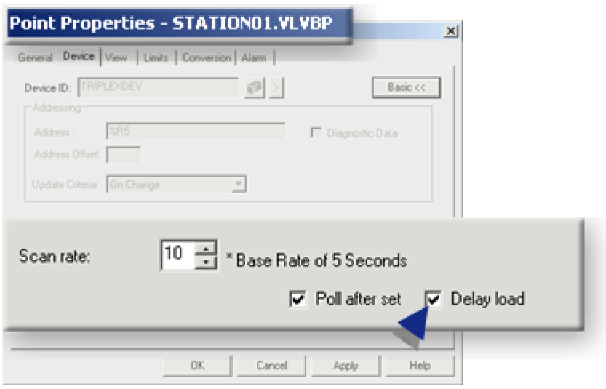
When Delay Load is checked in a Data Item dialog box

- Just-In-Time is disabled.
- A scan rate is required.



1. Instantiation: Point Properties Result

Delay Load is read/write in the object's Point Properties dialog box, whether or not it is checked in the Data Item dialog box.



1. Benefits and Disadvantages: Delay Load

Delay Load benefits and disadvantages are the same whether the point is included in an instantiated object or is created in the project's Workbench. They are as follows.

Benefits	Disadvantages	Non-Supported Functions
• Loaded when needed • No memory usage until point is loaded • Written to point database • Available in point list • Definition can be modified	• Slower access • Configuration updates can be lengthy on larger systems • Trigger settings and Alarm options are not available	• Trend buffering • Alarming • Cannot be used within another point's configuration, e.g. Safety point, Trigger, Availability Trigger point, etc. • Cannot be used in a calculation point equation

Just-In-Time

When Just-In-Time (JIT) is checked, data item instantiated point configuration is created by the Point Manager when an application requests it.

When the application is done, the data item is removed from the Point Manager. It is not written to the point database.

JIT Data Items can be used in:

- CimView,
- Point Control Panel,
- Trending, and
- Scripting.

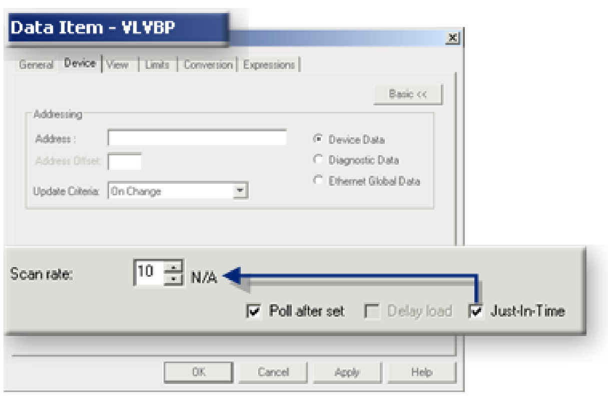
Important: Change approvalis not supported with Just-In-Time points.

A (page 50)	Class: Data item configuration
B (page 50)	Instantiation: Point properties result.
C (page 51)	Benefits and disadvantages: Just-In-Time

1. Class: Data Item Configuration

When Just-In-Time is checked:

- Delay Load is disabled.
- A scan rate is required.

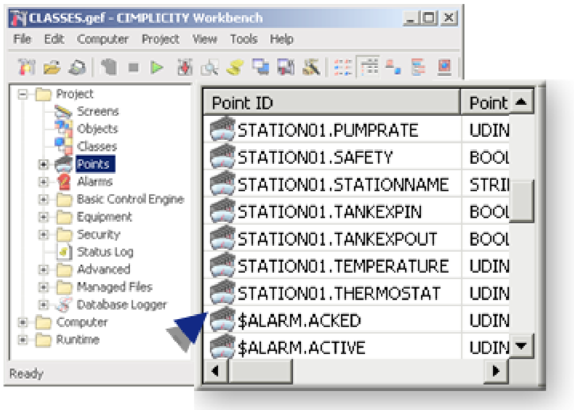


1. Instantiation: Point Manager Result

An instantiated object does not list a point for the Just-In-Time data item in the Workbench.

Example

The data item VLVBP in a class TANKR was configured to be a Just-In-Time point. It is not included in the instantiated point list when an object is created from the class.



1. Benefits and Disadvantages: Just-in-Time

Just-In-Time benefits and disadvantages are as follows.

Benefits	Disadvantages	Non-supported Functions
• Created on demand when needed • No memory usage until point is created • Project uses less disk space • Point database is reduced	• Not available in point lists • Slowest access • Device points only • Does not reside in point database • Definition cannot be modified • Exclusive to classes • Trigger settings and Alarm options are not available **Note:** Are not recommended for points that are accessed often.	• Trend buffering • Alarming • Attribute set • Analog deadband • Cannot be used within another point's configuration, e.g. Safety point, Trigger, Availability Trigger point, etc. • Cannot be used in a calculation point equation • Cannot use another point within its configuration, e.g. Safety point, Trigger, Availability Trigger point, etc. • Cannot be marked as an enterprise point.

Standard Memory Load

When Delay Load and Just-In-Time check boxes are clear in a Data Item dialog box, an instantiated point has standard memory usage.

Standard memory load benefits and disadvantages are the same whether the point is included in an instantiated object or is created in the project's Workbench. They are as follows.

Benefits	Disadvantages	Non-supported Functions

• Accessible any time • Quick access • Written to point database • Available in point list • Definition can be modified	• Uses memory in point database • Project uses more memory • Configuration updates can be lengthy on larger systems	None—all functions supported

2.2. Data Item Expressions

2.2. Data Item Expressions

Data item expressions are evaluated from class using object attributes or constants to derive the value of a data item field.

Note: The data items become points when the object is instantiated.

- Overview: Data item expressions.
- Configuration: Data item expressions.

Overview: Data Item Expressions

As a result, the a point that is instantiated for different objects from a single data item can include different values in the same field, e.g. Alarm High or Setpoint Low. If a value was entered directly into a data item field, that value will be the same read-only value for every point instantiated from that data item.

Example

How the value of the Alarm High field for a single data item is determined depends on whether or not it was assigned an expression or a value.

- Alarm High field assigned an expression.
- A data item includes an expression for the **Alarm High** field.
- The value for the **Alarm High** field is entered in the Object dialog box for three class objects.
- The value in the **Alarm High** field for each instantiated point is as follows.

OBJECT1	5000
OBJECT2	2500
OBJECT3	15000

- Alarm High data item field assigned a value.
- The value entered in the **Alarm High** field in a Data Item dialog box is 5000.
- There is no configuration at the Object level for the **Alarm High** field.

- The value in the **Alarm High** field for each instantiated point is as follows.

OBJECT1	5000
OBJECT2	5000
OBJECT3	5000

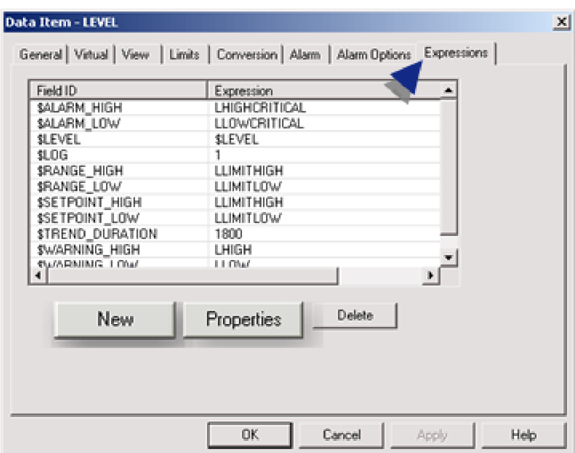
Note: A class attribute is a string. As a result the attribute cannot be entered in fields that require numeric entries, e.g. the Alarm Limits field on the data item's alarm tab.

Configuration: Data Item Expressions

2.2.1 (page 53)	Open an Expression dialog box.
2.2.2 (page 54)	CIMPLICITY Field ID's.
2.2.3 (page 54)	Field ID expression configuration.
2.2.4 (page 59)	Data item: Additional expression examples.

2.2.1. Open an Expression Dialog Box

Buttons on the Expressions tab in the Data Item dialog box enable you to open an Expression dialog box for a new or existing Field ID.



rect 113, 181, 190, 207 (page 54)
rect 33, 181, 110, 207 (page 54)

- New Field ID.

- Existing Field ID

Note: You can delete any field ID by selecting an ID and clicking Delete. If you delete data item expressions in a class, there will not be any changes made to existing data items. They will keep their last values. However, the changes will apply to any new data items you create.

New Field ID

1. Click **New** on the Expressions tab in the Data Item dialog box.

   A New Expression dialog box opens.

2. Select a field ID in the dropdown list.

   Note: If the field has already been configured for the data item, it will not be included in the list.

3. Click **OK**.

   Result: An Expression dialog box for the new field ID opens.

   Existing Field Properties

4. Select an existing field ID.

5. Click Properties.

The Expression dialog box with specifications for the existing field opens.

2.2.2. CIMPLICITY Field ID's

$ALARM_DELAYDelay interval (seconds) for all alarm states.If $ALARM_DELAY is assigned to a data item with delay fields for other alarm states (e.g. $ALARM_HIGH_DELAY),

1. $ALARM_DELAY is evaluated first.
2. Specific alarm delay entries are evaluated next and override $ALARM_DELAY values.

2.2.3. Field ID Expression Configuration

The Expression dialog box provides the fields either one operand or two operands connected by an operator.

- Numeric entries.
- Attribute entries.
- Expression with two operands

Numeric Entries

An Expression for a field ID that is assigned to a data item can be a number. The number will be entered in the selected field for the data item's instantiated point in every object created from the class.

A (page 55)	Class: Expression configuration.
B (page 56)	Object: Object creation.
C (page 56)	Instantiation: Point properties result.

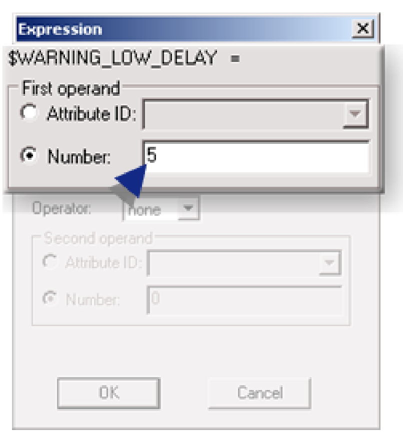
1. Class: Expression Configuration

The number that should be assigned to the field is entered in the **First operand**>**Number** field in the Expression dialog box.

Example

Every instantiated point for a data item, LEVELRES, in the TANKV class should have a 5 second alarm delay for the Warning Low (Lo) alarm.

5 is entered as the First operand for the `$WARNING_LOW_DELAY` field.



1. Object: Object Creation

When a number only is used in the expression, object level entries for that field will not affect the points that instantiated from the selected data item.

Example

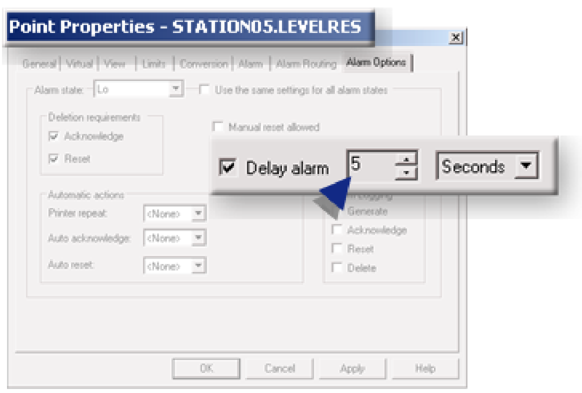
An object STATION05 is created from the TANKV class.

1. Instantiation: Point Properties Result

The number entered in the Expression dialog box displays in selected field in the instantiated point's Point Properties dialog box.

Example

The STATION05.LEVELRES Point Properties dialog box has 5 seconds entered in the **Delay alarm** field for the Warning Low (Lo) alarm level on the Alarm Options tab.



Attribute Entries

An Expression for a field ID that is assigned to a data item can be an attribute *(page 23)*. The value is entered at the object level each object can have a different value.

A *(page 57)*	Class: Expression configuration.
B *(page 57)*	Object: Field values entered.
C *(page 58)*	Instantiation: Point properties result.

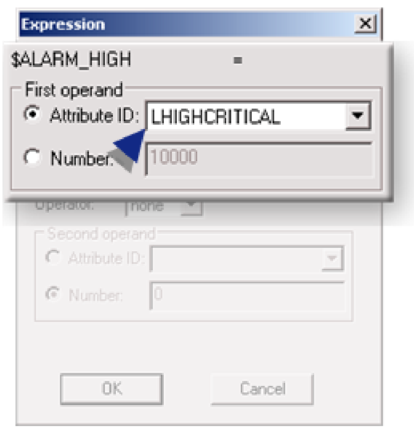
1. Class: Expression Configuration

All of the class analog attributes are listed in the Attribute ID fields' dropdown list. Any of those attributes can be applied to the selected field.

Example

The Alarm High value for a data item, LEVEL, in the TANKN class may be different for each object.

In order to apply a value entered at the object level, the LEVEL Expressions list includes the $ALARM_HIGH field, which has the custom attribute `LHIGHCRITICAL` as the expression.



1. Object: Field Values Entered

When an attribute is selected as a data item's field expression, the data item's field value is assigned at the object level

Object Created

Class objects are created from the class.

Example

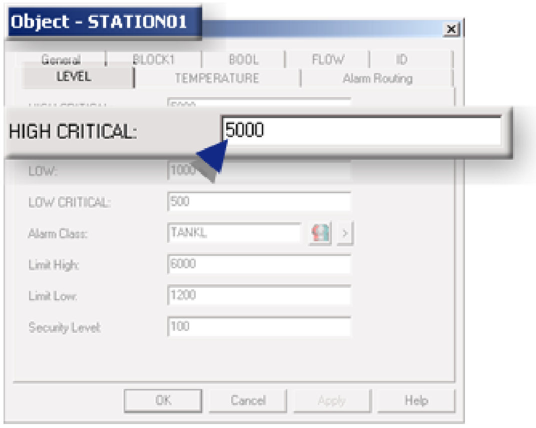
A class object created from the class TANKN is named: STATION01.

Field Expressions Assigned Values

Values that are entered in the Object dialog box fields will be applied to the instantiated points that have the attributes applied to field ID's.

Example

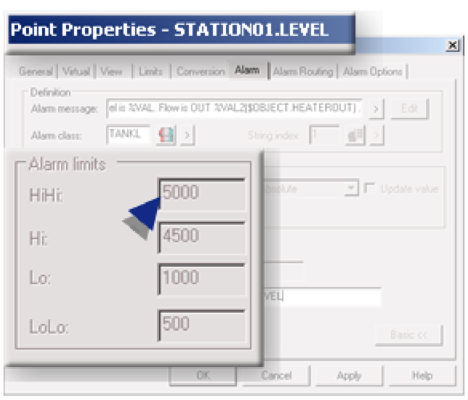
5000 is entered in the HIGH CRITICAL field, which is the prompt *(page 26)* for the `LHIGHCRITICAL` attribute.



1. Instantiation: Point properties Result

The value entered for the attribute in the Object dialog box displays in the Point Properties dialog box associated field.

Example



Expression with Two Operands

Expression versatility is expanded by using two operands instead of just one.

Operands can include:

- Two numbers.
- Two attributes.
- One attribute and one number.

Operators that are available to create the expression are:

Operator	Description
-	Minus
*	Times
/	Divided by
+	Plus
<	Less an
<>	Not equal to
=	Equal to
>	Greater than
none	No operand (Disables the second operand)

2.2.4. Data Item: Additional Expression Examples

- Example: Alarm Enable
- Example: `$DATA_ITEM_EXISTS`

Example: Alarm Enable

Note: When the enable state fields are included for a data item, an object designer can enable/disable the included alarm levels for each object.

The instantiated analog point for a data item named LEVELEXP requires:

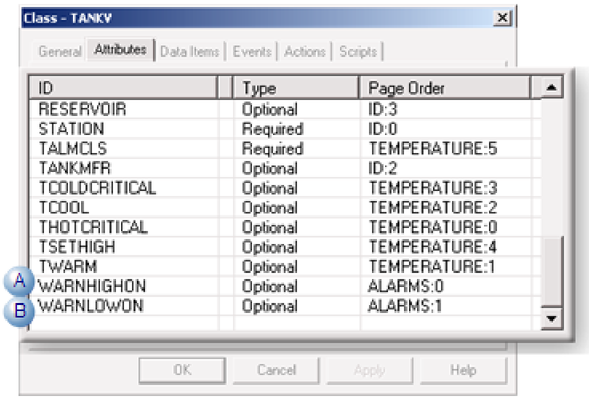
- Four alarm states for some objects.
- Alarm High and Alarm Low states only for other object.

A *(page 59)*	Class: Data item configuration.
B *(page 60)*	Object: Object defined.
C *(page 61)*	Instantiation: Point properties result.

1. Class: Data Item Configuration

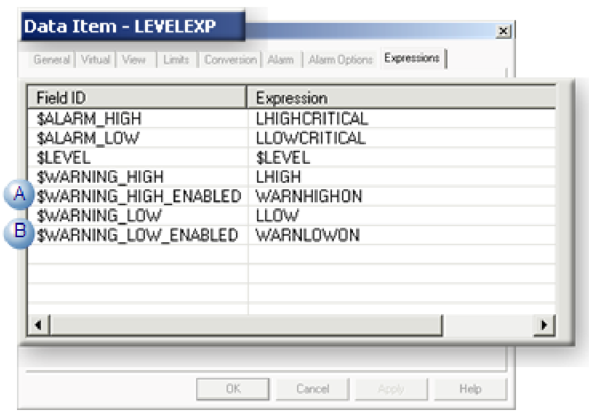
Two analog attributes are created that will display as fields in the Object dialog box.

The attributes are as follows.



	Field ID	Field Name for Attribute Dialog Box
A	$WARNING_HIGH_ENABLED	WARN HIGH ON
B	$WARNING_LOW_ENABLED	WARN LOW ON

Field ID's are assigned to the LEVELEXP data item.



	Field ID	Expression Example
A	$WARNING_HIGH_ENABLED	WARNHIGHON analog attribute.
B	$WARNING_LOW_ENABLED	WARNLOWON analog attribute.

1. Object: Object Defined

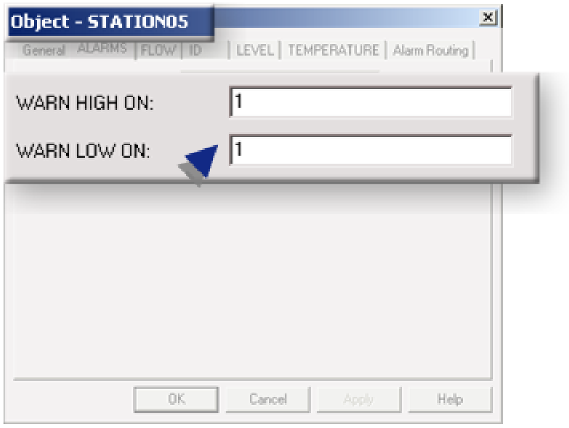
Two objects are defined in the TANKV class.

STATION05

One object, STATION05, requires all alarm states.

A value that is greater than 0 is required to enable the alarm states.

1 is entered for each

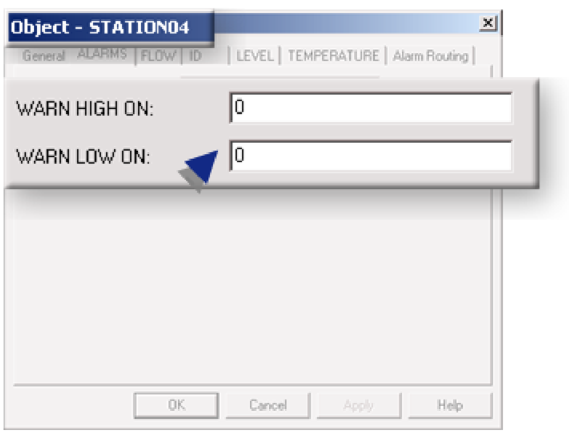


STATION04

One object, STATION04, requires only the Alarm High and Alarm Low alarm states.

The value 0 is required to disable the alarm states.

0 is entered for each

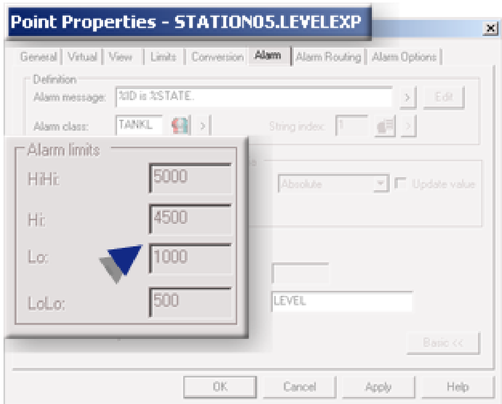


1. Instantiation: Point Properties Result

The alarm state values for the instantiated points reflect the entries in the Object dialog box.

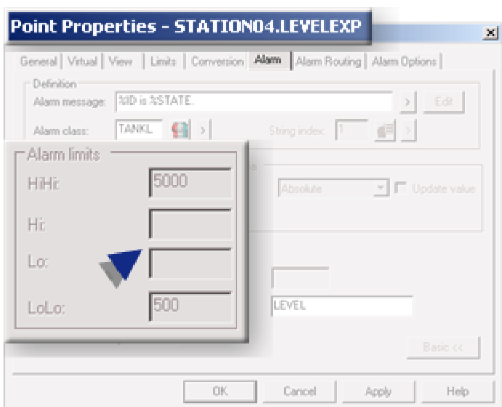
STATION05

STATION05.LEVELEXP has four enabled alarm states.



STATION04

STATION04.LEVELEXP has Alarm High and Alarm Low only enabled.



A data item can be included or excluded from being instantiated into a point based on the expression in an assigned `$DATA_ITEM_EXISTS` field ID.

Example: $DATA_ITEM_EXISTS

A class includes data items, that will be instantiated into points based on which control valve is selected for an object.

The data items require the following control valves.

Data Item	CVALVE01	CVALVE02	CVALVE03
TANK01	X		
TANK02		X	
TANK03			X

One data item will be instantiated into a point for each object; the data item that is instantiated depends on which valve is selected for that object.

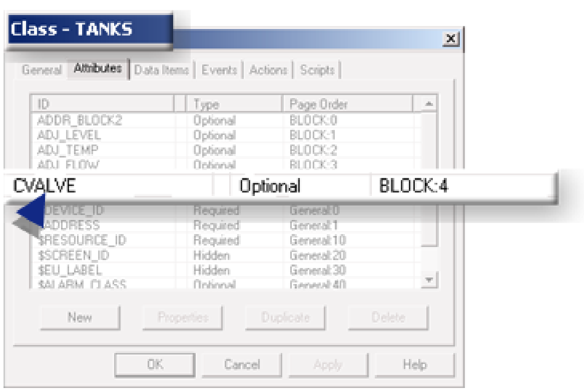
A (page 63)	Class: Data item configuration.
B (page 64)	Object: Object defined.
C (page 65)	Instantiation: Point properties result.

1. Class: Data Item Configuration

In the class configuration data items will be assigned to the same attribute.

Attribute Created

An analog attribute, `CVALVE`, is created that will be used to identify the Valve that should be selected for each data item.

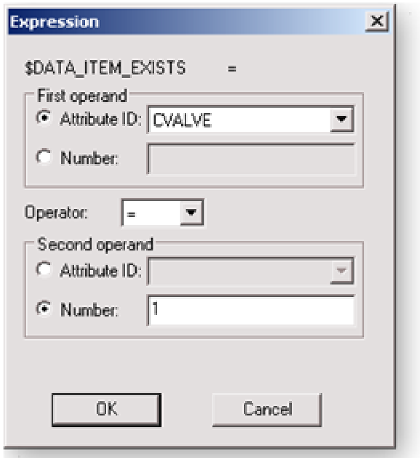


Note: The Prompt in the Attribute dialog box will be **CONTROL VALVE**.

Data Item Configuration

Three TANK data items are created.

The first data item, TANK01 is assigned `$DATA_ITEM_EXISTS`, with the following expression to associated Control Valve 1 with TANK01.



Field	Example Value Assigned	
First operand	Attribute ID	CVALVE
Operator		=
Second operand	Number	1

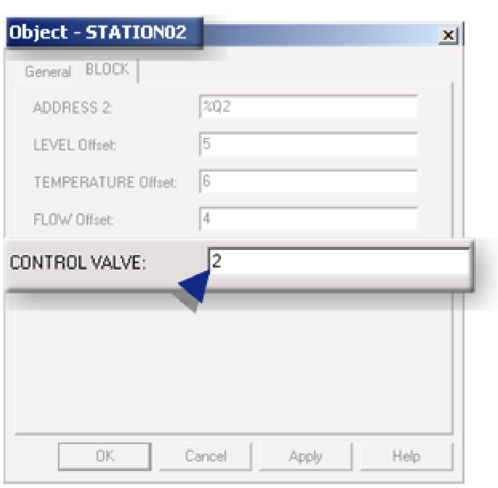
Two additional data items are assigned the field ID, `$DATA_ITEM_EXISTS`.

The three data items and expressions for `$DATA_ITEM_EXISTS` are as follows.

Data Item	$DATA_ITEM_EXISTS Expression
TANK01	CVALVE=1
TANK02	CVALVE=2
TANK03	CVALVE=3

1. Object: Object Defined

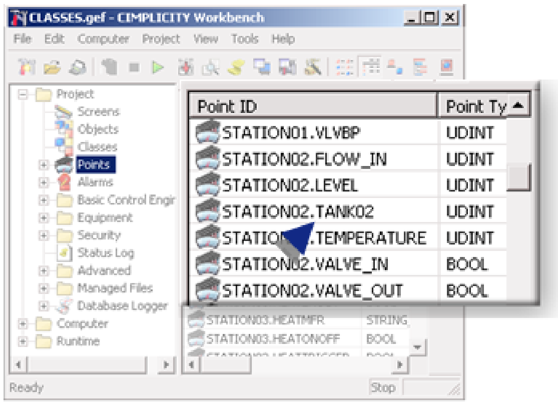
2 is entered in the **CONTROL VALVE** field for an object, STATION02.



1. Instantiation: TANK Points Existing

TANK02 is instantiated for the STATION01 object.

TANK01 and TANK03 do not exist for that object.



2.3. Address Adjustment Expressions

Address adjustment expressions provide a powerful tool that automates applying addresses to devices at the object level.

Address adjustment expressions are computed against the base `$ADDRESS` attribute value that is entered for an object.

- Overview: Address adjustment.
- Configuration: Numeric $ADDRESS_ADJ entry.
- Configuration: Attribute $ADDRESS_ADJ entry.
- Additional base addresses.
- Do not configure $ADDRESS_ADJ if a class data item is configured with a device address.

Overview: Address Adjustment

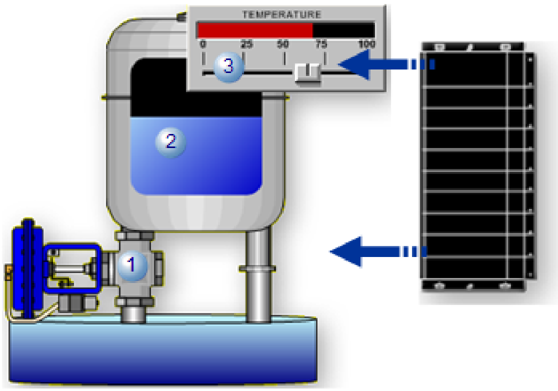
Address adjustment options enable you to do any of the following:

- Enter a numeric value that specifies the address adjustment at the class data item level.
- Create numeric attributes that will enable the address adjustment to be entered at the object level.
- Create a string attribute that provides a field in the Object dialog box to enter an additional base address.

Example

A section of a water heating system that is connected to a PLC requires different address assignments. These assignments can be made using address adjustment.

Three assignments demonstrate the flexibility that address adjustment options provide.



1	Input valve		Numeric $ADDRESS_ADJ entry	BLOCK 1
2	Level		Attribute $ADDRESS_ADJ Entry	BLOCK 1
3	Temperature		Additional Base Address	BLOCK 2

Note: An address can be entered for an individual data item in its Data Item dialog box. That address is assigned to its instantiated point. This assignment overrides `$ADDRESS` or `$ADDRESS_ADJ` assignments.

Configuration: Numeric $ADDRESS_ADJ Entry

A simple numeric value at the class level can be used for address adjustment.

Conditions that can use a simple entry include the following.

- Class objects will require one base address only.
- The base address will be different for different objects.
- The offset from the base address for an instantiated point can be the same for all objects.

Note: When `$ADDRESS_ADJ` is selected an Octal checkbox appears in the Expression dialog box.

Check Octal to define the adjustment as an Octal number.

A (page 67)	Class: $ADDRESS_ADJ configuration.
B (page 67)	Object: Address defined.

C (page 68)	Instantiation: Point properties address result.

1. Class: $ADDRESS_ADJ Configuration

1. Select the Expressions tab in a device Data Item dialog box,

   Example

   A data item, VALVE_IN, will include a basic address adjustment from the base address.

2. Click New.

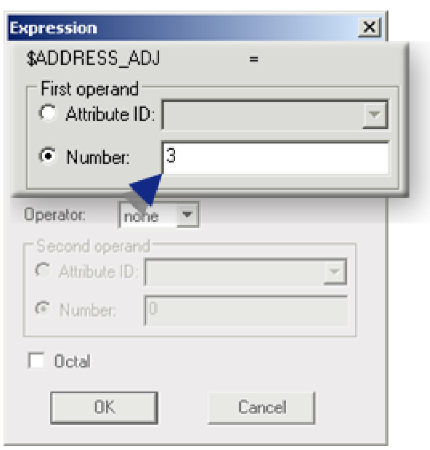
3. Select $ADDRESS_ADJ in the dropdown **Field ID** list.

   An Expression dialog box opens.

4. Enter a number in the **Number** field.

   Example

   3 is entered in the **Number** field.



   a. Object: Address Defined

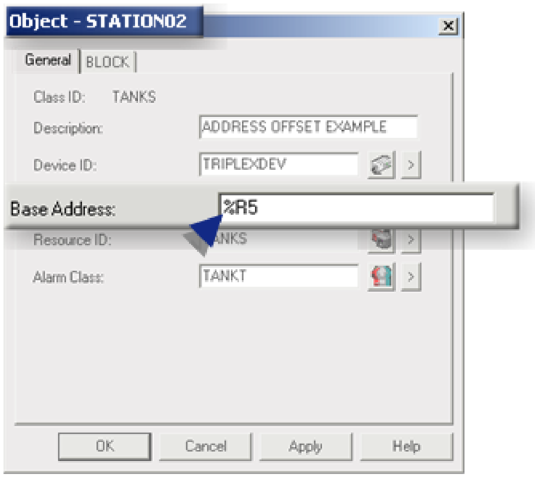
An address is entered in the **Address** field in the Object dialog box.

Note: The Address field is created from the CIMPLICITY class attribute, `$ADDRESS`.

Example

An object named STATION02 is created.

%R5 is entered in the STATION02 Object dialog box **Address** field.

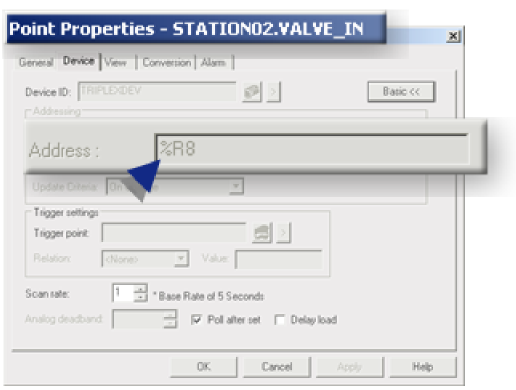


a. Instantiation: Point Properties Address Result

When the point is instantiated, the device address is the `Base address + $ADDRESS_ADJ` value.

Example

The read-only device address for the instantiated STATION02.VALVE_IN point is %R8.



Configuration: Attribute $ADDRESS_ADJ Entry

A custom attribute can be used for address adjustment.

Conditions that can use a custom attribute include the following.

- Class objects will require one base address only.
- The base address will be different for different objects.

- The offset from the base address for an instantiated point may also be different for different objects.

A (page 69)	Class: $ADDRESS_ADJ configuration.
B (page 70)	Object: Address defined.
C (page 71)	Instantiation: Point properties address result.

a. Class: $ADDRESS_ADJ Configuration

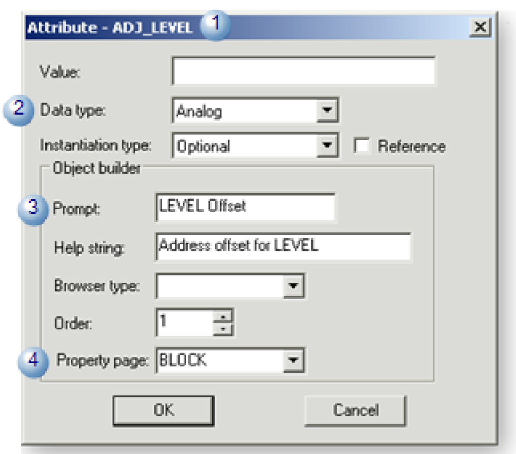
Attribute Configuration

An analog attribute can be created to display as a numeric field in the Object dialog box.

Example

The address adjustment will be required for instantiated points that are associated with the LEVEL value; the adjustment will be different for different objects.

A class attribute is created as follows.



	Field	Example Entry
1	Attribute Name	ADJ_LEVEL
2	Data type	Analog
3	Prompt	LEVEL Offset
4	Property Page	BLOCK

$ADDRESS_ADJ Configuration.

5. Select the Expressions tab in a device Data Item dialog box,

   Example

   A data item, LEVEL, will include an attribute address adjustment against the base address.

6. Click New.

7. Select $ADDRESS_ADJ in the dropdown **Field ID** list.

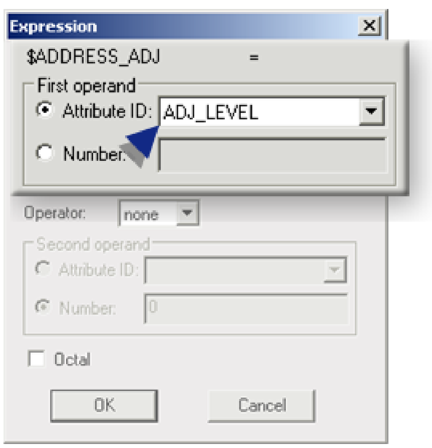
   

   An Expression dialog box opens.

8. Select the attribute that should be applied to the **$ADDRESS_ADJ** field.

   Example

   The attribute, `ADJ_LEVEL`, is selected.

   a. Object: Address Defined

   Both of the following are entered in the Object dialog box.

   Base Address

   An address is entered in the **Address** field.

   Note: The Address field is created from the CIMPLICITY class attribute, `$ADDRESS`.

   Example

An object named STATION02 is created.

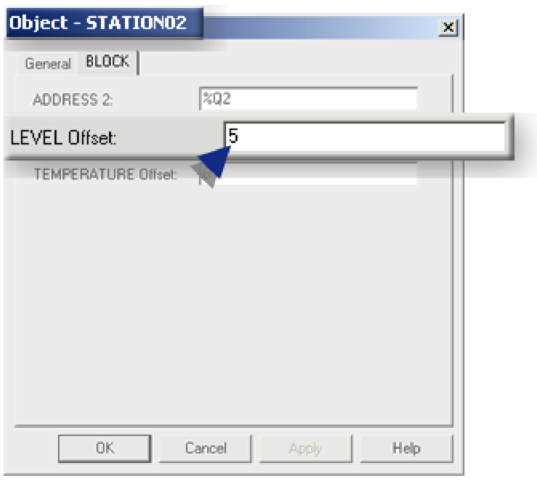
%R5 is entered in the STATION02 Object dialog box **Address** field.

Adjustment

An offset value is entered in the field created by the custom analog class attribute.

Example

5 is entered in the **LEVEL Offset** field that was created by the `ADJ_LEVEL` attribute.

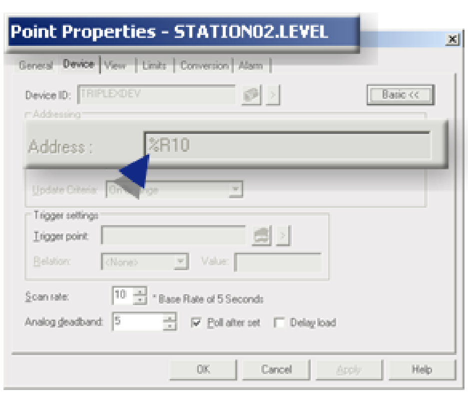


a. Instantiation: Point Properties Address Result

When the point is instantiated, the device address is the `Base address + $ADDRESS_ADJ` value entered in the Object dialog box.

Example

The read-only device address for the instantiated STATION02.LEVEL point is %R10.



Additional Base Addresses

Address adjustment can enable applying more than one base address to an object.

Conditions that can use more than one base address include the following.

- Objects will be connected to two PLC blocks, each of which requires a unique base address.
- The base addresses will be different for different objects.
- The offset from the base address for an instantiated point may always be the same or may be different for different objects.

A (page 72)	Class: $ADDRESS_ADJ configuration.
B (page 73)	Object: Address defined.
C (page 74)	Instantiation: Point properties address result.

a. Class: $ADDRESS_ADJ Configuration

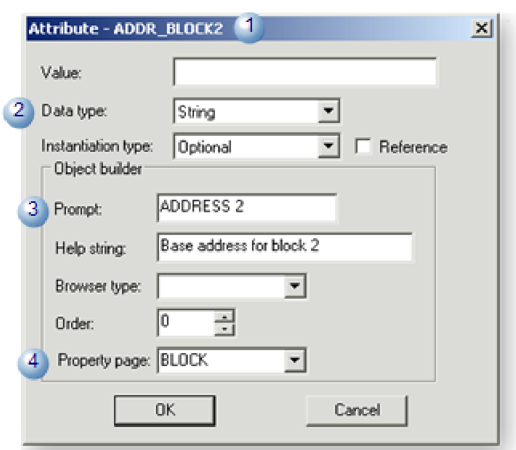
Attribute Configuration

A string attribute can be created to display as a string field in the Object dialog box.

Example

The second address will be required that will be used as the base address for some device instantiated points.

A class attribute is created as follows.



	Field	Example Entry
1	Attribute Name	ADDR_BLOCK2

2	Data type	String
3	Prompt	ADDRESS 2
4	Property Page	BLOCK

Data Item Device Address Configuration

Both of the following are entered in the Data Item dialog box.

9. An `$ADDRESS_ADJ` entry is entered in the Expression dialog box.

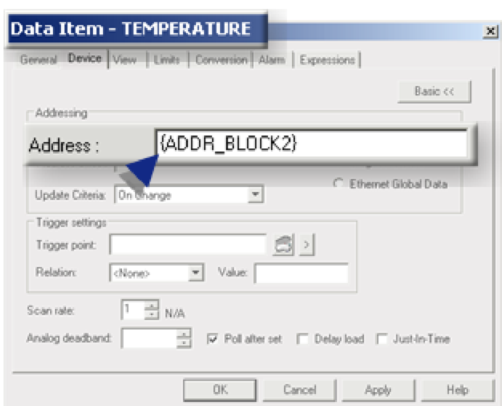
   Example

   A data item TEMPERATURE is assigned a custom `ADJ_TEMP` attribute for the **$ADDRESS_ADJ** field.

10. The string attribute created for the additional base address is entered as a string substitution in the Device tab>**Address** field.

   Example

   The string substitution `{ADDR_BLOCK2}` is entered in the Data Item dialog box for the FLOW_IN data item.



   a. Object: Address Defined

Both of the following are entered in the Object dialog box.

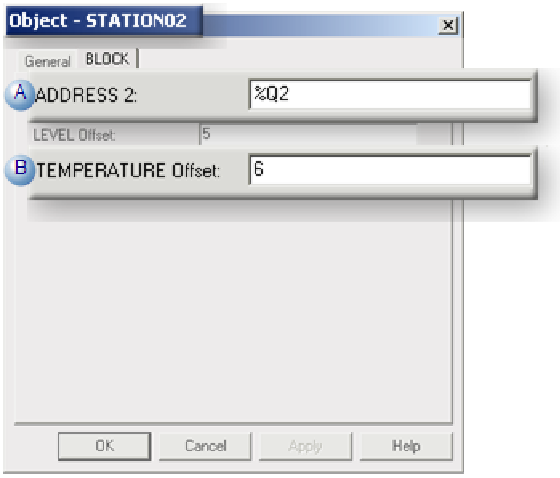
11. An address is entered in the field that was created for the second base address by the custom string attribute.

12. (Optional) An offset value is entered if fields are available.

   Example

   An object named STATION02 is created.

The following entries affect the instantiated FLOW_IN point.



Field	Example Entry
ADDRESS 2	%Q2
TEMPERATURE Offset	6

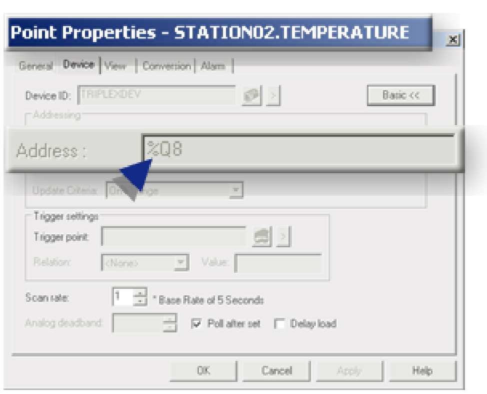
a. Instantiation: Point Properties Address Result

When the point is instantiated, the device address is the `Custom Base address + $ADDRESS_ADJ` value.

Note: The `$ADDRESS_ADJ` value may be based on a number entered in the class Expression dialog box or the Object dialog box.

Example

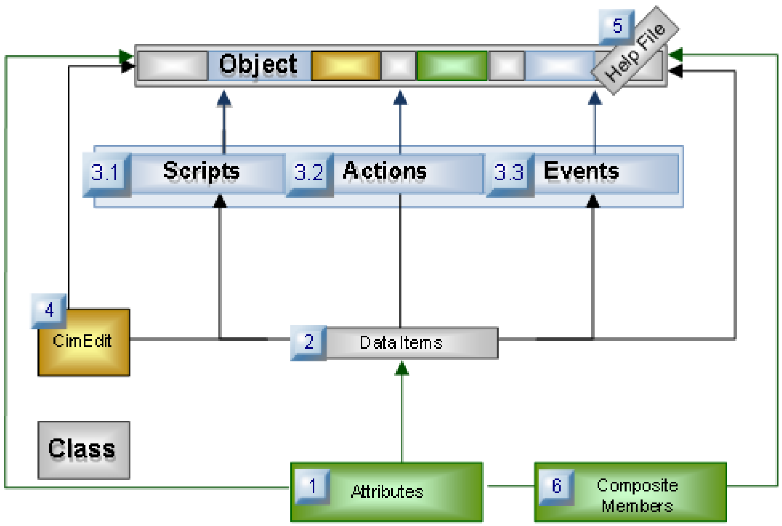
The read-only device address for the instantiated STATION02.TEMPERATURE point is %Q8.



3. Scripts, Actions, Events

3. Scripts, Actions, Events

CIMPLICITY classes can include scripts, actions and events that can interact with each other the same way they do in any CIMPLICITY project.



	Item	Applied to:
3.1 (page 75)	Script	Class as soon as they are created
3.2 (page 85)	Action	Each instantiated object where it was applied in the class template.
3.3 (page 92)	Event	Each instantiated object where it was applied in the class template.

3.1. Class Scripts

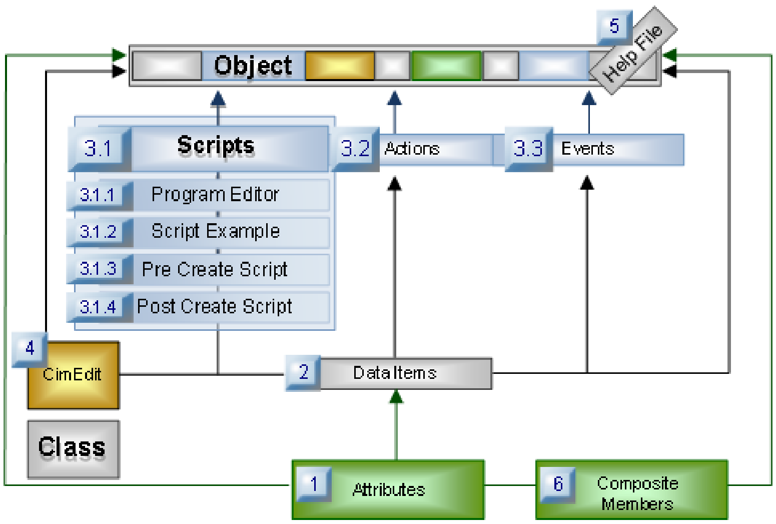
3.1. Class Scripts

- Overview: Scripts in class configuration.
- Overview: Script included in a class.
- Configuration: Class scripts.

Overview: Scripts in Class Configuration

Class scripts:

- Can be created and modified in the CIMPLICITY Program Editor by opening it through the Class dialog box.
- Can be available anywhere that standard CIMPLICITY project scripts are.
- Are stored in the Workbench Scripts folder, with a **class$** prefix, as soon as they are created.\



3.1.1 (page 78)	Open the CIMPLICITY Program Editor.
3.1.2 (page 81)	Class scripts definition.
3.1.3 (page 81)	Pre create script.
3.1.4 (page 83)	Post create script.

The **class$** script can be modified in the CIMPLICITY Program Editor by opening it through the Workbench Scripts folder.

Overview: Script Included in a Class

A class retains one copy only of each script.

A (page 77)	Class: Script configuration.
B (page 77)	Instantiation: Class script availability.

1. Class: Script Configuration

Class scripts are created and edited in the Program Editor. CIMPLICITY inserts `<Class Name>$` in front of the script name as soon as it is created.

The syntax for each class script name is:

`Class$Scriptname.bcl`

Where

`Class$` is the name of the class to which the script belongs.

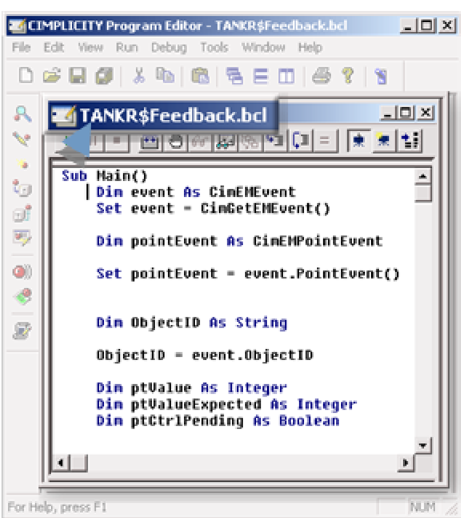
Scriptname.bcl is the name entered in the New Script Name dialog box.

You can modify the script either through the Scripts folder or through the Class dialog box.

Example

A script, feedback.bcl, has been created for a TANKR class.

When feedback.bcl is opened in the Program Editor, the name on the title bar is `TANKR $feedback.bcl`.

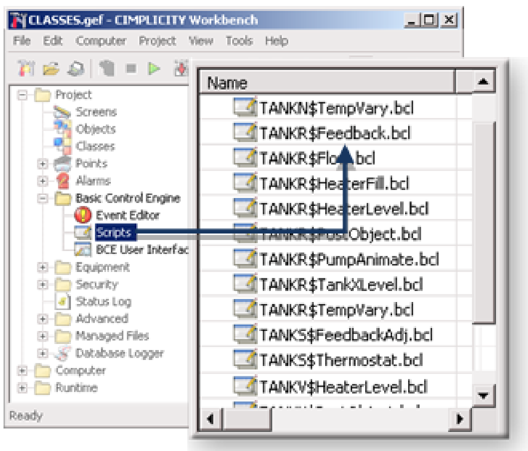


1. Instantiation: Class Script Availability

As soon as a class script is created *(page 79)* or duplicated *(page 80)* it is added to the Basic Control Engine>Scripts list and is available for objects.

Example

The TANKR$feedback.bcl is listed in the Workbench right pane. As soon as the Class is applied (clicking Apply or OK in the Class dialog box) the script will be applied to every object associated with TANKR.

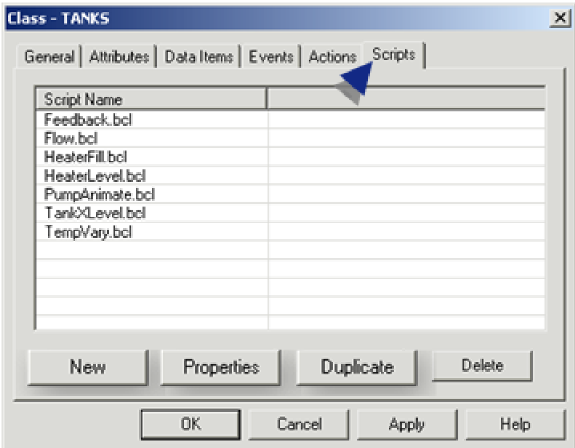


Configuration: Class Scripts

3.1.1 *(page 78)*	Open the CIMPLICITY Program Editor.
3.1.2 *(page 81)*	Class scripts definition.
3.1.3	Post create script.

3.1.1. Open a Script Editor

The buttons on the **Scripts** tab enable you to open a script editor to create a new script or modify an existing script.



rect 15, 215, 90, 238 *(page 79)*
rect 100, 216, 175, 239 *(page 80)*
rect 184, 216, 259, 239 *(page 80)*

- New script.
- Existing script.
- Duplicate script.

Note: All scripts that are listed for a class on the Scripts tab and in the Workbench **Basic Control Engine>Scripts** right pane will be included in the class if the class is exported/imported *(page 117)* to another project. They are listed even if you cancelled creating them and/or did not compile them. You can delete any action by selecting an ID and clicking Delete.

New Script

1. Click **New** on the Scripts tab in the Class dialog box.

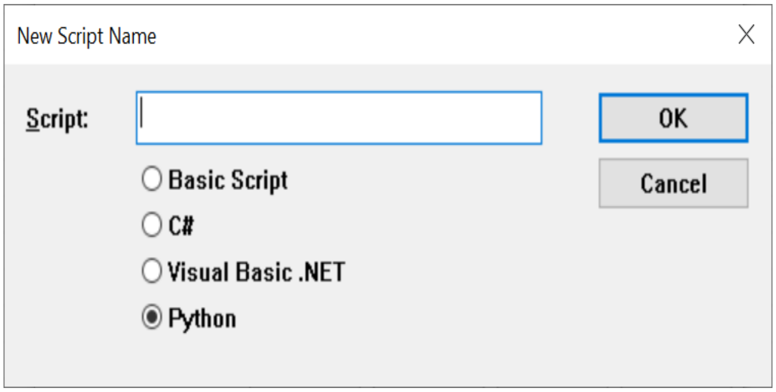
   The New Script Name dialog box displays.

2. Enter a unique name in the **Script** field.

   A Script name is

   - Is up to 15 characters.
   - Can have alphanumeric characters.
   - Can have underscores.
   - Cannot have spaces.

3. Select the type of script you want to create.



4. Click **OK**.

Result: A blank script editor opens.

Existing Script Properties

5. Select an existing script.
6. Click **Properties**.

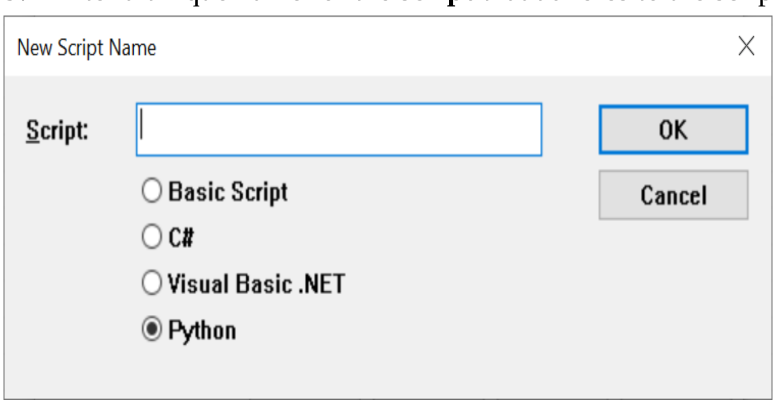
Result: A script editor with the existing script opens.

Duplicate Script

7. Select an existing script.
8. Click **Duplicate**.

A New Script Name dialog box opens.

9. Enter a unique name for the **script** that adheres to the script ID requirements *(page 79)*.



10. Click **OK**.

The duplicate script is added to the list on the **Scripts** tab.

3.1.2. Class Scripts Definition

Because class scripts will be used for more than one object, the script will require the flexibility to call fully instantiated points that have different names.

An extension in the CIMPLICITY Basic script, `CimEMEvent (object)` , enables the script to access the object that represents the selected event.

One of the attributes is the object that triggered this event.

That way you can make the fully qualified point ID's with the object ID as the prefix to the point ID.

Sample Script

The script below demonstrates how to identify the name of the object that is being used for an event.

As a result, that object ID can be used to construct the object's point ID's for which script is running.

```
Sub Main()
Dim event As CimEMEvent
Set event = CimGetEMEvent()
Dim pointEvent As CimEMPointEvent
Set pointEvent = event.PointEvent()
Dim ObjectID As String
ObjectID = event.ObjectID
Dim ptValue As Integer
Dim ptValueExpected As Integer
Dim ptCtrlPending As Boolean
ptValue = pointEvent.Value

ptCtrlPending = PointGet(ObjectID & ".AT")
If ptCtrlPending Then
ptValueExpected = PointGet(ObjectID & ".DO.$RAW_VALUE")
If ptValue = ptValueExpected Then
'Indicate that a control is complete
PointSet ObjectID & ".AT", 0
End If
Else
'Enable the alarm on the digital indicator
PointSet ObjectID & ".DI.QUALITY.ALARMS_ENABLED", 1
End If
End Sub
```

3.1.3. Pre Create Script

- Overview: Pre create script.
- Configuration: Pre create script.

Overview: pre Create Script

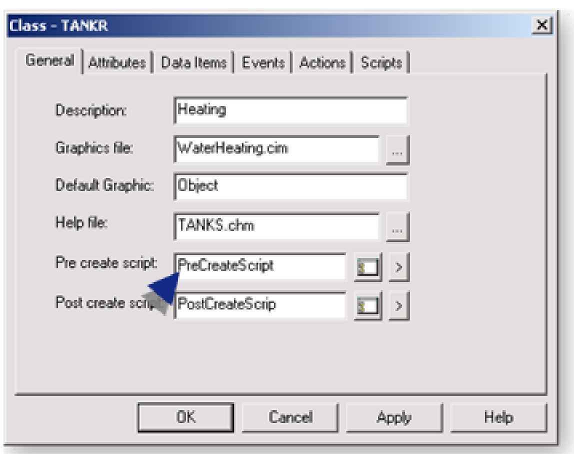
If class objects require additional functionality that is not configured in the class, the additional functionality can be included in a script. that will call the PreObjectCreateentry point before the object is created.

Tip: The CIMPLICITY Object Model includes many scripting objects that are developed specifically for classes, including the following.

	CimClassActionList (object)
	CimClassAttribute (object)
	CimClassAttributeList (object)
	CimClassDataItemExpression (object)
	CimClassDataItemExpressionList (object)
	CimClassDataItemList (object)
	CimClassEventActionList (object)
	CimClassEventList (object)
	CimClassInstance (object)
	CimClassList (object)
	CimClassScript (object)
	CimClassScriptList (object)

Configuration: pre Create Script

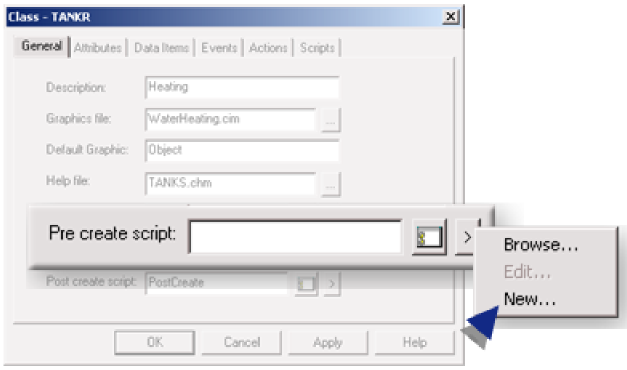
The content of the pre create script depends entirely on your class/object requirements. The one unique feature when configuring the class is that you enter its ID in the **pre create script** field on the Genera tab in the Class dialog box.



1. Class: Pre Create Script Configuration

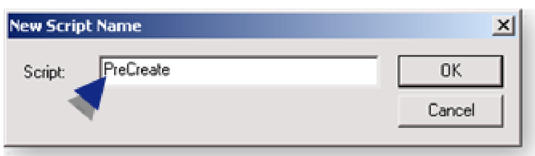
New Pre Create Script

1. Click the Popup menu button to the right os the Pre create script field; select New on the Popup menu.



   A New Script Name dialog box opens.

2. Enter a name in the **Script** field.



3. Click OK.

4. Click the Browse button to the right of the Pre create script field.
   A Select a Script browse window opens.

5. Select a script.

6. Click the Popup menu button to the right of the Pre create script field; select Browse.
   A Select a Script browse window opens.

7. Select a script.

3.1.4. Post Create Script

- Overview: Post create script.
- Configuration: Post create script.

Overview: Post Create Script

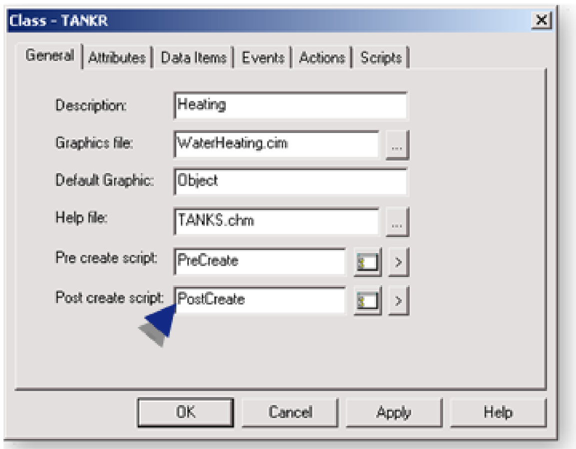
If class objects require additional functionality that is not configured in the class, the additional functionality can be included in a script that will call the PostObjectCreateentry point after the object is created.

Tip: The CIMPLICITY Object Model includes many scripting objects that are developed specifically for classes, including the following.

	CimClassActionList (object)
	CimClassAttribute (object)
	CimClassAttributeList (object)
	CimClassDataItemExpression (object)
	CimClassDataItemExpressionList (object)
	CimClassDataItemList (object)
	CimClassEventActionList (object)
	CimClassEventList (object)
	CimClassInstance (object)
	CimClassList (object)
	CimClassScript (object)
	CimClassScriptList (object)

Configuration: Post Create Script

The content of the post create script depends entirely on your class/object requirements. The one unique feature when configuring the class is that you enter its ID in the **Post create script** field on the Genera tab in the Class dialog box.

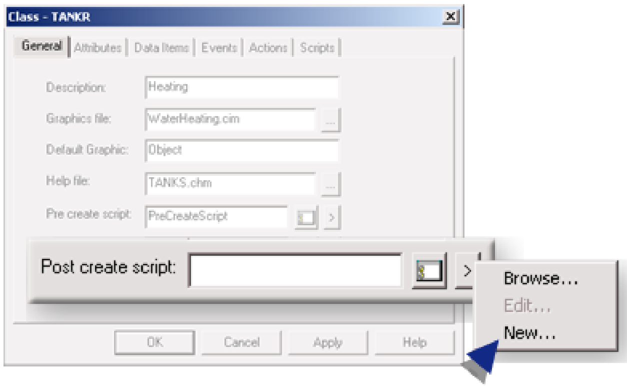


Class: Post create script configuration.	
Instantiation: Post create script.	

1. Class: Post Create Script Configuration

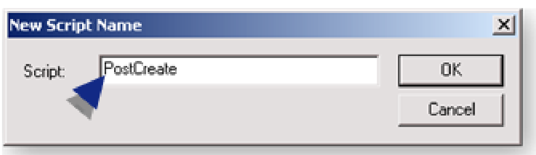
New Post Create Script

1. Click the Popup menu button to the right os the Post create script field; select New on the Popup menu.



A New Script Name dialog box opens.

2. Enter a name in the **Script** field.



3. Click OK.

4. Click the Browse button to the right of the Post create script field.
   A Select a Script browse window opens.

5. Select a script.

6. Click the Popup menu button to the right of the Post create script field; select Browse.
   A Select a Script browse window opens.

7. Select a script.

3.2. Class Actions

3.2. Class Actions

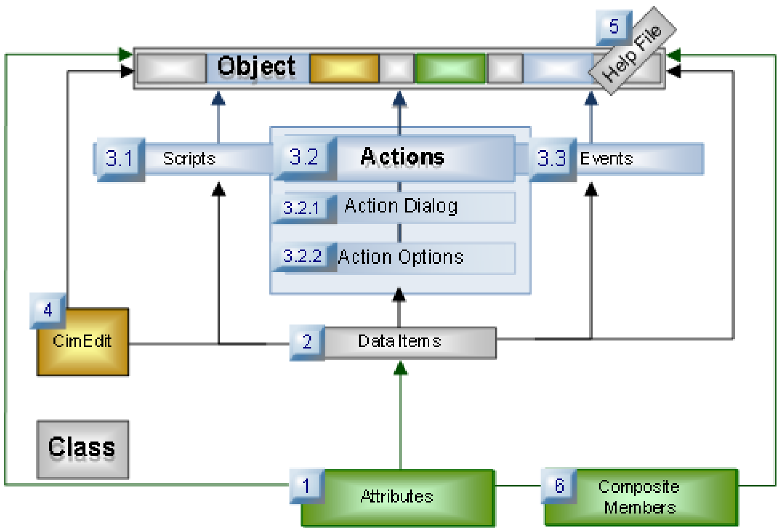
- Overview: Actions in class configuration.
- Overview: Class actions instantiated into an object action.

• Configuration: Class action.

Overview: Actions in Class Configuration

Class actions

• Are created through the Class dialog box.
• Appear in the Event Editor window when an object is created.
• In the Event Editor, actions:
• Are associated with an object. An object designer cannot change the association.
• Cannot be modified. All fields are read-only.
• Cannot be deleted.
• Can be associated with a non-object event. This enables an object to instantiate actions that can be called from other events.



3.2.1 (page 88)	Open an Action dialog box.
3.2.2 (page 90)	Class action definition.

Overview: Class Actions Instantiated into an Object Action:

Class actions are instantiated into object actions, as follows.

A (page 87)	Class: Action configuration.
B (page 87)	Object: Object created.

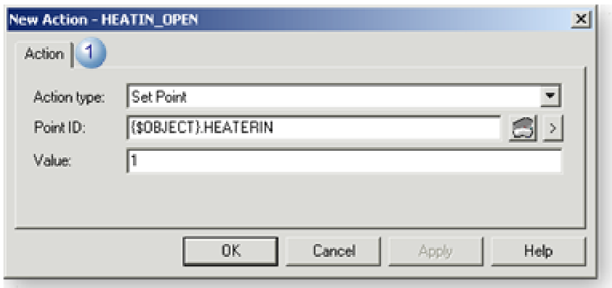
C (page 87)	Instantiation: Action result.

1. Class: Action Configuration

Actions are created in Action dialog boxes that are opened through the Class dialog box.

Example

A class action `HEATIN_OPEN` is created as a `Set Point` action.



Note: Class action ID's *(page 88)* are listed on the Actions tab in the Class dialog box.

1. Object: Object Created

Class objects are created from the class.

Example

A class object created from the class TANKS is named: STATION01.

1. Instantiation: Action Result

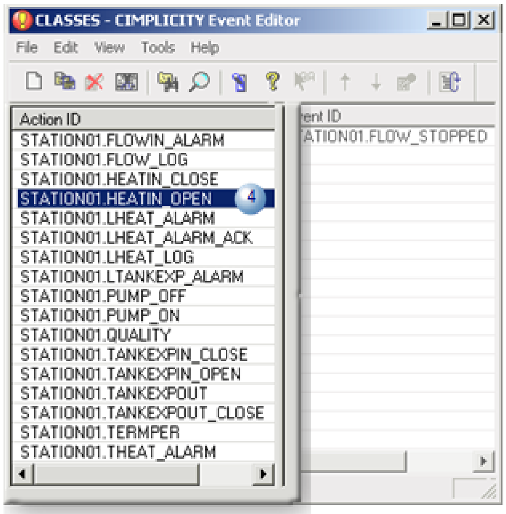
Class actions are instantiated into object actions.

Event Editor

The instantiated actions are listed in the Event Editor.

Example

The class action instantiated into a class object is named: STATION01.HEATIN_OPEN.

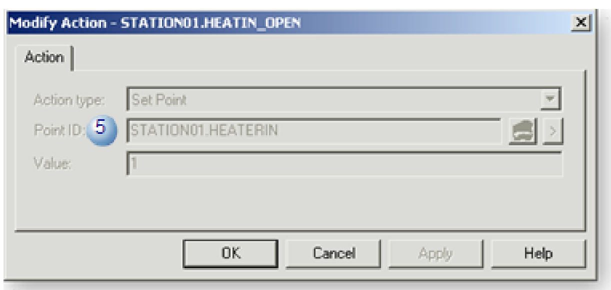


Modify Action Dialog Box

Instantiated object actions cannot be modified.

Example

The Modify Action dialog box for `STATION01.HEATIN_OPEN` is read-only.

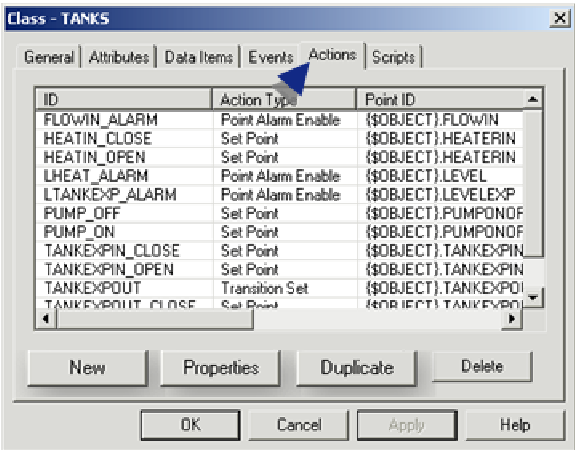


Configuration: Class Actions

3.2.1 (page 88)	Open an Action dialog box.
3.2.2 (page 90)	Class action definition.

3.2.1. Open an Action Dialog Box

Buttons on the Actions tab enable you to open an Action dialog box to create a new or modify an existing action.



rect 15, 216, 93, 238 (page 89)
rect 99, 216, 177, 238 (page 90)
rect 183, 216, 261, 238 (page 90)

- New action.
- Existing action.
- Duplicate action.

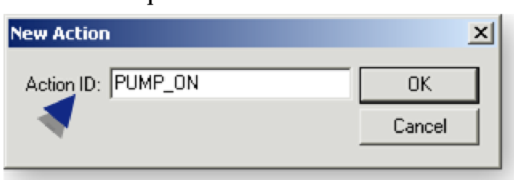
Note: You can delete any action by selecting an ID and clicking Delete.

New Action

1. Click **New** on the Actions tab in the Class dialog box.

   The New Action dialog box displays.

2. Enter a unique name in the **Action ID** field.

   The Action ID:

   - Is limited to 16 characters.
   - Can be composed of
   - Uppercase alphabetic characters,
   - Numeric characters and
   - Underscores.
   - Must begin with an alphabetic character.

A blank New Action dialog box opens.

3. Click **OK**.

   Result: A New Action dialog box for the new action opens.

   Existing Action Properties

4. Select an existing action.
5. Click Properties.

   Result: The Action dialog box with specifications for the existing action opens.

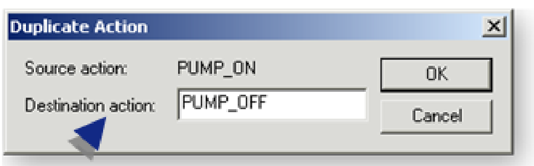
   Duplicate Action

6. Select an existing action.
7. Click Duplicate.

   A Duplicate Action dialog box opens.

   The selected source action displays.

8. Enter a unique name for the **Destination action** that adheres to the action ID requirements *(page 89)*.



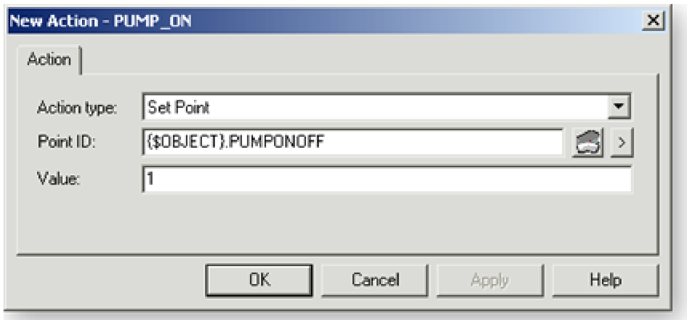
9. Click OK.

A New Action dialog box with specifications for the source action opens.

3.2.2. Class Action Definition

Note: The user can associate a non-object event with an object action. This allows an object to instantiate actions that can be called from other events.

The fields in the Action dialog box depend on the selected action type.



- Action type.
- Action fields.
- Supported string substitution for action fields.

Action Type

All CIMPLICITY action types (found in the Event Editor) are available except for:

- **Alarm Lookup** and
- **Recipe Upload/Download**.

The difference between configuring a class action and a standard action is that you:

- Select a data item instead of a point ID when you configure an action that requires a point ID.
- The CIMPLICITY **$OBJECT** variable is automatically entered when you select the data item.

Actions available in the drop down list for classes include:

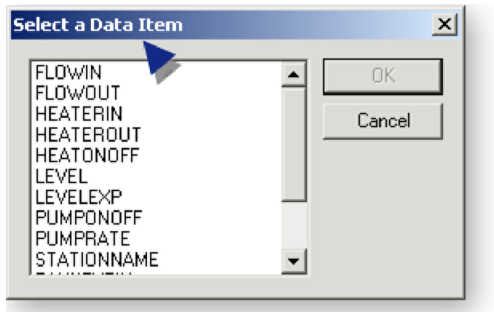
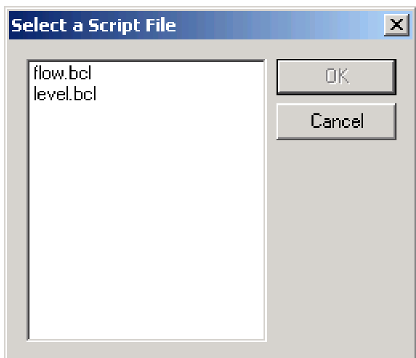
	Log Only
	Point Alarm Acknowledge
	Point Alarm Disable
	Point Alarm Enable
	Run Script
	Set Point
	Source Transition Set
	Transition Set

Action Fields

Fields vary based on your selected action. One or more field may display.

Note: The field's value can be selected from an associated browser.

Possible fields and buttons to open associated browsers are as follows.

Field	Button	Opens
Point ID		 **Note:** The CIMPLICITY `$OBJECT` variable appears in the **Point ID** field before the data item as follows: `{$OBJECT).DataItem`
Source		
Alarm ID		
Script		
Resource		CIMPLICITY Select a Resource browser

Result: When the Action dialog box is closed, the action, which is modified or created, displays on the Actions tab in the Class dialog box.

Supported String Substitution for Action Fields

String substitution is supported for the following action fields, which you enter in the associated Action dialog box field.

- Point ID.
- Resource ID.
- Point value.

3.3. Class Events

3.3. Class Events

- Class events configuration overview.
- Class events configuration steps.

- Configuration: Class events.

Overview: Events in Class Configuration

Class events provide a wide degree of flexibility for changing the business rules of a system.

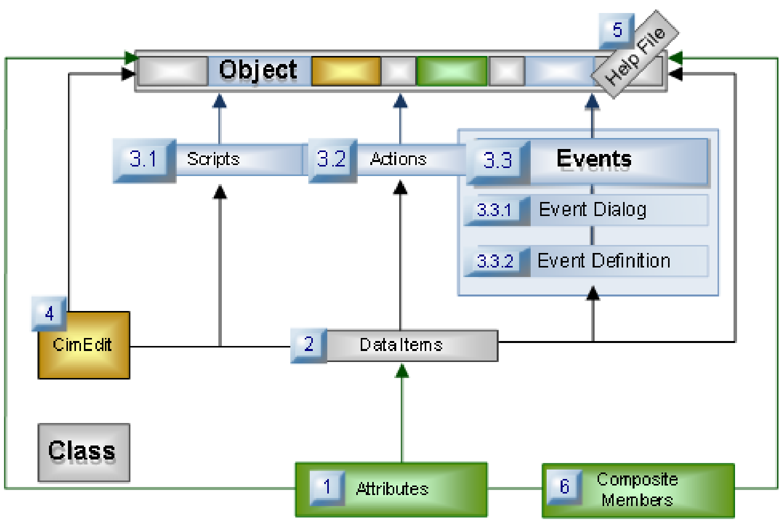
Class events:

- Are created through the Class dialog box.
- Appear in the Event Editor window when an object is created. In the Event Editor, events:
- Are associated with an object.

An object designer cannot change the association.

- Cannot be modified. All fields are read-only.
- Cannot be deleted.

Creating class events is similar to creating events in the Event Editor. All existing event types that are supported in the event editor are available for class event configuration.



3.3.1 *(page 96)*	Open an Event dialog box.
3.3.2 *(page 97)*	Class event definition.

Overview: Class Events Instantiated into an Object Event

Class events are instantiated into object events, as follows.

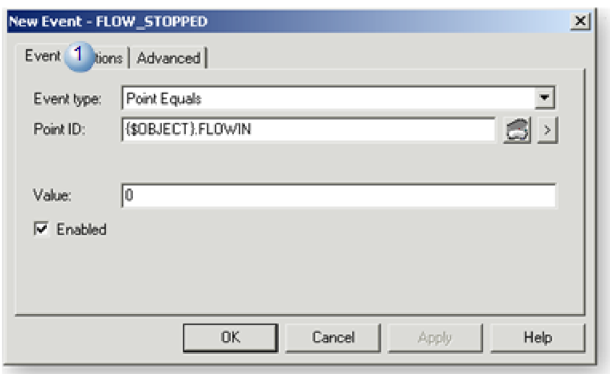
A (page 94)	Class: Event configuration.
B (page 94)	Object: Object created.
C (page 94)	Instantiation: Event result.

1. Class: Event Configuration

Actions are created in Event dialog boxes that are opened through the Class dialog box.

Example

A class event `FLOW_STOPPED` is created as a `Point Equals` event.



Note: Class event ID's *(page 96)* are listed on the Event tab in the Class dialog box.

Example

The event named `FLOW_STOPPED` is listed with other events on the Events tab in a class named TANKS.

1. Object: Object Created

Class objects are created from the class.

Example

A class object created from the class TANKS is named: STATION01.

1. Instantiation: Class Event Result

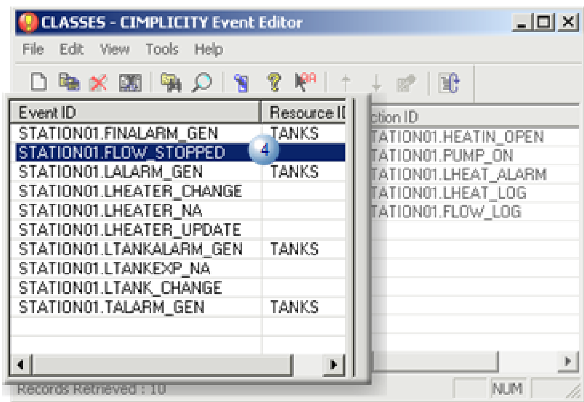
Class events are instantiated into object events.

Event Editor

The instantiated events are listed in the Event Editor.

Example

The class event instantiated into a class object is named: STATION01.FLOW_STOPPED.

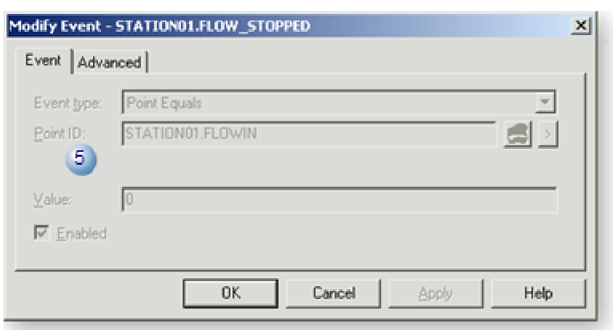


Modify Event Dialog Box

instantiated object events cannot be modified.

Example

The Modify Event dialog box for `STATION01.FLOW_STOPPED` is read-only.



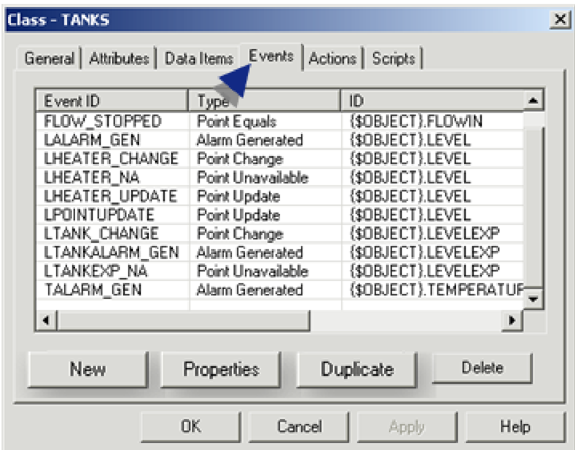
Configuration: Class Events

3.3.1 (page 96)	Open an Event dialog box.

3.3.2 (page 97)	Class event definition.

3.3.1. Open an Event Dialog Box

Buttons on the events tab enable you to open an Event dialog box for a new or existing event.



rect 13, 217, 91, 240 (page 96)
rect 97, 216, 175, 239 (page 97)
rect 182, 215, 260, 238 (page 97)

- New event.
- Existing event.
- Duplicate event.

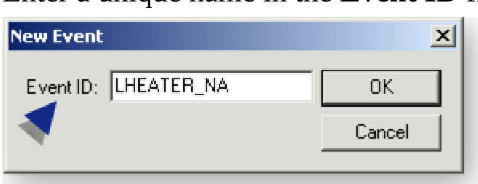
Note: You can delete any event by selecting an ID and clicking Delete.

New Event

1. Click **New** on the events tab in the Class dialog box.

   The New Event dialog box displays.

2. Enter a unique name in the **Event ID** field.



   The Event ID:

   - Is limited to 16 characters.

- Can be composed of
- Uppercase alphabetic characters,
- Numeric characters and
- Underscores.
- Must begin with an alphabetic character.

A blank New Event dialog box opens.

3. Click **OK**.

   Result: A New Event dialog box for the new event opens.

   Existing Event Properties

4. Select an existing event.
5. Click Properties.

   Result: The Event dialog box with specifications for the existing event opens.

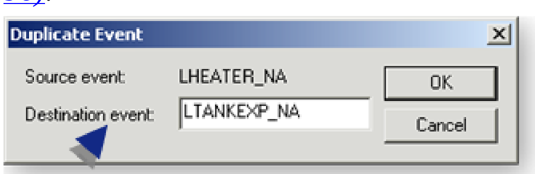
   Duplicate Event

6. Select an existing event.
7. Click Duplicate.

   A Duplicate Event dialog box opens.

   The selected source event displays.

8. Enter a unique name for the **Destination event** that adheres to the event ID requirements *(page 96)*.



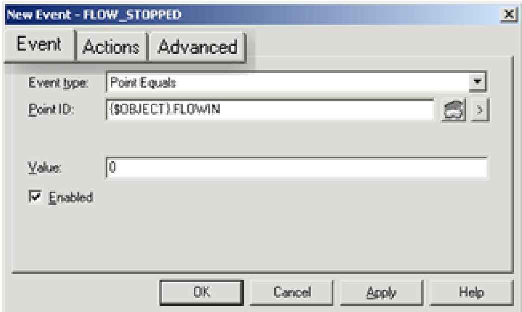
9. Click OK.

A New Event dialog box with specifications for the source event opens.

3.3.2. Class Event Definition

Class event configuration
Supported string substitution for event fields

Class event configuration

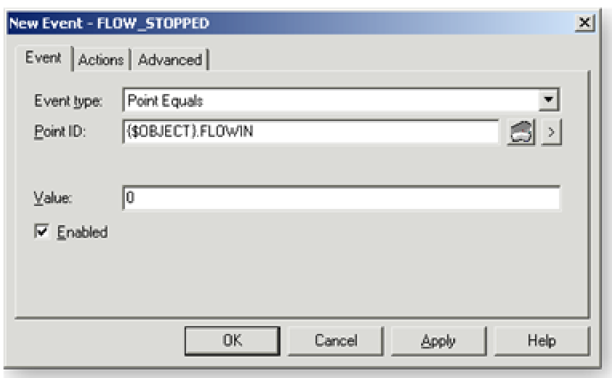


rect 2, 18, 47, 39 (page 98)
rect 92, 18, 151, 39 (page 103)
rect 45, 17, 90, 38 (page 100)

- Event tab
- Actions tab
- Advanced tab

Event Tab

The fields on the Event tab in the New Event dialog box depend on the selected event type.



- Event type
- Event fields

Event Type

All of the event types available in the Events Editor are available for a class event.

The difference between configuring a class event and a standard event is that you:

- Select a data item instead of a point ID when you configure an event that requires a point ID.

- The CIMPLICITY **$OBJECT** variable is automatically entered when you select the data item.

Events available in the drop down list for classes include:

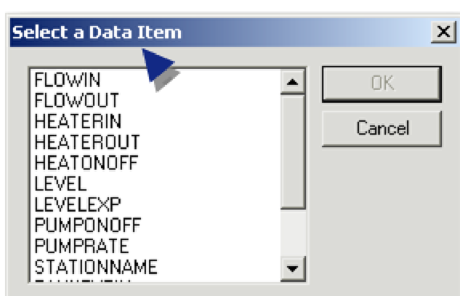
Alarm Acknowledged
Alarm Deleted
Alarm Generated
Alarm Reset
Point Change
Point Equals
Point Transition High
Point Transition Low
Point Unavailable
Point Update
Run Once
Timed

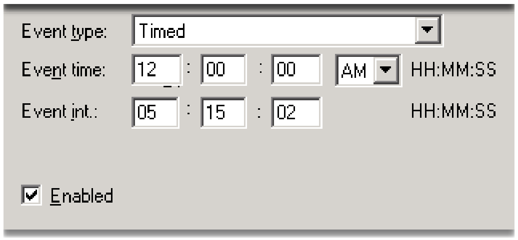
Event Fields

Fields vary based on your selected event. One or more field may display.

Note: The field's value can be selected from an associated browser.

Possible fields and buttons to open associated browsers are as follows.

Field	Button	Opens
Point ID		 **Note:** The CIMPLICITY `$OBJECT` variable appears in the **Point ID** field before the data item as follows: `{$OBJECT).DataItem`
Source		
Alarm ID		
Class ID		CIMPLICITY Select an Alarm Class browser
Resource ID		CIMPLICITY Select a Resource browser

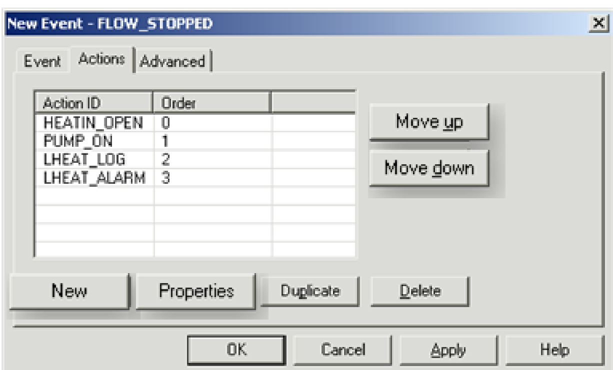
Event Time		Timed event provides fields to enter The start time and interval for an event. 
Event int		

Actions Tab

Actions listed on the Actions tab are associated with the selected event. .

Buttons on the Actions tab enable you to open a New Actions dialog box to add an additional action.

Note: You can remove any action's association with the event by selecting an ID and clicking Delete.



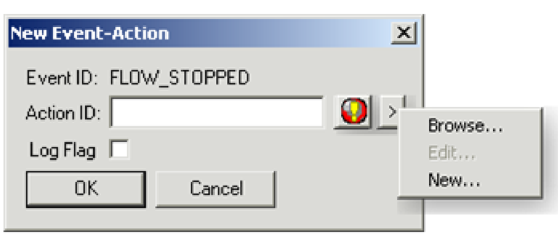
- New action association.
- Log Flag checkbox.

New Action Association

Click New on the Actions tab in the New Event dialog box.

A New Event-Action dialog box opens.

Selections are as follows.



Action ID

Methods to select an action ID are as follows.

- Browse existing actions.
- Create a new action.

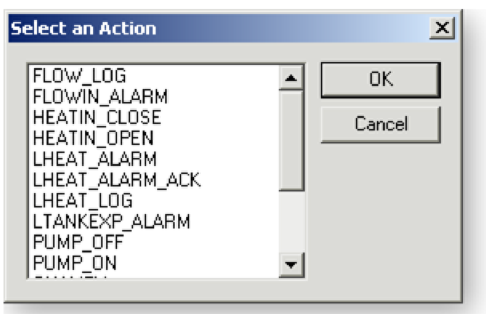
- Browse existing actions.

1. Do one of the following.

- Click the Browse Actions button  to the right of the **Action ID** field.
- Click the Popup Menu button to the right of the Action ID field; select Browse on the Popup menu.

A Select an Action browser opens.

1. Select an action; click OK.



- Create a new action.

1. Click the Popup Menu button to the right of the Action ID field; select New on the Popup menu.

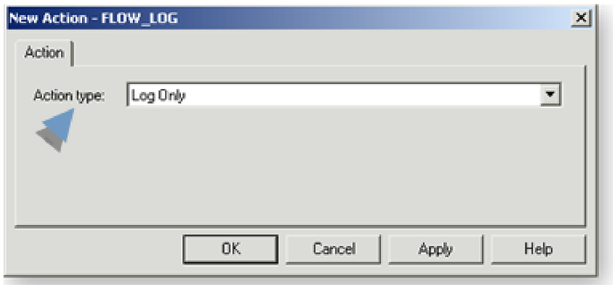
A New Action dialog box opens.

1. Enter a unique action name in the **Action ID** field.

1. Click OK.

A New Action dialog box opens for the entered action

1. Define the action.



1. Click OK.

Result: The New Event-Action dialog box displays when you use either method.

Log Flag Checkbox

Check Log Flag if the action should be logged to a database table.

Result: When the New-Event Action dialog box is closed a new class action is listed both on the Actions tab in the Class dialog box and at the end of the list of actions associated with the event.

Note: Select the action and click the Move Up button if it should be located higher up in the list.

Existing Event-Action Properties

1. Select an existing event-action.
2. Click Properties.

   The New Event-Action dialog box opens for the selected event-action.

3. Do one or both of the following.
   - Edit the selected action.

   a. Click the Popup button to the right of the Action ID field; select Edit.

   The New Action dialog box opens for the selected action.

   a. Make any required changes to the action configuration.
   b. Click OK.

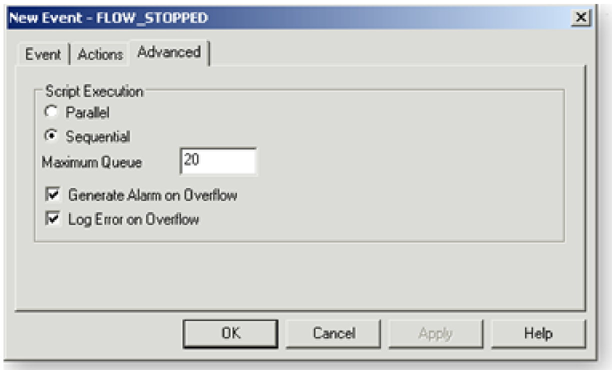
   The action is modified.

- Check of clear the Log Flag check box.

When the New-Event Action dialog box is closed the class action that is associated with the event is modified.

Advanced Tab

Advanced tab options are as follows.



Configure advanced specificationsthe same as you configure them for an event in the Events folder.

When an object is created from the class, the event displays in the CIMPLICITY Event Editor window as:

ObjectName.EventID

Where

ObjectName is the name of the object that has been created.

EventID is the name entered in the New Event dialog box.

Supported String Substitution for Event Fields

String substitution is supported for the following event fields, which you enter in the associated Event dialog box field.

- Point ID.
- Alarm ID.
- Resource ID.
- Alarm Class.
- Point Value.

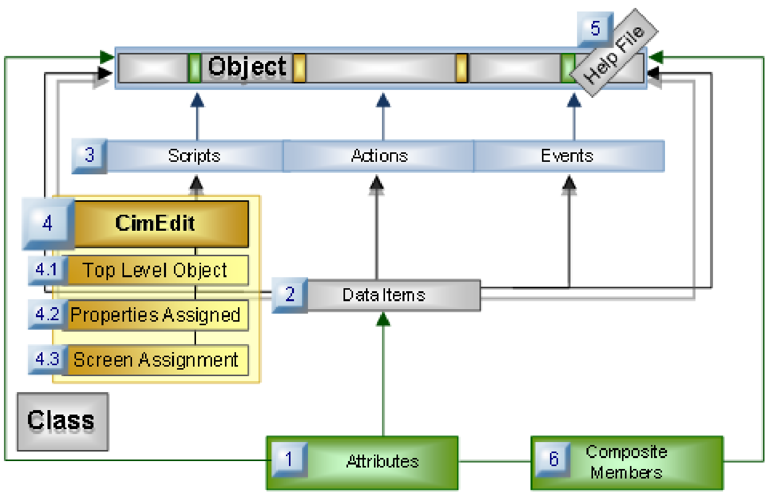
4. CimEdit Class Screens

4. CimEdit Class Screens

- Overview: CimEdit class screens configuration.
- Overview: CimEdit object instantiated into a class object screen.
- Configuration: CimEdit class screens.

Overview: CimEdit Class Screens Configuration

CimEdit screens can be created with graphic objects that include data items. When screens are applied to class objects, the configuration will be instantiated into a dynamic CimEdit/CimView screen that reports the selected class object values.



4.1 (page 107)	Top level CimEdit group object in a class.
4.2 (page 111)	Class properties assigned to a CimEdit object.
4.3 (page 112)	Class screen assignment.

Overview: CimEdit Object Instantiated into a Class Object Screen

A (page 105)	Class: CimEdit Group Object Configuration

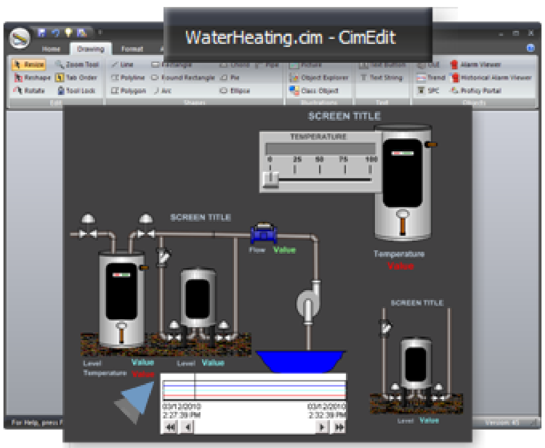
B (page 106)	Object: Object created.
C (page 106)	Instantiation: Object Screen

1. Class: CimEdit Group Object Configuration

One or more objects are configured on a CimEdit screen.

Example

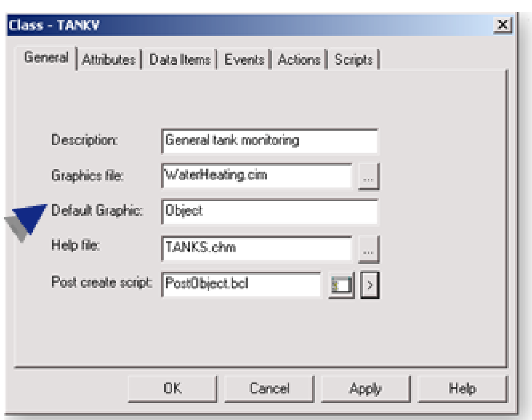
A CimEdit screen, WaterHeating.cim includes three group objects; the largest object is named Object,



The CimEdit screen and default object are indicated on the General tab in the Class dialog box.

Example

A class TANKV selects WaterHeating.cim as its graphics file and Object as its default graphic.



1. Object: Object Created

A class is selected to create an object.

Example

A class object created from the class TANKV is named: STATION04.

1. Instantiation: Object Screen

An Object screen can display one or more of the objects created on the CimEdit screen.

A class object the first object to display be the default graphic entered in the Class dialog box. That graphic can be changed after the screen is created.

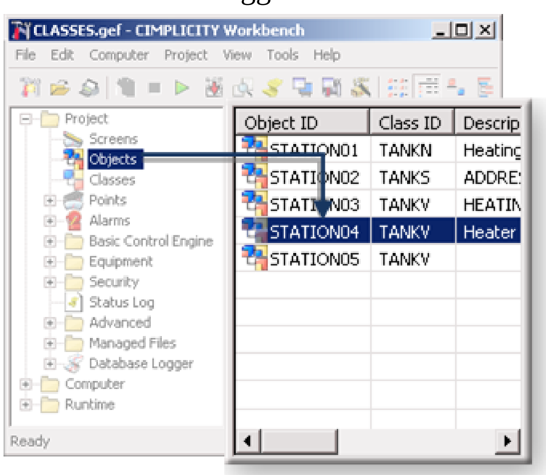
An object is dragged from the Workbench onto a CimEdit Screen; the default graphic displays on the screen.

The source object from the class CimEdit screen appears on the new screen. Object values are substituted for variables, e.g. **$OBJECT** , that were configured on the source screen.

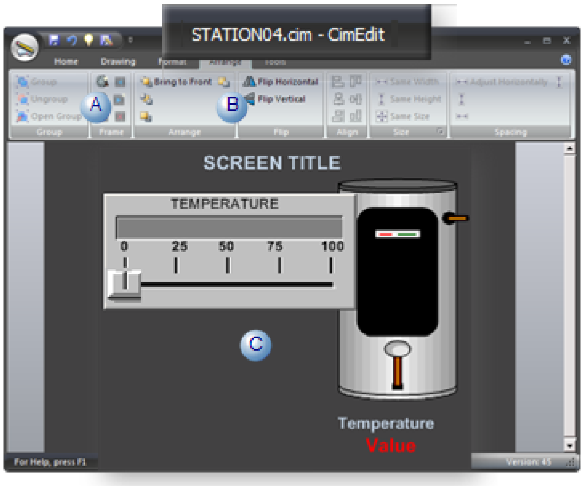
Note: Even though the class object is created from a group object, it cannot be opened; configuration in is disabled.

Example

1. STATION04 is dragged to a CimEdit screen.



2. The following occurs.



A	Configuration features (e.g. Ungroup, Open Group) are disabled.
B	Display features (e.g. Bring to front) are enabled.
C	Another graphic, TemperatureHeater, is selected from the list of graphics that are on the WaterHeating.cim screen.

Configuration: CimEdit Class Screens

4.1 (page 107)	Top level CimEdit group object in a class.
4.2 (page 111)	Class properties assigned to a CimEdit object.
4.3 (page 112)	Class screen assignment.

4.1. Top Level CimEdit Group Object in a Class

The CimEdit group object can be created to be the source for graphics that are created for a CIMPLICITY class object.

Define a top level group object as follows.

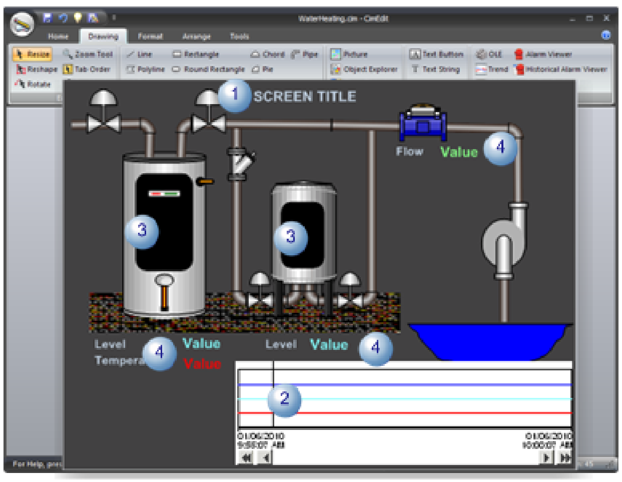
A (page 108)	Place objects on a CimEdit screen.
B (page 108)	Group the objects into a Class Group Object.

C (page 109)	Identify the group object.
D (page 110)	Assign an $OBJECT variable to the group object.
E (page 110)	Create additional objects.

1. Place Objects on a CimEdit Screen

You can use any of the thousands of CimEdit objects on a class CimEdit screen. The type and number of objects you place on the screen depends on the class requirements.

The following are examples of objects that can be replaced with values for a class object.



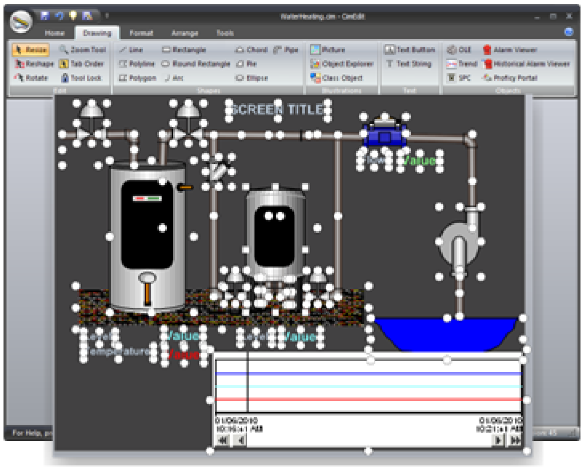
1	Text string.
2	Trend ActiveX object.
3	Fill in a tank group object from Object Explorer.
4	Numeric text to track tank data.

1. Group the Objects into a Class Group Object

The class group object becomes the source for graphics that are created for CIMPLICITY class objects.

1. Select the objects that should be included in the group object.

   Note: You can create more than one group object, then select one to be the default *(page 112)*
   .



2. Groupthe selected objects the same way you group any objects on a CimEdit screen.

3. Identify the Group Object

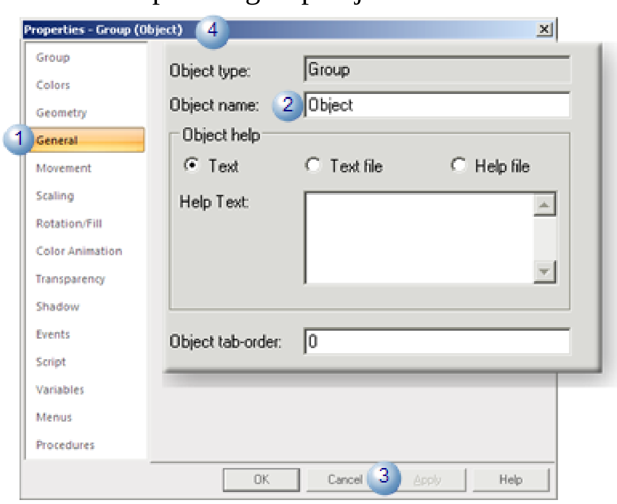
   A CimEdit class group object requires a name.

4. Select the class group object.

5. Open its Object - Group Properties dialog box, using any of the methods provided by CimEdit, for example:
   a. Right-click the group object.
   b. Select Properties on the Popup menu.

   The Properties - Group dialog box opens.

6. Name the top-level group object as follows.



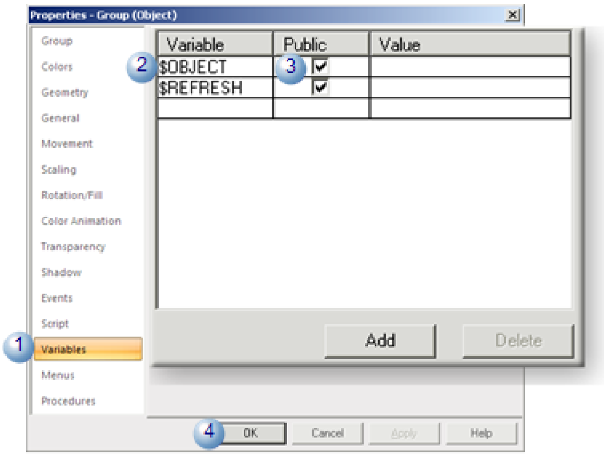
1	Select **General**.
2	Enter a name in the **Object name** field.

3	Click Apply.
4	The Object name (e.g. Object) displays in the Properties - Group dialog box title bar.

a. Assign an $OBJECT Variable to the Group Object

A variable must be assigned to the class (group) object that CIMPLICITY will substitute with the appropriate (class) object name.

Assign the CIMPLICITY **$OBJECT** variable to the top level group object as follows:



When an object that is created from the class is dragged into CimEdit, a class object graphic is created. The name of the object is substituted for {$OBJECT} in expressions during runtime.

1	Select Variables.
2	Enter `$OBJECT` in the **Variable** field.
3	Check Public.
4	Click OK.

Note: The `$OBJECT` variable on the Variables tab is read-only when a CimEdit screen designer opens the Properties - Class Object *(page 125)* dialog box.

1. Create Additional Objects

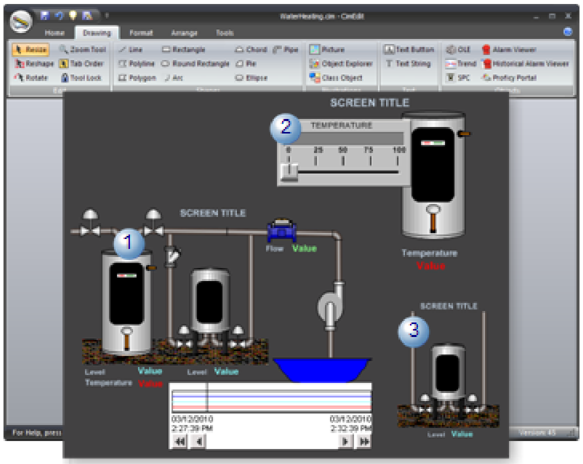
Follow the same procedure to create as many additional group objects as necessary.

Any of the group objects can be selected as the default object for a class object.

Example

A CimEdit screen named WaterHeating.cim includes three objects.

Each group object is assigned a unique name.



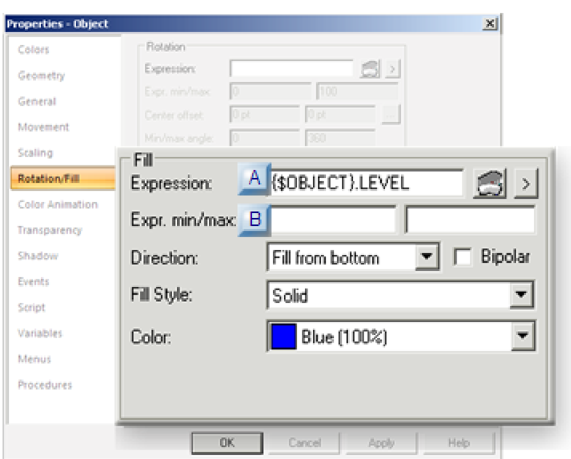
1	object
2	TemperatureHeater
3	TANKEXP

4.2. Class Properties Assigned to a CimEdit Object

You can assign expressions and variables to a group of objects and objects that are included in the top-level group the same way you assign them on any CimEdit screen.

The values you enter are values from the Class configuration, e.g. data items that become point IDs when an object is created from the class.

Two CimEdit Properties dialog box field types that can take advantage of class configuration are as follows.



rect 147, 120, 169, 138 (page 112)
rect 146, 94, 168, 112 (page 112)

A (page 112)	Expression field.
B (page 112)	Expr. min/max field.

A	Expression Field

The **Expression** field that is configured for a class requires the following.

- The data item((or data items) that is entered in the **Expression** field must be manually entered.

Note: Only points are listed in the Select a Point browser.

- The predefined variable, `{$OBJECT` (page 39) `}`, is required to hold the place for the object name.

The syntax is:

`{$OBJECT}.data item`

Where

`{$OBJECT}` references the object that is created from the class.

Data item indicates the class data item.

Example

`{$OBJECT}.LEVEL`

B	Expr. min/max Fields

When an instantiated object uses the configured graphic, CimEdit looks for the selected instantiated point's display values:

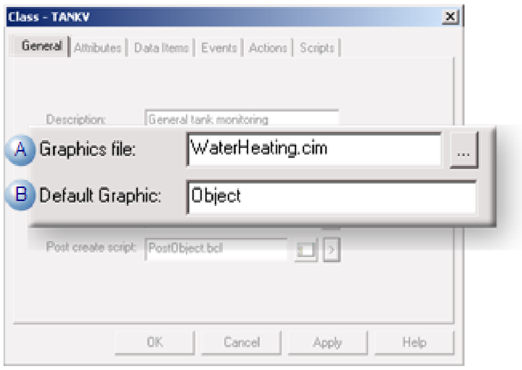
1. First in the CimEdit Properties dialog box.
2. Second in the point's **Display low** and **Display high** fields.

4.3. Class Screen Assignment

1. Open the Class dialog box for the class with which the screen will be associated.

2. Select the General tab.

3. Enter the following.



	Field	Enter
A	Graphics file	CimEdit screen name that will be associated with the class. The selected CimEdit screen contains the source objects that are available to object designers.
B	Default Graphic	Name of the CimEdit group object that displays as the default when a class object is dragged to a new CimEdit screen.

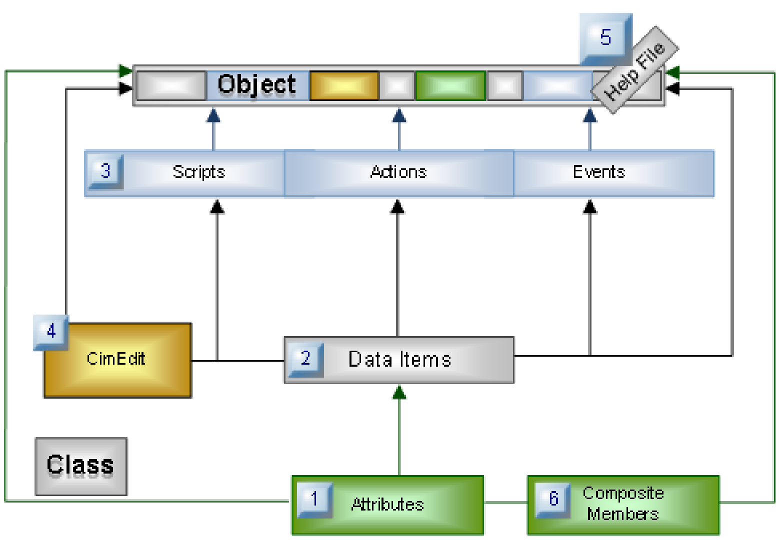
When an object that is created from the class is dragged from the Workbench into a new CimEdit screen, a class object graphic, which is linked to the class source graphic displays.

Note: An object designer can select another object in the default object's Properties dialog box.

5. Class Help File

You can associate a custom help (.hlp) file that is created using a third party tool with a class. The help file will assist object designers when they are configuring point objects based on the point class.

In order to make the help file specific to the class you are creating, it is recommended that you write the help file after you configure the other class components.



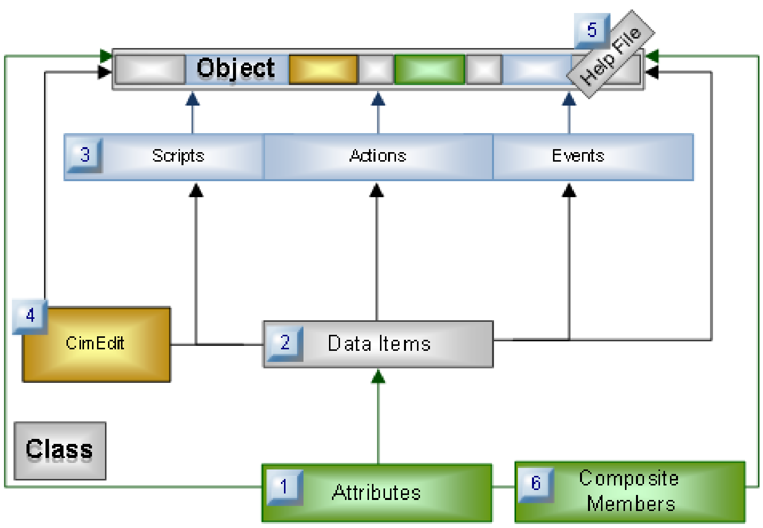
Note: Refer to the documentation from your help-based software for details about creating custom help files.

Associate a custom help file with a class

1. Select the General tab in the Class dialog box.
2. Enter the name of the help file in the **Help file** field.

When an object designer presses F1 or clicks Help in the Object dialog box, the table of contents for the specified help file displays.

6. Composite Classes and Objects



About Composite Classes and Objects

Beginning in CIMPLICITY 9.5, there are Composite Classes and Objects. Classes can reference other classes. You can modify a class dynamically and you can also delete a class dynamically if there are no references to the class or object instances of that class with composite references to the class.

Advantages of Composite Classes and Objects

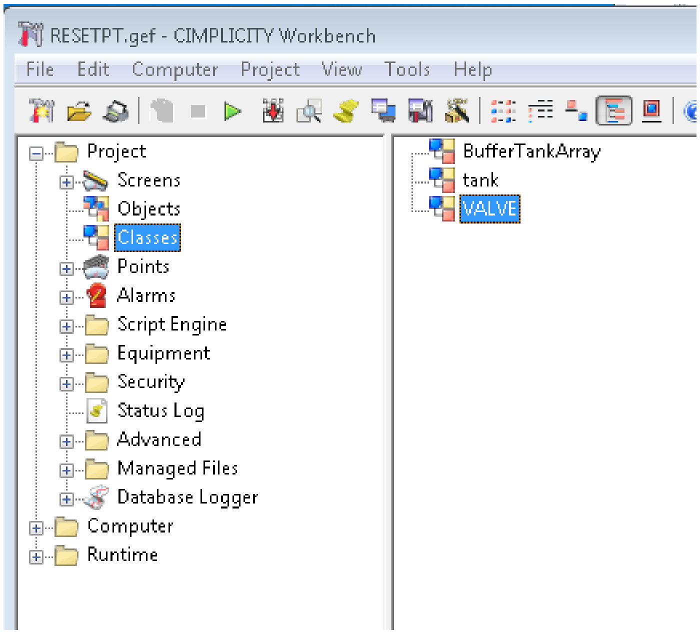
Well-planned composite classes and objects reduce the amount of work required to configure a CIMPLICITY project. Read more about Composite Classes here (page 114).

About Composite Classes

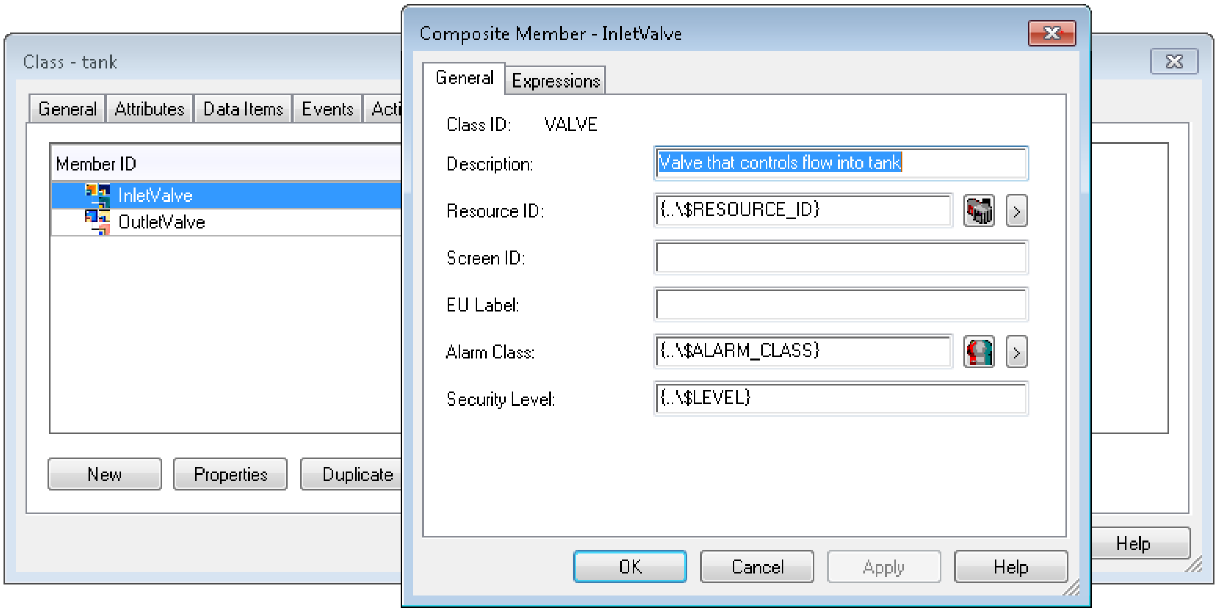
Composite Classes are classes that contain another class or classes. The Composite Class has all of the attributes of any class or classes it contains to reduce the work involved in creating CIMPLICITY objects.

Example of Composite Classes

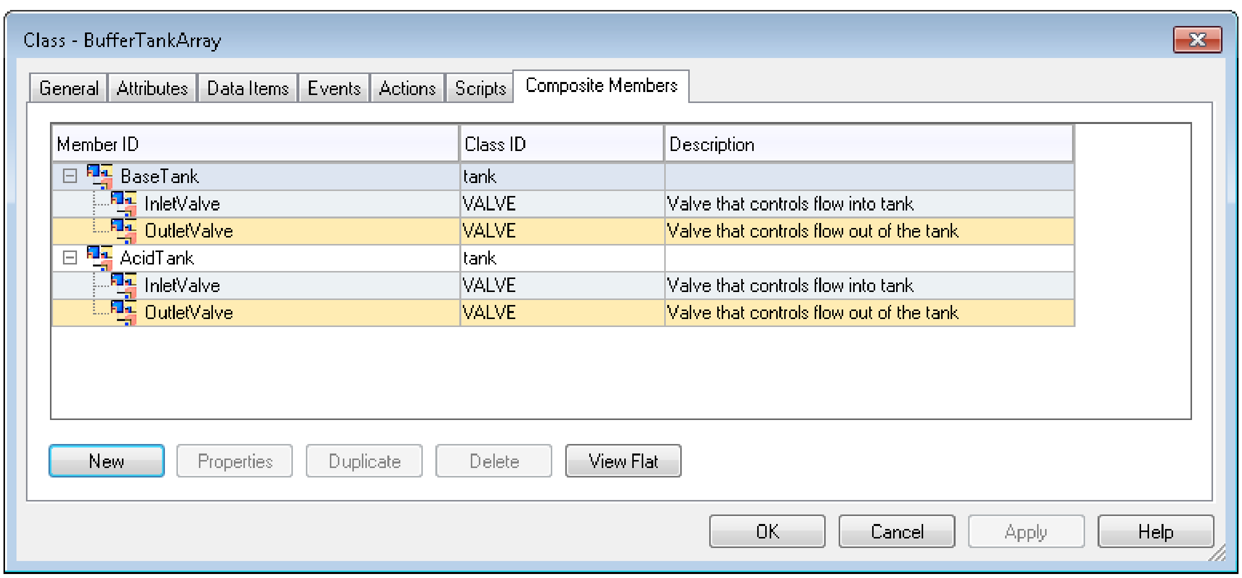
In this example, a CIMPLICITY project named RESETPT.gef, there are three classes, BufferTankArray, tank and VALVE.



The class named Valve is a low-level class. It has no composite members.



However, BufferTankArray, contains two composite classes. InletVale and OutletVale are both defined with the Class ID of "tank." Therefore, they contain any attributes contained in the class "Tank." Using composite classes, you only need to define the class "Tank" once and then reuse it over and over again when designing your project.



Note: Composite Class inherit just the attributes of the classes they refer to. They do not inherit scripts, events or other features of they classes they refer to.

Character Limit and Composite Classes

The composite member name is used in the name of the points for data item in the class that get created for each object instance, and there is a limit of 256 characters for the fully realized pointIDs.

To stay under the character limit, carefully consider the naming you use when creating composite member names. For example, a class that contains a composite member named FillValve, of class valve, would create points with the following names (if the Valve class had data items FlowRate, and ValvePosition.)

<$ObjectID>.FillValve.FlowRate

<$ObjectID>.FillValve.ValvePosition

Composite Members Can Refer in Both Directions

Not only do container classes automatically have the values of classes that it contains, you can have composite member attributes use the values of the attributes of its container classes. When supplying the value of a composite members attributes within the class definition, you can refer to the container attributes by using a syntax {..\<AttributeName>}. This syntax would refer to an attribute in the immediate container class and {..\..\<AttributeName>} refers to an attribute two levels up in the container hierarchy.

Export or Import a Class

Export or Import a Class

The power of the class becomes particularly apparent when you export a class from one project and import it into another.

Options include:

Option 1 *(page 117)*	Export a class.
Option 2 *(page 118)*	Import a class.

Option 1. Export a Class

You can export a point class.

Exported point class files:

- Have a **.soc** extension
- Can be imported into other CIMPLICITY projects.

Important: Any changes you make to a class within a project will not be reflected in an exported class until it is re-exported.

1. Right-click a class in the Workbench right pane.
2. Select **Export** on the popup menu.

   The Save As dialog box opens.

3. Select the folder in which the file should be saved.
4. Save the file using the syntax:

   `<ClassName>.soc`

   Example

   The class TANKV will be exported.

   The export file name is:

   TANKV.soc

5. Click Save.

CIMPLICITY gathers all the class components and includes them in the .soc file as follows:

Component	Folder Retrieved from:
Class configuration	Classes
Scripts (.bcl) files	Scripts
CimEdit (.cim) files	Screens

A class developer can import the class into a different project and work with the class configuration.

Option 2. Import a Class

You can import a class that has been exported to a **.soc** file into another CIMPLICITY project using the class popup menu options.

1. Right-click the **Classes** icon in the Workbench left pane.
2. Select **Import** on the popup menu.

The Open dialog box opens.

3. Open an import (.soc) file as follows.
4. Click Open.

   The Workbench displays.

5. Press **F5** to refresh the screen.

CIMPLICITY distributes the .soc file components as follows:

Component	Folder Distributed to:
Class configuration	Classes
Scripts (.bcl) files	Scripts
CimEdit (.cim) files	Screens

A class developer can now modify the class to meet any different requirements for the project.

An object designer can immediately implement objects with associated CimEdit/CimView screens.

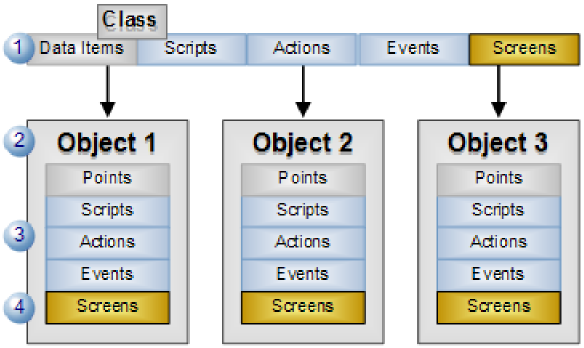
Class Object Configuration

About Class Objects

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.

When creating a class object, an object designer simply needs to specify the values that pertain to that class object. Once specified, the class object's features achieve full status within a CIMPLICITY project.

For example, class object points behave as any other points with the same type of configuration; class object events function the same as any other similar events.



1	A class file (.soc file) can have several components, including CimEdit screens and developer created help.
2	One or more objects can be created from the class.
3	Objects include instantiated parts that are included in the class.
4	One or more CimEdit screens can be created for each object.

Although class object features behave the same as their non-class counterparts, CIMPLICITY makes it easy for the object designer (or any project designer) to locate features for any class object. CIMPLICITY attaches the object name to the front of the feature name.

Important: Before you can configure a class object, a class must be created *(page 14)*. If the class exists, but is not in the current project, you simply need to import *(page 118)* it into the project.

Class Object Configuration

Class Object Configuration

Creating a class object is straight-forward. Because the object is based on a class template, most of the configuration is already done.

Following are steps to configure one or more class objects.

Step 1 *(page 121)*	Open an Object dialog box.
Step 2 *(page 124)*	Assign values to the class object.
Step 3 *(page 125)*	Display class object graphics in CimEdit/CimView

Step 4 (page 127)	Add a class object to a project.

Step 1. Open an Object Dialog Box

Step 1. Open an Object Dialog Box

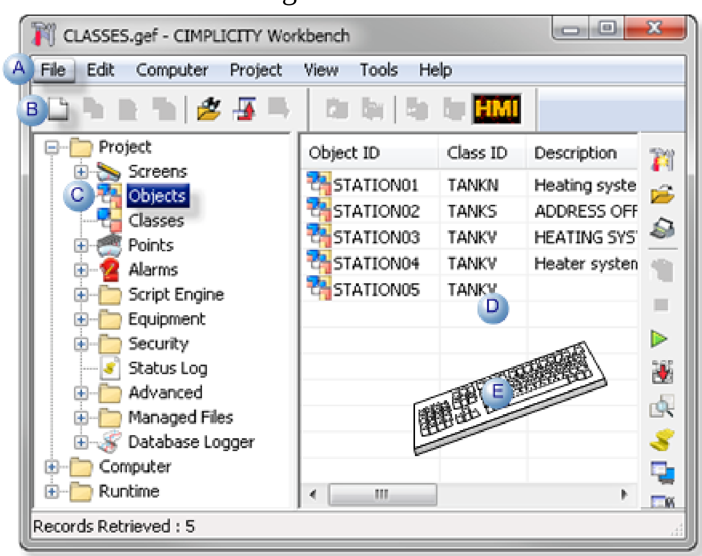
Note: Classes can be created in the same project in which objects are made or imported (page 118) from other projects.

Option 1.1 (page 121)	Create a new class object.
Option 1.2 (page 123)	Open an existing Object dialog box.

Option 1.1. Create a new Class Object

1. Select **Project>Objects** 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	

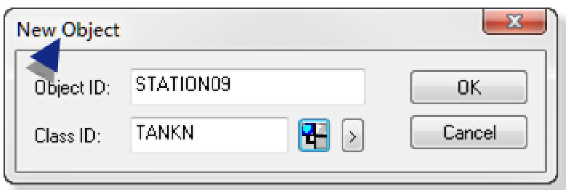
	Double click **Object**.
D	a. In the Workbench right pane. a. Right-click anywhere. b. Select New on the Popup menu.
E	Press Ctrl+N on the keyboard.

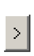
A New Object dialog box opens when you use any method.

3. Right-click anywhere.

4. Select New on the Popup menu.

5. Fill in the fields as follows.



Field	Description
Object ID	A unique name for the new object. The Object ID: • Can contain alphanumeric characters and underscores. • Must begin with an alphabetic character • Has a maximum length of 255 characters, which includes the following three components. a. Object name. b. . (separator). c. Data item name. The three components can total 255 characters, and the Object ID becomes the point name when the object is instantiated. Example a. An object ID is TANK02 b. A data item name is FLOWLEFT c. The separator is always . d. The point name is TANK02.FLOWLEFT. TANK02.FLOWLEFT is only 15 characters; therefore, it is a valid name.
Class ID	Class that are used as the template for the class object. (Optional) Click the buttons to do the following.
	 Opens the Select a Class browser.
	 Displays a New/Edit/Browse pop-up menu.

6. Object name.

7. . (separator).

8. Data item name.

The three components can total 255 characters, and the Object ID becomes the point name when the object is instantiated.Example

9. An object ID is TANK02

10. A data item name is FLOWLEFT

11. The separator is always .

12. The point name is TANK02.FLOWLEFT.
    TANK02.FLOWLEFT is only 15 characters; therefore, it is a valid name.

13. Click OK.

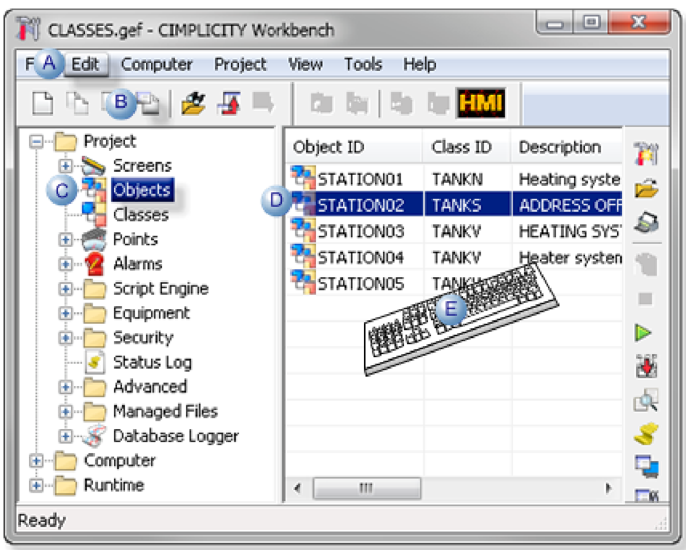
The Object - <object name> dialog box opens for the new object.

<u>Option 1.2. Open an Existing Object Dialog Box</u>

1. Select Project>Objects in the Workbench left pane.

2. Select an object 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 **Object**. b. Select Properties on the Popup menu.
D	In the Workbench, double-click an object.

E	Press Alt+Enter on the keyboard.

4. Right-click **Object**.

5. Select Properties on the Popup menu.

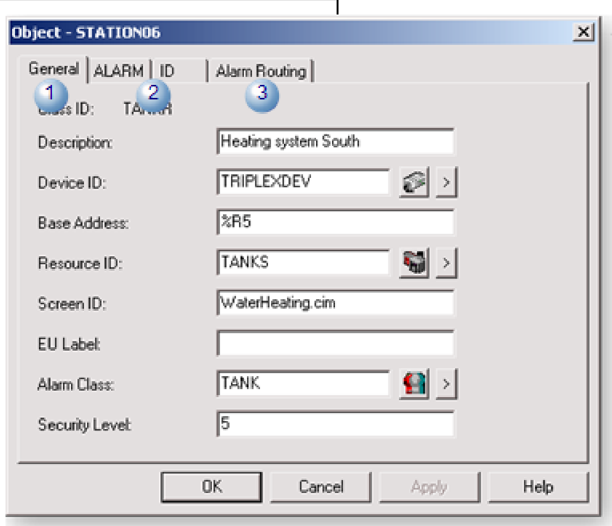
Step 2. Assign Values to the Class Object

The type of information available to you depends on the attribute properties created for the class.

When you create a class object, an Object dialog box that reflects the class configuration opens. (The object ID displays in the dialog box's title bar.)

The class object adheres to specified requirements based on the class definitions. You assign the values of these requirements in the Object dialog box.

Customized class features that display in the dialog box can include:

Customized Feature	Description
Fields	The fields that display and whether or not they are write-able come from the class configuration.
Default Entries	Any default entries in fields come from the class configuration.
Tabs	The Object dialog box displays.
	

Enter values in the Object dialog box fields that define the specific criteria for the class object that is being configured.

When appropriate, the **Browse** buttons that display throughout CIMPLICITY, display to aid you search for the correct entries.

Option	Browse Button that will display
Alarm Class	
Resource	
Device	
Role	
Point	
User	
Port	

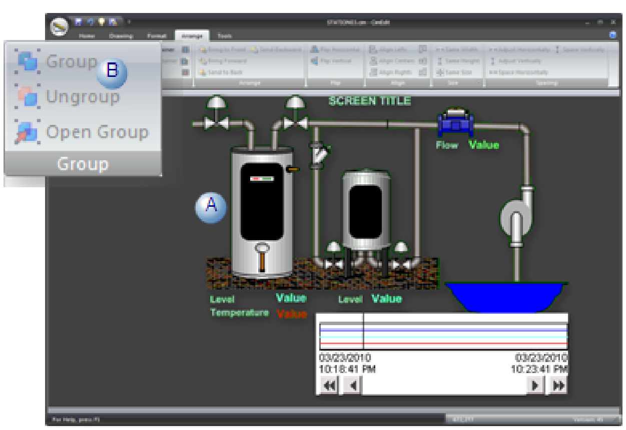
Step 3. Display Class Object Graphics in CimEdit/CimView

1. Make sure the project is running.
2. Open a CimEdit screen.

   The CimEdit screen can either be a new or existing screen.

3. Drag a class object onto the CimEdit screen.

   The following happens.



A	The default graphic for the class displays.
B	CimEdit functionality (e.g. Ungroup, Open Group) is disabled.

Enter Class object Specifications

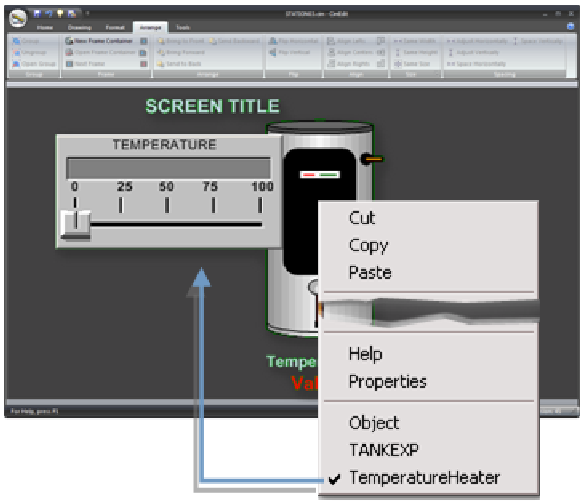
Limited changes can be assigned to the graphic object.

Right-click the CimEdit class object.

Popup Menu Options

The Graphic objects are listed on the Popup menu that included in the CimEdit screen that is assigned (page 104) to the class.

Select any listed object to replace the object that is currently selected.



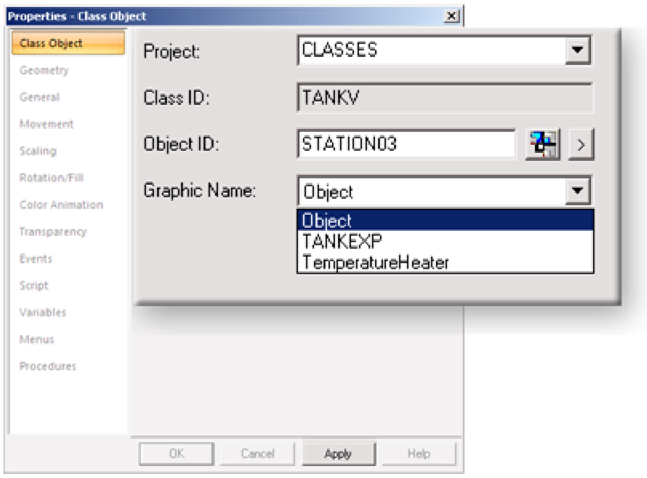
Properties Dialog Box Options

4. Right-click the class object graphic.
5. Select Properties from the Popup menu.

   The Properties - Class Object dialog box opens.

6. Select Class Object.

   Class object options are as follows.



Option	Description
Project	Project with which the class object is associated.
Class ID	Class that contains the CimEdit graphic (read-only).
Object ID	Selected class object (read/write). The ID for any object in the class can be selected if you want to associate the graphic with a different object from the one dragged onto the CimEdit screen. **Note:** Click the Browse button to the right of the **Object ID** field to display a list of all of the objects in the class.
Graphic Name	Selected class object graphic (read/write). Any graphic from the source CimEdit screen can be selected if you want to use a class graphic that is different from the default.
Apply	**Important:** Clicking **Apply** updates the `$OBJECT` variable to the new value. The `$OBJECT` variable, which was created during class object configuration *(page 39)*, is read-only on the Variables tab in the Class Object dialog box.

Result: The graphic will adhere to your specifications as follows:

7. During runtime the graphic will represent the values of the selected class object.
8. The graphic wills change to the graphic you selected, if it is different from the default, as soon as you close the Properties - Class Object dialog box.

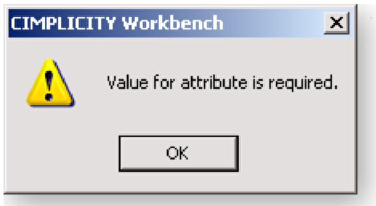
Step 4. Add a Class Object to a Project

When all of the required values have been assigned and any optional values to support your class object, add the class object to your project.

- Click **OK** in the Object Properties dialog box.

Result: CIMPLICITY takes the values you enter in the Object dialog box and applies them to the appropriate feature in CIMPLICITY, e.g. points, alarms. You can also do more specific configuration throughout the Workbench.

Important: You will not be able to complete a tab's configuration until you fill in all of the required fields. If you try, an error message opens telling you that a value for an attribute is required. This message continues to display when you attempt to exit the dialog box until all required fields are filled in.



Chapter 2. Logging and Archiving

Historian OPC Interface

About CIMPLICITY Integration with Historian

Historian enables you to archive data for historical retrieval.

CIMPLICITY provides a straightforward OPC interface that enables you to easily connect to Historian to archive CIMPLICITY point data.

Pre-requisites

Ensure that you have installed the following components:

Component	Help Link
Historian Server **Warning**: In Historian 9.0, when prompted for the UAA location during this install, be sure to select the URL where you have installed or where you will install Historian Web-clients.	https://www.ge.com/digital/documentation/historian/version90/t_hgs_installing_single_server_historian.html
Historian Collectors	https://www.ge.com/digital/documentation/historian/version90/c_about_installing_collectors.html
Historian Client Tools	https://www.ge.com/digital/documentation/historian/version90/t_hgs_installing_historian_client_tools.html
Historian Web-clients **Warning:** Generally, to work with Historian 9.0, Historian Web-clients must be installed in order to set up data and alarm collection from a CIMPLCIITY node unless upgrading from using previous versions of Historian with your CIMPLICITY project(s).	https://www.ge.com/digital/documentation/historian/version90/c_about_installing_web_based_clients.html

Historian Server must be fully configured, either on a CIMPLICITY Server or on a remote server. The Historian Database and Historian Alarm/Event Database must be configured and functional.

Note: Starting Historian 9.0, you will need to add OPC and OPC A & E collector instances using Configuration Hub. Refer https://www.ge.com/digital/documentation/historian/version90/t_add_collector_instance.html

Note:

- Historian provides detailed documentation to configure and use Historian. This documentation describes CIMPLICITY integration with Historian.
- Historian 3.1x and lower are not supported on Vista or Windows Server 2008.

Step Number	Description
Step 1 *(page 130)*	Select Alarm and Event Archiver during Historian installation.
Step 2 *(page 132)*	Enable the Historian OPC Interface.
Step 3 *(page 141)*	Select points to be logged to Historian.
Step 4 *(page 144)*	Open the Historian Administrator System Statistics window.
Step 5 *(page 145)*	Review CIMPLICITY point (tag) details in Historian.
Step 6 *(page 149)*	Display CIMPLICITY alarm data in Historian.
Step 7 *(page 151)*	Set up Historian connections to collect data.

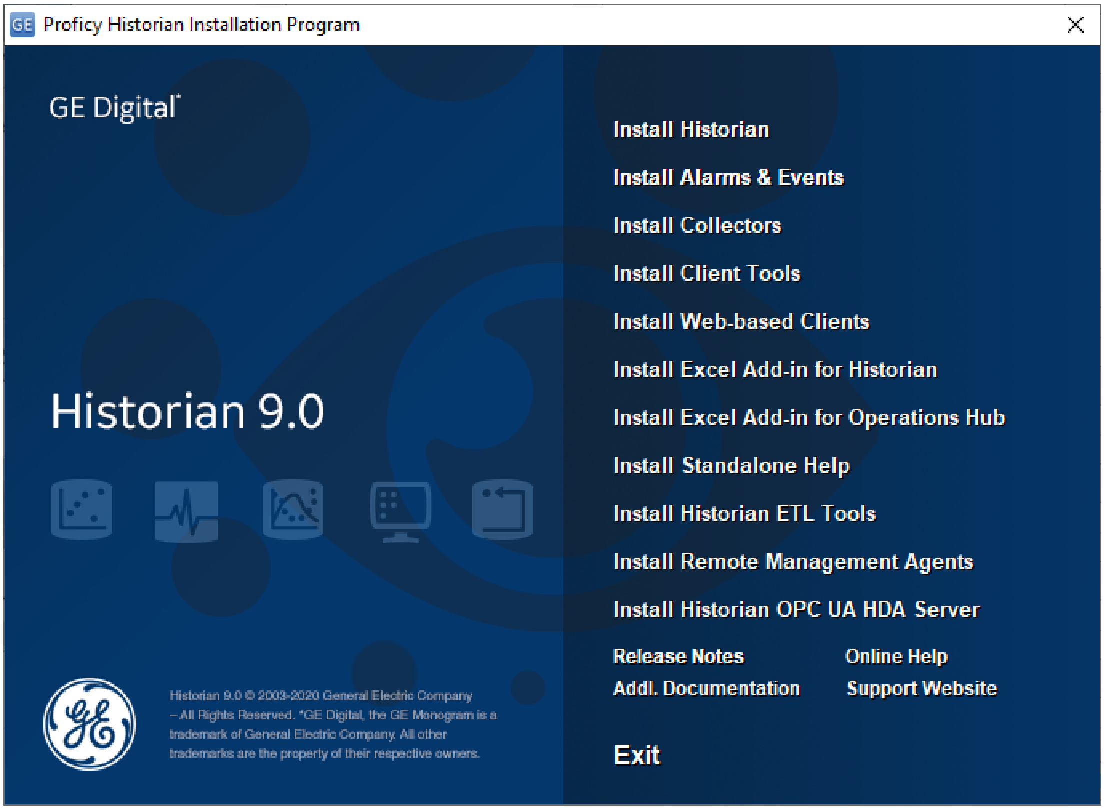
Step 1. Select Archive Features during Historian Installation

Pre-requisites:

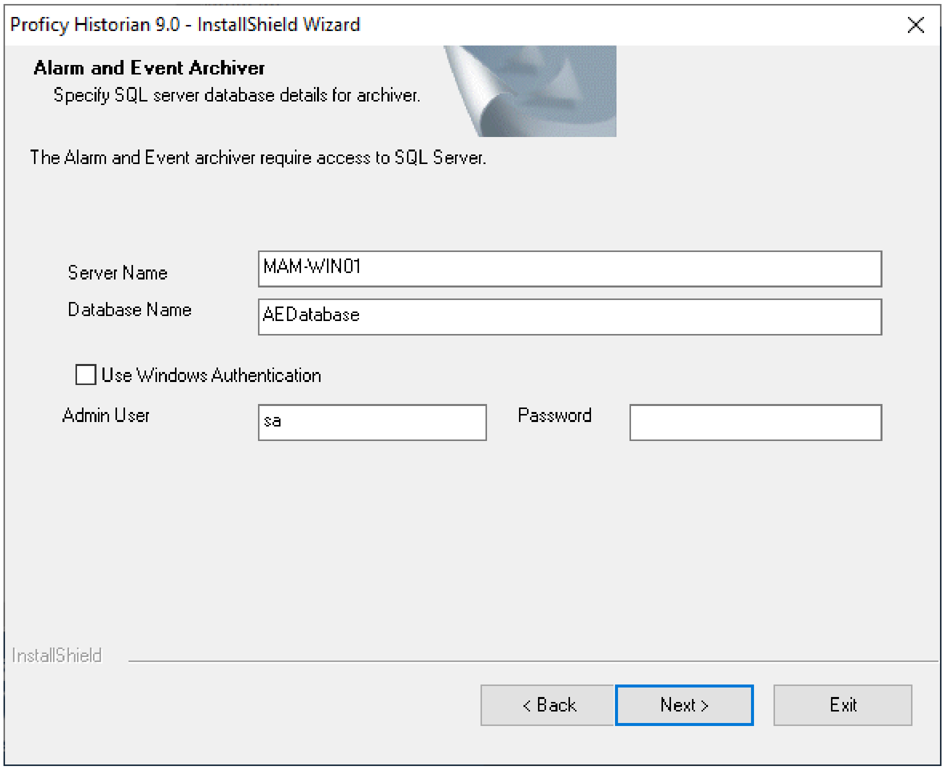
You must install Historian Server and Historian Viewer on all CIMPLICITY machines before configuring a project to log to the Historian. Install the following components on every node/ machine:

- Historian Client Tools
- OLE DB
- User API

1. The servers must have OPC Collectors installed for the OPC Interface and the OPC A&E Interface. If both options are used, both collectors must be installed.



To enable the archiving features select Install Alarms & Events and specify SQL Server database details during installation.



Note: To successfully create Historian Alarm and Event database during installation, you must ensure that SQL Server agent is in running state.

Result: The selected Historian archive features will enable Historian to store CIMPLICITY alarm and event data.

Step 2. Enable the Historian OPC Interface(s)

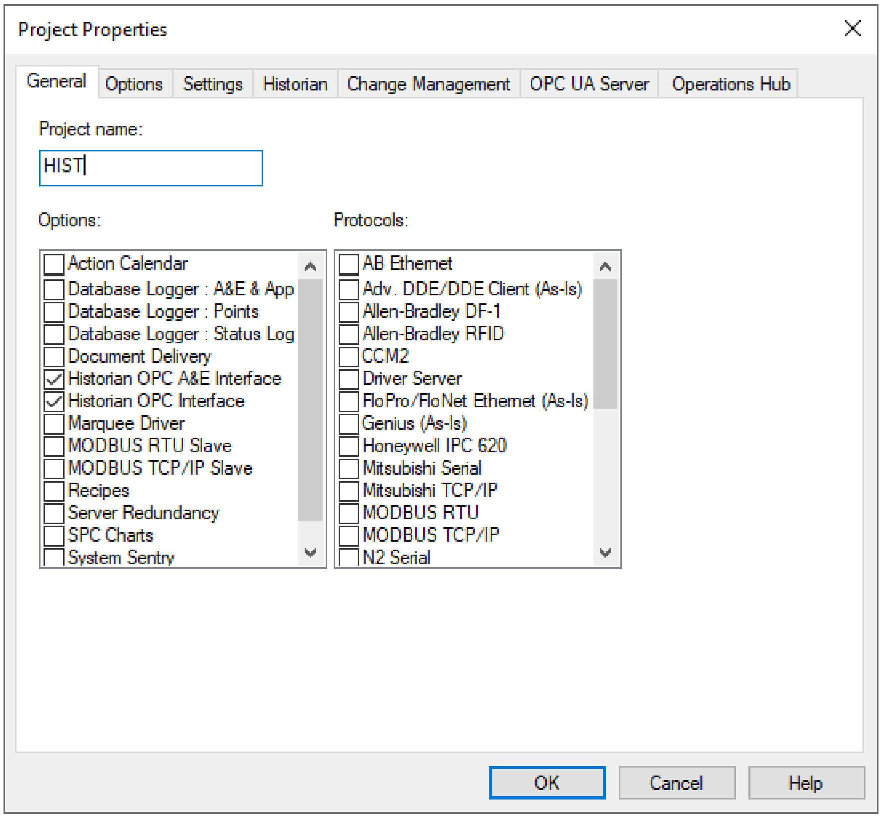
Step 2. Enable the Historian OPC Interface(s)

Important: Make sure Historian is installed and accessible to the CIMPLICITY project.

Step 2.1 *(page 132)*	Select the Historian logging option(s).
Step 2.2 *(page 134)*	Define the Historian connection.

Step 2.1. Select the Historian Logging Option(s)

1. Open the CIMPLICITY project with data that will be sent to Historian.
2. Open the **Project Properties** dialog box.
3. Select the **General** tab.
4. Select either or both of the following.



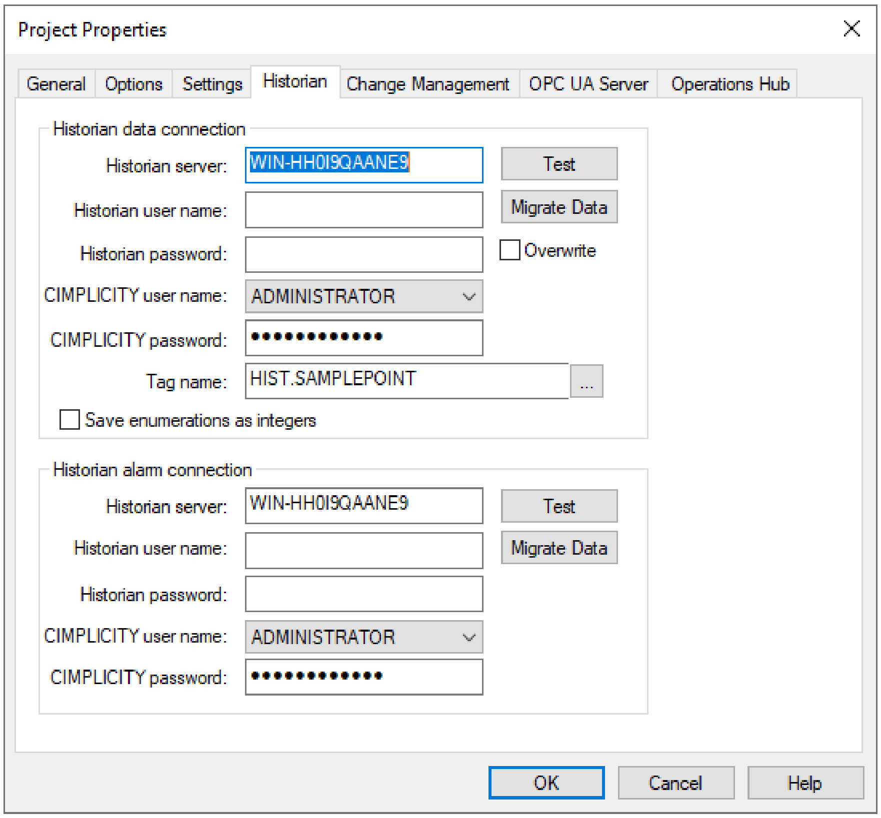
A	Historian OPC A&E Interface CIMPLICITY: • Logs alarm and event data to Historian through the OPC Alarm & Event server. • Maps *(page 297)* fields in the ALARM_LOG and EVENT_LOG to be consistent with fields provided by the OPC Alarm & Event server.
B	Historian OPC Interface CIMPLICITY • Logs data to Historian through the CIMPLICITY OPC server. • Maps *(page 297)* fields in the DATA_LOG to resolve differences in how data is identified between the CIMPLICITY DATA_LOG and the real-time Collector.

Note: This enables the Historian OPC Interface for this project.

A Historian tab displays in the Project Properties dialog box when you select either or both Historian OPC options.

Step 2.2. Define the Historian Connection

1. Open the CIMPLICITY Workbench and select **Project** from the menu bar.
2. Click **Properties** to open the Project Properties dialog per the example below.



NOTE: If you added Historian as part of your project setup, the **Historian** tab is visible. If not, you can select one or both of the Historian check boxes that appear in the list on the **General** tab.

If you select the **Historian OPC Interface** check box, you enable the Historian Data Server.

If you select the **Historian OPC A & E Interface** check box, you enable the Historian Alarm Server.

3. Select the **Historian** tab and complete the fields as described below.

For the Historian data and alarm connections:

4. In the top section, enter the name of your **Historian Data Server** and in the bottom section, enter the name of your **Historian Alarm Server.**

5. In each section, enter the **Historian username** and **Historian password** used to access your Historian Data Server and Historian Alarm Server, respectively. If these are not entered correctly, the logging will fail.

   These entries are not always required; instances where an entry should be made include the following:

   - The Historian Server and users who log into that server are different from the CIMPLICITY Server and user.
   - A user who is logged into the Historian Server may not have all of the privileges required to manage logging CIMPLICITY data. An entry in this field can specify a user with administrator privileges.

6. The **CIMPLICITY username** automatically populates from your project.

7. Enter the corresponding **CIMPLICITY password**. Note that privileges may differ between the Historian user and the CIMPLICITY user.

8. Keep the default Tag name convention or identify a new one. For information, see the Tag naming convention section.

9. Click **Test** to test your connection to the Historian server. One of the following messages appears:

Test Result	Message
The Historian server is incorrect or not available.	Failed to connect to the Historian server.
Historian does not recognize the user name or password.	The configured user does not have permission to write to Historian.
Connection succeeds.	Connected to the Historian server.

NOTE: when the Test button is clicked, global parameters *(page 420)* related to Historian Server information (e.g. HISTDATASERVER *(page 467)*, HISTDATAUSER *(page 467)*, HISTALMSERVER *(page 466)*) are created with appropriate values.

g. (Optional) Click Migrate Data to open the Historian Migration Utility *(page 296)* and migrate data in SQL databases to Historian. The Historian Migration Utility is also available through the CIMPLICITY Database Logger.

For the Historian Data Server only, select or clear the Overwrite check box to do the following.

- Select the check box to overwrite tag descriptions that already exist in Historian.
- Clear the check box to keep the tag descriptions that already exist in Historian.

IMPORTANT: You can revise *(page 146)* tag definitions, e.g. data type, in Historian. However, if Overwrite is checked, the changes will be overwritten when the Historian log is updated. However, collection *(page 149)* criteria are not overwritten.

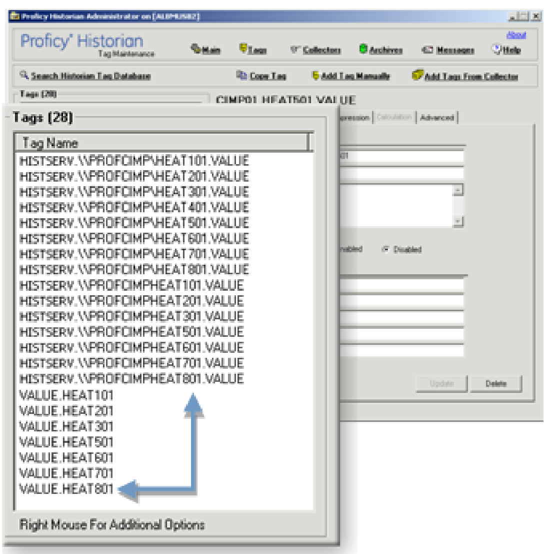
Tag Naming Convention

An example of the current Historian tag name displays the **Tag Name** field. This naming convention can be changed.

Note: If tags have previously been imported into Historian, changing the tag naming convention will result in duplicate tags, tags with the old naming convention and tags with the new naming convention.

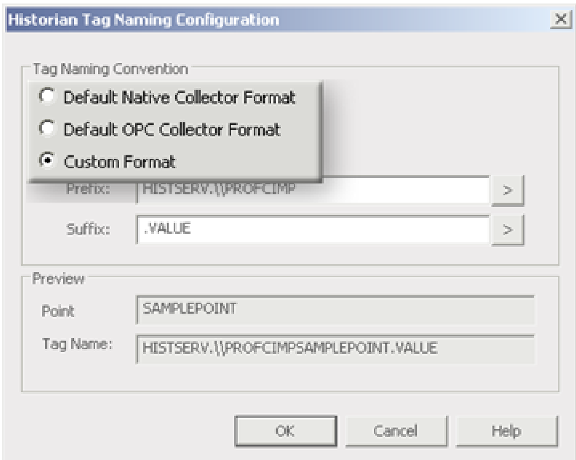
Example

The Historian tag name convention was changed for CIMPLICITY points that had been previously imported into Historian. The same CIMPLICITY points are listed as two separate tags in Historian.



Click the Open button to the right of the **Tag Name** field to change the displayed naming convention.

The Historian Tag Naming Configuration dialog box opens and displays the following options:



Default Native Collector Format

The default Native Collector format is available to help Historian users who used the Native Collector, which is not supported by CIMPLICITY v7.5 and higher.

When Default Native Collector Format is checked, Historian will use the Native Collector naming convention, which was

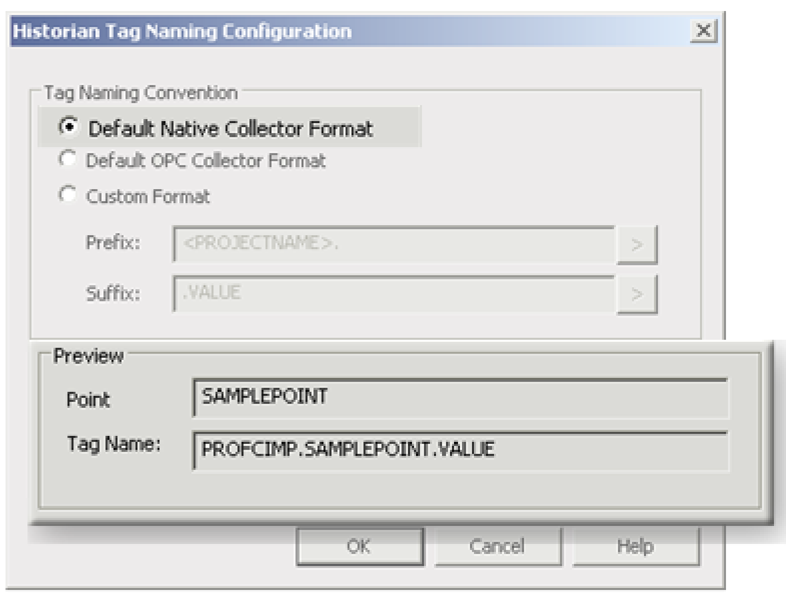
<PROJECTNAME>.<POINTNAME>.VALUE

Where

<PROJECTNAME> is the name of the CIMPLICITY project that the point is in.

<POINTNAME> is the CIMPLICITY point name.

A read-only naming convention preview displays for a sample point in the Tag Name field.



Default OPC Collector Format

When Default OPC Collector Format is checked, Historian will use the OPC Collector format as the naming convention for migrating CIMPLICITY points.

The OPC Collector format is:

<MACHINENAME>.\\<PROJECT NAME>\<POINTNAME>.VALUE

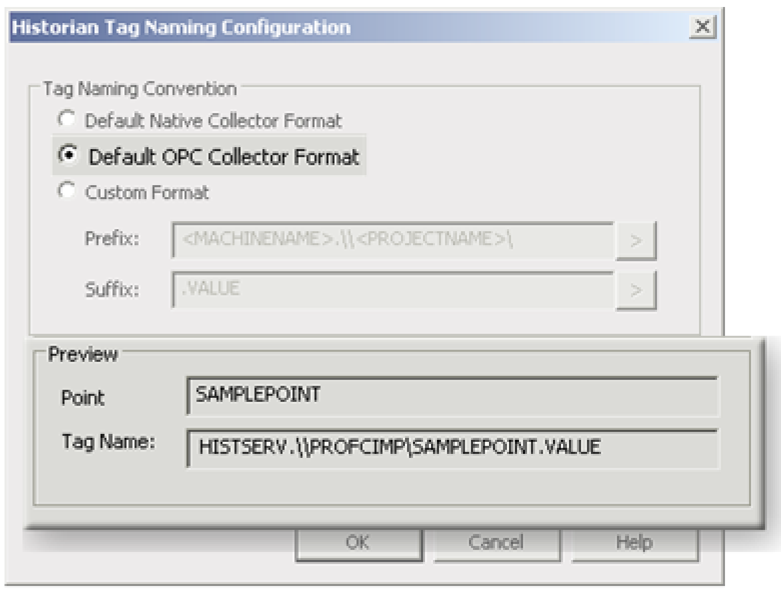
Where

<MACHINENAME> is the Historian server.

<PROJECTNAME> is the name of the CIMPLICITY project that the point is in.

<POINTNAME> is the CIMPLICITY point name.

A read-only naming convention preview displays for a sample point in the Tag Name field.

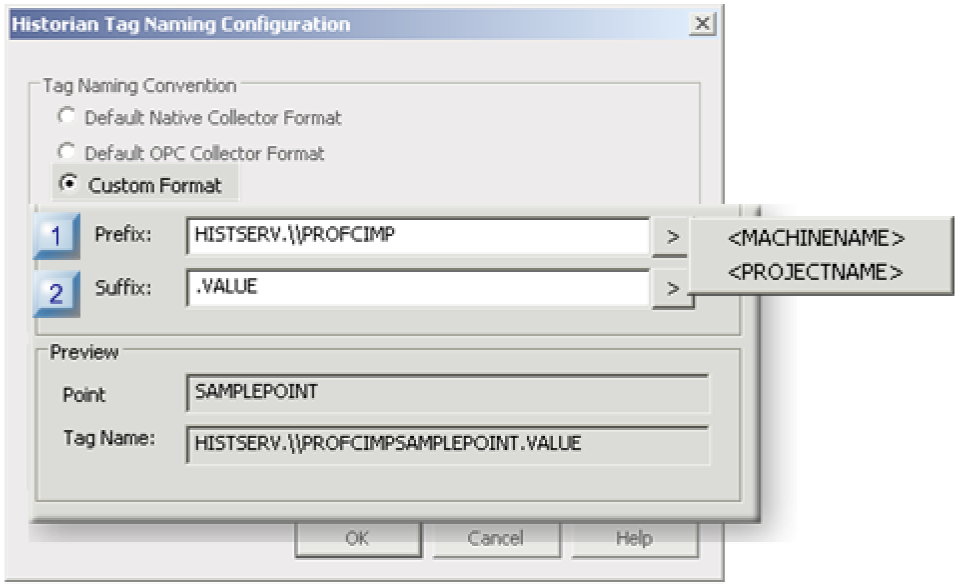


Custom Format

(Default)

When Custom Format is checked the entire naming convention can be customized and applied as the official Historian tag naming convention.

The convention is divided into a prefix and suffix, as follows.



Prefix

The default Prefix text for a Historian tag name is <MACHINENAME>.\\<PROJECTNAME>

Where

<MACHINENAME> is the Historian server.

<PROJECTNAME> is the name of the CIMPLICITY project that the point is in.

Guidelines

- Text can be entered instead of or in addition to the <MACHINENAME> and <PROJECTNAME> parameters.

Note: Click the Popup Menu button to the right of the Prefix field to select and automatically enter either parameter.

- The following should not be included in the prefix.

*

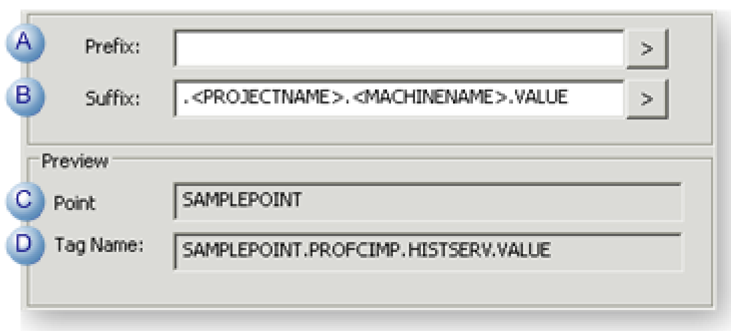
?

Spaces

- If the text in the Prefix field is empty then the prefix for the tag name will be the CIMPLICITY point name only.

Example

The following example describes how a Historian tag name is constructed when the Prefix field is blank.



A	Prefix	Blank.
B	Suffix	.<PROJECTNAME>.<MACHINENAME>.VALUE Important: Include the . character where it should be included in the Historian tag name.
C	Point	SAMPLEPOINT
D	Tag Name	SAMPLEPOINT.PROFCIMP.HISTSERV.VALUE Where the sample tag name parts are as follows. SAMPLEPOINT PROFCIMP HISTSERV VALUE

Suffix

The default Suffix text for a Historian tag name is .VALUE.

Guidelines

- The parameters <MACHINENAME> and <PROJECTNAME> can be used in the suffix instead of or in addition to being used in the prefix.

Note: Click the Popup Menu button to the right of the Prefix field to select and automatically enter either parameter.

- The following should not be included in the suffix.

*

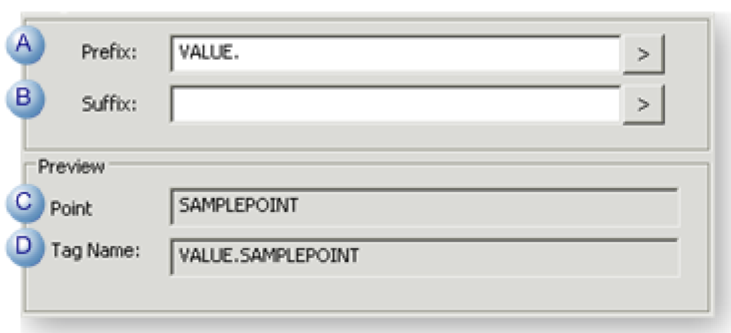
?

Spaces

- If the text in the Suffix field is empty then the suffix for the tag name will be the CIMPLICITY point name only.

Example

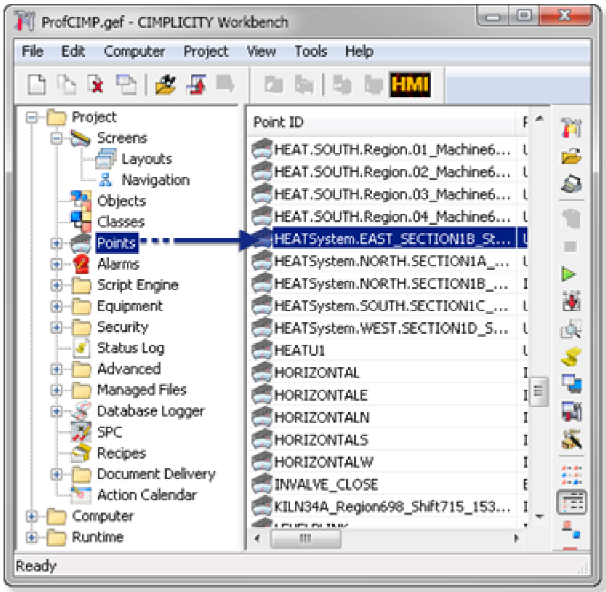
The following example describes how a Historian tag name is constructed when the Suffix field is blank.



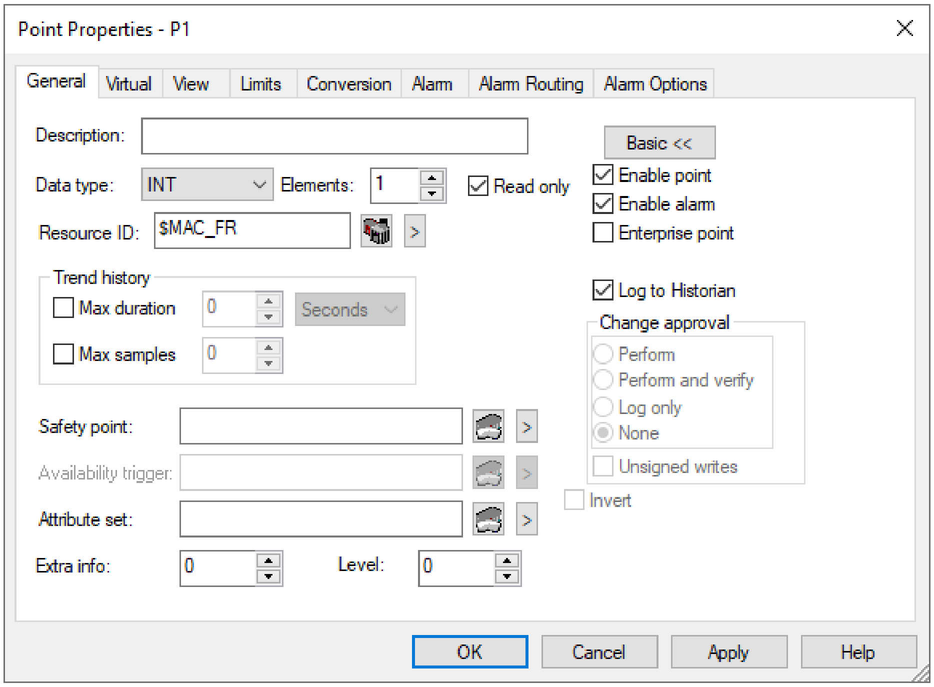
A	Prefix	VALUE. Important: Include the . character where it should be included in the Historian tag name.
B	Suffix	Blank.
C	Point	SAMPLEPOINT
D	Tag Name	VALUE.SAMPLEPOINT Where the sample tag name parts are as follows. VALUE SAMPLEPOINT

Step 3. Select Points to be Logged to Historian

1. (If the project is running) make sure dynamic configuration is enabled.
2. Select Points in the Workbench left pane.
3. Select a point that will be logged in the Workbench right pane.



4. Openthe point's Point Properties dialog box.
5. Select the General tab.
6. Check Log data.

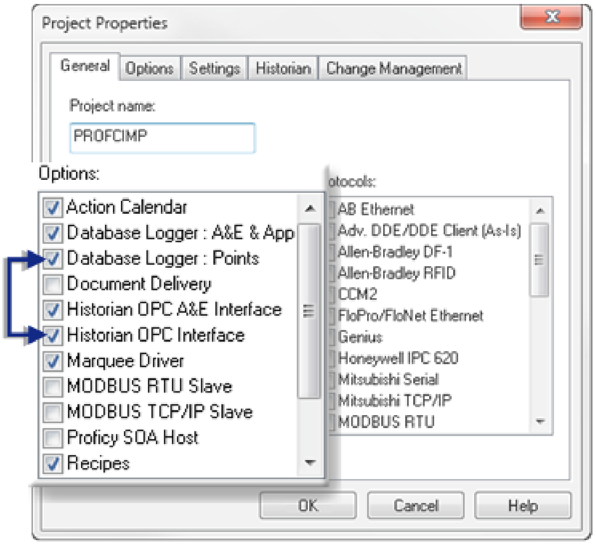


7. Click OK.

8. (If the project is not running) do the following:
    a. Do a Configuration Update.
    b. Startthe project.

The selected point and/or point alarm will be logged to Historian, based on the Historian options (page 132) that were selected.

Important: If Database Logger: Points is checked on the General tab in the Project Properties dialog box, point data will be logged to both the Historian and the Database Logger.



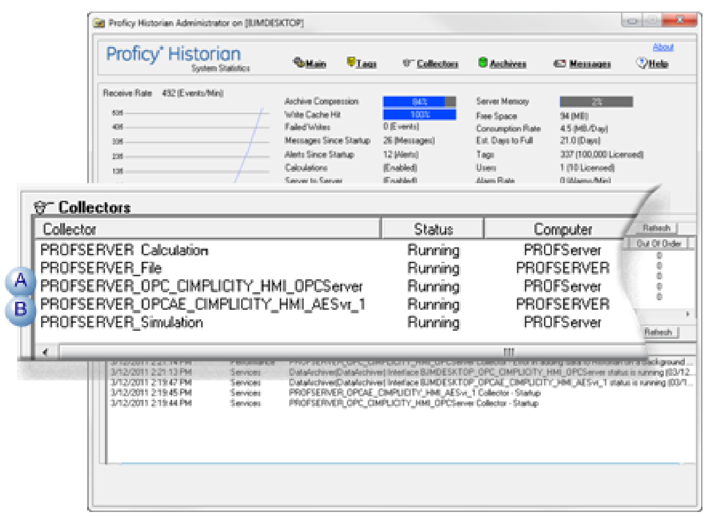
Step 4. Open the Historian Administrator System Statistics Window

1. Click Start on the Windows task bar.
2. Select (All) Programs>Proficy Historian>Historian Administrator.

   Note: The location on your start menu may differ from this path.

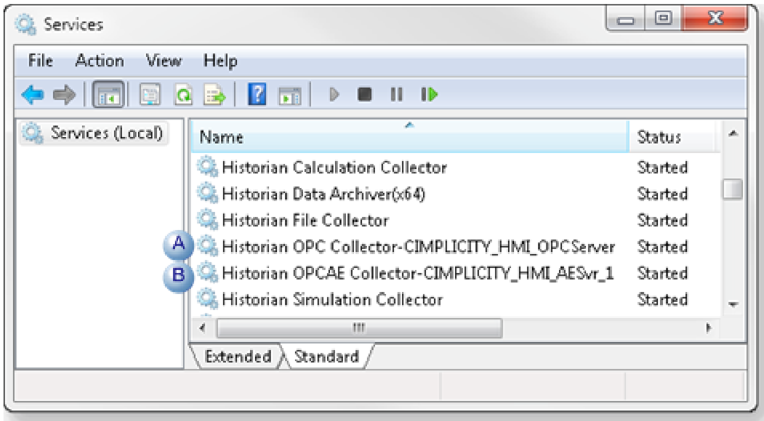
   The Historian Administrator System Statistics window opens.

   - The **Collectors** box reports if the selected Historian OPC collectors are running.
   - The Historian OPC Collectors are as follows.



A	Historian data server	<Historian OPC Collector Server Name>_OPC_CIMPLICITY_HMI_OPCServer
B	Historian alarm server	<Historian OPC Collector Server Name>_OPCAE_CIMPLICITY_HMI_AESvr_1

Note: The services status can be seen in the Microsoft Windows Services window.



A	Historian OPC Collector-_OPC_CIMPLICITY_HMI_OPCServer
B	Historian OPC Collector_OPCAE_CIMPLICITY_HMI_AESvr_1

Step 5. Review CIMPLICITY Point (Tag) Details in Historian

Step 5. Review CIMPLICITY Point (Tag) Details in Historian

Note:

- CIMPLICITY points are referred to as tags in Historian.
- The initial value for tags (points) that are unsolicited is collected twice.

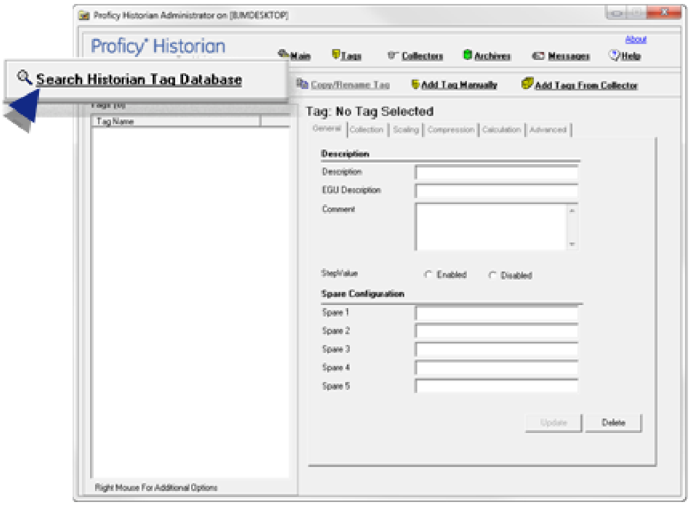
Step 5.1 *(page 145)*	List CIMPLICITY tags in Historian
Step 5.2 *(page 146)*	Display CIMPLICITY tag details that can be revised.

Step 5.1. Search the Historian Tag Database for CIMPLICITY Tags

1. Click Tags on the Historian menu bar.

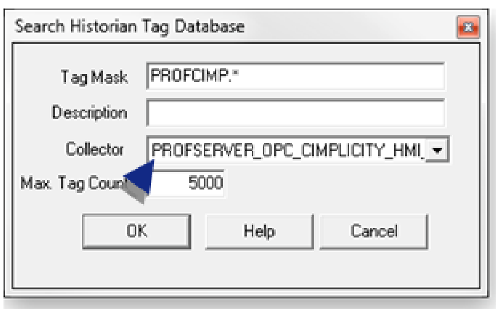
   The Tags Maintenance window opens.

2. Click **Search Historian Tag Database**.



A Search Historian Tag Database dialog box opens.

3. Select <Server Name>_OPC_CIMPLICITY_HMI_OPCServer in the **Collector** field.

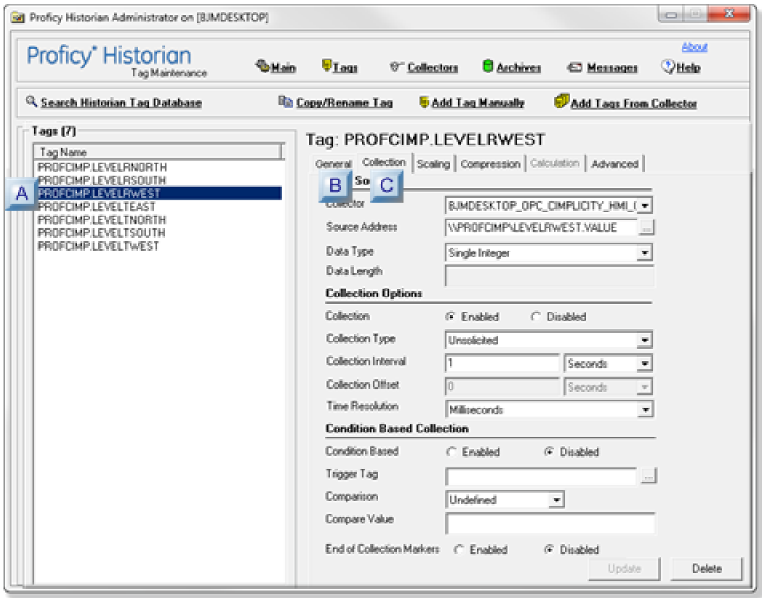


Note: Consult Historian documentation for details about the fields in the Search Historian Tag Database dialog box.

The CIMPLICITY tags that fulfill the criteria are listed in the Tags box.

Step 5.2. Display CIMPLICITY Tag Details that can be Revised

The following tag configuration was entered for the CIMPLICITY point entered by Historian migration to accommodate differences in data definitions between CIMPLICITY and Historian. The details can be changed in Historian. However, if Overwrite is checked on the Historian tab in the CIMPLICITY Project Properties dialog box, when the OPC Server detects these tags as new (e.g. CIMPLICITY and Historian are stopped and started) and changes will be overwritten with the CIMPLICITY data.



rect 229, 102, 251, 124 (page 148)
rect 196, 102, 218, 124 (page 148)
rect 0, 108, 22, 130 (page 147)

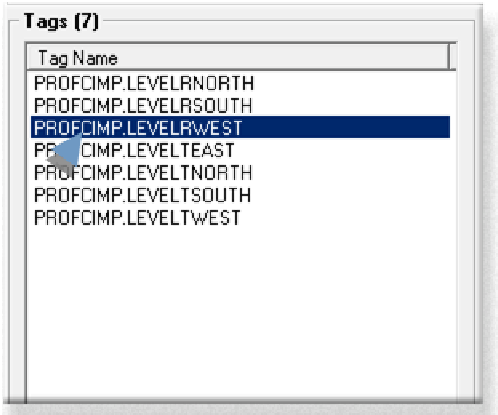
A (page 147)	Tag list
B (page 148)	General tab
C (page 148)	Collection tab

A	Tag list

- Tags that fulfill search criteria are listed in the Tag Maintenance window **Tags** box.
- Tag names for tags that came from CIMPLICITY points display as follows.

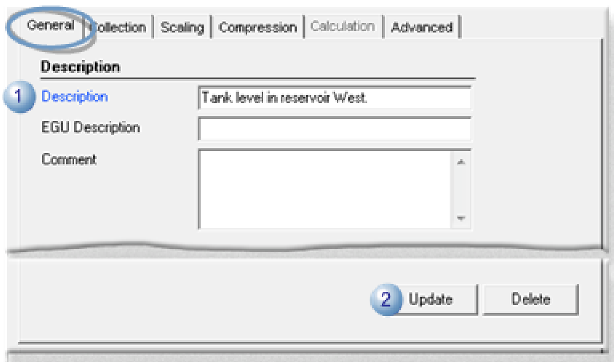
<OPC Server name >\\<CIMPLICITY project name>\<Point name>.<POINT VALUE>

- Select a tag to display its details on the Tag Maintenance window tabs.



B	General tab

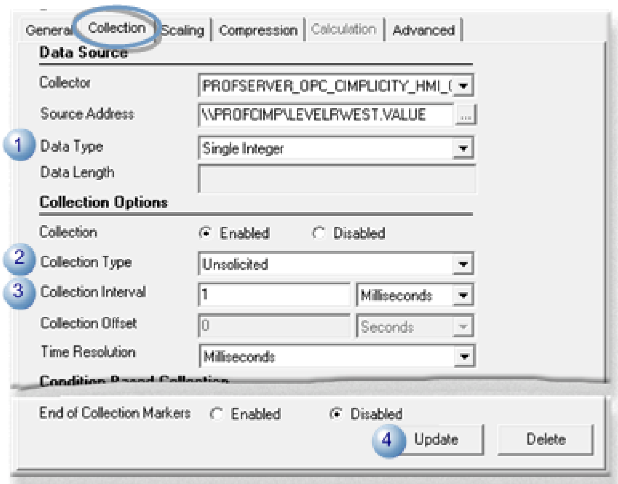
The following field on the General tab can be revised in Historian



	Field	Default Entry/Action
1	Description	Entry in the **Description** field in the CIMPLICITY Point Properties dialog box.
2	Update	Updates tag details in Historian when clicked.

C	Collection tab

The following fields on the Collection tab can be revised in Historian.



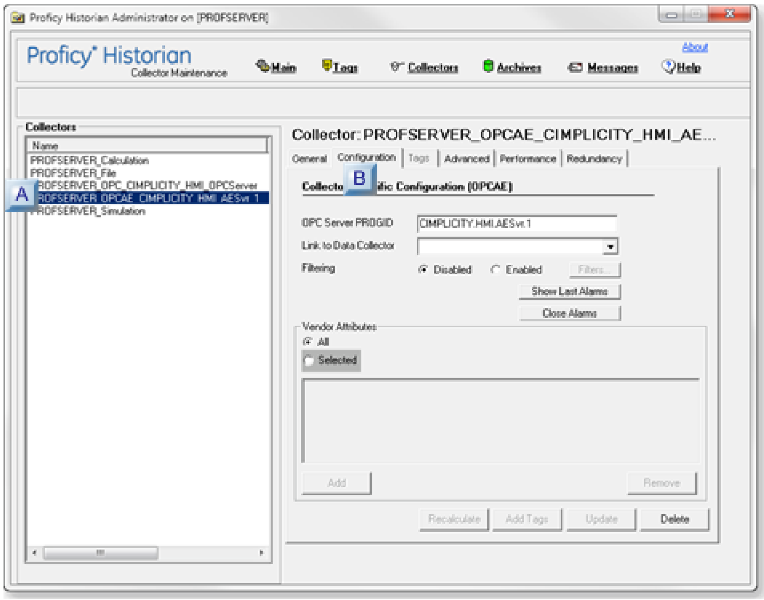
	Field	Default Entry/Action
1	Data Type	Historian tag data type that provides a long enough field to accommodate its corresponding *(page 159)* CIMPLICITY point data type.
2	Collection Type	Options are: • Unsolicited. • Polled.
3	Collection Interval	Options for an entered number are: • Milliseconds • Seconds • Minutes • Hours.
4	Update	Updates tag details in Historian when clicked.

Note: The Historian OPC data collector uses the Historian Collector default settings as the collection defaults. You can change the settings for a selected tag. If you change the Historian Collector default settings, the new defaults will apply to new tags. They will not overwrite your customized settings for selected tags.

Step 6. Display CIMPLICITY Alarm Data in Historian

Click Collectors on the Historian menu bar.

The Collector Maintenance window displays.



rect 0, 106, 23, 128 [(page 150)](#)
rect 207, 91, 237, 123 [(page 151)](#)

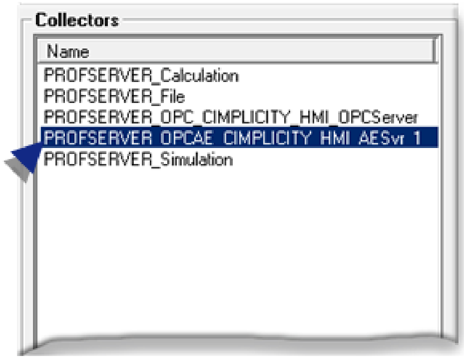
1 (page 150)	Collector list
2 (page 151)	Configuration tab

1	Collector list

- Collectors that are available for Historian are listed in the Collector Maintenance window **Collectors** box.
- Collector names display as follows.

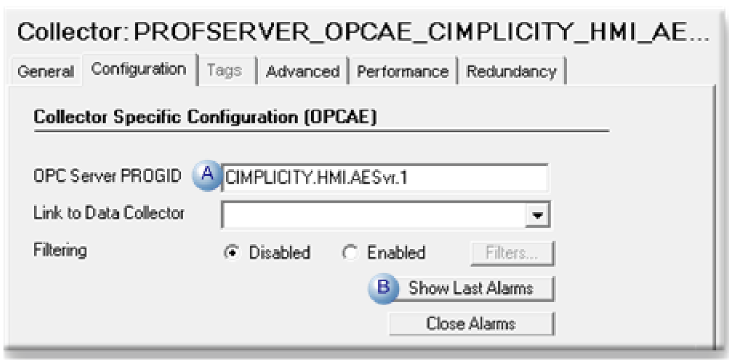
<Alarm & Event OPC Server name >\\<Collector name>

- Select the <server name>_OPCAE_CIMPLICITY_HMI_AESvr_1.



2	Configuration tab

The last 10 CIMPLICITY alarms and events can be displayed through the Configuration tab.



	Feature	Description
A	OPC Server PROGID	The CIMPLICITY.HMI.AE.Svr.1 was installed *(page 130)* with Historian.
B	Show Last Alarms	Opens a Show Alarms/Events window when clicked.

A Show Alarms/Events window lists details about the last 10 CIMPLICITY alarms and events.

Note: Review Historian documentation for more details about the Historian Collectors Maintenance Configuration tab.

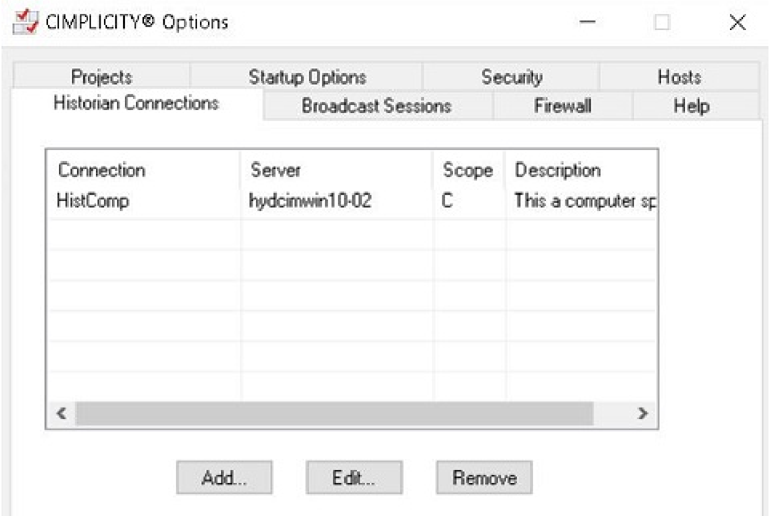
Step 7. Set up Historian Connections to Collect Data

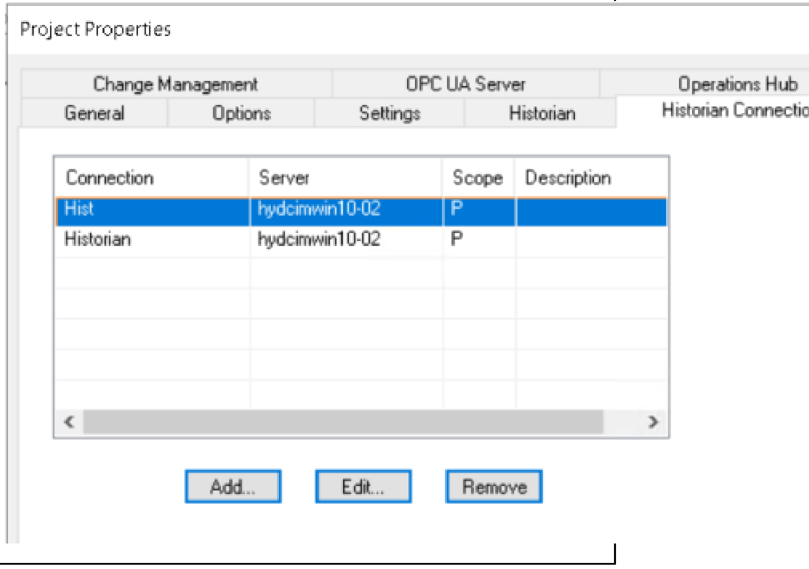
Historian connections can pull selected data from Historian and display it in CIMPLICITY applications.

Valid Historian connections that are listed at the following locations:

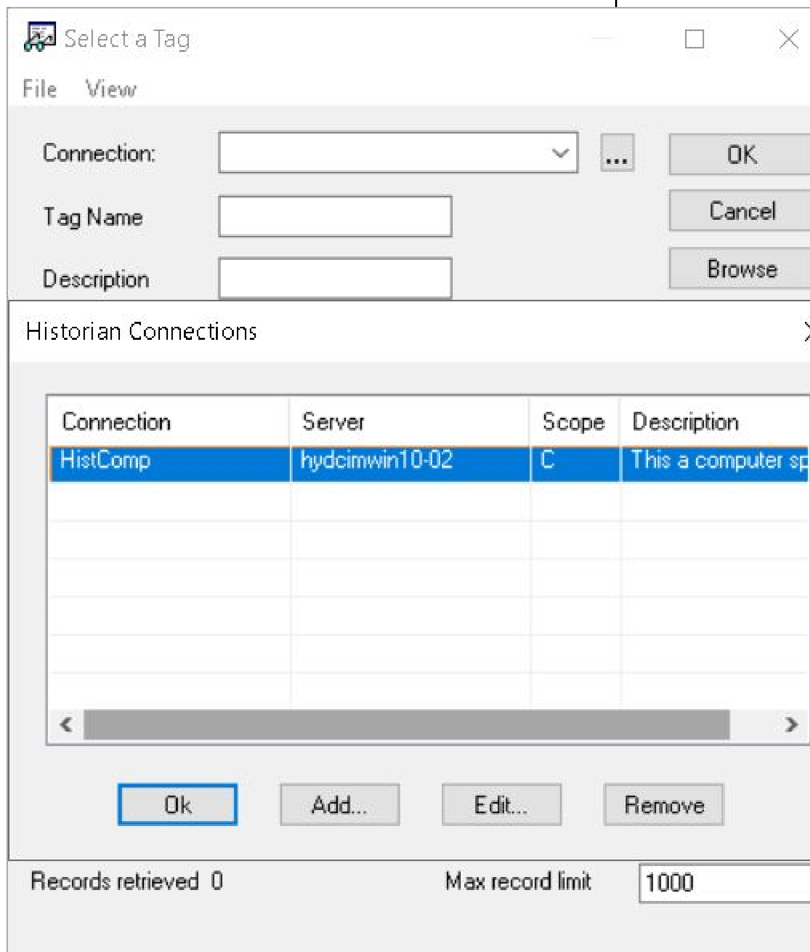
- **Historian Connections** tab in CIMPLICITY Options Dialog. (Only computer level connections are available)

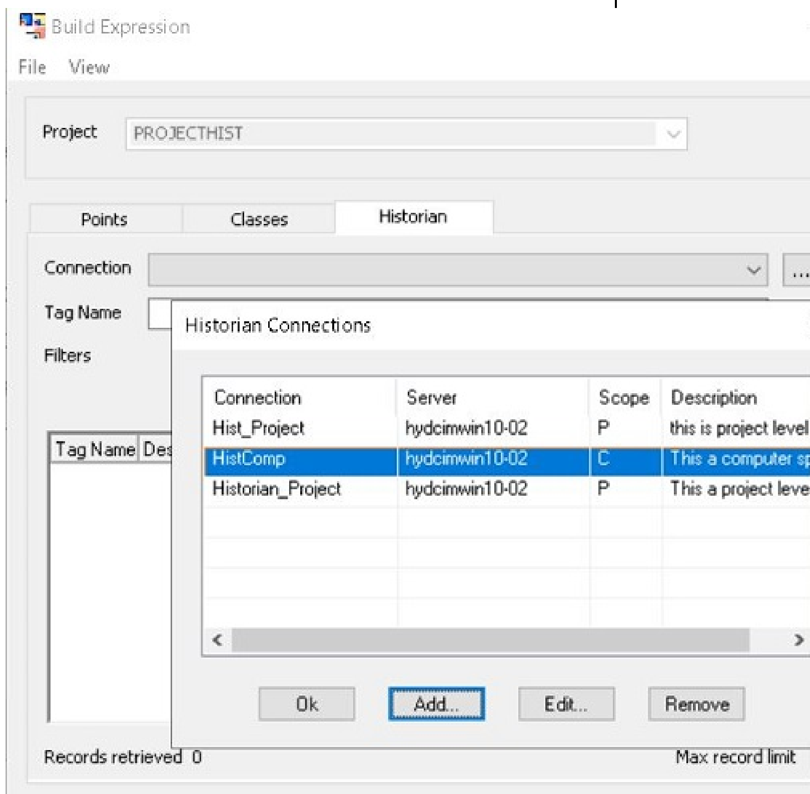
- **Historian Connections** tab of **Project Properties**. (Only project level connections are available).
- Tag browser that enables you to select and use Historian tags in CIMPLICITY, e.g. an **Expression** field in CimEdit. (Only computer level connections are available).
- Expression Builder of a derived point. (Both project level and computer level connections are available).

<table>
<tr><th>Location</th><th>Figure</th></tr>
<tr><td>Historian Connections tab in CIMPLICITY Options Dialog.</td><td>


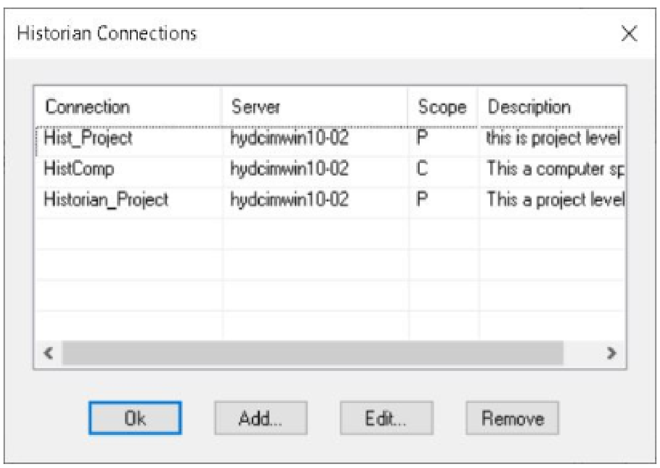
</td></tr>
<tr><td>Historian Connections tab of Project Properties.</td><td>


</td></tr>
</table>

Location	Figure
Tag browser	

Location	Figure
Expression Builder of a derived point.	

1. Select **Add** or **Edit** or **Remove** to add, edit and/or remove Historian connections.



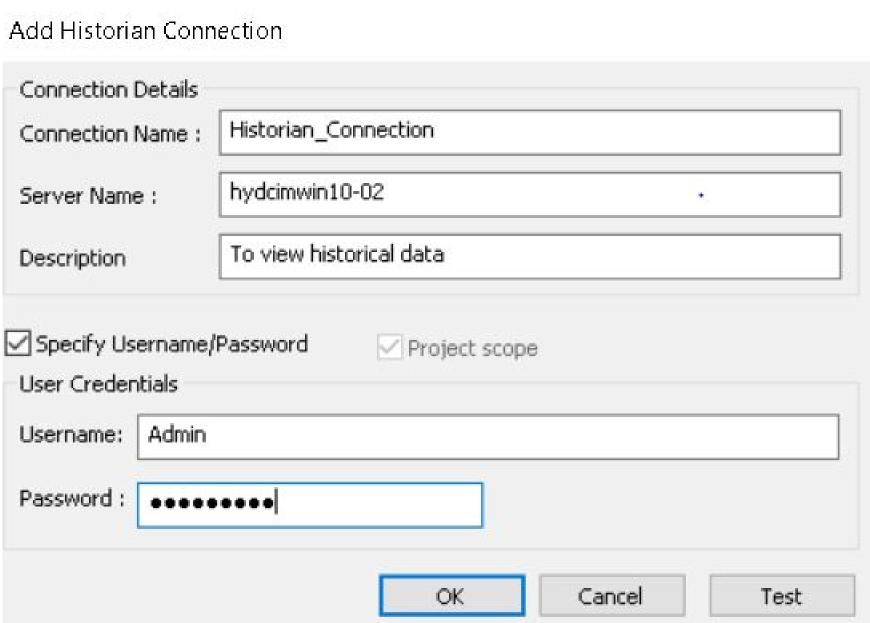
A (page 155)	Add button

B (page 156)	Edit button
C (page 156)	Remove button

Add, Edit or remove connections, as follows.

A	Add button

a. The **Add** button enables you to add a server/connection to the Historian Connections list. During runtime, CIMPLICITY can pull data from Historian tags in applications and fields that use and display Historian tag values.
b. Click **Add**. A blank Add Historian Connection dialog box opens.
c. Entries to define the Historian connections are as follows.



Field	Description
Connection Name	An alias that will make the connection easy to recognize.
Server Name	Name of Historian server.
Description	Additional detail to help identify the Historian connection.
Specify Username/ Password	Check to enable the Username and Password fields.
Username	Username that has access to the Historian Administrator.

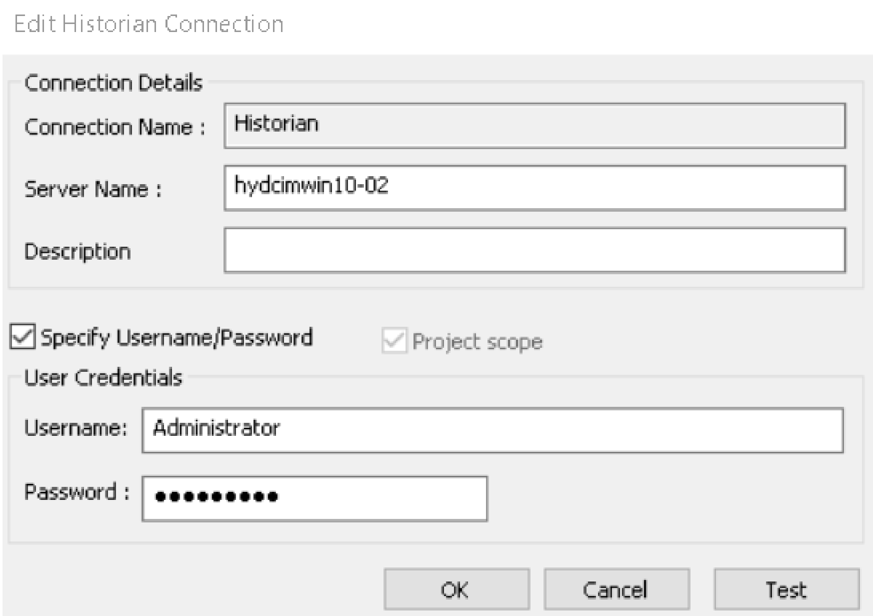
Password	Valid password for the entered user. **Note:** A valid password is required to connect if a user name is entered.	
Buttons	OK	Closes the dialog box; adds the connection/server to the list.
	Cancel	Cancels the addition.
	Test	Tests the connection to the Historian server.
Project Scope	Indicates that the Historian Connection is created at project level. **Note:** The **Project Scope** check box can be enabled/disabled only from the Expression Builder. You must disable the check box to add a computer scope connection.	

B	Edit button

Specifications for any connection can be edited.

a. Select the connection in the list to be edited; click **Edit**.

The **Edit Historian Connection** dialog box opens.



a. Make any required change.
b. Click **OK**.

The Edit Historian Connection dialog box closes; the edited connection/server replaces the original connection/server in the Historian Connections list.

C	Remove button

Any connection/server can be removed from the Historian Connection list.

1. Select a connection
2. Click **Remove**.

The connection is removed from the list; this connection will no longer be listed or be available when a user selects the connection/server that will supply Historian tag data for a feature, e.g. Historian Trend line.

Connection Guidelines

- The default Historian server is selected as follows.

Machine	Default Server is selected:
Server	Historian Administrator.
Viewer	During the Historian Client installation. **Note:** Historian Client can be installed during installation of the CIMPLICITY viewer.

- The Historian connection file:
- Is named histmgr_connection_config.xml.
- Is located in the ...\CIMPLICITY\data directory.
- Is stored at the computer level.
- Must be identical on the server and the viewer.

Tip: histmgr_connection_config.xml can be included in CIMPLICITY deployment *(page 356)* to insure that it is identical on the server and viewer.

Technical Reference: Historian Integration

Technical Reference: Historian Integration

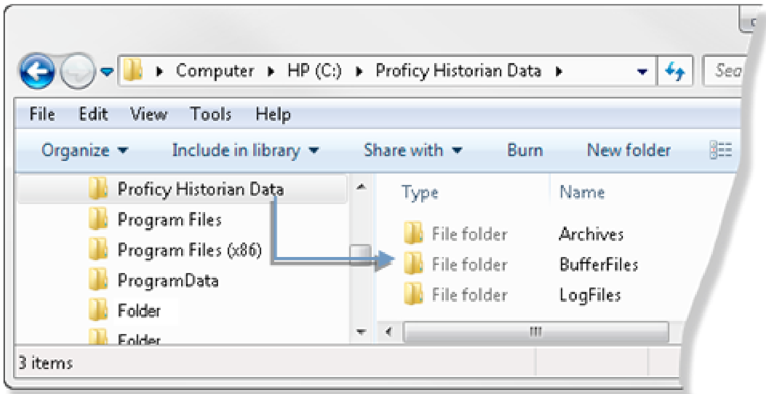
- CIMPLICITY data files' location in Historian.
- CIMPLICITY data types vs. Historian data types.
- CIMPLICITY tags selected for collection.

CIMPLICITY Data Files' Location in Historian

- Historian data location.
- Historian activity logs.
- Historian archives.

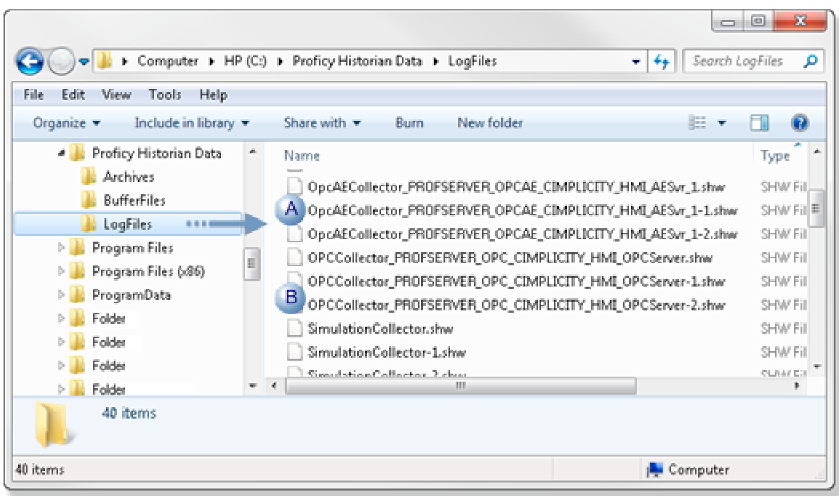
Historian Data Location

Historian data is located in the c:\Proficy Historian Data folder.



Historian activity logs

The Historian logs activity for the `ComputerName_OPC_` connection in the Historian>LogFiles.



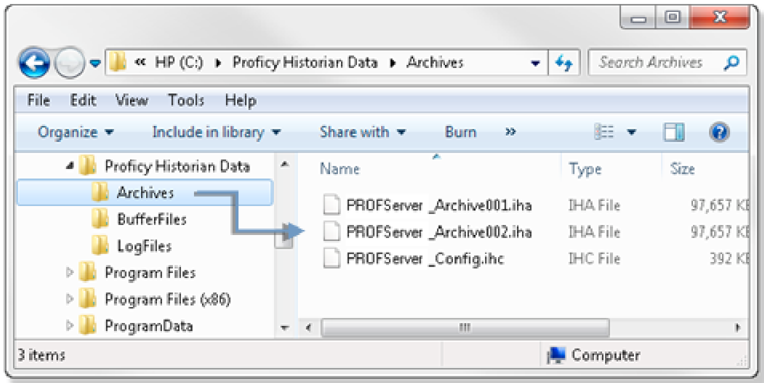
A	OpcAECollector_<server name>_OPCAE_CIMPLICITY_HMI_AESvr_<n>.shw
B	OPCCollector_<server name>_OPC_CIMPLICITY_HMI_OPCServer-<n>.shw

Historian archives

1. Open Microsoft Explorer.
2. Select C:\Program Files\Proficy\Historian\Archives.

   Note: This is the default path for Historian archives. The path to Historian in your system may be different.

When CIMPLICITY points are selected for data collection, Historian stores the data in Historian>Archives.



Note: Consult your Historian documentation for details about reviewing the data.

CIMPLICITY Data Types vs. Historian Data Types

CIMPLICITY ensures that the data type in Historian is long enough to accommodate the CIMPLICITY data type length.

Data type mapping is as follows.

Data Type	CIMPLICITY	Historian
Integer	DINT	Double integer
	INT	Double integer
	REAL	Double float
	SINT	Double integer
	UDINT	Double float
	UINT	Double integer
	USINT	Double integer
Boolean	BOOL	Integer
	BYTE	Double integer
	DWORD	Double integer
	WORD	Double integer
Text	STRING	Variable string
	STRING_20	Variable string
	STRING_8	Variable string

	STRING_80	Variable string

CAUTION: You can change the data type in Historian. However, if you checked Overwrite on the Historian tab in the CIMPLICITY Project Properties dialog box,

CIMPLICITY Tags Selected for Collection

CIMPLICITY Tags Selected for Collection

Historian provides options for selecting CIMPLICITY tags for collection.

Options include:

	Single CIMPLICITY tag for collection.
	Several CIMPLICITY tags for collection.

Single CIMPLICITY Tag for Collection

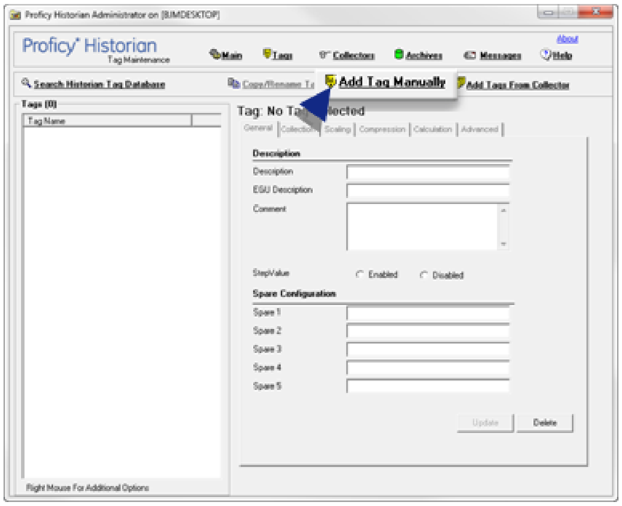
Tags that have not already been added to the Historian Collector can be added manually.

Important: One project has to be running for the Collector to be available; then all the projects connected to the OPC_Collector will be listed even if they are not running.

1 *(page 160)*	Open the Add Tag Manually dialog box.
2 *(page 161)*	Add a tag manually.

Open the Add Tag Manually dialog box

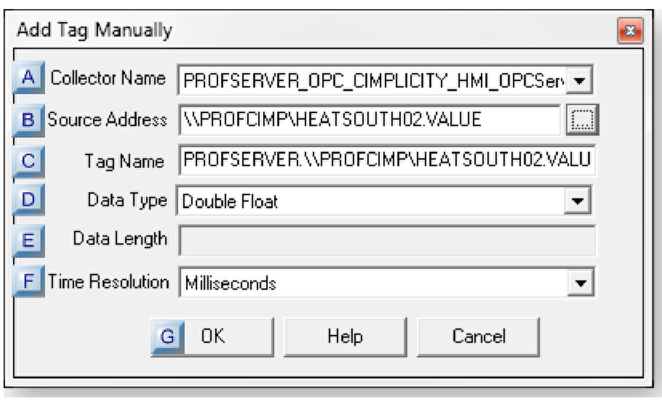
Click Add Tag Manually in the Historian Tag Maintenance window.



Result: The Add Tag Manually dialog box opens.

Add a tag manually.

Enter the following specifications.



rect 88, 193, 118, 216 [(page 164)]
rect 4, 159, 27, 182 [(page 164)]
rect 4, 131, 27, 154 [(page 163)]
rect 4, 105, 27, 128 [(page 163)]
rect 4, 82, 27, 105 [(page 163)]
rect 4, 56, 27, 79 [(page 162)]
rect 4, 30, 27, 53 [(page 162)]

A (page 162)	Collector Name.
B (page 162)	Source Address.

C *(page 163)*	Tag Name.
D *(page 163)*	Data type.
E *(page 163)*	Data Length.
F *(page 164)*	Time Resolution.
G *(page 164)*	OK.

A	Collector Name

Select <Server Name>_OPC_CIMPLICITY_HMI_OPCServer.

Example

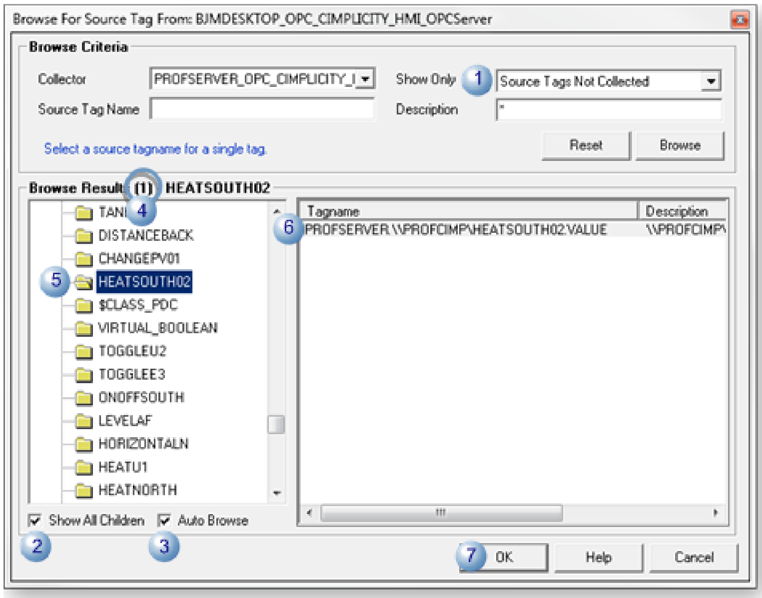
A project named PROFCIMPj has the Historian OPC Interface enabled.

The project's collector displays in Historian as `PROFSERVER_OPC_CIMPLICITY_HMI_OPCServer`.

B	Source Address

The Browse button to the right of Source Address opens the Browse for Source Tag window.

Do the following.



1	Show Only	Select Source Tags Not Collected.
2	Show All Children	(Optional) Check to show all tags listed under a selected folder, including points in subfolders.
3	Auto Browse	Check to display tagnames in the right-pane that are associated with the selected folder in the left-pane.
4	Browse Results	Displays the number of tags associated with a selected folder. **Note:** If Source Tags not Collected are the only tags that should display, the browse result will be 0 if all tags for a selected folder are already being collected.
5	Tag Folder	Selected folder displays its tags in the Browse for Source Tag dialog box right-pane.
6	Tagname	Select the tag that should be added to Historian collection. **Note:** Listed tagname(s) are associated with the listed folder.
7	OK	Click OK to close the Browse for Source Tag dialog box.

Result: The selected tag displays in the Add Tag Manually dialog box.

C	Tag Name

Tag selected in Browse for Source Tag dialog boxdisplays.

D	Data Type

Automatically filled in.

E	Data Length

Automatically filled in.

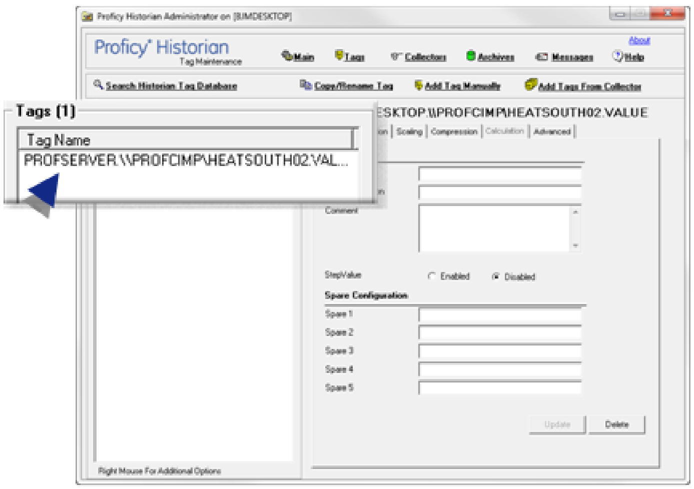
F	Time Resolution

Automatically filled in.

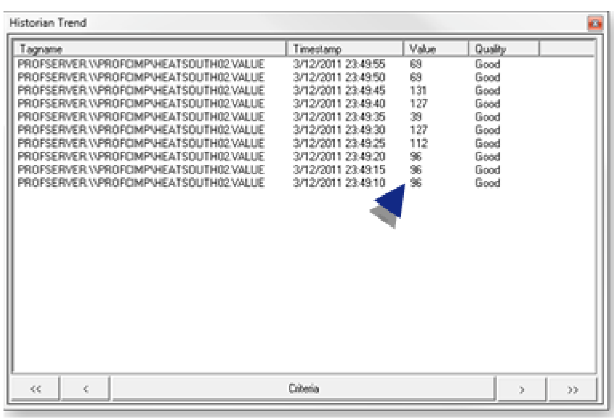
G	OK

Click OK to close the Add Tag Manually dialog box.

Result: The selected unique tag is added to the Porficy Historian Administrator Tags box.



Historian will now collect the tag's values from CIMPLICITY.



Note: Consult the Historian documentation for more information about configuring the Historian.

Several CIMPLICITY Tags for Collection

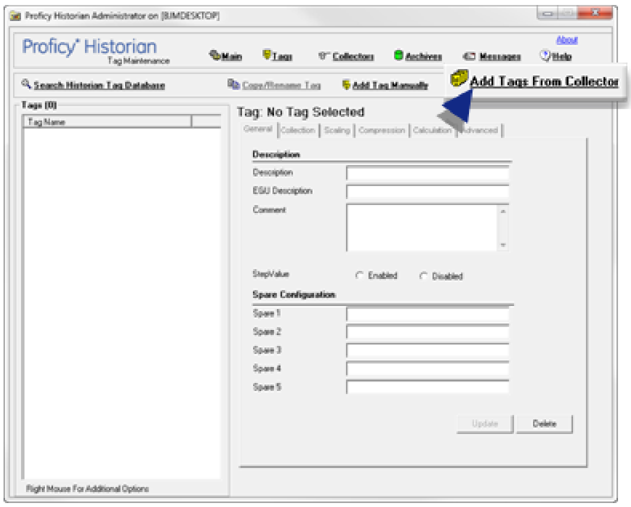
Tags that have not already been added to the Historian Collector can be added from the Collector.

Important: One project has to be running for the Collector to be available; then all the projects connected to the OPC_Collector will be listed even if they are not running.

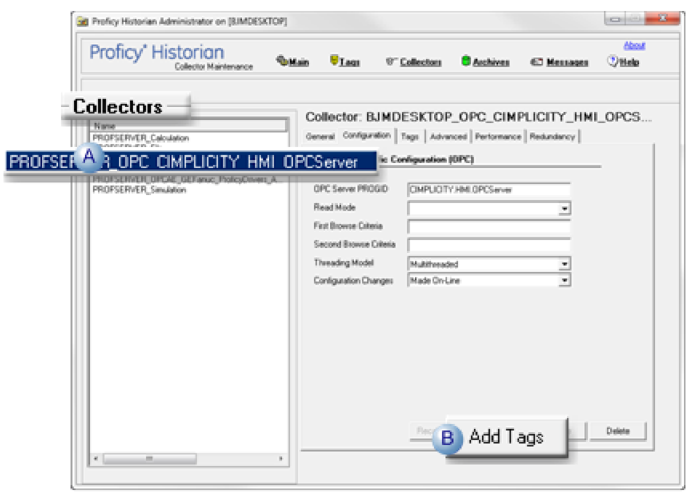
1. Open the Add Multiple Tags from Collector dialog box.

Do one of the following.

- Click Add Tag From Collector in the Historian Tag Maintenance window.



- Click Collectors in the Historian Administrator; do the following.

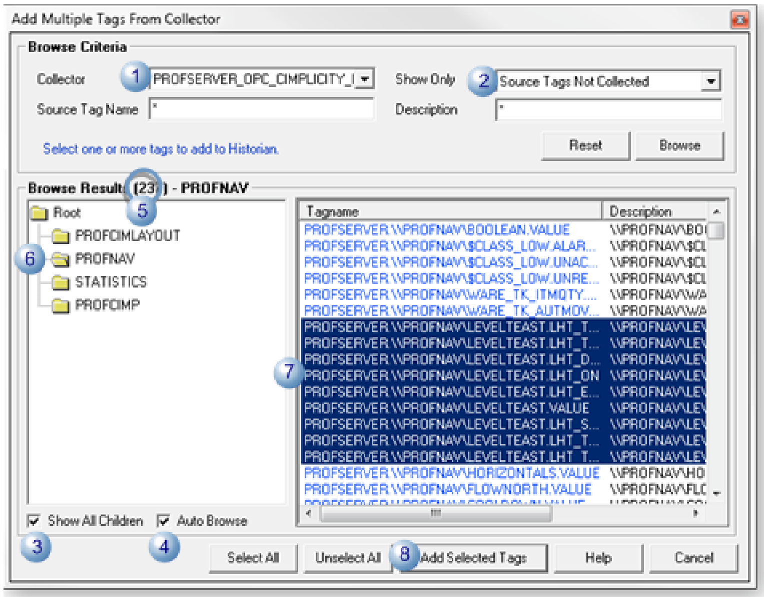


A	Select the collector that is collecting the required tag values.
B	Click Add Tags.

Result: The Add Multiple Tags from Collector window opens.

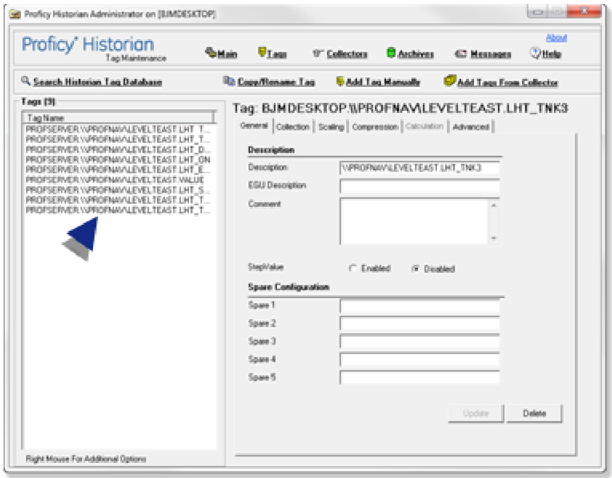
1. Select tags from the collector.

Select the following.



1	Collector	(Based on how the Add Multiple Tags from Collector dialog box was opened) select a collector if the correct collector is not already selected.
2	Show Only	Select Source Tags Not Collected.
3	Show All Children	(Optional) Check to show all tags listed under a selected folder, including points in subfolders.
4	Auto Browse	Check to display tagnames in the right-pane that are associated with the selected folder in the left-pane.
5	Browse Results	Displays the number of tags associated with a selected folder. **Note:** If Source Tags not Collected are the only tags that should display, the browse result will be 0 if all tags for a selected folder are already being collected.
6	Tag Folder	Selected folder displays its tags in the Browse for Source Tag dialog box right-pane.
7	Tagname	Select the tag that should be added to Historian collection. **Note:** Listed tagname(s) are associated with the listed folder.
8	OK	Click OK to close the Add Multiple Tags From Collector dialog box.

Result: The Historian Tag Maintenance window displays the selected tags; Historian will now collect the selected tags' values from CIMPLICITY.



Result: The CIMPLICITY Historian OPC Interface enables collection of data for the selected points.

Note: Consult the Historian documentation for more information about configuring the Historian.

Database Logger Configuration

About the Database Logger

The CIMPLICITY Database Logger 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.

The configuration is straightforward. You do not need to know about SQL or other ODBC database internals to configure Database Logger logging tables.

Overview of the Database Logger Functionality

In the Database Logger you easily:

- Use existing tables or create new tables that will log selected items for any one of the processes.
- Enter specifications for how, when and to what ODBC data source you want to log data.

When you start the project in which you configured the Database Logger, the Database Logger creates tables you configured based on the attributes you specified.

- When a project starts, the Database Logger:
- Creates or repair any missing or damaged databases.
- Creates any missing tables.

- Creates any missing columns in the tables.

Creating a report is also straightforward. CIMPLICITY provides sample reports that you use in Excel. You can use these sample reports to quickly generate a report or as a basis to create new reports.

The Database Logger option uses the standard ODBC interface to log your production data.

Refer to the **Readme** file that accompanies this release for a list of supported database interfaces.

Database Logger Configuration Overview

Database Logger Configuration Overview

Database Logger configuration is straightforward. The following list provides a logical order for reviewing the details when you are learning how to configure one or more log tables.

Database Logger configuration includes:

1 *(page 169)*	Hardware and database logger performance.	
2 *(page 169)*	Start Database Logger configuration.	
3 *(page 172)*	Database Logger configuration hierarchy review.	
4 *(page 182)*	Review available Database Logger file management functions.	
5 *(page 196)*	Configure Database Logger defaults.	
6	Configure any or all of the following database logs:	
	A *(page 207)*	Points.
	B *(page 237)*	Group of points.
	C *(page 249)*	Alarms.
	D *(page 265)*	Event alarms.
	E *(page 276)*	Event Management.
	F *(page 295)*	External applications.

7 (page 296)	Historian migration.

Hardware and Database Logger Performance

On a 400 MHz Pentium II with 200 MB memory

SQL Server	For Database Logging Peaks at Around
On-node	265 writes per second (2 field writes)
Off-node	100 writes per second (2 field writes). However, off-node is dependent on your network configuration as well.
Table Type	Write Definition
DATA	One point being logged. If you have 20 points, you are doing 20 independent writes (when is configurable individually for each point). The size of each write (number of fields) depends on how many attributes you are logging. Fields include: • 1 for timestamp • 1 for the point ID • Anything else you are logging (e.g., value, engineering units). The more attributes you have, the slower the writes will be.
GROUP	Is based on your table's logging conditions. The number of points and the attributes being logged (configurable independently for each point) determine how big the writes will be (how many fields there will be). Fields include: • 1 for the timestamp • Anything else you are logging. Example If you are logging the value of two points, your group log table will have 3 fields, timestamp and two fields for the points' values.

Bulk Insertion Peak Performance

CIMPLICITY Database Logger using bulk insertion.	Avg. bytes/ write	Approx. Writes/ sec	Approx. Total throughput/ sec
On-node	28	900-1000	25-30Kb
Off-Node	28	700-800	20-25Kb

Important: (For SQL Server) insert triggers fire only if bulk insertion is disabled.

Start Database Logger Configuration

Start Database Logger Configuration

Step 1 (page 170)	Enable Database Logger options.
Step 2 (page 170)	Open the Database Logger window.

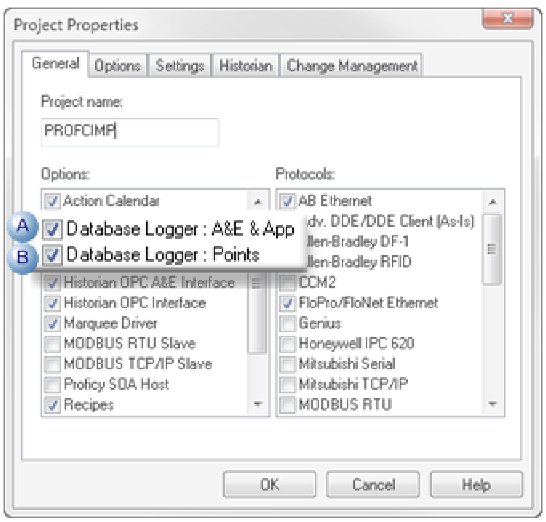
Step 1. Enable Database Logger Options

1. Do one of the following.
    - Click Project>Properties on the Workbench menu bar.
    - Click the Project Properties button  on the Workbench toolbar.

    The Project Properties dialog box opens.

2. Select the General tab.

3. Check the Database Logger options that your system will use.



	Option	Enables
A	Database Logger: A&E & App	ALARM_LOG EVENT_LOG EM_LOG Selected application tables
B	Database Logger: Points	POINT_LOG Existing GROUP_LOG tables

4. Click OK.

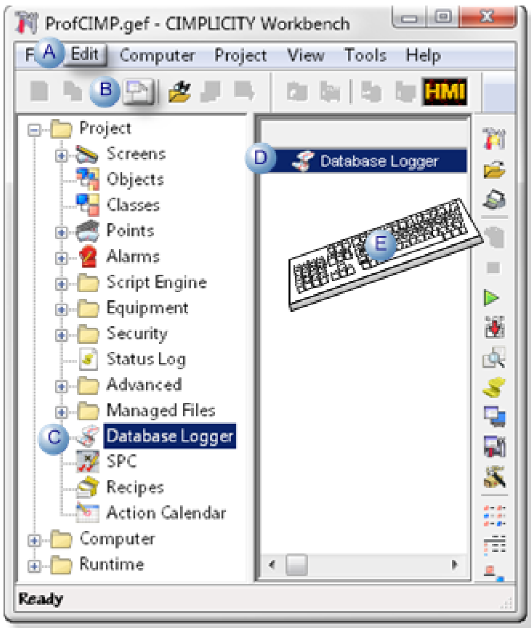
The logging tables for the selected options will be available for logging selected data.

Step 2. Open the Database Logger Window

1. Select **Project>Database Logger** in the Workbench left pane.

2. Select **Database Logger** in the 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:	
	Either	Or
	Double click **Database Logger**.	a. Right-click **Database Logger**. b. Select Properties on the Popup menu.
D	In the Workbench right pane:	
	Either	Or
	Double click **Database Logger**.	a. Right-click **Database Logger**. b. Select Properties on the Popup menu.
E	Press Alt+Enter on the keyboard.	

4. Right-click **Database Logger**.

5. Select Properties on the Popup menu.

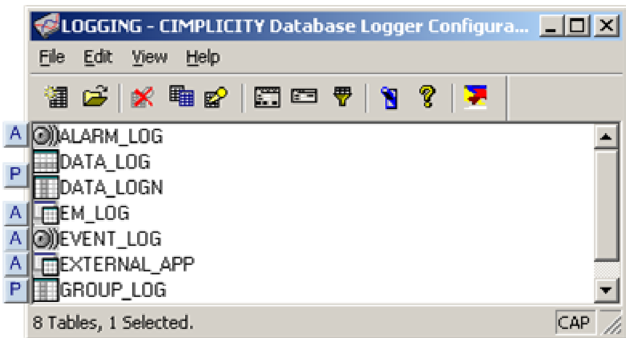
6. Right-click **Database Logger**.

7. Select Properties on the Popup menu.

Database Logger Configuration Hierarchy

Database Logger Configuration Hierarchy

When you open the Database Logger Configuration window, you see a list of the currently configured point data and group tables as well as the Alarm Log and Event Log tables.



rect -2, 64, 105, 91 *(page 172)*
rect -2, 120, 130, 138 *(page 172)*
rect -2, 136, 130, 154 *(page 172)*
rect -2, 152, 130, 170 *(page 172)*
rect -2, 168, 105, 186 *(page 173)*
rect -2, 89, 105, 120 *(page 172)*

The icon to the left of each table indicates its type as follows:

*	Icon	Table	Log
A	*(page 249)*	ALARM_LOG *(page 249)*	Selected alarms that appear in the Alarm Viewer.
P	*(page 207)*	Data *(page 207)*	Selected point values individually.
A	*(page 276)*	EM_LOG *(page 276)*	Event Manager activities.
A	*(page 265)*	EVENT_LOG *(page 265)*	Selected system alarms that do not appear in the Alarm Viewer.
A	*(page 295)*	External application *(page 295)*	Actions in certain CIMPLICITY applications, e.g., Tracker.

P	(page 237)	Group (page 237)	Selected point values in parallel.

* Indicates the option checked in the Project Properties dialog box, as follows.

A	Database Logger: A&E & App
P	Database Logger: Points

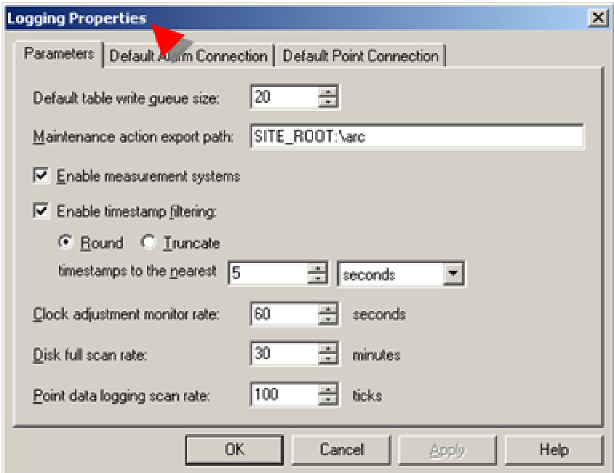
In order to configure how, when and where logs will collect and report data you can start at the Database Logger level to specify certain properties and become more specific in your configuration as follows:

Level 1 (page 173)	Configure Database Logger defaults that apply to all tables in the Database Logger.
Level 2 (page 174)	Configure Table defaults that apply to a selected table in the Database Logger. Override selected Database Logger defaults, if necessary.
Level 3 (page 174)	Configure Item logging conditions for selected items. These conditions override a table default, if necessary.

When you complete your configuration to can easily create a report for reviewing logged data.

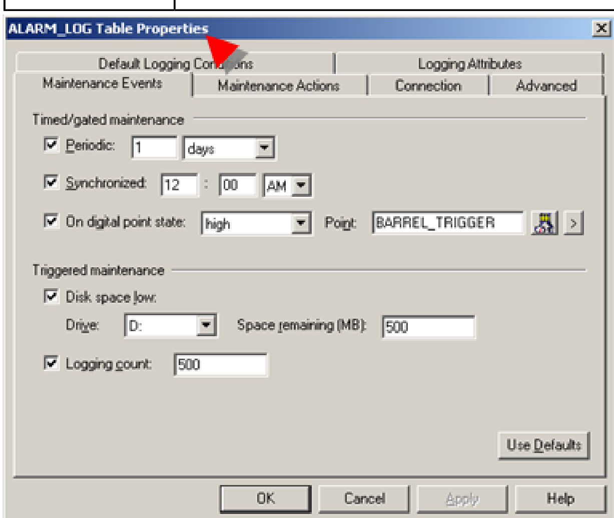
Level 1 Configuration–Database Logger Defaults

Configuration	Create Database Logger defaults for all the tables, including:		
	•	Several basic parameters.	
	•	Data logging sources for alarm and point logs. **Note:** The tabs in the dialog box display based on what options were check in the Project Properties dialog box.	
		Tab	Option Checked
		Default Alarm Connection	Database Logger: A&E & App
		Default Point Connection	Database Logger: Points
Configuration Tool	Logging Properties dialog box.		



Level 2 Configuration–Database Logger Table

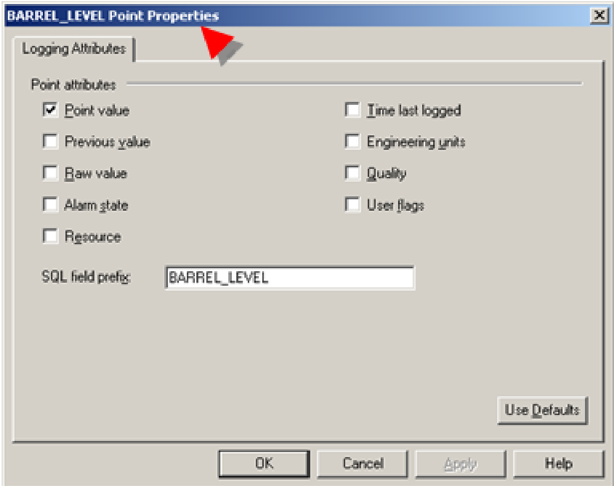
Configuration	Configure and create defaults for individual tables including: • Default logging conditions, • Logging attributes, • Connection, • Maintenance events, • Maintenance actions and • Advanced specifications. **Note:** The specific configuration depends on the process being configured. Table defaults can override Database Logger defaults.
Configuration Tool	Table Properties dialog box.



Level 3 Configuration–Item in a Database Logger Table

Configuration	Enter logging conditions that are different from the table defaults for individual items in the table.

Configuration Tool	Item's Properties dialog box.



Result: When configuration for any table is completed you can open Excel and generate reports from the samples that are included in CIMPLICITY or create your own reports.

Note: The Event Manager Log (EM_LOG) and logs for external applications require new reports.

Log Reports included with CIMPLICITY

Configuration	Create reports to display the data logged as a result of configuration in the Database Logger.	
Configuration Tool	Reports created on site or Excel reports supplied with CIMPLICITY that report:	
	Alarms	Alarms reported in the Alarm Viewer or other alarms.
	Data	Points logged individually.
	Trends	Points logged in parallel.

SQLDATA97.XLS

	A	B	C	D	E	F	G	H	I
1	timestamp	point_id	_ALRM	_ENG	_TIME	_RES	_PREV	_VAL	_RAW
32	17-Dec-1999 10:00:52	S90_900	Normal			90-30		4000	4000
33	17-Dec-1999 10:00:52	S90_350	Alarm High			90-30		14969	14969
34	17-Dec-1999 10:00:52	S90_550	Alarm High			90-30		2154	2154
35	17-Dec-1999 10:00:52	S90_500	Normal			90-30		220	220
36	17-Dec-1999 10:02:52	S90_350	Alarm High		12/17/1999 10:00:52	90-30	14369	14969	14969
37	17-Dec-1999 10:02:52	S90_500	Normal		12/17/1999 10:00:52	90-30	220	220	220
38	17-Dec-1999 10:02:52	S90_550	Alarm High		12/17/1999 10:00:52	90-30	2154	2154	2154
39	17-Dec-1999 10:02:52	S90_900	Normal		12/17/1999 10:00:52	90-30	4000	4000	4000
40	17-Dec-1999 10:03:52	CAR_MOVEMENT	Normal			WASH		0	0
41	17-Dec-1999 10:03:52	COLOR_CHANGE	Normal			DEMO		111	111
42	17-Dec-1999 10:04:52	S90_350	Alarm High		12/17/1999 10:02:52	90-30	14369	14969	14969

Control | **Data** | Error Log | Parameters | ParmDialog

Logging Table Columns Guidelines

guide: Guidelines for logging table columns include:

- A table can have, at most, 250 columns. The number of columns you can actually have in a group table depends on the type of data you are storing.
- A record being added to the database can have at most 2 KB of data.

With overhead, 2 KB amounts to approximately 222 8-byte floating-point numbers. (Points with Engineering Units conversion are stored in floating point format.)

If you are storing a number of floating-point numbers or long text strings in a group record, it is recommended that you verify that the 2-KB limit is not being exceeded.

Review Log Table Configuration

Review configuration for CIMPLICITY:

- Alarm Logging *(page 249)* (ALARM_LOG)
- Status Log Logging *(page 286)* (COR_LOG)
- Point Data Logging *(page 207)* (Data Log)
- Event Management Logging *(page 276)* (EM_LOG)
- Event Alarm Logging *(page 265)* (EVENT_LOG)
- Group Point Logging *(page 237)* (Group log)
- Application Logging *(page 295)* (An external application log)

Review Advanced Configuration

Review configuration for CIMPLICITY:

- ALARM-LOG. See Step 3.7. Do Advanced Alarm Logging Configuration *(page 263)*.
- COR_LOG. See Step 2.5. Perform advanced COR_LOG Configuration *(page 293)*.
- Data Log. See Step 2.7. Do advanced Logging Configuration *(page 234)*.
- EM_LOG. See Step 2.6. Do advanced EM_LOG Configuration *(page 285)*.
- EVENT_LOG. See Step 3.6. Advanced Event Logging Configuration *(page 274)*.
- Group Log. See Step 2.7. Do Advanced Group Log Configuration *(page 247)*.
- An external application log. See Application Logging *(page 295)*.

Automatic Report Configuration

Automatic Report Printing Configuration

Important: You can only use the automatic report printing capability with Microsoft Access (As-Is product) and SQL Server databases.

You can use OLE automation from CIMPLICITY scripts to trigger the running and printing of reports from Microsoft Excel spreadsheets that you have configured to produce reports

The CIMPLICITY script may be:

- Triggered by Database Logger Maintenance Events, or
- Triggered by any event in the Event Manager, or
- Attached to an object a user can trigger on a CimView screen.

The steps to implement automatic report printing include:

Step 1 *(page 177)*	Configure the CIMPLICITY Service to access a printer.
Step 2 *(page 179)*	Create a script to generate the report and print it.
Step 3 *(page 180)*	Create an event to trigger the script.

Step 1. Configure the CIMPLICITY Service to Access a Printer

1. Open the Windows Services window.
2. Double-click the CIMPLICITY HMI Service.

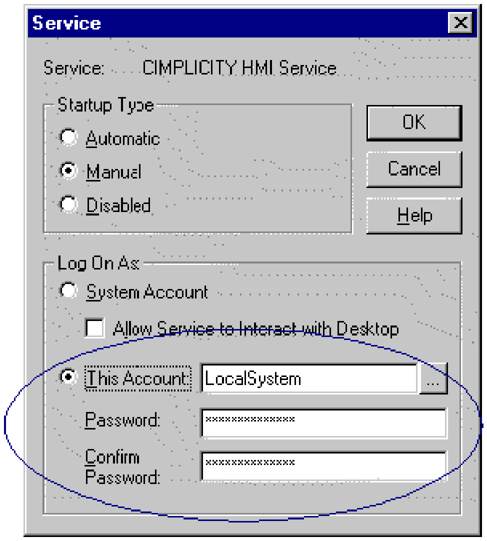
   The CIMPLICITY Service Properties dialog box opens.

3. Choose one of the following options to continue:

   For systems with single users:

   a. Click **This Account** in the Service dialog box, under **Log On As.**
   b. Select a CIMPLICITY user account.
   c. Enter the correct password and confirm it, if the account requires a password.

Service Dialog Box: One user example

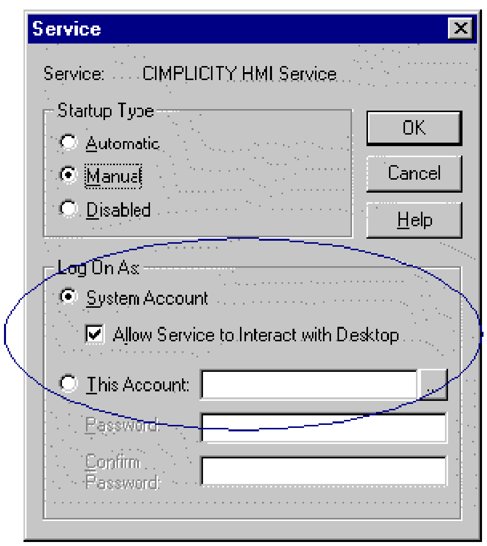


Note: If no one is logged in to the account you select here, when the script executes, no printers will be available. The report will be generated, but it will not be printed.

For systems with multiple users:

a. Click **System Account** in the Service dialog box, under **Log On As.**
b. Check the **Allow Service to Interact with Desktop** check box.

Service Dialog Box: One user example



4. Click OK to accept your changes and close the Service dialog box.

5. Click OK to close the Services dialog box.

6. Close the Control Panel.

Service access is configured according to your specifications.

Step 2. Create a Script to Generate the Report and Print it

1. Expand the Basic Control Engine folder in the CIMPLICITY Workbench left pane.
2. Double-click Scripts.

   The CIMPLICITY Program Editor opens.

3. Create the script to run the report and print it.

   You can use the sample script below as a template.

4. Compile the script and create an executable.

   Note: The following script opens the **SQLALARM.xls** spreadsheet, generates a report, and prints it. You can use it as a template for creating your own scripts:

```
Sub Main ()
'------------------------------------------------------------------------------------
'  This section sets REPORT_TRIGGER back to 0 so that the script
 doesn't continue to run.
'  This is not needed if the report is triggered directly from a
 CimView screen or by a
'  TIMED event in the Database Logger or Event Manager.
'
  Dim PT As New point
  PT.id = "\\PROJECT\REPORT_TRIGGER"
  PT.value = 0
  PT.set
'------------------------------------------------------------------------------------
'  The code from this point on is for printing.
'  This example uses one of the existing Excel spreadsheets for
 extracting and printing
'  the data from the cimplog.mdb alarm database.
'  You will need to customize this code so that it prints what you
 need. It is also
'  possible in some applications to run a specific macro from the
 command line.
'  You can then use the SHELL command.
'
  Dim Excel As Object
Set Excel = CreateObject ("Excel.Application")
'
'      Open the workbook for alarm reporting
  Excel.application.workbooks.open "C:\CIMPICITY\REPORT\SQLALARM.XLS"
  Excel.application.workbooks("SQLALARM.XLS").activate
'
```

```
'       Run the macro to get the data
  Excel.application.Run "SQLALARM.XLS!GenerateReport"
'
'       Print the data
  Excel.application.workbooks("SQLALARM.XLS").worksheets("Data").activate
  Excel.application.workbooks("SQLALARM.XLS").activesheet.printout
'
'       Quit without saving the data
  Excel.application.workbooks("SQLALARM.XLS").saved = TRUE
  Excel.application.quit
End Sub
```

Step 3. Create an Event to Trigger an Automatic Report

Step 3. Create an Event to Trigger an Automatic Report

There are many ways to trigger a script to run and print a report from Database Logger tables.

Following are two options.

Option 3.1 *(page 180)*	Use the Database Logger Maintenance Actions.
Option 3.2 *(page 181)*	Use the Event Manger.

Option 3.1. Use the Database Logger Maintenance Actions

1. Open *(page 170)* the Database Logger in the project's Workbench.
2. Open the Table Properties dialog box for the table whose report will be printed.
3. Select the Maintenance Events tab for the table.
4. Specify the event(s) you want to trigger the script.
5. Select the Maintenance Actions tab for the table.
6. Check **Run program**.
7. Enter the fill path and name of the executable you created from the script.
8. Click **OK.**

   The Database Logger saves your changes and closes the Table Properties dialog box.

The report automatically generates and prints whenever any of the maintenance events you specified occur.

Option 3.2. Use the Event Manager

- Trigger a report automatically from an event
- Allow users to trigger a report from a CimView screen

Trigger a report automatically from an event

1. Openthe **Event Editor** in the project's Workbench.
2. Create any event you wish to trigger the report.
3. Create an action for this event to run the script.

   Result: The report automatically generates and prints whenever the event occurs.

   Allow users to trigger a report from a CimView screen

4. Open a screen in CimEdit.
5. Select or create the object you want to trigger the report.
6. Display the object's property sheets.
7. On the Script tab, click **Edit**.
8. In the Edit Script window, enter a script that will trigger the report. You can generate the report directly, or trigger an event that invokes an action that generates the report.
9. Close the Edit Script window.
10. Display the Events tab.
11. Create a new event and assign it a new procedure.
12. In the Procedure Information dialog box, create an Invoke Script action and specify the script you created.
13. Click **OK** to save the new procedure information.
14. Click **OK** to save the object's properties.

When a user executes the event you configured, the report automatically generates and prints.

Database Logger File Management Functions

Database Logger File Management Functions

The Database Logger window provides you with the ability to carry out several file management functions to manage CIMPLICITY tables, including:

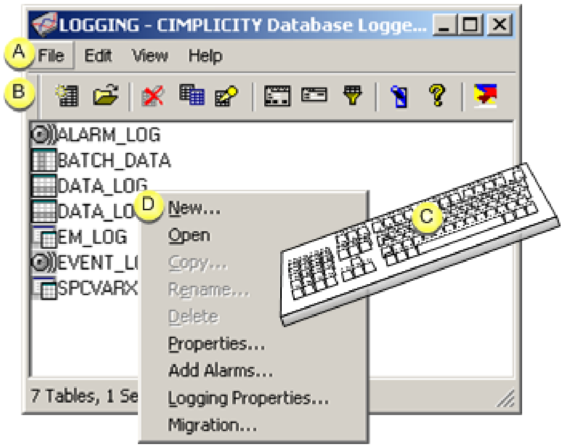
1 *(page 182)*	Create a new logging table.
2 *(page 184)*	Open an existing logging table.
3 *(page 185)*	Copy an existing logging table to a new logging table.
4 *(page 186)*	Rename a logging table.
5 *(page 187)*	Delete a logging table.
6 *(page 188)*	Filter table lists.
7 *(page 191)*	Activate dynamic configuration.
8 *(page 191)*	Reconcile a CIMPLICITY table with a logging database.
9 *(page 195)*	Close a logging table.
10 *(page 195)*	Exit the Database Logger Configuration window.

1. Create a new Logging Table

Note: The logging tables that are available and that can be created depend on the options you selected in the Project Properties *(page 170)* dialog box.

You can create as many new CIMPLICITY data, group or applications tables as you need.

1. Do one of the following:



A	Click File>New Table on the Database Logger Configuration window menu bar.
B	Click the **New Table** button on the Database Logger Configuration window toolbar.
C	Press Ctrl+N on the keyboard.
D	a. Right-click any icon in the Database Logger Configuration window's list. b. Select New… on the Popup menu.

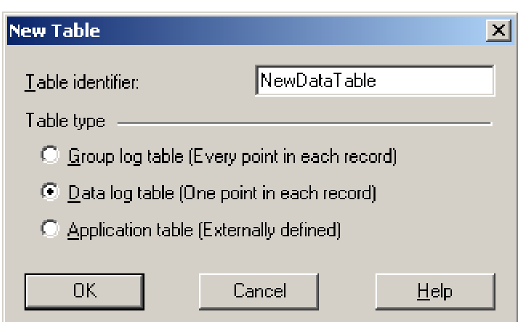
The New Table dialog box opens when you use any method.

2. Right-click any icon in the Database Logger Configuration window's list.

3. Select New… on the Popup menu.

4. Enter and select the following.

Note: Table types are enabled based on the Database Logger options selected in the Project Properties *(page 170)* dialog box.



Option	Description

Table identifier field	Name for the table The name: • Can contain • Alphanumeric characters (**A** through **Z**, **0** through **9**) and • Underscores (_). • Should be no longer than 16 characters. If you enter more than 16 characters, the name will automatically be truncated to the first 16 characters when you click **OK**. • Cannot be the name of a currently existing table. • Should not use any words that are reserved for the database management system being used. **Example** If you are using a SQL Server, do not use BULK, which is a reserved word for that DBMS.
Radio buttons	Check the type of table to create in the Table type box.

5. Click **OK**.

A Table Properties dialog box opens for you to configure the new table.

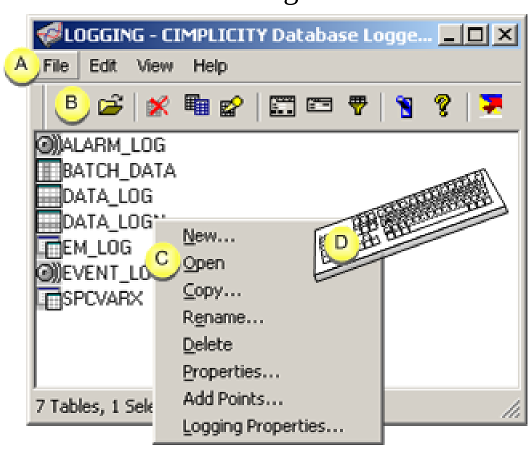
Important: You can only create new group, data and application log tables, when the appropriate Database Logger options are enabled in the Project Properties *(page 170)* dialog box.

When the Database Logger: A&E & App option is enabled, only one alarm table (ALARM_LOG), event table (EVENT_LOG) and Event Manager (EM_LOG) table exist in the Data Logger. You cannot delete these tables, and you cannot create additional Alarm, Event or Event Manager tables. When the option is not enabled, these tables are not available.

2. Open an Existing Logging Table

1. Select the logging table in the Database Logger Configuration window.

2. Do one of the following.



A	Click File>Open on the Database Logger Configuration window menu bar.
B	Click the **Open Table** button  on the Database Logger Configuration window toolbar.

C	a. Right-click the selected table. b. Select Open on the Popup menu.
D	Press Ctrl+O on the keyboard.

3. Right-click the selected table.

4. Select Open on the Popup menu.

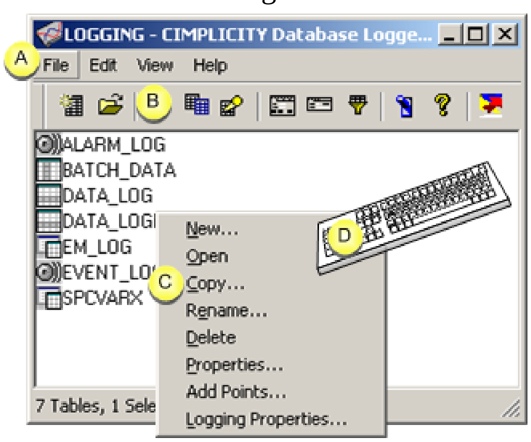
3. Copy an Existing Logging Table to a New Logging Table

The following log tables can/cannot be copied.

Can be copied	• CIMPLICITY data logs • CIMPLICITY group logs
Cannot be copied	• ALARM_LOG • EM_LOG • EVENT_LOG • External application log

1. Select the logging table you want to copy.

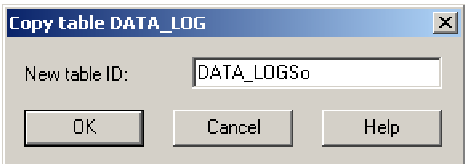
2. o one of the following.



A	Click File>Copy Table on the Database Logger Configuration window menu bar.
B	Click the **Copy Table** button  on the Database Logger Configuration window toolbar.
C	a. Right-click the selected table. b. Select Copy on the Popup menu.
D	Press Ctrl+C on the keyboard.

A Copy table dialog box opens.

3. Right-click the selected table.
4. Select Copy on the Popup menu.
5. Enter the name of the new table in the **New table ID** field.



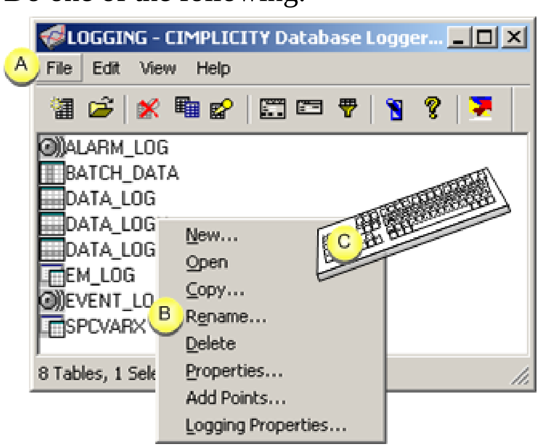
6. Click **OK**.

4. Rename a Logging Table

The following log tables can/cannot be renamed.

Can be renamed	• CIMPLICITY data logs • CIMPLICITY group logs
Cannot be renamed	• ALARM_LOG • EM_LOG • EVENT_LOG • External application log

1. Select the logging table you want to rename.
2. Do one of the following.



A	Click File>Rename Table on the Database Logger Configuration window menu bar.

B	a. Right-click the selected table. b. Select Rename on the Popup menu.
C	Press Ctrl+R on the keyboard.

A Rename table dialog box appears.

3. Right-click the selected table.
4. Select Rename on the Popup menu.
5. Enter the new name for the table in the **New table ID** field.

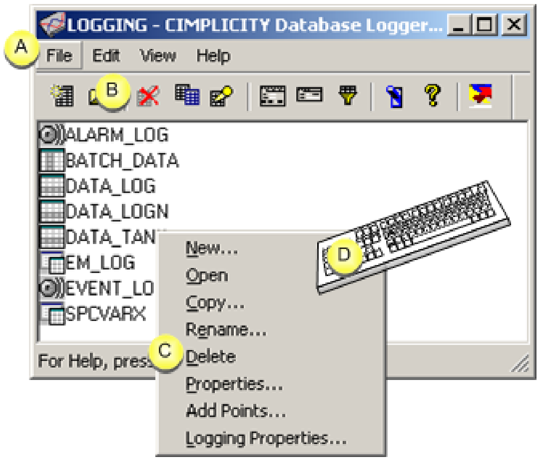
6. Click **OK**.

5. Delete a Logging Table

The following log tables can/cannot be deleted.

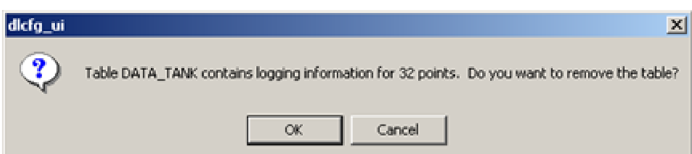
Can be deleted	• CIMPLICITY data logs • CIMPLICITY group logs • External application log
Cannot be deleted	• ALARM_LOG • EM_LOG • EVENT_LOG

1. Select the logging table you want to delete.
2. Do one of the following.



A	Click File>Delete Table on the Database Logger Configuration window menu bar.
B	Click the **DeleteTable** button on the Database Logger Configuration.
C	a. Right-click the selected table. b. Select Delete on the Popup menu.
D	Press Delete on the keyboard.

A message opens to inform you of the number of items in the table and confirm deletion.



3. Right-click the selected table.

4. Select Delete on the Popup menu.

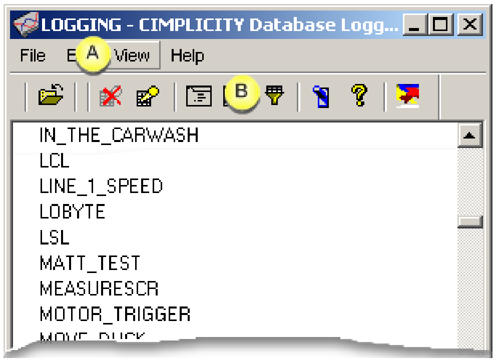
5. Click **OK**.

6. Filter Log Table Lists

- Define filter.
- Wildcard guidelines.

Define filter

1. Do one of the following in the Database Logger Configuration window or any open table.



A	Click View>Filter... on the Database Logger Configuration window menu bar.
B	Click the **Filter** button on the Database Logger Configuration window or a table's toolbar.

A Filter <Type> dialog box opens.

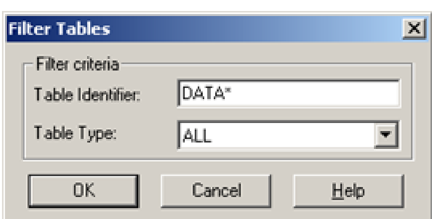
Where

<Type> depends on the table or window in which the filter operation is being used.

2. Enter filter specifications based in the associated dialog box as follows.
   - Database Logger Configuration window.

A Filter Tables dialog box opens.

Enter the following.

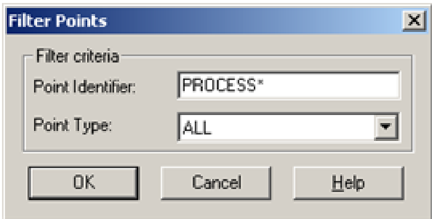


Field	Description
Table Identifier	Full or partial table ID. Use the ? and * wildcards to perform partial string searches.
Table Type	Options are: • ALARM • ALL • DATA • EXTERNAL • GROUP • Data log or group log table

- Data log or group log table.

A Filter Points dialog box opens.

Enter the following.

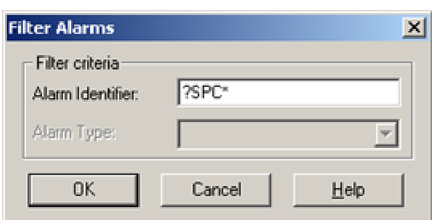


Field	Description
Point Identifier	Full or partial point ID. Use the ? and * wildcards to perform partial string searches.
Point Type	Options are: • 4STATE • ALL • ANALOG • APPL • DIGITAL • TEXT

- ALARM_LOG table and EVENT_LOG tables.

A Filter Alarms dialog box opens.

Enter the following.



Field	Description
Alarm Identifier	Full or partial alarm ID. Use the ? and * wildcards to perform partial string searches.
Alarm Type	Not available.

3. Do one of the following.
   a. Click **OK** to filter the list according to the information you've entered.
   b. Click **Cancel** to cancel the filter function.

Tip: To display all the items in the table, enter an asterisk (*****) in the <Type> **Identifier** field.

Wildcard guidelines

You can use the following wild cards in the Database Logger user interface:

wildcard	Description

*	Search for any number of characters at this point in the string.
?	Search for any single character in this place in the string.
_	If you are calling ODBC functions directly, you cannot use the ***** and **?** wild cards. The underscore character, _,is the only available wild card. Use it to search for any character in this place in a string.

Note:

- There are no implied wild cards.

If you do not include or terminate your search string with an asterisk, only those items that match your request exactly will be returned.

- If you are calling SQL functions directly, you can use the ***** and **?** wild cards.

7. Activate Dynamic Configuration

You can dynamically configure several logging properties through the Database Logger. You can:

- Add, remove or modify all point table configuration.
- Add and remove alarms from ALARM_LOG and EVENT_LOG tables.
- Modify logging conditions in ALARM_LOG and EVENT_LOG tables. That includes the default logging conditions in the table and the logging conditions of individual alarms.

To activate dynamic configuration in the Database Logger:

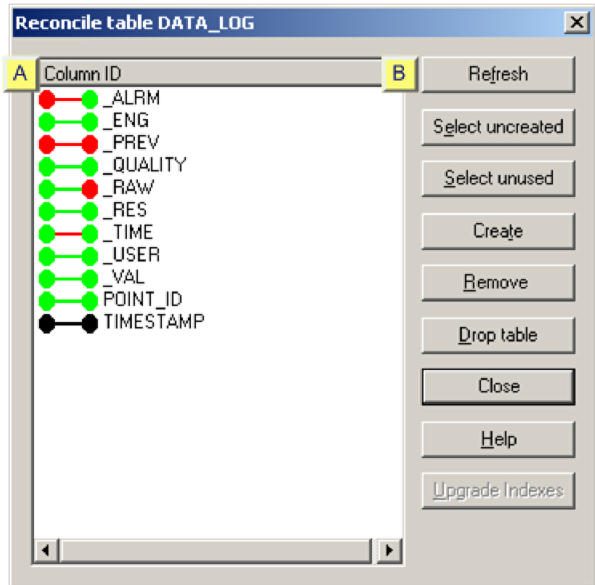
Click the **Dynamic** button  on any table's toolbar.

When you activate Dynamic Configuration, CIMPLICITY updates your project's configuration automatically. You don't have to return to the Workbench and do a project update in order for your changes to take affect.

8. Reconcile a CIMPLICITY table with a Logging Database

1. Open the CIMPLICITY table in the Database Logger that you want to reconcile.
2. Click Edit>Reconcile Table on the table's menu bar.

   The Reconcile table dialog box opens.



rect 265, 33, 368, 58 [(page 193)]
rect 265, 65, 368, 90 [(page 193)]
rect 262, 129, 365, 154 [(page 193)]
rect 265, 95, 368, 120 [(page 193)]
rect 265, 162, 368, 187 [(page 193)]
rect 265, 195, 368, 220 [(page 194)]
rect 265, 292, 368, 317 [(page 194)]
rect -3, 21, 150, 225 [(page 192)]
rect 233, 28, 271, 63 [(page 193)]

A (page 192)	Table reconcile status color code
B (page 193)	Reconcile buttons

A	Table Reconcile Status Color Code

The icon colors display the differences between the CIMPLICITY table (Database Logger configuration) and the logging database as follows.

	Field in the CIMPLICITY	Field in the	
Icon	Database Logger	Logging Database	Comment
	Yes	Yes	
	Yes	Yes	Types don't match.
	No	Yes	Unused
	Yes	No	Uncreated

	Removed	Removed	

B	Reconcile Buttons

Reconcile buttons are available to perform the following actions.

- Refresh the view.

Refresh	Refresh

Click Refresh.

- Create uncreated fields that are in the CIMPLICITY table but not in the logging database.

Select uncreated	Select uncreated
Create	Create

a. Select the fields that display the icon.

Tip: Do one of the following.

- Press the **Shift** or **Ctrl** key to select multiple fields
- Click **Select Uncreated** to select all the uncreated fields in the table.

a. Click **Create**.

The fields are created in the logging database. The fields' icons change to:

- Remove unused fields that are in not in the Database Logger, but are in the logging database.

Select unused	Select unused
Remove	Remove

For Microsoft Access (As-Is product)

a. Select the fields that display the icon.

Tip: Press the **Shift** or **Ctrl** key to select multiple fields, or click **SelectUnused** to select all the unused fields in the table.

a. Click **Remove**.

SQL Server and Oracle

SQL Server and Oracle do not directly support dropping columns in a database table.

a. Copy the table to a temporary table.
b. Drop the table from the database.
c. Recreate the table with the fields you wish to use.

d. Copy the data from the temporary table to the new table.
e. Delete the temporary table.
f. Consult your SQL Server or Oracle documentation for details on how to do this.

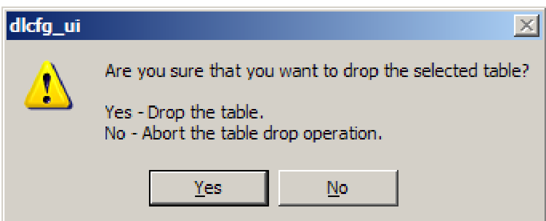
The unused fields are removed from the database. The fields' icons change to .

- Drop table

	Drop Table

1. Click **Drop Table** to drop the table from the database.

A message displays asking you to confirm dropping the table.



1. Click Yes to drop the table from the database.

Note: The table is not dropped from the CIMPLICITY Database Logger.

- Upgrade Indexes

	Upgrade Indexes

Note: Upgrade Indexes is enabled if you are looking at indexes that have not updated from a 5.0 or older log.

Important: Updating indexes to CIMPLICITY v5.5 may take a long time to complete. The amount of time required depends on the size of the table. This operation is very CPU intensive and will degrade the database server performance while it is executing.

1. Make sure your project is shut down.
2. Disconnect any other clients accessing the table whose indexes you will be upgrading.
3. Click **Upgrade Indexes**.

A message appears warning you about making sure that no one is logging to the table.

CIMPLICITY:

1. Drops all old indexes that it had created,
2. Creates new Version 5.5 indexes and
3. Disables the **Upgrade Indexes** button.

9. Close a CIMPLICITY Logging Table

9. Close a CIMPLICITY Logging Table

When a CIMPLICITY logging table is open, you can close the table.

Option 9.1 *(page 195)*	Return to the Database Logger Configuration window.
Option 9.2 *(page 195)*	Exit the Database Logger from an open table.

Option 9.1. Return to the Database Logger Configuration Window

1. Right-click on any item in the table.
2. Select Close on the Popup menu.

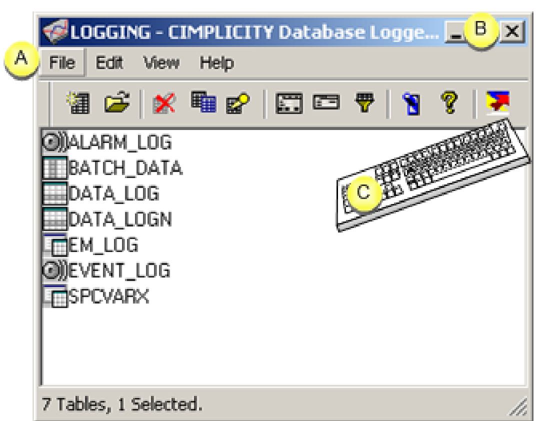
Option 9.2. Exit the Database Logger from an Open Table

Click the **Exit Window** button ☒ on the top right corner of the table.

Result: The Database Logger closes.

10. Exit the Database Logger Configuration Window

Do one of the following in the Database Logger Configuration window.



A	Click File>Exit on the Database Logger Configurationwindow menu bar.
B	Click the **Exit Window** button on the top right corner of the table.
C	Press Alt+F+C on the keyboard.

Result: The Database Logger Configuration window closes.

Note: You can also exit the Database Logger Configuration window by clicking the **Close Window** button on the top right corner of any open table.

Table Renamed or Copied

Choose either:

Copy a table.
Rename a table.

Database Logger Default Logging Properties

Database Logger Default Logging Properties

The Database Logger Configuration window provides you with the capability to set global defaults that specify:

- Open the Logging Properties dialog box. See Open the Logging Properties Dialog Box *(page 196)*.
- Database Logger Default general parameters. See Database Logger Default General Parameters *(page 196)*.
- Database Logger Defaults for alarm, event, status log, and application logs. See Database Logger Defaults for Alarm, Event, Status Log, and Application Logs *(page 201)*.
- Database Logger Defaults for point and group point logs. See Database Logger Defaults for Point and Group Point Logs *(page 205)*.

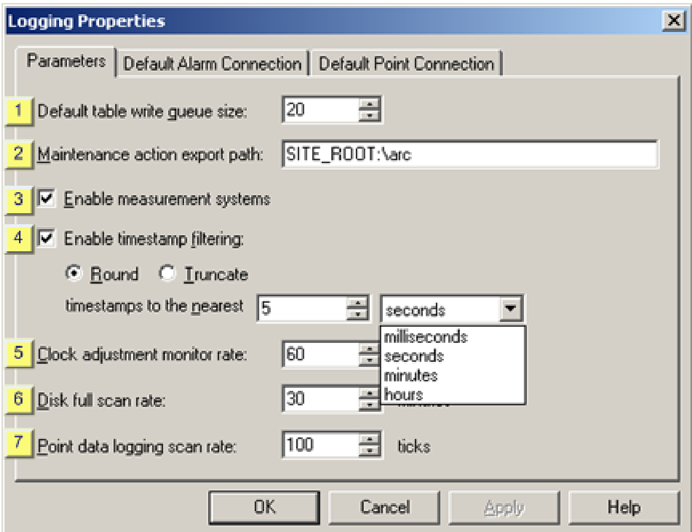
You can change these defaults for any individual table when you are configuring that table's properties.

Open the Logging Properties Dialog Box

1. Right-click the Database Logger window.
2. Select Logging Properties on the Popup menu.

Database Logger Default General Parameters

Because there are several parameters that may be the same for all your logging tables, the Database Logger provides you with the ability to specify defaults. The default you set will be the initial value a table uses. However, you can change the values for individual tabls.



rect -2, 263, 281, 290 [(page 200)]
rect 2, 110, 241, 137 [(page 198)]
rect -1, 135, 363, 207 [(page 199)]
rect -1, 205, 238, 232 [(page 200)]
rect -3, 230, 236, 257 [(page 200)]
rect 3, 78, 426, 105 [(page 198)]
rect 1, 53, 248, 80 [(page 198)]

Select the Parameters tab in the Logging Properties dialog box.

1 (page 198)	Default table write queue size
2 (page 198)	Maintenance action export path
3 (page 198)	Enable measurement systems
4 (page 199)	Enable timestamp filtering
5 (page 200)	Clock adjustment monitor rate
6 (page 200)	Disk full scan rate

7 (page 200)	Point data logging scan rate

1	Default table write queue size

A queue size that supports the worst-case logging during activity bursts.

For example, if 10 points are being logged to the DATA_LOG table, you should have a queue size of at least 10 in case the points' logging conditions all occur simultaneously.

Guidelines

The default queue size is the size of the queue of logged data for each table in the **Database queue size** field.

- If the database queue for a table overflows a message is logged to your project's Status Log and additional write requests are dropped until there is room in the queue for them.
- If there is more logging than can be supported by the benchmarked data rate for the database increasing the size of the database queue will not resolve the problem of lost data.
- Increasing the size of the Database Logger queue will increase the amount of memory used by the database logger. Since each table pre-allocates memory for the queue, this growth will take place at startup time. For each additional entry in the queue, the following additional memory will be used:

Alarm Log Table	200 bytes per entry.
Event Log Table	200 bytes per entry.
Data Tables	Size of logged data plus 36 bytes.
Group Tables	Size of logged data plus 36 bytes.
Database Logger Default:	**20**

2	Maintenance action export path

The system path to the directory that holds the files created by table maintenance export and purge actions in the, if it is different from the default.

Default	**SITE_ROOT:\ARC** places the data into the **ARC** subdirectory of the project.

Guidelines

You may use the logical directory name SITE_ROOT to specify a different subdirectory of your project, or use any other full path specification such as D:\EXPORT.

3	Enable measurement systems

Checked	Database Logger logs data for point values using the active measurement system converted value.

Note: The Database Logger logs values for device points and virtual points are as follows:

Device points	Checked and the measurement system is active	Database Logger logs the data in the specified units. **Note:** You can also log the raw value for device points by checking **Raw Value** on the Log Attribute tab in a table's Properties dialog box.
	Clear or no measurement system is active	Database Logger logs the EU converted value.
Virtual points	The database logger logs the derived value of the point.	

Example

If you configure a point to be in inches, if you are using a metric system and that point is showing as centimeters, the Database Logger is still going to log it as inches unless you check Enable Measurement Systems.

4	Enable timestamp filtering

Checked	Limits the number of characters in a table's timestamp column. You can choose to either round or truncate the timestamp.	
	Default	Clear

Option 1–Rounding up or down

1. Check Round to round the actual time is rounded in the timestamp to the nearest specified number of units.
2. Enter the nearest number and unit to which the timestamp will be rounded in the **timestamps to the nearest** fields.

Example

You specify that timestamps be rounded to the nearest 5 seconds.

The time is **10:50:22:05**. It will be rounded to **10:50:20**.

The time is **10:50:24:05**. It will be rounded to **10:50:25**.

Option 2–Truncating

1. Check **Truncate** to truncate the actual time in the **timestamp to** the nearest specified unit.
2. Enter the unit to which the timestamp will be truncated.

Example

You specify that timestamps be truncated to seconds.

The time is 10:50:22:05. It will be truncated to 10:50:22.

The time is 10:50:24:55. It will be truncated to 10:50:24.

5	Clock adjustment monitor rate

Rate (in seconds) at which system clock adjustments are checked. All synchronized timed events are adjusted when a system clock adjustment occurs.

Default	60

Valid values from 1 through 3600.

6	Disk full scan rate

An interval (in minutes) the Database Logger should wait between disk scans.

Default	30

Guidelines

The Database Logger scans the disk at intervals to determine if it is full. The **Disk full scan rate (min)** determines how long the Database Logger waits after completing one scan before it scans the disk again.

The default for the Disk full scan rate affects tables for which you have enabled Disk Space Low maintenance event.

Note: The Disk Full condition is only useful for Microsoft Access (As-Is product) databases. The SQL Server forces pre-allocation of disk space for data tables.

7	Point data logging scan rate

A value in ticks (100 ticks=1 second) in the **Point data logging scan rate** field that is:

- Faster than the fastest table scan rate in ticks at which the Point Data Logger will submit data for logging.
- Large enough so that all the data for a PLC scan comes into the database together.

Default	100

Guidelines

The Point Data Logging scan rate is the rate at which the Point Data Logger submits data for logging.

Result: The pathname you enter is validated when the Parameters tab is closed.

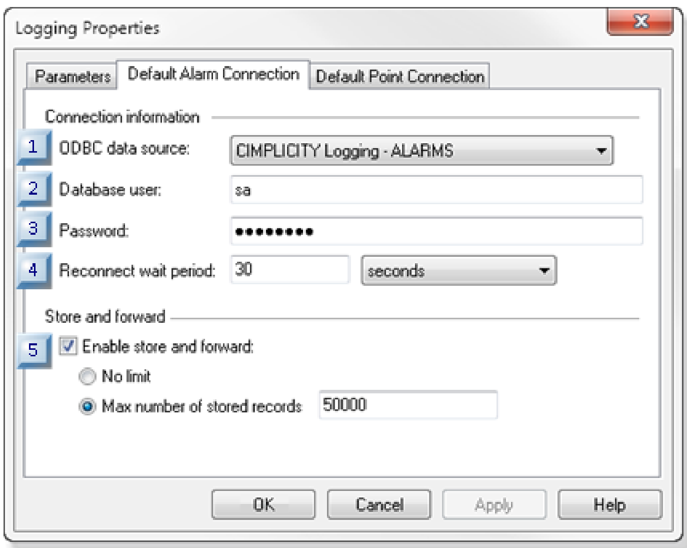
Database Logger Defaults for Alarm, Event, Status Log, and Application Logs

You can specify a single global ODBC data source that will be used when logging to Alarm, Event, Status Log, and Application database files.

The Default Alarm Connection section is displayed when you select the Database Logger: A&E & App option in the Project Properties dialog box.

When you configure each table, you can choose to use these defaults or you can specify an ODBC data source that is unique for that table.

In the Logging Properties dialog box, select the Default Alarm Connection tab.



rect 3, 75, 32, 98 *(page)*
rect 4, 100, 32, 126 *(page)*
rect 5, 126, 32, 150 *(page)*
rect 5, 152, 32, 176 *(page)*
rect 4, 203, 31, 227 *(page)*

1. ODBC data source.

2. Database user.

3. Password.

4. Reconnect wait period.

5. Enable store and forward.

6. Additional information that may be needed.

ODBC Data Sources

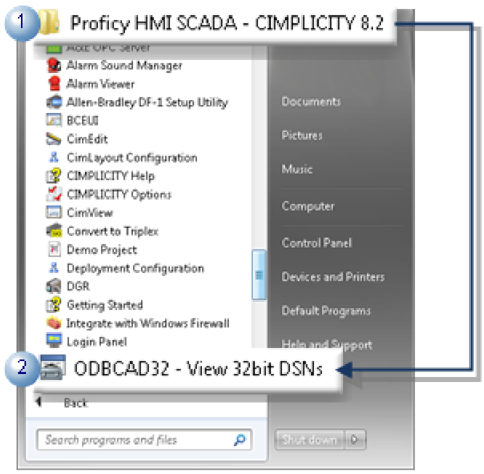
Issues that apply to the ODBC data source for alarm, event, application, point and group point logs include the following.

- ODBC Data Source on 64-bit System
- ODBC Connection to SQL Server 2012 using Windows Authentication
- Database to Database Information

ODBC Data Source on 64-bit System

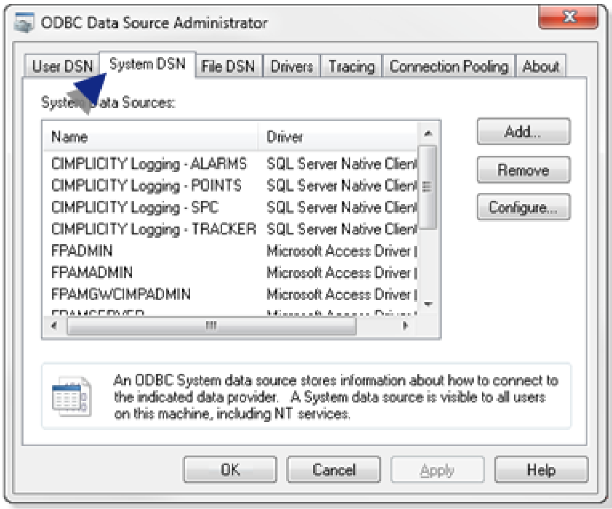
Important: System Data Sources may not be listed on the System DSN tab when you open The ODBC Data Source Administrator on a 64-bit system.

1. From the CIMPLICITY Start menu, expand All Programs>HMI SCADA - CIMPLICITY on the Windows start menu.



2. Click ODBCAD32 - View 32bit DSNs.
3. From the ODBC Data Source Administrator opens, select the DSN tab. The System Data Sources will be listed.
4. Click the Start button on the Windows task bar.
5. Enter the following in the **Command Line** field: c:\windows\syswow64\odbcad32.exe
6. Press Enter. The ODBC Data Source Administrator opens.
7. Select the DSN tab. The System Data Sources will be listed.
8. Select from available options or configure an existing or a new ODBC data source.
9. Select from the one or more Microsoft Access (As-Is product) or SQL Server options in the drop down menu. Basic options include:
    - CIMPLICITY Logging - Alarms Logs data to the cimplog.mdb file in the ARC directory of your CIMPLICITY project.
    - CIMPLICITY Logging - Points Logs data to the pointlog.mdb file in the ARC directory of your CIMPLICITY project.
10. Configure an existing or add a new ODBC data source.

11. Click the **ODBC Data Source** button to the right of the **ODBC data source** field. The ODBC Data Source Administrator opens.



12. Select the System DSN tab.
13. Configure the new ODBC data source based on the data source procedures. Consult Microsoft documentation.
14. Close the ODBC Data Source Administrator.
15. Select the new ODBC data source from the drop down menu.

ODBC Connection to SQL Server 2012 using Windows Authentication

The NT AUTHORITY/SYSTEM role must have administrative privileges for the Database Logger to connect to ODBC. Beginning with SQL Server 2012 NT AUTHORITY/SYSTEM does not have administrative privileges by default. The privilege needs to be granted in the Microsoft SQL Server Management Studio.

1. Open the Microsoft SQL Server Management Studio.
2. Expand **Security>Logins** in the Object Explorer left-pane.
3. Right-click **NT AUTHORITY\SYSTEM**.
4. Select Properties on the Popup menu. The Login Properties - NT AUTHORITY\SYSTEM dialog box opens.
5. Select **Server Roles** in the Login Properties - NT AUTHORITY\SYSTEM dialog box left pane.
6. Check sysadmin in the right-pane.
7. Save your changes; close the Microsoft SQL Server Management Studio.

The Database Logger should be able to connect to ODBC using Windows Authentication. Consult Microsoft documentation for additional details.

Database to Database Information

Information needed varies from database to database. In general if you are connecting to:

- A Microsoft Access database, you may be prompted for a file name. Microsoft Access (As-Is product) does not fully support the long point IDs or alarm messages that were introduced in CIMPLICITY V9.0. Consult Microsoft documentation for details about maximum character support.

- A SQL Server, you may be prompted for a database name.
- An Oracle database, you may be prompted for a Server ID. Enter the Alias Name for the Oracle database in this field.

If the Data Logger is unable to connect to the selected database, validation fails.

Database user

User who will connect to the selected database driver. A database user name is required if you are connecting to a SQL Server. Make sure that the name is a valid database user name.

Password

Password needed to connect to the selected database driver. A password is required if you are connecting to a SQL Server. Make sure that the password is a valid database password.

Reconnect wait period

The amount of time that the Database Logger waits between reconnect attempts when the connection to the database is lost.

Enter a value from **0** (continuous retries) through **24** hours. Default: 30 seconds.

Enable store and forward

Check Enable store and forward to enable the ALARM_LOG table perform store and forward options:

Option	Check if you want:
No Limit	The Database Logger to store an unlimited number of records when its connection to the database is down. The number of records actually stored is determined by the amount of time the connection is lost and by the amount of free disk space you have.
Max number of stored records	To select the number of records the Database Logger will store when its connection to the database is down. Enter a number between **1** and **4294967285**.

Additional information that may be needed

Information needed varies from database to database. In general if you are connecting to:

- A Microsoft Access database, you may be prompted for a file name.

- A SQL Server, you may be prompted for a database name.
- An Oracle database, you may be prompted for a Server ID. Enter the Alias Name for the Oracle database in this field. If the Data Logger is unable to connect to the selected database, validation fails.

The Database Logger validates your entries. You will be prompted if additional information is required to connect to the database.

Database Logger Defaults for Point and Group Point Logs

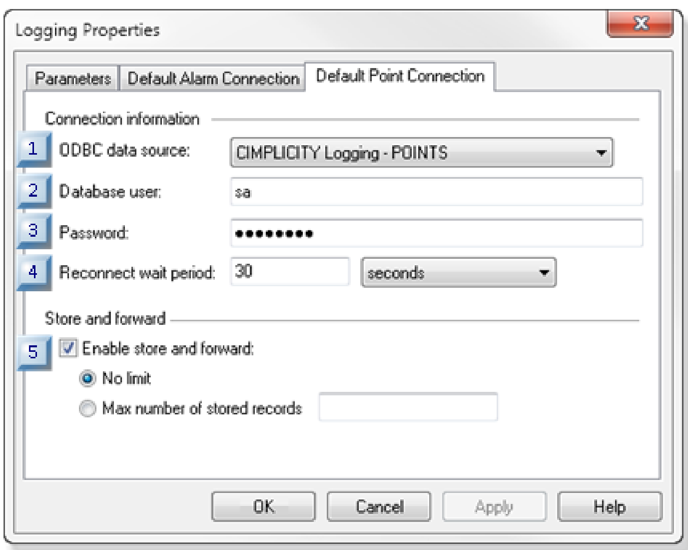
You can specify a single global ODBC data source that will be used when logging data to CIMPLICITY data and group tables.

Note:

1. The Default Point Connection tab displays when the Database Logger: Points option is checked in the Project Properties dialog box.

2. When you configure each table, you can choose to use these defaults or you can specify an ODBC data source that is unique for that table.

   Select the Default Point Connection tab in the Logging Properties dialog box.



rect 4, 203, 33, 227 *(page)*
rect 3, 153, 32, 177 *(page)*
rect 3, 125, 32, 149 *(page)*
rect 3, 101, 32, 125 *(page)*
rect 3, 75, 32, 99 *(page)*

Configure the default parameters for the CIMPLICITY point and group log tables the same way you configure the parameters *(page 201)* for the CIMPLICITY alarm, event and application log tables.

1 *(page)*	ODBC data source
2 *(page)*	Database user.
3 *(page)*	Password.
4 *(page)*	Reconnect wait period.
5 *(page)*	Enable store and forward.
6 *(page)*	Additional information that may be needed.

Review Database Logger Default Logging Connections

Review Database Logger Defaults for CIMPLICITY:

- Database Logger Defaults for Alarm, Event, Status Log, and Application Logs *(page 201)*
- Database Logger Defaults for Point and Group Point Logs *(page 205)*

Review Logging Connections

Review Logging Connections for CIMPLICITY:

- ALARM_LOG. See Step 3.6. Configure the Alarm Log Connection *(page 262)*.
- COR_LOG. See Step 2.4. Configure the COR_LOG Connection *(page 291)*.
- Data log table. See Step 2.6. Configure the Logging Connection *(page 229)*.
- EM_LOG. See Step 2.5. Configure an EM_LOG Connection *(page 284)*.
- EVENT_LOG. See Step 3.5. Configure the EVENT_LOG Connection *(page 273)*.

- Group log table. See Step 2.6. Configure the Group Connection *(page 246)*.
- An external application log. See Application Logging *(page 295)*.

Review Default Logging Conditions for Points

Review configuration for a CIMPLICITY:

Data table.
Group table.

Point Data Logging

Point Data Logging

Data Logging	Enables you to log point data individually.
Default point data log table	DATA_LOG

Point data logging provides you with a straightforward process to create and maintain records for analyzing the performance of selected points.

The point performance represents the actual performance of equipment and processes in your system. As a result, point logging provides you with an in depth record that can help you determine if action is required to improve the performance of any equipment or process in your system.

Configuration steps in an open *(page 184)* data log table are as follows.

Step 1 *(page 209)*	Add points to a data table.
Step 2 *(page 212)*	Configure CIMPLICITY point DATA logging properties.
Step 3 *(page 236)*	Configure logging conditions for a single point.

Important: When using fully qualified points, you must provide remote project log inconfiguration with the project.

Indexes, Columns, Rows in CIMPLICITY Data Log Tables

- Indexes in data logging tables.

- Columns and rows in data logging tables.

Indexes in data logging tables

Indexes in data logging tables include a:

Index	Description	
Primary key	Is on the Joined Point ID and Timestamp columns. Includes the following if:	
	The project name table attribute is selected.	Project name column
	milliseconds is enabled.	An MSEC column
Secondary key	Joined Point ID and timestamp_utc columns.	
Secondary key	Is on the timestamp only.	
Secondary key	Is on the timestamp_utc only.	

Columns and rows in data logging tables

Columns and **rows** in data logging tables are as follows.

Column Name	Data Type	Description
Constant fields		
timestamp	date/time	Timestamp of the logging event
timestamp_utc	date/time	UTC timestamp of the logging event.
point_id	string	CIMPLICITY identifier for the point
Attribute fields		
_ALRM	string	Alarm state of the point .
_ENG	string	Engineering units label of the point.
_PREV	As configured. Default is string	Previous logged value of the point for the same logging event.
_VAL	As configured. Default is string	Point value.
_TIME	date/time	Previous time the point was logged for the same logging event.
_TIME_UTC	UTC date/time	Previous UTC time the point was logged for the same logging event.
_RAW	As configured. Default is string	Raw value of the point.
_RES	string	CIMPLICITY Resource associated with the point.
_QUALITY	number	Point quality flags.
_USER	number	Value of the user defined flags.
Optional fields		
project	string	Name of the CIMPLICITY project

msec	number	Actual number of milliseconds in the timestamp

Review Logging Attributes

Review Logging Attributes for CIMPLICITY:

ALARM_LOG
EM_LOG
EVENT_LOG
An external application log.

Review Logging Attributes for Points

Review Logging Attributes for CIMPLICITY:

Data log table
Group log table

Step 1. Add Points to a Data Log Table

Step 1. Add Points to a Data Log Table

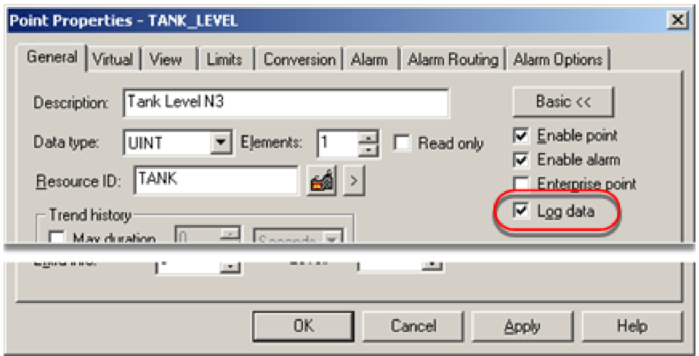
You can add points to a data or group log through either the:

Step 1.1 *(page 209)*	Point Properties dialog box
Step 1.2 *(page 210)*	Database Logger

Step 1.1. Add Points to Log Tables via a Point Properties Dialog Box

1. Openthe Point Properties dialog box for a selected device or virtual point.
2. Select the General tab.
3. Check Log data.

   Note: The Log data checkbox displays when the option Database Logger: Points is checked in the Project Properties dialog box.



4. Click OK or Apply.

CIMPLICITY adds the point to its default DATA_LOG. You can apply more specifications in the Database Logger.

Step 1.2. Add Points to Data Log Tables through Database Logger

Add points to a data log table through the:

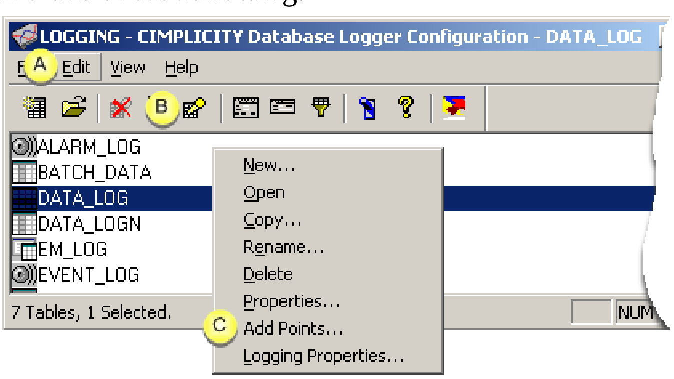
- Database Logger Configuration window.
- Data log table

Database Logger Configuration window

1. Open the CIMPLICITY Database Logger Configuration window.

2. Select a DATA_LOG table.

3. Do one of the following.

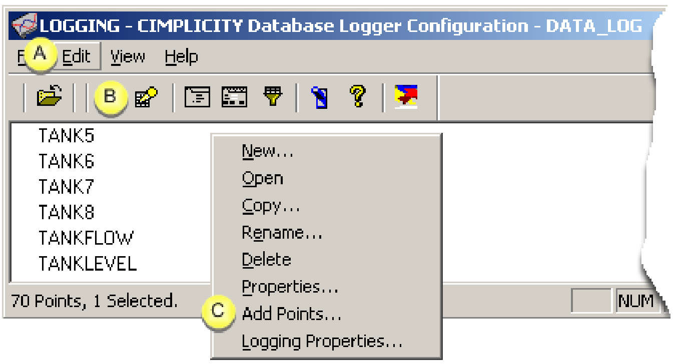


A	Click Edit>Add Points on the CIMPLICITY Database Logger Configuration window menu bar.
B	Click the Add Item button on the Database Logger Configuration window toolbar.
C	a. Right-click in the CIMPLICITY Database Logger Configuration window.

	a. Select Add Points... on the Popup menu.

Data log table

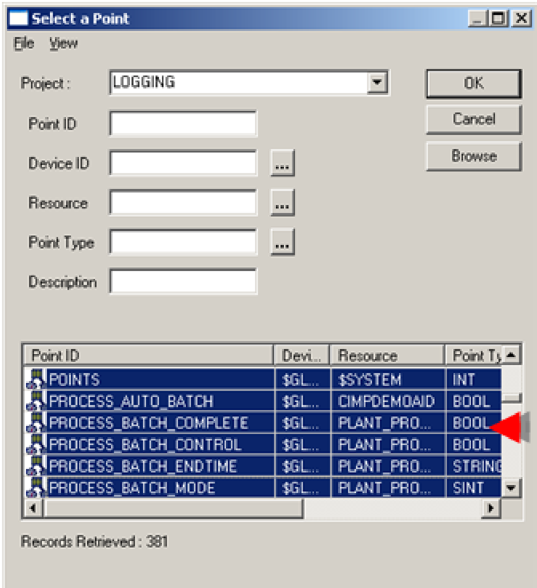
4. Right-click in the CIMPLICITY Database Logger Configuration window.
5. Select Add Points... on the Popup menu.
6. Select a data log table.
7. Do one of the following.



A	Click File>Add Points on the table menu bar.
B	Click the Add Item button  on the log table toolbar.
C	a. Right-click in the log table window. b. Select Add Points… on the Popup menu.

The Select a Point browser opens when you use any method.

8. Right-click in the log table window.
9. Select Add Points… on the Popup menu.
10. Select the Point IDs you want to add.



11. Click **OK**.

The Point IDs display in the data table window and will be logged according to your specifications.

Step 2. Configure CIMPLICITY Point Data Logging Properties

Step 2. Configure CIMPLICITY Point Data Logging Properties

You configure the point logging properties for all points in a data log table in a Table Properties dialog box. You can adjust the logging conditions for a single point in the (Database Logger's) Point Properties dialog box..

The tasks to configure the point logging properties include:

Step 2.1 *(page 213)*	Open a CIMPLICITY data log table Properties dialog box.
Step 2.2 *(page 214)*	Configure data logging conditions.
Step 2.3 *(page 222)*	Configure data logging attributes.
Step 2.4 *(page 224)*	Configure data log maintenance events.
Step 2.5 *(page 227)*	Configure maintenance actions.

Step 2.6 (page 229)	Configure the CIMPLICITY point data log connection.
Step 2.7 (page 234)	Do advanced configuration for a data log table.

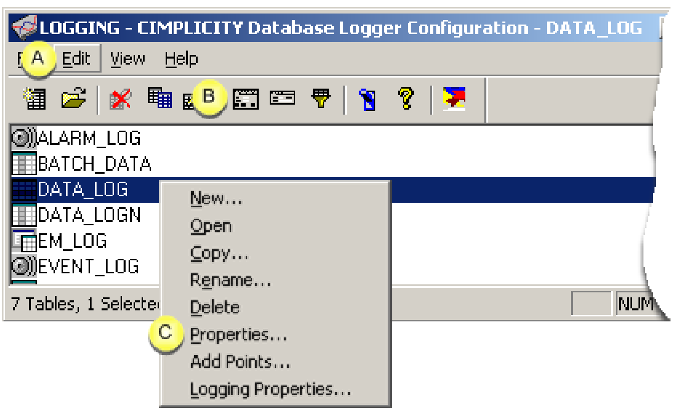
Step 2.1 Open a Table Properties Dialog Box

- Database Logger Configuration window.
- Data log table.

Database Logger Configuration window

1. Select a data table (e.g. DATA_LOG) in the Database Logger Configuration window.

2. Do one of the following.

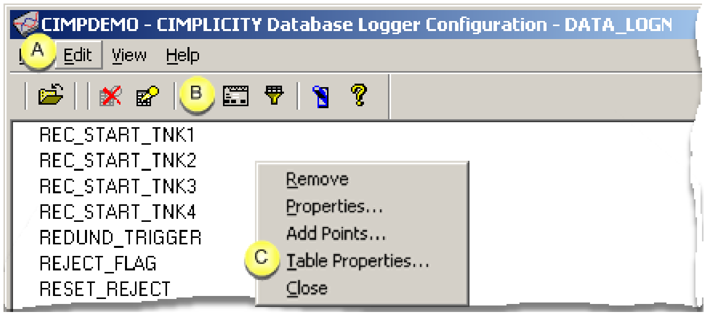


A	Click Edit>Table Properties... on the Database Logger Configuration window's menu bar.
B	Click the Table Properties button  on either window's toolbar.
C	a. Right-click the data table. b. Select Properties... on the Popup menu.

Result. The data table, e.g. DATA_LOG Table Properties dialog box opens when you use any method.

Data log table

Do one of the following.



A	Click Edit>Table Properties... on the data table window's menu bar.
B	Click the Table Properties button on either window's toolbar.
C	a. Right-click the data table. b. Select Table Properties... on the Popup menu.

Result. The data table, e.g. DATA_LOG Table Properties dialog box opens when you use any method.

3. Right-click the data table.
4. Select Properties... on the Popup menu.
5. Right-click the data table.
6. Select Table Properties... on the Popup menu.

Step 2.2. Configure Default Logging Conditions

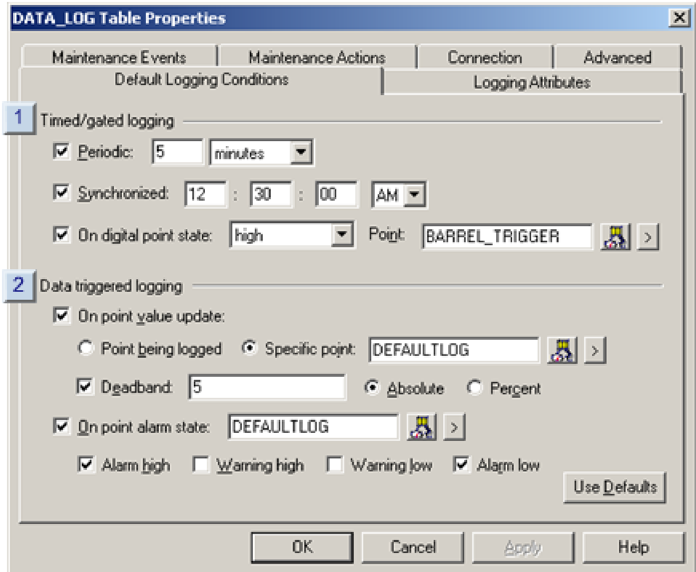
Step 2.2. Configure Default Logging Conditions

Logging Conditions determine when data will be logged into the table.

Note: For a CIMPLICITY data log table, you can change the table's default logging conditions for an individual point in the (Database Logger's) Point Properties dialog box.

Select the Default Logging Conditions tab.

Configuration includes the following.



Option 2.2.1 *(page 215)*	Time/gated logging
Option 2.2.2 *(page 219)*	Data triggered logging.

Tip: Click **Use Defaults** to reset the Logging Conditions to their default values. The default is to log data periodically every 5 minutes.

Option 2.2.1. Timed/Gated Logging and Timed/Gated Maintenance

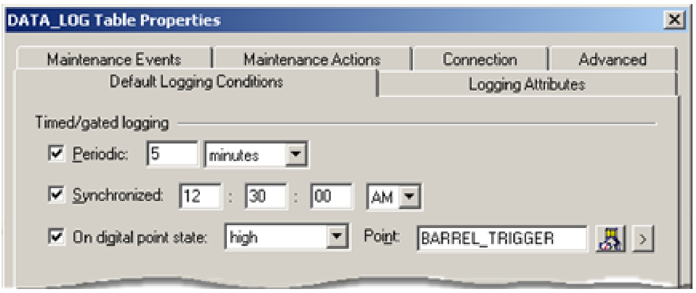
Option 2.2.1. Timed/Gated Logging and Timed/Gated Maintenance

- Options.
- Option combinations.

Options

Options to configure timed/gated logging are as follows.

Note: One or more options can be selected.



rect 19, 78, 198, 102 (page 216)
rect 20, 107, 289, 131 (page 216)
rect 21, 135, 415, 159 (page 216)

Option	Description	
Periodic	Logs data into the table at regular intervals. Configuration for the Periodic configuration group includes:	
	Checkbox	Checked enables periodic logging.
	Value field	Enter the time value.
	Interval field	Select the time interval. Valid intervals are • DAYS • HOURS • MINUTES • SECONDS • **TICKS** (100 ticks = 1 second).
Synchronized	Synchronizes log data to a selected daytime. Configuration for the Synchronized group includes:	
	Checkbox	Checked enables synchronizing.
	Time boxes	Select hour, minutes and seconds (12 hours).
	AM/PM	Select AM or PM from the drop-down list.
On digital point state	Logs data while a digital point is in a selected state. Configuration for the On digital point state group includes:	
	Checkbox	Checked enables On digital point state.
	State	Valid states in the drop-down list are: • high • low • transition (high or low state) If a periodic time interval is not specified, logging will be triggered when the selected digital point transitions to the selected state, or any time the digital point transitions (if transition is selected). If you select a **TRANSITION** point state, together with a periodic or synchronized time, the gate will be ignored.
	Point	Point whose state is monitored for logging. **Note:** If a device point is configured as On Scan, the point value is updated on every scan even though its value remains unchanged.

Note: Fields also appear for timed/gated logging on a:

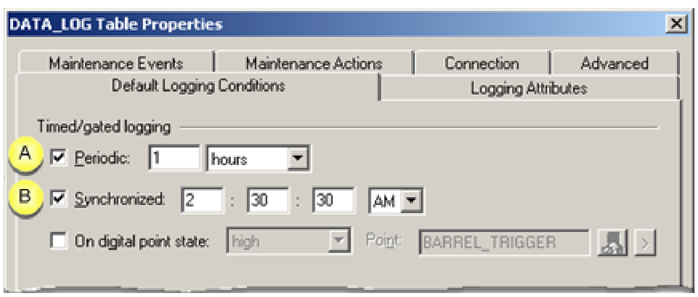
- Data log table's default logging conditions tab.
- Group log table's Logging Conditions tab.

Option combinations

Option 2.2.1.1 (page 217)	Periodic and synchronized.
Option 2.2.1.2 (page 217)	Periodic and On Digital Point State.
Option 2.2.1.3 (page 218)	Synchronized and On Digital Point State.
Option 2.2.1.4 (page 218)	Periodic, Synchronized and On Digital Point State.

Option 2.2.1.1. Periodic and Synchronized

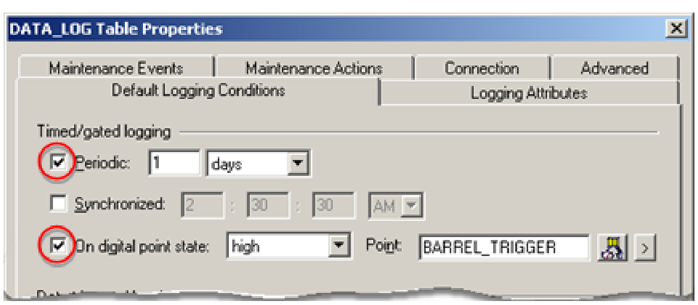
The combination of periodic and synchronized actions are executed as follows.



A	Periodic	Periodically at the specified time interval
B	Synchronized	Beginning at the specified time of the day

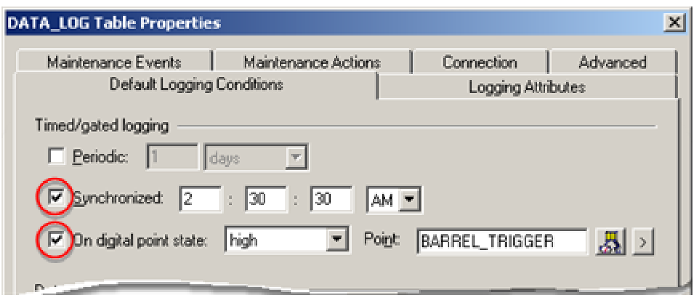
Option 2.2.1.2. Periodic and on Digital Point State

The combination of periodic and On digital point state actions are executed as follows.



1	If the Digital Point state is:	HIGH	LOW	TRANSITION
	Logging and maintenance actions are executed when:			
2	The Digital Point transitions to:	HIGH	LOW	HIGH or LOW
3	And: • At the specified time interval			
	• While the digital point state is:	HIGH	LOW	HIGH or LOW

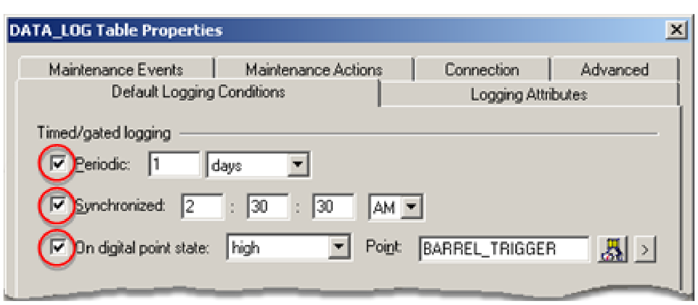
Option 2.2.1.3. Synchronized and On Digital Point State



For Synchronized and On digital point state:

1	If the Digital Point state is:	HIGH	LOW	TRANSITION
	Logging and maintenance actions are executed:			
2	• At the specified day time.			
	• While the digital point state is:	HIGH	LOW	HIGH or LOW

Option 2.2.1.4. Periodic, Synchronized and On Digital Point State

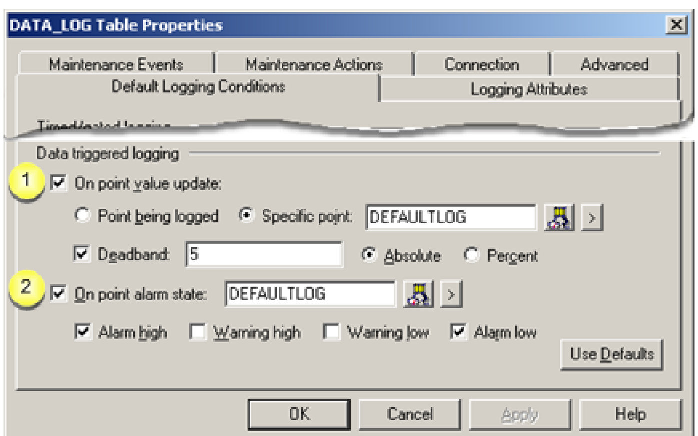


For Periodic, Synchronized and On digital point state as follows.

1	If the Digital Point state is:	HIGH	LOW	TRANSITION
	Logging and maintenance actions are executed:			
2	• At the specified day time			
	• If the Digital Point state is:	HIGH	LOW	HIGH or LOW
	And			
3	• At the Specified Time Interval			
	• While the Digital Point state is:	HIGH	LOW	HIGH or LOW

Option 2.2.2. Data Triggered Logging

Options to configure data triggered logging are as follows.



1 (page 220)	On point value update
2 (page 221)	On point alarm state
3 (page 222)	Data triggered logging guidelines

1	On point value update

On point value update logs values when a selected point is updates.

Configuration for the **On point value update** group includes:

Option	Description		
Checkbox	Enables On point value update option.		
Radio buttons	Check one option to determine the conditions for when the values of all the points in the table will be logged.		
	Point being logged	Each point that was added to the table after **Point being logged** was checked (and does not have a customized logging condition) is logged when its value is updated.	
	Specific point	Selected point is updated. Enter a Point ID for this option. • Click the **Browser** button to open the Select a Point browser or • Click the **Popup Menu** button to display options for selecting a point.	
Deadband	filter out changes in the value of the selected point type before triggering a logging event–for either the point being logged or the specific point. Configuration in the Deadband group includes:		
	Checkbox	Checked enables deadband.	
	Value	Floating point number. Deadband value	
	Radio buttons	Check one to specify how to evaluate the entered value to trigger another logging event.	
		Absolute	Absolute value
		Percent	Percent of the last value.
	The point's value must change more than the Deadband value before another logging event will be triggered.		

Deadband Example 1. Absolute Value

An Absolute Deadband value is 5.0.

Point Value	Result
200.0	First logging event
201.3	No Logging event
204.2	No Logging event
205.1	Logging event triggered.

Deadband Example 2. Percent Value

A Percentage Deadband value is 5.0

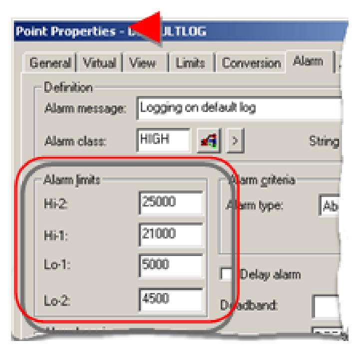
Point Value	Result
200.0	First logging event
201.3	No Logging event
204.2	No Logging event
205.1	No Logging event
210.2	Logging event triggered.

Note: An explicit deadband of **0** specifies that the point must have actually changed as opposed to having been updated by the point manager.

2	On point alarm state

On point alarm state logs point values when a selected point goes into one or more selected alarm states.

Configuration for the On point alarm state group includes

Option		Description
Checkbox	Enables the on point alarm state option.	
Field	Entered the point to be evaluated. Aids to find the point include:	
		Opens the Select a Point browser
		Displays a Popup menu with options for selecting a point.
	Important: For On point alarm state to work successfully, you must define alarm limits for the trigger point. For example, if you select a Point ID and choose its Alarm High state, logging in the table will occur each time the point enters the Alarm High state.	
Alarm limits	One or more alarm states must be checked. The values for these alarm states is specified on the Alarm tabin the selected point's Properties dialog box.	
	Alarm high	
	Warning high	
	Warning low	
	Alarm low	

Tip: Click **Use Defaults** to reset the Logging Conditions to their default values. The default is to log data periodically every 5 minutes.

3	Data triggered logging guidelines

- You can combine **On point value update** logging with **On point alarm state** logging.
- If you do not define any logging conditions under Timed/Gated Logging, then data will only be logged into the table when the trigger occurs.
- You can combine triggers. When you combine triggers, logging occurs when either of the triggers occurs.

Example

You want to log a set of points when POINT_A is updated or when POINT_B is in Alarm High state. Do the following.

1. Check the **On point value update** check box.
2. Enter POINT_A in the **Point** field.
3. Check the **On point alarm state** check box.
4. Check the **Alarm High** check box.
5. Enter POINT_B in the **Point** field.

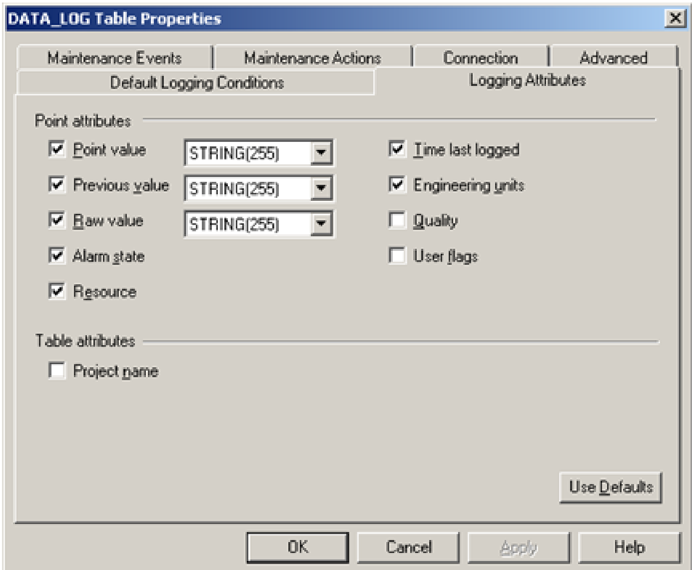
Step 2.3. Configure Data Logging Attributes

Logging Attributes for a data log table (e.g. DATA_LOG) enables you to:

- Select the current point attributes that will be logged to the table.
- Log the name of the CIMPLICITY project.

1. Select the Logging Attributes tab.

2. Check any of the following point attributes to log.



rect 27, 83, 214, 104 (page 223)
rect 27, 102, 214, 123 (page 223)
rect 27, 122, 214, 146 (page 223)
rect 27, 145, 106, 169 (page 223)
rect 27, 167, 106, 191 (page 223)
rect 237, 79, 333, 103 (page 224)
rect 237, 101, 333, 125 (page 224)
rect 237, 123, 333, 147 (page 224)
rect 237, 145, 333, 169 (page 224)
rect 27, 217, 106, 241 (page 224)

Point attributes	Log the
Point Value	Point's current value. Select a Data Type. The Database Logger will attempt to "force" the logged data into the type you select. Data type options are: • BOOL • SINT • USINT • INT • UINT • DINT • UDINT • REAL • STRING (255)
Previous Value	Previous value recorded in this table for this point. Data type options are the same as for Point Value *(page 223)*.
Raw Value	Point's raw value. Data type options are the same as for Point Value *(page 223)*.
Alarm State	Point's alarm state when the value is being logged.
Resource	Point's Resource ID.

Time Last Logged	Time the point was last logged in this table.
Engineering Units	Point's engineering units.
Quality	Is the value of the **QUALITY** attribute. .
User Flags	Value of the **USER_FLAGS** attribute.
Table attributes	
Project name	Project name.

Tip: Click **Use Defaults** to reset the Logging Attributes to their default values. The defaults are to:

- Log **Point Value (**STRING(255) in the DATA_LOG**)** and
- Not log the **Project Name**.

Step 2.4. Configure Logging Maintenance Events

Maintenance Events define when export, purge, and command actions will be performed on the table.

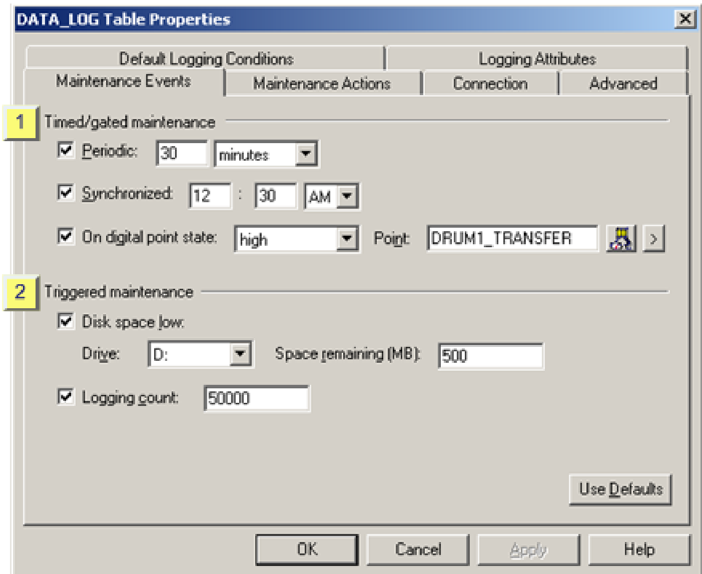
You can specify that maintenance be triggered when:

- A condition, such as low disk space, occurs or
- On a periodic or gated interval

Any specified event will trigger maintenance actions.

Select the Maintenance Events tab in a data Table Properties dialog box.

Configuration options are as follows.



rect 4, 55, 418, 158 [(page 225)](#)
rect 4, 156, 418, 259 [(page 226)](#)

1 (page 225)	Timed/gated maintenance
2 (page 226)	Triggered maintenance

1	Timed/gated maintenance

Specifies when maintenance will be triggered.

Use any combination of the options.

Option	Description	
Periodic	Triggers maintenance at regular intervals. Configuration for the Periodic configuration group includes:	
	Checkbox	Checked enables periodic maintenance.
	Value field	Enter the time value.
	Interval field	Select the time interval. Valid intervals are • minutes • hours • days
Synchronized	Triggers maintenance at a selected time of day. Configuration for the Synchronized group includes:	
	Checkbox	Checked enables synchronizing.

	Time boxes	Select hour (12 hours) and minutes.
	AM/PM	Select AM or PM from the drop-down list.
On digital point state	Logs data while a digital point is in a selected state. Configuration for the On digital point state group includes:	
	Checkbox	Checked enables On digital point state.
	State	Valid states in the drop-down list are: • high • low • transition (high or low state) If a periodic time interval is specified Maintenance actions will be executed when the selected digital point is in the selected state. If a periodic time interval is not specified Maintenance actions are triggered when the specified digital point transitions to the selected state, or any time the digital point transitions (if transition is selected).
	Point	Valid states are **HIGH**, **LOW**, and **TRANSITION.** If a periodic time interval is specified, maintenance actions will be executed when the digital point you choose is in the selected state. If a periodic time interval is not specified, the maintenance actions are triggered when the specified digital point transitions to the selected state, or any time the digital point transitions (if transition is selected).

Up *(page 224)*

2	Triggered maintenance

Specify if and when maintenance events should be triggered.

Use either or both of the following conditions.

Condition	Description	
Disk space low	A drive's space falls below a specified size. Configuration for the **Disk space low** group includes:	
	Check box	Checked enables Disk space low.
	Drive	Select the drive.
	Space remaining	Megabytes Maintenance events occur when space falls below the threshold specified in this field.
	Note: The disk is checked at an interval that you specify on the Parameters tab the Logging Properties dialog box.	
Logging count	The logging count exceeds a specified number.	
	Check box	Checked enables Logging count.

	Field	Number of records. Maintenance actions will be executed when the number of records logged to the table reaches a multiple of this number. **Example** You specify 500. Maintenance actions are executed when the count reaches 500, 1000, 1500, etc. **Note**: If you check both conditions maintenance is triggered when either condition occurs.

Important: Record deletion may not actually free disk space; this can depend your DBMS. Consult your DBMS documentation.

Up *(page 224)*

Step 2.5. Configure the Logging Maintenance Actions

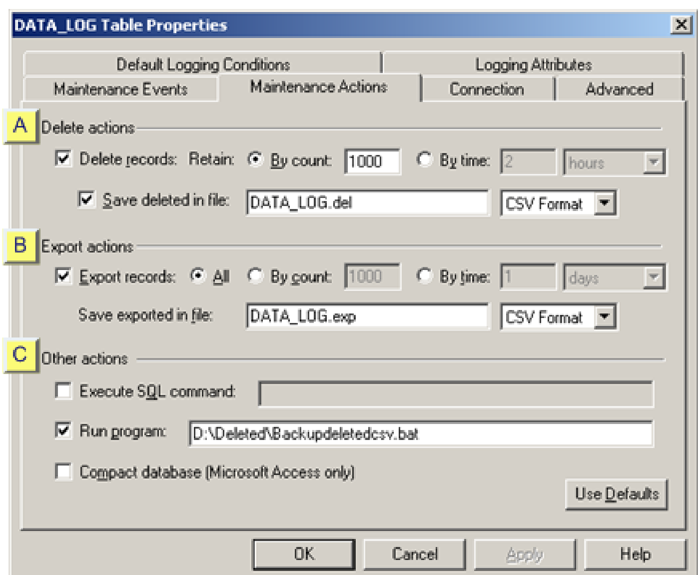
Maintenance Actions define the type of actions that take place when a Maintenance Event occurs.

You can configure a data table to:

- Export records to an export file,
- Purge records from a logging table and/or
- Incorporate command line actions.

Select the Maintenance Actions tab in the DATA_LOG Table Properties dialog box.

Options are as follows.



rect -3, 54, 418, 134 *(page 228)*
rect -3, 201, 418, 304 *(page 228)*
rect -3, 132, 418, 202 *(page 228)*

1 *(page 228)*	Delete actions

2 (page 228)	Export actions
3 (page 228)	Other actions

1	Delete actions

Delete actions specify the criteria for how many records will be retained when data is deleted from a logging table when a maintenance event occurs.

Options in the Delete actions group include:

Option	Description	
Delete records	Determines the method of counting the number of records to retain in the table.	
	Checkbox	Check to enable Delete records.
	Check either radio button.	
	By count	Delete all but the last (most recent)<n> records in the table.
	By time	Delete all but the records entered in the table for the last <n> days, hours or minutes.
Save deleted in file	Saves the purged data to a .csv format file before removing it from the table.	
	Checkbox	Check to enable Save deleted in file.
	Field	Name of the .csv file.

2	Export actions

Export actions specify how many records will be exported when a maintenance event occurs.

Export actions specify

Option	Description	
Export records	Exports specified data. Configuration includes:	
	Checkbox	Checked to enable **Export records**.
	Check either radio button.	
	By count	Export last (most recent)<n> records in the table.
	By time	Export all the table for the last <n> days, hours or minutes.
Save exported in file	Saves the exported data in a .csv file.	
	Field	Name of the .csv file.

3	Other actions

(Optional) Select one of the other actions.

Option	Description	
Execute SQL command	Enables you to incorporate a SQL command line action. Configuration includes:	
	Check box	Checked enables Execute SQL command
	Field	Enter a SQL command Example Records should be counted in the data table from the current time minus one day (point values logged in the last 24 hours.) The field entry is: INSERT NTO MYTABLE VALUES NOW(), SELECT COUNT(*) FROM DATA_LOG Where DATA_LOG is the data log table.
Run program	Runs a selected program. Configuration includes:	
	Check box	Checked enables Run program.
	Field	Enter a program path name in the accompanying field. The entry can include command-line arguments for the program. Example Purge actions have been configured to save the records to a .csv file. A program has been written that transfers the files to a zip drive. The field entry is: D:\Deleted\ Backupdeletedcsv.bat Where D:\Deleted\ is the path to the program. Backupdeletedcsv.bat is the program that will be run when a maintenance event occurs.
Compact database	Compacts the Microsoft Access (As-Is product) database where the table resides during a maintenance event.	
	Check box	Checked enables Compact database.

Step 2.6. Configure the Logging Connection

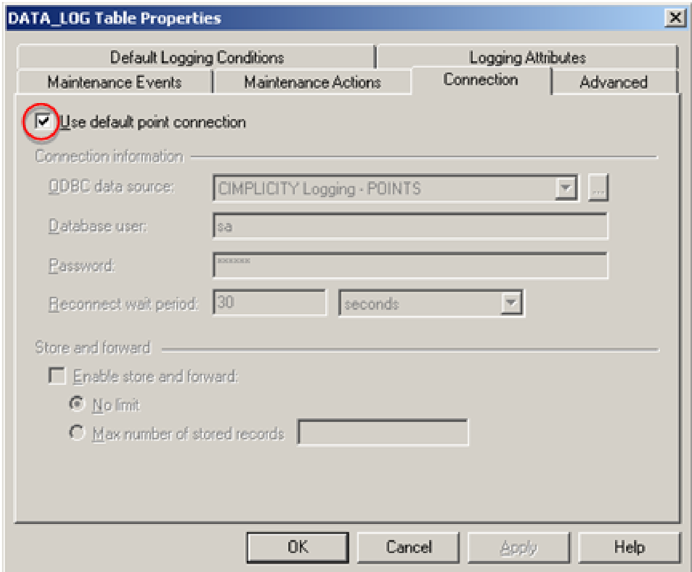
Step 2.6. Configure the Logging Connection

Connection defines the database connection and the Store and Forward properties for a table.

Option 2.6.1 *(page 229)*	Default logging connection.
Option 2.6.2 *(page 230)*	Customized logging connection.

Option 2.6.1. Default Logging Connection

1. Select the Connection tab.
2. Check **Use default point connection**.

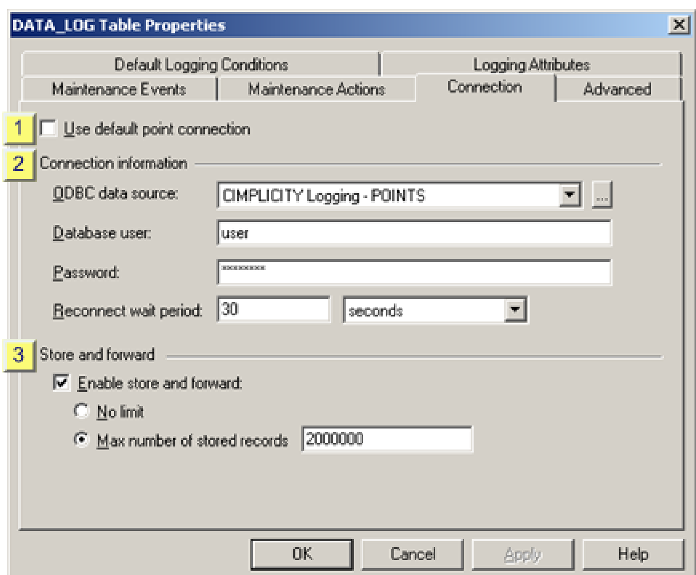


The Connection *(page 205)* tab is dimmed and unavailable for configuration. The data log table uses the connections specified on the Default Point Connection tab in the Logging Properties dialog box.

Option 2.6.2. Logging Connection Selected for a Single Table

Select the Connection tab.

Configuration is as follows.



rect 3, 60, 166, 84 *(page 231)*
rect 2, 203, 399, 291 *(page 233)*
rect 3, 82, 400, 205 *(page 231)*

1 (page 231)	Use default point logging.
2 (page 231)	Connection information
3 (page 233)	Store and forward

1	Use default point connection

Clear Use default point connection.

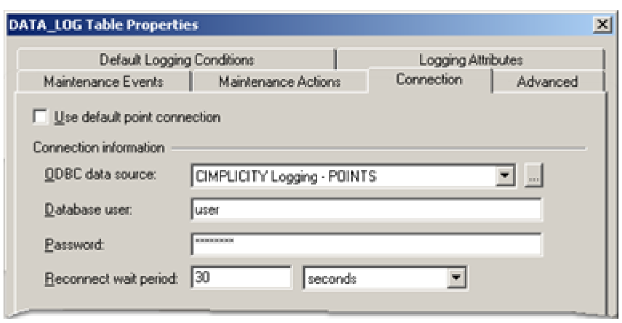
The Connection tab is enabled to enter specifications for the selected table.

2	Connection information

The connection information enables you to select an ODBC data source that is different from the default.

Connection options
ODBC data source selection procedures.

Connection Options



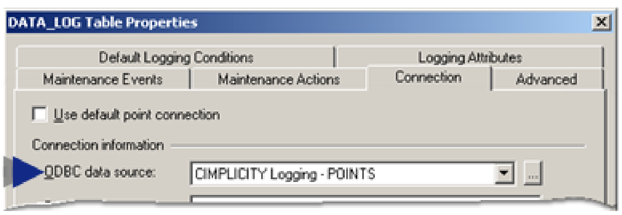
rect 14, 91, 346, 112 (page 231)
rect 14, 110, 346, 134 (page 232)
rect 15, 133, 347, 157 (page 232)
rect 14, 155, 310, 179 (page 232)

Field	Description	
ODBC data source	An ODBC data source can be:	
	A	Selected from the field's drop-down list
	B	A re-configured existing source or a new ODBC data source.

Database user	User who will connect to the selected database driver. **Important:** This field is required if you are connecting to a SQL Server.	
Password	Needed to connect to the selected database driver. **Important:** This field is required if you are connecting to a SQL Server	
Reconnect wait period	A value between 0 (continuous retries) and 24 hours. This value specifies the amount of time **that** the Database Logger waits between reconnect attempts when the connection to the database is lost. Time unites include: • Seconds • Minutes • Hours	
	Default	30 seconds.

Up *(page 231)*

ODBC data source selection procedures



1. Select an ODBC data source from the drop down list in the **ODBC data source,** field. You can select from available options or configure an existing or a new ODBC data source:

   Basic options include, but are not limited to:

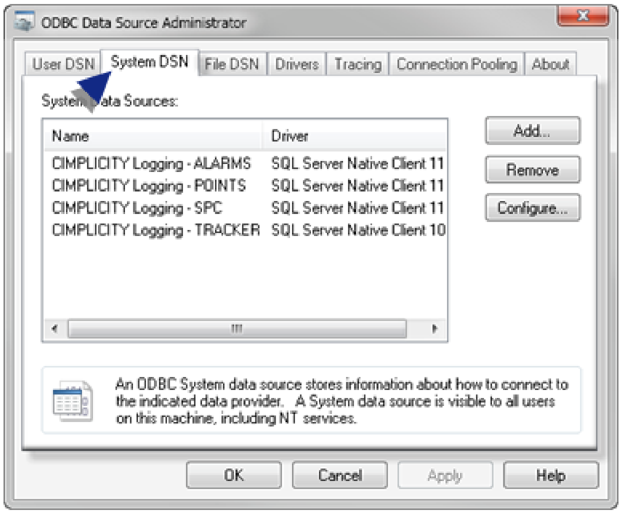
Selection	SQL Server is:	Logs to:
CIMPLICITY Logging - Alarms	Installed	SQL Server database
	Not installed	cimplog.mdb in the ...\Proficy CIMPLICITY\ARC directory
CIMPLICITY Logging - Points	Installed	SQL Server database
	Not installed	pointlog.mdb in the ...\Proficy CIMPLICITY\ARC directory

   Note: If you have an Oracle database, you may see the ODBC data source that you created for Oracle.

2. Configure an existing or add a new ODBC data source.
   a. Click the ODBC Data Source button  to the right of the **ODBC data source** field.

   The ODBC Data Source Administrator opens.

   a. Select the System DSN tab.



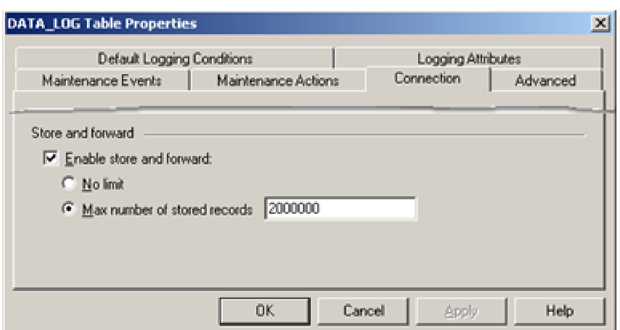
a. Configure the new ODBC data source based on the data source procedures.
b. Close the ODBC Data Source Administrator.
c. Select the new ODBC data source from the drop down menu.

Up (page 231)

3	Store and forward

When Enable store and forward is checked the DATA_LOG will perform store and forward functions.

Options are:



rect 32, 97, 185, 117 (page 233)
rect 33, 115, 269, 135 (page 233)

Option	Description
No Limit	Checked enables No Limit. The Database Logger will store an unlimited number of records when its connection to the database is down. The number of records actually stored is determined by the amount of time the connection is lost and by the amount of free disk space you have.
Max number of stored records	The Database Logger will store a specified number of records when its connection to the database is down. Enter a number between **1** and **4294967285**.

	Checked	Enables Max number of stored records.
	Field	Number that will be the maximum stored. Valid numbers are from 1 to 4294967285.

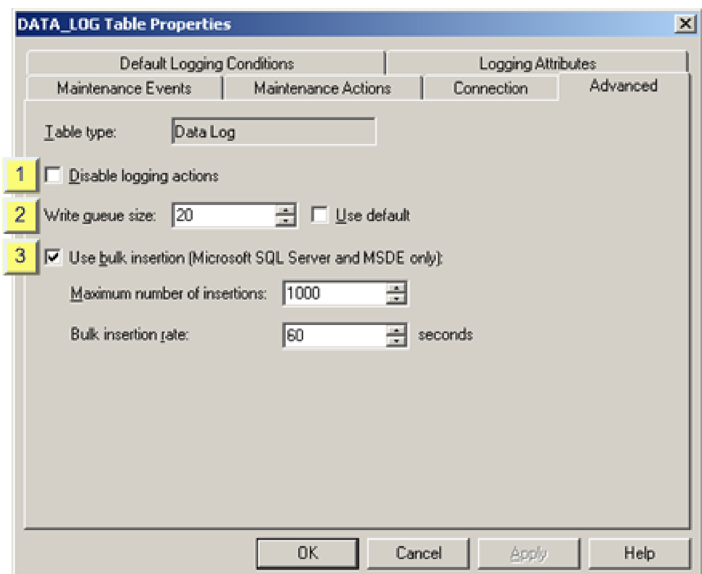
Step 2.7. Do advanced Logging Configuration

The Advanced tab in the data log's Table Properties dialog box enables you to:

- Disable logging actions for the table that you are configuring.
- Override the queue size that is specified in the Database Logger's Logging Properties dialog box.
- Select and control bulk insertion for Microsoft SQL Server or MSDE.

Select the Advanced tab in the data log's Table Properties dialog box.

Configuration options are as follows.



rect 1, 87, 169, 117 *(page 235)*
rect 0, 141, 298, 229 *(page 235)*
rect 0, 113, 266, 143 *(page 235)*

1 *(page 235)*	Disable logging actions
2 *(page 235)*	Write a queue size/Use default
3 *(page 235)*	Use bulk insertion (Microsoft SQL Server and MSDE only)

1	Disable logging actions

Select whether or not to log data to the table.

Do one of the following in the Disable logging actions checkbox.

Option	Description
Check	Do not log data to the selected table.
Clear	Log to the selected table when the CIMPLICITY project is running.

2	Write a queue size/Use default

Choose one of the following.

Option	Description
Use default	Check to use the universal size specified on the Parameters tab in the Database Logger's Logging Properties dialog box.
Write queue size field	1. Clear Use default. 2. Enter a number in the that supports the worst-case logging during activity bursts. Example: If 10 points are being logged to the table, you should have a queue size of at least 10 in case the points' logging conditions all occur simultaneously.

3	Use bulk insertion (Microsoft SQL Server and MSDE only)

Use bulk insertion enables you to take advantage of the Database Logger's bulk insertion capability (Microsoft SQL Server or MSDE only).

1. Check Use bulk insertion.
2. Select the Maximum number of insertions that your system can handle.
3. Select the interval between insertions in the **Bulk insertion rate** field.

The Database Logger will be triggered to insert records, based on whichever specification occurs first.

Example

You specify the:

Maximum number of insertions	=	1000
Bulk insertion rate	=	60 seconds

As a result:

If the Database Logger:	Then the Database Logger:
Collects 1000 records within 60 seconds.	Inserts the records into the log.
Does not collect 1000 records within 60 seconds.	Inserts whatever records have been collected.

Important: For SQL Server, insert triggers fire only if bulk insertion is disabled.

Step 3. Configure Logging Conditions for a Single Point

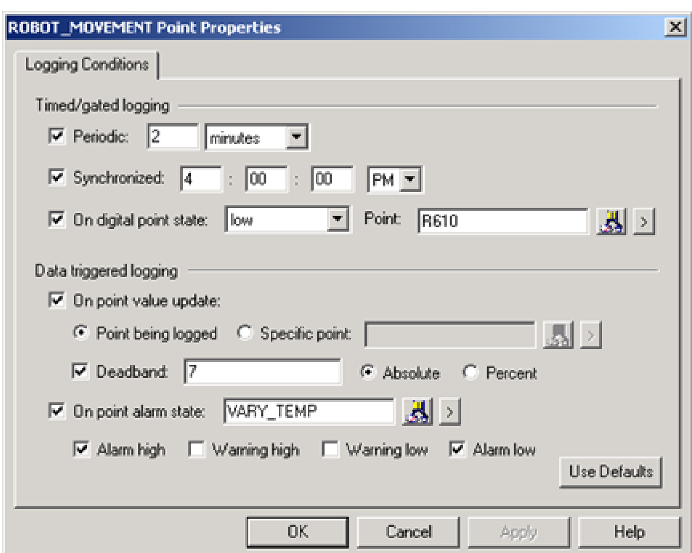
1. Open a CIMPLICITY data table's window.
2. Select the point you want to log under different conditions from the default.
3. Do one of the following.
   - Double-click the selected Point ID.
   - Click the Item Properties button on the window's toolbar.
   - Click Edit>Point Properties... on the Table window's menu bar.
   - Use the Popup menu.

   a. Click the right mouse button.

   b. Select Properties… on the Popup menu.

   Result: The (Database Logger's) Point Properties dialog box opens when you use any method.

   The Point Properties dialog box displays either the default logging conditions or conditions that were previously checked for the selected point.



4. Select the logging conditions that apply to the selected point.

   Options are included for:

Timed/gated logging
Data triggered logging.

5. Click **OK**.

The selected point will be logged when its logging conditions occur.

Note: If point data is being logged to Historian and the point's logging conditions configuration is changed in the CIMPLICITY Database Logger, the Historian Collection Options *(page 146)* will be changed for the tag.

Group Point Logging

Group Point Logging

Group Point Logging	Enables you to log data for a selected group of points in parallel.
Benefit	Gives better performance and uses disk space more efficiently than data logging
Default group point data log table	GROUP_LOG

Group Point logging provides you with a straightforward process for creating and maintaining records to analyze the performance of selected points, whose values and selected attributes are logged at the same time.

The point performance represents the actual performance of equipment and processes in your system. As a result, point logging provides you with an in depth record that can help you determine if action is required to improve the performance of any equipment or process in your system.

Steps to configure group logging include:

Step 1 *(page 239)*	Add points to a group log table.
Step 2 *(page 241)*	Configure group logging properties.
Step 3 *(page 248)*	Configure logging attributes for a single point.

Important: When using fully qualified points, you must provide remote project login configuration with the project.

Indexes, Columns, Rows in Group Tables

- Indexes in group data logging tables.
- Columns and rows in group data logging tables.

Indexes in group data logging tables

Indexes in Group logging tables include a:

- Unique primary key index on the timestamp column.
- Unique primary key index on the timestamp_utc column.

If you have selected the project name table attribute, the primary key index also includes the project name column.

The primary key includes an MSEC column, if milliseconds is enabled.

Columns and rows in group data logging tables

Columns and **rows** in Group logging tables are as follows.

Note: The prefix of each attribute column is based on the SQL Field Prefix specified for the point.

The index of each column is based on the point array element. Non array points always have index value of '0' (e.g. <prefix>_VAL0)

Column Name	Data Type	Description
Constant fields		
timestamp	date/time	Timestamp of the logging event
timestamp_utc	date/time	UTC timestamp of the logging event
Attribute fields		
<prefix>_ALRM<index>	string	Alarm state of the point .
<prefix>_ENG<index>	string	Engineering units label of the point.
<prefix>_PREV<index>	Depends on point type	Previous logged value of the point for the same logging event.
<prefix>_VAL<index>	<prefix>_VAL<index>	Value of the point.
<prefix>_TIME<index>	date/time	Previous time the point was logged for the same logging event.
<prefix>_TIME_UTC<index>	UTC date/time	Previous UTC time the point was logged for the same logging event.
<prefix>_RAW<index>	Depends on point type	Raw value of the point.
<prefix>_RES<index>	string	CIMPLICITY Resource associated with the point.

<prefix>_QUALITY<index>	number	Point quality flags.
<prefix>_USER<index>	number	Value of the user defined flags.
Optional fields		
project	string	Name of the CIMPLICITY project
msec	number	Actual number of milliseconds in the timestamp

Note: The maximum number of columns a table can have is SBMS specific. Consult your DBMS documentation to ensure that your Group table does not exceed these limits. Default MSDE DBMS is **1024** columns, **8060** Bytes per row.

Important: Since the Microsoft Access (As-Is product) and Oracle format does not support sub-second timestamp data, you cannot log points to a Group table at sub-second rates. Attempting to do so will cause the duplicate-keyed records to be dropped from the database, unless you have enabled millisecond logging.

Step 1. Add Points to a CIMPLICITY Group Table

You can add points to a group log through the Database Logger Configuration window and in a Group Log window.

Add points to a data log table through the:

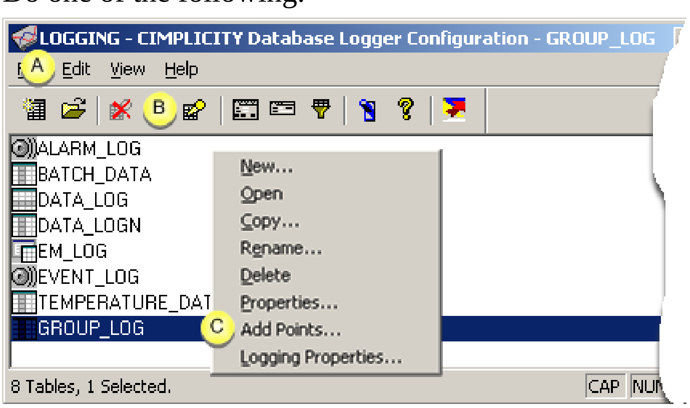
- Database Logger Configuration window.
- Group log table.

Database Logger Configuration window

1. Open the CIMPLICITY Database Logger Configuration window.

2. Select a GROUP_LOG table.

3. Do one of the following.



A	Click Edit>Add Points on the CIMPLICITY Database Logger Configuration window menu bar.
B	Click the Add Item button on the Database Logger Configuration window toolbar.
C	a. Right-click in the CIMPLICITY Database Logger Configuration window. b. Select Add Points... on the Popup menu.

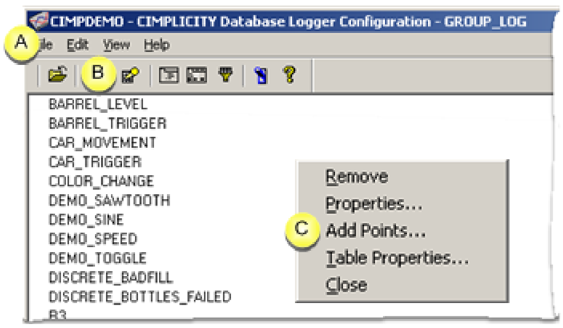
Group log table

4. Right-click in the CIMPLICITY Database Logger Configuration window.

5. Select Add Points... on the Popup menu.

6. Select a data log table.

7. Do one of the following.



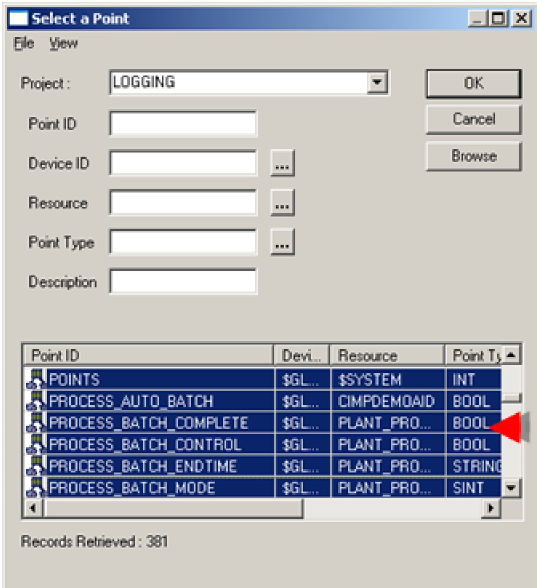
A	Click File>Add Points on the table menu bar.
B	Click the Add Item button on the log table toolbar.
C	a. Right-click in the log table window. b. Select Add Points… on the Popup menu.

The Select a Point browser opens when you use any method.

8. Right-click in the log table window.

9. Select Add Points… on the Popup menu.

10. Select the Point IDs you want to add.



11. Click **OK**.

The Point IDs display in the group data table window and will be logged according to your specifications.

Step 2. Configure Group Logging Properties

Step 2. Configure Group Logging Properties

You configure the point logging properties for all points in a in a CIMPLICITY group table's Table Properties dialog box. (You can adjust the logging conditions for a single point in the (Database Logger's) Point Properties dialog box.).

Step 2.1 (page 242)	Open the CIMPLICITY group Table Properties dialog box.
Step 2.2 (page 242)	Configure logging conditions.
Step 2.3 (page 243)	Configure logging attributes.
Step 2.4 (page 244)	Configure maintenance events.
Step 2.5 (page 245)	Configure maintenance actions.

Step 2.6 (page 246)	Configure the CIMPLICITY group point log connection.
Step 2.7 (page 247)	Configure advanced properties for a group log table.

Step 2.1. Open a Group Table Properties Dialog Box

1. Either:
   - Select a group table (e.g., GROUP_LOG) in the Database Logger Configuration window, or
   - Open the group (page 184) table (e.g., GROUP_LOG) window..

2. Do one of the following.

   Method 1

   Click the **Table Properties** button on either window's toolbar.

   Method 2

   a. Click Edit on either window's menu bar.
   b. Select Table Properties…

   Method 3

   a. Right-click a group table in the Database Logger Configuration window or any Point ID in the group table's window.
   b. Select Table Properties… from the popup menu.

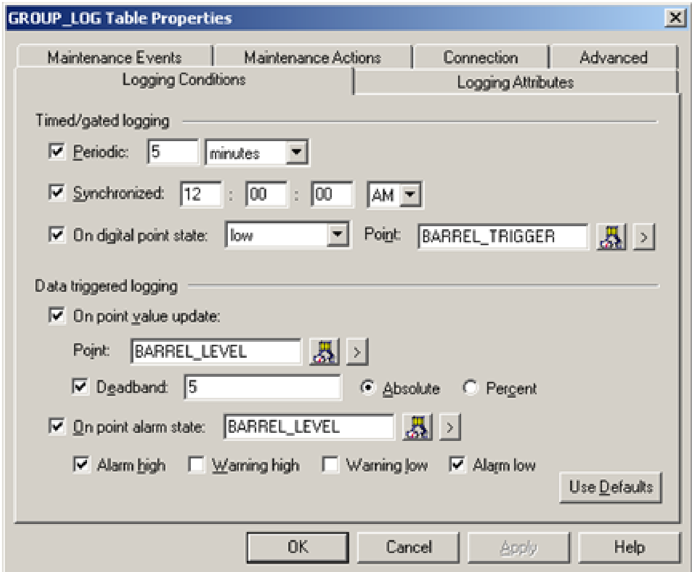
   Result. The group table's (e.g., GROUP_LOG) Table Properties dialog box opens when you use any method.

Step 2.2. Configure Group Logging Conditions

Logging Conditions determine when data will be logged into the table

Note: Because, the points in a group table are logged in parallel, the logging conditions are the same for all. However, you can select attributes that will be logged for individual points.

1. Select the Logging Conditions tab.



The conditions you check define the conditions for when the values of all the points in the table will be logged.

2. Specify logging conditions the same way you specify them for a data log *(page 214)* table.
3. Check **On point alarm state** to log point values when a specified point goes into one or more selected alarm states.

Step 2.3. Configure Group Logging Attributes

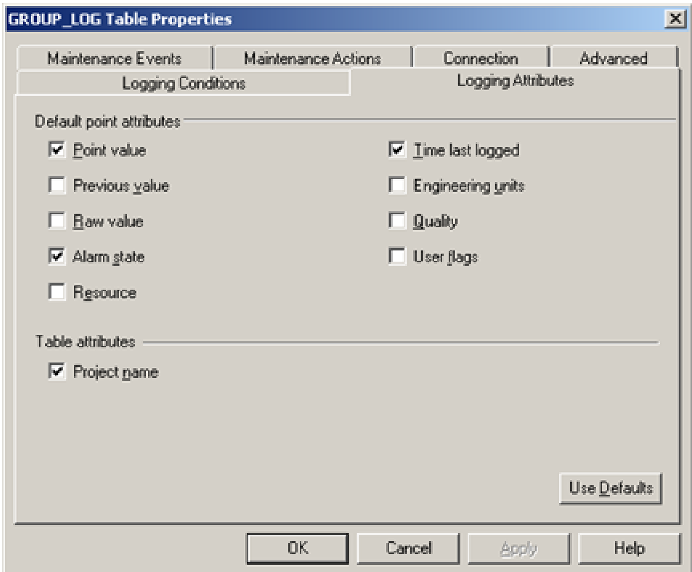
Logging Attributes for a group table (e.g. GROUP_LOG) enables you to:

- Select the current point attributes that will be logged to the table.
- Log the name of the CIMPLICITY project.:

The checked logging attributes will be the default logging attributes for additional points you add to the group table.

Note: You can also specify that selected attributes be logged for individual points in the group table.

1. Select the Logging Attributes tab.



2. Check any of the following point attributes to log:

Point Attribute	Log the:
Point Value	Point's current value.
Previous Value	Previous value recorded in this table for this point.
Raw Value	Point's raw value.
Alarm State	Point's alarm state when the value is being logged.
Resource	Point's Resource ID.
Time Last Logged	Time the point was last logged in this table.
Engineering Units	Point's engineering units.
Quality	Is the value of the QUALITY attribute.
User Flags	Value of the USER_FLAGS attribute.
Table attributes	
Project name	Project name.

Tip: Click **Use Defaults** to reset the Logging Attributes to their default values. The defaults are to:

- Log **Point** and
- Not log the **Project Name**.

Note: This also applies to BATCH_DATA attributes.

Step 2.4. Configure Group Maintenance Events

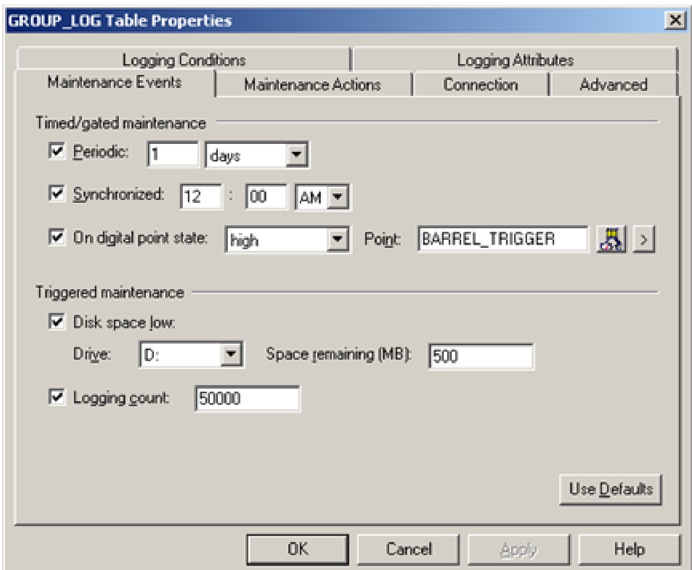
Maintenance Events define when export, purge, and command actions will be performed on the group table.

You can specify that maintenance be triggered when:

- A condition, such as low disk space, occurs or
- On a periodic or gated interval

Any specified event will trigger maintenance actions.

1. Select the Maintenance Events tab in a data Table Properties dialog box.



2. Specify actions the same way you specify them for a data log *(page 224)* table.

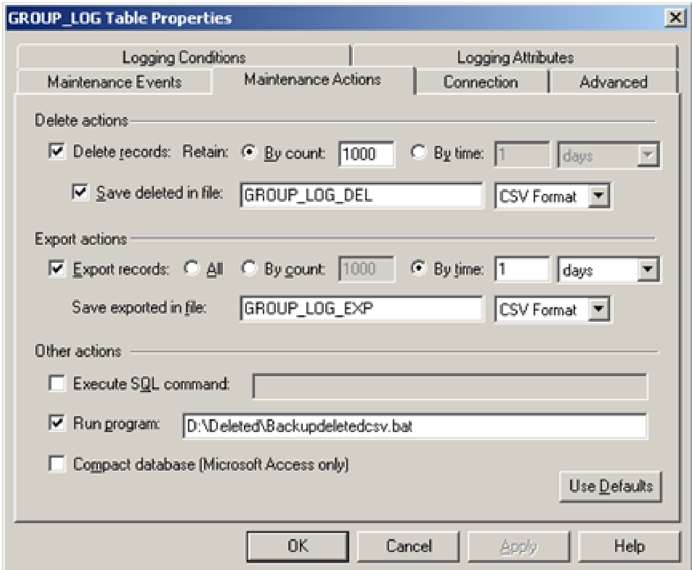
Step 2.5. Configure Group Maintenance Actions

Maintenance Actions define the type of actions that take place when a Maintenance Event occurs.

You can configure a group table to:

- Export records to an export file,
- Purge records from a logging table and/or
- Incorporate command line actions.

1. Select the Maintenance Actions tab in the group log Table Properties dialog box.



2. Specify actions the same way you specify them for a data log *(page 227)* table.

Step 2.6. Configure the Group Connection

Connection defines the database connection and the Store and Forward properties for the table.

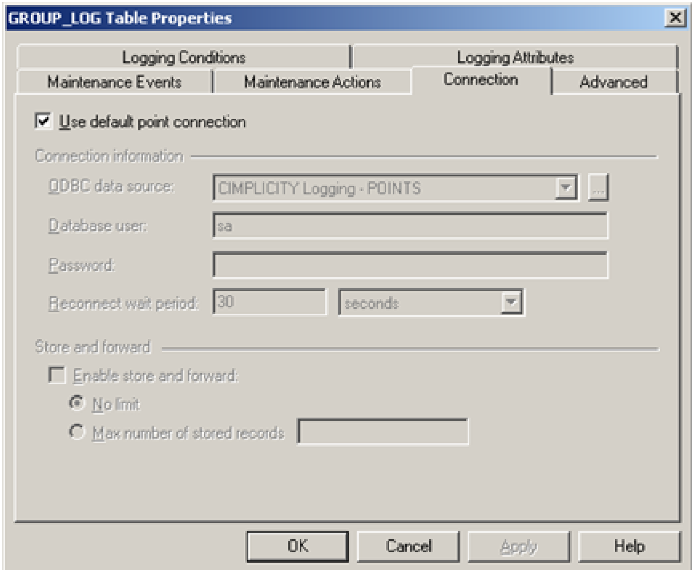
Do one of the following.

- Use the default connection.
- Specify logging connection specific to the GROUP_LOG table.

Use the default connection

1. Select the Connection tab in the GROUP_LOG Table Properties dialog box.

2. Check Use default point connection.

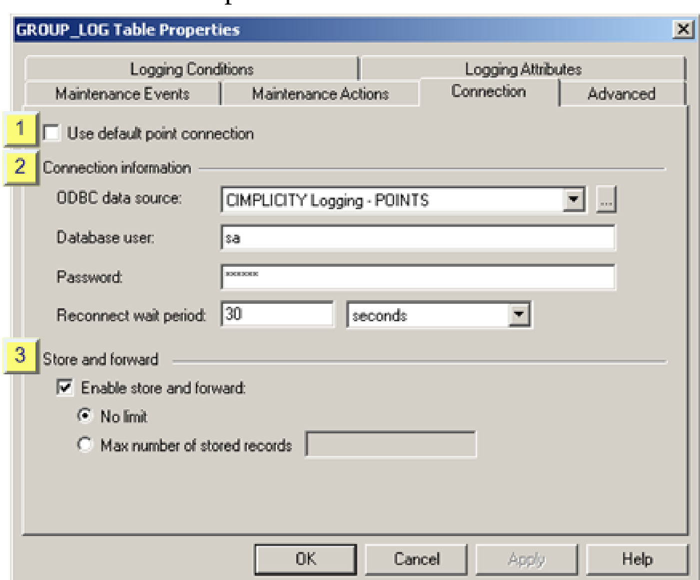


3. Do default connection configuration the same way you do for a data log *(page 229)* table.

   Specify logging connections specific to the GROUP_LOG table

4. Select the Connection tab in the GROUP_LOG Table Properties dialog box.

5. Clear Use default point connection.



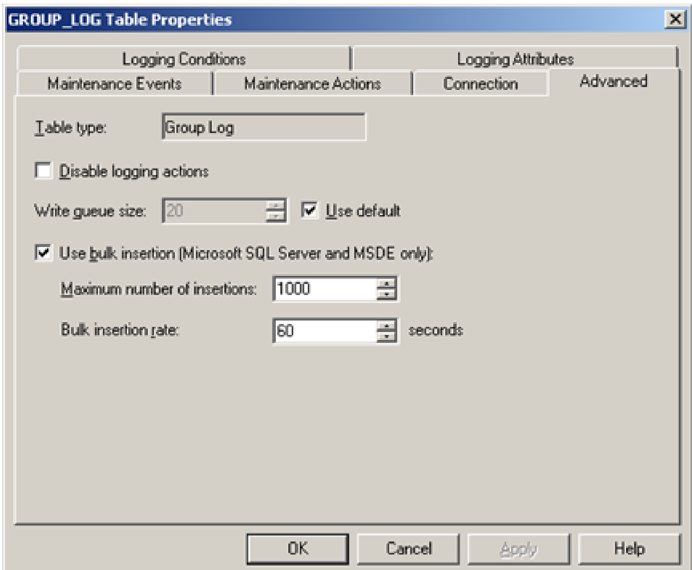
6. Do custom connection configuration the same way you do for a data log *(page 230)* table.

Step 2.7. Do Advanced Group Log Configuration

The Advanced tab in the group log's Table Properties dialog box enables you to:

- Disable logging actions for the table that you are configuring.
- Override the queue size that is specified in the Database Logger's Logging Properties dialog box.
- Select and control bulk insertion for Microsoft SQL Server or MSDE.

1. Select the Advanced tab in the group log's Table Properties dialog box.



2. Do advanced configuration the same way you do for a data log *(page 227)* table.

Important: For SQL Server, insert triggers fire only if bulk insertion is disabled.

Step 3. Configure Logging Attributes for a Single Point in a Group Table

1. Open a group table (e.g. GROUP_LOG) window.
2. Select the point for which you want to log different attributes from the default.
3. Do one of the following.

   Method 1

   Double-click the selected Point ID.

   Method 21

   Click the Item Properties button on the group log table toolbar.

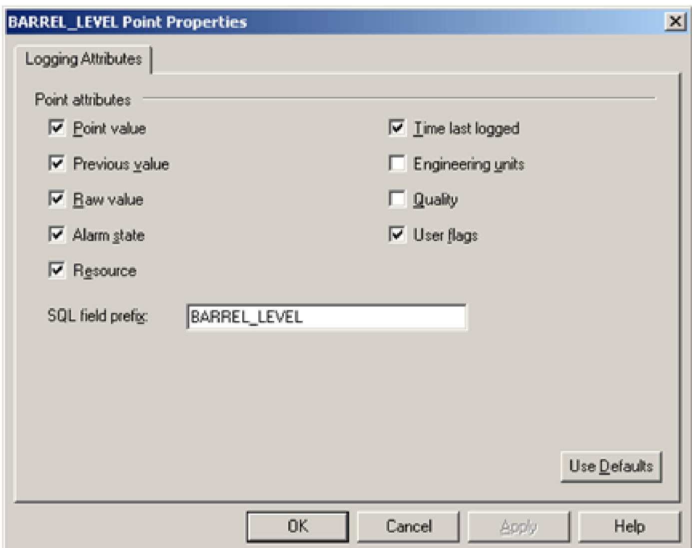
   Method 3

a. Click Edit on the window's menu bar for the group log table.
b. Select Point Properties…

Method 4

a. Click the right mouse button.
b. Select Properties… from the popup menu.

The (Database Logger's) Point Properties dialog box opens when you use any method displaying either the default logging conditions or conditions that were previously checked for the selected alarm.



4. Check the attributes that you want logged for the selected point.

5. Click **OK**.

The attributes you select will be logged for the selected point. The selection does not change if you change the table defaults.

Alarm Logging

Alarm Logging

Alarm logging provides you with a straightforward process for creating and maintaining records to analyze the alarm state of equipment and processes in your system. As a result, alarm logging helps you determine if action is required to repair or maintain your system.

The Database Logger enables you to:

- Log data about selected alarms or all alarms to an alarm log table.

- Save the log files, until specified criteria occur.
- Have the logged data discarded, based on specified criteria.
- Save the logged data to another file for future analysis.

You can display, print and analyze that information in a report that you create or in an Excel CIMPLICITY Alarm Report that is included in CIMPLICITY..

Note: Data logged for alarms includes:

- Alarm ID,
- Time the Alarm was Generated,
- Alarm Class, Resource,
- Alarm Message,
- Alarm comments that are entered in the Alarm Viewer.
- Logged By
- Other optional attributes that you select.

Steps to configure alarm logging include:

Step 1 *(page 250)*	Review system alarms included in the alarm log table.
Step 2 *(page 252)*	Add additional alarms to the alarm log table.
Step 3 *(page 255)*	Configure alarm logging properties.
Step 4 *(page 264)*	Configure logging conditions for a single alarm.

Note: You cannot copy or create a second ALARM_LOG.

Review Logging Conditions for a Single Alarm

Review configuration for a single CIMPLICITY:

Alarm.
Event alarm.

Step 1. Review System Alarms in the ALARM_LOG

Step 1. Review System Alarms in the ALARM_LOG

The following system alarms are recorded in the Alarm Log table:

Alarm ID	Description
$DEVICE	Specified device has a problem.
$DEVICE_DOWN	Specified device has failed.
$DEVICE_FAILOVER	Device failure for Server Redundancy.
$REDUND_DEV_DOWN	Redundant device down for Server Redundancy.
AMSI_ALARM	Reserved for future use.
DB_CONN_DOWN	Connection lost to specified database for Database Logger.
DB_START_FORWARD	Forwarding files found for Database Logger Store and Forward.
MCP_PROC_DOWN	Specified CIMPLICITY process has terminated unexpectedly.

Indexes, Columns, Rows in the ALARM_LOG Table

Indexes in Alarm logging tables include a:

- Primary key index on the joined timestamp and sequence number columns.
- Secondary key index on the joined timestamp_utc and sequence number columns.
- Secondary index on the timestamp alone.
- Secondary index on the timestamp_utc alone.

If you have selected the project name table attribute, the primary key index also includes the project name column.

The primary key includes an MSEC column, if milliseconds is enabled.

Columns and **rows** in Alarm logging tables are as follows:

Column Name	Data Type	Description
Constant fields		
timestamp	date/time	Timestamp of the logging event.
timestamp_utc	date/time	UTC timestamp of the logging event.
sequence_number	number	Identity column to ensure uniqueness.
alarm_id	string	CIMPLICITY alarm identifier.
alarm_class	string	CIMPLICITY alarm class associated with the alarm.
resource	string	CIMPLICITY resource associated with the alarm.
logged_by	string	CIMPLICITY process that logged the alarm.
reference	string	Reference information for the alarm logged.

prev_state	string	Previous state of the alarm. Possible values.	
		N	Normal
		G	Generate
		R	Reset
		A	Acknowledge
		D	Delete
log_action	string		State of the alarm that caused log event. Possible values.
		N	Normal
		G	Generate
		R	Reset
		A	Acknowledge
		D	Delete
final_state	string		Final state of the alarm. Possible values N=G=R=A= D=
		N	Normal
		G	Generate
		R	Reset
		A	Acknowledge
		D	Delete
alarm_message	string	Message associated with the alarm.	
generation_time	date/time	Timestamp the alarm was generated.	
generation_time_utc	UTC date/time	UTC timestamp the alarm was generated.	
Optional fields			
project	string	Name of the CIMPLICITY project.	
msec	number	Actual number of milliseconds in the timestamp.	

Step 2. Add Point Alarms and Alarms to the ALARM_LOG Table

Step 2. Add Point Alarms and Alarms to the ALARM_LOG Table

You can add the following to the CIMPLICITY alarm log table:

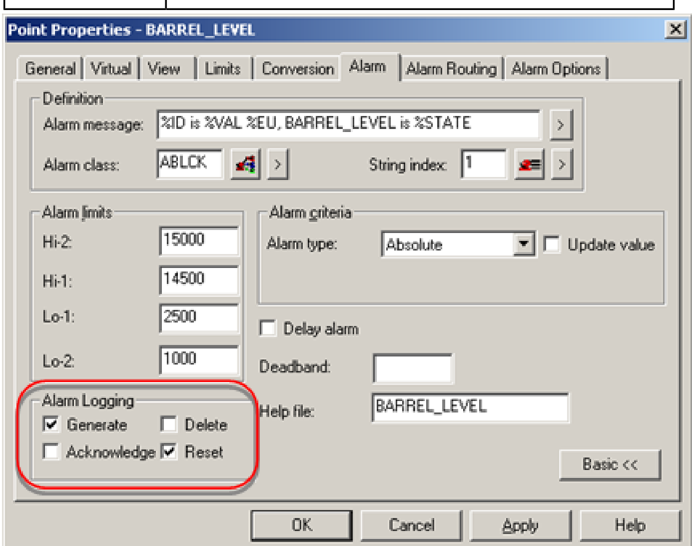
Option 2.1 *(page 253)*	Add Point Alarms to the ALARM_LOG Table via a Point Properties dialog box.

Option 2.2 (page 253)	Add Alarms to ALARM_LOG through the Database Logger.

Option 2.1. Add Point Alarms to the ALARM_LOG Table via a Point Properties Dialog Box

1. Select **Points** in the Workbench left pane.
2. Select the point for which you want to log alarms in the Workbench right pane.
3. Open the point's Properties dialog box.
4. Select the Alarm tab.
5. Check if and when you want an alarm to be logged in the Alarm Logging box on the Alarm tab of the Point Properties dialog box. You can select any or all of the options:

Condition	Alarm data is logged when the alarm:
Generate	Occurs.
Acknowledge	Is acknowledged by the operator or the system.
Reset	Is reset by the operator or the system.
Delete	Is manually deleted from the system.



6. Click **OK** or **Apply**.

CIMPLICITY adds the point to its default ALARM_LOG. You can apply more specifications in the Database Logger.

Option 2.2. Add Alarms to ALARM_LOG through the Database Logger

1. Do one of the following in the Database Logger Configuration window or the ALARM_LOG window.

   In the Database Logger Configuration window

   a. Select ALARM_LOG
   b. Do one of the following:

   Method 1

   Click the **Add Item** button  on the Database Logger Configuration window toolbar.

   Method 2

   a. Click Edit on the Database Logger Configuration window menu bar.
   b. Select Add Alarms…

   Method 3

   a. Right-click ALARM_LOG.
   b. Select Add Alarms… from the popup menu.

   In the ALARM_LOG window

   Do one of the following.

   Method 1

   Click the **Add Item** button  on the ALARM_LOG toolbar.

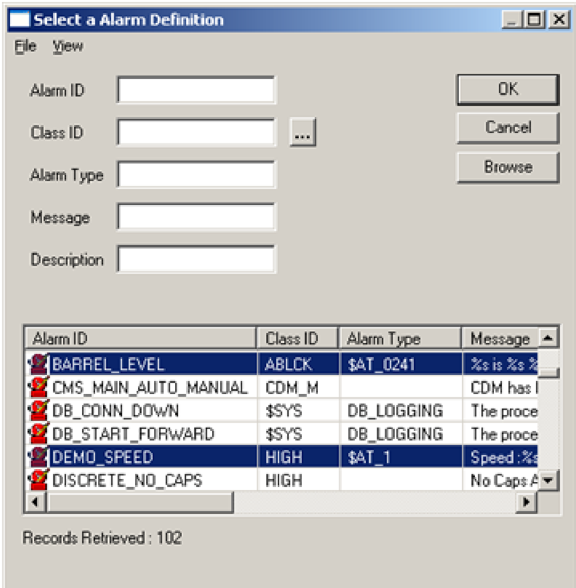
   Method 2

   a. Click File on the ALARM_LOG window menu bar.
   b. Select Add Alarms…

   Method 3

   a. Right-click in the ALARM_LOG window.
   b. Select Add Alarms… from the popup menu.

   The Select an Alarm Definition browser opens when you use any method.



2. Select the Alarm IDs you want to add.

3. Click **OK**.

The alarms display in the ALARM_LOG window and will be logged according to your specifications.

Step 3. Configure Alarm Logging Properties

Step 3. Configure Alarm Logging Properties

You configure the alarm logging properties for all included alarms in the ALARM_LOG Table Properties dialog box. (You can adjust the logging conditions for a single alarm in the Alarm Properties dialog box.).

Step 3.1 *(page 256)*	Open the ALARM_LOG Table Properties dialog box.
Step 3.2 *(page 256)*	Configure default logging conditions.
Step 3.3 *(page 257)*	Configure logging attributes.
Step 3.4 *(page 260)*	Configure maintenance events.

Step 3.5 (page 261)	Configure maintenance actions.
Step 3.6 (page 262)	Configure the ALARM_LOG connection.
Step 3.7 (page 263)	Configure advanced properties for the ALARM_LOG table.

Step 3.1. Open the ALARM_LOG Table Properties Dialog Box

1. Either:
    - Select ALARM_LOG in the Database Logger Configuration window, or
    - Open *(page 184)* the ALARM_LOG window.

2. Do one of the following.

    Method 1

    Click the **Table Properties** button on either window's toolbar.

    Method 2

    a. Click Edit on either window's menu bar.
    b. Select Table Properties…

    Method 3

    a. Right-click ALARM_LOG in the Database Logger Configuration window or any alarm in the ALARM_LOG window.
    b. Select Table Properties… from the popup menu.

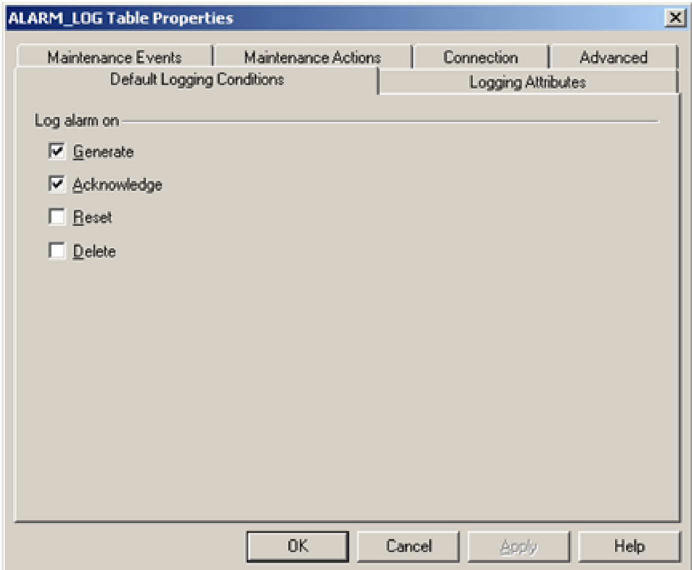
    Result. The ALARM_LOG Table Properties dialog box opens when you use any method.

Step 3.2. Configure Default Alarm Logging Conditions

Logging Conditions determine when data will be put into the log table. .

Note: You can change the conditions you specify in the ALARM_LOG Table Properties for a single alarm.

1. Select the Default Logging Conditions tab.



The conditions you check define the default conditions for when the alarms in the table will be logged.

2. Modify the existing default conditions, if required.

   The conditions are:

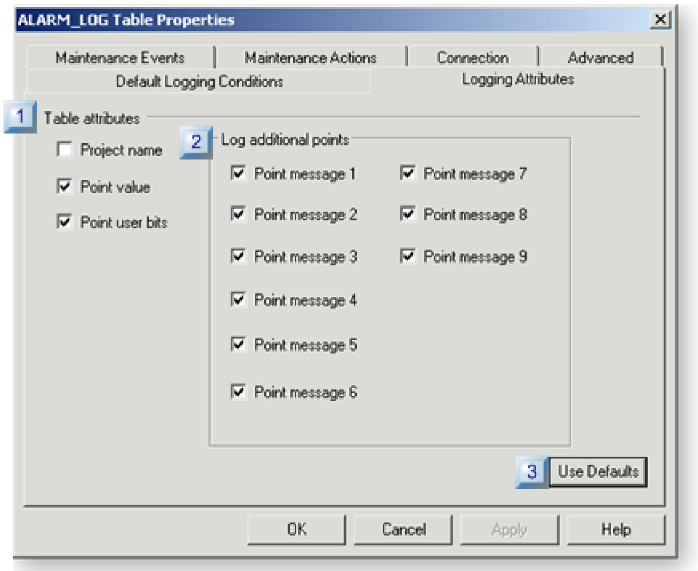
Condition	Alarm data is logged when the alarm:
Generate	Occurs.
Acknowledge	Is acknowledged by the operator or the system.
Reset	Is reset by the operator or the system.
Delete	Is manually deleted from the system.

Step 3.3. Configure Alarm Logging Attributes

Logging Attributes for an ALARM_LOG table enables you to log additional information about the logged alarms.

Select the Logging Attributes tab in the Table Properties dialog box.

Options are as follows.



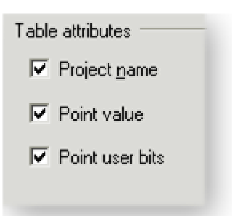
rect -1, 57, 23, 84 *(page 258)*
rect 108, 73, 132, 100 *(page 259)*
rect 319, 279, 343, 306 *(page 260)*

1 *(page 258)*	Table attributes.
2 *(page 259)*	Log addition points.
3 *(page 260)*	Use defaults.

1	Table attributes

Check the table attributes that should be included in the ALARM_LOG table.

Note: These attributes can also be included in the DATA_LOG



Attribute	SQL Field	Description
Project name	project	Project the point is in.

Point value	point_val	Point value of the alarm being logged.			
Point user bits	user_bits	The value of the point's 64 bit user set usage. Even though CIMPLICITY is a 32 bit application a point can access the lowest 32-bits and the highest 32 bits in a 64 bit set. The logged value: • Displays the combined 64 bit value. • Is logged as a decimal.			
		Example A point writes the following values to EXTENDED_USER_FLAGS_HIGHand EXTENDED_USER_FLAGS_LOW.			
		Bit Set	Decimal	=	Hex
		HIGH	35	=	0x23
		LOW	15	=	0xf
		CIMPLICITY combines the EXTENDED_USER_FLAGS_HIGH and EXTENDED_USER_FLAGS_LOW.			
		The HEX value is:			0x000000230000000f
		Converted to Decimal:			150323855375
		Result: CIMPLICITY logs 150323855375 to the ALARM_LOG in SQL.			

Note: If you select or de-select the project name table attribute you must drop the table so that it is properly re-created with or without the new project field as part of the primary key. Failing to do so can cause the database logger to fail to log data into the table.

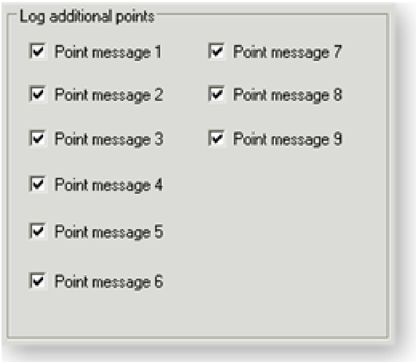
2	Log additional points

The status of any component in a system may be affected by the operation of other components in that system.

Therefore, if a point is in alarm state, even if related points are not in alarm state, another value or combination of values may be causing or influencing the alarm state value.

To help analyze an alarm status, CIMPLICITY enables you to log additional point values in the SQL table row that contains the selected point's alarm state information.

Each point message checkbox that is checked adds two columns to the ALARM_LOG table, as follows.



Checkbox	Column 1	Column 2
Point message <n>	point_id_<n>	point_val_<n>

- When a point's alarm messageincludes related point values, CIMPLICITY enters the point ID and value in the fields that are assigned in the alarm message.
- If a Point message <n> is checked, but is not included in a point's alarm message, there is no entry (e.g. NULL) for that point in the ALARM_LOG.

3	Use Defaults

Click Use Defaults to return the ALARM_LOG to its default logging attribute settings.

The default logging attribute settings are as follows.

- The first time you open the ALARM_LOG all check boxes are clear.

When you click Use Default, all check boxes are once again cleared.

- The next time you open the ALARM_LOG the settings that were selected the last time is was closed become the default settings

When you click Use Default, check boxes that were:

- Checked when the previous session closed, are checked.
- Clear when the previous session closed, are clear.

Step 3.4. Configure Alarm Log Maintenance Events

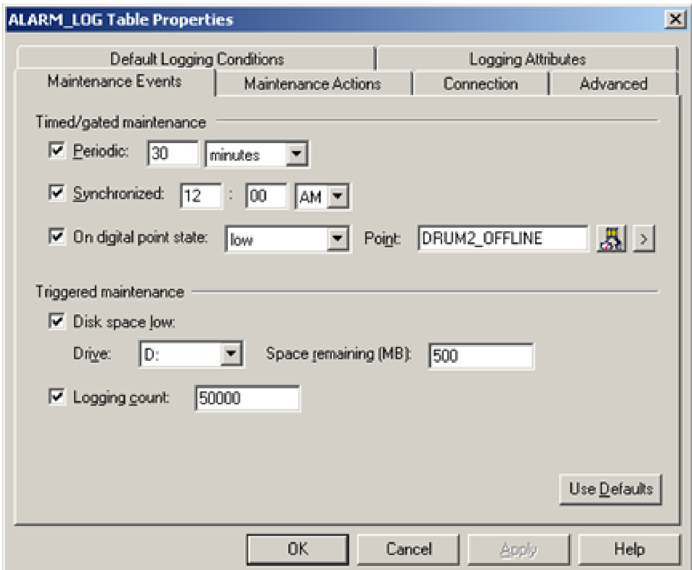
Maintenance Events define when export, purge, and command actions will be performed on the Alarmtable.

You can specify that maintenance be triggered when:

- A condition, such as low disk space, occurs or
- On a periodic or gated interval

Any specified event will trigger maintenance actions.

1. Select the Maintenance Events tab in a alarm Table Properties dialog box.



2. Specify actions the same way you specify them for a data log *(page 224)* table.

Important: (For SQL Server) insert triggers fire only if bulk insertion is disabled.

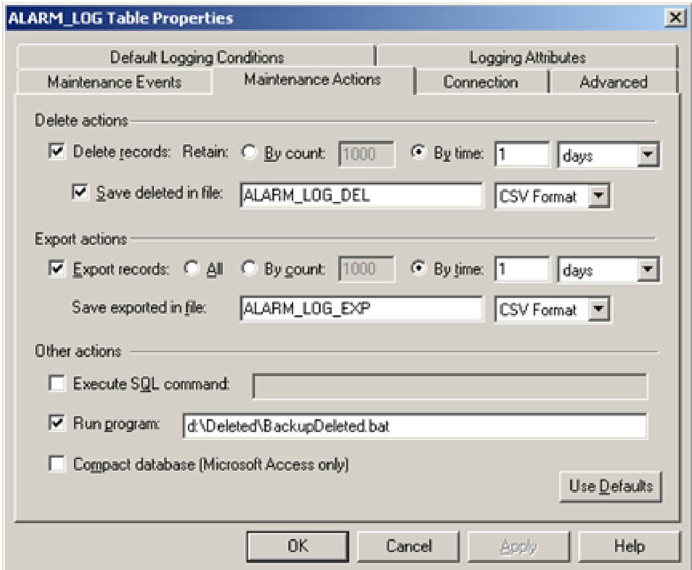
Step 3.5. Configure Alarm Log Maintenance Actions

Maintenance Actions define the type of actions that take place when a Maintenance Event occurs.

You can configure the ALARM_LOG table to:

- Export records to an export file,
- Purge records from a logging table and/or
- Incorporate command line actions.

1. Select the Maintenance Actions tab in the ALARM_LOG Table Properties dialog box.



2. Configure the maintenance actions the same way you configure maintenance actions for a CIMPLICITY data *(page 227)* table.

Step 3.6. Configure the Alarm Log Connection

Connection defines the database connection and the Store and Forward properties for a table.

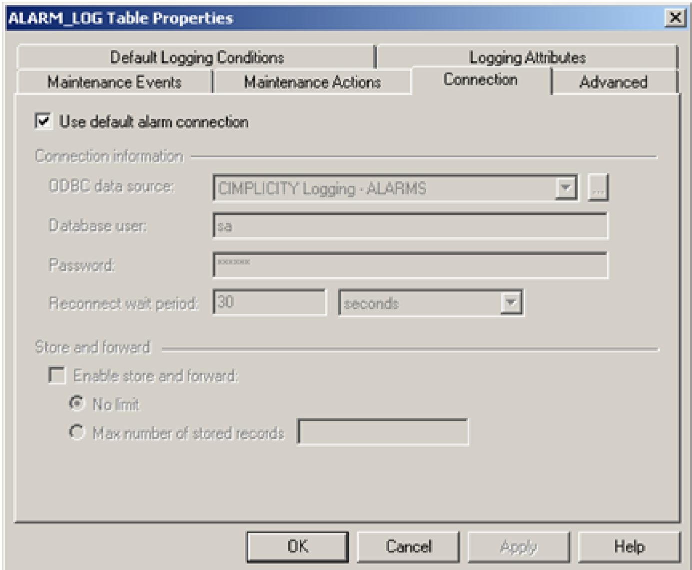
Do one of the following.

- Use the default connection.
- Specify logging connection specific to the ALARM_LOG table.

Use the default connection

1. Select the Connection tab in the ALARM_LOG Table Properties dialog box.

2. Check Use default point connection.

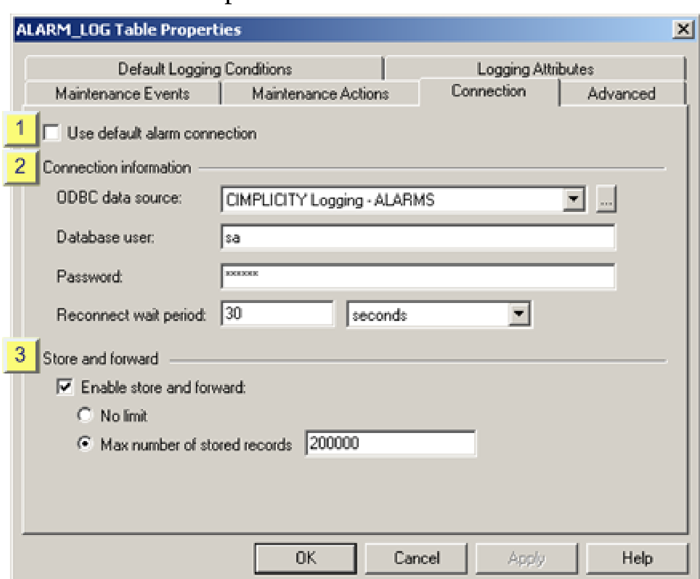


3. Do default connection configuration the same way you do for a data log *(page 229)* table.

   Specify logging connections specific to the ALARM_LOG table

4. Select the Connection tab in the ALARM_LOG Table Properties dialog box.

5. Clear Use default point connection.



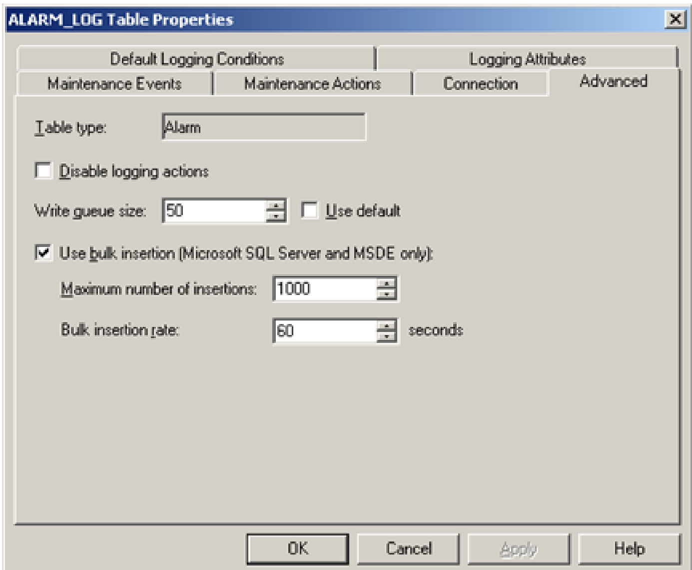
6. Do custom connection configuration the same way you do for a data log *(page 230)* table.

Step 3.7. Do Advanced Alarm Logging Configuration

The Advanced tab in the ALARM_LOG Table Properties dialog box enables you to:

- Disable all logging actions for the table that you are configuring
- Override the queue size that is specified in the Database Logger's Logging Properties dialog box.
- Select and control bulk insertion for Microsoft SQL Server or MSDE.

1. Select the Advanced tab in the ALARM_LOG Table Properties dialog box.



2. Do advanced configuration the same way you do for a DATA_LOG *(page 229)* table.

Important: For SQL Server, insert triggers fire only if bulk insertion is disabled.

Step 4. Configure Logging Conditions for a Single Alarm

1. Select the alarm you want to log under different conditions from the default.

2. Do one of the following.

Method 1

Double-click the selected Alarm ID.

Method 2

Click the Item Properties button  on the ALARM_LOG toolbar.

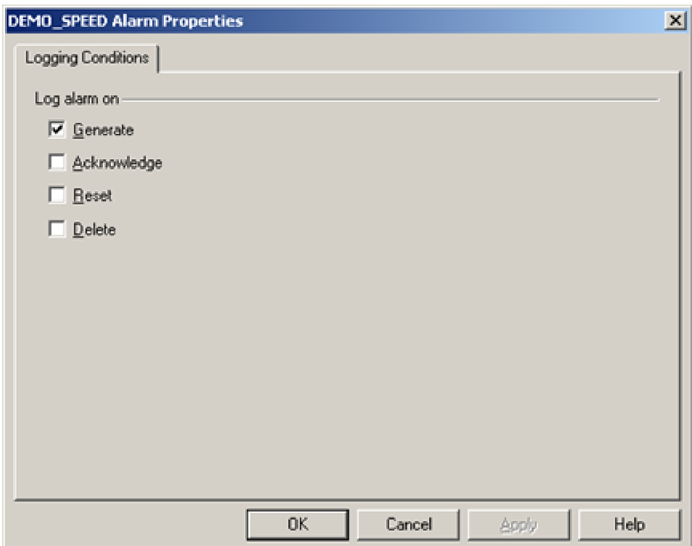
Method 3

a. Click Edit on the ALARM_LOG menu bar.
b. Select Alarm Properties…

Method 4

a. Click the right mouse button.
b. Select Properties… from the popup menu.

The Alarm Properties dialog box opens when you use any method displaying either the default logging conditions or conditions that were previously checked for the selected alarm.



3. Check the logging conditions that apply to the selected alarm.

4. Click **OK**.

The selected alarm will be logged in the ALARM_LOG when its logging conditions occur.

Event Alarm Logging

Event Alarm Logging

Event Alarm Logging enables you to log alarms for events that do not need to be acknowledged or reset. These events are normally system-based events, such as logging in and logging out. They do not appear in the Alarm Viewer. As a result, Event Logging is a convenient way for you to track a system issue that you are concerned about, but for which you do not want to have alarms appear in the Alarm Viewer where acknowledgement and resetting issues arise. :

The Database Logger enables you to:

- Log data about the specially selected alarms to an ALARM_LOG table.
- Save the log files, until specified criteria occur.
- Have the logged data discarded, based on specified criteria.

- Save the logged data to another file for future analysis.

You can display, print and analyze that information in a report that you create or in an Excel CIMPLICITY Alarm Report that is included in CIMPLICITY.

Steps to configure event alarm logging include:

Step 1 *(page 250)*	Review system alarms included in the EVENT_LOG table.
Step 2 *(page 267)*	Add event alarms through the Database Logger.
Step 3 *(page 269)*	Configure event alarm logging properties.
Step 4 *(page 275)*	Configure logging conditions for a single event alarm.

Note: You cannot copy or create a second EVENT_LOG.

Step 1. Review System Alarms in the EVENT_LOG

Step 1. Review System Alarms in the EVENT_LOG

The following system events are recorded in the Event Log table:

Event ID	Description
$ALARM_DISABLED	Alarming for specified point is disabled.
$ALARM_ENABLED	Alarming for specified point is enabled.
$ALARM_MODIFIED	Alarm limits for specified point modified by user.
$ALARM_RESTORED	Alarm limits for specified point restored to original values.
$AM_STATUS	Alarm Management throughput statistics. Issued every half-hour.
$DL_FILE_FULL	Reserved for future use.
$DOWNLOAD	Setpoint was downloaded to a specified point.
$DYN_CFG	Dynamic configuration enable/disable has been performed by a specified user.
$LOGIN_DISABLED	Specified use failed to log in to the specified CIMPLICITY project.
$LOGON	Specified user has logged in to a specified CIMPLICITY project.
$LOGOUT	Specified user has logged out of a specified CIMPLICITY project.
$RTR_LINK_DOWN	Router link has been lost to specified node.

DEMO_SPEED	Carwash Demo speed alarm is generated.

Indexes, Columns, Rows in the EVENT_LOG Table

Indexes in the EVENT_LOG include a:

- Primary key index on the joined timestamp and sequence number columns.
- Secondary key index on the joined timestamp_utc and sequence number columns.
- Secondary index on the timestamp alone.
- Secondary index on the timestamp_utc alone.

If you have selected the project name table attribute, the primary key index also includes the project name column.

The primary key includes an MSEC column, if milliseconds is enabled.

Columns and **rows** in Event logging tables are as follows:

Column Name	Data Type	Description
Constant fields		
timestamp	date/time	Timestamp of the logging event.
timestamp_utc	date/time	UTC timestamp of the logging event.
sequence_number	number	Identity column to ensure uniqueness.
alarm_id	string	CIMPLICITY alarm identifier.
alarm_class	string	CIMPLICITY alarm class associated with the alarm.
resource	string	CIMPLICITY resource associated with the alarm.
logged_by	string	CIMPLICITY process that logged the alarm.
reference	string	Reference information for the alarm logged.
alarm_message	string	Message associated with the alarm.
generation_time	date/time	Timestamp the alarm was generated.
generation_time_utc	UTC date/time	UTC timestamp the alarm was generated.
Optional fields		
project	string	Name of the CIMPLICITY project
msec	number	Actual number of milliseconds in the timestamp

Step 2. Add Event Alarms through the Database Logger

1. Do one of the following in the Database Logger Configuration window or the EVENT_LOG window.

In the Database Logger Configuration window

a. Select EVENT_LOG
b. Do one of the following:

Method 1

Click the **Add Item** button  on the Database Logger Configuration window toolbar.

Method 2

a. Click Edit on the Database Logger Configuration window menu bar.
b. Select Add Alarms…

Method 3

a. Right-click EVENT_LOG.
b. Select Add Alarms… from the popup menu.

In the EVENT_LOG window

Do one of the following.

Method 1

Click the **Add Item** button  on the EVENT_LOG toolbar.

Method 2

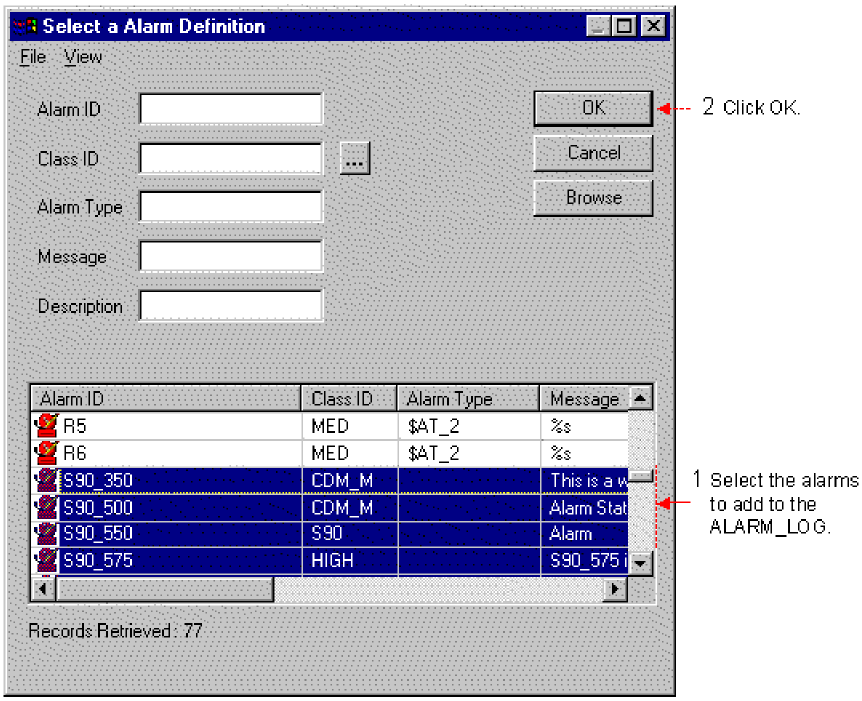
a. Click File on the EVENT_LOG menu bar.
b. Select Add Alarms…

Method 3

a. Right-click in the EVENT_LOG window.
b. Select Add Alarms… from the popup menu.

The Select an Alarm Definition browser opens when you use any method.

Alarm Browser Example



2. Select the Alarm IDs you want to add.

3. Click **OK**.

The alarms display in the EVENT_LOG window and will be logged according to your specifications.

Step 3. Configure Event Alarm Logging Properties

Step 3. Configure Event Alarm Logging Properties

You configure the event logging properties for all alarms in the EVENT_LOG Table Properties dialog box. (You can adjust the logging conditions for a single alarm in the Alarm Properties dialog box.). .).

Step 3.1 *(page 270)*	Open the EVENT_LOG Table Properties dialog box.
Step 3.2 *(page 270)*	Configure event logging attributes.
Step 3.3 *(page 271)*	Configure EVENT_LOG maintenance events.

Step 3.4 *(page 272)*	Configure EVENT_LOG maintenance actions.
Step 3.5 *(page 273)*	Configure the EVENT_LOG connection.
Step 3.6 *(page 274)*	Advanced event logging configuration.

Step 3.1. Open the EVENT_LOG Table Properties Dialog Box

1. Either:
   - Select EVENT_LOG in the Database Logger Configuration window, or
   - Open *(page 184)* the EVENT_LOG window.

2. Do one of the following.

   Method 1

   Click the **Table Properties** button  on either window's toolbar.

   Method 2

   a. Click Edit on either window's menu bar.
   b. Select Table Properties…

   Method 3

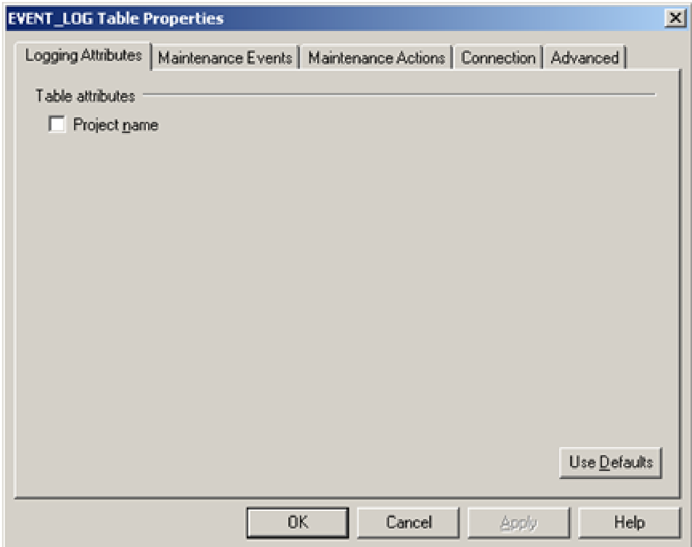
   a. Right-click EVENT_LOG in the Database Logger Configuration window or any alarm in the EVENT_LOG window.
   b. Select Table Properties… from the popup menu.

   Result. The EVENT_LOG Table Properties dialog box opens when you use any method.

Step 3.2. Configure Event Logging Attributes

Logging Attributes for a CIMPLICITY EVENT_LOG table enables you to log the name of the CIMPLICITY project.

1. Select the Logging Attributes tab in the EVENT_LOG Table Properties dialog box.



2. Check **Project name** to log the project name.

 Tip: Click **Use Defaults** to set the logging attributes to their default value. The EVENT_LOG will be set to not log the **Project Name**.

 Warning: If you select or de-select the project name table attribute you must drop the table so that it is properly recreated with or without the new project field as part of the primary key. Failing to do so can cause the database logger to fail to log data into the table.

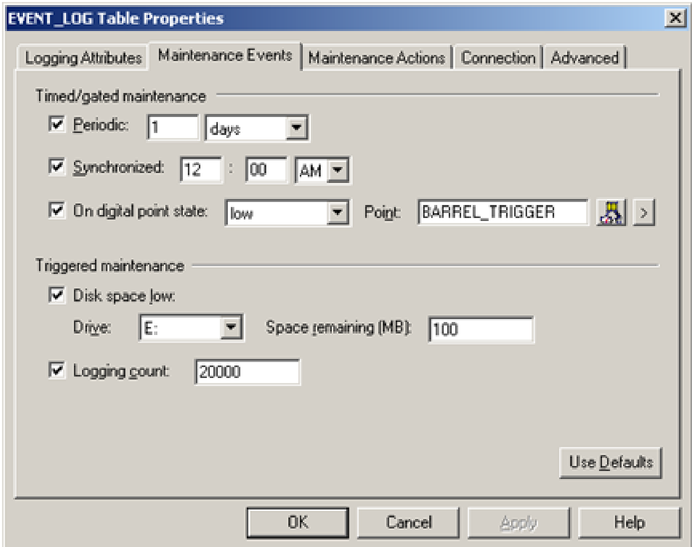
Step 3.3. Configure EVENT_LOG Maintenance Events

Maintenance Events define when export, purge, and command actions will be performed on the table. .

You can specify that maintenance be triggered when:

- A condition, such as low disk space, occurs or
- On a periodic or gated interval

1. Select the Maintenance Events tab in the EVENT_LOG Table Properties dialog box.



2. Specify events the same way you specify them for an DATA_LOG *(page 224)*.

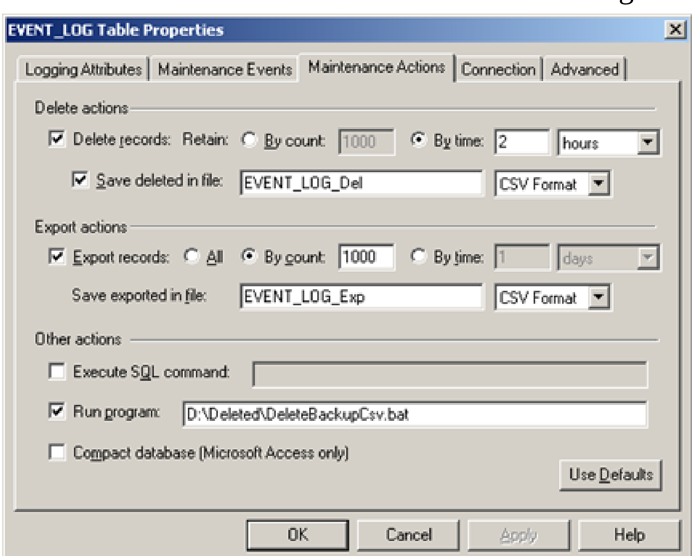
Step 3.4. Configure the EVENT_LOG Maintenance Actions

Maintenance Actions define the type of actions that take place when a Maintenance Event occurs.

You can configure an event table to:

- Export records to an export file,
- Purge records from a logging table and/or
- Incorporate command line actions.

1. Select the Maintenance Actions tab in the event log Table Properties dialog box.



2. Specify actions the same way you specify them for a data log *(page 227)* table.

Step 3.5. Configure the EVENT_LOG Connection

Connection defines the database connection and the Store and Forward properties for the table.

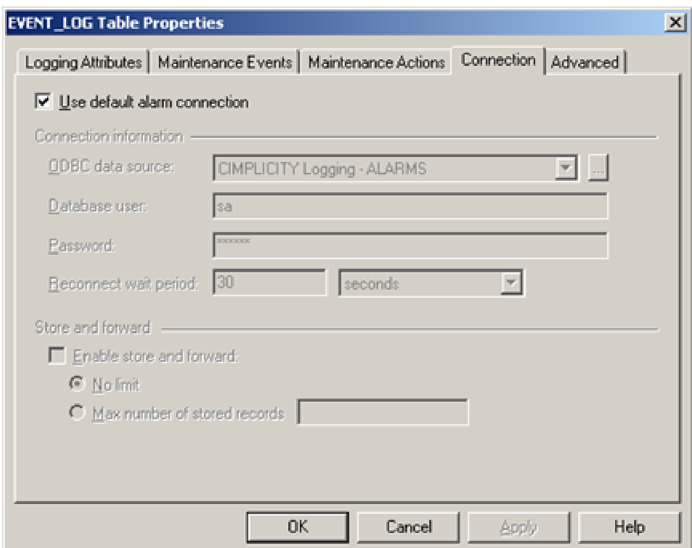
Do one of the following.

- Use the default connection.
- Specify logging connection specific to the EVENT_LOG table.

Use the default connection

1. Select the Connection tab in the EVENT_LOG Table Properties dialog box.

2. Check **Use default alarm connection**.

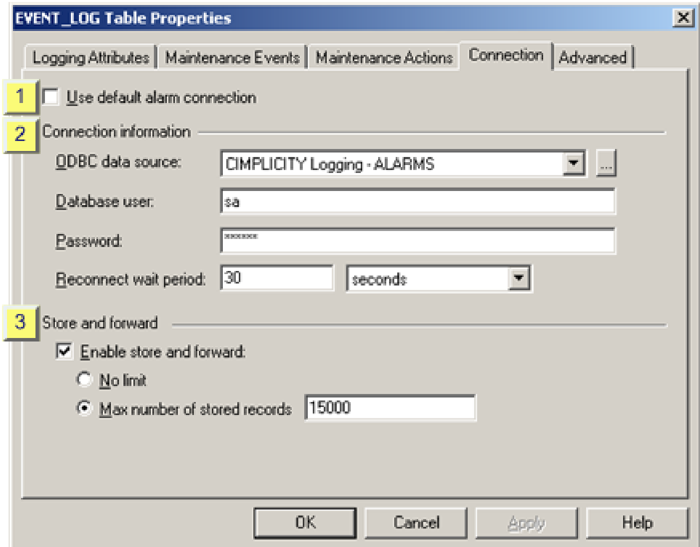


Result: The Connection tab is dimmed and unavailable for configuration. The event log uses the properties specified on the Default Alarm Connection tab in the Logging Properties dialog box.

Specify logging connections specific to the EVENT_LOG table

3. Select the Connection tab in the EVENT_LOG Table Properties dialog box.

4. Clear Use default point connection.



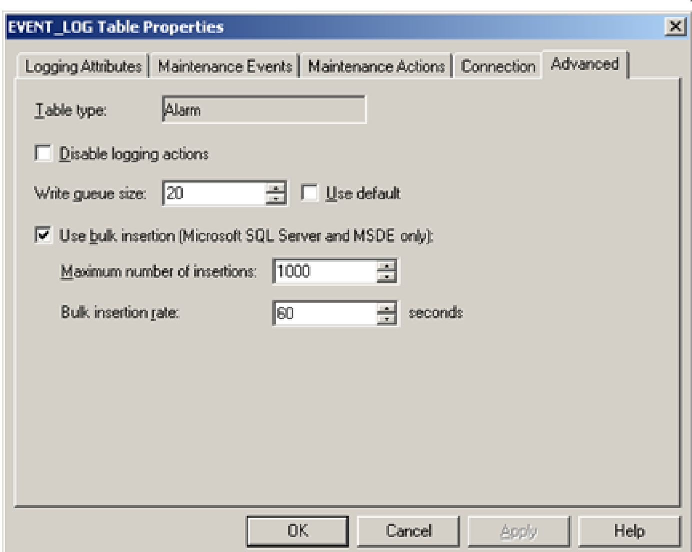
5. Do custom connection configuration the same way you do for a data log *(page 230)* table.

Step 3.6. Advanced Event Logging Configuration

The Advanced tab in the EVENT_LOG Table Properties dialog box enables you to:

- Disable logging actions for the table that you are configuring.
- Override the queue size that is specified in the Database Logger's Logging Properties dialog box.
- Select and control bulk insertion for Microsoft SQL Server or MSDE.

1. Select the Advanced tab in the EVENT_LOG Table Properties dialog box.



2. Do advanced configuration the same way you do for a DATA_LOG *(page 229)* table.

Important: For SQL Server, insert triggers fire only if bulk insertion is disabled.

Step 4. Configure Logging Conditions for a Single Event Alarm

1. Select the alarm you want to log under different conditions from the default.
2. Do one of the following.

   Method 1

   Double-click the selected Alarm ID.

   Method 2

   Click the Item Properties button on the EVENT_LOG toolbar.

   Method 3

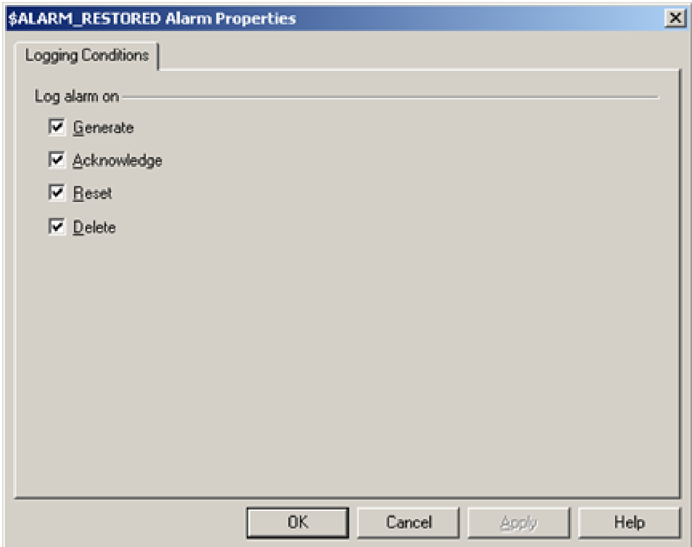
   a. Click Edit on the EVENT_LOG menu bar.
   b. Select Alarm Properties…

   Method 4

   a. Click the right mouse button.
   b. Select Properties… from the popup menu.

   The Alarm Properties dialog box opens when you use any method displaying either the default logging conditions or conditions that were previously checked for the selected alarm.

3. Check the logging conditions that apply to the selected alarm.



4. Click **OK**.

The selected alarm will be logged in the EVENT_LOG when its logging conditions occur.

Event Management Logging

Event Management Logging

The CIMPLICITY Database Logger enables you to keep a record of event management event actions using the EM_LOG table. When the event triggers actions that have been selected to be logged, the Event Manager sends a message to the logger to log..

The steps to configure the event actions logging properties include

Step 1 *(page 279)*	Add event actions to the EM_LOG.
Step 2 *(page 280)*	Configure EM_LOG properties.

Indexes, Columns and Rows in EM_LOG Tables

Indexes in EM logging tables include a:

- Primary key index on the joined timestamp and sequence number columns.
- Secondary key index on the joined timestamp_utc and sequence number columns.
- Secondary index on the timestamp alone.

- Secondary index on the timestamp_utc alone.

If you have selected the project name table attribute, the primary key index also includes the project name column.

A column type **datetime2(7)** is used for the timestamp, timestamp_utc, script_trigger_time and script_trigger_time_utc that includes the 100 nanosecond resolution.

Columns and rows in EM logging tables are as follows.

Column Name	Data Type	Description
Constant fields		
timestamp	date/time	Timestamp of the logging event.
timestamp_utc	date/time	UTC timestamp of the logging event.
sequence_number	number	Identity column to ensure uniqueness.
time	string	Local timestamp of the previous time the point was logged.
time_utc	string	UTC timestamp of the previous time the point was logged.
event_type	string	The type of Event Manager event associated with the event source.
event_source	string	Identifier that triggered the event.
action_type	string	The type of Event Manager action associated with the action source.
action_target	string	Identifier of the action's target.
script_trigger_time	string	The time when the script was triggered.
script_trigget_time_utc	string	The UTC time when the script was triggered.
Optional fields		
project	string	Name of the CIMPLICITY project.
msec	number	Actual number of milliseconds in the timestamp.

NOTE: The following types of action are logged into EM_LOG.

action_type	Description
RUN SCRIPT or RUN DOTNET SCRIPT	Logged when the script starts executing.
RUN SCRIPT DONE or RUN DOTNET DONE	Logged if the script execution is completed successfully.
RUN SCRIPT ABORT or RUN DOTNET ABORT	Logged if the script execution is aborted.
RUN SCRIPT ERROR or RUN DOTNET ERROR	Logged if an error occurs during script execution.

For every RUN SCRIPT action_type, a corresponding RUN SCRIPT DONE, or RUN SCRIPT ABORT, or RUN SCRIPT ERROR action_type would be logged. This enables you to know the status of an action.

For every RUN SCRIPT DONE or RUN DOTNET DONE action_type, script_trigger_time and script_trigget_time_utc are logged . This enables you to know the duration of script execution.

The duration of script execution is the difference between timestamp and script_trigger_time logged in the row that contains RUN SCRIPT DONE or RUN DOTNET DONE action_type.

You can execute the following query to calculate the duration of script execution:

```
SELECT[timestamp]

      ,[timestamp_utc]

      ,[script_trigger_time]

      ,[script_trigger_time_utc]

      ,DATEDIFF(mcs, [script_trigger_time], [timestamp])/1000000.0 as
 'Duration in seconds'

      ,[sequence_number]

      ,[action_type]

      ,[action_target]

      ,[event_type]

      ,[event_source]

  FROM [EM_LOG]

  WHERE [action_type] LIKE 'RUN SCRIPT%' AND NOT [script_trigger_time] IS
 NULL

GO
```

Review Maintenance Events

Review Maintenance Events for CIMPLICITY:

- ALARM_LOG. See Step 3.4. Configure Alarm Log Maintenance Events *(page 260)*.
- COR_LOG. See Step 2.2. Configure Status Log Logging Maintenance Events *(page 289)*.
- Data log table. See Step 2.4. Configure Logging Maintenance Events *(page 224)*.
- EM_LOG. See Step 2.3. Configure Event Management Logging Maintenance Events *(page 282)*.
- EVENT_LOG. See Step 3.3. Configure EVENT_LOG Maintenance Events *(page 271)*.
- Group log table. See Step 2.4. Configure Group Maintenance Events *(page 244)*.

- An external application log. See Application Logging *(page 295)*.

Review Maintenance Actions

Review Maintenance Actions for CIMPLICITY:

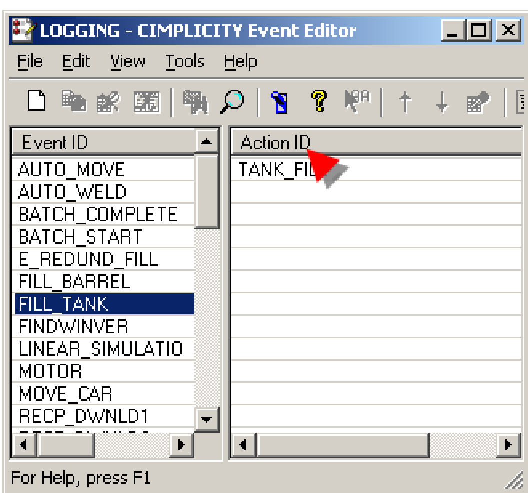
- ALARM_LOG. See Step 3.5. Configure Alarm Log Maintenance Actions *(page 261)*.
- COR_LOG. See Step 2.3. Configure Status Log Logging Maintenance Actions *(page 290)*.
- Data log table. See Step 2.5. Configure the Logging Maintenance Actions *(page 227)*.
- EM_LOG. See Step 2.4. Configure Event Management Logging Maintenance Actions *(page 283)*.
- EVENT_LOG. See Step 3.4. Configure the EVENT_LOG Maintenance Actions *(page 272)*.
- Group log table. See Step 2.5. Configure Group Maintenance Actions *(page 245)*.
- An external application log. See Application Logging *(page 295)*.

Step 1. Add Event Actions to the EM_LOG

1. Expand the Basic Control Engine folder in the Workbench left pane.
2. Double-click **Event Editor**.

   The Event Editor window opens.

3. Select the Event ID in the left pane to which you want to attach the action.
4. Left-click the Action ID column.

5. Open the New Event-Action dialog box using any of the following methods.

   Method 1

Click the **New Event_Action** button  on the CIMPLICITY Event Editor toolbar.

Method 2

a. Click File on the CIMPLICITY Event Editor menu bar.
b. Select New Event_Action.

Method 3

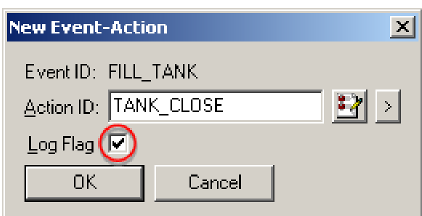
a. Click the right-mouse button in the right pane.
b. Select New Event_Action from the popup menu.

Method 4

Press **Ctrl+N** on the keyboard.

The New Event-Action dialog box opens when you use any of these methods.

6. Enter a name for the **Action ID**.
7. Check **Log Flag**.



8. Click **OK**.

When the selected event triggers the action, the Event Manager sends the data to the Database Logger.

Step 2. Configure EM_LOG Properties

Step 2. Configure EM_LOG Properties

You configure the event action logging properties in the EM_LOG Properties dialog box.

The tasks to configure the Event Management action logging properties include:

Step 2.1 *(page 281)*	Open the EM_LOG Properties dialog box.
Step 2.2 *(page 281)*	Configure logging attributes.

Step 2.3 *(page 282)*	Configure maintenance events.
Step 2.4 *(page 283)*	Configure maintenance actions.
Step 2.5 *(page 284)*	Configure the EM_LOG connection.
Step 2.6 *(page 285)*	Configure advanced properties for the EM_LOG.

Step 2.1. Open the EM_LOG Properties Dialog Box

1. Select EM_LOG in the Database Logger Configuration window.
2. Do one of the following.

   Method 1

   Click the **Table Properties** button  on the Database Logger Configuration window toolbar.

   Method 2

   a. Click the right mouse button.
   b. Select Properties… from the popup menu.

   Method 3

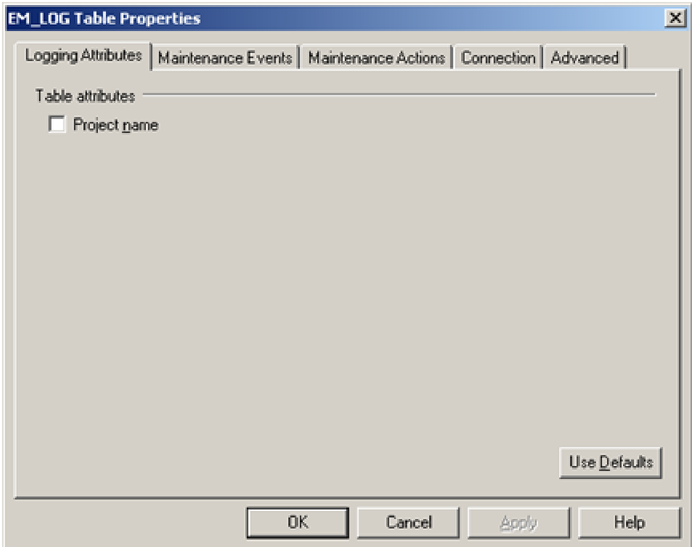
   a. Click Edit on the Database Logger Configuration window menu bar.
   b. Select Table Properties…

The EM_LOG Table Properties dialog box opens when you use any method.

Step 2.2. Configure Event Management Logging Attributes

Logging Attributes for an EM_LOG enables you to log the name of the CIMPLICITY project.

1. Select the Logging Attributes tab in the Table Properties dialog box.



2. Check **Project name** to log the project name.

Tip: Click **Use Defaults** to set the logging attributes to their default value. The EM_LOG will be set to not log the **Project Name**.

Warning: If you select or de-select the project name table attribute you must drop the table so that it is properly recreated with or without the new project field as part of the primary key. Failing to do so can cause the database logger to fail to log data into the table.

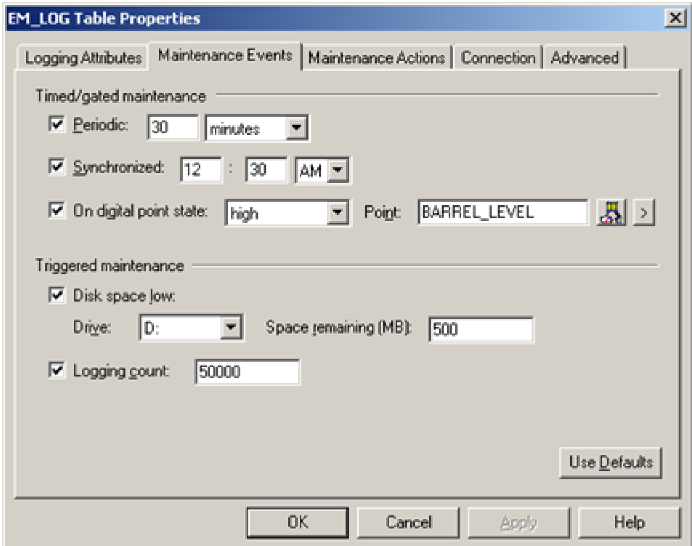
Step 2.3. Configure Event Management Logging Maintenance Events

Maintenance Events define when export, purge, and command actions will be performed on the table.

You can specify that maintenance be triggered when:

- A condition, such as low disk space, occurs or
- On a periodic or gated interval

1. Select the Maintenance Events tab in the EM_LOG Table Properties dialog box.



2. Specify actions the same way you specify them for a data log *(page 224)* table.

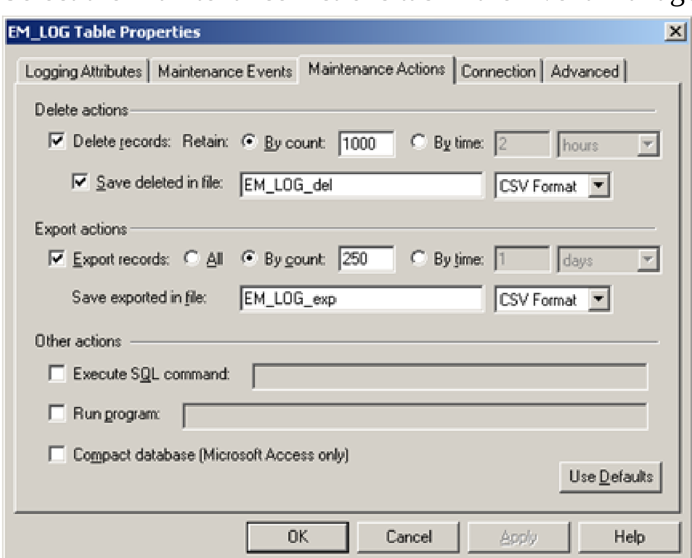
Step 2.4. Configure Event Management Logging Maintenance Actions

Maintenance actions define the type of actions that take place when a Maintenance Event occurs.

You can configure the EM_LOG table to:

- Export records to an export file
- Purge records from a logging table and/or
- Incorporate command line actions.

1. Select the Maintenance Actions tab in the Event Manager Log Table Properties dialog box.



2. Specify events the same way you specify them for a data log *(page 227)* table.

Step 2.5. Configure an EM_LOG Connection

Connection defines the database connection and the Store and Forward properties for the table.

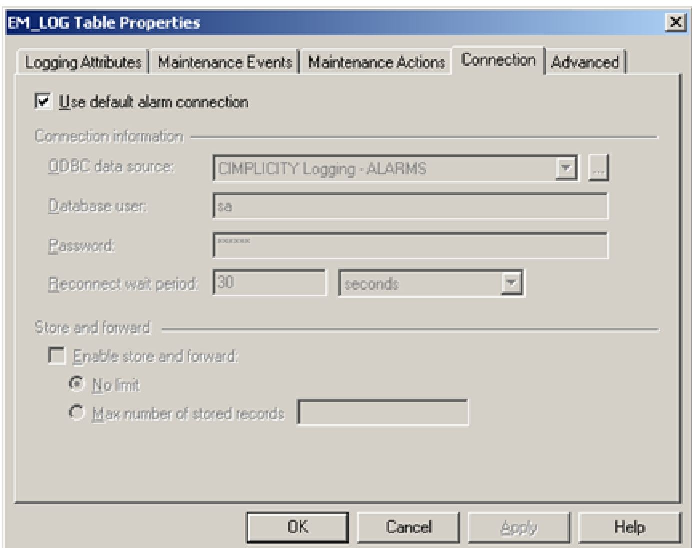
Do one of the following.

- Use the default connection.
- Specify logging connection specific to the EM_LOG table.

Use the default connection

1. Select the Connection tab in the EM_LOG Table Properties dialog box.

2. Check **Use default point logging**.

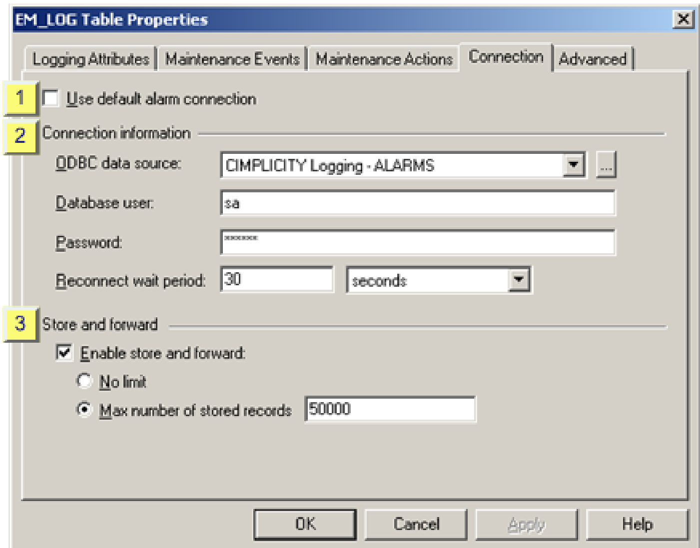


Result: The Connection tab is dimmed and unavailable for configuration. The EM_LOG uses the properties specified on the Default Alarm Logging tab in the Logging Properties dialog box.

Specify logging connections specific to the EM_LOG table

3. Select the Connection tab in the EM_LOG Table Properties dialog box.

4. Clear Use default point connection.



rect 0, 43, 330, 72 *(page 231)*
rect 1, 184, 409, 277 *(page 233)*
rect 0, 70, 408, 186 *(page 231)*

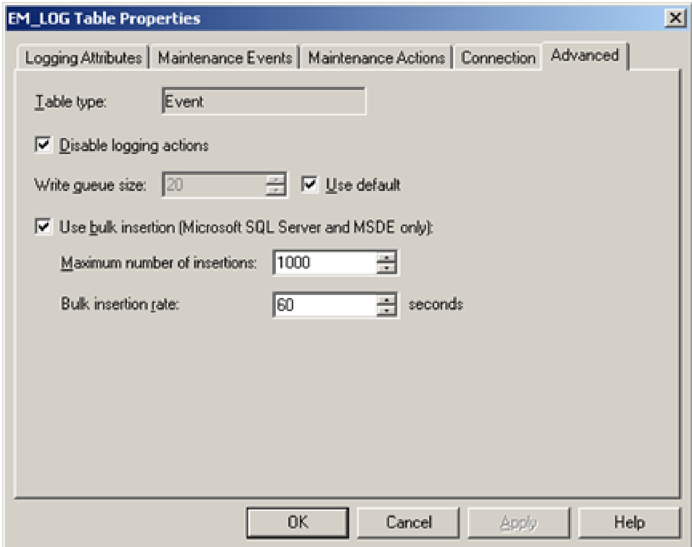
5. Do custom connection configuration the same way you do for a data log *(page 230)* table.

Step 2.6. Do advanced EM_LOG Configuration

The Advanced tab in the EM_LOG Table Properties dialog box enables you to:

- Disable logging actions for the table that you are configuring.
- Override the queue size that is specified in the Database Logger's Logging Properties dialog box.
- Select and control bulk insertion for Microsoft SQL Server or MSDE.

1. Select the Advanced tab in the EM_LOG Table Properties dialog box.



2. Do advanced configuration the same way you do for a data log *(page 229)* table.

Important: (For SQL Server) insert triggers fire only if bulk insertion is disabled.

Status Log Logging

Status Log Logging

The CIMPLICITY Database Logger enables you to record a project's status log messages to the COR_LOG database table in addition to the cor_recstat.cl2 file in the project’s \log directory. The project is scanned periodically, and any records added to it are also added to the COR_LOG table.

To configure Status Log logging, you must:

- Step 1. Enable Status Log Logging *(page 287)*
- Step 2. Configure COR_LOG Properties *(page 288)*
- Step 3. Configure Scan Time *(page 294)*

Indexes and Columns in COR_LOG Tables

Indexes in COR_LOG tables include the following:

- A primary key index in the joined timestamp_utc and sequence number columns.
- A secondary key index in the joined timestamp and sequence number columns.
- A secondary index in the timestamp_utc alone. This is the clustered index for the table.

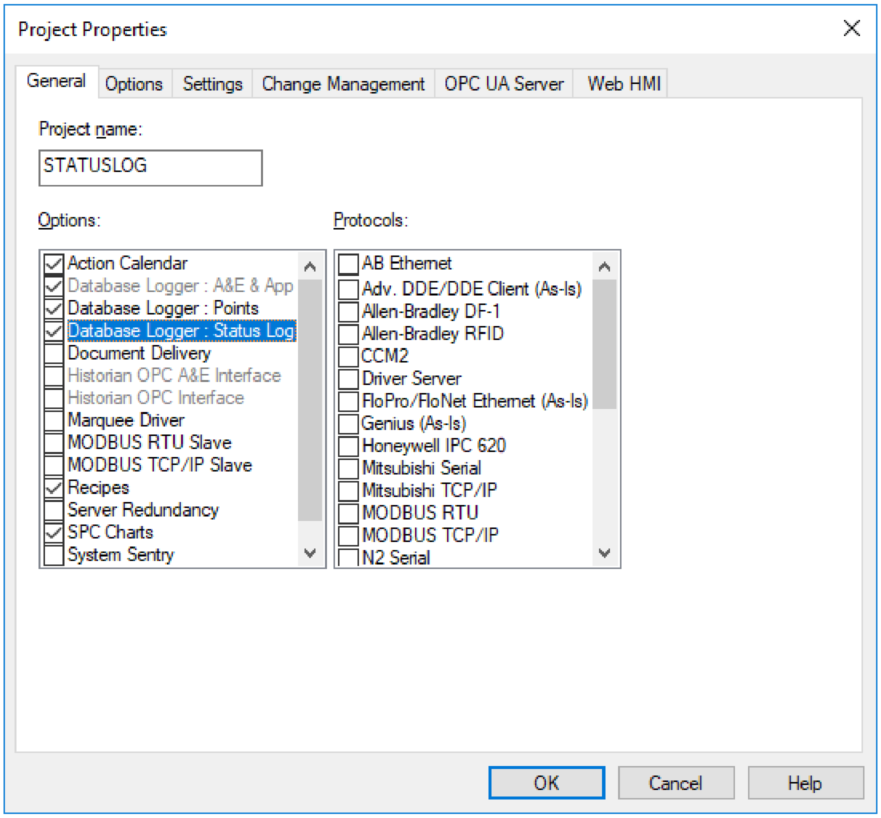
The columns and rows in COR_LOG tables are as follows:

Column Name	Data Type	Description
timestamp	datetime	Timestamp of the logging event.
timestamp_utc	datetime	UTC timestamp of the logging event.
sequence_number	number	Identity column to ensure uniqueness.
date_time	datetime	Date/Time field in Status Log.
project	varchar(21)	Name of the project from where this Status Log record originated.
severity	int	Severity field in the Status Log.
pid	int	Process id of the process logging the Status record.
process	varchar(33)	Process field in the Status Log.
procedure	varchar(33)	Procedure field in the Status Log.
status	int	Status field in the Status Log.
reference	int	Reference field in the Status Log.
code	int	Code field in the Status Log.
source	varchar(21)	Source field in the Status Log.
message	varchar(1024)	Message field in the Status Log.

Step 1. Enable Status Log Logging

1. In the **CIMPLICITY Workbench** window, select **Project**, and then select **Properties**.
2. In the **Project Properties** window, in the **General** section, select the **Database Logger : Status Log** option.

    Note: This will enable the Database Logger : A&E & App option.



Status logging is now enabled.

Step 2. Configure COR_LOG Properties

Step 2. Configure COR_LOG Properties

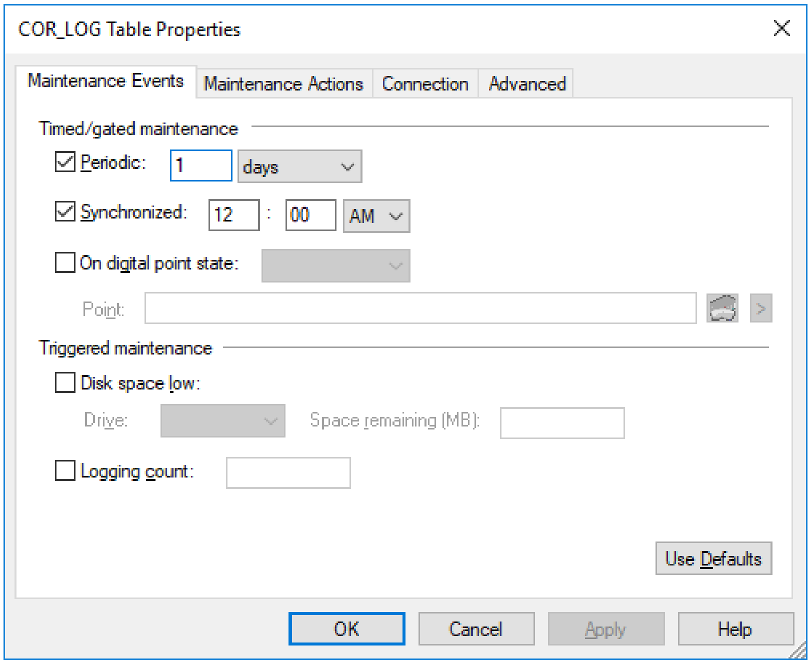
The CIMPLICITY Database Logger enables you to keep a record of event management event actions using the EM_LOG table. When the event triggers actions that have been selected to be logged, the Event Manager sends a message to the logger to log..

The steps to configure the event actions logging properties include:

- Step 2.1. Access the COR_LOG Properties Dialog Box *(page 288)*
- Step 2.2. Configure Status Log Logging Maintenance Events *(page 289)*
- Step 2.3. Configure Status Log Logging Maintenance Actions *(page 290)*
- Step 2.4. Configure the COR_LOG Connection *(page 291)*
- Step 2.5. Perform advanced COR_LOG Configuration *(page 293)*

Step 2.1. Access the COR_LOG Properties Dialog Box

1. In the **CIMPLICITY Workbench** window, in the left pane, expand **Project**, and then select **Database Logger**.



2. Double-click the **Database Logger** file.

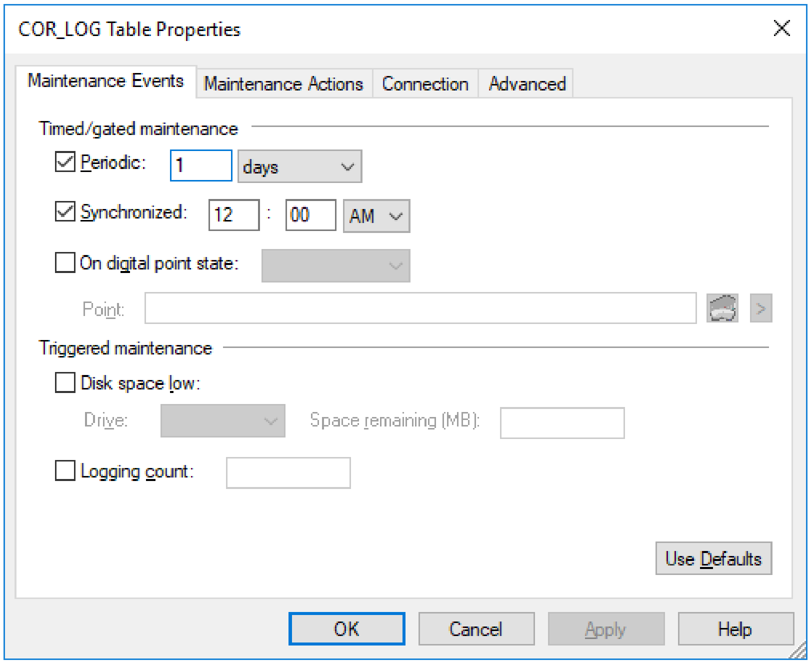
3. In the **Database Logger Configuration** window, select COR_LOG.

4. Select **Edit**, and then select **Table Properties**. Alternatively, you could select  , or right-click, and then select Properties. You can now access the COR_LOG Properties dialog box.

Step 2.2. Configure Status Log Logging Maintenance Events

The Maintenance Events section defines when the export, purge, and command actions will be performed on the table. You can specify that maintenance is triggered when either a condition, such as low disk space, occurs; or a periodic or gated interval of time passes. Use these steps to configure maintenance events.

1. In the **COR_LOG Table Properties** dialog box, select the **Maintenance Events** tab.

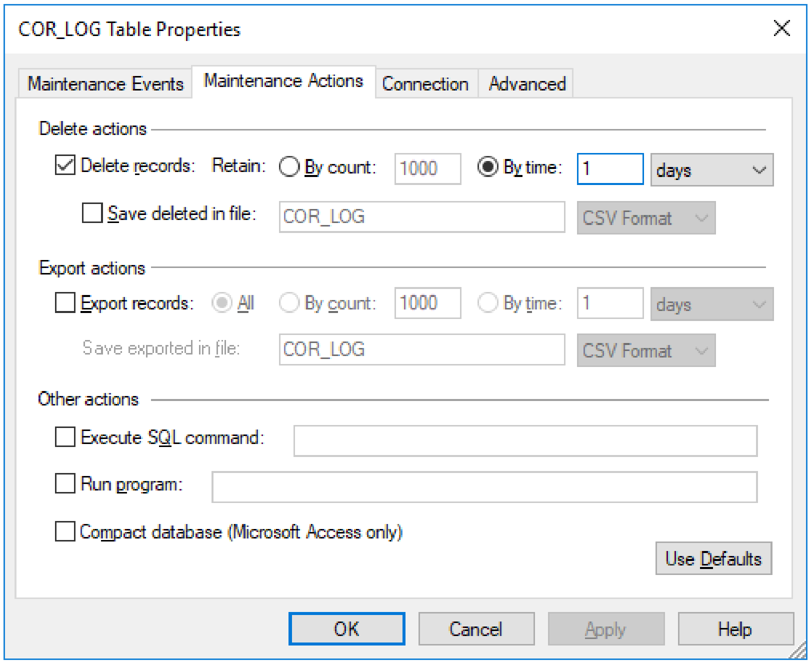


2. Specify actions the same way you specify them for a data log table, as described for Point Data Logging in Step 2.4. Configure Logging Maintenance Events *(page 224)*.

Step 2.3. Configure Status Log Logging Maintenance Actions

The Maintenance Actions section defines the type of actions that take place when a maintenance event occurs. You can configure the COR_LOG table to: Export records to an export file, Purge records from a logging table, and/or Incorporate command line actions. Use these steps configure maintenance actions for a table.

1. In the COR_LOG Table Properties dialog box, select the Maintenance Actions tab.



2. Specify events the same way you specify them for a data log table, as described for Point Data Logging in Step 2.4. Configure Logging Maintenance Events *(page 224)*.

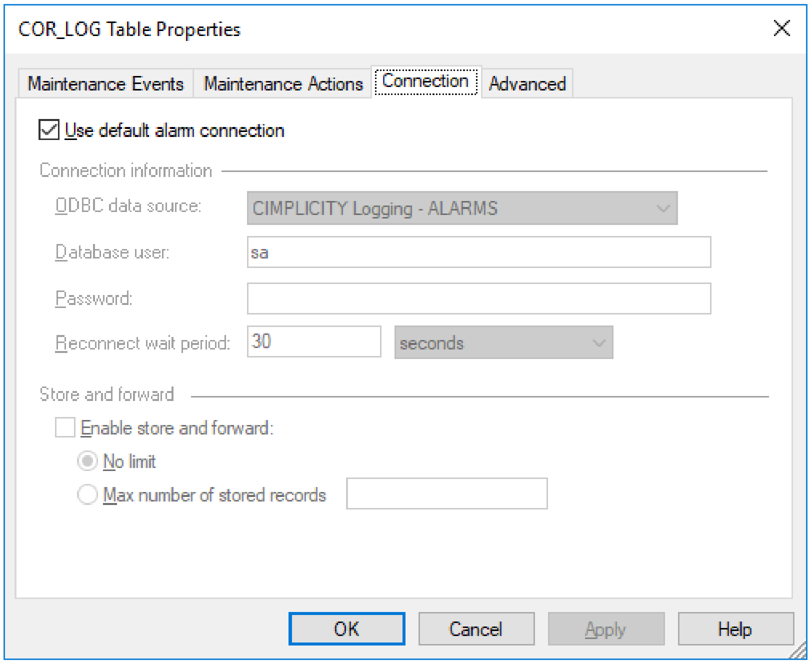
Step 2.4. Configure the COR_LOG Connection

The Connection section defines the database connection and the store and forward properties for the table. Do one of the following: Use the default connection, or Specify logging connection specific to the COR_LOG table.

Use the default connection

1. In the **COR_LOG Table Properties** dialog box, select the **Connection** tab.

2. Select the **Use default alarm connection** check box.

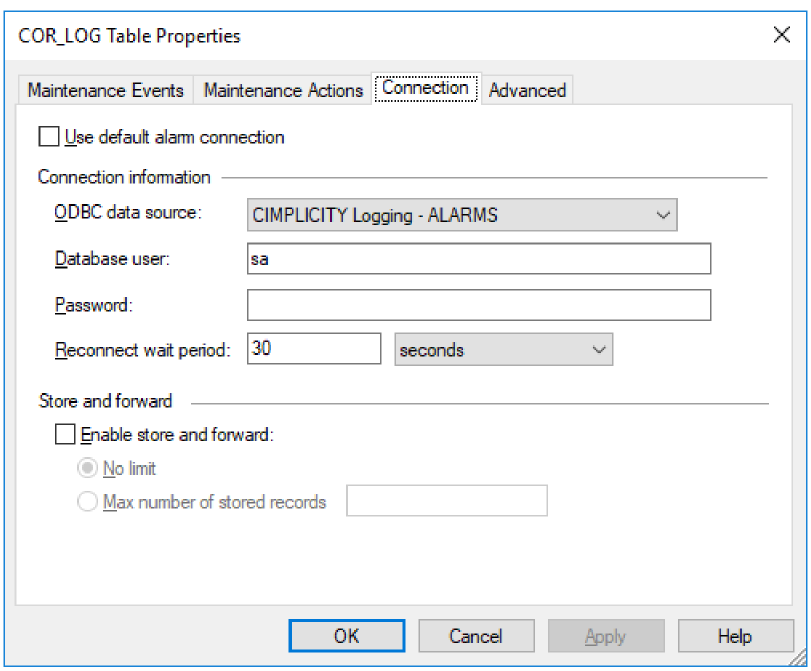


The COR_LOG table now uses the properties specified in the Default Alarm Connection section in the Logging Properties dialog box.

<u>Specify logging connections specific to the COR_LOG table</u>

1. In the **COR_LOG Table Properties** dialog box, select the **Connection** tab.

2. Clear the **Use default alarm connection** check box.

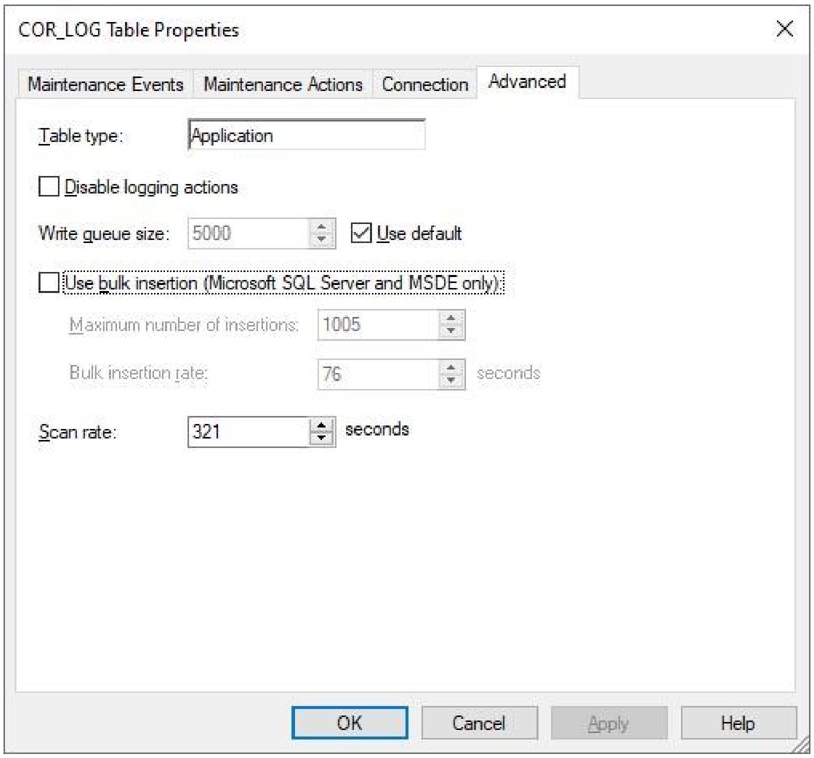


3. Perform the custom connection configuration, and store and forward configuration as it is done for a data log table, as described for Point Data Logging in Step 2.4. Configure Logging Maintenance Events *(page 224)*.

Step 2.5. Perform advanced COR_LOG Configuration

The Advanced section in the COR_LOG Table Properties dialog box enables you to: Disable logging actions for the table that you are configuring, Override the queue size that is specified in the Database Logger's Logging Properties dialog box, and Select and control bulk insertion for Microsoft SQL Server. Use these steps to perform advanced configuration.

1. In the **COR_LOG Table Properties** dialog box, select the **Advanced** tab.



2. Perform advanced configuration the same way you specify them for a data log table, as described for Point Data Logging in Step 2.4. Configure Logging Maintenance Events *(page 224)*.

   Note: In the**Scan rate** field, specify the periodic rate at which you want to scan the Status Log file for added records.

   Important: For SQL Server, insert triggers are fired only if bulk insertion is disabled.

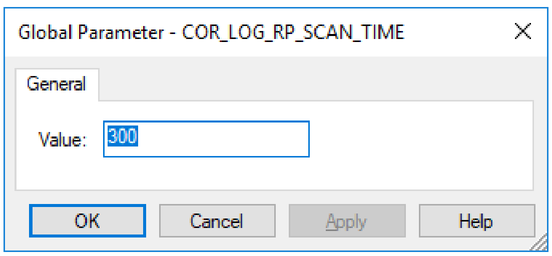
Step 3. Configure Scan Time

The Status Log file is periodically scanned to check for records that have been added to it since the last scan. You can specify the periodic rate at which the Status Log file is scanned for added records value, using the global parameter **COR_LOG_RP_SCAN_TIME** or from the **Advanced** tab of COR_LOG Table Properties.

Following are the steps to set COR_LOG scan time from Global Parameters:

1. **From Global Parameters**: In the **CIMPLICITY Workbench** window, in the left pane, expand **Project**, expand **Advanced**, and then select **Project Parameters**.

2. In the list of parameters, double-click **COR_LOG_RP_SCAN_TIME**, enter a value, and then select **OK**.



Note: The default value of **COR_LOG_RP_SCAN_TIME** is 300 seconds. The minimum value allowed is 15 seconds and the maximum value allowed is 3600 seconds.

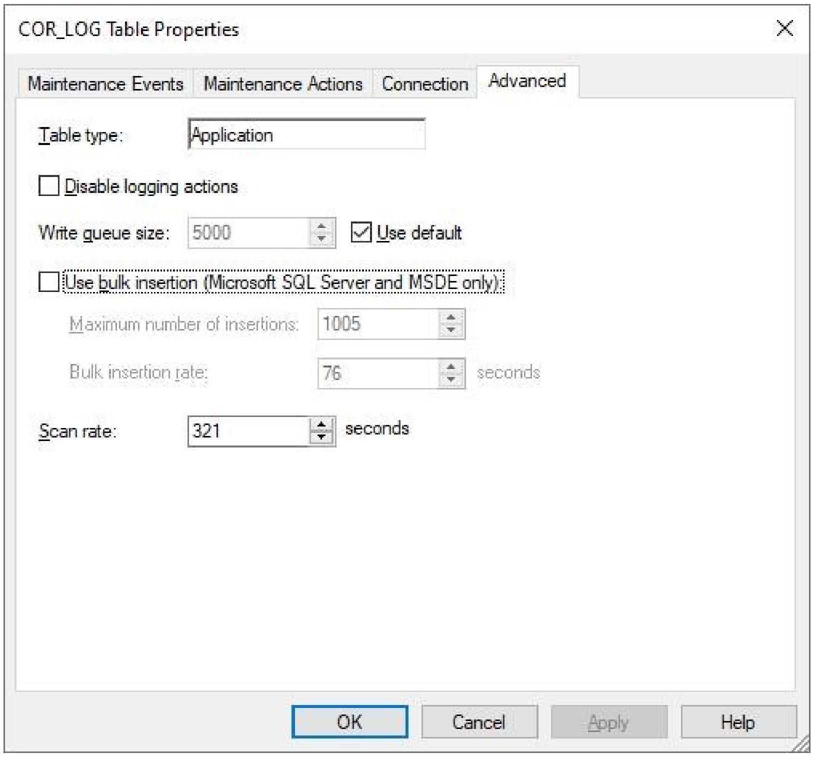
From COR_LOG Table Properties

Following are the steps to set COR LOG scan time from the **Advanced** tab of COR_LOG Table Properties.

1. In CIMPLICITY Workbench, double-click **Database Logger**.

2. Right-click COR_LOG, and then select **Properties**.

3. Select the **Advanced** tab.



4. In **Scan rate** field, enter the value at which you want to scan the Status Log for the records.

   Note: The default value of **COR_LOG_RP_SCAN_TIME** is 300 seconds. The minimum value allowed is 15 seconds and the maximum value allowed is 3600 seconds.

5. Select **OK**.

Application Logging

Application Logging

Application Logging enables you to log actions of certain CIMPLICITY applications, such as the Basic Control Engine. To take advantage this feature, you must configure an Application Table for the product option.

The application chooses what data to log and when to log it.

Tip: You can configure the following properties for an Application logging table.

- Logging attributes.
- Maintenance actions.
- Maintenance events.

- Logging properties.

Follow the same steps that you follow for the EM_LOG *(page 276)* .

Indexes, Columns, Rows in Application Logging Tables

Indexes in Application logging tables include a:

- Primary key index on the joined timestamp and sequence number columns.
- Secondary key index on the joined timestamp_utc and sequence number columns.
- Secondary index on the timestamp alone.
- Secondary index on the timestamp_utc alone.

If you have selected the project name table attribute, the primary key index also includes the project name column.

Columns in Application logging tables include:

- Columns that are specific to that table.
- An additional column for each table attribute you have selected.

Rows in all Application logging tables include:

- A timestamp column.
- A timestamp_utc column
- An auto increment sequence number that ensures that each row is unique.

Proficy Historian Migration

Historian Migration

- Overview
- Historian migration configuration steps

Overview

When you update CIMPLICITY logging to use the Historian logging database, you can preserve previously logged data by using the Proficy a Migration tool.

The Historian Migration utility migrates all or selected data from the previously used log database, e.g. CIMPLICITY SQL, to Historian.

Tables that can be migrated are:

- DATA_LOG
- ALARM_LOG
- EVENT_LOG

You can migrate as many records that the Historian license permits.

Tables that cannot be migrated include:

- SPC tables
- GROUP_LOG

Historian Migration Configuration Steps

Migration is straightforward.

When you open the Historian Migration tool, you open a migration session. You specify what data you want migrated. When the data is migrated and you close the Historian Migration tool, the migration session is ended. If all of the data you want to migrate has been migrated you may not use the tool again. If there is more data to migrate, you can simply, re-open the tool, select your criteria and migrate your selection.

Note: The previously used database will not be removed after migration.

Step 1 *(page 300)*	Open the Historian Migration tool.
Step 2 *(page 300)*	Select data to migrate
Step 3 *(page 303)*	Enter advanced migration specifications
Step 4 *(page 307)*	View log files

Historian Migration Mapping

Important: There are some differences in how data is identified between the CIMPLICITY and the real-time Collector.

The following lists define how data is mapped from CIMPLICITY to Historian to resolve these differences.

- DATA_Log migration mapping.

- ALARM_LOG and EVENT_LOG migration mapping.
- Mapping guidelines.

DATA_LOG migration mapping

The Historian Migration utility maps data from the CIMPLICITY DATA_LOG to Historian as follows.

1 (page 298)	Quality mapping
2 (page 298)	Data Type mapping
3 (page 299)	Point/Tag attributes mapping
4 (page 299)	Data attribute mapping

Quality Mapping

CIMPLICITY Quality Attribute Value	Historian Quality, Subquality
IS_AVAILABLEand IS_IN_RANGE and STALE=FALSE	Good, ihOPCNonspecific
IS_AVAILABLE and IS_IN_RANGE and STALE=TRUE	Good, ihOPCLastKnowValue
NOT IS_AVAILABLE	Bad, ihOPCCommFailure
NOT IS_IN_RANGE	Bad, ihOPCOutOfService
DISABLE_WRITE ALARMED ALARMS_ENABLED	No effect on Historian quality No effect on Historian quality No effect on Historian quality
USER Qualityflags	Bad, ihOPCNonspecific

Data Type Mapping

CIMPLICITY Point Types	Historian Data Type
BOOL SINT INT USINT	Integer Double integer Double integer Double integer
REAL UDINT	Double float Double float
DINT UINT	Double Integer Double Integer
STRING STRING_20 STRING_8 STRING_80	Variable String Variable String Variable String Variable String
3D_BCD 4D_BCD BYTE	Not supported Not supported Not supported

Point/Tag Attribute Mapping

CIMPLICITY	Historian
Point_id	Tagname
_ENG	Engineering Units
FROM SQL Table	DataType

Data Attribute Mapping

CIMPLICITY	Historian
Timestamp	Timestamp
_VAL	Value
_QUALITY	Quality

Note: Hi and Lo Engineering Units do not exist in CIMPLICITY SQL table; these fields are ignored during the migration.

ALARM_LOG and EVENT_LOG migration mapping

During the Historian migration fields in the ALARM_LOG and EVENT_LOG are mapped to be consistent with fields provided by the OPC Alarm & Event server.

CIMPLICITY	Historian
alarm_message	Description
Project, resource, alarm_id	Source (project/resource/alarm_id)
timestamp	TimeStamp
generation_time	Endtime Condition Subcondition Event Category
HIGH,MED,LOW,$SYS	Severity Quality

Mapping Guidelines

- `AlarmType` is as follows.

AlarmType	When read from
ihALARM_CONDITION	ALARM_LOG table
ihALARM_SIMPLE	EVENT_LOG table

- `Condition` and `SubCondition` are as follows.

For the:	Condition and Subcondition
Alarm log	Exist, only for the Alarm log.
Tag alarm	Are set to **Level** to match the real time collector.
System tag	Use **System**.

- `EventCategory` use the following.

For the:	EventCategory
Tag alarm	**Level"**
System alarm	**System**

- `Severity` is configurable by user.

Default mapping is as follows.

CIMPLICITY	Historian
HIGH $SYS	850
MED	500
LOW	150

- `DataSource` is `PEAE_MIGRATION`.
- `Source` is combination of project/resource/alarm_id.

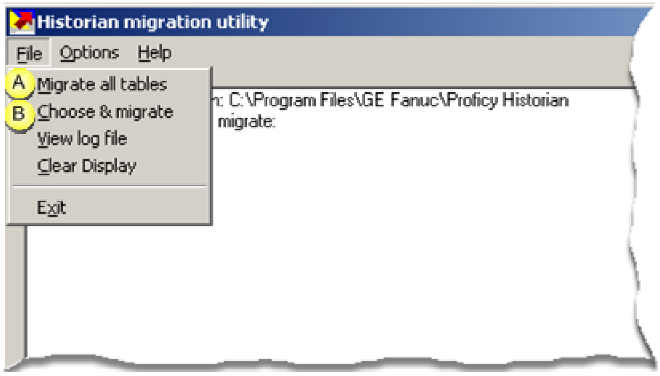
Step 1. Open the Historian Migration Utility

1. Right-click any icon in the Database Logger Configuration window's list.
2. Select Migration… on the Popup menu.

Step 2. Select Data to Migrate

1. Click File on the Historian migration utility menu bar.

2. Select either of the migration options.



rect 3, 41, 113, 63 (page 301)
rect 3, 61, 113, 81 (page 301)

A	Migrate all tables

Migrate all available tables in the SQL database to Historian.

Available tables include:

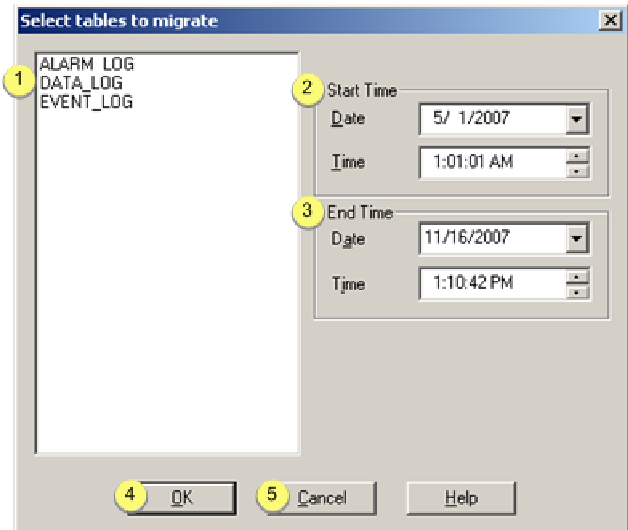
- DATA_LOG
- ALARM_LOG
- EVENT_LOG

Important: SPC log tables and GROUP_LOG tables are not included in the migration.

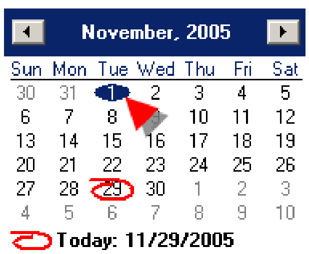
B	Choose & migrate

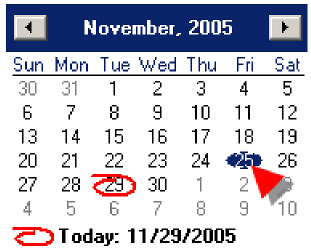
Migrate a subset of tables.

Choices include the following.



rect 0, 25, 177, 139 [(page 302)]
rect 178, 39, 373, 117 [(page 302)]
rect 180, 117, 375, 195 [(page 303)]
rect 64, 290, 154, 327 [(page 303)]
rect 156, 292, 237, 326 [(page 303)]

1	Tables available for migration		One or more of the available tables can be selected for migration.	
2	Start Time	Date	Month/Day/Year of data that should be selected as the first data in the migration. **Note:** Click the Down Arrow button to open a calendar on which you can make the selection. 	
			Default	1/ 1/1969
		Time		Hour:Minute Second (AM or PM) of data that should be selected as the first data in the migration.
			Default	1:01:01 AM
			The default date and time migrates all data up to the selected end time.	

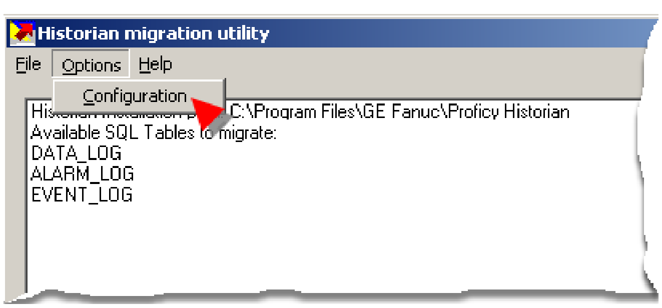
3	End Time	Date	Month/Day/Year of data that should be selected as the last data in the migration. **Note:** Click the Down Arrow button to open a calendar on which you can make the selection. 	
			Default	Current date.
		Time		Hour:Minute Second (AM or PM) of data that should be selected as the last data in the migration.
			Default	Current time.
			The default migrates data from the start date to the most current.	
4	OK	Closes the Select tables to migrate dialog box. The migration begins as soon as the dialog box is closed.		
5	Cancel	Cancels the migration.		

Important: The same start time and end time are used for each selected table during the migration session. The selections apply to the one migration session only. Once completed, settings revert to the default settings.

Migration occurs after you select either option. A status bar displays migration progress.

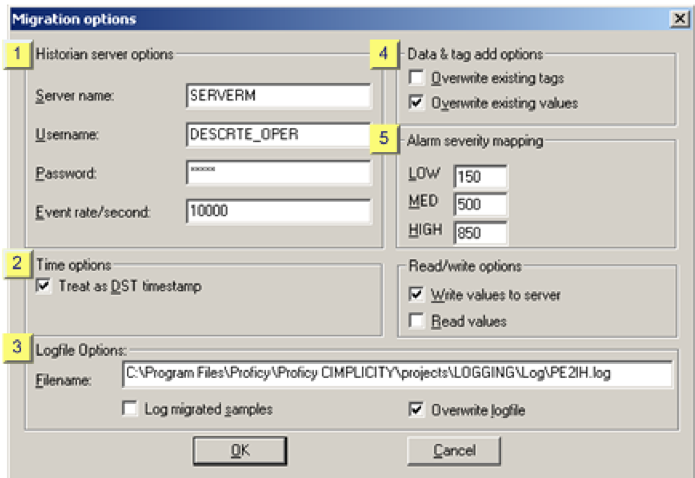
Step 3. Enter Advanced Migration Specifications

Click Options>Configuration on the Historian migration utility menu bar.



The Migration options dialog box opens.

Configuration options are as follows.



rect -1, 11, 222, 148 [(page 304)]
rect 221, 70, 410, 149 [(page 306)]
rect 220, 9, 409, 72 [(page 305)]
rect 2, 203, 434, 254 [(page 305)]
rect 0, 146, 223, 197 [(page 305)]

1 (page 304)	Historian server options
2 (page 305)	Time options
3 (page 305)	Logfile Options
4 (page 305)	Data & tag add options
5 (page 306)	Alarm severity mapping
6 (page 307)	Read/write options

1	Historian server options

Option	Description

Server name	Historian Server name.	
	Default	• CIMPLICITY Server if the Historian Server is the same as the CIMPLITY Server. • Blank if the CIMPLICITY Server is different from the Historian Server.
Username	Valid Historian user name. If the name is not recognized by Historian, the migration will fail.	
Password	Password required for the specified Historian user. If the password is incorrect, the migration will fail.	
Event rate/ second	Number of data per second that can be sent to the Historian Server. **Note:** This can be used to slow data flow for Historian Server processing requirements.	
	No limitation	0
	Default	10000

2	Time options

Treat as DST timestamp directs CIMPLICITY to do the following.

Checked	Converts the time stamp entered when the data was logged from Local to Universal Tmestamp format.
Clear	Leaves the time stamp in the designated Local format.

3	Logfile options

Logfile options provide instructions for logfile maintenance.

Option	Description	
Filename	Location and name of the file that contains the log data.	
	Default	...\Project Path\Project Name\Log\<Logfile.log>
Log migrated samples	Checked	Every sample sent to Historian is logged on screen
	Clear	Samples sent to Historian are not logged on screen.
	Default	Clear
Overwrite logfile	Checked	Overwrites the existing log file with data from the last migration.
	Clear	Creates a new logfile for the current migration.
	Default	Checked

4	Data & tag add options

Tags and values that already exist in Historian can be overwritten if specified.

Checkbox	Description

Overwrite existing tags	Checked	Overwrites the tag properties in Historian if the tag name already exists in Historian. Associated tag properties are: • datatype • Sourceaddr(projectname\PointID.VALUE) • Engineering Unit (if available).
	Clear	If the tag name already exists in Historian, the migrating tag's properties will not written to Historian. **Note:** If the tag's properties already in Historian are different from the properties of the tag that will be migrated, the new tag properties will not be migrated to Historian.
	Default	Clear
Overwrite existing values	Checked	Overwrites the existing values for the existing Historian tag.
	Clear	Appends new values for the existing Historian tag in the Historian log.
	Default	Checked

5	Alarm severity mapping

- A CIMPLICITY project

Has the following default classes for its alarms.

- LOW
- MED
- HIGH
- $SYS
- $ACAL

May have one or more user defined alarm classes.

- The OPC Alarm & Event Server defines alarm categories as integers, not strings.

During migration CIMPLICITY alarm classes are mapped to the integer value entered in the **LOW**, **MED** and **HIGH** fields..

Field	Description (after migration)	
LOW	The LOW alarm class is identified as the entered value.	
	Default	150
MED	The MED alarm class is identified as the entered value.	
	Default	500

HIGH	The following alarm classes are identified as the entered value. • HIGH • SYS • User defined alarm classes	
	Default	850

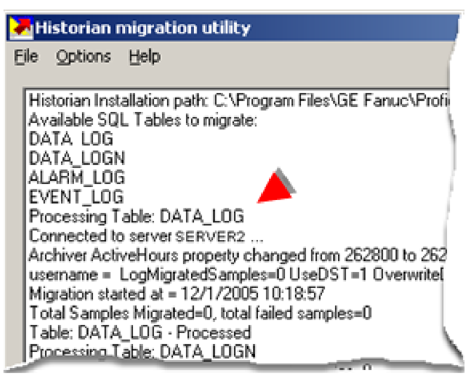
6	Read/write options

Read/write options provides the tools to validate the migration.

Checkbox	Description	
Write values to server	Checked	Writes values to Historian during the migration.
	Clear	A simulation is conducted to determine if the data can get written from CIMPLICITY logging to Historian. Values are not actually written to Historian during the migration.
	Default	Checked
Read values	Checked	Enables validating data if it was written successfully and can be read. When the migration is complete a window will open for validation purposes. **Important:** Validation reads each tag and its data back from Historian. This process can be very time consuming.
	Clear	Validation is not performed for reading values.
	Default	Clear

Step 4. View Migration Reports

A migration progress summary displays in the historian migration utility window during the migration session after each migration.



Tip: Click File>Clear Display on the Historian migration utility menu bar to clear the window.

More detailed migration information is included in the following.

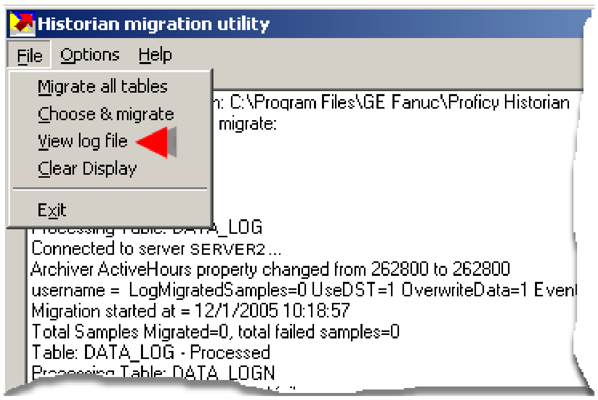
- Migration log file

• Data migration report

Migration Log File

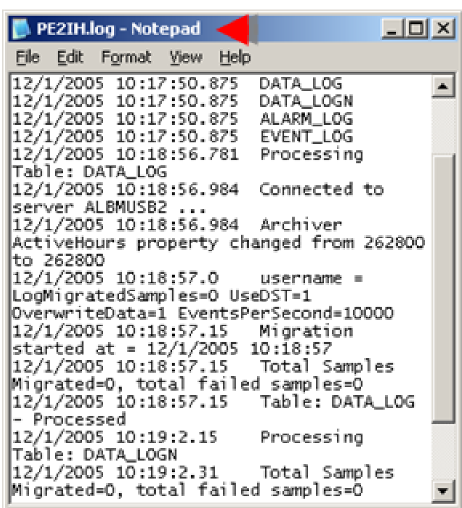
1. Click File on the Historian migration utility menu bar.

2. Select View log file.



The log file that is named in the Migration options dialog box opens in Notepad.

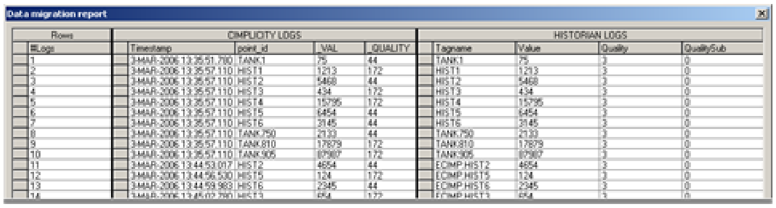
The file displays details of the last migration during the open session.



Data Migration Report

Note: Make sure you check Read Values *(page 307)* in the Migration Options dialog box if you want the Migration report to open when the migration is completed

The Data migration report includes the following columns.



Data migration report

Rows	CIMPLICITY LOGS				HISTORIAN LOGS			
#Logs	Timestamp	point_id	_VAL	_QUALITY	Tagname	Value	Quality	QualitySub
1	3-MAR-2006 13:35:51.780	TANK1	75	44	TANK1	75	3	0
2	3-MAR-2006 13:35:57.110	HIST1	1213	172	HIST1	1213	3	0
3	3-MAR-2006 13:35:57.110	HIST2	5468	44	HIST2	5468	3	0
4	3-MAR-2006 13:35:57.110	HIST3	434	172	HIST3	434	3	0
5	3-MAR-2006 13:35:57.110	HIST4	15795	172	HIST4	15795	3	0
6	3-MAR-2006 13:35:57.110	HIST5	6454	44	HIST5	6454	3	0
7	3-MAR-2006 13:35:57.110	HIST6	3145	44	HIST6	3145	3	0
8	3-MAR-2006 13:35:57.110	TANK750	2133	44	TANK750	2133	3	0
9	3-MAR-2006 13:35:57.110	TANK810	17879	172	TANK810	17879	3	0
10	3-MAR-2006 13:35:57.110	TANK905	87987	172	TANK905	87987	3	0
11	3-MAR-2006 13:44:53.017	HIST2	4654	44	ECIMP HIST2	4654	3	0
12	3-MAR-2006 13:44:56.530	HIST5	124	172	ECIMP HIST5	124	3	0
13	3-MAR-2006 13:44:59.983	HIST6	2345	44	ECIMP HIST6	2345	3	0
14	3-MAR-2006 13:45:02.780	HIST3	654	172	ECIMP HIST3	654	3	0

CIMPLICITY columns are:

- Timestamp
- Point_ID
- _VAL
- _QUALITY
- _RES
- Project
- _ENG
- _ALRM

Historian columns are:

- Timestamp
- Tagname
- Value
- Quality
- QualitySub

Note: Historian quality is OPC quality.

Database Logging Management

About Database Logging Management

- ODBC configuration.
- Database management for SQL Server.
- About creating tables.
- Table characteristics.
- Database-disconnect-recovery.

Important: The Microsoft Access (As-Is product) ODBC driver is only supported on Intel-based computers.

ODBC Configuration

ODBC Configuration

When you install CIMPLICITY if you:

- Install the (MSDE) database server option, a Microsoft SQL-compatible database server, the ODBC driver will be SQL Server data sources.
- Do not install the server, CIMPLICITY redistributes ODBC using Microsoft data access components.

You do not need to purchase or configure additional software to use these drivers.

Tip: Install the database server because, as a SQL server, it provides better scaling when you want to go to a more robust SQL server to support Enterprise type solutions.

For improved performance and network extensibility, you may wish to log to a third-party database product such as Microsoft SQL Server or Oracle.

ODBC Drivers and Data Sources for Database Logging Overview

ODBC Drivers and Data Sources for Database Logging Overview

An overview of ODBC configuration options for database logging includes:

- ODBC drivers.
- CIMPLICITY default ODBC data sources.
- Custom ODBC data sources.
- ODBC configuration and moving projects.

ODBC Drivers for Database Logging

ODBC Applications use an ODBC driver to communicate with database applications. The drivers supported for use with the CIMPLICITY Database Logger are:

Driver	Description
Microsoft Access (As-Is product)	An entry-level database which requires no additional software to use.

Microsoft SQL Server or Microsoft SQL Server Express 2012, 2014, 2016	A high-performance database for larger applications. You must purchase SQL Server to use this driver. Supported SQL versions are: • SQL Server 2012, 2014, 2016 • SQL Server 2008 can be used but is no longer supported. **Important:** SQL Native Client is not currently supported.
Oracle ODBC Driver	A high-performance database for large applications.

Note: These drivers are supported on CIMPLICITY supportedoperating systems.

CIMPLICITY Default ODBC Data Sources

An ODBC Data Source is a specific configuration of an ODBC driver. When you install the CIMPLICITY Database Logger, several data sources are automatically configured:

Data Source	Description
CIMPLICITY Logging - Alarms	If you: • Installed the database server when you installed , this data source will use the Microsoft SQL Server driver and log onto the local SQL server. • Did not install the database server, the data source uses the Microsoft Access (As-Is product) driver to log to **CIMPLOG.MDB** in your distribution ARC directory.
CIMPLICITY Logging Points	If you: • Installed the database server when you installed , this data source will use the Microsoft SQL Server driver and log onto the local SQL server. • Did not install the database server, the data source uses the Microsoft Access (As-Is product) driver to log to **POINTLOG.MDB** in your distribution ARC directory.
CIMPLICITY SQL Server Logging	The data source uses the Microsoft SQL Server driver to log to an on-node (local) SQL Server.

Custom ODBC Data Sources

You may configure additional data sources on your computer, or customize the data sources that are configured by CIMPLICITY installation. However, you must observe the following restrictions:

- Use only supported ODBC drivers. Other drivers may not meet the data requirements of CIMPLICITY software.
- Because CIMPLICITY runs as a service, it must use system data sources.
- Remember that data sources are not part of your CIMPLICITY project, and will not automatically follow a project that is taken to a different node.
- Be aware of any driver-specific restrictions (below) which may apply.

Note: If you want to trend Logged or Logged Point data that is stored in a remote database, you must add a custom data source to each Viewer.

ODBC Configuration and Moving Projects

If you copy a project to another computer, the ODBC configuration associated with that project will not be automatically copied.

- If the project uses the default CIMPLICITY data sources, you may need to modify your configuration depending on whether the default data sources use SQL Server or Access (As-Is product).

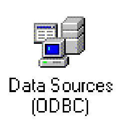
Note: Normally there will not be any additional configuration. However, if you install CIMPLICITY differently on one machine from a machine to which you move the project, you may need to adjust the new machine.

Example

You installed the database server on Machine A. However, you did not install the database server on Machine B. You moved the project to machine B. CIMPLICITY logging expects to log to Access (As-Is product). In fact it won't because ODBC is pointing to the SQL database server. You will need to re-configure Machine B to point to the correct data source.

- If the project uses custom data sources, these must be re-configured on the target machine.

SQL Server Data Sources Configuration

1. Click **Start** on the Windows task bar.
2. Select Settings>Control Panel.
3. Click the **ODBC Sources** icon 

.

   The ODBC Data Source Administrator dialog box opens.

4. Select the System DSN tab.
5. Select either:
   - CIMPLICITY Logging - Alarms or
   - CIMPLICITY Logging - Points.
6. Click **Configure**.

A first Microsoft SQL Server DSN Configuration wizard displays.

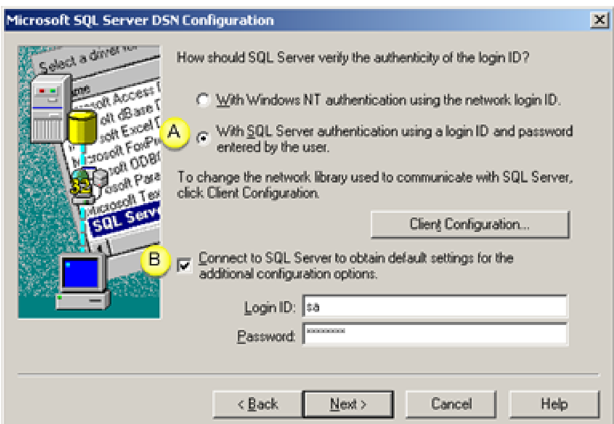
7. Configure the first Microsoft SQL Server DSN Configuration wizard as follows:
   a. Select the **Server** as follows:

For	Select
The local machine	One of the following: • `(Local)` • `Machine Name\Proficy`
A remote machine	The computer name from the drop down list.

   a. Click **Next**.

A second Microsoft SQL Server DSN Configuration wizard opens.

8. Configure the second Microsoft SQL Server DSN Configuration wizard as follows:



	Feature	Check	
A	Authenticate the login ID	With SQL Server authentication using a login ID and password entered by the user.	
B	Connect to SQL	Connect to SQL Server to obtain default settings for the additional configuration options.	
		Enter	Description
		Login ID	Valid for the selected SQL Server.
		Password	Required with the log in ID.

9. Click **Next**.

10. Finish configuration leaving the defaults on the remaining dialog boxes.

Microsoft Access (As-Is Product)Data Sources Configuration

1. Run the ODBC Administrator (from the ODBC program group) or select the ODBC option from the CIMPLICITY supportedWindows Control Panel.
2. Click **System DSN** to select the list of System Data Sources.
3. Click **Add...** to add a new data source.
4. Select Microsoft Access Driver from the list, and click **OK**.
5. Enter a unique **Data Source Name**.

   Use any combination of letters, numbers, and white space; but avoid punctuation characters.
6. Enter a Data Source Description, if desired.
7. Either:
    - Click **Select...** to select an existing database file, or
    - Click **Create...** to create a new .MDB file.
8. Specify a file name In the Select Database or New Database dialog, and click **OK**.
9. From the ODBC Microsoft Access Setup dialog, click **Options**.
10. Verify that **Exclusive** is not checked.
11. Configure separate data sources for CIMPLICITY Point Logging and Alarm Logging.

    Note: Due to file locking issues in the Microsoft Access driver, you must never request Exclusive access to a database that is used by CIMPLICITY for logging. In addition, the Point Logging and Alarm Logging data sources must refer to different database files.

Oracle Configuration Guidelines

- ODBC versions are neither forward nor backward compatible.
- Database aliases configured on the client node should also be configured on the server node.

Note: Configuring the aliases on both nodes helps avoid confusion about the database name.

- The Oracle server's IP address must be in the CIMPLICITY computer's HOST file if DNS or Wins resolution is not being used.
- In the ODBC Datasource Administrator dialog box, use the Microsoft ODBC for Oracle Driver and enter the database alias name in the **Server Name** field.
- Starting with Oracle 9i, the Oracle services are started by default.

Oracle for CIMPLICITY Configuration

Oracle for CIMPLICITY Configuration

Step 1 *(page 315)*	Turn on Allow Service to Interact with desktop.
Step 2 *(page 315)*	Setup CIMPLICITY software to log to an Oracle database.

Step 1. Turn on Allow Service to Interact with Desktop

1. Open the Control Panel.
2. Double-click **Services** 

.

   The Services dialog box opens.

3. Either:
   - Double-click CIMPLICITY Service in the list of services, or
   - Click **Startup**....

   The (CIMPLICITY) Service dialog box opens.

4. Select **System Account** in the Log On As box.
5. Check **Allow Service to Interact with Desktop**.
6. Click **OK**.

   The (CIMPLICITY) Services dialog box closes.

7. Click **Close**.

   The Services dialog box closes.

Step 2. Setup CIMPLICITY Software to Log to an Oracle database.

1. Install the Oracle9i or 10g Server on a clean machine.

2. Install HMI SCADA - CIMPLICITY 7.0 on a different machine that has a CIMPLICITY supported operating system.

3. Install the Oracle 9i or 10g Client on the CIMPLICITY 7.0 Server.

   Important: The required Oracle client version is v9.2.08 or above. Any version lower than 9.2.08 is not supported.

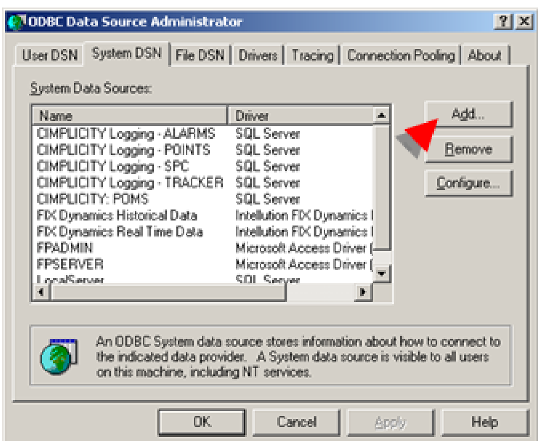
   Note: The installation is a standard Oracle installation.

   Consult Oracle documentation for details.

4. Open the ODBC Data Source Administrator through the Microsoft Control Panel.

5. Select the DSN tab.

6. Click Add.



   A Create New Datasource dialog box opens.

7. Create a data source for one of the following.

   Important: Make sure the driver is installed before you attempt to configure the data source.

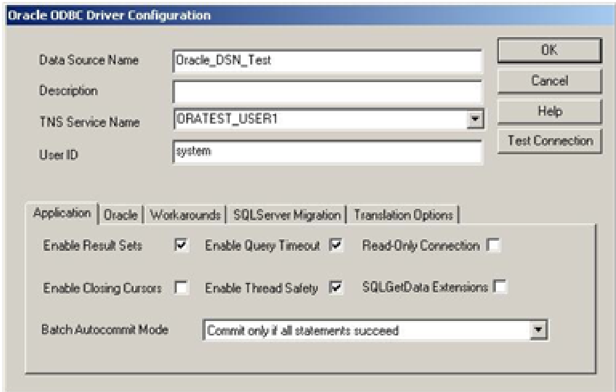
Oracle OraHome92 driver
Microsoft ODBC for Oracle driver
Data Direct 5.1 for Oracle driver

   Oracle OraHome92 driver

   a. Select the Oracle In OraHome92 driver in the Create New Data Source list of drivers.
   b. Click Finish.

   An Oracle ODBC Driver Configuration dialog box opens.

a. Enter the following.



Option	Description	
Data Source Name	Name that identifies the data source, e.g. Oracle_DSN_Test.	
Description	(Optional) Description to help users identify the data source.	
TNS Service Name	<HostName>_<ServerName> selected from the drop down list. Where	
	HostName =	SID created while installing Oracle 9i/10g server on Server machine.
	ServerName =	Oracle server name.
User ID	Name must be an authorized user, created during the Oracle 9i/10g server installation.	

a. Click Test Connection.

An Oracle ODBC Driver Connect dialog box opens, displaying the **Service Name** and **User Name**.

a. Enter a valid password (that goes with the user ID).

Note: The password was created during the Oracle 9i/10g Server Installation.

a. Click OK.

A Connection Successful message box opens.

a. Click OK.

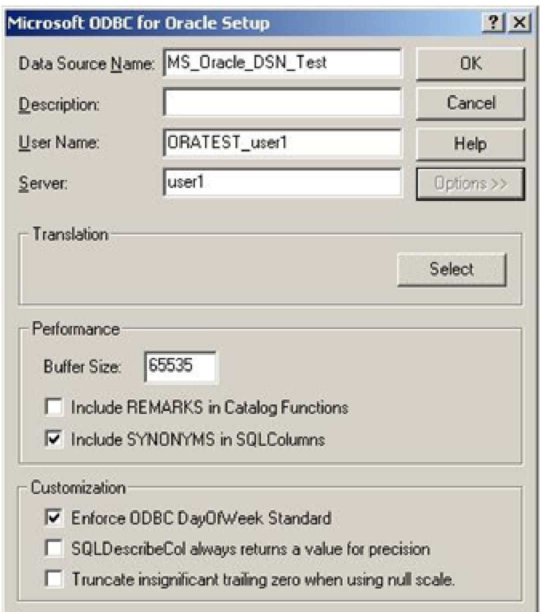
A new DSN is created and listed in System DSN list.

Microsoft ODBC for Oracle driver

1. Select the Microsoft ODBC for Oracle driver in the Create New Data Source list of drivers.
2. Click Finish.

A Microsoft ODBC for Oracle Setup dialog box opens.

1. Enter the following.



Field	Description	
Data Source Name	Name that identifies the data source, e.g. Oracle_DSN_Test.	
Description	(Optional) Description to help users identify the data source.	
User Name	Name must be an authorized user, created during the Oracle 9i/10g server installation, e.g. <HostName>_<ServerName>	
	HostName =	SID created while installing Oracle 9i/10g server on Server machine.
	ServerName =	Oracle server name.
Server	Name of the Oracle Server.	

A new DSN is created and listed in System DSN list.

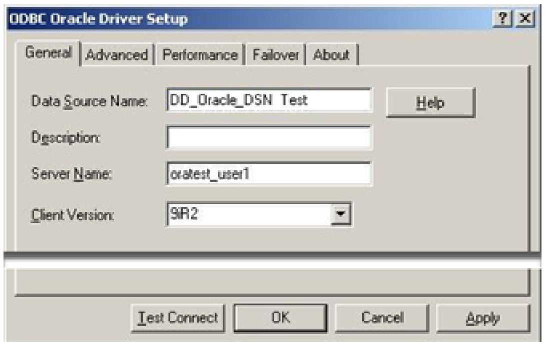
Note: A valid password will be required when the ODBC data source is selected in the Database Logger's Logging Properties *(page 196)* dialog box.

Data Direct 5.1 for Oracle driver

1. Select the Data Direct 5.1 Oracle driver in the Create New Data Source list of drivers.
2. Click Finish.

An ODBC Oracle Driver Setup dialog box opens.

1. Enter the following.



Field	Description	
Data Source Name	Name that identifies the data source, e.g. Oracle_DSN_Test.	
Description	(Optional) Description to help users identify the data source.	
Server Name	Oracle server name, e.g. <HostName>_<ServerName>	
	HostName =	SID created while installing Oracle 9i/10g server on Server machine.
	ServerName =	Oracle server name.
Client Version	Oracle version used for the client.	

1. Click Test Connect.

A Log on to Oracle dialog box opens displaying the **Server Name**.

1. Enter the following.

Field	Description
User Name	Authorized User ID.

Password	Valid password (that goes with the user ID).

1. Click OK.

A Connection Successful message box opens.

1. Click OK.

A new DSN is created and listed in System DSN list.

Oracle Software Requirements

The Oracle client software must be installed in the same computer as the CIMPLICITY software.

Database Management for SQL Server

Database Management for SQL Server

Database management for SQL server includes:

- Configuring additional locks.
- Checking if Microsoft SQL Server Service is running.
- Handling default database server limitations.
- Handling the transaction log.

Additional Locks Configuration

If the SQL Server generates an error that looks like this on rollover:

Error #S1000 SQLSERVER has run out of locks. Re-run your command when …

You can do the following to increase the number of locks:

1. Open the SQL Enterprise Manager in the Microsoft SQL Manager program group.
2. From the **Server** menu, select **SQL Server**, then **Configure**. The Server Configuration/Options dialog opens.
3. In the Server Configuration/Options dialog, select the Configuration page.
4. Find **locks** in the list.
5. Increase the number of locks.
6. Select **OK**.
7. Exit the SQL Enterprise Manager.

The change takes place the next time the SQL Server is restarted.

Check if Microsoft SQL Server Service is Running

1. Expand the Database Logger in the Workbench left pane.
2. Double-click Service Manager.

   The SQL Server Service Manager dialog box appears.

3. Click the **Start/Continue** button, if it is enabled, to start the Microsoft SQL Server.

Default Database Server Limitations

The default Database Server has the following limitations:

- The maximum per database is 2 Gigabytes.
- Performance begins to degrade after 5 connections.

Connections can include:

1. Alarm Logging (`DLRP`).
2. Point Logging (`PTDL`).

   When forwarding data (in store and forward), forward is a separate connection to the database.

3. Alarm Logging forward connection.
4. Point Logging forward connection.

   If the table is enabled for bulk insertion, bulk insertion requires a dedicated connection to the database in order to work. As a result, maintenance actions for bulk insert tables will need to allocate a separate connection. Once the action is completed, it will free the connection.

5. Bulk insertion maintenance action.
6. Bulk insertion forward connection.

   If there are too many tables enabled for bulk insertion there, resulting in too many connections at once, there will be a performance degrade.

   If your system configuration regularly exceeds 5 connection, it is recommended that you upgrade to CIMPLICITY SQL, which is a more robust server.

Transaction Log

All SQL Server transactions are logged to the Transaction Log file. The Transaction Log contains an audit trail by default and can grow to be quite large. To conserve disk space consider doing one of the following:

- Use a truncated Transaction Log by default.
- Use a CIMPLICITY maintenance action truncate the Transaction Log periodically.

Truncate the Log as a System Default

1. Open the SQL Enterprise Manager in the Microsoft SQL Manager program group.
2. From the Manage menu, select Databases. The Manage Databases dialog opens.
3. In the Manage Databases dialog, double click on the database used by the CIMPLICITY Database Logger. The Edit Database dialog opens.
4. In the Edit Database dialog, select the Options tab.
5. Select the **Truncate Log on Checkpoint** option.
6. Click **OK**.
7. Close the Manage Databases dialog.

   Result: The change takes place the next time the SQL Server is restarted.

   Truncate the Log as a CIMPLICITY Maintenance Action

8. Open the Database Logger in the CIMPLICITY program group.
9. Select a table from the list of tables.
10. Open the Table Properties dialog for the table.
11. Select the Maintenance Actions tab.
12. In the **Command Action** field, enter:

    dump transaction DATABASENAME with no_log

    This command will dump the Transaction Log without saving the log.

    If you want to save the log to a file, see your SQL Server documentation on how to do this.

13. Select the Maintenance Events page.
14. Configure the event that will trigger the action.

    This change takes place the next time you update the project's configuration and then restart the project.

    It is recommended truncating the log as a system default.

    todo: To set truncate the log as a system default:

15. Open the SQL Enterprise Manager.
16. Select the database you are logging to.
17. Right-click on the database.
18. Select Properties from the popup menu.
19. Select the Options tab.
20. Make sure that **Truncate Log On Checkpoint** is checked.

Create Tables

Create Tables

When a project is started, the Database Logger creates any tables you have configured for that project based on the attributes you have specified. This means you don't have to know anything about databases to log data from your CIMPLICITY project.

Specifically, when a project starts, the Database Logger will:

- Attempt to create or repair any missing or damaged databases.
- Create any missing tables.
- Create any missing columns in the tables.

Microsoft Access (As-Is Product)Note

To successfully create or repair tables, the Database Logger requires an exclusive lock on the Access database. If another application is using the Access database, the Database Logger cannot exclusively lock the database, and it will terminate. If this happens, the following message will be logged in the Status Log for the project:

Unable to repair database

This problem occurs, for example, if a user is running reports on the Access database while a project is being started.

Table Characteristics

The Database Logger tables have the following characteristics:

Column Definitions

The following columns are defined for each type of log file:

- Data logging tables contain a timestamp and Point ID column, plus an additional column for each table and point attribute you have selected. The number of columns is independent of the number of points in the table. For example, if you have a table that logs the point value and previous value for all points configured for data logging, the table will have four columns.
- Alarm, Event, and Application logging tables have columns that are specific to that table, plus an additional column for each table attribute you have selected. All table rows include a timestamp column and an auto-increment sequence number that ensures that each row is unique.
- Group logging tables contain a timestamp column, plus an additional column for each table and point attribute you have selected. For example, if you have a table logging the value and alarm state of five points, the table will have eleven columns.

Note: Points with Engineering Units conversion are stored in floating point format

Key Definitions

The following keys are defined for each type of log file:

- Group logging tables have a unique primary key index on the timestamp column. If you have selected the project name table attribute, the primary key index also includes the project name column.
- Data logging tables have a primary key index on the joined timestamp and Point ID columns, and a secondary index on the timestamp alone. If you have selected the project name table attribute, the primary key index also includes the project name column.
- Alarm, Event, and Application logging tables have a primary key index on the joined timestamp and sequence number columns, and a secondary index on the timestamp alone. If you have selected the project name table attribute, the primary key index also includes the project name column.

Important: Since the Microsoft Access (As-Is product) format and Oracle do not support sub-second timestamp data, you cannot log points to a Data or Group table at sub-second rates. Attempting to do so will cause the duplicate-keyed records to be dropped from the database.

Database-Disconnect-Recovery

Database-Disconnect-Recovery

If the Database Logger loses its connection to a database, it generates a DB_CONN_DOWN alarm. When it successfully reconnects to the database, the alarm is cleared.

You can control the amount of:

	Time to wait between connection requests.	Reconnect wait period.
	Data to be saved locally then forwarded to the database when the reconnection is made.	Store and forward.

Reconnect Wait Period

You can use the **Reconnect wait period** to define the amount of time to wait between connection requests whenever the Database logger loses its connection to the database. Each time a reconnect request fails, a DB_CONN_DOWN alarm is generated.

You can define a separate wait period for the Alarm Logging and Point Logging.

After the database connection is reestablished, if you have also enabled the **Store and forward** feature, all of the stored data is automatically forwarded to the database.

Note: The DB_CONN_DOWN alarm is not stacked by default, but you can change it in Alarm Configuration.

Store and Forward

Store and Forward

The **Store and Forward** feature is available for supported databases. This feature lets you write records for Alarm Logging and Point Logging to storage files whenever the Database Logger loses its connection to the database.

When the Database Logger successfully reconnects to the database and **Store and Forward** is enabled when the Database Logger starts, the Database Logger checks for stored data files (that is, .SQL files) in the storage directory. If it finds files, it generates a DB_START_FORWARD alarm then checks the files for stored data. If there is no stored data in the files, the Database Logger logs the following message to the Status Log:

No forward data found in <store_path>

and clears the **DB_START_FORWARD** alarm.

If there is data to forward, the Database Logger goes through each file and forwards the data to the database. When all the data has been forwarded, or if the Database Logger loses its connection to the database, it clears the DB_START_FORWARD alarm and stops processing the storage files.

Note: Stored data and new logged data are sent in parallel to the database, and all Maintenance actions are ignored while data is being stored.

After a file is forwarded to the database, the Database Logger sends the following message to the Status Log:

Forward: <file_name> succ: <no_suc> fail <no_fail>

If there are failures, the following message will also be logged:

Forward: See log <filename> for errors.

If the Database Logger cannot process a SQL statement in a storage file, it generates a log file with the same name as the storage file and the extension .LOG. The log file contains the statement forwarded to the database and the error message returned by the database.

If the Database Logger encounters no problems when forwarding data, no .LOG file is generated.

Note: If you have **Store and Forward** enabled and storage files exist when you shut down the Database Logger and you then disable **Store and Forward**, the Database Logger will not forward records from the storage files when it restarts.

You can open a storage file and examine its contents in, for example, Notepad.

Important: Store and Forward is supported only for local Microsoft Access (As-Is product) databases, not remote Access databases.

Store and Forward Enabled

When you enable this feature, you can choose to store an unlimited, or a maximum number of records in the storage file for the database. You can define separate storage options for Alarm Logging and Point Logging.

- If you choose to store unlimited records, the Database Logger continues to save data records until it runs out of disk space. The following message is then logged in the Status Log:

Unable to write to store and forward file, <filename>

and the Database Logger continues to try to save additional records.

- If you choose to store a maximum number of records, the Database Logger saves the configured number of records then stops, and the following message is logged to the Status Log.

Number of stored records exceeded maximum <max_no>

The records are stored in one or more files in a storage directory. The number of files depends on the maximum number of records you select and the maximum number of records per file.

Store and Forward File Names

For Alarm Logging and Point Logging storage files, the default directory is **%SITE_ROOT%\arc** , and the default maximum number of records per file is 100.

The filenames for the storage files have the following format:

```
<prcnam>_<dbmsid>_<date>_<time>
.sql
```

Where

<prcnam> is process name

Example

For point logging (group and data logs), the process name is **MAC_PTDL** .

For alarm logging, the process name is **MAC_DL**

<dbms_id> is the database's DBMS ID,

<date> is the date and

<time> is the time the file was created.

Example

MAC_PTDL_$PTLOG_20000913_174548.SQL

How Maximum Number of Records Works

Once the maximum number of records have been stored, the Database Logger will not store any more records until it can forward some of the currently stored records.

For example, you configure Store and Forward for a maximum of five records.

- After the Database Logger loses its connection to the database, it stores five records and logs the "Number of stored records exceeded" message to the Status Log:
- The Database Logger then reconnects to the database and is able to forward three of the stored records before it loses its connection again.
- The Database Logger will now store three records and log the "Number of stored records exceeded" message to the Status Log:

If the Database Logger is shut down, and there are still records in the storage files, when the Database logger restarts, these records are not counted against the maximum. Under these conditions, you can have more than the maximum number of records stored to disk.

Note: You specify the maximum number of stored records on the Connection tabs in the Database Logger Logging Properties dialog box.

Change Directory and Record Defaults

The default directory for **Store and Forward** files is **%SITE_ROOT%\arc** , and the default number of records per file is 100. You can change both of these defaults.

Important: Please consider carefully when changing the maximum number of records per stored file. Setting too high a number will result in unmanageable files, while setting too low a number will cause excessive disk I/O.

1. Click Tools>Command Prompt on the Workbench menu bar.
2. At the command prompt type:

   cd master

   idtpop dbms_def

   notepad dbms_def.idt

   The file looks like this:

```
|-* IDT file generated by IDTPOP utility v1.0
* RECORD: DBMS_DEF ODBC Database Definitions
*   0 DBMS_ID                          Name of the database/service
 for logging
*   1 driver                           ODBC name of DMBS driver
*   2 connect_string                   Connection string for ODBC
 driver
*   3 wait_period                      reconnect wait period
*   4 wait_units                       reconnect wait period units
*   5 store_forward                    0=off 1=on
*   6 store_path                       path of stored file
*   7 recs_per_file                    max recs per stored file
```

```
*   8 limited                           0=unlimited store else max num
 of recs
*
$LOGGING|Microsoft Access Driver <*.mdb>|-
DSN=CIMPLICITY Alarm Logging;UID=admin;DBQ=BSM_ROOT:\arc
\CIMPLOG.MDB|-1|0|0||100|0.000000
$LOGGING|Microsoft Access Driver <*.mdb>|-
DSN=CIMPLICITY Point Logging;UID=admin;DBQ=BSM_ROOT:\arc
\POINTLOG.MDB|-1|0|0||100|0.000000
```

Note that the Alarm Logging and Point Logging databases have separate records.

3. To define a new path for the stored records for a database, enter the pathname in the **store_path** field for the database. You can set a different path for each database.
4. To define a new maximum number of records per stored file for a database, enter the new maximum in the **recs_per_file** field for the database. You can set a different maximum for each database.
5. Exit Notepad and save the updated file.
6. Enter the following commands to regenerate the data file and close the command window:

   scpop dbsm_def

   exit
7. Shut down your project, perform a **Configuration Update**, and then restart the project.

This change to takes effect when the project restarts.

CIMPLICITY Log Files

About CIMPLICITY Log Files

CIMPLICITY software consists of a large number of interrelated programs. Not all programs are run interactively by users. The non-interactive programs, therefore, cannot report problems directly to a user's terminal. Instead, they use the various log files available on the system.

Project log files

You can find these log files in your project's log directory. All messages related to a project are logged to these files.

These files include:

CIMPLICITY Version	File	File Name
Lower t than CIMPLICITY v9.0	Status Log	cor_recstat.clg
CIMPLICITY v9.0 and higher	Status Log	
All CIMPLICITY versions	All CIMPLICITY processes other than user processes	*.out and *.err
All CIMPLICITY versions	Backup for all CIMPLICITY processes other than user processes	*.out_<n> and *.err_<n> .

Note:

- If a project has been upgraded from an earlier CIMPLICITY version both the cor_recstat.clg and cor_recstat.cl2 files will be available.

If cor_recstat.cl2 was used a new file should be cor_recstat_1.cl2, etc.

- The message lengths *(page 337)* for the status log files are as follows.

cor_recstat.clg	80 characters
cor_recstat.cl2	1024 characters.

- If a long point ID that is being used in CIMPLICITY applications is deleted, error messages displayed in a cor_recstat.cl2 log that are associated with that long point ID will display the internally generated short point ID (32 characters or less) instead of the long point ID created by the user.

Notes:

- Examples of applications that might be using the long point ID are CimView, Trend, an event in the Event Editor.
- The message will report that there is a request for a nonexistent point; the reported point ID will be the short point ID.

System log files

You can find the **system** log files in the ..\<CIMPLICITY Installation>\log directory under your main CIMPLICITY directory. All system messages are logged to these files.

These files include:

CIMPLICITY Version	File	File Name
Lower t than CIMPLICITY v9.0	System Status Log	cor_recstat.clg

CIMPLICITY v9.0 and higher	System Status Log	cor_recstat.cl2
All CIMPLICITY versions	Router	w32rtr.out and w32rtr.err

Use the following to view log files.

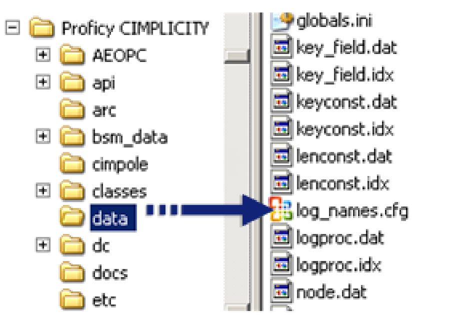
Use	To view information in
Status Log Viewer	Status Log files for projects or the system.
Notepad	.out, .out_<n>, .err, and .ERR_<n> files for projects.

Set the Maximum Records Allowed in the Cor_Recstat.clg/ Cor_Recstat.cl2

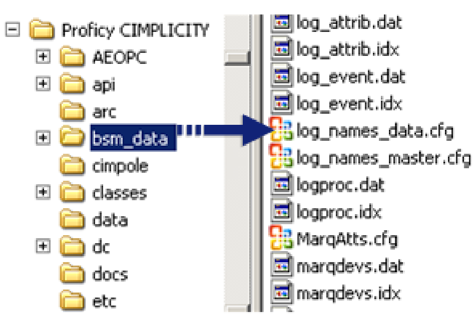
You can set the maximum number of records allowed in the status log file.

1. Make sure that no CIMPLICITY processes are running.
2. Open each of the following files.

...\<CIMPLICITY Installation>\data\Data\log_names.cfg

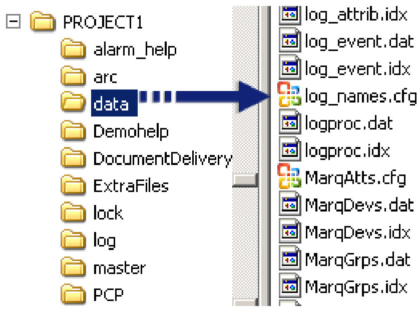


....\<CIMPLICITY Installation>\bsm_data\log_names_data.cfg



...\<Project Installation>\data\log_names.cfg

Note: This file needs to be edited in each project's directory, for all involved projects.



3. Add the following lines to each of the files identified above.

`LOG_MAXRECORDS|S|default|10|1000`

`LOG_MAXRECORDS|P|default|10|1000`

Where

Component	Description	
LOG_MAXRECORDS	Command for maximum records.	
S	System	
P	Project	
default	Makes the entries the default.	
10	Length of the string that can be changed.	
	Default	10
1000	Maximum number of records that can be listed in the CIMPLICITY Log Viewer window. **Note:** The view can be filtered *(page 342)* so the records you need to review will be listed.	
	Default	1000
Comment	The maximum number of records has been successfully tested up to 16,000. **Note:** When there are large records the Log Viewer takes time to populate this high number.	

Note: If you have an existing cor_recstat.clg or cor_recstat.cl2 and:

Increase maximum records	The existing records are retained.
Decrease maximum records	If the log contains a greater number of records, the cor_recstat.cl2 (or cor_recstat.clg) is re-initialized.

Examine Status Log, Output, and Error Files

Examine Status Log, Output and Error Files

You can resolve problems by examining the various log files in order to identify

- Errors.
- Error sources.
- Other relevant logging information.

Step 1 (page 333)	Open the Status Log Viewer.
Step 2 (page 337)	Review Status Log Viewer tools.
Step 3 (page 341)	Work in the Status Log Viewer
Step 4 (page 352)	(Optional) Check System Output and Error Files
Step 5 (page 354)	(Optional) Check Process Output and Error Files

Step 1. Open the Status Log Viewer

Step 1. Open the Status Log Viewer

Option 1.1 (page 333)	Open a project Status Log.
Option 1.2 (page 336)	Open the system Status Log.

1.1. Open a Project Status Log

CIMPLICITY provides several methods to open a project Status Log.

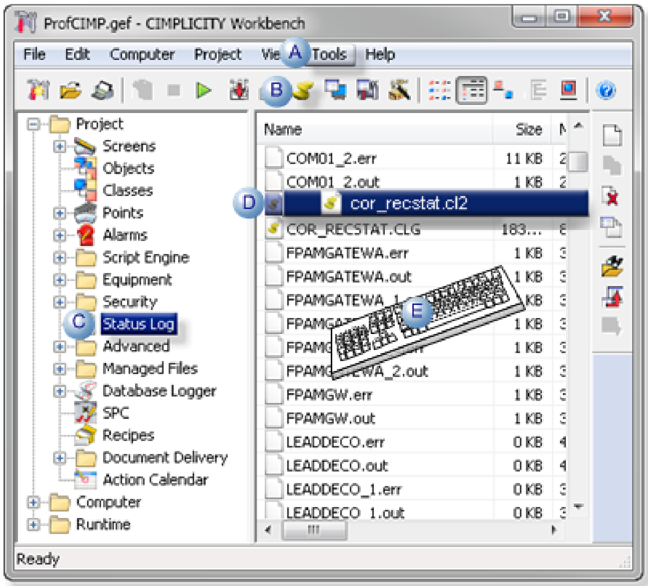
- Workbench
- Start menu

Workbench

1. (Optional) Select Project>Status Log in the Workbench left-pane.

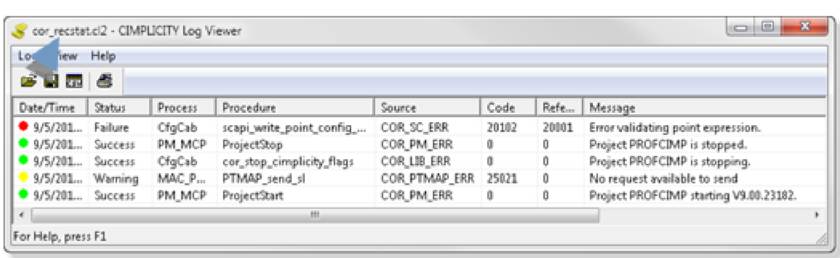
2. (Optional) Select cor_recstat.clg in the Workbench right-pane.

3. Do one of the following.



A	Click Tools>Status Log on the Workbench menu bar.	
B	Click the Status Log button on the Workbench toolbar.	
D	In the Workbench right-pane:	
	Either	Or
	Double-click COR_RECSTAT.cl2.	a. Right-click COR_RECSTAT.CL2. b. Select Open on the Popup menu.
	Note: If the project has been upgraded from a version lower than CIMPLICITY v8.2 COR_RECSTAT.CLG will also be listed in the Workbench right-pane.  	
E	Press Ctrl+L on the keyboard.	

Result: The Project status log displays in the Status Log viewer.



Note: The cor_recstat.cl2 file will open automatically when the Status Log button or Tools>Status Log menu is used.

An additional option to open the COR_RECSTAT.clg *(page 330)* log file is as follows.

4. Right-click COR_RECSTAT.CL2.

5. Select Open on the Popup menu.

6. Do either of the following.
   - Select Log>Select Log on the CIMPLICITY Log Viewer menu bar.
   - Click the Open File button on the CIMPLICITY Log Viewer toolbar.

   The Find CIMPLICITY Log browser opens.

7. Select the project's Log folder.

8. Change the .cl2 to .clg extension in the **File Name** field.

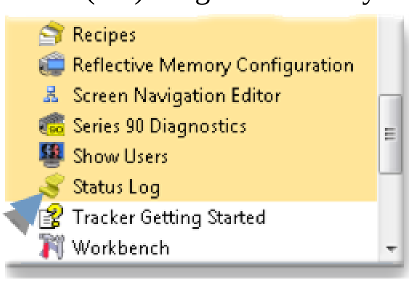
9. Click Open.

   Result: The project's COR_RECSTAT.clg file will open in the CIMPLICITY Log Viewer.

   Start Menu

10. Click Start on the Windows Task bar.

11. Select (All) Programs>Proficy HMI SCADA - CIMPLICITY version>



   Result:

   - If the System Log was the last log opened.

   The System Log opens.

12. Click Log>Select Log on the Status Log window menu bar.

13. Select COR_RECSTAT.cl2 in the ...\<project name>\Log directory.

The selected status log opens in the Status Log window.

Tip: Change the .cl2 to .clg in the Find CIMPLICITY Log browser>**File Name** field you want to open the COR_RECSTAT.clg *(page 330)* file.

- If a project status log was the last log opened.

The last used status log opens.

Either review the open log or select another.

1.2. Open the System Status Log

CIMPLICITY provides several methods to open the system Status Log.

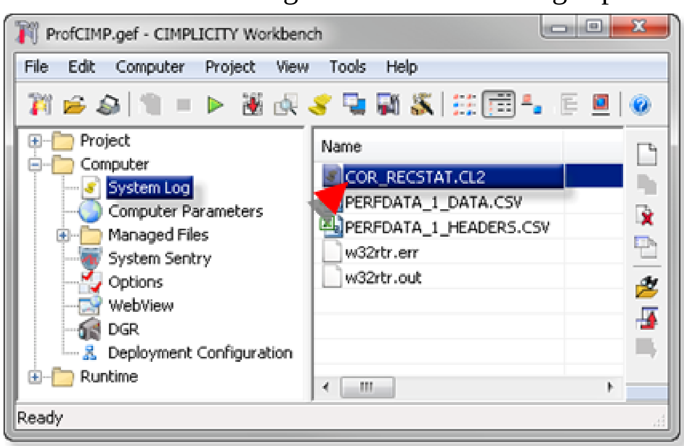
- Workbench
- Start menu

Workbench

1. Select **Computer>System Log** in the Workbench left-pane.

2. Select COR_RECSTAT.CL2 in the Workbench right-pane.

3. Do one of the following in the Workbench right-pane.



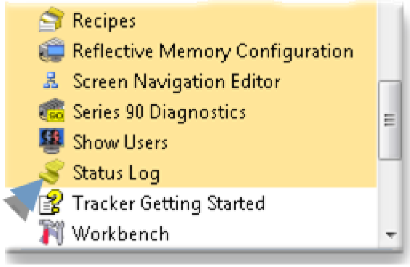
- Double-click COR_RECSTAT.CL2.
- Right-click COR_RECSTAT.CL2 select **Open** on the Popup menu.

Result: The system Status Log opens when you use either method.

Start Menu

4. Click Start on the Windows Task bar.

5. Select (All) Programs>Proficy HMI SCADA - CIMPLICITY version> Status Log.



Result:

- If the System Log was the last log opened.

The System Log opens.

- If a project status log was the last log opened.

The last used status log opens.

Click Log>View System Log on the Status Log window menu bar.

The System Log opens.

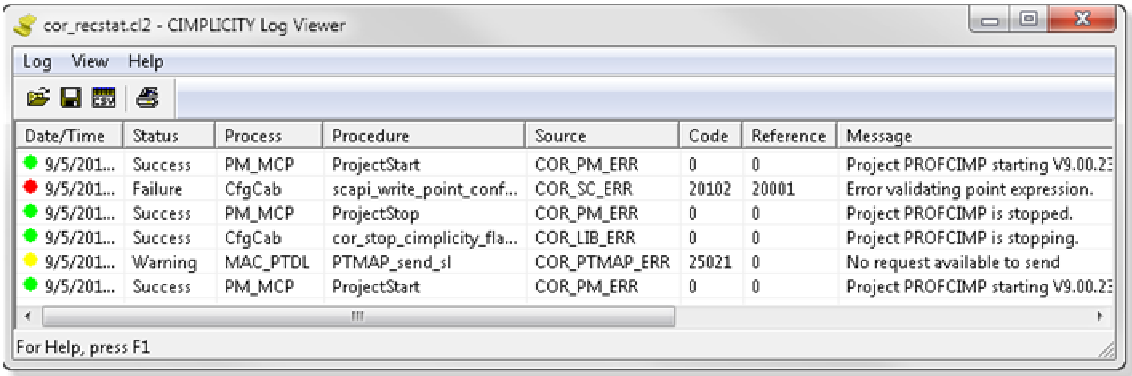
Step 2. Review Status Log Viewer Tools

Step 2. Review Status Log Viewer Tools

2.1 (page 337)	Status Log columns.
2.2 (page 339)	Status Log menus, toolbars, shortcut keys.

2.1. Status Log Columns

The CIMPLICITY Log Viewer screen displays the following information for each record that it finds in the status log file:



Column	Description		
Date/Time	Date and time the message was logged.		
Status	Type of message. This can be: "Failure", "Warning" or "Success".		
		Red	Failure
		Yellow	Warning
		Green	Success
Process	Name of the process that generated the log message.		
Procedure	Name of the procedure that generated the log message.		
Source	Symbolic name for the error class.		
Code	Primary value used by software for expressing the type of error.		
Reference	Number that can be used to determine the location of the condition that caused the error.		
Message	Explanation of the condition that caused the log message.		
	Message Maximum Length	1024 characters (cor_recstat.cl2) **Important:** The message length in cor_recstat.clg, which is available in upgraded projects, continues to be 80 characters.	

Note: Guidelines

- The **Procedure**, **Source**, **Reference**, and **Code** fields are primarily for use by CIMPLICITY support and should be reported if you are contacting CIMPLICITY support for troubleshooting assistance.
- The list of messages is initially sorted in descending order (newest to oldest) by **Date/Time**.

You can click on any of the column title buttons to sort the messages alphanumerically by that message attribute.

Example

To view all the messages generated by the `MAC_PTDL` process, click the Process header.

Result: All the messages generated by MAC_PTDL will be grouped together.

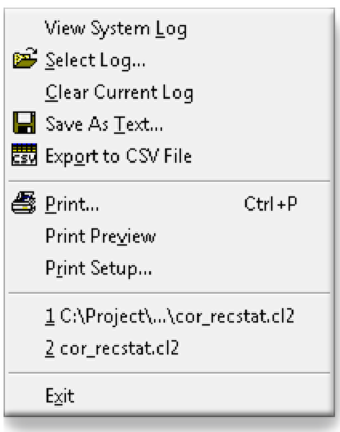
2.2. Status Log Menus, Toolbars, Shortcut Keys

- Status Log menus.
- Toolbar buttons.
- Shortcut keys.

Status Log Menus

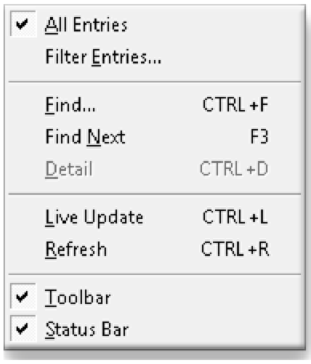
- Log menu.
- View menu.
- Help menu.

Log Menu



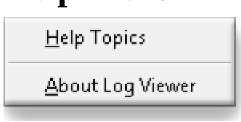
Option	Description
View System Log	Displays the system log file.
Select Log	Selects the log file you want to view.
Clear Current Log	Clears all messages from the current log file.
Save as Text...	Saves the current contents of the Status Log file as a text file.
Export to CSV File	Exports contents of the open log to a .csv file format. to a csv (but is separated by |) and the
Print...	Prints the contents of the Status Log file to a printer.
Print Preview...	Shows you what your print request will look like.
Print Setup...	Configures the printer, form and page setup for your Status Log print request.
Exit	Exits the Status Log function.

View Menu



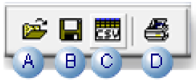
Option	Description
All Entries	Displays all entries in the Status Log.
Filter Entries...	Filters the entries by Status, Process, Procedure, Source, Reference, or Code.
Find...	Locates an entry based on the **Status**, **Process**, **Procedure**, **Source**, **Reference**, and **Code** information that you enter.
Find Next	If you already selected **Find**, this command locates the next entry based on the information you entered. If you have not entered any **Find** parameters, this command locates the next entry in the log file.
Detail...	Displays all of the detailed information for the selected log message.
Live Update	Automatically updates the Status Log list as messages are generated. You cannot display message details when in this mode.
Refresh	Refreshes the Status Log list on request. This option is available when **Live Update** is disabled. In addition, you can display message details when in this mode.
Toolbar	Enables/Disables the display of the Toolbar at the top of the screen.
Status Bar	Enables/Disables the display of the Status Bar at the bottom of the screen.

Help Menu



Option	Description
Help Topics	Displays the main Help window for the Status Log Viewer.
About Log Viewer...	Displays program information, version number, and copyright for Status Log Viewer.

Toolbar Buttons



A	Open	Selects a log file to view.
B	Save	Saves the active document.
C	Export	Exports the log to a .csv file.
D	Print	Prints the active document.

Shortcut Keys

Shortcut	Description
Ctrl+P	Prints the contents of the Status Log file.
F3	Finds the next Status Log message that fits the search criteria.
Ctrl+D	Displays details for the selected message.
Ctrl+L	Enables/Disables Live Updates.
Ctrl+R	Refreshes the display.

Step 3. Work in the Status Log Viewer

Step 3.4. Work in the Status Log Viewer

The Status Log Viewer lets you locate messages that have a warning or failure status, and provides information to help determine the source of the problem.

3.1 *(page 342)*	Filter status log messages.
3.1 *(page 343)*	Find status log messages.
3.3 *(page 343)*	Display status log message details.
3.4 *(page 345)*	Print the status log file.
3.5 *(page 347)*	Save the status log as a text file.

3.6 (page 349)	Export to a CSV file.
3.7 (page 350)	Select a different status log to view.

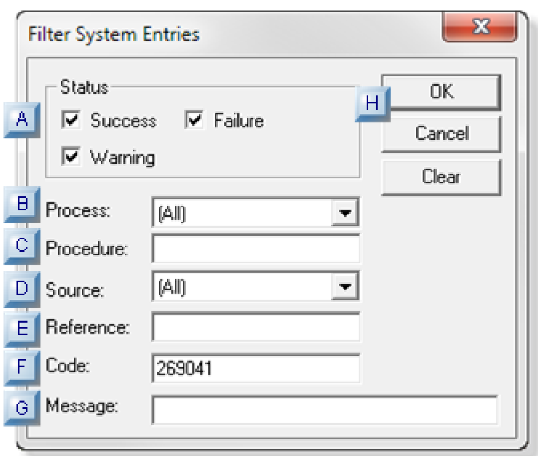
3.1. Filter the Status Log Messages

The Status Log can be filtered to display a particular set of messages. The filter selection types correspond to the CIMPLICITY Log Viewer columns.

Click View on the status log Viewer menu bar; select Filter Entries.

The Filter System Entries dialog box opens.

Options are as follows.



rect 0, 236, 22, 259 (page 343)
rect 0, 210, 22, 233 (page 343)
rect 0, 186, 22, 209 (page 343)
rect 0, 161, 22, 184 (page 343)
rect 0, 133, 22, 156 (page 343)
rect 0, 109, 22, 132 (page 343)
rect 0, 54, 22, 77 (page 342)
rect 219, 46, 241, 69 (page 343)

	Option	Description
A	Status	Check one or more message status types. • Success • Warning • Failure **Important:** If you do not check at least one status type, no messages will display..

B	Process	CIMPLICITY process that may require attention (e.g. CfgCab). **Tip:** Click the drop-down list to display the list of processes that are currently in the log file and can be selected.	
C	Procedure	Internal procedure (software module) name/names. Enter either of the following: • An entire procedure name (e.g. PTMAP_add_pt_list). • The first n characters (e.g. PTMAP). **Note:** Procedures, which are more specific than processes, can help narrow the search for a problem.	
D	Source	Selected error class name (e.g. COR_PM_ERR). **Tip:** Click the drop-down list to display the list of sources that are currently in the log file and can be selected.	
E	Reference	Exact number to determine the location of the condition that caused the error (e.g. 105).	
F	Code	Exact primary value used by software for expressing the type of error (e.g. 25009).	
G	Message	Message text associated with a log message. **Note:** The entry must be exactly as it appears in the status log.	
H	Buttons	Buttons do the following.	
		OK	Log Viewer window displays with the filtered list of messages. If no messages match the filter, the window is blank.
		Cancel	Closes the dialog box without filtering.
		Clear>OK	Clears filtering and re-displays all status log messages

3.2. Find Status Log Messages

1. The Find Status Log Messagesdialog box closes.
2. The next message in the status log that matches the filter is highlighted.
   If no message is found, the highlight remains at the current message.

   Tip: Once you have set the filters in the Find System Entry dialog box and found the first message that matches the filters, you can do either of the following:

   - Select View>Find Next on the status log menu bar.
   - Press F3

The next message in the configured direction that matches the filter is highlighted.

3.3. Display Status Log Message Details

Details for any message can be viewed in a read-only dialog box.

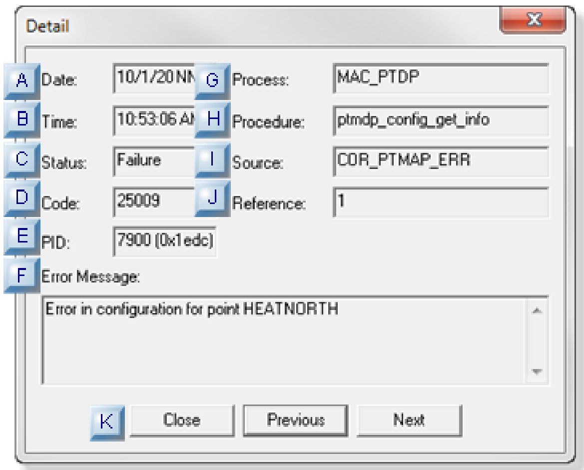
Do any of the following for a selected message.

- Double-click the message.
- Press Ctrl+D.

• Select View>Detail on the CIMPLICITY Log Viewer menu bar.

The Detail dialog box opens.

Message details are as follows.



rect 53, 250, 74, 271 [(page 345)]
rect 120, 108, 141, 129 [(page 345)]
rect 120, 84, 141, 105 [(page 345)]
rect 120, 60, 141, 81 [(page 345)]
rect 120, 36, 141, 57 [(page 345)]
rect 1, 157, 22, 178 [(page 345)]
rect 1, 132, 22, 157 [(page 344)]
rect 1, 107, 22, 132 [(page 344)]
rect 1, 82, 22, 107 [(page 344)]
rect 1, 58, 22, 83 [(page 344)]
rect 1, 35, 22, 60 [(page 344)]

This dialog box displays the following information

	Field	Description
A	Date	Date the message was generated.
B	Time	Time the message was generated.
C	Status	The status of the reported process/procedure/source. • Failure • Warning • Success
D	Code	Primary value used by the software for expressing the type of error.
E	PID	**Process ID** of the process that logged the message. **Note:** The number is displayed in both decimal and hexadecimal format.

F	Error Message	Explanation of the condition that caused the message to be logged.	
G	Process	Process that logged the message (e.g. MAC_PTDP).	
H	Procedure	Software module that logged the message (e.g. ptmdp_config_get_info).	
I	Source:	Error class name (e.g. COR_PTMAP_ERR).	
j	Reference	Number passed by the procedure to assist in determining the location of the condition that caused the error.	
K	Buttons	The buttons do the following.	
		Close	Close the dialog box and return to the Log Viewer screen.
		Previous	Display the previous message in the log file. **Note:** If you are at the first message in the file, you will be asked if you want to continue from the end.
		Next	Display the next message in the log file. **Note:** If you are at the last message in the file, you will be asked if you want to continue from the beginning.

Important: If you are contacting CIMPLICITY support for troubleshooting assistance about a particular message or set of messages, the information in these fields is of critical importance:

- Status
- Code
- Process
- Procedure
- Source
- Reference
- Error Message

3.4. Print the Status Log File

Printing a status log is similar to printing any other text document. Many printing features, of course, depend on the printer type and model.

Printing can be initiated from either of the following.

- Print Preview window.
- Print dialog box.

Print Preview Window

A status log can be printed through the Print Preview window.

A benefit to using print preview is being able to determine what pages should be printed.

Click Log>Print Preview on the CIMPLICIT Log Viewer menu bar.

The Print Preview window opens; sections are as follows.

rect 1, 18, 24, 40 (page 346)
rect -1, 147, 22, 169 (page 346)
rect 36, 339, 59, 361 (page 346)

	Item	Description	
A	Toolbar	Standard print preview toolbar buttons include:	
		Print	Opens the Print dialog box
		Next Page	Displays the page after the current page.
		Prev Page	Displays the page before the current page.
		Two Page	Switches to a 2-page display.
		Zoom In	Magnifies the current page.
		Zoom Out	Reduces the current page.
		Close	Closes the Print Preview window; returns to the CIMPLICITY Log Viewer window.
B	Content	Status log content in print page format.	
C	Status Bar	Number(s) of the displayed page(s).	

Print Dialog Box

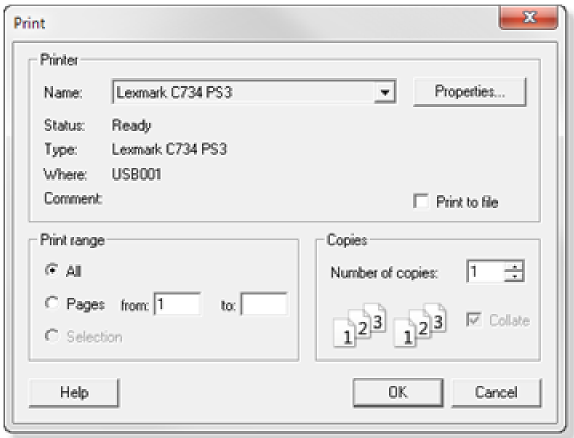
The status log can be printed directly through the CIMPLICITY Log Viewer window or through the Print Preview window.

Do one of the following.

- Click Log>Print on the CIMPLICITY Log Viewer menu bar.
- Press Ctrl+P on the keyboard.
- Click Print on the Print Preview window toolbar.

The Print dialog box opens.

Basic options are as follows.



Option	Description
Name	Currently selected printer. Click the dropdown list to select from the list of available printers.
Print Range	Print all of the status log messages or select pages. **Notes** • If you applied a Filter *(page 342)* to the Status Log display, only the filtered messages will print. • Use the Print Preview window to determine what pages to print.
Copies	Number of copies to print.
Print to file	Check to save the printout in a file in addition to printing.
Properties	Opens dialog box to display and change printer properties.
OK	Prints the status log or selected pages in the status log.
Cancel	Cancels print.

Note: The header line on each page of the printout shows the computer name, the full path name for the Status Log file and a page number.

3.5. Save the Status Log as a Text File

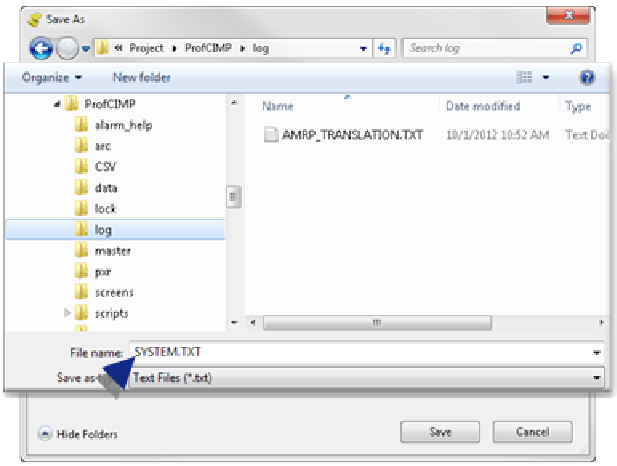
The Status Log files are binary files, and cannot be viewed directly.

You can, however, copy the contents of a Status Log file to a text file that you can view with a text editor (e.g. Notepad).

This feature can be applied to both cor_Recstat.clg and cor_recstat.cl2 files.

1. Click Log>Save as Text on the CIMPLICITY Log Viewer menu bar.

   The Save As dialog box opens.

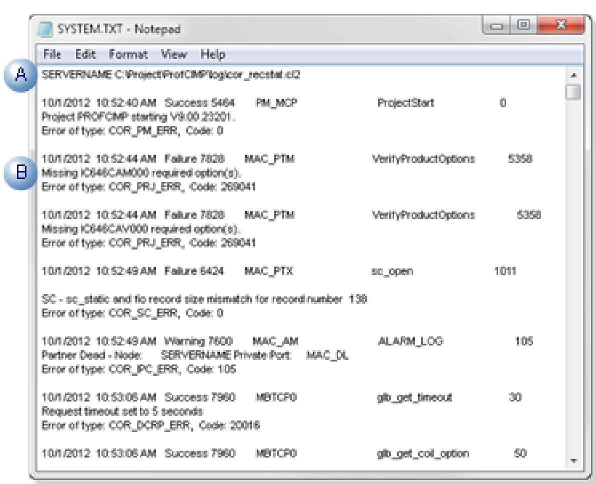
   

2. Select the folder in which the file will be saved.
3. Enter a name in the **File name** field.

   Default: SYSTEM.TXT.

4. Click OK.

The text file displays the following.



A	First line of the text file shows the computer name and the full path name for the Status Log file.
B	Other lines display messages that were listed in the CIMPLICITY Log Viewer window when the file was saved. **Note:** If you applied a Filter *(page 342)* to the Status Log display, only the displayed messages were saved to the text file.

Tip: You can

- Configure the CIMPLICITY project to create and automatically update an ASCII text version of the unfiltered Project Status Log.

Edit log_names.cfgin the DATA folder under the current folder.

Add the following line:

`LOG_ASCII|P|default|1|T`

Add the line anywhere in the file, e.g. under Standard Logical Names,

Make sure there is no asterisk at the start of the line.

The next time you start the project, CIMPLICITY will create a text file version of this status log, COR_STATUS.LOG, in the project's Log folder.

CIMPLICITY keeps this file updated with the latest project status information.

- Log system status to a text file with the same name in the ...\<CIMPLICITY Installation>\log folder.

Add the same line, above, to the file LOG_NAMES.CFG in the ...\<CIMPLICITY Installation>\data folder.

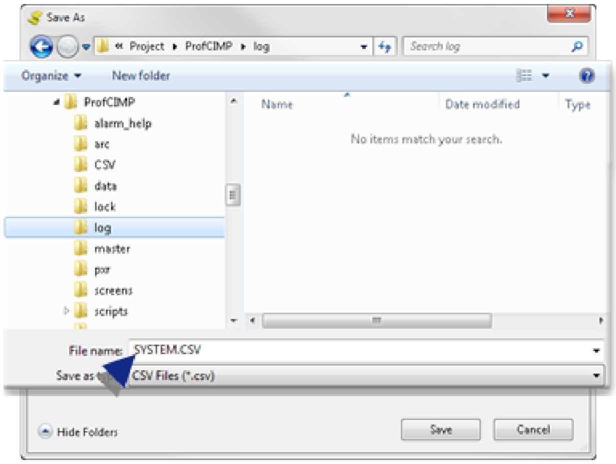
3.6. Export to a CSV File

You can, export the contents of a cor_recstat.cl2 log to a CSV format.

Note: If you applied a Filter *(page 342)* to the Status Log display, only the displayed messages will be exported.

1. Click Log>Export to CSV File on the CIMPLICITY Log Viewer menu bar.

   The Save As dialog box opens.



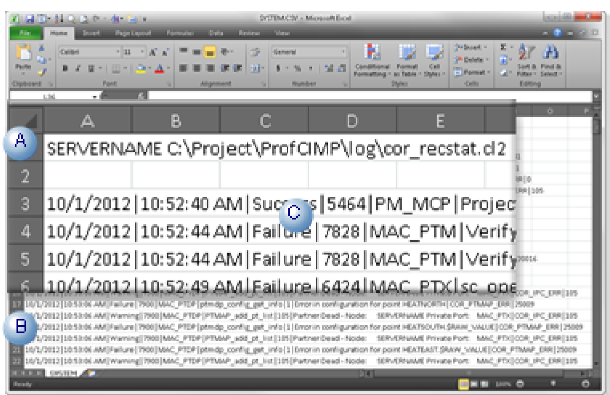
2. Select the folder to which the file will be exported.

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

   Default: SYSTEM.CSV.

4. Click OK.

   Result:: The CSV file, which can be opened in Excel, displays the following.

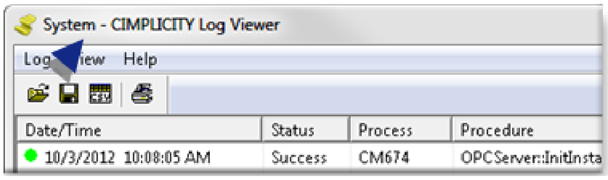


A	First line in the file shows the computer name and the full path name for the Status Log file.
B	Other lines display messages that were listed in the CIMPLICITY Log Viewer window when the file was exported. **Note:** If you applied a Filter *(page 342)* to the Status Log display, only the displayed messages were saved to the text file.
C	Pipes (|)separate information to emulate column breaks.

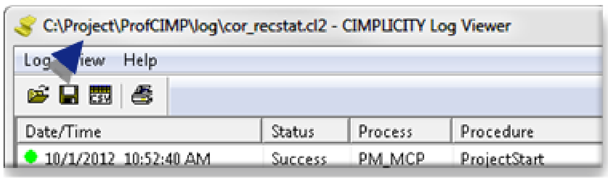
3.7. Select a Different Status Log to View

CIMPLICITY includes a:

- System Status Log file.



- Project Status Log file for each project.



When launched the Status Log Viewer opens either the System Status Log file or the Project Status Log for the current project depending on the method used *(page 333)* . You can, however, select a different log file for display at any time (e.g. system status, current project status, or the status log for another project).

Do one of the following:

- Select Log>View System Log on the CIMPLICITY Log Viewer menu bar.

The System - CIMPLICITY Log Viewer displays.

- Select Log>Select Log on the CIMPLICITY Log Viewer menu bar.

A Find CIMPLICITY Log File dialog box opens.

The CIMPLICITY Log Viewer can display CIMPLICITY Status Log files .cl2 and older .clg files.

You can search the directory structure on any drive you are connected to for log files.

The default Status Log file name is COR_RECSTAT.CL2.

Status Log files are generally found in the ..\<CIMPLICITY Installation>\log directory and in each project log directory.

1. Locate the file in the ..\<CIMPLICITY Installation>\log directory or a project log directory.
2. Click OK.

Step 4. (Optional) Check System Output and Error Files

Step 4. (Optional) Check System Output and Error Files

The current set of w32rtr.out and w32rtr.err files for the Router may contain additional information if you are experiencing problems with your CIMPLICITY software.

1. Stop the router.

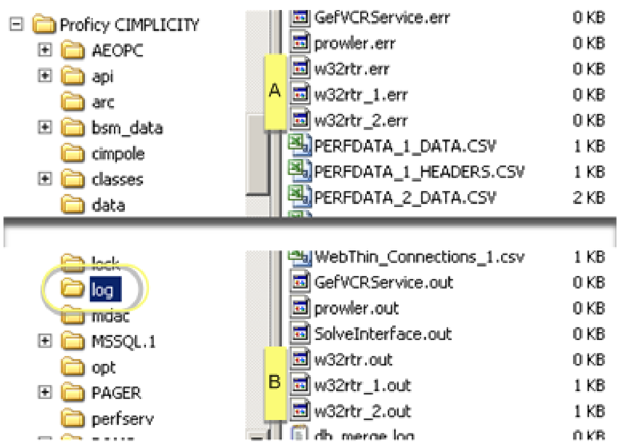
   The router must be stopped to examine these files.

   Warning: All projects on the computer stop when you stop the router.

2. Open the Windows Explorer.
3. Select the Proficy CIMPLICITY\Log.

   Note: The default directory is C:\Program Files (86)\Proficy\Proficy CIMPLICITY\Log.

4. Check for:



A	w32rtr.err files with a size greater than 0. These files have data.	
	Example	
	w32rtr.err	Newest .err file
	w32rtr_1.err	Next newest backup
	w32rtr_2.err	Oldest backup.
B	w32rtr.out files with a size greater than 0. These files have data.	
	Example	
	w32rtr.out	Newest .out file

	w32rtr_1.out	Next newest backup
	w32rtr_2.out	Oldest backup

5. Use Notepad to look at the contents of these files (right-click and select Open With... Then select Notepad in the dialog box that appears).

Tip: Drag any one of these files from the Workbench **Computer**>**System Log** right-pane into a text editor, e.g. Notepad, to view the contents.

Process Names in the Status Log

Processes that log messages in the Status Log include:

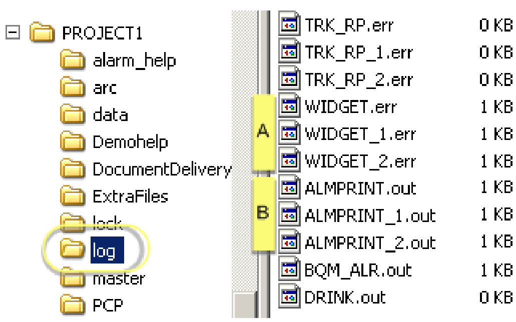
Name	Description
`AB_ETH`<n>	Allen-Bradley Ethernet where <n> is the port number
`AV<n>`	Alarm Viewer session where <n> is the session number
`CfgCab`	Configuration
`COM<n>`	Serial Devcom where <n> is the port number
`CM<n>`	CIMVIEW session where <n> is the session number
`CW<n>`	CWSERV session where <n> is the session number
`MAC_AM`	Alarm Manager
`MAC_DL`	Data Logger
`MAC_DYN`	Dynamic Configuration Manager
`MAC_EMRP`	Basic Control Engine
`MAC_PTDL`	Point Data Logger
`MAC_PTDP`	Derived Point Processor
`MAC_PTM`	Point Manager
`MAC_PTX`	Point Translator
`PB`	Point Bridge
`PDC_DS`	Tracker Decision Control Data Server
`PM_MCP`	Main Control Program
`PRT_DC`	Tracker Data Collector
`PRT_DS`	Tracker Data Server
`PRT_GRD`	Tracker Graphic Display
`PRTCLNT`	PRT Client Resident process

`PRTCNT`	PRT Count Resident process
`RCODB_RP`	RCO Resident process
`TADB_RP`	TADB Resident process
`TCPIP<n>`	Series 90 Ethernet where <n> is the port number
`TCPRD<n>`	TCP/IP Redundancy where <n> is the port number
`TRK_RP`	Tracker Resident Process
`TRKEA`<id>	Tracker Collector Service Where id is the process ID
`W32RTR`	Message router

Step 5. Check Process Output and Error Files

Step 5. (Optional) Check Process Output and Error Files

1. Stop the project that has the problem.
2. Open Windows Explorer.
3. Select the <project name>Folder>Log.
4. Check for:



A	.err files with a size greater than 0. These files have data.	
	Example	
	TCPIP0.err	Newest file for this process.
	TCPIP0_1.err	Next newest backup.
	TCPIP0_2.err	Oldest backup.
B	.out files with a size greater than 0. These files have data.	
	Example	
	COM01.out	Newest file for this process.
	COM01_1.out	Next newest backup.

	COM01_2.out	Oldest backup.

Note: Each .err and .out file can have up to two backups.

5. Use Notepad to look at the contents of these files.

Note: You can only examine the newest.err and .out files while the project is not running. You can examine backup .err and .out files while the project is running.

Tip: Drag any one of these files from the Workbench **Project>Status Log** right-pane into a text editor, e.g. Notepad, to view the contents.

Basic CIMPLICITY Filenames and Associated Programs

Filename	Associated Process
AB_ETH<n>	Allen-Bradley Ethernet device communications where <n> is the port number
COM0<n>	Serial device communications where <n> is the port number
MAC_AM	Alarm Manager
MAC_DL	Data Logger
MAC_DYN	Dynamic Configuration
MAC_EMRP	Basic Control Engine
MAC_PTDL	Point Data Logger
MAC_PTDP	Virtual Point Processor
MAC_PTM	Point Manager
MAC_PTX	Point Translation
MAC_UR	User Registration
PB	Point Bridge
PCIM<n>	Genius device communications where <n> is the port number
TCPIP<n>	Series 90 TCP/IP device communications where <n> is the port number.
TCPRD<n>	Series 90 TCP/IP redundancy where <n> is the port number.
W32RTR	Message Router

In addition to these files, you may also find files for alarm log printers and device communication drivers.

Chapter 3. System Management

Server to Viewer File Deployment

About Server to Viewer File Deployment

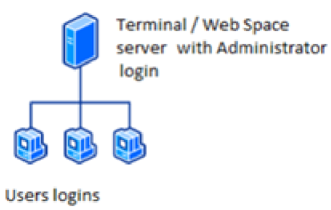
The CIMPLICITY file deployment system provides the tools to automatically synchronize *(page 357)* selected files, including CimView screens, global scripts and global parameter files, from a CIMPLICITY deployment server to its viewers.

- Deployment core files.
- Deployment configuration steps.

Deployment core files

Server to viewer deployment is based on two core files.

File	Description
gefdepl.dplcfg	A configuration file: • Is created or modified when the deployment configuration is saved in the CIMPLICITY Configuration Editor. • Includes the details of the files that should be deployed, from where they should be deployed, to whom they should be deployed, and when they should be deployed. • Will be copied to the Viewers that are included in the synchronization. **NOTE:** The location on the Viewer is in the installation\Data folder: <localhost>\Program Files (x86)\Proficy\Proficy CIMPLICITY\Data

File	Description
cimsync.exe	• Cimsync.exe is a standalone executable running under user context. • When starting this executable, the user must be logged into the machine for the synchronization to occur. • Cimsync is located on each Viewer included in the synchronization, in the exe folder: **<localhost>\Program Files (x86)\Proficy\Proficy CIMPLICITY\exe** • In Terminal/Web Space servers, the user must log in with administrator credentials before launching the cimsync.exe file, and must remain logged in for the synchronization to occur. Only users connected to the Terminal/Web Space servers can see the changes.: 

Deployment configuration steps

Step Number	Description
Step 1 *(page 360)*	Plan the deployment configuration
Step 2 *(page 365)*	Configure deployment on the server.
Step 3 *(page 406)*	Set up a deployment viewer.

Deployment Overview

A deployment server can hold one or more folders with files that are automatically deployed to one or more viewers whenever they are modified. The deployment insures that the viewers automatically have the latest file version.

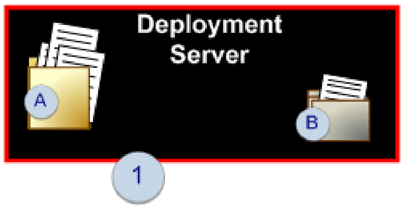
A summary overview of one deployment server synchronizing files with one viewer is as follows.

1 *(page 358)*	Deployment server setup.
2 *(page 358)*	Viewer setup.

3 (page 359)	CimSync.exe source files deployment.
4 (page 359)	CimSync.exe configuration file deployment.

Deployment server setup

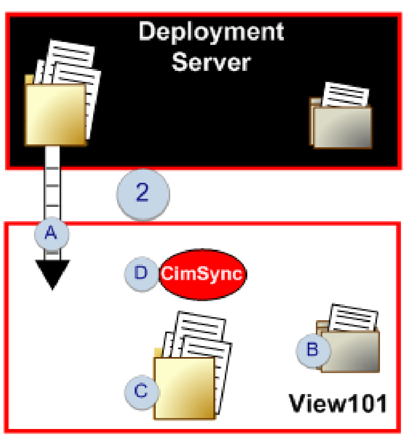
The deployment server requires two basic components.



A	A shared folder with files that are periodically updated and need to be deployed to one or more viewers. **Note:** The shared folder will be accessed by the Viewer by a UNC (page 359) path that is identified (page 375) in the Deployment Configuration Editor.
B	A shared folder with a configuration file, gefdepl.dplcfg, that lists the details of the deployment configuration. **Note:** Gefdepl.dplcfg is created on a CIMPLICITY server or Viewer and is always saved to the installation \Data directory. ...\Program Files\Proficy\Proficy CIMPLICITY\Data. However, it can be used on any server designated as the deployment server. • The deployment server does not have to be a CIMPLICITY server. • The configuration file location is entered in the file when it is being configured.

Viewer setup

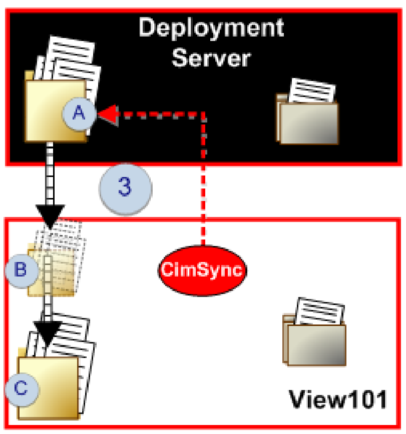
The viewer requires four basic components.



A	A UNC path to the server's shared files folder. **Notes** • The viewer must have read access to the deployment server source files folder. • You can also map a drive to the folder; however it is recommended that you use a UNC path, which is not dependent on each viewer.
B	A local folder with a configuration file, gefdepl.dplcfg, that lists the details of the deployment configuration. **Note:** On the Viewer, this file must always be located in the installation \Data directory. ...\Program Files\Proficy\Proficy CIMPLICITY\Data.
C	A local folder that holds the files for the viewer to use.
D	A file, CimSync.exe, that implements the deployment. **Note:** This file is installed with CIMPLICITY.

CimSync.exe source files deployment

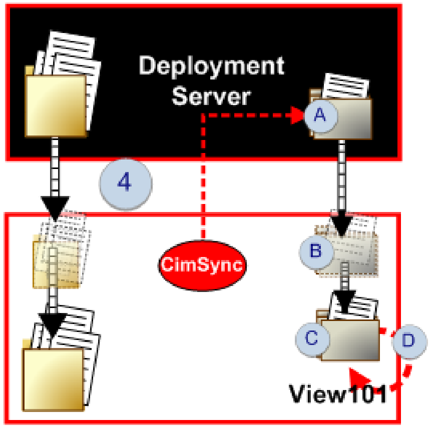
CimSync.exe does the following.



A	Detects changes or additions to files using the UNC path to the deployment server's source files folder.
B	(After a specified time period) Deploys the files to a temporary folder on the Viewer. **Note:** The temporary folder insures that all the source files are transferred correctly and provides a more stable environment than the network for them to be deployed. The files are deployed at one time to the target folder or folders. Updating all of the involved files at one time will insure that the integrity of any dependence among them is maintained.
C	Deploys the file to the viewer's target file folder where it overwrites existing files and adds new files.

CimSync.exe configuration file deployment

CimSinc.exe does the following.



A	Detects a change in the gefdepl.dplcfg file using the UNC path to the deployment server's configuration file folder.
B	(If the viewer is set to have the configuration file synchronized with the deployment server) Deploys it immediately a temporary folder on the viewer. **Note:** The temporary folder insures that all the source files are transferred correctly and provides a more stable environment than the network for them to be deployed. The files are deployed at one time to the target folder or folders. Updating all of the involved files at one time will insure that the integrity of any dependence between the configuration file and other files is maintained.
C	Overwrites the existing gefdepl.dplcfg version in the viewer's installation \Data directory.
D	Reads the new configuration file version and uses the new parameters from that point on.

Step 1. Plan the Deployment Configuration

Step 1. Plan the Deployment Configuration

The key to developing a straight-forward deployment system is to map out the system's viewer/server requirements and specifications before beginning configuration.

Planning includes determining specifications for:

1.1 (page 360)	Plan configuration for selected viewers.
1.2 (page 363)	Plan configuration for a 'COMMON' Viewer

Step 1.1. Plan Configuration for Selected Viewers

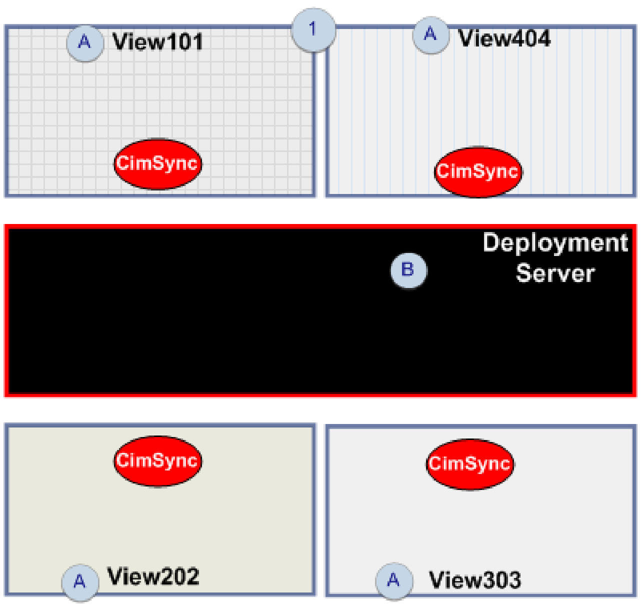
The key to developing a straight-forward deployment system is to map out the following.

1 (page 361)	Identify the Deployment Server and Selected Viewers.

2 (page 361)	Identify what files require synchronization on selected viewers.
3 (page 362)	Identify what Viewers should have the configuration file automatically synchronized.

Identify the deployment server and selected viewer names.

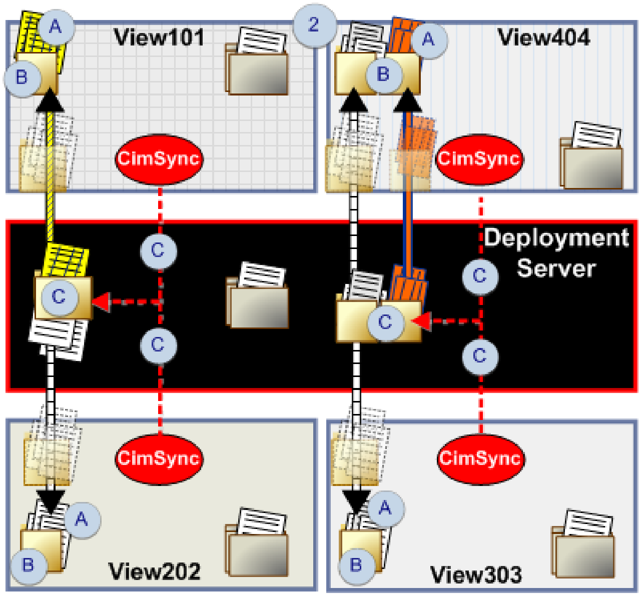
Make a note of the names of the viewers that will be singled out for configuration and the deployment server name.



A	Each viewer that will be singled out in the deployment must be identified with the exact name identified on the Computer Name tab in the Windows System properties dialog box.
B	The deployment server name must be the exact name identified on the Computer Name tab in the Windows System properties dialog box.

Identify files and folders that will be synchronized.

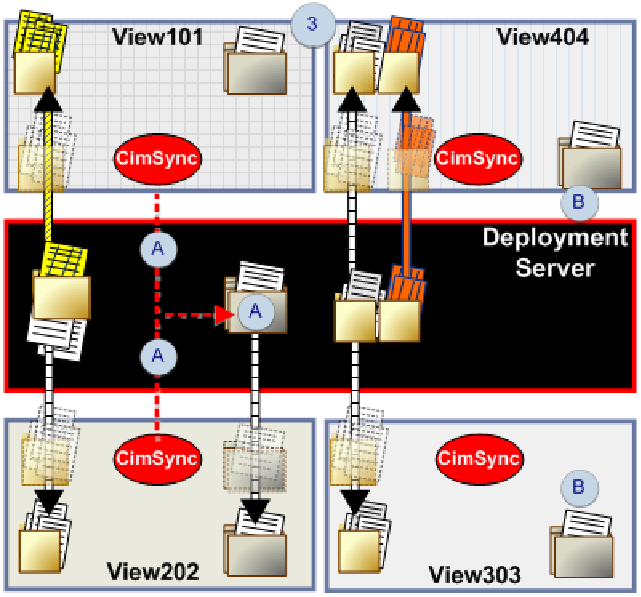
Identify the following.



A	Exactly what files and/or file types on the viewer will require synchronization.
B	The target folder path on the viewer that holds the identified files. There can be more than one local path on a viewer. **Note:** The temporary *(page 359)* folder on the viewer is automatically created.
C	Each shared folder on the deployment server that holds the source files that will be deployed to the viewer. • There can be more than one folder on the server. • Subfolders that can optionally hold files to be deployed.

Identify what Viewers should have the configuration file automatically synchronized.

Identify the following for each viewer.



A	The UNC path on the deployment server for viewers that should have the configuration file, gefdepl.dplcfg, automatically updated. **Notes** • The viewer must have read access to the Server folder to enable synchronization. • The target folder on the viewer is always in the Installation \Data directory. • The temporary *(page 360)* folder on the viewer is automatically created.
B	Viewers can have a configuration file that can only be updated manually because it will not be synchronized with the server.

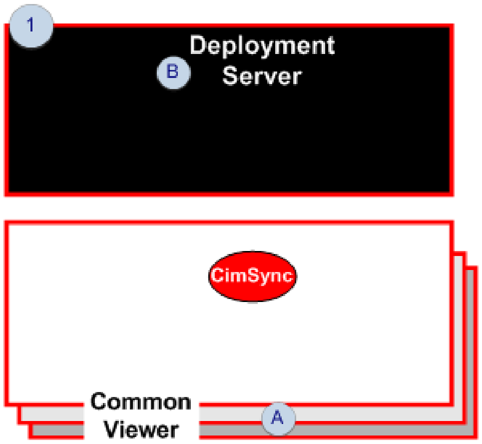
Note: Make sure the Viewers have enough hard drive space to support deployment.

Step 1.2. Plan Configuration for a 'COMMON'Viewer

When the Viewer specifications are planned out, there may be several viewers that are the same. These viewers can be grouped under a "COMMON' Viewer designation.

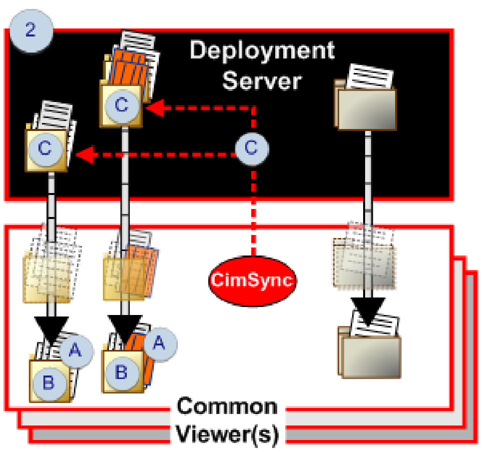
1 *(page 364)*	Group together viewers that have the same specifications.
2 *(page 364)*	Identify files and folders that will be synchronized.
3 *(page 365)*	Synchronize the configuration file.

Group together viewers that have the same specifications.



A	Viewers grouped under a 'COMMON' Viewer do not have to be identified by name.
B	The deployment server name must be the exact name identified on the Computer Name tab in the Windows System properties dialog box.

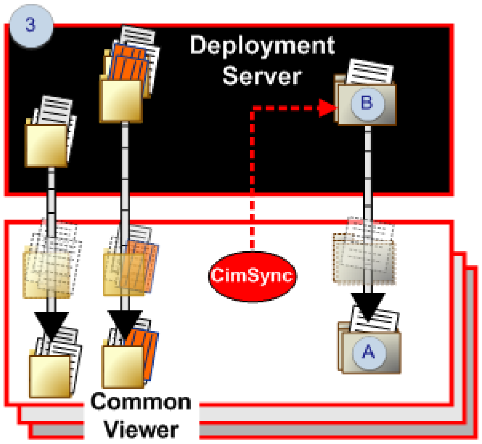
Identify files and folders that will be synchronized.



A	Exactly what files and/or file types on the common viewers will require synchronization.
B	The local (target) folder path on the viewers that hold the identified files. The local path must be the same for every viewer included in the synchronization. **Note:** The temporary *(page 359)* folder on the viewer is automatically created.
C	Each shared folder on the deployment server that holds the source files that will be deployed to the viewer. • There can be more than one folder on the server. • Subfolders that can optionally hold files to be deployed.

Synchronize the configuration file.

A 'COMMON' viewer should use an automatically synchronized configuration file.



A	The target folder on the viewer is always in the Installation \Data directory.
B	The UNC path on the deployment server for viewers that should have the configuration file, gefdepl.dplcfg, automatically synchronized with the server. **Notes** • The viewer must have read access to the Server folder to enable synchronization. • The temporary *(page 360)* folder on the viewer is automatically created.

Step 2. Configure Deployment on the Server

Step 2. Configure Deployment on the Server

When the deployment configuration has been mapped out, the organization and parameters can be identified and written to the gefdepl.dplcfg deployment configuration file.

CIMPLICITY provides a user interface, the CIMPLICITY Configuration Editor, in which you can identify the configuration and save it to the gefdepl.dplcfg file.

Step 2.1 *(page 366)*	Open the CIMPLICITY (Deployment) Configuration Editor.
Step 2.2 *(page 371)*	Create a new configuration file.
Step 2.3 *(page 374)*	Identify the deployment server and file.

Step 2.4 *(page 378)*	Identify 'COMMON' Viewer parameters and folders.
Step 2.5 *(page 395)*	Identify a selected viewer's parameters and folders.
Step 2.6 *(page 404)*	Complete configuration on the server.

Step 2.1. Open the CIMPLICITY (Deployment) Configuration Editor

Step 2.1. Open the CIMPLICITY (Deployment) Configuration Editor

- Open the CIMPLICITY (Deployment) Configuration Editor
- Review the CIMPLICITY (Deployment) Configuration Editor tools.

Open the CIMPLICITY (Deployment) Configuration Editor

Do one of the following.

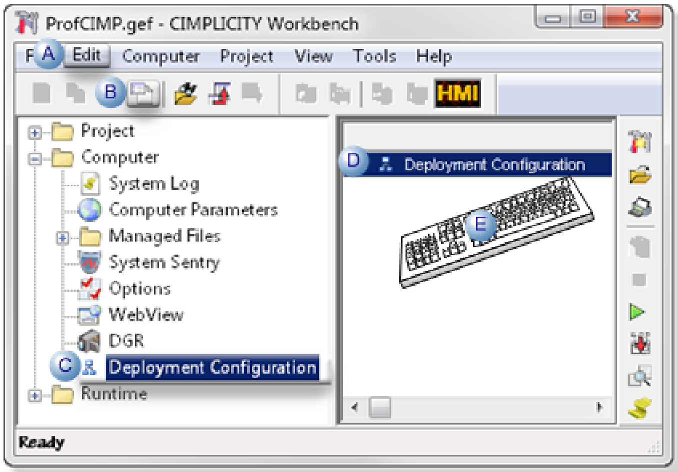
- Start menu
- Workbench

Start Menu

1. Click Start on the Windows Task bar.
2. Select All Programs>HMI/SCADA CIMPLICITY version>Deployment Configuration.

   Workbench

3. Select Computer>Deployment Configuration in the Workbench left pane.
4. Select Deployment Configuration in the right pane.
5. 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:	
	Either	Or
	Double click **Deployment Configuration**.	a. Right-click **Deployment Configuration**. b. Select Properties on the Popup menu.
D	In the Workbench right pane:	
	Either	Or
	Double click **Deployment Configuration**.	a. Right-click **Deployment Configuration**. b. Select Properties on the Popup menu.
E	Press Alt+Enter on the keyboard.	

6. Right-click **Deployment Configuration**.
7. Select Properties on the Popup menu.
8. Right-click **Deployment Configuration**.
9. Select Properties on the Popup menu.
10. Select the option *(page 371)* that addresses your requirements.
11. Click Next.

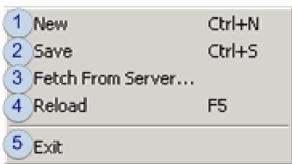
2.1.1. CIMPLICITY Deployment Configuration Editor: Menus

The CIMPLICITY Configuration Editor menu bar includes the following features.

Note: The features display on both the server and viewer. However, some features apply to one or the other.

rect 4, 1, 27, 36 *(page 368)*
rect 34, 1, 57, 36 *(page 368)*
rect 66, 1, 89, 36 *(page 369)*
rect 100, 1, 140, 36 *(page 370)*
rect 144, 1, 184, 36 *(page 370)*

1. File menu



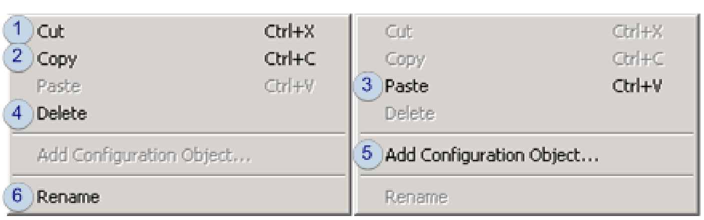
	Item	Description	Server	Viewer
1	New	Creates a new configuration file.	X	X
2	Save	Saves the current configuration file	X	X
3	Fetch from Server	Opens a Fetch From Server browser to find and open a gefdepl.dplcfg file from the deployment server	X	X
4	Reload	Refreshes the CIMPLICITY Configuration Editor display.	X	X
5	Exit	Closes the CIMPLICITY Configuration Editor.	X	X

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.

Another Popup menu opens when an value is selected in edit mode.

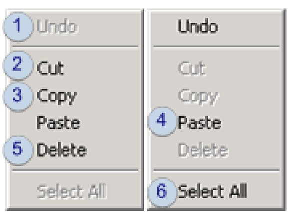
Edit Menu/Popup



	Item	Description	Server	Viewer

1	Cut	Cuts a selected entity, e.g. a folder; the entity can be pasted in a selection location.	X	X
2	Copy	Copies a selected entity, e.g. a folder; the entity can be pasted in a selected location	X	X
3	Paste	Pastes a cut or copied entity in a selected location. A number is appended to the name. Example z:\ViewAll under the 'COMMON' viewer is copied and pasted under a z:\View101 folder. The pasted name is z:\ViewAll(1).	X	X
4	Delete	Deletes the selected entity.	X	X
5	Add Configuration Object	Adds a new entity, e.g. a viewer or folder to the tree.	X	X
6	Rename	Makes a selected entity available for editing.	X	X

Edit Value Popup menu



	Item	Description	Server	Viewer
1	Undo	Undoes the last action.	X	X
2	Cut	Cuts the value being edited; the selection can be pasted in a selected location.	X	X
3	Copy	Copies the value being edited; the selection can be pasted in a selected location.	X	X
4	Paste	Pastes a cut or copied value in a selected location. A number is appended to the name. Example c:\ViewAll under the 'COMMON' folder is copied and pasted under a z:\View101 folder. The pasted name is z:\ViewAll(1).	X	X
5	Delete	Deletes the selected value.	X	X
6	Select All	Selects the entire value for editing.	X	x

1. View menu

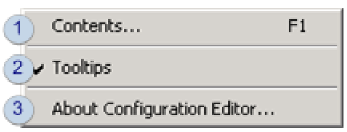
	Item	Description		Server	Viewer
1	Toolbar	Select to:		X	X
		Check	Display the toolbar.		
		Clear	Hide the toolbar.		

2	Status Bar	Select to:		X	X
		Check	Display the status bar.		
		Clear	Hide the status bar.		
3	Split	Enables resizing the panes using the arrow keys on the keyboard.		X	X

1. CimSync menu

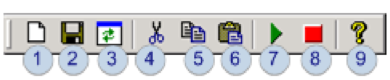
	Item	Description	Server	Viewer
1	Starts	Starts synchronization with the deployment server.		X
2	Stop	Stops synchronization with the deployment server.		X

1. Help menu



	Item	Description		Server	Viewer
1	Contents	Opens the Deployment topic in the CIMPLICITY documentation.		X	X
2	Tooltips	Select to:		X	X
		Check	Display tool tips		
		Clear	Hide tool tips.		
3	About Configuration Editor.	Opens details about the CIMPLICITY version and build.		X	X

2.1.2. CIMPLICITY Deployment Configuration Editor: Toolbar



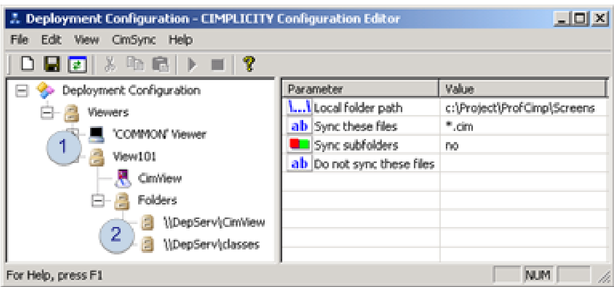
	Description	When Enabled
1	Create a new configuration file.	Always
2	Save the configuration file.	Always
3	Reload the configuration file from the disk.	After the configuration file is saved for the first time.
4	Cut the selected entity	When an entity, e.g. a folder or viewer, is selected.
5	Copy a selected entity or value.	When an entity, e.g. a folder or viewer, is selected.

6	Paste a cut/copied entity or value.	When an entity, e.g. a folder or viewer, is selected.
7	Run synchronization with the server.	On a viewer when synchronization is stopped.
8	Stop synchronization with the server.	On a viewer when synchronization is running.
9	Open deployment documentation.	Always

2.1.3. CIMPLICITY Deployment Configuration Editor: Left Pane

The left pane basically lists the sources of the deployed files.

Basic listed information is as follows.

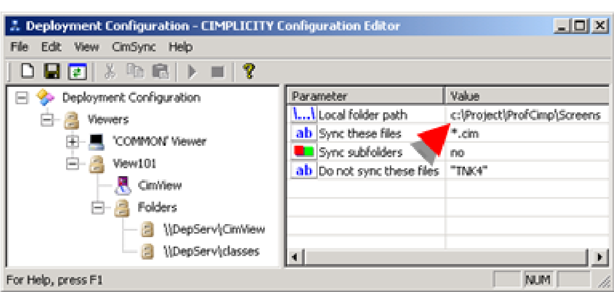


A	Viewer names.
B	Shared folders on the server with deployable files configured for this viewer.

2.1.4. CIMPLICITY Deployment Configuration Editor: Right Pane

The CIMPLICITY Configuration Editor right pane basically defines:

- Where the files are deployed to on the viewer
- What files are deployed
- When the files should be deployed.



Step 2.2. Create a new Configuration File

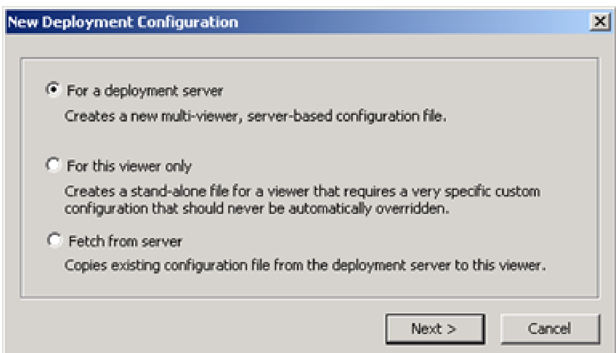
Use any of the following methods to create a new configuration file.

- Open (page 366) the CIMPLICITY Configuration Editor when a configuration file is not in the installation \Data directory.
- Click the New button on the CIMPLICITY Configuration Editor toolbar.
- Select File>New on the CIMPLICITY Configuration Editor menu bar.

A New Deployment Configuration window opens.

1. Check an option.
2. Click Next.

   Options are as follows.



rect 21, 38, 310, 77 (page 372)
rect 23, 85, 347, 136 (page 373)
rect 23, 134, 347, 177 (page 374)
rect 304, 188, 380, 214 (page 374)

- For a deployment server
- For this viewer only
- Fetch from server
- Cancel

Important: You can create more than one configuration file when you are working on the CIMPLICITY server and setting up your configuration and copy the version to its appropriate viewer or the deployment server.

However, the file will always be saved to the CIMPLICITY installation \Data directory. If you plan to create more than one version make sure that you copy the version you created from the Data directory into another directory before creating a second version. An existing version in the data directory will be overwritten.

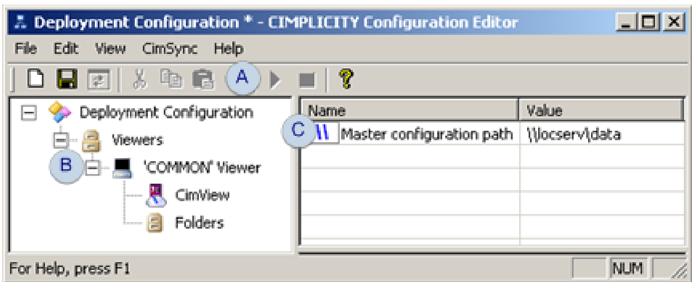
Check one of the following.

For a deployment server

The file created will be included in the automatic synchronization from the server to the viewer.

Always check For a deployment server for the version that will be on the deployment server.

The deployment server configuration file template opens with the following features.



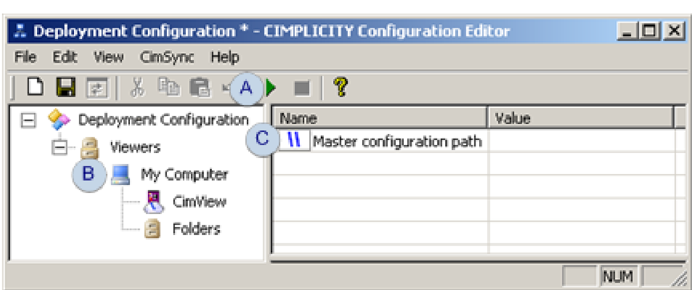
A	Run and Start buttons are disabled. **Note:** The buttons will be enabled when the file is deployed to a viewer.
B	A 'COMMON' Viewer is available in the left pane.
C	The default Master configuration path in the right pane is: `\\`<local server>`\data` Where <local server> is the CIMPLICITY server. `data` is the C:\Program Files\Proficy\Proficy CIMPLICITY\data folder. **Note:** The actual value can be changed *(page 375)*.

For this viewer only

A stand-alone file can be important for a viewer that requires a very specific custom configuration that should never be automatically overridden.

When For this viewer only is checked, the viewer to which this version is copied will not be included in an automatic configuration file synchronization with the server.

The stand-alone configuration file opens with the following features.



A	The Run button is enabled, since the template is meant for a viewer.

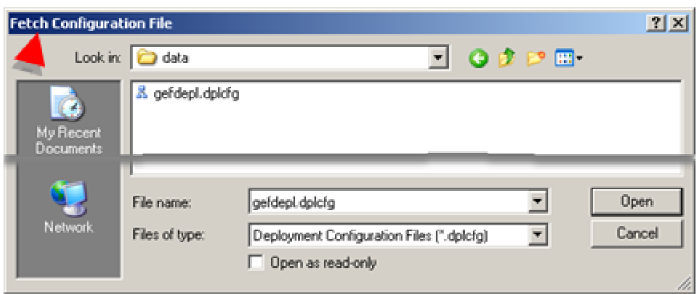
B	My Computer is available as the viewer in the left pane.
C	The **Master configuration path** field is empty, since the template is meant for a viewer.

Fetch from Server

Fetch from Server opens a Fetch Configuration File browser that enables you to quickly find an existing gefdepl.dplcfg file that is on the same or a different server in the network.

Tip: If the server drive that contains the CIMPLICITY project is mapped to the local computer, the Fetch Configuration File browser opens to the data directory and displays the gefdepl.dplcfg file.

Deployment also supports older versions of the configuration file, gefdepl.cimcfg files; that file will display if *.cimcfg is selected in the **Files of type** field.



Cancel

The Cancel button closes the New Deployment Configuration window closes.

Step 2.3. Identify the Master Configuration Path

The Master configuration path is the UNC path to the server and folder from which the configuration file is distributed and, if specified, updated whenever it is modified.

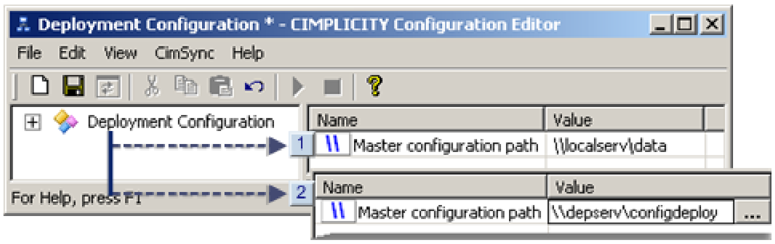
When the configuration is saved in the CIMPLICITY Configuration Editor, the configuration file, gefdepl.dplcfg, is saved in the CIMPLICITY server installation Data directory.

However, the master configuration path

CIMPLICITY does not have to be installed on the deployment server. However, the path must be a path that has read access from the Viewer.

Select Deployment Configuration in the CIMPLICITY Configuration Editor left pane.

Do either of the following to select the Master configuration UNC path.



rect 174, 69, 450, 97 *(page 375)*
rect 175, 106, 482, 144 *(page 375)*

1 *(page 375)*	Default UNC path
2 *(page 375)*	UNC path that is not the default
	Important guidelines

1	Default UNC path

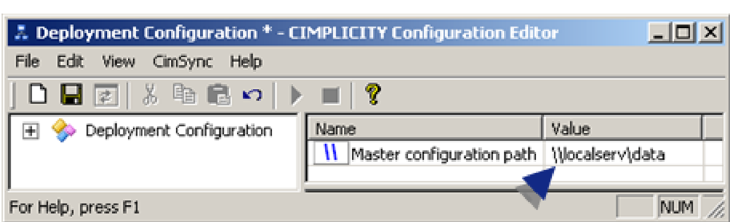
The default UNC path is \\<Local Server>\Data

Where

<Local Server> is the name of the CIMPLICITY server.

`\Data` is the Data folder in the CIMPLICITY installation path, C:\Program Files\Proficy\Proficy CIMPLICITY\Data.

Leave the local server\data UNC path in the **Master configuration path** field if the local server and CIMPLICITY\Data folder should be the path to the deployment server.



2	UNC path that is not the default.

Do either of the following to enter a Master configuration UNC path that is different from the default.

Enter the Master configuration path.
Use the Browse button

Enter the Master configuration path

A UNC path can be typed in the **Master configuration path** field, if it is different from the default path.

The format is \\<Server name>\<Folder Name>

Where

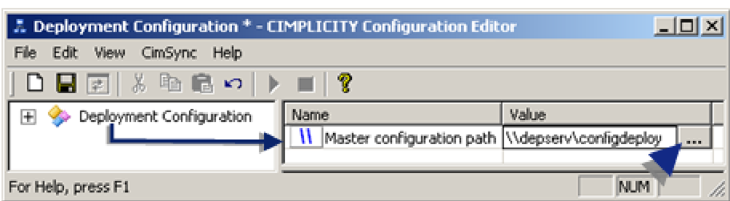
<Server name> is the server on which the deployment configuration file is located.

<Folder name> is the name of the folder the file is in.

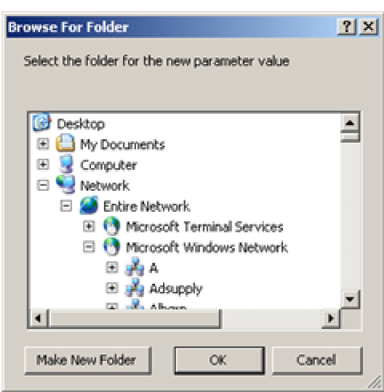
Result: The selected UNC path will be entered in the Master configuration path field.

Use the Browse button

1. Click the Browse button to the right of the **Master configuration path** field.



A Browse for computer browser opens.



1. Find and select the server and configuration file deployment folder on your network.
2. Click OK.

Result: The selected UNC path will be entered in the Master configuration path field.

guide: Important Guidelines

- When you modify and save gefdepl.dplcfg, it is always updated in the CIMPLICITY installation Data directory. If you do not use the default location, you will have to copy and paste the updated version into the Master configuration path in order for it to be deployed to the viewers.
- If the selected server or path is not accessible when you save the configuration file, a message box will issue a warning . You can still save the file and continue configuration. Simply make sure that when the configuration file needs to be available for deployment that it is placed in the specified path.
- Make sure the selected deployment folder is available to the Viewers involved in the deployment.

Example using a non-default Master configuration UNC path

1. Gefdepl.dplcfg is always automatically created and saved in the CIMPLICITY installation data directory on the CIMPLICITY server.

   The directory is:

   c:\Program Files\Proficy\Proficy CIMPLICITY\Data

2. A second server, depserv, will be used as the deployment server.

3. Gefdepl.dplcfg is copied/pasted to c:\ConfigDeploy on the deployment server

   Note: If specified, gefdepl.dplcfg will be automatically updated on every viewer, whenever it is modified

   ConfigDeply will be the official master configuration share folder name on the deployment server..

   The following is entered as the Master configuration path.

   \\depServ\ConfigDeploy

   Where

   \\depserv is the deployment server name.

   \ConfigDeploy is the share folder name.

4. Whenever Gefdepl.dplcfg is updated, it is copied/pasted from the ...\Data folder into the folder specified in the **Master configuration path** field.

Step 2.4. Identify 'COMMON' Viewer Parameters and Folders

Step 2.4. Identify 'COMMON' Viewer Parameters and Folders

Files that need to be deployed to several viewers can be placed in folders on the deployment server under a 'COMMON' Viewer.

This will eliminate duplication of effort and insure that the single correct file version is deployed at one time to every required viewer.

Identification steps in the CIMPLICITY Configuration Editor are as follows.

- Select 'COMMON' Viewer
- Configure 'COMMON' Viewer

Select 'COMMON' Viewer

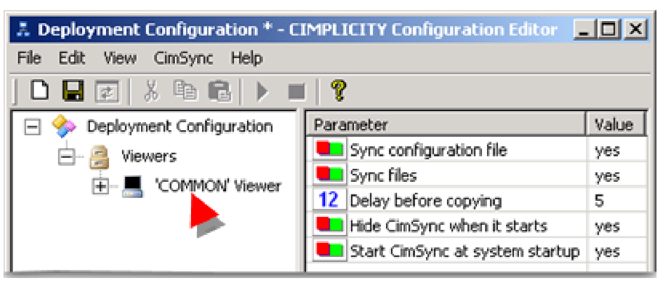
A 'COMMON' Viewer:

- Is included in the CIMPLICITY Configuration Editor.
- Lists default synchronization values.

The default values:

- Are used for all viewers that are not specifically listed in the configuration, but are included in the synchronization.
- Can be changed to conform to your system's requirements.
- Can be changed on an individual viewer to conform to the viewer's requirements.

Select 'COMMON' Viewer in the CIMPLICITY Configuration Editor left pane.



Configure 'COMMON' Viewer

Steps to configure the 'COMMON' Viewer are as follows.

Step 2.4.1 (page 379)	Define Parameters for the 'COMMON' Viewer.
Step 2.4.2 (page 383)	Enter CimView Reload Criteria for the 'COMMON' Viewer.
Step 2.4.3 (page 393)	Add a common folder for the Server/Viewer synchronization
Step 2.4.4 (page 394)	Define parameters for a 'COMMON' Viewer folder.

Step 2.4.1. Define Parameters for the 'COMMON' Viewer

A 'COMMON' Viewer:

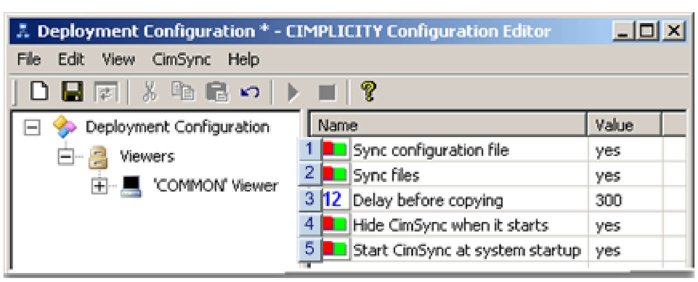
- Is included in the CIMPLICITY Configuration Editor.
- Lists default synchronization values.

The default values:

- Are used for all viewers that are not specifically listed in the configuration, but are included in the synchronization.
- Can be changed to conform to your system's requirements.
- Can be changed on an individual viewer to conform to the viewer's requirements.

Select 'COMMON' Viewer in the CIMPLICITY Configuration Editor left pane.

Available parameters are as follows.



rect 177, 75, 426, 94 (page 380)
rect 177, 92, 425, 111 (page 380)
rect 177, 109, 426, 128 (page 381)
rect 177, 126, 427, 142 (page 383)
rect 177, 140, 429, 161 (page 383)

1 (page 380)	Sync configuration file
2 (page 380)	Sync files
3 (page 381)	Delay before copying
4 (page 383)	Hide CimSync when it starts
5 (page 383)	Start CimSync at system startup

1	Sync configuration file

Select one of the following.

yes	Local copy will be synchronized with the deployment server file.
no	Local copy will be stand-alone, not synchronized with the deployment server.
Default	yes

Note:

The **Sync configuration file** value, yes or no, in the configuration file that is in each Viewer's local folder enables or disables synchronizing the server's configuration file with that viewer.

To insure that synchronization continues, make sure that **Sync configuration file** is set to yes in all configuration files that are deployed to the viewer. If a configuration file is deployed to the viewer's local folder with **Sync configuration file** set to no, the Viewer will no longer look for configuration updates.

The configuration file will have to be manually fetched on the Viewer, the value reset to yes and synchronization restarted *(page 410)* , to resume configuration deployment from the server to that viewer. At that time the viewer will only look to the server for a new configuration file; it will not look to see if there is a non-deployed configuration file that might have been placed in its own temporary folder.

2	Sync files

Select one of the following.

yes	Files in the Server's deployment folders will be synchronized with the files in the target folders on the Viewer.
no	Files will not be synchronized, even if they are in the deployment folder.

Default	yes

3	Delay before copying

- Is applied to deployment files only.

Important:

- The configuration file, gefdepl.dplcfg, is immediately deployed to the temporary folder on the viewer and then the local folder. Available system speed is the only factor that will impact its deployment time.
- CimView files have additional parameters *(page 383)* in order to insure that updated files do not disrupt running screen activity. The total deployment time is based on the value and conditions influencing those parameters and the **CopyDelay** parameter.
- Applies to both the server and the viewer as follows.

The value entered is applied in total to each, the server and viewer.

- Base number of seconds CIMPLICITY deployment will wait after a file has been added or updated in the server's deployment folder before it is synchronized with the viewer.
- Base number of seconds the files waits in the viewer's temporary folder before it is copied to the local folder.

guide: Delay before copying Guidelines

The **Delay before copying** value is referred to as the base number of seconds because the actual time may exceed the minimum deployment time, which is 2x the entered value.

The actual deployment time, when only **Delay before copying** is considered (not the CimView parameters *(page 383)*), depends on when new or updated files are placed in the server's deployment folder. The following conditions affect the total deployment time.

Important: Delay before copying affects both the configuration file and other files. If sync configuration file is set to no, and the configuration file is updated on the server, it will be deployed to the Temp folder on the Viewer after the Delay before copying interval. However, it will remain in the Viewer's Temp folder; it will not overwrite the configuration file that is already in the Viewer's local folder.

The only way to update the current local configuration file version with the version in the Temp folder is to manually fetch it. If sync config file is reset to yes, the version that is in the Temp folder already on the Viewer will not overwrite the local version. Only an updated version on the Server can automatically be deployed and synchronized.

Condition 1

1. New files are placed in the server's deployment folder and travel through the Copy/Delay cycle on both the server and viewer

2. No new files are placed in the deployment folder during that period.

   Result: The files are copied from the viewer's temporary to local folders in the specified CopyDelay (X2) time.

   Example

   The **Delay before copying** parameter entered value is 300 seconds.

	Deployment Activity	Seconds
1	The new files stay in the Server folder and then are deployed to the temporary folder.	300
2	The new files stay in the temporary folder	300
	No new files are added to the server's deployment folder during this cycle.	
3	The new files are copied to the local folder(s).	
	Total deployment time	600

   Condition 2

3. New files are placed in the server's deployment folder.

4. Additional new files are placed in the deployment folder during the deployment cycle for the first group of files.

The first deployment cycle timing stops; the clock is reset to 0; the entire cycle starts again for both groups of files.

Example

The **Delay before copying** parameter entered value is 300 seconds.

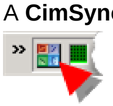
		Seconds	
	Deployment Activity	Group 1	Group 2
1	Group 1 files stay in the Server folder and then are deployed to the temporary folder.	300	
2	Group 1 files are in the viewer's temporary folder.	180	
3	Group 2 files are placed in the server's deployment folder while Group 1 files are still in the viewer's temporary folder.		
	Group 2 stays in the deployment folder		300
	Group 1 remains in the viewer's temporary folder	300	

4	Group 2 files are deployed to the viewer's temporary folder.		
	Both Group 1 and Group 2 stay in the viewer's temporary folder for 300 seconds.	300	300
	No new files are added to the server's deployment folder.		
5	Group 1 and Group 2 are copied to the viewer's local folders.		
	Total deployment time	1080	600

Delay before copying Default: 300 seconds

4	Hide CimSync when it starts

Select one of the following.

yes	CimSync runs in the background on 'COMMON' viewers; the **CimSync** icon does not display.
no	A **CimSync** icon is placed on the Windows Task bar when **CimSync** is running.  The icon opens a CimSync report window, which lists the synchronization activity.
Default	yes

5	Start CimSync at system startup

Select one of the following.

yes	CimSync will start up when a 'COMMON' Viewer is re-booted,
no	CimSync will have to be manually started from the CIMPLICITY Configuration Editor on the Viewer or by double-clicking CimSync.exe.
Default	yes

Step 2.4.2. Enter CimView Reload Criteria for the 'COMMON' Viewer

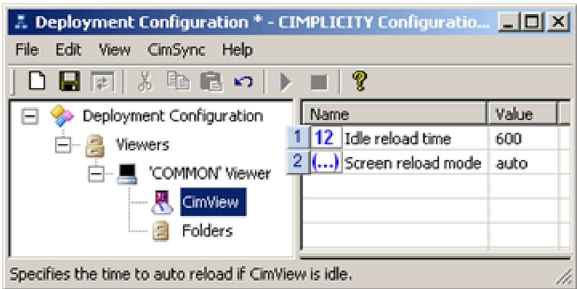
CIMPLICITY deployment can be instructed when an updated CimView file should be reloaded on the 'COMMON' Viewer, in order to insure the integrity of a screen that is currently running in CimView.

Important: The reload_mode parameter does not affect the server configuration file. When the server configuration file changes, it is automatically copied to the local directory irregardless of the reload_mode setting.

1. Expand Deployment Configuration>Viewers>'COMMON' Viewer in the CIMPLICITY Configuration Editor left pane.

2. Select CimView.

Available parameters are as follows.



rect 173, 70, 355, 90 [(page 384)](page 384)
rect 173, 88, 355, 108 [(page 385)](page 385)

1 (page 384)	Idle reload time
2 (page 385)	Screen reload mode
3 (page 385)	Sample conditions for screen reload.

1	Idle reload time

Idle reload time:

- Is used when the notify option is selected for the **screen reload mode**.
- Provides a gauge that CimSync uses to determine when it should automatically copy CimView files from the viewer's temporary folder to its local folder(s) even though the selection is to initially notify the user. This feature insures that the viewer remains up-to-date when it is idle.
- Is the number of seconds that CimView must be completely idle after CimSync sends a request to copy files and does not receive a response.

The actual number of seconds that the CimView files remain in the temporary folder before they override **notify** and are automatically copied to the local folders depends on several factors, including:

- Delay before copying *(page 381)* time.
- Idle reload time.
- When new files are added to the server's deployment folder.
- Actual time in the deployment cycles that the screen goes idle.

Default: 600 seconds

Note: When the reload mode is set to Notify and when the configuration file on the server changes the following happens.

3. The configuration file is automatically copied to the Viewer
4. A prompt displays.
   - Confirming the prompt accepts all environment changes that CimSync performs.
   - When the new configuration file is accepted a re-sync with the server will be performed to pull down any new files specified in the new configuration file.
5. The change in the local configuration file restarts the monitoring after reading the changed configuration file.

2	Screen reload mode

Screen reload mode options are:

Auto

CimSync automatically copies the files to the local directory during synchronization after the number of seconds determined by the Delay before copying *(page 381)* parameter value.

Note: It is recommended that auto reload be used if CimView is running in Terminal Services sessions.

Notify

CimSync:

6. Notifies the Viewer's user that new files have been in the temporary folder after the **Delay before copying** seconds have elapsed.
7. Asks the user if it can copy the files into the local folder(s).

Result: Exactly when CimSync copies the files from the temporary folder into the local folder depends on the following conditions.

- Delay before copying *(page 381)* time.
- Idle reload time.
- When new files are added to the server's deployment folder.
- Actual time in the deployment cycles that the screen goes idle.

Default: auto

a. Sample Conditions for Screen Reload

The following sample conditions provide examples of how the total deployment time is computed when Notify is selected as the **Screen reload mode**:

Condition 1 (page 386)	User grants permission.
Condition 2 (page 387)	User denies permission.
Condition 3 (page 388)	User denies permission/additional files are added to the Server Deployment folder.
Condition 4 (page 390)	User ignores the request but the screen is active.
Condition 5 (page 391)	User denies the request and then the screen becomes idle.
Condition 6 (page 392)	CimView is not running.
Condition 7 (page 392)	Terminal Services is being used.

Condition 1: User Grants Permission

8. New files are placed in the server's deployment folder and travel through the Copy/Delay cycle on both the server and viewer

9. No new files are placed in the deployment folder during that period.

10. An Environment Changed message pops up in the lower, right-hand corner of the CimView screen.

    Note: The Environment Changed message is how CimSync sends a request to copy the files from the temporary to the local folders.

11. The user immediately clicks Reload.

    Clicking Reload grants CimSync permission to copy the files into the local folders.

Results:

- The Environment Changed message closes.
- CimSync copies the files from the viewer's temporary to its local folders.

Example

The Delay before copying *(page 381)* parameter value is 300 seconds.

The **idle reload time** parameter value is 600 seconds.

Deployment Activity		Seconds
1	The new files stay in the Server folder and then are deployed to the temporary folder.	300
2	The new files stay in the temporary folder	300
3	No new files are added to the server's deployment folder during this cycle.	
	The user is notified and grants permission.	0
4	The new files are copied to the local folder(s).	
	Total deployment time	600

Condition 2: User Denies Permission

12. New files are placed in the server's deployment folder and travel through the Copy/Delay cycle on both the server and viewer

13. No new files are placed in the server's deployment folder during that period.

14. An Environment Changed message pops up in the lower, right-hand corner of the CimView screen.

    Note: The Environment Changed message is how CimSync sends a request to copy the files from the temporary to the local folders.

15. The user clicks Snooze.

    Clicking Snooze denies CimSync permission to copy the files into the local folders.

    Results:

    - The Environment Changed message closes.
    - CimSync resends another request to copy after the Copy/Delay period (from the temporary folder to the local folder) has elapsed for a second time.

    CimSync continues to send requests at the **Delay before copying** interval if:

16. The user:

a. Is interacting with the screen.
b. Continues to deny the request.

17. No other files are placed in the server's deployment folder.

    Example

    The Delay before copying *(page 381)* parameter value is 300 seconds.

    The **idle reload time** parameter value is 600 seconds.

Deployment Activity		Seconds
1	The new files stay in the Server folder and then are deployed to the temporary folder.	300
2	The new files stay in the temporary folder	300
3	No new files are added to the server's deployment folder during this cycle.	
	The user is notified and denies permission.	0
4	The files continue to stay in the viewer's temporary folder.	
	CimSync sends another request after the **Delay before copying** time in the temporary folder.	300
5	The user denies permission. New files still have not been added to the server's deployment folder.	
	The files continue to stay in the viewer's temporary folder.	300
7	CimSync sends another request after the **Delay before copying** time in the temporary folder.	
	The user grants permission	0
6	The new files are copied to the local folder(s).	
	Total deployment time	1200

    Condition 3: User Denies Permission/Additional files are added to the Server Deployment Folder

18. New files are placed in the server's deployment folder and travel through the Copy/Delay cycle on both the server and viewer

19. No new files are placed in the deployment folder during that period.

20. An Environment Changed message pops up in the lower, right-hand corner of the CimView screen.

    Note: The Environment Changed message is how CimSync sends a request to copy the files from the temporary to the local folders.

21. The user clicks Snooze.

Clicking Snooze denies CimSync permission to copy the files into the local folders.

22. New files have been placed in the server's COMMON deployment folders before CimSync sends another copy request to the user.

    Results"

    - The Environment Changed message closes.
    - The clock is reset so the Delay before copy second count starts again.
    - CimSync notifies the user at the end of the new server/viewer cycle.

    This amount of time is the total of the number of seconds that have elapsed since the last request plus the total seconds in the Copy/Delay cycle.

    Example

    The Delay before copying *(page 381)* parameter value is 300 seconds.

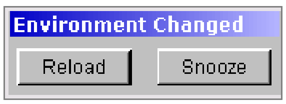
The **idle reload time** parameter value is 600 seconds.		Seconds	
Deployment Activity		Group 1	Group 2
1	Group 1 files stay in the Server folder and then are deployed to the temporary folder.	300	
2	Group 1 files are in the viewer's temporary folder.	300	
3	The user is notified and denies permission.		
	Group 1 files continue to stay in the viewer's temporary folder.	180	
4	Group 2 files are placed in the server's deployment folder, while Group 1 files are still in the viewer's temporary folder.		
	Group 2 stays in the deployment folder		300
	Group 1 remains in the viewer's temporary folder	300	
4	Group 2 files are deployed to the viewer's temporary folder.		
	Both Group 1 and Group 2 stay in the viewer's temporary folder.	300	300
5	No new files are added to the server's deployment folder.		
	CimSync sends another request to copy the files from the temporary to the local folder(s).		
	The user grants permission	0	0
6	Group 1 and Group 2 are copied to the viewer's local folders.		
	Total deployment time	1380	600

Condition 4: User Ignores Request but the Screen is Active

23. New files are placed in the server's deployment folder and travel through the Copy/Delay cycle on both the server and viewer
24. No new files are placed in the server's deployment folder during that period.
25. An Environment Changed message pops up in the lower, right-hand corner of the CimView screen.

    Note: The Environment Changed message is how CimSync sends a request to copy the files from the temporary to the local folders.

26. The user ignores the prompt, but is interacting with the screen.



Results:

- The Environment Changed message remains open on the screen.
- Another Environment Changed message pops up in the lower right corner of the CimEdit screen after the next Delay before copying time period has elapsed.

Example

The Delay before copying *(page 381)* parameter value is 300 seconds.

The **idle reload time** parameter value is 600 seconds.

Deployment Activity		Seconds
1	The new files stay in the Server folder and then are deployed to the temporary folder.	300
2	The new files stay in the temporary folder	300
3	No new files are added to the server's deployment folder during this cycle.	
	The user is notified and ignores the request. **Note:** The message remains on the screen.	
	The files continue to stay in the viewer's temporary folder.	
	CimSync sends another request after the **Delay before copying** time in the temporary folder.	300
4	The user ignores the request. **Note:** The message remains on the screen. New files still have not been added to the server's deployment folder.	
	The files continue to stay in the viewer's temporary folder.	
	The user grants permission before the **Delay before copying** time has elapsed.	180
5	The new files are copied to the local folder(s).	

	Total deployment time	1080

Condition 5: User Denies the Request and then the Screen Becomes Idle

27. New files are placed in the server's deployment folder and travel through the Copy/Delay cycle on both the server and viewer

28. No new files are placed in the server's deployment folder during that period.

29. The user clicks Snooze.

    Clicking Snooze denies CimSync permission to copy the files into the local folders.

30. The screen goes idle and remains idle after the user denies the prompt.

CimSync:

1. Resends a request one Delay before copying time period after the first request and discovers the screen is idle.
2. Waits the entered idle time delay.
3. Automatically copies the files from the temporary to the local folders.

Example

The Delay before copying *(page 381)* parameter value is 300 seconds.

The **idle reload time** parameter value is 600 seconds.

Deployment Activity		Seconds
1	The new files stay in the Server folder and then are deployed to the temporary folder.	300
2	The new files stay in the temporary folder	300
3	No new files are added to the server's deployment folder during this cycle.	
	The user is notified and denies permission.	
	The files continue to stay in the viewer's temporary folder.	
	CimSync sends another request after the **Delay before copying** time in the temporary folder.	300
4	The user denies permission. New files still have not been added to the server's deployment folder. The screen goes idle before the **Delay before copying** time has elapsed.	
	The files continue to stay in the viewer's temporary folder.	

	CimSync sends another request after the **Delay before copying** time (starting from the last time the user denied permission).	300
5	CimSync discovers that the screen is idle.	
	New files still have not been added to the server's deployment folder. The screen continues to be idle.	
	CimSync waits the specified idle reload time.	600
7	The new files are copied to the local folder(s).	
	Total deployment time	1800

Condition 6: CimView is not Running

CimView is not running on the viewers.

CimSync will copy the files to the local directory automatically.

Example

The Delay before copying *(page 381)* parameter value is 300 seconds.

The **idle reload time** parameter value is 600 seconds.

Deployment Activity		Seconds
1	The new files stay in the Server folder and then are deployed to the temporary folder.	300
2	The new files stay in the temporary folder	300
3	No new files are added to the server's deployment folder during this cycle.	
	CimView is not running.	
4	The new files are copied to the local folder(s).	
	Total deployment time	600

Condition 7: Terminal Services is being Used

Terminal Services is being used for configuration deployment.

There is no clearly defined or expected behavior when CimView is running on multiple sessions and is prompted that new files have been added to the COMMON folders.

CimSync will:

1. Check to see if CimView is running on the console (Terminal Services) server only.
2. Prompt the server before copying files to the local directory.
3. Copy or not copy the files based on the console users response to the CimView prompt.

Note: It is recommended that auto reload be used if CimView is running in Terminal Services sessions.

Step 2.4.3. Add a Common Folder for the Server/Viewer Synchronization

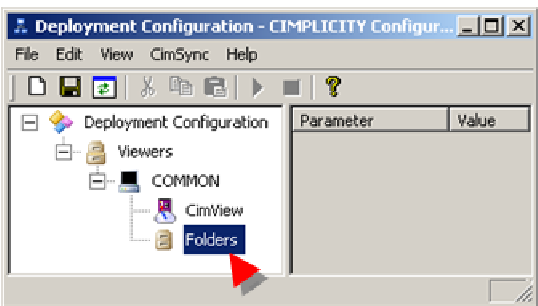
Note:

- The 'COMMON' Viewer can contain folders that can deploy files to every viewer that:
- Has read access to the shared folder on the deployment server.
- Shares the common configuration file that includes the selected folder.
- Do not have their own viewer name included in the configuration.
- The folder name in the left pane (tree view) identifies the viewers' mapped drive that contains the source files.

1. Expand Deployment Configuration>Viewers>'COMMON' Viewer in the CIMPLICITY Configuration Editor left pane.

2. Select Folders.

   Note: The **Folders** item has no parameters.



3. Do one of the following.
   - Right-click Folders; select Add on the Popup menu.
   - Select Edit>Add on the CIMPLICITY Configuration Editor menu bar.

   An Add configuration Object dialog box opens.

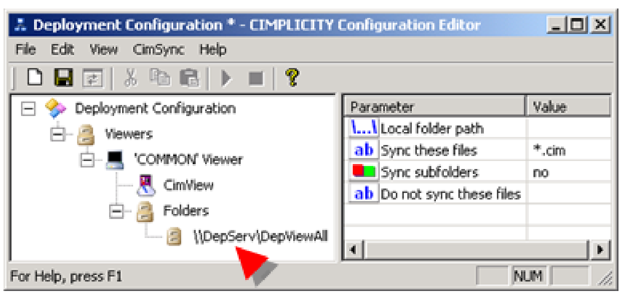
4. Specify the following.

Name	The UNC path to the deployment server folder that will contain source files. **Note:** The viewer must have read-access to the shared folder. Example `\\DepServ\DepViewAll` Where	
	`DepServ`	Is the deployment server
	`DepViewAll`	Is the shared drive on the deployment server.
Type	Folder	
Icon	Select a folder.	

Note: You can use a mapped drive. Every viewer using the 'COMMON' folder must have the same mapped drive.

5. Click OK.

CIMPLICITY deployment adds the folder to the 'COMMON' Viewer's list of folders in the CIMPLICITY Configuration Editor left pane.

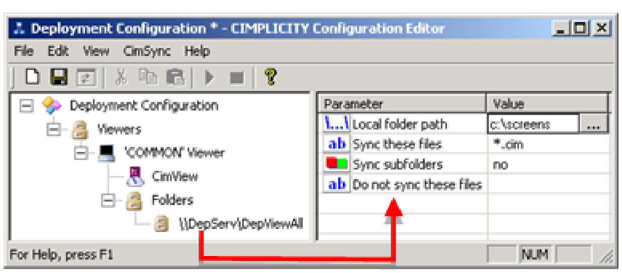


Step 2.4.4. Define Parameters for a 'COMMON' Viewer Folder

Select a source folder in the CIMPLICITY Configuration Editor left pane.

Parameters in the right pane will identify what files should go to what target folders on the viewers.

Available parameters are as follows.



Parameter	Description

Local folder path	The Local folder path is the actual path and folder on the Viewer that will receive the deployed files. Example The files that the Viewers use are in the c:\Screens folder The value entered is: c:\Screens **Note:** All viewers that use this folder must have a c:\screens folder. **Tip:** Click the Browse button to open the Browse for a folder browser.	
Sync these files	• Any number or type of file can be put in the source folder. However, only files that fill one or more of the entered specifications will be synchronized. • File specifications should be separated by semi-colons ;and can contain the following wildcard characters: *?	
	.	All files.
	No entry	No files.
	Default	*.cim
Sync subfolders	Choose one of the following.	
	yes	Synchronizes files in the selected folder's subfolders, if there are any.
	no	Does not synchronize sub-folder files.
	Default	no
Do not sync these files	• Files in the source folder (or subfolders, if included) that fill one or more entered specifications will not be synchronized. • File specifications should be separated by semi-colons ;and can contain the following wildcard characters: *?	

Add Configuration Object Dialog Box

The Add Configuration Object dialog box is used to add the following to the deployment configuration.

Add a common folder for the 'COMMON' Viewer.
Add a selected viewer.
Add a source folder for a selected viewer.

Step 2.5. Identify a Selected Viewer's Parameters and Folders

Step 2.5. Identify a Selected Viewer's Parameters and Folders

If a selected viewer has configuration requirements that are different from the "COMMON' viewers, it can be identified and set up separately.

Important: If a viewer is set up separately it will not be included in the 'COMMON' viewer deployment.

Step 2.5.1 (page 396)	Add a viewer to be included in synchronization.
Step 2.5.2 (page 397)	Define parameters for an added viewer.
Step 2.5.3 (page 400)	Enter CimView reload criteria for a selected viewer.
Step 2.5.4 (page 402)	Add a source folder for the server/viewer synchronization.
Step 2.5.5 (page 403)	Define parameters for the source folder.

Step 2.5.1. Add a Viewer to be Included in Synchronization

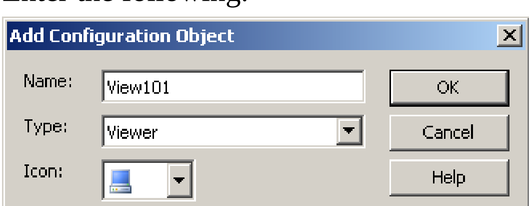
1. Expand Deployment Configuration in the CIMPLICITY Configuration Editor left pane.

2. Select Viewers.

3. Do one of the following.
   - Click Edit>Add on the CIMPLICITY Configuration Editor menu bar.
   - Right-click Viewers; select Add on the Popup menu.

   An Add configuration Object dialog box opens when you use either method.

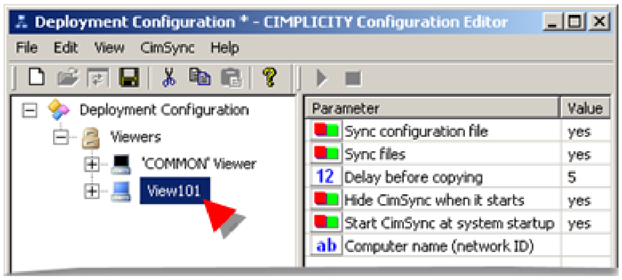
4. Enter the following.



Field	Description
Name	The viewer name. **Note:** The name must be the name that displays on the Computer Name tab in the Windows Systems Properties dialog box.
Type	Viewer
Icon	Select a computer.

5. Click OK.

The Viewer is added to the list in the Configuration Window left pane.



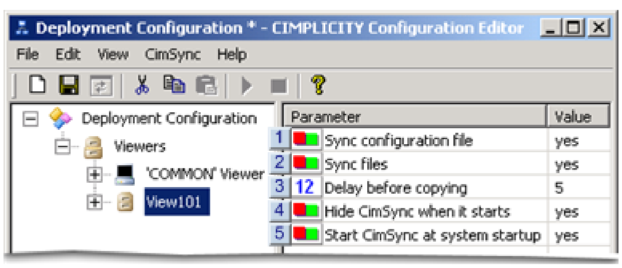
Step 2.5.2. Define Parameters for an Added Viewer

Note:

- An added Viewer provides the parameters included with the 'COMMON' Viewer plus a parameter for identification.
- Values entered for the selected viewer's parameters override values entered for the 'COMMON' Viewer. If a parameter value is blank, the 'COMMON' Viewer value is used.

Select the viewer to be configured.

Available parameters are as follows.



1 (page 398)	Sync configuration file.
2 (page 398)	Sync files.
3 (page 398)	Delay before copying.
4 (page 399)	Hide CimSync when it starts.

5 (page 399)	Start CimSync at system startup.

1	Sync configuration file

Select one of the following.

yes	Local copy will be synchronized with the deployment server file.
no	Local copy will be stand-alone, not synchronized with the deployment server.
Default	yes

Note:

The **Sync configuration file** value, yes or no, in the configuration file that is in each Viewer's local folder enables or disables synchronizing the server's configuration file with that viewer.

To insure that synchronization continues, make sure that **Sync configuration file** is set to yes in all configuration files that are deployed to the viewer. If a configuration file is deployed to the viewer's local folder with **Sync configuration file** set to no, the Viewer will no longer look for configuration updates.

The configuration file will have to be manually fetched on the Viewer, the value reset to yes and synchronization restarted *(page 410)* , to resume configuration deployment from the server to that viewer. At that time the viewer will only look to the server for a new configuration file; it will not look to see if there is a non-deployed configuration file that might have been placed in its own temporary folder.

2	Sync files

Select one of the following.

yes	Files in the Server's deployment folders will be synchronized with the files in the target folders on the Viewer.
no	Files will not be synchronized, even if they are in the deployment folder.
Default	yes

3	Delay before copying

- Is applied to deployment files only.

Number	(Seconds) The default number of seconds CIMPLICITY deployment will wait after a file has been added or updated in the deployment folder before it is synchronized with the viewer. Based on the steps that deployment takes the actual time delay will be:	
	gefdepl.dplcfg	3x the number entered
	Deployment files	2x the number entered.

Default		300
	gefdepl.dplcfg	900 seconds
	Deployment files	600 seconds
Note: The additional delay is necessary for the temporar *(page 359)*y file and configuration *(page 360)* file deployment.		

Important:

- The configuration file, gefdepl.dplcfg, is immediately deployed to the temporary folder on the Viewer. Once deployed, it will prompt you before it is copied to the Viewer's installation ...\data directory.
- CimView files have additional parameters *(page 383)* in order to insure that updated files do not disrupt running screen activity. The total deployment time is based on the value and conditions influencing those parameters and the **CopyDelay** parameter.
- Applies to both the server and the viewer as follows.

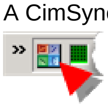
The value entered is applied in total to each, the server and viewer.

- Base number of seconds CIMPLICITY deployment will wait after a file has been added or updated in the server's deployment folder before it is synchronized with the viewer.
- Base number of seconds the files waits in the viewer's temporary folder before it is copied to the local folder.

Delay before copying Guidelines

4	Hide CimSync when it starts

Select one of the following.

yes	CimSync runs in the background; the CimSync icon does not display.
no	A CimSync icon is placed on the Windows Task bar when CimSync is running.  The icon opens a CimSync report window, which lists the synchronization activity.
Default	yes

5	Start CimSync at system startup

Select one of the following.

yes	CimSync will start up when a Viewer is re-booted,
no	CimSync will have to be manually started from the CIMPLICITY Configuration Editor on the Viewer or by double-clicking CimSync exe.

Default	yes

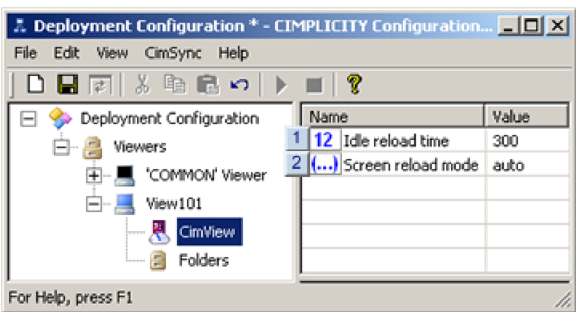
Step 2.5.3. Enter CimView Reload Criteria for a Selected Viewer

CIMPLICITY deployment can be instructed when an updated CimView file should be reloaded on a selected Viewer, in order to insure the integrity of a screen that is currently running in CimView.

Important: The reload_mode parameter does not affect the server configuration file, . When the server configuration file changes, it is automatically copied to the local directory irregardless of the reload_mode setting.

1. Expand Deployment Configuration>Viewers> <Selected Viewer> in the CIMPLICITY Configuration Editor left pane.
2. Select CimView.

   Available parameters are as follows.



rect 174, 74, 351, 92 (page 400)
rect 174, 90, 351, 109 (page 401)

1 (page 400)	Idle reload time
2 (page 401)	Screen reload mode

1	Idle reload time

Idle reload time:

- Is used when the notify option is selected for the **screen reload mode**.
- Provides a gauge that CimSync uses to determine when it should automatically copy CimView files from the viewer's temporary folder to its local folder(s) even though the selection is to initially notify the user. This feature insures that the viewer remains up-to-date when it is idle.
- Is the number of seconds that CimView must be completely idle after CimSync sends a request to copy files and does not receive a response.

The actual number of seconds that the CimView files remain in the temporary folder before they override **notify** and are automatically copied to the local folders depends on several factors, including:

- Delay before copying *(page 381)* time.
- Idle reload time.
- When new files are added to the server's deployment folder.
- Actual time in the deployment cycles that the screen goes idle.

Default: 600 seconds

Important: When the reload mode is set to Notify and when only the configuration file on the server changes the following happens.

3. The configuration file is automatically copied to Viewer

Note: When the reload mode is set to Notify and when the configuration file on the server changes the following happens.

4. The configuration file is automatically copied to the Viewer

5. A prompt displays.
   - Confirming the prompt accepts all environment changes that CimSync performs.
   - When the new configuration file is accepted a re-sync with the server will be performed to pull down any new files specified in the new configuration file.

6. The change in the local configuration file restarts the monitoring after reading the changed configuration file.

2	Screen reload mode

Screen reload mode options are:

Auto

CimSync automatically copies the files to the local directory during synchronization after the number of seconds determined by the Delay before copying *(page 381)* parameter value.

Note: It is recommended that auto reload be used if CimView is running in Terminal Services sessions.

Notify

CimSync:

7. Notifies the Viewer's user that new files have been in the temporary folder after the **Delay before copying** seconds have elapsed.

8. Asks the user if it can copy the files into the local folder(s).

Exactly when CimSync copies the files from the temporary folder into the local folder depends on the following conditions.

- Delay before copying *(page 381)* time.
- Idle reload time.
- When new files are added to the server's deployment folder.
- Actual time in the deployment cycles that the screen goes idle.

Default: auto

Step 2.5.4. Add a Source Folder for the Server/Viewer Synchronization

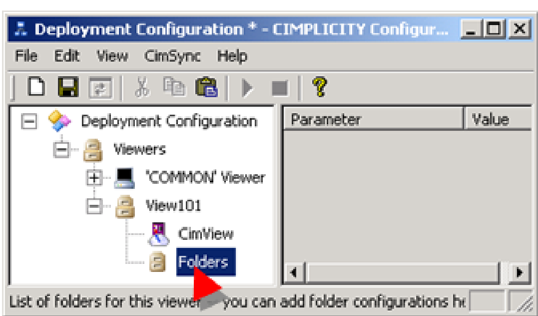
1. Expand Deployment Configuration>Viewers><Viewer Name> in the CIMPLICITY Configuration Editor left pane.

   Where

   `Viewer Name` is the viewer you are targeting for synchronization.

2. Select Folders.

   Note: The **Folders** item has no parameters.



3. Do one of the following.
   - Right-click Folders; select Add on the Popup menu.
   - Select Edit>Add on the CIMPLICITY Configuration Editor menu bar.

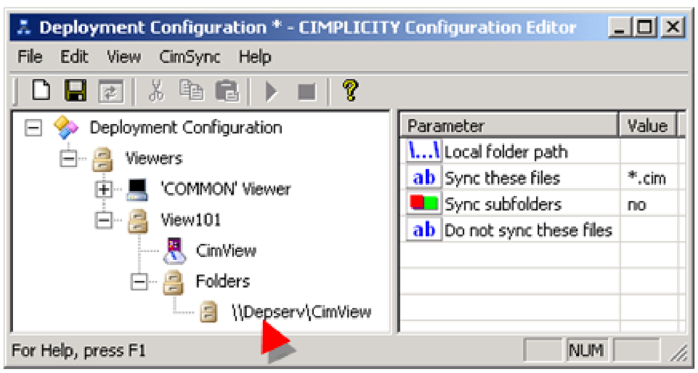
   An Add configuration Object dialog box opens.

4. Specify the following.

Name	The UNC path on the configuration server of the folder that will contain source files. Example `\\DepServ\CimView` Where `\\DepServ` is the configuration server. `\CimView` is the shared folder.
Type	Folder
Icon	Select a folder.

5. Click OK.

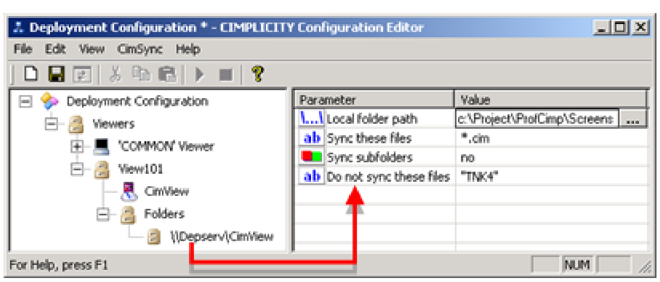
CIMPLICITY deployment adds the folder to the viewer's list of folders in the CIMPLICITY Configuration Editor left pane.



Step 2.5.5. Define Parameters for the Source Folder

Select a source folder in the CIMPLICITY Configuration Editor left pane.

Available parameters are as follows.



Parameter	Description

Local folder path	The Local folder path is the actual path and folder on the selected Viewer that will receive the deployed files. Example The files that the Viewer uses are in its c:\Project\PPSCimp\Screens folder The value entered is: c:\Project\PPSCimp\Screens **Tip:** Click the Browse button to open the Browse for a folder browser.	
Sync these files	• Any number or type of file can be put in the source folder. However, only files that fill one or more of the entered specifications will be synchronized. • File specifications should be separated by semi-colons ;and can contain the following wild characters: *?	
	.	All files.
	No entry	No files.
	Default	*.cim
Sync subfolders	Choose one of the following.	
	yes	Synchronizes files in the selected folder's subfolders, if there are any.
	no	Does not synchronize sub-folder files.
	Default	no
Do not sync these files	• Files in the source folder (or subfolders, if included) that fill one or more entered specifications will not be synchronized. • File specifications should be separated by semi-colons ;and can contain the following wild characters: *?	

Step 2.6. Complete Configuration on the Server

Step 2.6. Complete Configuration on the Server

A *(page 404)*	Save the configuration file.
B *(page 405)*	Set up the files and folders.

1. Save the configuration file.

1. Add and configure as many Viewers and folders as your system requires.
2. Do one of the following to save the configuration.
   - Click the Save button on the CIMPLICITY Configuration Editor toolbar.

- Select File>Save on the CIMPLICITY Configuration Editor menu bar.
- Press Ctrl+S on the keyboard.

3. Click File>Exit on the CIMPLICITY Configuration Editor menu bar.

The CIMPLICITY Configuration Editor closes. You are now ready to set up (or finish setting up) the identified paths and folders.

1. Set up the files and folders.

Do the following.

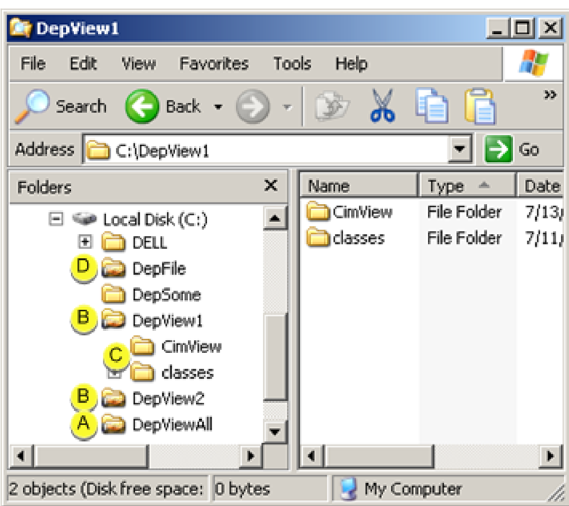
Step 2.6.1 (page 405)	Create the source folders on the deployment server.
Step 2.6.2 (page 406)	Make sure that gefdepl.dplcfg is in the Master Configuration folder.

Step 2.6.1. Create the Source Folders on the Deployment Server

Make sure that all of the folders shared in the CIMPLICITY Configuration Editor are created and shared on the deployment server.

This includes the folder for the gefdepl.dplcfg file.

For example, folders should include:



A	One or more folders shared as 'COMMON' Viewer folders
B	Folders shared for selected viewers.
C	Any required sub-folders.

D	Folder shared as the master configuration folder.

Step 2.6.2. Make sure that gefdepl.dplcfg is in the Master Configuration Folder

1. Find the gefdepl.dplcfg file that was just created in the CIMPLICITY installation Data directory.

   ...\Program Files\Proficy\Proficy CIMPLICITY\Data

2. Do one of the following.

Master configuration folder is the :	Do the following:
CIMPLICITY Installation Data folder.	Copy the file to paste in the Viewer. **Note:** The file can also be fetched from the Viewer.
Another deployment server.	Copy and paste the file into the deployment server folder that is identified as the master configuration *(page 374)* folder.

Step 3. Set up a Deployment Viewer

Step 3. Set up a Deployment Viewer

Configuration can be done on a viewer. However, it is recommended that it be done mainly on the Server.

- Files are not automatically deployed from a viewer to the server or other viewers, so any global changes will not be deployed.
- If the viewer is set up to automatically synchronize the configuration file, any changes to the viewer configuration will be overwritten.

Important: A Viewer must have enough disk for each local deployment folder and the temporary directory in order to insure successful deployment.

Calculations include space that is padding and that is at least the amount used on the server, as follows.

Directory	Viewer space and Padding Requirements	
Local	Free space	Free space that is at least the size of the corresponding source directories on the server
	Padding	1. Whichever of the following is larger: 2. • 1% of the deployed directories on the server. 3. • 1 MB.

Temporary	Free space	A minimum free space of the sum of all the configured server directories.
	Padding	1. Whichever of the following is larger: 2. • 1% of the deployed directories on the server. 3. • 1 MB.

Steps for the viewer include:

Step 3.1 (page 407)	Place the correct configuration file on the viewer.
Step 3.2 (page 409)	Open the CIMPLICITY Configuration Editor.
Step 3.3 (page 410)	Start synchronization.
Step 3.4 (page 411)	Review synchronization progress.
Step 3.5 (page 412)	Stop synchronization.

Step 3.1. Place the Correct Configuration File on the Viewer

Use any of the following methods to place the correct configuration file on the Viewer.

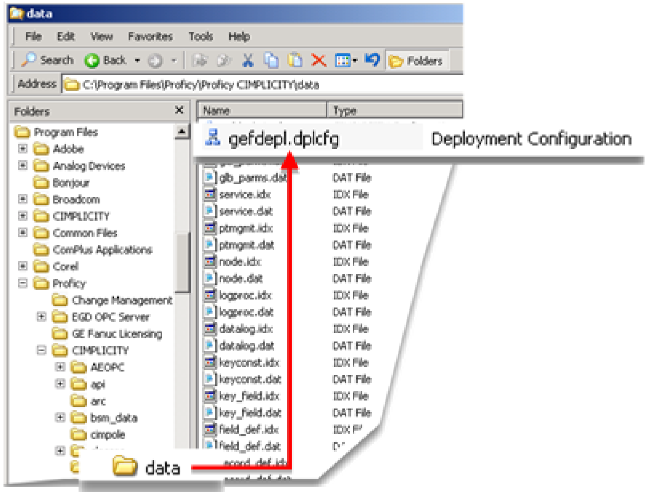
- Paste gefdepl.dplcfg into the Data directory.
- Fetch gefdepl.dplcfg from the deployment server.
- Create a new configuration file.

Paste Gefdepl.dplcfg into the Data directory

1. Open Windows Explorer.
2. Select the Viewer's installation data directory

   ..\Program Files\Proficy\Proficy CIMPLICITY\data

3. Paste the gefdepl.dplcfg file that was saved on the deployment server into the Viewer's installation data directory.



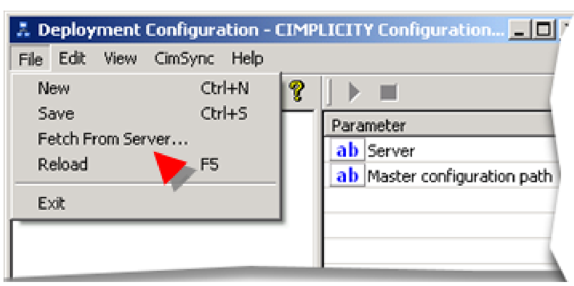
Note: Make sure that cimsync.exe is in the Viewer's installation exe directory.

..\Program Files\Proficy\Proficy CIMPLICITY\exe

Fetch Gefdepl.dplcfg from the deployment server

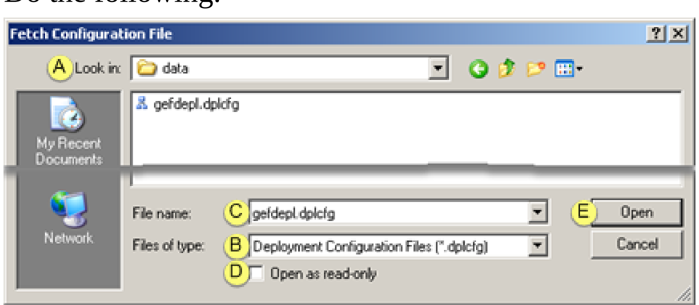
You can fetch gefdepl.dplcfg through the CIMPLICITY Configuration Editor.

4. Open the CIMPLICITY Configuration Editor.
5. Click File>Fetch from Server on the CIMPLICITY Configuration Editor menu bar.



The Fetch Configuration File browser opens.

6. Do the following.



A	Look in.	Find the directory on the deployment server that was assigned to deploy the gefdepl.dplcfg file.
B	Files of type	Select *.dplcfg **Note:** The file type *.cimcfg. is also available, to be backward compatible with older deployment configuration files
C	File name	Select gefdepl.dplcfg. **Note:** The file gefdepl.cimcfg is also available, to be backward compatible with an older deployment configuration file.
D	Open as read-only	(Optional) Check to deploy the file as a read-only file.
E	Open	Click Open.

Gefdepl.dplcfg is automatically deployed to the viewer's installation Data directory. If a configuration file was in the directory, it will be overwritten.

Note: If the correct configuration does not display in the CIMPLICITY Configuration Editor, click File>Reload on the CIMPLICITY Configuration Editor menu bar.

Create a New Configuration File

You can create a configuration file on a Viewer, the same way you do on a server.

If you do, it is recommended that you only create a stand-alone file for that viewer.

- Any configuration will not be deployed to any other computer.
- A stand-alone file will not be overwritten by the deployment server version.

Step 3.2. Open the CIMPLICITY Configuration Editor

Do one of the following.

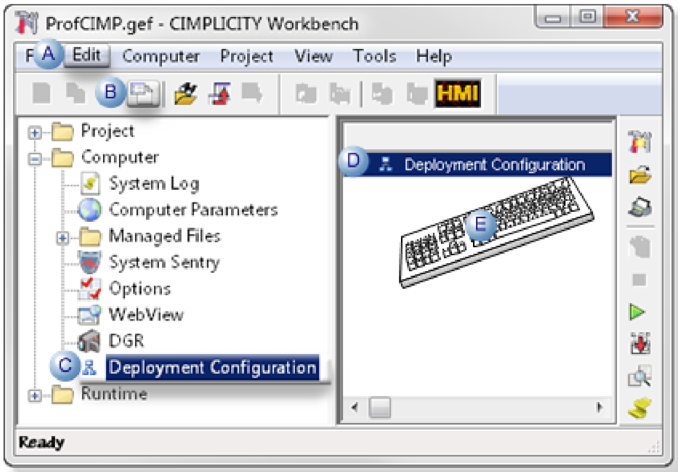
- Start menu
- Workbench

Start Menu

1. Click Start on the Windows Task bar.
2. Select All Programs> HMI/SCADA CIMPLICITY version>Deployment Configuration.

   Workbench

3. Select Computer>Deployment Configuration in the Workbench left pane.
4. Select Deployment Configuration in the right pane.
5. 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:	
	Either	Or
	Double click **Deployment Configuration**.	a. Right-click **Deployment Configuration**. b. Select Properties on the Popup menu.
D	In the Workbench right pane:	
	Either	Or
	Double click **Deployment Configuration**.	a. Right-click **Deployment Configuration**. b. Select Properties on the Popup menu.
E	Press Alt+Enter on the keyboard.	

a. Result: One of the following will happen.

A configuration file Is:	Result
In the Installation \Data directory	The CIMPLICITY Configuration Editor opens the existing gefdepl.dplcfg file.
Not in the Installation \Data directory	You will be provided with prompts *(page 371)* to create a new file.

6. Right-click **Deployment Configuration**.

7. Select Properties on the Popup menu.

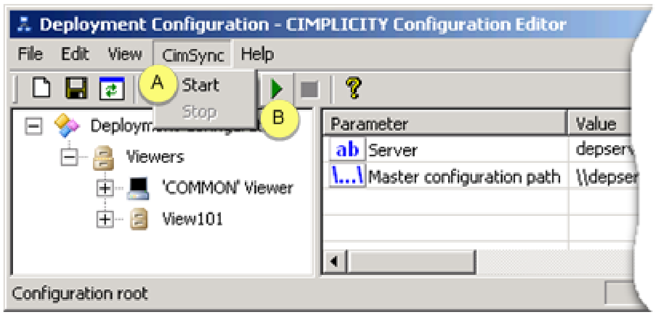
8. Right-click **Deployment Configuration**.

9. Select Properties on the Popup menu.

Step 3.3. Start Synchronization

CIMPLICITY deployment provides several methods for starting synchronization on a viewer.

- Do one of the following in the CIMPLICITY Configuration Editor.



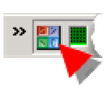
A	Click CimSync>Start on the menu bar.
B	Click the Run button  on the toolbar.

- (If deployment is set to start at system startup) reboot the Viewer.

Result: Synchronization will begin and run according to the specifications defined for the viewer.

Step 3.4. Review Synchronization Progress

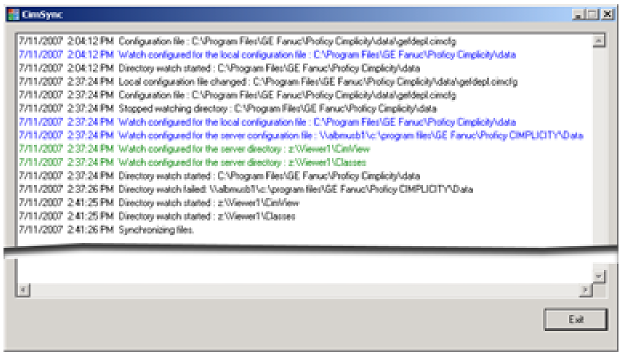
If the Configuration File instructs deployment not to hide CimSync when it starts *(page 397)* , a **Deployment** icon is placed on the Windows Task bar.



1. Do one of the following.
   - Double-click the Deployment icon.
   - Right-click the Deployment icon; select Restore on the Popup menu.

   A CimSync report window opens when you use either method.

   The report provides synchronization details, enabling you to confirm that deployment is proceeding accurately and, if not, what are the problems.



2. (Optional) Do one of the following to save the report as text.
   - Click the **CimSync** icon on the CimSync report title bar; select Save as Text on the menu.
   - Right-click the **CimSync** icon on the Windows Task bar; select Save as Text on the Popup menu.

3. Click the **CimSync** icon on the CimSync report title bar; select Minimize on the menu to hide the report.

Step 3.5. Stop Synchronization

1. Do any of the following if synchronization must be stopped on a viewer.
   - Click the Stop button on the CIMPLICITY Configuration Editor toolbar.
   - Click CimSync>Stop on the CIMPLICITY Configuration Editor menu bar.
   - Click Exit in the CimSync report window.
   - Right-click the **CimSync** icon on the Windows Task bar; select Close on the Popup menu.

   A message opens when you use any method asking if you want to stop the CIMPLICITY synchronization on this viewer.

2. Click Yes.

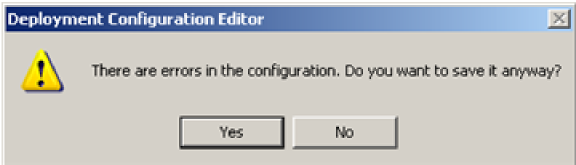
Synchronization with the viewer is stopped.

Deployment Configuration Error Reporting

Deployment Configuration Error Reporting

When you try to save the configuration file in the Deployment Configuration Editor, Deployment will validate your configuration. If it finds errors a message box will open to report that there is or are errors in the configuration and ask you if you want to save it.

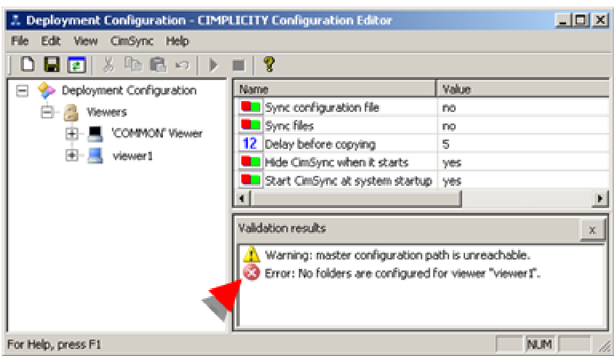
Click one of the following.



Yes	Saves the file and...
No	Does not save the file...

Results

- The message box closes after you click either button.
- The bottom half of the Configuration Editor right-pane displays the errors.



Deployment Error Messages

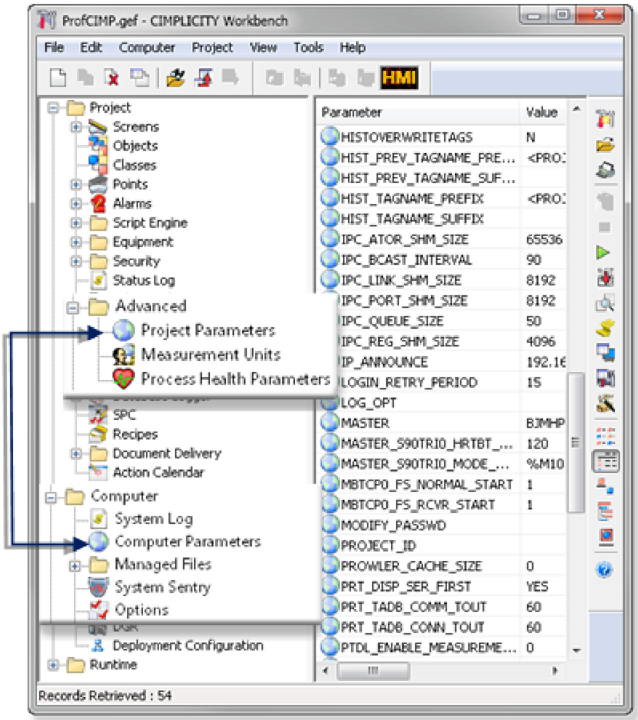
Check the named Viewer in the Deployment Configuration left pane and remove any invalid characters.Example

1. view1 is listed in the Deployment Configuration Editor left pane with two back slashes.
   \\view1
2. An error message displays in the lower right pane as follows.
   Error: "\\view1" Viewer name contains one or more invalid characters: \/:*?"<>|
3. \\view1 is renamed and is correctly listed as view1.

Global Parameters

About Global Parameters

The CIMPLICITY Workbench provides an easy way to view and, when instructed, to add or modify existing global parameters.



A Global Parameter:

- Defines parameters for the CIMPLICITY Base System and options that override default parameters in the software.
- Controls either:
- The system or
- A project.
- Can be changed in one of the following ways:
- Through user interfaces in CIMPLICITY Configuration functions.
- Through the Workbench in the System or Project section.

CIMPLICITY software stores the project's global parameter values in a global_parms.idt file. This interface enables you to avoid opening and editing the file. As a result, the Workbench application saves you time and helps you avoid accidental errors. However, even with this tool, you should only edit global parameters when it is absolutely necessary.

Warning: Improper global parameter configuration can cause adverse system effects. Do not add, modify or delete global parameters unless specifically instructed to by the documentation or a GE Intelligent Platforms representative.

Global Parameter Configuration

Global Parameter Configuration

The Workbench Global Parameters application enables you to edit a global parameter value without tampering with the global_parms.idt file to which the values are written. .

You can:

1 (page 415)	Add a global parameter to the Workbench list.
2 (page 418)	Edit a global parameter value.
3 (page 418)	Delete a global parameter from the list.

The procedure for each of these management functions is simple. However, the result of any modification can have a powerful impact on the system. Therefore, you will see warning messages every step of the way. They are there for a reason.

1. Add a Global Parameter to the Workbench List

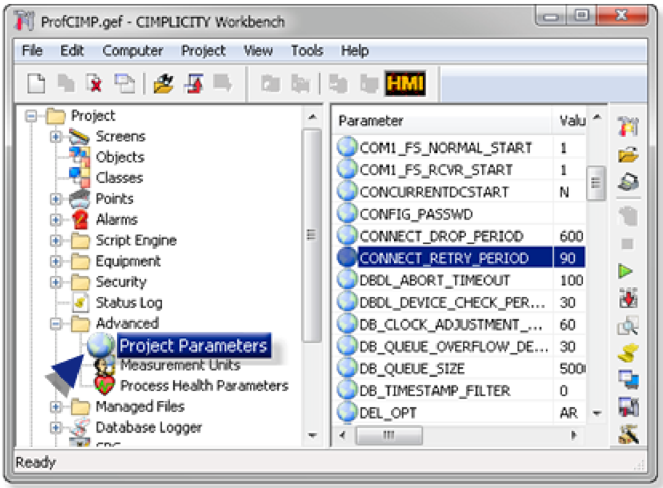
CIMPLICITY supports several global parameters that are not in the default list when you initially display the global parameters in the Workbench. Those global parameters do not have a value. Therefore, CIMPLICITY is not using them. You can easily add any of the global parameters to the Workbench list, set their values, update your configuration and, as a result, activate them.

Important: A global parameter applies to either the project or the system. It is important that you add the parameter to the correct folder. If you add it to the wrong folder, it will not be workable.

1. Display Project or Computer parameters

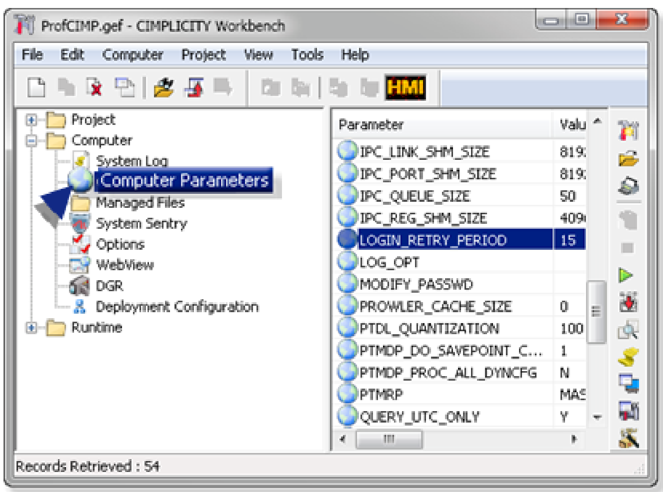
Project Parameters

Select **Project>Advanced>Project Parameters** in the Workbench left pane.



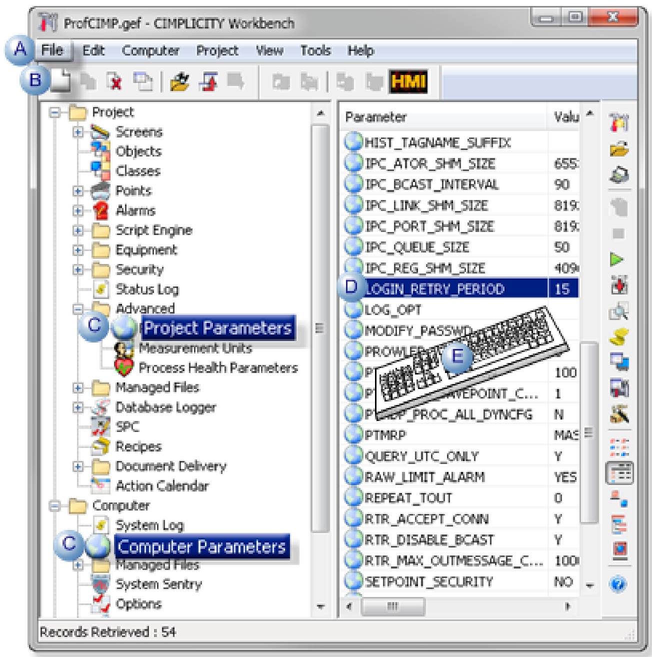
Computer (System) parameters

Select **Computer>Computer Parameters** in the Workbench left pane.



1. Add a Project or Computer parameter

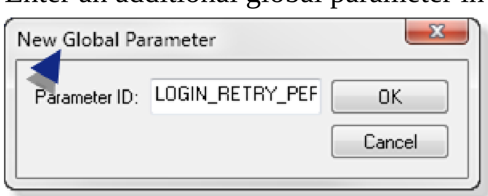
Do one of the following:



A	Click File>New on the Workbench menu bar.
B	Click the New Object button on the Workbench toolbar.
C	In the Workbench left pane, double click one of the following **Project Parameters or C omputer Parameters**.
D	1. In the Workbench right pane, right-click any global parameter and select New on the Popup menu.
E	Press Ctrl+N on the keyboard.

A New Global Parameter dialog box opens when you use any method for either project or computer parameters.

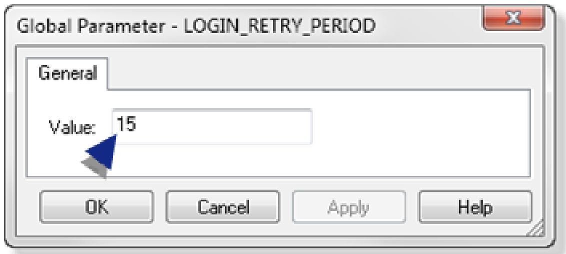
1. Enter an additional global parameter in the **Parameter ID** field.



2. Click OK.

   A Global Parameter - <name> dialog box opens.

3. Enter a value that applies to the global parameter in the **Value** field.



4. Click OK.

   A message box opens, warning that improper configuration can cause adverse effects.

5. Click OK if you are sure you want to continue.
6. Do a project configuration update.

   A message box opens asking you if you want to copy master configuration data to run-time data.

7. Click OK.

The global parameters are activated and their entered values are now the values that the CIMPLICITY system and/or project use.

2. Edit a Global Parameter Value

In the Workbench left pane:

1. Right-click one of the following.
   - **Project Parameters**.
   - **Computer Parameters**.
2. Select Properties on the Popup menu.
3. Right-click one of the following
   - **Project Parameters**.
   - **Computer Parameters**.
4. Select Properties on the Popup menu.

3. Delete a Global Parameter from the List

If it is necessary to de-activate a global parameter, you can by deleting it from the Workbench list.

Warning: Delete a global parameter only when instructed to by documentation or a GE Intelligent Platforms representative.

1. Select *(page 415)* either of the following.

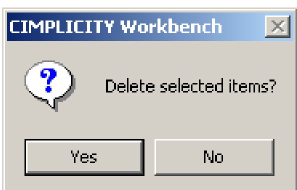
Select	To delete a:
Project Parameters	Project global parameter.
Computer Parameters	System global parameter.

2. Select the global parameter(s) you want to delete.

3. Do one of the following:

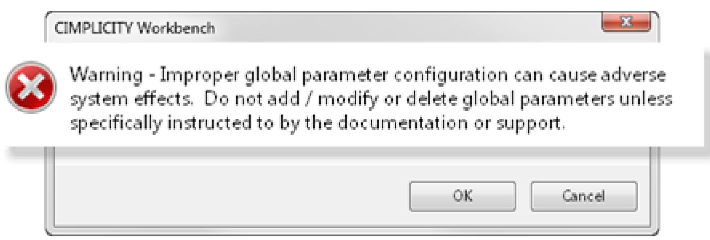
   - Click the Delete button  on the Workbench toolbar.
   - Click the right-mouse button on your selection; select Delete on the Popup menu.
   - Click Edit>Delete>Object on the Workbench menu bar.
   - Press Del on the keyboard.

   A message box opens asking you if you want to delete the selected items.

4. Click Yes.

   A message box opens warning you that improper configuration can cause adverse effects.

5. Click OK if you are sure you want to delete the global parameter.

   The global parameter is deleted from the list.

6. Do a CIMPLICITY configuration update.

The global parameter is deleted from the list and de-activated in the CIMPLICITY system and/or project.

Global Parameters

Global Parameters

Warning: Improper global parameter configuration can cause adverse system effects.

The information in this section should be used to add, modify or delete global parameters only if you are specifically instructed to by the documentation or GE Intelligent Platforms.

$AM_STATUS

For	Alarms Project
Purpose	Enable/disable the $AM_STATUS message.

Value	Enter one of the following.	
	Y	Enable the $AM_STATUS alarm message.
	N	Disable the $AM_STATUS alarm message
Default Value	N	

ACK_TOUT

For	Points and Alarms Project
Purpose	To specify the default time in minutes before an alarm is automatically acknowledged by the Alarm Management Resident Process.
Default for	**Acknowledge Timeout** field in the Alarm Options Properties dialog box for points and alarms.
Value	Number (of minutes).
Default Value	0 (no automatic timeout).

AE_STARTUP_CFG

For	A&E OPC Server System	
Purpose	To select when the A&E OPC Server will read the project configuration and populate the server configuration database.	
Value	The A&E OPC Server populates the server configuration database:	
	Y	At startup time. This might take some time (a few minutes) for very large projects to connect to Historian.
	N	When alarms are generated. This option will provide instant project connection.. **Note:** In this case server becomes fully configured only after every alarm has been generated at least once
Default Value	N	

ALARM_SAVE_COMMENTS

For	Alarms Project	
Purpose	To instruct the Alarm Viewer to remove or store alarm comments.	
Default for	Store alarm comments checkbox in the Alarm Properties dialog box.	
Value	Enter one of the following:	
	NO	Available only while an Alarm ID appears in the Alarm Viewer. When the Alarm ID has been reset, deleted, or automatically removed, the comment ceases to exist, or

	YES	Stores comments until 20 comments have been listed for one Alarm ID, regardless of whether the alarm has been acknowledged, reset, deleted, etc. When 20 comments have accumulated, the first comment entered is deleted to make room for the newest comment.
Default Value	NO	

ALLOW_UPDATE_WHEN_DISABLED

For	Ports Project	
Purpose	To specify if setpoints and unsolicited data should be processed when a port is dynamically disabled.	
Value	Enter one of the following:	
	YES	Process setpoints and unsolicited data continue to be processed. Polled data will continue to display old values.
	NO	When a port is dynamically disabled: • The associated devices are marked **Down.** • Setpoints and unsolicited data are not processed.
Default Value	NO	

AM_OLD_POINT_RESET

For	Alarm Manager Project	
Purpose	To control whether the reset of a point alarm will cause **Time** value displayed in the AMV to change from the alarm's generated time to the reset time. (Updating to the reset time only applies to point alarms, not general system alarms.) **Note:** The generated time value is not updated to the reset time after a manual reset.	
Value	Enter one of the following.	
	Y	Update the generated time to the reset time.
	N	Do not update the generated time to the reset time.
Default Value	N	

AM_RESET_ACK_STATE

For	Alarm Management Project
Purpose	To reset the Acknowledge state of an acknowledged alarm that has an Acknowledge and Reset deletion requirement to No after the alarm has been cleared. This forces a user to acknowledge the alarm again before it is deleted from the list of alarms.
Value	Enter one of the following:

	YES	Resets the acknowledge state to No
	NO	Does not reset the acknowledge state to No
Default Value	NO	

AM_SAVE_ACK_STATE

For	Points and Alarms Project	
Purpose	To retain the current Acknowledge state of an alarm when the point transitions from the HiHi (Alarm High) to the Hi (Warning High) state or from the LoLo (Alarm Low) to the Lo (Warning Low) state	
Value	Y	Retains the current acknowledge state
	N	Does not retain the current acknowledge state.
Default Value	N	

AMLP_HIGHWATERMARK

For	Alarm Line Printer Project
Purpose	To stop spooling alarm messages if more than a certain number of alarm messages are in the job queue for the alarm printer. When this global parameter is set to its default value, all alarm messages go to the spooler for the alarm printer. If you enter a number greater than zero (0), the AMLP program will spool alarm messages to the printer until the high watermark is reached. Subsequent alarm messages will be dropped until the low watermark, defined by AMLP_LOWWATERMARK, is reached.
Value	Number (of jobs in the print spooler)
Default Value	0 (no high watermark)

AMLP_LOSTALARMSPAGE

For	Alarm Line Printer Project	
Purpose	To print a page indicating that alarm messages were dropped because the AMLP program detected a high watermark for the print spooler. This global parameter is used only when AMLP_HIGHWATERMARK is set to a non-zero value. The page is printed when the AMLP program detects a low watermark and resumes sending messages to the print spooler.	
	Enter one of the following:	
Value	Y	Print a page indicating that alarms have been dropped.
	N	Do not print a page when alarms are dropped.
Default Value	N	

AMLP_LOWWATERMARK

For	Alarm Line Printer Project
Purpose	To restart printing alarms if the number of jobs being spooled falls below this number. This global parameter is used only when the AMLP_HIGHWATERMARK global parameter is set to a non-zero value.
Value	Number (of jobs in the print spooler)
Default Value	0

AMLP_MAX_QUEUE

For	Alarm Line Printer Project
Purpose	The Alarm Line Printer program (AMLP) assumes that there is no restriction for the size of the alarm message queue. If the output device is disabled, a virtual memory overflow can result AMLP_MAX_QUEUE global parameter can restrict the size of the alarm message queue. After you implement this global parameter, if the output device is disabled and the number of alarms in the alarm queue exceeds the value you specify, the alarm message **The alarm dropped has exceeded the configured size of alarm queue** is generated.
Value	Maximum number of messages in the queue.
Default Value	None

AMLP_USE_GEN_TIME

For	Alarm Line Printer Project	
Purpose	To specify whether an acknowledged or deleted alarm should be printed at the generation time or the time of the Acknowledge or Delete action.	
Value	Enter one of the following:	
	YES	The generation time is printed.
	NO	The time of the action (acknowledge or delete) is printed.
Default Value	YES	

AMLP_USE_RESET_TIME

For	Alarm Line Printer Project
Purpose	To have the alarm printer print the time an alarm was reset. Important: The global parameter, AM_OLD_POINT_RESET must be set along with AMLP_USE_RESET_TIME, as follows:
Value	Choose one of the following: To print the reset time, set: `AM_OLD_POINT_RESET = Y` and `AMLP_USE_RESET_TIME = Y` To print the generated time, set: `AM_OLD_POINT_RESET = N` and `AMLP_USE_RESET_TIME = Y`

Default Value	N

AMV_SOUND_LOOP

For	Alarm Sound Manager Project
Purpose	To provide additional support for continuous playback of alarm sound audio files.
Comment	When the attribute value is set to Y or y: 1. Alarm sound will be repeated in loop till the specified stop criteria is satisfied. 2. Sound will stop immediately if stop criteria is satisfied. 3. The alarm audio file will continue to play in loop, even if the replay option is not selected. 4. Alarm Sound file will pause immediately if muted and will resume playing when cleared. When the attribute value is set to N or n the existing behavior prevails.
Value	Y or y, N or n
Default Value	N or n

AMVS_BOOTPRJ

For	Alarm Sound Manager System
Purpose	To make the Alarm Sound Manager wait until a specific project, (which has also been configured to auto-start) has completely started before it (the Alarm Sound Manager) completes its startup
Value	The project name that should be completely started before the Alarm Sound Manager completes its startup. Example Project2
Comment	If more than one project is configured to start at boot time (e.g. on the Startup Options tab in the CIMPLICITY Options dialog box) enter the project name that the Alarm Sound Manager should wait for.
	Note: Add these system parameters to the Computer Parameters list. The list is located in the Workbench's **Computer**>**Computer Parameters** section.

AMVS_TIMEOUT

For	Alarm Sound Manager System	
Purpose	To specify the amount of time that the Alarm Sound Manager should wait for any project to completely start before it (the Alarm Sound Manager) completes its startup.	
Value	Values are as follows.	
	Min	1 minute

	Max	60 minutes
Default Value	10 minutes (If AMVS_TIMEOUT is not defined.)	
	Note: The Alarm Sound Manager will completely start as soon as Windows is started if the Alarm Sound Manager is configured to auto-start but no project is configured to auto-start.	

AUTO_CONFIG_AE_SRC

For	Points (alarm configured) System (computer)	
Purpose	To log Setpoint Audit trail events (i.e. $DOWNLOAD alarms) for Alarm configured points that are specified at Filter by Source to the Historian server.	
Value	Enter one of the following.	
	Y	Will be logged.
	N	Will not be logged.
Default Value	N	
Comment	The format of the string for the **Source** field, in this case, for the Set Point Audit trial event is modified to match the source of Filter, as follows. <PROJECT>/<RESOURCE ID>/<Ref ID>/<Ref ID>	

BASIC_LOAD_SCRIPT_EXTENSION

For	Basic Control Engine Project
Purpose	To direct Event Manager whether to run a read-only *.bclrt file or a read/write *.bcl file if both exist with the same name
Value	Y or y
	EMRP will use the following sequence till a script is found.
	1. Configured extension. 2. Based on the configured extension. • .bcl if the configured extension is .bclrt. • .bclrt if the configured extension is .bcl. • .bcl if the configured extension is neither. 1. .bclrt. **Note:** This third pass only happens if .bclrt has not already been tried.
	Not defined Not one of the values indicated above

	EMRP will use the following sequence till a script is found.	
	1. .bcl 2. .bclrt.	
Default Value	Y	**Note:** The default value applies to new projects.

BASIC_MULTITHREAD_COM

For	Basic Script Engine Project or System	
Purpose	To enable multi-thread script COM calls.	
Comments	Script COM calls prior to CIMPLICITY 7.0 occurred in a single-thread as follows. • COM calls were marshaled over to a single script thread to be executed. • The script thread would wait for the COM thread to complete a call. If a thread was already making a COM call, then the next thread had to wait for it to finish before it could make a COM call. Each thread would have to wait for completion of the prior COM call. Beginning with CIMPLICITY 7.0: • Each COM call can occur in its execution thread instead of being marshaled over to a single thread. • BASIC_MULTITHREAD_COM provides the ability to choose between using the single-thread or multi-thread COM calls in the Basic Script Engine. The initialization routine for the Basic Script Engine includes a flag that controls the threading behavior of the script engine. If BASIC_MULTITHREAD_COM is enabled, the flag will call COM objects in the thread of execution instead of being marshaled over to a single thread. **Note:** Some limited releases before CIMPLICITY 7.0 may also include this feature.	
Value	Choose either of the following.	
	Y	Initializes multi-thread COM calls.
	N	Initializes single-thread COM calls.
Default Value	N	**Note:** The default has been set to N in order to insure backward compatibility with scripts created in previous CIMPLICITY versions.

BCE_ENABLE_TRACING

For	Basic Control Engine Project

Purpose	To specify if the Trace command will output any trace information. The Event Manager Resident Process (EMRP) reads BCE_ENABLE_TRACING when the EMRP is started during project startup. If BCE_ENABLE_TRACING is set to N or n, the Trace command does not output any trace information. It simply returns before checking whether or not TraceEnable has been called. TraceEnable/ TraceDisable commands only affect the output of a single running event script thread. However, BCE_ENABLE_TRACING affects all scripts run in the EMRP on a single server.	
Value	Enter one of the following.	
	Y	Enables the parameter
	N	Disables the parameter
Default Value	N	

BCO_ARCHIVE_AFTER_N_SWEEPS

Advantages of using BCO_ARCHIVE_AFTER_N_SWEEPS include:

1. Since these activities are carried out in two different threads (parallel processing), there will not be any delay in posting the jobs to the device.
2. Archive/Delete can be delayed; archive/delete is not as high a priority as posting jobs. This will be helpful there is a high volume of broadcast messages in the plant.

BCO_AUTO_ARC_DEL

For	Broadcast (Order Execution Mgt) Project	
Purpose	To perform archiving or deletion of jobs from the History folder, based on the history queue limits, which are configurable through the Broadcast Queue Monitor Configure Device GroupsWeb page.	
Value	0	(Or any value other than 1 or 2) Auto archive and auto delete are disabled.
	1	Auto-archive the Job from History folder to Archive folder
	2	Auto-delete the Jobs from History folder.
Default Value	0	

BCO_DATA_FETCH_ERROR

For	Broadcast (Order Execution Mgt) Project			
Purpose	Determines what Broadcast will do when there is a data fetch error. Broadcast does the following based on whether you disable or enable BCO_DATA_FETCH_ERROR.			
	Disable BCO_DATA_FETCH_ERROR			
	Even when there is a data fetch error • Jobs are sent to the device. • An alarm and log are generated.			
		Note: In a normal scenario (BCO_DATA_FETCH_ERROR is disabled or enabled) the following occurs.		
		Whether or not there is an error, W or A is added to the filename, as follows.		
			WYSIWYG	W is added.
			Example 0001094033735734;**W**;111-11;111-11;All in One-SOLVE_PRT_EXPRT;221;YANE;111-11;.cwf	
			ASCII	A is added.
			Example 0001094033735734;**A**;111-11;111-11;All in One-SOLVE_PRT_EXPRT;221;YANE;111-11;.caf	
	Enable BCO_DATA_FETCH_ERROR			
	When there is a data fetch error the following occurs.			
	• Jobs: • Do not go to the device. • A zero is added to the filename, before the W or A. (W and A are described in the Disable BCO_DATA_FETCH_ERROR section.)			
			WYSIWYG	0 is added before the W.
			Example 0001094033735734;**0W**;111-11;111-11;All in One-SOLVE_PRT_EXPRT;221;YANE;111-11;.cwf	
			ASCII	0 is added before the A.
			Example 0001094033735734;**0A**;111-11;111-11;All in One-SOLVE_PRT_EXPRT;221;YANE;111-11;.caf	

	• Stay in the active folder of the device. This does not allow any of the subsequent jobs to be sent to the device. • A log is generated in the cor log. • An alarm is generated. The user must: 1. Open the file manually 2. Enter correct data for the objects / fields which had data fetch error 3. Rename the file by removing the appended 0, Result: the jobs will start being posted to the device.				
Value	0	(Or any value other than 1) Disabled			
	1	Enabled			
Default Value	0	Disabled			

BCO_DFO_NOT_AUTO

To enable or disable printing fail over from the secondary printer back to the primary printer.BCO_DFO_NOT_AUTO enabledWhen the:

1. Primary device of a device group goes down the jobs are sent to secondary device
2. Secondary device goes down broadcasts are sent to the primary device automatically, if it is available.
   BCO_DFO_NOT_AUTO disabledWhen the:
3. Primary device of a device group goes down the jobs are redirected to the secondary device.
4. Secondary goes down the jobs are not redirected to the primary automatically.
   The user will have to reset the devices from Broadcast Queue Monitor Web pages.**Note:** The fail over will occur when the secondary fails, e.g. runs out of paper, and the primary is back on line.

BCO_SWEEP_DELAY

For	Broadcast (Order Execution Mgt.) Project	
Purpose	To introduce a time delay between two successive sweeps.	
Value	Number (of seconds).	
Default Value	1	second

BIND_ADDR

Obsolete

BIND_ADDR is not used in CIMPLICITY v6.2 and higher.

To specify which IP address to use on a computer that has multiple IP addresses, and is not using Cabling Redundancy, enter the node name and matching IP address for the computer on the Hoststab in the CIMPLICITY Options dialog box.

Configuration must be for 2 NIC's as well as for cable redundancy.

BROWSE_TREE_UNDERSCORE

For	Project (for servers) System (for viewers)
Purpose	When viewing point data in a tree view, the tree view uses the period as the delimiter between branches. Using this global parameter allows the underscore character to be used.
Value	Any value turns BROWSE_TREE_UNDERSCORE on. CIMPLICITY code just looks to see if parameter is defined.
Default Value	Undefined

CE_MAX_DELAY

For	Basic Control Engine Project
Purpose	To specify the maximum delay time in seconds after which a late event will not be executed. For example, events may be delayed when there is a heavy load on the system.
Value	**Number** (of seconds).
Default Value	60

CE_MAX_THREADS

For	Basic Control Engine Project
Purpose	To specify the maximum number of simultaneous scripts allowed.
Value	**Number** (of scripts)**.**
Default Value	30

CE_MIN_STANDBY_THREAD_COUNT

For	Basic Control Engine Project
Purpose	Project Purpose To specify the number of threads that the Event Manager can allow to be in the idle state indefinitely.

Value	**Number** (of threads).
Default Value	0

CE_POOL_THREADS

For	Basic Control Engine Project
Purpose	To specify the maximum number of threads in a thread pool.
Value	**Number** (of threads).
Default Value	0 (will be assigned a value that is twice the number of logical processors in the system).

CE_THREAD_TIMEOUT

For	Basic Control Engine Project
Purpose	To specify the idle cache time in seconds before the threads are freed.
Value	**Number** (of seconds).
Default Value	900

CLR_TOUT

For	Alarm Management Project
Purpose	To specify a default time in minutes before an alarm is automatically reset by the Alarm Management Resident Process.
Default for:	**Reset Timeout** field in the Alarm Options Properties dialog box for points and alarms
Value	**Number** (of minutes).
Default Value	0

CONNECT_DROP_PERIOD

For	Login System
Purpose	To specify the time in seconds before an inactive login connection is dropped by the project. The time starts when all CIMPLICITY application windows have been exited. A user who opens a CIMPLICITY application window during this period will not be required to log in to CIMPLICITY software again.
Value	**Number** (of seconds).
Default Value	600 (Equals 10 minutes)

CONNECT_RETRY_PERIOD

For	Login System
Purpose	To specify the time interval for between retrying the connection to a remote project.
Value	**Number** (of seconds).
Default Value	90 **Note:** 90 = Equals 1½ minutes

Note: Entering a value of less than 30 seconds is not recommended.

COR_LOG_RP_SCAN_TIME

For	Database Logger Project
Purpose	To specify the periodic rate at which the Status Log file is scanned for added records.
Value	Number of seconds.
Default Value	300

DB_CLOCK_ADJUSTMENT_MONITOR

For	Database Logger Project
Purpose	Defines the rate at which system clock adjustments are checked. All synchronized timed events are adjusted when a system clock adjustment occurs.
Value	Number (of seconds). Enter a value from 1 through 3600
Default For	Clock adjustment monitor rate in the Database Logging Properties dialog box.
Default Value	60

DB_COMPACT_QUEUE_SIZE

For	Database Logger Project **CAUTION:** Do not modify this option unless instructed to by GE Intelligent Platforms support personnel.
Purpose	For Microsoft Access (As-Is product) databases, the compact queue holds the database requests. This global parameter defines the queue size for maintenance actions.
Value	Number
Default Value	1000

DB_COUNTER_FIELD

For	Database Logger Project
Purpose	To modify the name of the **seq_num** field that specified the record sequence number of each logged transaction in the following Database Logger Tables: • Alarm Management tables (ALARM_LOG and EVENT LOG) • Event Manager table (EM_LOG) • SPC tables • HDA tables • Tracker tables
Value	An up to 80 character name. The name may include or be entirely composed of uppercase characters (for example, SEQNUM) which is needed for certain database client applications that have problems accessing lowercase field names from an Oracle DMBS. Note: When you use the DB_COUNTER_FIELD global parameter, make sure that you drop all affected tables before running the project. Otherwise, the Database Logger will not be able to recreate them with the new sequence number field name specified by the DB_COUNTER_FIELD global parameter.
Default Value	`seq_num` (as a lowercase string)

DB_DBMS_QUEUE_SIZE

For	Database Logger Project **CAUTION:** Do not modify this option unless instructed to by GE Intelligent Platforms support personnel.
Purpose	Defines the database's queue size for non-table requests. For example, connect requests are non-table requests
Value	Enter the queue size. The minimum size is 5.
Default Value	1000

DB_DEBUG

For	Database Logger Project	
Purpose	To enable the dumping of diagnostic information for the Database Logger to the MAC_DL.out and MAC_PTDL.out files in your project's **log** directory.	
Value	Enter one of the following: A sum of any combination of:	
	0	Stop dumping diagnostic information–Must be explicitly specified to stop dumping diagnostic information to the log files.
	1	Print full syntax of ODBC error messages.
	2	Print details of database connection when established.

	4	Print syntax of ODBC statements if they fail.
	256	Print application-specific details.
	512	Print details of bulk insertion errors.
	1024	Rename/keep store and forward files after loading them. If you need additional diagnostic information, a value of 7 (1+2+4) is recommended. Additional undocumented values should only be used by GE Intelligent Platforms engineers as they send a great number of messages to MAC_DL.out and MAC_PTDL.out that could quickly fill your available disk space.
Default Value	0	

DB_ENABLE_MSEC

For	Database Logger System or Project	
Purpose	To create a separate column in which the CIMPLICITY Data Logger will store the sub-second portion of the timestamps to the nearest millisecond. This capability can be important when a database is not capable of storing time values with a timestamp accuracy of 1/1000 of a second, which is the accuracy that CIMPLICITY Data Logger achieves. In fact, most databases can only handle storing time values to the nearest second. Therefore when timestamps are inserted into the database, their values are either truncated or rounded, by the database, to a value that complies with the database's storage capabilities. You can create DB_ENABLE_MSEC in either the Project or the System folder. The result is as follows:	
	Folder	Global Parameter will be used by:
	System	Millisecond Logging compatible clients whom are started outside the scope of the project environment and copied into all new CIMPLICITY projects that you create (e.g. Starting a CimView screen outside of a project).
	Project	Only the project's logging processes and Millisecond Logging compatible clients started within the context of the project environment.
Value	Values include:	
	Y	Creates a separate column.
	N	Does not create a separate column.
Default Value	`msec` **is the default name of a field that** CIMPLICITY creates in the database. Note: CIMPLICITY names the field **msec**. To change this name, use the global parameter DB_MSEC_FIELD.	

Important:

1. If you want to enable DB_ENABLE_MSEC, create or enable the parameter before you create any tables in the database.

   CIMPLICITY creates the field as part of a table's primary key when it creates the table. CIMPLICITY creates tables when you start your project for the first time or reconcile a table through the Database Logger Configuration program. Therefore, you will have to drop any table that exists before you create or enable DB_ENABLE_MSEC so CIMPLICITY can recreate them with the new field.

2. If you want to disable DB_ENABLE_MSEC, drop existing tables and let CIMPLICITY recreate them without the parameter enabled.

   Failure to do this will cause logging of data to malfunction because the database will reject the inserted data due to constraint violations.

3. If you are logging data from multiple projects to the same database then all projects must have the feature identically configured.

4. If you are using DB_ENABLE_MSEC on more than one computer, including servers and viewers, add DB_ENABLE_MSEC on each.

5. Only Point Logging, Alarm Logging and Trending clients fully support this feature.

   To alter the name that CIMPLICITY uses to refer to or create the field in the database, create another global parameter, name it DB_MSEC_FIELD, and provide the appropriate string value for the field name.

DB_ENABLE_TRANSACTIONS

For	Database Logger Project	
Purpose	To override the default database functionality as requested by ODBC. The default functionality is determined by the ODBC driver. If the driver supports transactions, that is the default. If you set triggers on your tables, you should set this global parameter to N.	
Value	Enter one of the following:	
	Y	Forces logging to use transactions.
	N	Forces logging to not use transactions.
Default Value	Determined by the ODBC driver.	

DB_MSEC_FIELD

For	Database Logger System or Project
Purpose	To alter the default name that CIMPLICITY uses to refer to a field that reports the sub-second portion of timestamps in the database.
Value	String that will be the fractional portion of timestamps field name.
Default Value	msec

1. If you want to enable DB_MSEC_FIELD, create or enable the parameter before you create any tables in the database.

CIMPLICITY creates the field as part of a table's primary key when it creates the table. CIMPLICITY creates tables when you start your project for the first time or reconcile a table through the Database Logger Configuration program. Therefore, you will have to drop any table that exists before you create or enable DB_MSEC_FIELD so CIMPLICITY can recreate them with the new field.

2. If you are logging data from multiple projects to the same database then all projects must have the feature identically configured.
3. Only Point Logging, Alarm Logging and Trending clients fully support this feature.

DB_POINT_ID_FIELD

For	Database Logger Project
Purpose	To change the name of the **point_id** field in the following Database Logger tables: • Point Management data log tables
Value	You can define a field name of up to 80 characters, or the maximum supported by your database. Note: When you use the DB_POINT_ID_FIELD global parameter, make sure that you drop all affected tables before running the project. Otherwise, the Database Logger will not be able to recreate the tables with the new Point ID field specified by the DB_POINT_ID_FIELD global parameter.
Default Value	`point_id` (as a lowercase string).

DB_PROJECT_FIELD

For	Database Logger Project
Purpose	To change the name of the project field in the following Database Logger tables: • Alarm Management (ALARM_LOG and EVENT_LOG) • Event Manager table(EM_LOG) • Point Management tables (data and group log tables) • SPC tables • HDA tables • Tracker tables
Value	You can define a field name of up to 80 characters, or the maximum supported by your database. Note: when you use the DB_PROJECT_FIELD global parameter, make sure that you drop all affected tables before running the project. Otherwise, the Database Logger will not be able to recreate the tables with the new project field specified by the DB_PROJECT_FIELD global parameter
Default Value	`project` (as a lowercase string)

DB_QUEUE_OVERFLOW_DELAY

For	Database Logger Project **CAUTION:** Do not modify this parameter unless instructed to by GE Intelligent Platforms support personnel.
Purpose	To specify the delay in seconds between logging overflow errors for the Database Logger.
Value	**Number** (in seconds). This timeout must be configured to be: • Greater than 5 seconds and • Less than 86400 seconds (1 day).
Default Value	30

DB_QUEUE_SIZE

For	Database Logger System
Purpose	To specify a default queue size that supports the worst-case logging during database table's logging activity bursts.
Default for	**Database queue size** field in the Database Logging Properties dialog box.
Value	**Number** (for queue size).
Default Value	5000

DB_STATUS_LOG_OFF

For	Database Logger Project	
Purpose	To turn on or off the database logging messages in the Status Log. If this parameter is not defined or set to zero (0), the messages are logged.	
Value	Enter for one or both of the following bits:	
	1	Turns off Status Log messages that display when the logging data type has been superceded by a column type in the database table. For examples, messages would display if a UINT point type is changed to an UDINT point type after the table has been created.
	2	Turns off status log messages alerting a user that the Alarm Manager resident process cannot be found. For example, messages might appear on a Viewer where there is no Alarm Manager resident process.
Default Value	No entry	

Note: Remaining bits are reserved for future use.

DB_TABLE_FAILURE_RETRY

For	Database Logger Project

Purpose	Defines the number of seconds to wait before attempting to reconnect to a table after a table connection failure.	
Value	Number (of seconds)	
Default Value	0	Try to reconnect immediately

DB_TIME_FORMAT

<table>
<tr><td>For</td><td colspan="4">Database Logger Project</td></tr>
<tr><td>Purpose</td><td colspan="4">To define the time stamp format for your database so that time and date fields will be written correctly in the Store and Forward files. This parameter is for a database that does not use the default Microsoft Access (As-Is product) and SQL Server time stamp format, which is used by CIMPLICITY.</td></tr>
<tr><td>Value</td><td colspan="4">Timestamp (is case sensitive):</td></tr>
<tr><td></td><td colspan="4">Uppercase</td></tr>
<tr><td></td><td>Y</td><td colspan="3">Year</td></tr>
<tr><td></td><td>M</td><td colspan="3">Month</td></tr>
<tr><td></td><td>H</td><td colspan="3">Format options for the Hour specification are:</td></tr>
<tr><td></td><td></td><td>HHHH</td><td colspan="2">24-hour Note: The 24-hour format is used for Alarm logging and Point logging files stored by default in the %SITE_ROOT%\arc directory.</td></tr>
<tr><td></td><td></td><td>HH</td><td colspan="2">12-hour Include one of the following to specify AM or PM.</td></tr>
<tr><td></td><td></td><td></td><td>A</td><td>AM</td></tr>
<tr><td></td><td></td><td></td><td>P</td><td>PM</td></tr>
<tr><td></td><td colspan="4">Lowercase</td></tr>
<tr><td></td><td>d</td><td colspan="3">Day</td></tr>
<tr><td></td><td>m</td><td colspan="3">Minutes</td></tr>
<tr><td></td><td>s</td><td colspan="3">Seconds</td></tr>
<tr><td></td><td>t</td><td colspan="3">Ticks (100 ticks=1 second)</td></tr>
<tr><td>Default Value</td><td colspan="4">Microsoft Access and SQL Server times tamp format.</td></tr>
</table>

DB_TIMESTAMP_FIELD

For	Database Logger Project

Purpose	To modify the name of the unique **timestamp** field which specifies the date and time of each logged transaction in the following Database Logger tables. • Alarm Management tables (ALARM_LOG and EVENT_LOG) • Event manager table (EM_LOG). • Point management tables (both data and group log tables) • SPC tables • HDA tables • Tracker tables
Value	An up to 80 character name. The name may include or be entirely composed of uppercase characters, e.g. DATETIME, which is needed for certain database client applications that have problems accessing lowercase field names from an Oracle DBMS. When you use the DB_TIMESTAMP_FIELD global parameter, make sure that you drop all affected tables before running the project. Otherwise, the database logger will not be able to recreate them with the new timestamp field name specified by the DB_TIMESTAMP_FIELD global parameter.
Default Value	timestamp (as a lower case string).

DB_TIMESTAMP_FILTER

For	Database Logger Project		
Purpose	To allow either rounding or truncation of all timestamps in records logged by database logging. This includes the Point, Alarm and Event Logging, SPC and Tracker.		
Default For	**Enable timestamp filtering** field in the Parameters tab of the Logging Properties dialog box.		
Value	Enter one of the following:		
	A	Rounds the timestamp up or down in milliseconds.	
	-A	Truncates the timestamp at milliseconds.	
	Example If a timestamp filter of 30000 is specified, the following timestamps will be logged as follows:		
	Actual Timestamp		Logged Timestamp
	3/1/99 12:22:37.730		3/1/99 12:22:30.000
	3/1/99 14:37:24.102		3/1/99 14:37:30.000
	3/1/99 23:59:51.107		3/2/99 00:00:00.000
	If a timestamp filter of -30000 is specified, the following timestamps will be logged as follows:		
	Actual Timestamp		Logged Timestamp
	3/1/99 12:22:37.730		3/1/99 12:22:30.000
	3/1/99 14:37:24.102		3/1/99 14:37:00.000
	3/1/99 23:59:51.107		3/1/99 23:59:30.000
Default Value	0		

DB_TIMESTAMP_PRECISION

For	Database Logger Project **CAUTION:** Do not create or modify this option unless instructed to by GE Intelligent Platforms support personnel.
Purpose	To specify an alternate timestamp precision for an ODBC data source that does not accept the default provided by the CIMPLICITY Database Logger.
Value	Use 27.7 unless you are instructed otherwise by GE Intelligent Platforms support personnel.
Default Value	27.7

DB_TIMESTAMP_UTC_FIELD

For	Database Logging Project
Purpose	To rename the existing **Timestamp_utc** column.
Value	New name for the column.
Default Value	**Timestamp_utc**

DBDL_DEVICE_CHECK_PERIOD

For	Database Logger Project
Purpose	To specify a default interval (in minutes) that the Database Logger should wait between disk scans for the **Disk full scan rate (min)** field in the Database Logger Logging Properties dialog box.
Value	**Number** (of minutes).
Default Value	30

DC_CACHE_DIAGNOSTICS

For	Device Communications Project
Purpose	To disable the caching of internal points for all device communications in a project.
Comments	A message is logged to the OUT file if the diagnostics are disabled. Configure <PORT>_CACHE_DIAGNOSTICS *(page 478)* to enable or disable diagnostic caching for a single port.
Value	Enter one of the following:

	N or n	Does not disable the caching.
	Y or y	Disables the caching.
Default Value	N	

DC_RETRY_ONE_DEVICE

For	Device Communications Project or System	
Purpose	When an enabled device fails to respond to a read or write request from a device communications driver for several consecutive attempts, it is said to be down and will be polled at the Retry interval rate until it responds. By default, a device communication driver tries all down devices at each Retry interval. You can use this global parameter to change the behavior so that the driver will retry only one down device per Retry interval. The down devices are retried using a Round Robin algorithm (at the first Retry interval, the first down device is retried, at the next interval, the second down device is tried, etc.). This reduces the elapsed time for the Retry interval and is useful when many enabled devices are down.	
Value	Enter one of the following:	
	TRUE	Poll one device per Retry interval
	FALSE	Poll all devices at each Retry interval
Default Value	FALSE	

DC_UNAVAIL_NAN

For	Device Communications Project Level	
Purpose	To enable the device communications interface to detect a floating point that has a value of NAN.	
Comment	Device points with NAN value will be marked as unavailable.	
Value	Enter one of the following.	
	Y	Enables floating point with NAN value detection.
	N	Does not enable floating point with NAN value detection.
	Default	N

DEL_OPT

For	Alarms Project	
Purpose	To specify the default requirements for alarm deletion.	
Default For	**Deletion requirements** on the Alarm Options tab of the Alarm Definition dialog box.	
Value	Enter one of the following:	
	A	Acknowledged only
	R	Reset only

	AR	Acknowledge and reset
Default Value	AR	

DEVICE_DOWN_DEVICE_REF

To set the Device ID as the reference for the alarmA $DEVICE_DOWN alarm occurs when a device stops talking to a PLC.There are three components that make up a unique alarm

1. The Alarm ID which in this case is $DEVICE_DOWN;
2. The Resource ID, which in this case it takes it from the device's resource and
3. A Reference ID.
   As a result, each device will have a unique alarm, regardless of its resource. The Alarm ID is static for all $DEVICE DOWN alarms; the resource changes based on the device; now the **Reference ID** is the **Device ID**.

DEVICE_TIMESTAMP_UTC

For	Time stamp Project
Purpose	To change the default local time stamp reference that is encoded in the unsolicited communication from a selected device that uses a UTC time base. This applies to the following communications devices: Modbus TCP/IP or S90 Triplex.
Value	Y: UTC time base is used as default. N: Local time stamp is used as default.
Default Value	N

DGR_STOPPED_UPDT_DELAY

For	Dynamic Graphic Replay System
Purpose	To set a delay (in milliseconds) that the DGR waits to activate the User Interface buttons after a last point is added from a client application.
Comment	Setting `DGR_STOPPED_UPDT_DELAY` helps make it less likely that an operator will press buttons before the DGR is ready. **Note:** This system global parameter must be set on all viewers.
Value	2000 milliseconds or higher. **Note:** The value is ignored if it is less than 2000.
Default	2000

DOMAIN_USER_AUTOLOGIN

For	Domain User Authentication Project
Purpose	To support authentication of domain users in CIMPLICITY after explicit logout.
Value	Enter one of the following:
	Y or y Allows domain user to log in automatically after explicit logout.
	N or n User must provide login credentials after explicit logout.
Default Value	Y or y

DONT_VERIFY_ESPOINT_FRID

For	Enterprise Server Points Project	
Purpose	To change the behavior of Setpoint Security for Enterprise Server points.	
Value	Enter one of the following:	
	Y	Setpoint Security for all points from a given provider are enforced against the **Resource ID**s as configured on the provider. This means that a setpoint against any of these points with resources not configured on the Enterprise Server project will always fail since a resource that is not configured cannot be in a user's view.
	N	If an Enterprise Server project contains the same **Resource ID**s as a given provider, then Setpoint Security for points from the provider are enforced against the Enterprise Server Resource IDs. If a resource for a given point is not configured on the Enterprise Server, then Setpoint Security for that point is enforced against the remote project's resource.
Default Value	N	

DOWNLD_PASSWD

For increased security, this parameter is obsolete beginning with CIMPLICITY v10.0 .

In prior versions of CIMPLICITY the DOWNLD_PASSWD was stored in "clear text." During a project upgrade if CIMPLICITY finds a DOWNLD_PASSWD parameter, it creates a secure way of storing the parameter, and fills the old DOWNLD_PASSWD parameter with random data. If you use the setpoint password functionality you should upgrade all viewers to the most recent version. If you decide to continue using old viewers that require setpoints, you need to put a clear text password in the current project global parameter, but this reduces recent security enhancements to CIMPLICITY.

DT_UPD_INTERVAL

For	Date/Time Points Project
Purpose	To set the interval in seconds to update a Project's Date & Time system points.

Value	**Number** (of seconds).
Default Value	1

EM_SCRIPT_COMPILE_WAIT

For	Event Manager Project
Purpose	To specify the time to wait before proceeding to use the generated DLL.
Value	**0 - 30000**
Default Value	**10000**

EM_SCRIPT_RECOMPILE_ALWAYS

For	Event Manager Project	
Purpose	To specify if the script must be forcefully recompiled on every project start.	
Value	**Y or y**	.NET Assemblies will be recompiled on every project start
	N or n	.NET Assemblies will not be recompiled on every project start
Default Value	**Y or y**	

EM_SCRIPT_TRACING

For	Event Manager Project
Purpose	Automatically trace the execution of all Event Manager scripts. The trace files are: • Located in the project's log directory and • Named <EVENT_ID>_<THREAD>.TXT Where • `Event ID` is the event that triggered the event • `Thread` is the Basic thread object that is running it. The output file contains the following information: <time>,<script name>,<line number> Where • `time` is the number of seconds since the start of time. • `script name` is the name of the bcl file and • `line number` is the line within the file.
Value	Y or y **Note:** Y or y =Trace values generated as the scripts run.

Default Value	

Note: Enabling this feature will increase CPU utilization and make scripts run slower. This is a diagnostic tool.

EMLOG_FLUSH_LIMIT

For	Event Manager Project
Purpose	To allow users to configure how many events can occur to force the log buffer to be flushed.
Value	An integer from 0-10
Default Value	0

Note: If a value attempts to be greater than 10 it defaults to 0. This is due to a typical message size where the number of segments per message is 10.

ENABLE_MILLISEC_FOR_TREND_TIME

For	Trend Control Time System or Project		
Purpose	To specify is the Slider Value Time should or should not display in milliseconds.		
Value	Enter one of the following.		
	Y/y/1 or Not listed	Enabled	Displays `Hour:Minute:Seconds:Milliseconds`.
	N/n/0	Disabled	Follows the regional time settings.
	Default	Enabled	

EU_AUDIT_TRAIL

For	Roles Project	
Purpose	To provide control over setpoint event logging results by logging the point value in the setpoint audit trail ($DOWNLOAD event) in the EU and Measurement Unit converted value. **Note:** Normally the point value logged for a setpoint is the raw value of the point.	
Value	YES	The EU converted value of the point will be logged (assuming that an EU conversion is defined on the Conversion tab in the Point Properties dialog box.)
	YESLABEL	Any defined EU label for the point will be included just after the point value in addition to printing the EU converted value.
	NO	Trail is in device units; produces no action
Default Value	NO	

EXPRESSION_TRACE_LEVEL

For	Historian Connection Project/System	
Purpose	To trace Historian related problems with connections and expressions.	
Value	0	No trace file is generated.
	1	A trace file is generated with any unavailable tags (incorrect tag names).
	2	A trace file contains all of the errors including bad tags, which includes incorrect tag types, bad quality and no data available.
Default Value	None	
Results	Results are entered in the following TRACE files.	
	Trace File	Generated when:
	CimView.TRACE	When CimView is launched.
	MAC_PTDP.out	PTMDP runs
	Note: The TRACE file location depends on whether or not `EXPRESSION_TRACE_LEVEL` is configured for a selected project or as a system (computer) parameter. The file locations are as follows.	
	For	Location
	Project	...\<project folder>\log
	System	...\Proficy\Proficy CIMPLICITY\LOG

<FACEPLATE NAME>_ZOOM

For	Proficy Process Systems Project	
Purpose	To specify the zoom size of a selected faceplate to conform to your system's resolution.	
Value	Number that represents the percent zoom. The acceptable range is 20 to 250. Example	
	70	70% of the size of the original faceplate.
	50	50% of the size of the original faceplate
	200	200% of the size
Default Value	100	
Comments	<Faceplate_Name> is the name of the faceplate file that will zoom to the specified size. Example Global parameters for the following faceplate files are as follows.	
	Faceplate File	Global Parameter
	ADV_PID_fp.cimrt	ADV_PID_fp_ZOOM
	AI_fp.cimrt	AI_fp_ZOOM

	ALARM_A_fp.cimrt	ALARM_A_fp_ZOOM
	ALARM_D_fp.cimrt	ALARM_D_fp_ZOOM
	AO_fp.cimrt	AO_fp_ZOOM
	DC2S_fp.cimrt	DC2S_fp_ZOOM
	DC3S_fp.cimrt	DC3S_fp_ZOOM
	DI_fp.cimrt	DI_fp_ZOOM
	DOUT_fp.cimrt	DOUT_fp_ZOOM
	IND_A_fp.cimrt	IND_A_fp_ZOOM
	IND_D_fp.cimrt	IND_D_fp_ZOOM
	INTERLOCK_fp.cimrt	INTERLOCK_fp_ZOOM
	MANUAL_SP_fp.cimrt	MANUAL_SP_fp_ZOOM
	PBUTTON_fp.cimrt	PBUTTON_fp_ZOOM
	PID_fp.cimrt	PID_fp_ZOOM
	RAMPSOAK_fp.cimrt	RAMPSOAK_fp_ZOOM
	TOTALIZE_fp.cimrt	TOTALIZE_fp_ZOOM

FIRST_WEEK_DAYS

For	Project Management System
Purpose	To set the minimum number of days in a week in the current year required to qualify for the first week of the year. This global parameter is used for the $PROJECT.DATE.WEEK system point.
Value	**Number** (of days).
Default Value	1

FP_LEFT

For	Proficy Process Systems Project
Purpose	To specify the location of the faceplate's left edge when it is opened from a mimic object. This left-coordinate overrides positioning the left edge of the faceplate at the cursor location.
Value	Number representing the Pixel location starting from left of the primary monitor screen where the left edge of the faceplate will be positioned.
Default Value	Cursor location left coordinate.

FP_TOP

For	Proficy Process Systems Project
Purpose	To specify the location of the faceplate's top edge when it is opened from a mimic object. This top coordinate overrides positioning the top edge of the faceplate at the cursor location.
Value	Number representing the Pixel location from the top of the primary monitor screen where the top edge of the faceplate will be positioned.
Default Value	Cursor location upper coordinate.

GSM_ANNUN_ALARM_H1

For	CimView System or Project	
Purpose	To specify the color and blink state to be displayed when the alarm state for an analog point is Warning High.	
Value	**Color Number** and **Blink** option Format is <color_number><blink>	
	Color_number	
	A number from **0** through **15**	Selects a color **Note:** The color associated with each number is defined in the rgb.dat file.
	Blink	
	Y	Blink
	N	No blink.
	Example 3Y **Note:** 3Y = Blue, Blink	
Default Value	14N **Note:** 14N = Aqua, No Blink	

GSM_ANNUN_ALARM_H2

For	CimView System or Project	
Purpose	To specify the color and blink state to be displayed when the alarm state for an analog point is High.	
Value	**Color Number** and **Blink** option Format is <color_number><blink>	
	Color_number	
	Number from 0 through 15	Selects the color **Note:** The color associated with each number is defined in the rgb.dat file.
	Blink	
	Y	Blink

	N	No blink.
	Example 13Y **Note:** 13Y = Fuschia, Blink	
Default Value	15N **Note:** 15N = (Yellow, No Blink)	

GSM_ANNUN_ALARM_L1

For	CimView System or Project	
Purpose	To specify the color and blink state to be displayed when the alarm state for an analog point is Warning Low.	
Value	**Color Number** and **Blink** option Format is <color_number><blink>	
	Color_number	
	A number from 0 through 15	Selects the color **Note:** The color associated with each number is defined in the rgb.dat file.
	Blink	
	Y	Blink
	N	No blink.
	Example 8Y **Note:** 8Y = Teal, Blink	
Default Value	10N **Note:** 10N = Green, No Blink	

GSM_ANNUN_ALARM_L2

For	CimView System or Project	
Purpose	To specify the color and blink state to be displayed when the alarm state for an analog point is Low.	
Value	**Color Number** and **Blink** option Format is <color_number><blink>	
	Color_number	
	Number from 0 through 15	Selects the color **Note:** The color associated with each number is defined in the rgb.dat file.
	Blink	
	Y	Blink
	N	No blink.

	Example 4Y **Note:** 4Y = Maroon, Blink		
Default Value	13N **Note:** 13N = Fuchsia No Blink		

GSM_ANNUN_DIG_OFF

For	CimView System or Project	
Purpose	To specify the color and blink state to be displayed when the Boolean point is 0.	
Value	**Color Number** and **Blink** option Format is <color_number><blink>	
	Color_number	
	A number from 0 through 15**.**	Selects the color **Note:** The color associated with each number is defined in the rgb.dat file.
	Blink	
	Y	Blink
	N	No blink.
	Example 13Y **Note:** 13Y = Fuchsia Blink	
Default Value	6N **Note:** 6N = Purple, No Blink	

GSM_ANNUN_DIG_ON

For	CimView System or Project	
Purpose	To specify the color and blink state to be displayed when the Boolean point is 1.	
Value	**Color Number** and **Blink** option Format is <color_number><blink>	
	Color_number	
	A number from 0 through 15	Selects the color. **Note:** The color associated with each number is defined in the rgb.dat file.
	Blink	
	Y	Blink
	N	No blink.
	Example 6N 6N = Purple No Blink	
Default Value	7N **Note:** 7N = White, No Blink	

GSM_ASC_FONT_NAME

For	CimEdit/CimView (Classic ASCII File Imports) System or Project
Purpose	To specify the name of the font to use for text on an ASCII (.ASC) screen.
Value	**Font Name**. **Note:** The font must be a TrueType font.
Default Value	**MS Console**.

GSM_ASC_FONT_SIZES

For	CimEdit/CimView (Classic ASCII File Imports) System or Project
Purpose	To allow manipulation of (non-scalable) font sizes when a screen is imported from the Classic system to CimView. GSM_ASC_FONT_SIZES uses a list of the 16 values to use for the 16 text sizes in an ASCII (.ASC) file (GRE text sizes 0 to 15, which are stored in the .ASC file as -1 to 14).
Value	An increase or decrease of one or all default values to increase or decrease the font display. **Example** You want the fonts to display 25% larger on a CimView screen that you imported from the Classic system. Enter values that are 25% larger than the default values as the GSM_ASC_FONT_SIZES value. 140, 175, 210, 350, 455, 700, 753, 753, 753, 753, 753, 753, 753, 753, 753, 753
Default Value	112, 140, 168, 280, 364, 560, 602, 602, 602, 602, 602, 602, 602, 602, 602, 602.

GSM_ASC_SCALE

For	CimEdit/CimView (Classic ASCII File Imports) System or Project
Purpose	To specify a floating-point number (for a CimView file that was imported from the Classic system) that represents the number of document units per world coordinates when converting ASCII (.ASC) screens to .CIM format. ASC files store screen information in floating point "world coordinates". The screens are 100.0 world coordinates wide by 60.0 world coordinates high. Document units in .CIM files are integers in TWIPS (twentieth of a point, 1440 TWIPS/inch).
Value	**Number** (for **doc. units/world coord)** Example If you want your .ASC screens to be about six inches wide on the display, you should use a scale factor of (1440 doc. units/inch)*(6 inches/screen)/(100 world coord./screen) which equals 86.4 doc. units/world coord.
Default Value	**86.4**

GSM_CACHE_FILE

"<FileName>" [Lib] [Lock] [Project <ProjectName>] [Setvar <VarName> <Var Value>]

For	CimView System or Project
Purpose	New format for each line in the cache file.
Comment	

Value	Filename	Path for the cache file.
	Lib	Indicates that the screen should be loaded or not loaded into the library cache.
	Y	Load the screen.
	N	Do not load the screen.
	Project	Provides the name of the project for unqualifed points in the screen.
	Lock	Indicates the screen should be locked in the cache.
	Y	Lock the screen in cache.
	N	Do not lock the screen in cache.
	SetVar	Sets the value for the screen. A valid entry is a string with the name of the variable and the value of the variable. There may be multiple Setvar statements for the cache entry.
	VarName	Name of the variable.
	VarValue	Value of the variable.

Items in [ ] indicate that it is an optional entry.

GSM_CACHE_SIZE

For	CimView System or Project
Purpose	To increase the cache size if you require faster screen repaint times when switching between screens.
Value	**Number** (of screens that should be kept in cache memory).
Default Value	8

Note: Increasing this number uses additional memory and may have a negative effect on the performance of other transactions.

GSM_CACHE_USE_VARS

For	CimView System or Project	
Purpose	To improve the screen caching algorithm.	
Comment	You should explicity add GSM_CACHE_USE_VARS to the project parameters.	
	Y	The cache logic searches for a screen with the same screen name, project, and initial variables.
	N	The cache logic searches for a screen with the same screen name and project.
Default Value	N	

GSM_DEBOUNCE_OLD_EVENTS

For	CimEdit System	
Purpose	To disable event debouncing on CimView screens created before CIMPLICITY v7.0.	
Value	Enter one of the following.	
	Y	Enables event debouncing.
	N	Disables event debouncing
Default Value	Y	
Comment	When `GSM_DEBOUNCE_OLD_EVENTS` is set to N, the same event can be queued up multiple times before it completes execution.	

GSM_DELAY_PARSEEXPRESSIONS

For	CimEdit Project or System (computer)	
Purpose	To delay parsing the expressions on a CimView screen until after the screen is displayed.	
Value	Enter one of the following.	
	Y	Delays parsing expressions until after the screen is displayed.
	N	Does not delay parsing expressions.
Default Value	N	

GSM_EXPONENT_PRECISION

For	CimView System or Project
Purpose	To do the following: If: A Text object on a CimView screen has a "General" format and The number of digits exceeds this parameter, Then: The number is displayed in scientific notation with that number of significant digits.
Value	**Number** (of digits that will prompt a scientific notation display). Example 3 **Note:** The number 1234 displays as 1.23e+3.
Default Value	6

GSM_FRAMES_EXPOSE_POINT_TARGETS

For	CimView System or Project
Purpose	To expose objects with screen level frame containers to the point target tools as if the objects were at the screen level.
Value	Enter one of the following:

	Y	Exposes the objects. All other right mouse button tools still see the same objects.
	N	Preserves the frame behavior.
Default Value	N	

GSM_GLOBAL_SCRIPT

For	CimEdit/CimView System or Project
Purpose	To load and compile the specified scripts when CimEdit or CimView starts. You can specify multiple files in the global parameter by separating them with semicolons (;). Any script that fails to compile will not be available after startup.
Value	**String** (of files that can be loaded and compiled) Example `D:\SCR\MISCR.TXT;D:\SCR\TKSETS.TXT`
Default Value	None

GSM_GLOBAL_SCRIPTCFG

For	CimEdit/CimView System or Project
Purpose	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 `"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. • GSM_GLOBAL_SCRIPTCFG is supported by CimEdit global script features. • The command line argument `/LoadScriptCfg` can also be used to specify the global configuration file.
Value	**String** File name Example GlobalScripts.txt
Default Value	None

GSM_LIB_CACHE_SIZE

For	CimEdit/CimView System or Project
Purpose	To increase the cache size for library screens (screens that contain link source objects or class object graphics). This is similar to GSM_CACHE_SIZE**.** This parameter affects console CimView sessions.
Value	**Number** (of screens that should be kept in cache memory).
Default Value	4

Note: Increasing this number uses additional memory and may have a negative effect on the performance of other transactions.

GSM_OVERRIDE_POINT_FMT

For	CimEdit/CimView System or Project
Purpose	To adjust the format used to display point values. This is used in most, but not all, parts of CimView. The format is specified using a "printf" style format string, just like in the Custom display format of a text object in CimEdit.
Value	A "printf" style format string. Example `%-8.3e`
Default Value	nothing (does not override the display)

GSM_SETPOINT_WAIT_TIMELIMIT

For	CimView Project or System (computer)	
Purpose	To control the timeout (milliseconds) for doing setpoint actions before the setpoint action gets canceled.	
Value	Values are as follows:	
	Min	200 milliseconds
	Max	N/A
Default Value	4000 milliseconds (If GSM_SETPOINT_WAIT_TIMELIMIT is not defined.)	
	Note: If a value smaller than the minimum is entered, the default value of 4000 ms will be used.	

GSM_SPCONFIRM_DEFAULT

For	CimView System or Project
Purpose	To specify a button other than **OK** to be the default in procedure or action dialog boxes. Procedure dialog boxes contain two buttons: OK and Cancel. Action dialog boxes contain three buttons: OK, Skip, and Cancel.
Value	Select one: • OK, • Skip or • Cancel.
Default Value	OK

Note: If you select a default that is not available in the dialog box, CimView uses OK as the default button.

GSM_STALE_COLOR_KEY

For	CimView System or Project	
Purpose	To display a last available data value in a color rather than display the default text if a point being displayed in a Text object on a CimView screen is unavailable and there is last available data. If no last available data exists, the default text is displayed.	
Value	**Numbers** to specify RGB color indices. Format is: RGB(<red>,<green>,<blue>) where <red>,<green>,<blue> are the red, green, and blue color indices from 0 to 255. Examples	
	`RGB(0,0,0)`	**Note:** The value = black
	`RGB(255,255,255)`	**Note:** The value = white.
Default Value	The Text object displays the default text when the point is unavailable, regardless of whether a last available value exists or not.	

GSM_TERMSERV_CACHE_SIZE

For	CimView using Terminal Services System or Project
Purpose	To limit the CimView cache when running on a terminal server.
Value	Number (of screens)
Default Value	0

GSM_UNAVAIL_COLOR_KEY

For	CimView System or Project
Purpose	To specify the color index number (0 to 255) used to indicate an unavailable point value on CimView screens.
Value	**Number** from 0 (Black) through 255 (White) **Note:** The color associated with each number is defined in the rgb.dat file.
Default Value	0

GSM_UNAVAILABLE_TIMELIMIT

For	CimEdit/CimView System or Project
Purpose	To adjust how long CimView waits before displaying unknown points as unavailable.
Value	Number (of milliseconds to wait)
Default Value	2000 **Note:** 2000 = 2 seconds

HIST_FORCE_HIST45_TYPES

For	Historian Project	
Purpose	To allow the Historian v4.5 types to be forced to be created. This covers the case where the local Historian client tools are Historian v5.0 and higher and the remote Historian archive is v4.5 or older.	
Comment	If the local client tools are Historian v4.5, this parameter is not needed; Historian v4.5 compatible types will be used automatically.	
Value	Y or y	Force the use of Historian v4.5 compatible tag types.
Default Value	Not defined	

HIST_PREV_TAGNAME_PREFIX

For	iHistOPC service Project
Purpose	To hold the previous prefix text for a Historian tag name.
Comment	HIST_PREV_TAGNAME_PREFIX will be created the first time the user stops a running project; from then on the value will be updated accordingly.
Value	Previous prefix text for a Historian tag name.
Default Value	No default value.

HIST_PREV_TAGNAME_SUFFIX

For	iHistOPC service Project
Purpose	To hold the previous suffix text for a Historian tag name.
Comment	HIST_PREV_TAGNAME_SUFFIX will be created the first time the user stops a running project; from then on the value will be updated accordingly.
Value	Previous suffix text for a Historian tag name.
Default Value	No default value.

HIST_TAGNAME_PREFIX

For	Viewers (e.g. CimView, DGR) iHistOPC service for Historian Tag Generation Project
Purpose	To hold the prefix text for a Historian tag name.
Value	Prefix text for a Historian tag name.
Default Value	<PROJECTNAME> Where <PROJECTNAME> is the token which will be replaced with the actual project name when this global parameter is used.

HIST_TAGNAME_SUFFIX

For	Viewers (e.g. CimView, DGR iHistOPC service for Historian Tag Generation Project
Purpose	To hold the Suffix text for a Historian tag name.
Value	Suffix text for a Historian tag name.
Default Value	.VALUE

HIST_TIME_STAMP_TYPE

For	Viewers Project	
Purpose	To configure how Historian Tag Timestamp is processed from CIMPLICITY project.	
Comment	By default, the parameter is not defined or the expected value not specified.	
Value	X	CIMPLICITY Timestamp is logged to Historian for every configured tag.
	Y	Collector Timestamp is logged to Historian for every configured tag.
	Z	Collector configurable default Timestamp is logged to Historian for every configured tag.

HISTALARMOPC

For	Viewers (e.g. CimView, DGR) Project
Purpose	To hold name of the CIMPLICITY user who will have access to the corresponding OPC Server for alarms.
Comment	HISTALMOPC must be configured on the server so remote viewers can read the information.
Value	CIMPLICITY user name.
Default Value	Administrator.

HISTALDATAOPC

For	Viewers (e.g. CimView, DGR) Project
Purpose	To identify the CIMPLICITY user who will have access to the corresponding OPC data Server.
Comment	HISTALDATAOPC must be configured on the server so remote viewers can read the information.
Value	CIMPLICITY user name
Default Value	Administrator.

HISTALMPASS

For	Viewers (e.g. CimView, DGR) Project
Purpose	To hold the password required for the specified Historian alarm server user.
Comment	HISTALMPASS must be configured on the server so remote viewers can read the information.
Value	Valid password for the specified Historian alarm server user.
Default Value	Empty

HISTALMSERVER

For	Viewers (e.g. CimView, DGR) Project
Purpose	To hold the name of the Historian alarm server.
Comments	• The HISTALMSERVER value is used by Viewers (CimView, DGR etc.,) to connect to Historian for Historical Data. • HISTALMSERVER must be configured on the server so remote viewers can read the information.
Value	Historian server name.
Default Value	Local Server name.

HISTALMUSER

For	Viewers (e.g. CimView, DGR) Project
Purpose	To hold the valid Historian alarm server user name.
Comment	HISTALMUSER must be configured on the server so remote viewers can read the information.
Value	Valid Historian alarm server user name.
Default Value	Empty.

HISTDATAPASS

For	Viewers (e.g. CimView, DGR) Project
Purpose	To hold the password required for the specified Historian data server user.
Comment	HISTDATAPASS must be configured on the server so remote viewers can read the information.
Value	Valid Historian data server user password.
Default Value	Empty.

HISTDATASERVER

For	Viewers (e.g. CimView, DGR) Project
Purpose	To hold the name of the Historian data server.
Comments	• The HISTDATASERVER value will be used by Viewers (e.g. CimView, DGR) to connect to the Historian for historical point data. • HISTDATASERVER must be configured on the server so remote viewers can read the information.
Value	Historian data server name.
Default Value	Local server name.

HISTDATAUSER

For	Viewers (e.g. CimView, DGR) Project
Purpose	To hold the valid Historian data server user name.
Comment	HISTDATAUSER must be configured on the server so remote viewers can read the information.
Value	Valid Historian data server user name.
Default Value	Empty.

HISTOVERWRITETAGS

For	Viewers (e.g. CimView, DGR) Project	
Purpose	To indicate whether or not tag descriptions that already exist in Historian should be overwritten.	
Comment	HISTOVERWRITETAGS must be configured on the server so remote viewers can read the information.	
Value	Checked	Overwrite tag descriptions that already exist in Historian.
	Clear	Do not overwrite tag description that already exists in Historian.
Default Value	Clear	

IPC Global Memory Use Parameters

For	CIMPLICITY Interprocess Communication (IPC) System
Purpose	To modify the parameters IPC uses to determine what is "reasonable" growth in the use of system memory. CIMPLICITY Interprocess Communication (IPC) will accommodate "reasonable" periods of temporary growth in the use of system memory, yet try to keep errant clients from causing IPC Memory growth to consume all resources.. There are two sets of global parameters. The first set is for On-Node message "ports"; the other set is for Off-Node communications through "sockets".

On-Node Message "Ports"	
Global Parameter	Specifies:
PORT_PERIOD_SECS	The number of seconds in a calculation period. The default is 5.
PORT_GROWTH_PERIODS	The number of periods to allow that message production exceeds message consumption. After this number of periods messages will be dropped. Defaults to 6.
PORT_NONE_CONSUMED	The number of periods to allow with no message consumption. Defaults to 3. When the queue of port overflows and exceeds the above parameters the following message will be logged: **Router to Application queue overflow on port** If you see the above message tuning the above parameters may help. Typically the problem occurs because data is being produced faster than it can be consumed and changing these parameters will only delay the problem not solve it. Typically the solution to the problem is to slow down production (e.g. reduce scan rates) or speed up production (e.g. faster computer, more RAM)
Off node Communications through "Sockets"	
Global Parameter	Specifies
SOCK_PERIOD_SECS	The number of seconds in a calculation period. Defaults to 15.
SOCK_GROWTH_PERIODS	The number of periods to allow that message production exceeds message consumption. After this number of periods messages will be dropped. Defaults to 6.
SOCK_NONE_CONSUMED	The number of periods to allow with no message consumption. Defaults to 3. When the queue of the "socket" overflows and exceeds the above parameters the following message will be logged: **Router dropped off node message to…** If you see the above message tuning the above parameters may help. Typically the problem occurs because data is being produced faster than it can be consumed and changing these parameters will only delay the problem not solve it. Typically the solution to the problem is to slow down production (e.g. reduce scan rates) or speed up production (e.g. faster network, faster computers, more RAM).

IPC_QUEUE_SIZE

For	Router System **CAUTION:** Do not modify this option unless instructed to by GE Intelligent Platforms support personnel.
Purpose	To specify the maximum number of messages buffered by the Router in its Router to application queue. The first time a queue overflows, it logs the following message to the Status Log: **Router to Application queue overflow on port <port_name>** Further queue overflows on the same port do not result in any more error messages. When the queue overflows, if the new message is not a response required message, it is dropped. If it is a response-required message, it is not dropped and the queue size can temporarily exceed the specified value.
Value	**Number** (of messages).
Default Value	50

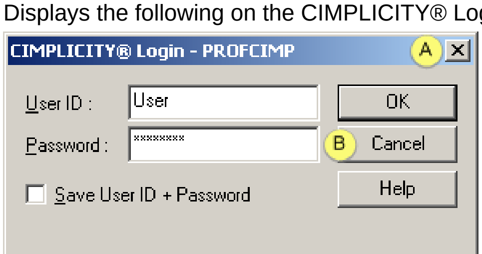
LICENSE_HT_DEBUG

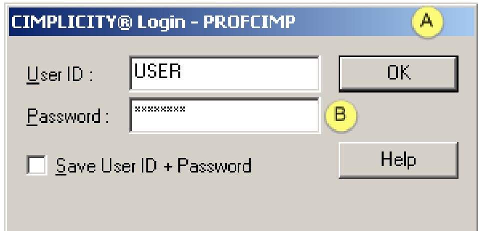
For	Router System	
Purpose	To log the current hyper-threading capability in addition to the number of physical and logical processors, which are logged at router startup.	
Value	Y	Logs the current hyper-threading capability at router startup.
	N	Does not log the current hyper-threading capability at router startup.
Default Value	N	

LOG_OPT

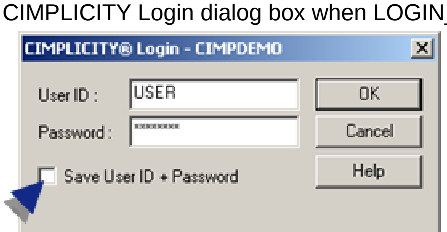
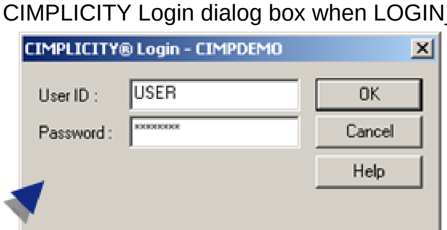
For	Database Logging Project	
Purpose	To define the default logging conditions for an alarm. You can choose any of the following:	
Value	Enter any combination of the following:	
	G	Alarm is generated.
	A	Alarm is acknowledged.
	R	Alarm is reset.
	D	Alarm is deleted by a user.
Default Value	No logging options	

LOGIN_CANCEL_OFF

For	CIMPLICITY Login Project		
Purpose	To prevent users from closing the CIMPLICITY® Login dialog box before logging in or cancelling the log in.		
Value	Select one of the following.		
	1	Displays the following on the CIMPLICITY® Login dialog box. 	
		A	X button enables a user to close the dialog box without logging in.
		B	Cancel enables a user to cancel the log in.

	0	(Or the global parameter does not exist.) Hides the following on the CIMPLICITY® Login dialog box. 	
		A	No X button prevents a user from closing the dialog box without logging in.
		B	No Cancel button prevents a user from cancelling the log in.
Default Value	0		

LOGIN_NOSAVE

For	CIMPLICITY Login Project
Purpose	To remove the Save Username and Password checkbox from a CIMPLICITY Login dialog box.
Comments	CIMPLICITY Login dialog box when LOGIN_NOSAVE is not added to the project.  CIMPLICITY Login dialog box when LOGIN_NOSAVE is added to the project.  **Important:** If you are using a Viewer LOGIN_NOSAVE must be set on the Viewer as a System (computer *(page 416)*) level global parameter.
Value	LOGIN_NOSAVE can be added and will function without a value.
Default Value	None

LOGIN_RETRY_PERIOD

For	CIMPLICITY Login Project. Do not modify this option unless instructed to by GE Intelligent Platforms support personnel.

Purpose	To specify the retry period in seconds for connecting to a project whose User Registration program is down.
Value	**Number** (of seconds).
Default Value	15

MARQ_POINT_LIMIT_LEN

Purpose	Defines the number of display characters for text points.
Value	A number between 20 and 80.
Default Value	80.

MARQ_PROC_NEW_ALARMS

Purpose	Determines how a new alarm will be displayed within the cycling marquee queues.	
Value	Either of the following.	
	YES	Display new alarm messages immediately.
	NO	Append new alarms to the end of the marquee queues.
Default Value	YES	

MARQ_RESERVED_NULL_CHAR

Purpose	Defines the ASCII character that will be reserved to specify the NULL character.
Value	ASCII character between 1 and 255.

Comment	Internally, the Marquee driver uses the NULL character to designate the end of marquee messages. You must specify the alternate character that the driver should use to terminate strings. After you specify the reserved NULL character, you cannot use it in the body of any of your messages. For example, if you specify ASCII character 255 as the NULL character, then ASCII character 255 will be used internally to

Default Value		

MARQ_VARIABLE_NULL_CHAR

Purpose	Enables the use of a variable or changing NULL termination character on a message by message basis.
Value	An ASCII character between 1 and 255.
Comment	The code is expects a sequence of characters `~$xx`, where `xx` is the hex representation of the NULL termination character zero filled (e.g. `0f`), on the beginning of a message. This new NULL termination character will be used until another message changes it.
Default Value	0

MARQ_WORD_WRAP_ON

Purpose	Enables or disables the word wrap feature.	
Value	Either of the following.	
	YES	To cause a character counter to count the characters in the message to be displayed. The character counter will insert a newline character when a word will be broken across lines of a marquee. Note that not all marquees support the word wrap feature.
	NO	Disable the word wrap feature.
Default Value	NO	

MARQ_WRAP_HF

Purpose	Enable or disable the word wrap feature for header and footer messages.	
Value	Either of the following.	
	YES	Include the characters within the header and footer messages for the word wrap feature.
	NO	Disable the word wrap feature for header and footer messages.
Default Value	NO	

MAX_ALARM_CLASSES

For	Alarm Classes System
Purpose	To specify the maximum number of alarm classes that can be created in a system.
Value	Number (of classes).
Default Value	550

You activate this parameter by creating a globals.ini file.

todo: To activate MAX_ALARM_CLASSES:

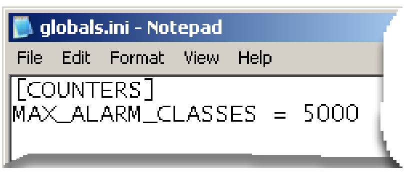
1. Open Microsoft Notepad (or another text editor).
2. Enter the following two lines.

   `[COUNTERS]`

   `MAX_ALARM_CLASSES =`<number>

   Where

   <number>= The maximum number of alarm classes allowed for the CIMPLICITY computer.

3. Save the file as Globals.ini.
4. Place Globals.ini in the ...\Proficy\Proficy CIMPLICITY\Data directory.
5. Restart CIMPLICITY for the new maximum number of alarm classes to take effect.

   Warning:

   - Do not change the globals.ini value once you have any CIMPLICITY process running. Changing the globals.ini once any CIMPLICITY process is running could cause memory corruption, or an access violation.
   - It is recommended that you reboot after changing the value in the globals.ini, since it is used to set up the shared memory file for the performance counters used in all the projects on a computer.

MAX_TREND_BUF

Be aware that increasing this parameter will affect system performance.

For	Trending Project
Purpose	To specify the maximum number of point values a point buffers for Trending. Point buffering is used when a chart is first displayed and there is some initial data. **Note:** If a value is entered in the **Max samples** field on the General tab in the Point Properties dialog box, the lower number (or MAX_TREND_BUF) determines the ceiling of buffered values.
Value	**Number** (of point values).
Default Value	200

MULTICAST_HOSTNAME

For	IP Address Traffic System
Purpose	To change the name used in the DNS lookup. MULTICAST_HOSTNAME takes the host name of the multicast host machine as input and returns its IP address.
Value	IP address. **Note:** If the multicast host machine does not have an IP address, the value of the MULTICAST_IP_ADDR global parameter is returned.
Default Value	`CIMPMULTIP`

MULTICAST_IP_ADDR

For	IP Address Traffic System
Purpose	To select different multicast addresses.
Value	**IP address** Guidelines • IP multicast addresses, also known as group addresses, are in the class D range of **224.0.0.0** to **239.255.255.255** as defined by setting the first four high order bits to 1110. In network prefix or Classless Inter-Domain Routing (**CIDR)** notation, IP multicast addresses are summarized as **224.0.0.0/4**. Multicast addresses in the range **224.0.0.0** to **224.0.0.255** (**224.0.0.0/24**) are reserved for the local subnet and are not forwarded by IP routers regardless of the Time to Live (TTL) in the IP header. • The IP multicast addresses from: • 224.0.1.0 to 238.255.255.255 are either reserved or assigned to a multicasting application. • 239.0.0.0 to 239.255.255.255 (239.0.0.0/8) are reserved for applications that can be administratively scoped. • The following are examples of reserved IP multicast addresses: • 224.0.0.1–all hosts on this subnet. • 224.0.0.2–all routers on this subnet. • 224.0.0.5–Open Shortest Path First (OSPF) Version 2, designed to reach all OSPF routers on a network. • 224.0.0.6–OSPF Version 2, designed to reach all OSPF designated routers on a network. • 224.0.0.9–Routing Information Protocol (RIP) Version 2. • 224.0.1.1–Network Time Protocol. For the latest list of reserved multicast addresses, visit the Information Sciences Institute web site.
Default Value	224.0.0.29

CAUTION: If you change the address, make sure that you change it on all the machines, in order to insure that they will continue "talking" with each other.

MULTICAST_TTL

For	IP Address Traffic System
Purpose	To specify the time a multicast message lives. This can be used to control how many routers will forward the message.
Value	Number (of routers). **Note:** Every time a message goes through a router one unit is deducted from the value. When the value reaches 0, the data transmission stops.
Default Value	10

OPCAE_TRACE_FLAGS

For	A&E OPC Client Project		
Purpose	To control tracing for an A&E OPC client.		
Value	Bit mask values are as follows.		
	1	Basic	Errors/warnings only.
	2	Notifications	From the OPC server.
	4	Actions	User ACKs and processing of the Event Queue so it will also record OPC events from the server.
	7	Most comprehensive trace	Includes 1 + 2 + 4.
Default Value	No tracing.		

PB_DIAGS

For	Point Bridge Project	
Purpose	To enable or disable Point Bridge diagnostics.	
Value	Numeric values in the following format: LEVEL=<num1>,FILECLOSE=<num2>,DBGCHK_SECS=<num3>,DUMPPOINT where:	
	<num1>	controls the amount/detail of diagnostic output. The value ranges from 0 to 9, where 0 means no debugging output, and 9 means the maximum amount of debugging output.
	<num2>	is a value for NT operating systems that may handle output files in a manner that prohibits multiple process access (as sometimes occurs with NT on DEC Alpha platforms). This value controls approximately how often the output file will actually be closed/re-opened, in seconds. A value of 0 (zero) means that this period closing will not occur. Any non-zero value of less than 60 will be forced to 60.
	<num3>	provides a number of seconds for an internal timer that rechecks the parameters in the Global Parameters file. This value cannot be less than zero. The default value is equivalent to five minutes.
	DUMPPOINT	initiates a one-shot dump of the Point Bridge internal information regarding the state of points it is servicing. This parameter is only acted upon once when any of the other arguments are changed, and the resulting change leaves LEVEL with a non-zero value.

Default Value		

PCM_ENH_AUDIT

For	Change Management System	
Purpose	To enable or disable enhanced auditing	
Comment	PCM_ENH_AUDIT: • Is added automatically if you check Enable enhanced auditing on the Project Properties dialog box>Change Management tab. • Can be added manually.	
Value	Enter one of the following:	
	Y	Enables enhanced auditing
	N	Disables enhanced auditing
Default Value	One of the following.	
	Doesn't exist	Enable enhanced auditing has never been checked on the Project Properties dialog box>Change Management tab.
	Y	Enable enhanced auditing is checked on the Project Properties dialog box>Change Management tab.

PERF_COUNTERS_ENABLED

For	InterProcess Communications System	
Purpose	To enable or disable performance counters.	
	PERF_COUNTERS_ENABLED should be defined on nodes that lack support for performance counters, such as XP embedded, which may not have support installed for performance counters.	
Value	Enter one of the following:	
	Y	Enables performance counters.
	N	Disables performance counters.
Default Value	Y	

PPS_OPC_SCANRATE

For	Proficy Process Systems Project	
Purpose	To set the scan rate for all configured PPS OPC Group settings in the project.	
Value	Scan rate time in microseconds (ms).	
Default Value	0	ms

<PORT>_CACHE_DIAGNOSTICS

For	Device Communications Project	
Purpose	To enable or disable diagnostics cache on a single port.	
Comments	DC_CACHE_DIAGNOSTICS *(page 447)* is evaluated first and then <PORT>_CACHE_DIAGNOSTICS. A message is logged to the OUT file if the diagnostics are disabled.	
Value	Enter one of the following:	
	N or n	Disables diagnostics.
	Y or y	Enables diagnostics
Default Value	N	

<PORT>_DEVICE_TIMESTAMP_UTC

For	Time stamp Project Port
Purpose	To change the default local time stamp to a UTC time base for all unsolicited messages coming to a port. that have the time stamp embedded. This applies to the following communications devices.

		Modbus TCP/IP
		S90 Triplex
Value	Y	UTC time base is used.
	N	Local time stamp is used.
Default Value	N	

<PORT>_OVRD_SCAN

For	Scan Rate Project		
Purpose	To define the scan rate scale and unit for a selected port.		
Comment	The port level has the highest precedence for the base scan rate.		
Value	Point ID	Of a configured INT array point that has 2 elements	
	Elements are:		
	Element_1	Scale	Between 1 and 1000 inclusively.
	Element_2	Units	Between 1 and 4
		1	TICKS (1/100 seconds)
		2	Seconds
		3	Minutes
		4	Hours
	Note: The scale and units and be changed during runtime.		
Default	No point ID is identified.		

<PORT>_SYNC_ONLY

For	Device Communications that support asynchronous polling by default, including:	
		Triplex
		AB Ethernet
		Mitsubishi TCP/IP
		Sharp
Purpose	To disable asynchronous polling for all devices on a port if poll after setis required. **Note:** Check Poll after seton the Device tab in the Point Properties dialog box to enable the functionality.	
Comments	Replace <PORT> with the actual port name. Example S90TRI1_SYNC_ONLY	
Value	Y	Polling for all devices on the port is synchronous only.
	N	Asynchronous polling for all devices on the port, where the feature is supported by the device communication interface, is enabled.

Default Value	N	

Note: Where it is available, asynchronous polling typically provides better data collection performance than does synchronous polling.

<PORT>_USE_OVRD_SCAN

For	Scan Rate Project	
Purpose	To enable the override of the scan rate for a selected port.	
Value	Point ID	• Of a configured Boolean virtual point. • Set to 1 enables the base scan rate override.
	Note: The point ID can be set to 1 during runtime.	
Default Value	No point ID is identified.	

PROJECT_ID

For	Multiple Projects Option Project **CAUTION:** Do not modify this option unless instructed to by GE Intelligent Platforms support personnel.
Purpose	To specify the one character **Project ID** for the project.
Value	Enter a letter A through Z, or number 0 through 9.
Default Value	None

PRT_AUTOMOVE_BY_LOCATION

For	PRT Project
Purpose	To include an additional `STRING_80 array` point named <existing auto move point name>.LOC or <existing auto move point name>_ LOC for each region with an auto move point based on the parameter value. The array point provides a single update when the entire move operation has finished. **Note:** The array point must be of type STRING_80. Any other point type will result in undesirable results.
Value	Values are as follows.

	1	Populates the `STRING_80 array` point <existing auto move point name>.LOC with point IDs. **Important:** The `STRING_80 array` point <existing auto move point name>.LOC must be configured by the user ahead of time or a warning message will be logged on project startup.
	2	Populates the `STRING_80 array` point <existing auto move point name>_LOC with point IDs. **Important:** The `STRING_80 array` point <existing auto move point name>_LOC must be configured by the user ahead of time or a warning message will be logged on project startup.
	Any other value or Not created	Does not incorporate the `STRING_80 array` point, if it exists.
Default Value	Not created	

Note:

- Values that were in the regular auto-move point sequence, will be in the same sequence in the `STRING_80 array` point's array elements.
- The number of elements in the new location based auto-move point will match the number of configured items per location for that region.
- A new location based point will be updated with the ID of each item moved in its own array element in the same order that they transitioned as a single update.
- If no auto-move point updates occur as a result of a move operation, the new point will not be updated.
- The configuration of the array point name with _LOC is in line with the usage of classes, which do not allow the use of "." in point names.

PRT_BASIC_USE_EX

For	PRT Project	
Purpose	To change the way the CIMPLICITY behaves with regard to getting a list from a project. If the user wants ALL PRT Basic scripts used the new EXtended PRT_APIs they can define a Project level Global Parameter.	
Value	Enter one of the following.	
	0	Use old PRT API methods.
	1	Use new Extended methods and allow across project lists properly.
	2	Use new Extended methods.
Default Value	0	

<table>
<tr><td>Comments</td><td>In addition to this global parameter you can specify these parameters on a script basis. This is done through the following methods:
<pre>
' Use old PRT_API methods
PrtSetProjectSearch PRT_PROJECT_SEARCH_OLD
' Use new Extended methods with search restricted in local project
PrtSetProjectSearch PRT_PROJECT_SEARCH_CURRENT
' Use new Extended methods with search in all connected projects
PrtSetProjectSearch PRT_PROJECT_SEARCH_GLOBAL
This change will affect all of the following calls:
Affects:
PRT.GetItemList
PRT.GetRegionList
PRT.GetGroupList
PRT.GetServiceList
PRTService.GetItemList
PRTService.GetRegionList
PRTService.GetGroupList
PRTGroup.GetItemList
PRTGroup.GetRegionList
PRTRegion.GetItemList
</pre></td></tr>
<tr><td>Example</td><td>This example script assumes that GRP1 exists in both projects.</td></tr>
<tr><td></td><td><pre>
Sub Main()
Dim main As New Prtgroup
' Use new Extended methods with search restricted in local project
PrtSetProjectSearch PRT_PROJECT_SEARCH_CURRENT
' Associated with appropriate project
main.projectid = "TRACKER1"
main.Id = "GRP1"
' Get a list of items from group "MAIN"
'   and display them one by one in a message box.
main.GetItemList
for j = 0 to main.ItemCount - 1
      ' return all items from TRACKER1 in GRP1
      MsgBox main.Item(j).ItemId
next j
' Associate with different project
main.projectid = "TRACKER2"
main.Id = "GRP1"
' Get a list of items from group "MAIN"
'   and display them one by one in a message box.
main.GetItemList
for j = 0 to main.ItemCount - 1
      ' return all items from TRACKER 2 in GRP1
      MsgBox main.Item(j).ItemId
next j
End Sub
</pre></td></tr>
<tr><td></td><td>In this example:

If you change "PrtSetProjectSearch PRT_PROJECT_SEARCH_OLD" then the:
Second association of putting the main.projectid to "TRACKER2" will have no impact.
Subsequent GetItemlist would be against the original project "TRACKER1".
The same can be said for changing to "PrtSetProjectSearch PRT_PROJECT_SEARCH_GLOBAL" except it will use the new Extended API methods.
</td></tr>
</table>

PRT_GUID_DISABLE_REFID

<table>
<tr><td>For</td><td colspan="2">PRT Project</td></tr>
<tr><td>Purpose</td><td colspan="2">To stop PRT from generating reference ID's.</td></tr>
<tr><td>Value</td><td>1</td><td>PRT will not generate reference ID's.</td></tr>
</table>

	0	PRT will generate reference ID's.
Default Value	None	PRT will generate reference ID's.

PRT_TADB_COMM_TOUT

For	PRT/TADB Project
Purpose	Number of seconds PRT can take for a command to be successful in the TADB before it times out. Time out will trigger an alarm to alert the user that the command failed.
Value	Number (of seconds to wait before command time out)
Default Value	60

PRT_TADB_CONN_TOUT

For	PRT/TADB Project
Purpose	Number of seconds PRT can take to make a connection with the TADB before it times out. Time out will trigger an alarm to alert the user that the connection failed.
Value	Number (of seconds to wait before connection time out)
Default Value	60

PRTC_TADB_SYNCHRONIZE

For	PRT/TADB Project	
Purpose	Controls whether or not **prtc.exe** will perform synchronization of PRT and TADB during startup.	
Value	Enter one of the following:	
	Y	Will perform synchronization during startup.
	N	Will not perform synchronization during startup.
Default Value	N	

PRTC_TADB_VALIDATION

For	PRT/TADB Project	
Purpose	Controls whether or not the PRT Collector will generate the XML report	
Value	Enter one of the following:	
	0	Do not generate a report.
	1	Generate a report before synchronization.
	2	Generate a report before and after synchronization.

	3	Generate a report after synchronization.
Default Value	0	

PRTCNT_USE_TADB

For	Tracker Project
Purpose	To enable PRTCNT to ignore the TADB option even if it is enabled.
Value	Enter one of the following: Y PRTCNT will use TADB if the project has TADB Option enabled. N PRTCNT will assume that the project does not have TADB Option enabled.

PTDL_ENABLE_MEASUREMENTS

For	Measurement System (Data logging) Project	
Purpose	To specify whether point values are to be logged in the currently active measurement system.	
Default For	**Enable measurement** systems field on the Parameters tab of the Logging Properties dialog box.	
Value	Enter one of the following:	
	0	Don't use a measurement system.
	1	Use the currently active measurement system.
Default Value	0	

PTDL_QUANTIZATION

For	Data Logging Project
Purpose	To specify the rate in ticks at which the Point Data Logger submits data for logging.
Default for:	**Point data logging scan rate** field in the Database Logger Logging Properties dialog box.
Value	Number (of ticks) that is: • Faster than the fastest table scan rate in ticks at which the Point Data Logger will submit data for logging. • Large enough so that all the data for a PLC scan comes into the database together. (100 ticks=1 second)
Default Value	100

PTEXP_ANA_EQ_NACK_AND_AL

For	Application expressions Project
Purpose	`ANA` behavior changed in CIMPLICITY V.6.1
	`PTEXP_ANA_EQ_NACK_AND_AL` can control which `ANA` behavior is used.
Value	**n**, **N**, **f** or **F** then ANA displayed TRUE if an alarm is not acknowledged. **y**, **Y**, **t** or **T** then ANA displays TRUE only if a point is in alarm state and has not been acknowledged.
Default Value	for new Projects (V9.5 and higher) **y**, **Y**, **t** or **T** for older Projects (prior to V9.5) **n**, **N**, **f** or **F** Note: ANA displayed TRUE if an alarm is not acknowledged i.e same as NACK, irrespective of PTEXP_ANA_EQ_NACK_AND

PTM_AM_DELAY_VAL_UPDATE

For	Point Management
Purpose	If point alarms are configured to update the point value in the message the updates will be delayed by this amount
Value	Number (of seconds)
Default Value	0

PTM_TIMESTAMP_FMT

For	Points Project	
Purpose	To define the time format for timestamps that are converted to strings.	
Value	`m/d/yy  HHHH:MM:SS.TTTTTT A` Where	
	m	month
	d	day
	yy	year
	HHHH	hour
	MM	minute
	SS	seconds
	TTTTTT	milliseconds
	A	microseconds
Default Value	`m/d/yy  HHHH:MM:SS.TTT A`	
Examples	m/d/yy HHHH:MM:SS.TTTTTT A m/d/yy HHHH:MM:SS A	

PTMAP_TIMED_POINTS

For	Points Project
Purpose	Allow the user to have off node ON_CHANGE requests updated only after a specified number of seconds.
Value	Number (of seconds)
Default Value	No entry

PTMDP_BATCH_UPDATING

For	Derived Points Project	
Purpose	To control updating of derived points with batch processing or a memory-efficient approach.	
Value	Enter one of the following.	
	Y	Batch processing
	N	Memory-efficient approach.
	Default	Y

PTMDP_DO_EU_CONV

For	Point Configuration Project	
Purpose	To automatically convert Virtual Point ID values that you use in the Expression Editor to engineering units without having to use the EU_CONV(<point_id>) function.	
Value	Enter one of the following:	
	0	Do not automatically convert **Point ID** values to engineering units in expressions.
	1	Automatically convert **Point ID** values to engineering units in expressions.
Default Value	0	

PTMDP_DO_SAVEPOINT_CACHE

For	Point Configuration System and Project	
Purpose	To define when virtual point values will be stored.	
Default For	**Store values** radio buttons in the Point Setup dialog box.	
Value	Enter one of the following:	
	0	Point values will be stored on point update.
	1	Point values will be stored on project shutdown

Default Value	1	

PTMDP_DO_SAVEPOINT_COMPACT

For	Point Configuration Project	
Purpose	To define whether virtual point storage is to be compacted on project startup.	
Default For	Compact on project startup check box in the Point Setup dialog box.	
Value	Enter one of the following:	
	0	Do not compact the point storage on project startup
	1	Compact the storage on project startup
Default Value	0	

PTMRP_ALARM_DELAY_STATE

For
Point Management

System

Purpose To change alarm state immediately to Point Management Clients.

Value Values are:

Y The alarm state changes immediately, and alarm delay is not applicable.

N The alarm state delay is enabled.

PTMRP_EXTERNAL_ALARM_OVERRIDE

For	Enterprise Project System	
Purpose	To enable external point alarm states to be reset (e.g. Point Bridge or Enterprise points).	
Value	Values are:	
	1	Enables `PTMRP_EXTERNAL_ALARM_OVERRIDE`.
	Not 1	Does not enable `PTMRP_EXTERNAL_ALARM_OVERRIDE`. If a Source project point was in an alarm state and was reset, the external point will not be reset; it will continue to display in an alarm state.
	Default	Not 1

PTMRP_FORCE_PT_READ_MAN_MODE_OFF

For
Enterprise Project

System

Purpose	Enables you to handle the way point values are read from device when the manual mode is disabled.
Value	Values are: Y When you disable the manual mode, the point values of unsolicited points are read from device irrespective of the value change in PLC. N When you disable the manual mode, the point values of unsolicited points are read from device only when the point value changes in PLC.
Default	N

PTMDP_MAX_RESPONSES_PER_CALLBACK

For	Point Management
Purpose	To control the number of point updates the Derived Point Process will process at a given time. The Derived Point Process will process up to this number of point updates before performing calculations on configured derived points.
Value	Number (of updates)
Default Value	100

PTMDP_PROC_ALL_DYNCFG

For	Point Configuration Project	
Purpose	To ensure that when class objects are created in dynamic mode, all expressions will be updated,	
Comment	This global parameter is available for the rare instances when it is found that some calculated expressions are not evaluated and updated.	
Value	Enter the following.	
	Y	Ensures that all expressions will be updated.
	N	Does not ensure that all expressions will be updated.
Default Value	N	

PTMRP

For	Point Management Project **CAUTION:** Do not modify this option unless instructed to by GE Intelligent Platforms support personnel.
Purpose	To define the default Point Manager to be used for point processing.
Value	**Name** (of the default Point Manager).
Default Value	`MASTER_PTM0_RP`

PTMRP_DELAY_ALARM_STATE

For	Point Management Project
Purpose	To delay the alarm states that the Point Manager sends to all the applications with point value updates using the PTMAP API. **Note:** The state update can be seen in: • CimView using the AL, A1, A2, AL1, AL2, AH1 and AH2 functions. • The Point Control Panel with the coloration of the text line that is used to display the point and its data.
Comment	Setting `PTMRP_DELAY_ALARM_STATE` is not required for the Alarm Manager, for both point and event alarms. The Point Manager always delays notification of alarms to the Alarm Manager regardless of this parameter setting. The Alarm Manager then sends alarms to both the alarm client API's and the Alarm Viewer.
Value	Enter one of the following: Y Do not enable the alarm state delay; the alarm state changes immediately N (or does not exis.) Enable the alarm state delay.
Default Value	Parameter does not exist.

PTMRP_EXTERNAL_ALARM_OVERRIDE

For	Enterprise Project System	
Purpose	To enable external point alarm states to be reset (e.g. Point Bridge or Enterprise points).	
Value	Values are:	
	1	Enables `PTMRP_EXTERNAL_ALARM_OVERRIDE`.
	Not 1	Does not enable `PTMRP_EXTERNAL_ALARM_OVERRIDE`. If a Source project point was in an alarm state and was reset, the external point will not be reset; it will continue to display in an alarm state.
	Default	Not 1

PTX_MAX_CACHED_POINTS

For	Point Translation Project
Purpose	To limit the size of the Point Translation cache on systems that have a large number of points. Point Translation adds points to the cache when Point Management applications (for example, scripts, CimView windows, Point Control Panel, etc.) requests point information. **CAUTION:** If the applications on a project are likely to cycle through all the points, limiting the cache size may cause point information to update more slowly than expected.
Value	Enter 0 or delete the global parameter if you do not want to limit the cache size. Otherwise, enter the number of points to be put into the cache.

Default Value	0

PTX_MUTE_DC_POINT_CHANGES

For	Point Project	
Purpose	To stop sending point configuration change requests to the device communications, thus allowing them to operate in an efficient manner while dynamic configuration is occurring.	
Value	Select one of the following.	
	Y	Stops configuration change notifications.
	N	Allows configuration change notifications.
Default Value	N	

PW_BLOCK_SIZE

PW_BLOCK_SIZE defines the IP address block size that the Project Wizard uses to determine how many IP address requests to make at one time. This is particularly useful in Windows XP SP2 and Windows 2003 Server SP1, whose security enhancements limit the number of outgoing unanswered TCP/IP requests. The security limitation can prevent the Project Wizard from finding existing PLC's in a network.If the Project Wizard does not find any PLC's that you know are on your network do the following.

1. Open the Windows Administrative Tools>Event Viewer
2. Review the Event Viewer to see if there is a TCP/IP error.
   If there is a TCP/IP error the Windows security enhancements are interfering with the Project Wizard's functionality.
3. Add PW_BLOCK_SIZE to the project.
4. Assign a value from 1 through 64.
   Recommended: Assign a value that is less than 5. The slower the network, the lower the number must be.

PW_BLOCK_TIMEOUT

For	Project Wizard Project
Purpose	To adjust the time out value for each block of IP's that the Project Wizard scans. **Note:** Adjusting the time out can help configure slower networks.
Value	The number of seconds as an unsigned integer within the valid range of 0 - 4,294,967. **Note:** If a negative number is entered, it will be converted to it's corresponding unsigned number.

Default Value	2 seconds

QT_ENABLE_SQL_WINAUTHEN

For	Windows Authentication Project	
Purpose	To support Windows Authentication with Quick Trends.	
Value	Enter one of the following.	
	1 or Y	Supports Windows Authentication with Quick Trends.
	N or NA	Does not support Windows Authentication with Quick Trends.
Default Value	NA	

QUERY_UTC_ONLY

Note: QUERY_UTC_ONLY should be used only for projects that were created for CIMPLICITY v8.0 or higher; for projects that were created in CIMPLICITY versions lower than v8.0, old data will contain NULLs in the **timestamp_utc** column; in most cases set the value to N.

When using the Historical Alarm Viewer on Viewers that have CIMPLICITY versions lower than v8.0 set the `QUERY_UTC_ONLY` global parameter value to X, as follows.

1. On the Server
2. For a CIMPLICITY v6.2 project
3. At the project level.
4. Before the project is started.

RAW_LIMIT_ALARM

For	Point Alarms Project	
Purpose	To enable or disable the generation of alarms for point values that are outside their raw limits. The options are:	
Value	Enter one of the following:	
	YES	Raw limit alarms are enabled.
	NO	Raw limit alarms are disabled.
Default Value	YES	

RCO_IGNORE_INIT_PT_UDP

For	RCO Project

Purpose	To set the RCO trigger functionality to what it was in CIMPLICITY versions that were lower than CIMPLICITY 6.2.	
Comments	The initial point update in:	
	6.2 and higher	Is used to prime the initial value of TRANS-HIGH and TRANS-LOW points.
	Less than 6.2	Is read and thrown away.
	If a point is unavailable, then the initial point will not come until the point becomes available. Therefore in:	
	6.2 and higher	RCO site will ignore the point values that are set before the RCO site started.
	Less than 6.2	RCO does not ignore the first update; RCO uses the first unavailable to available point as a trigger.
Value	0	Ignore the initial point value before RCO started.
	1	Do not ignore the initial point value before RCO started. (Use the less than 6.2 behavior).
Default Value	0	

RCODB_CONN_TOUT

For	RCO Project
Purpose	To specify the wait time during project startup for RCO to connect with the database. Time out will occur at the end of the specified time.
Comments	If it takes longer than the default 15 seconds to connect to the database. an error message will display that the data source is invalid. Assigning RCODB_CONN_TOUT a longer connection time overcomes this issue.
Value	Number of seconds before RCO connection to the database times out.
Default Value	15 seconds

RCODB_QRY_TOUT

For	RCO Project
Purpose	To specify the wait time for RCO to get a query result.
Comments	With SQL Server 2005, it might take longer than the default 15 seconds to query the database. If it does take longer, an error message will display that the database is invalid. Assigning RCODB_QRY_TOUT a longer connection time overcomes this issue.
Value	Number of seconds for RCO to get a query result from the database, if getting a query result takes longer than the default 15 seconds.

Default Value	15 seconds

REDUND_LINK_SLEEP

For	Server Redundancy System
Purpose	To make the Router wait a period of time before creating the link to the standby node.
Comments	The User Registration (UR) processes on the primary and secondary nodes need to synchronize with each other at startup. This can normally occur within a 30-second period. On slower computers this might not be enough time. `REDUND_LINK_SLEEP` provides the ability to make the appropriate adjustments.
Value	**Number** (of seconds).
Default Value	5

REDUND_PROBE_COUNT

For	Server Redundancy Project (On both primary and secondary computer)	
Purpose	To specify the number of missed probes (within a failover period) before an application is failed over. **Warning:** The failover rate should never be modified to less than 3 seconds.	
Comments	The failover period is defined as: `REDUND_PROBE_DELAY (page 493) * (REDUND_PROBE_COUNT + 1)` If more than the `REDUND_PROBE_COUNT` probes are missed, a failover is started.	
Value	Number (of probes)	
Default Value	Count	5
	Failover period	15 seconds

REDUND_PROBE_DELAY

For	Server Redundancy Project (On both primary and secondary computer)	
Purpose	To specify the number of milliseconds between probes to contact an application. **Warning:** The fail over rate should never be modified to less than 3 seconds.	
Comments	The fail over period is defined as: `REDUND_PROBE_DELAY * (REDUND_PROBE_COUNT (page 493) + 1)` If more than the `REDUND_PROBE_COUNT` probes are missed, a failover is started.	
Value	Number of milliseconds	
Default Value	Delay	3000 milliseconds
	Failover period	15 seconds

REDUND_PROBE_PORT

For	Server Redundancy Project (On both primary and secondary computer)
Purpose	To specify the TCP/IP port number used to implement the server redundancy probe mechanism.
Comments	Change this parameter only if it conflicts with other software. The number of missed probes before a failover can be specified using the `REDUND_PROBE_COUNT` *(page 493)* and `REDUND_PROBE_DELAY` *(page 493)* global parameters.
Value	Port number
Default Value	4000

REPEAT_TOUT

For	Alarm Management Project
Purpose	To specify the time in minutes before an alarm is automatically repeated to all interested processes by the Alarm Management Resident Process.
Default for	**Minutes** field in the Alarm Options tab of the Alarm Definition dialog box when **Auto repeat** is set to **Timed**.
Value	**Number** (of minutes)
Default Value	0 (alarms are not repeated).

RTR_ACCEPT_CONN

For	Network Options System	
Purpose	To specify whether or not you want to connect to other projects in your enterprise and you want other projects to connect to this project.	
Default for	Use the **Accept Connections** field in the CIMPLICITY® Options dialog box.	
Value	Enter one of the following:	
	Y	Accept the connections.
	N	Stand alone.
Default Value	Y	

RTR_DISABLE_BCAST

For	Router System

Purpose	To specify if you want to disable broadcast of the project name on the network. When you broadcast the project name, users on other nodes that request point data can use the project name in fully qualified points. Otherwise, they can only use the node name in fully qualified points. Your specification is used as the default in the **Enable Project Broadcast** field in the Workbench Project Properties dialog box.	
Value	Enter one of the following:	
	Y	Disable broadcast.
	N	Do not disable broadcast.
Default Value	N	

RTR_MAX_OUTMESSAGE_COUNT

For	Router System **CAUTION:** Do not modify this option unless instructed to by GE Intelligent Platforms support personnel.
Purpose	To specify the maximum number of messages buffered by the Router for a Router to Router link. If the queue overflows, the Router drops new messages. When it drops the first message, it logs the following error to the Status Log: **Router dropped offnode message to <remote_node_name>** Further dropping of messages to the same node do not result in additional error messages as long as the remote node stays active.
Value	(maximum) Number (of messages).
Default Value	1000

SECURE_SOCKETS

For	Network System
Purpose	To secure a PC with encryption. When you activate SECURE SOCKETS encryption on one PC, then only PCs that support this parameter (PCs that have CIMPLICITY 4.01-Service Pack 2 or higher installed) will be able to communicate with the secure PC. All data transfer to the secure PC will be encrypted.

You activate this parameter by creating a globals.ini file.

todo: To activate SECURE_SOCKETS:

1. Open Microsoft Notepad (or another text editor).
2. Enter the following two lines.

   `[ROUTER]`

   `SECURE_SOCKETS=Y`

   Where

Y= Yes, activate encryption

N= No, de-activate encryption

3. Save the file as Globals.ini.

4. Place Globals.ini in the ...\Proficy\Proficy CIMPLICITY\Data directory.

5. Restart CIMPLICITY for encryption to take effect.

Note: Only computers that have CIMPLICITY 4.01 service pack 2 or higher installed will be able to communicate with a SECURE_SOCKET encrypted PC.

SERVER_UP_INTERVAL

For	Router System and Project **CAUTION:** Do not modify this option unless instructed to by GE Intelligent Platforms support personnel.
Purpose	To specify the Router (IPC) System Name Broadcast Period in seconds.
Value	Number (of seconds).
Default Value	75

SETPOINT_SECURITY

For	Point Management Security Project	
Purpose	To specify if a user can perform setpoints on only those points whose resources are in the user's view. This is the default for the Use the Enable Setpoint Security check box in the Project Properties box.	
Value	Enter one of the following:	
	YES	Enable setpoint security.
	NO	Do not enable setpoint security.
Default Value	NO	

SHORT_FILENAMES

For	Filenames Project **CAUTION:** Do not modify this option unless instructed to by GE Intelligent Platforms support personnel.	
Purpose	To enable or disable support for short (8 characters or less) filenames.	
Value	Enter one of the following:	
	YES	Users cannot use filenames greater than 8 characters.
	NO	Users can use any length filenames.
Default Value	NO	

SECONDARY_STARTUP_TIMEOUT

For	Server Redundancy Project
Purpose	To delay starting the project on the secondary server until after the project on the primary server starts if the project on both the primary (active) and secondary (standby) servers is configured to start at boot.
Comments	`SECONDARY_STARTUP_TIMEOUT` helps avoid race conditions between the two servers when they are trying to determine which server is the active time out server.
Value	Number (of minutes).
Default Value	0

SOLVEENGINEDEBUG

For	Order Execution Mgt Query Engine System and Project	
Purpose	To display debug information about a query failure in a HMI\log\SolveInterface.out file.	
Value	Enter one of the following:	
	0	A query message will report that a query has failed. Debug information will be discarded.
	1	A query message will report that a query has failed. Debug information will be displayed in HMI\log \SolveInterface.out
Default Value	0	

SPC_DB_CONNECT_TIMEOUT

For	CIMPLICITY Statistical Process Control Project
Purpose	To reduce the number of milliseconds CimView will wait to connect to an SPC database (for an inserted SPC ActiveX control) before the wait times out.

Comment	SQL Express 2005 and higher require a user name and password to connect to the database. If SPC has not been configured with the correct user name and password and one or a group of CimView screens includes an SPC Control CimView will appear to be hanging for the default time (60 seconds) while trying to connect. Do either of the following. • Enter the correct credentials in the SPC Document Properties dialog box. • Use SPC_DB_CONNECT_TIMEOUT to reduce the wait time.	
Value	Number (of milliseconds)	
	Minimum	250 ms
	Maximum	60000 ms
Default Value	60000	

SPC_RESIZE_OUT_OF_BOUNDS

For	CIMPLICITY Statistical Process Control Project	
Purpose	To re-scale and re-center the graph when extremely out of control points are plotted so the values won't be drawn outside the bounds of the graph.	
Value	Enter one of the following.	
	Y or y	Enables the parameter
	N or n	Disables the parameter
Default Value	Y	

STARTUP_TIMEOUT

For	Project Project
Purpose	To set the number of minutes to wait before timing out when starting up a project.
Value	**Number** (of minutes).
Default Value	10

SVC_RETRY_COUNT

For	External Services Project **CAUTION:** Do not modify this option unless instructed to by GE Intelligent Platforms support personnel.
Purpose	To set the number of retries waiting for an external service to start up.
Value	**Number** (of retries).

Default Value	30

SVC_RETRY_DELAY

For	External Services Project **CAUTION:** Do not modify this option unless instructed to by GE Intelligent Platforms support personnel.
Purpose	To set the delay in ticks between retries waiting for an external service to start up.
Value	**Number** (of ticks). **Note:** 100 ticks=1 second
Default Value	100

SYSNAME

For	Project Project **CAUTION:** Do not modify this option unless instructed to by GE Intelligent Platforms support personnel.
Purpose	To specify the project name.
Value	Name (for the project).
Default Value	The name that was assigned to the project when it was created.

TERMSERV_ALLOW_SETPOINTS

<table>
<tr><td>For</td><td colspan="2">Points using remote connections or WebSpace sessions System or Project</td></tr>
<tr><td>Purpose</td><td colspan="2">To enable or disable setpoints from user-authenticated CIMPLICITY tools that are launched through remote desktop or WebSpace sessions.</td></tr>
<tr><td>Value</td><td colspan="2">Enter any of the following.</td></tr>
<tr><td></td><td>Any value except F, FALSE, N, or NO.</td><td>This will enable setpoints for remote desktop or WebSpace sessions.

Note: Setpoints will also be enabled if the TERMSERV_ALLOW_SETPOINTS parameter does not exist or contains an unexpected value.</td></tr>
<tr><td></td><td>• F
• FALSE
• N
• NO</td><td>This will disable setpoints for remote desktop or WebSpace sessions, but not for physical console sessions.

Note: The values specified are not case-sensitive.</td></tr>
<tr><td>Default Value</td><td colspan="2">T</td></tr>
</table>

TREND_DISABLE_READNESTED

For	Trending System / Project		
	TREND_DISABLE_READNESTED is assigned to the system or project, as follows.		
	Computer	Assign to	When
	Viewer	System	Always
	Server	System	CimView is started outside a project.
	Server	Project	CimView is started inside a project.
Purpose	To disable database sub-queries for time stamps.		
Value	Enter one of the following:		
	0	Enable time stamps.	
	1	Disable time stamps.	
Default Value	0		

TRK_ERROR_RETRIES

For	Tracker System	
Purpose	To specify how many times the Extended Attributes database access will be retried if the first attempt returns an error.	
Value	Number from 0-10 retries	
	0	Minimum retries (None)
	10	Maximum retries
Default Value	0	

TRK_PRJMON_INTERVAL

For	Tracker System	
Purpose	To specify how frequently, in seconds, the Tracker Collector checks to see if any projects have abnormally terminated.	
Value	Number from 3-60 seconds	
	3	Minimum seconds
	60	Maximum seconds
Default Value	15	

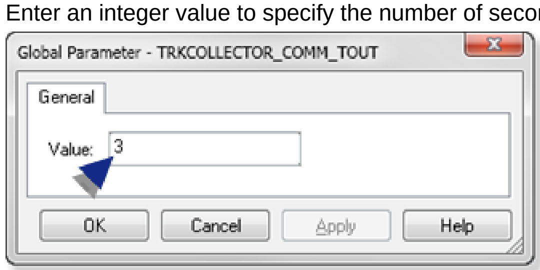
TRK_PRJMON_TIMEOUT

For	Tracker System	
Purpose	To specify how long a project has to check in with the Tracker Collector before the Collector declares that it has abnormally terminated.	
Value	Number from 2-30 seconds	
	2	Minimum seconds
	30	Maximum seconds
Default Value	10	

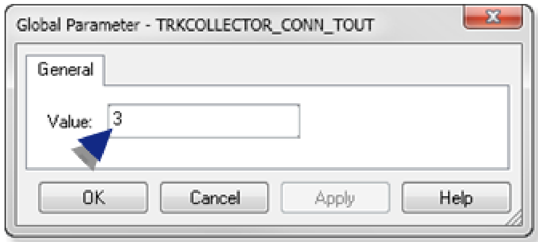
TRK_RETRY_DELAY

For	Tracker System	
Purpose	To specify the delay, in milliseconds, between retry attempts when access to the Extended Attribute database fails.	
Value	Number from 100-3000 milliseconds	
	100	Minimum milliseconds
	3000	Maximum milliseconds
Default Value	1000	

TRKCOLLECTOR_COMM_TOUT

For	Tracker System	
Purpose	To specify the number of seconds before TrkCollector gives up on a SQL command.	
Value	Enter an integer value to specify the number of seconds.  	
	***	No maximum
	1	Minimum
	0	No time out
Default Value	3	

TRKCOLLECTOR_CONN_TOUT

For	Tracker System	
Purpose	To specify the number of seconds TrkCollector can take to get a SQL connection.	
Value	Enter an integer value to specify the number of seconds. 	
	***	No maximum
	1	Minimum
	0	No time out
Default Value	3	

TRKCOLLECTOR_ITEM_CACHE

For	Tracker System	
Purpose	To Enable/Disable support for caching extended attributes in TrkCollector.	
Value	Y	Extended attributes will be cached by TrkCollector
	N	Caching of extended attributes by TrkCollector will be disabled.

TRUNCATE_OBJ_DESCRIPTION

For	Proficy Process Systems Project	
Purpose	Enable PPS Object Builder to create PPS objects by truncating user created data item descriptions to the 80 character limit, when the limit is exceeded.	
Value	Values are:	
	Y or y	Truncate user created descriptions that replace the data item default $DESCRIPTION, so the object can be built. **Note:** An error is logged to the status log reporting that the data-item description has been truncated to the allowed maximum length.
	N or no parameter	Does not create the class object if the user created description exceeds 80 characters.
	Default	N or no parameter

UR_LOGIN_FAILURES

For	User Interface Project
Purpose	To define the consecutive number of login failures before the user is locked out of the Login dialog box.
Value	Number greater than 1.
Default Value	3

USE_HIST_TIMEFMT

1. Whether or not to display milliseconds.
   Note: Milliseconds are truncated and display as .000.

2. What to display when a timestamp value is not available.

3. 24-hour format.

4. milliseconds.

5. A blank field if there is no timestamp.

6. Does not display milliseconds.

7. Displays:
   a. Localized format (e.g. AM/PM
   b. The earliest value that could be displayed for a timestamp (e.g. around 1/1/1970; the actual value being 1/1/1970 with the Time Zone offset from UTC applied.) if there is no timestamp value recorded.

VALIDATE_PASSWORD_CHANGE

Script filename without the extension and without a path.**Note:** CIMPLICITY looks for the script file in the following order.

1. .bcl
2. .bclrt

WAIT_PROJECT_TIMEOUT

For	CimView Project
Purpose	To increase the time before a project times out when a CimView script logs into a project that is not running to get a point value and it take longer than the default time to start the Router.
Value	Number of seconds.

Default Value	20	Seconds

Device Communications Parameters

The following global parameters are modified on a per-project basis for particular Communications options.

	Allen-Bradley Communications
	Allen-Bradley DF1 Communications
	CCM2 Communications
	DDE Communications (As-Is-Local only))
	FloPro/FloNet Communications
	Honeywell IPC 620 Communications
	Marquee network configuration
	Mitsubishi Serial Communication
	Mitsubishi TCP/IP Communications
	Modbus RTU Communications
	Modbus TCP/IP Communications
	OMRON Host Link Communications
	OMRON TCP/IP Communications
	Series 90 TCP/IP Triplex Communications
	Smarteye Communications
	SNP Communications
	SNPX Communications
	Square D SY/MAX Communications
	Toyopuc Ethernet Device Communications

Allen-Bradley Communications

	AB_WS_UNSO_PLC5_FLOAT
	ABI_MAXDEF
	ABETH_PLC_POLL_TIMEOUT
	ABETH_PLC_REQUEST_TIMEOUT
	ABETH_PLC_RESPONSE_TIMEOUT

	ABETH_UNSO_QUEUE_SIZE

Allen-Bradley DF1 Communications

	ABDF1_<PORT>_USESPFP

CCM2 Communications

	<PORT>_TURN_AROUND_DELAY

DDE Communications (Legacy)

	DDE_UNAVAIL_ON_ILLEGAL_VAL

FloPro/FloNet Communications

	DC_RETRY_ONE_DEVICE
	FLOPRO_RESPONSE_TIMEOUT
	FLOPRO_STATIC_MODEL

Honeywell IPC 620 Communications

	HWABC_DEBUG
	<PORT>_DEBUG

Marquee network configuration

	TSERV_<PORT>
	TSERV_<COM>

Note: Other Marquee global parameters are in the main list *(page 424)* .

Mitsubishi Serial Communication

	COM<PORT>_TO

Mitsubishi TCP/IP Communications

	DC_CONNECT_MS
	DC_TCP_POLL_MS
	DCQ_CONNECT_URETRY_CNT
	DCQ_DEAD_TIME

	MMAX_SYNC_TICKS
	MSYNC_TICKS
	<PORT>_DC_CONNECT_MS
	<PORT>_DC_TCP_POLL_MS
	<PORT>_DCQ_DEAD_TIME
	<PORT>_MSYNC_TICKS
	<PORT>_SOCKET_PORT

Modbus RTU Communications

	MB_COMM_TIMEOUT
	MB_LOG_PROTOCOL
	MDBC
	<PORT>_LOG_PROTOCOL

Modbus TCP/IP Communications

	<DEVICE>_READ_REQUEST_TIMEOUT
	<DEVICE>_REQ_RETRY
	<DEVICE>_TIMEOUT_RETRY_DELAY
	DeviceId_CONN_SECONDARY
	DeviceId_CONSERVES_CONN
	DeviceId_ONE_COIL_WRITE
	DeviceId_ONE_REG_WRITE
	MBEDC_<device_id>
	MBETH_ASYNC_CONNECTION
	MBETH_DISABLE_IO_ERRLOG
	MBETH_DISABLE_UNSO_DATA
	MBETH_ENABLE_PROTOCOL_DEBUG
	MBETH_READ_REQUEST_TIMEOUT
	MBETH_REQ_MILLISECOND_TIMEOUT
	MBETH_REQ_RETRY
	MBETH_REQ_TIMEOUT
	MBETH_SOCKET_PORT
	MBETH_TIMEOUT_RETRY_DELAY

	MBETH_UNSO_DATA_QUEUE_SIZE
	<PORT>_ASYNC_CONNECTION
	<PORT>_DISABLE_IO_ERRLOG
	<PORT>_DISABLE_UNSO_DATA
	<PORT>_ENABLE_PROTOCOL_DEBUG
	<PORT>_MAX_BUFFER_SIZE
	<PORT>_READ_REQUEST_TIMEOUT
	<PORT>_REQ_MILLISECOND_TIMEOUT
	<PORT>_REQ_RETRY
	<PORT>_REQ_TIMEOUT
	<PORT>_SOCKET_PORT
	<PORT>_TIMEOUT_RETRY_DELAY
	<PORT>_UNSO_DATA_QUEUE_SIZE

OMRON Host Link Communications

	COM<PORT>_TO
	OMRON_MAX_BUFFER_SIZE

OMRON TCP/IP Communications

	OMRON_MAX_BUFFER_SIZE
	<PORT>_OMRON_SA1
	<PORT>_OMRON_SERVICE
	<PORT>_OMRON_SNA
	<PORT>_SEQUENCE_SID
	<PORT>_TO

Series 90 TCP/IP Triplex Communications

	<DEVICE>_MAX_POLL_MSG
	<DEVICE>_MAX_REQUEST_CACHE
	<PORT>_ALLOW_MULTIMSG
	<PORT>_DISABLE_KEEPALIVE
	<PORT>_MAX_CACHE
	<PORT>_MAX_POLL_MSG

	<PORT>_MAX_REQUEST_CACHE
	S90TCP_ALLOW_MULTIMSG
	S90TCP_ALLOW_UNSO
	S90TCP_DC_BIT_REVERSE
	S90TCP_<DEVICE_ID>_IS_SE
	S90TCP_DISABLE_KEEPALIVE
	S90TCP_MAX_CACHE
	S90TCP_MAX_POLL_MSG
	S90TCP_MAX_REQUEST_CACHE
	S90TCP_RECONNECT_DELAY

Smarteye Communications

	<PORT>_LOG_WARNING
	<PORT>_MODE
	<PORT>_POLL_LIMIT
	<PORT>_RESTART_SEA
	<PORT>_SEA_HANDSHAKE_TIMEOUT
	SE_LABEL_LEN
	SEA_HANDSHAKE_TIMEOUT

SNP Communications

	SNP_SEND_BREAK
	SNP_IDLE_TIME

SNPX Communications

	<DEVICE>_VALIDATE
	SNPX_BROADCAST_TIME
	SNPX_VALIDATE

Square D SY/MAX Communications

	SYMAX_<PORT>_WORD_SWAP

Toyopuc Ethernet Device Communications

	<PRCNAM>_CONNECTION_TIMEOUT
	<PRCNAM>_READ_WRITE_TIMEOUT
	<PORT>_BIND_ADDR

Global Parameter Files

Global Parameter Files

When you modify the global parameter list in the Workbench and perform a configuration update, CIMPLICITY modifies the system or project's global parameters file (glb_parms.idt).

You can view a global parameter file by opening it in Windows Notepad through an MS DOS window.

Locate a Global Parameter File

You can locate the **glb_parms.idt** file through an MS DOS window.

1. Click Tools on the Workbench menu bar.
2. Select Command Prompt.

   An MS DOS window opens.

3. Enter the following commands in the Command Prompt window:

   `cd <dir>`

   where

   <dir> is the directory where you need to make the changes.

   For global parameters that apply to the:

   - System (all projects), enter `cd %BSM_ROOT%data`
   - Current project, enter `cd %SITE_ROOT%master`

   `idtpop glb_parms`

   `notepad glb_parms.idt`

   A Windows Notepad opens displaying a record of the global parameters in the all projects or specified project directory. The list of global parameters in this file corresponds to the list that displays in the Workbench.

4. Find the parameter you want to check.

   You can edit the value in this file if you need to.

5. Save the file if you have modified it.

6. Exit Notepad.

7. Type `scpop glb_parms` at the DOS prompt. A new binary version of the Global Parameters file will be produced and put in the master directory.

8. ype `exit` to exit the Command Prompt window.

   When you are ready to implement the change in the run-time system, you will have to stop the project(s), perform a Configuration Update, and restart the project(s). The Global Parameters file that is used on a Viewer is located in the CIMPLICITY installation directory in a subdirectory called data (e.g., c:\cimplicity\hmi\data).

   Note: The Global Parameters file that is used on a Viewer is located in the CIMPLICITY installation directory in a subdirectory called data (e.g., c:\cimplicity\hmi\data

Sample GLB_PARMS.IDT File

The following is a sample glb_parms.idtfile:

|-* IDT file generated by IDTPOP utility v1.0

* RECORD: GLB_PARMS SYSTEM GLOBAL PARAMETERS

*

*   0 PARM_ID   Global Parameter Identifier

*   1 parm_type   CH-0,S-1,IS-2,I-3,BT-4,WRD-5,LG-6,BL-7

*   2 parm_value Parameter Value up to 80 chars

*

ACK_TOUT|3|0

CLR_TOUT|3|0

CONNECT_DROP_PERIOD|3|600

CONNECT_RETRY_PERIOD|3|90

DBDL_ABORT_TIMEOUT|1|50

DBDL_DEVICE_CHECK_PERIOD|1|30

DB_QUEUE_OVERFLOW_DELAY|1|30

DB_QUEUE_SIZE|1|20

DEL_OPT|1|AR

DOWNLD_PASSWD

DYN_CFG|1|YES

GSM_UNAVAIL_COLOR_KEY|1|0

IPC_BCAST_INTERVAL|1|90

LOGIN_RETRY_PERIOD|3|15

LOG_OPT|1|

PTDL_QUANTIZATION|1|100

PTMRP|1|MASTER_PTM0_RP

RAW_LIMIT_ALARM|1|YES

REPEAT_TOUT|3|0

RTR_ACCEPT_CONN|1|Y

RTR_DISABLE_BCAST|1|Y

SETPOINT_SECURITY|1|NO

SHORT_FILENAMES|1|NO

SVC_RETRY_COUNT|1|30

SVC_RETRY_DELAY|1|100

SYSNAME|1|MMI_ONE

Process Health Parameters

About Process Health Parameters

Process Health Parameters enables you to have any selected CIMPLICITY project process automatically monitored.

Warning: Keep in mind, restarting a process can have serious consequences, for example losing point values or disconnecting users who are logged in from other locations.

If selected, monitoring can include attempted process restarts, process failure and/or project failure.

Step 1 *(page 512)*	Open the Process Health dialog box.
Step 2 *(page 513)*	Enter Process Monitor specifications.

Note: If Configuration Securityis enabled, only roles that have Projectschecked on the Configuration tab in the Roles dialog box will have access to the Process Health parameters configuration.

Note: The Event Manager Resident Process (EM_RP) reports bad health when it encounters scenarios where string and/or stack space has been used up or overrun. You can use the Process Health Parameters application to choose appropriate actions for this scenario.

Step 1. Open a Process Health Dialog Box

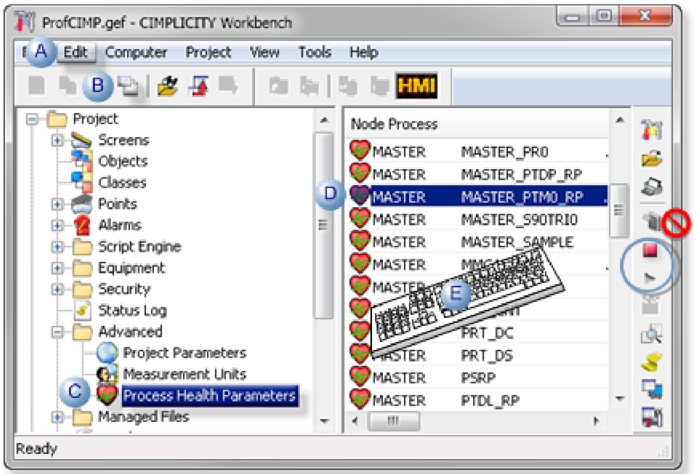
1. Startthe CIMPLICITY project.

   Note: Do not enable dynamic configuration.

2. Select **Project>Advanced>Process Health Parameters** in the Workbench left pane.
3. Select a process in the Workbench right pane.

   Note: All the processes that are included in the CIMPLICITY project are listed in the Workbench right pane, including device communications processes.

4. 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 **Process Health Parameters**. b. Select Properties on the Popup menu.	
D	In the Workbench right pane:	
	Either	Or
	Double-click a process.	a. Right-click a process. b. Select Properties on the Popup menu.
E	Press Alt+Enter on the keyboard.	

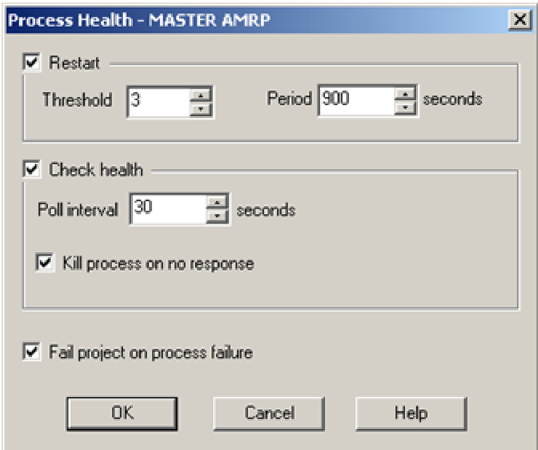
5. Right-click **Process Health Parameters**.
6. Select Properties on the Popup menu.
7. Right-click a process.
8. Select Properties on the Popup menu.

Step 2. Enter Process Monitor Specifications

- Process Health configuration.
- process_dependents.cfg file.

Process Health configuration

Check any of the following groups to enable the feature for the selected process.



rect 11, 28, 322, 86 *(page 514)*
rect 7, 94, 318, 193 *(page 514)*
rect 7, 193, 318, 239 *(page 515)*

Field	Description	
Restart	Restarts the process a specified number of times within a selected number of seconds. **Note:** The process you are configuring may have dependent processes (processes that start after the selected process starts). If it does, you must configure a process_dependents.cfg *(page 515)* file to insure that the dependent processes will stop and restart based on the selected process behavior. Example 1. Process #2 (e.g. Derived Point Process) • Is dependent on another process #1 (e.g, Point Manager). • Assumes that process #1 is started first. 2. Process #1 fails and is restarted 3. Dependent process #2 must be stopped and restarted immediately following Process #1.	
	Threshold	Number of times the process can be restarted, within the number of seconds specified by the Restart period, before it is failed.
	Period	Number of seconds in which the restart threshold is operational.
	Default	Disabled.
Check health	Send messages to the process to determine if it is running correctly. For example, a process may appear to be running when, in fact, it is hung.	
	Poll interval	Number of seconds that elapse between polls.
	Kill process on no response	If the process does not respond when it is polled, it will be stopped.
	Default	Disabled
	Note: Check health is enabled only if the process supports active health checks.	

Fail project on process failure	(For clusters only) The project stops when the selected process fails.	
	Default	Disabled

Important:

- Process health does not support dynamic configuration. Specifications for the process must be entered when the project is stopped, then incorporated through a project configuration update.
- If you change a process name you must identify the process in the process_dependents.cfg file.

Process_Dependents.cfg file

- The process_dependents.cfg file must be edited if:
- Default PRT service names have been changed.
- Customized processes that have dependencies are configured to restart.
- The default process_dependents.cfg is located in the ...\Proficy CIMPLICITY\bsm_data folder.
- The default process_dependents.cfg file is as follows.

|-*

* The first field is the name of the process that has died

* The remaining fields are the names of processes that will be stopped and restarted if they are currently running

PRT_DC|PRT_DS

Process Control

About Process Control

Program Control is an interactive process that lets you start, stop, and display the current state of CIMPLICITY processes. This section describes the functions available in Program Control and procedures for running it.

Step 1 *(page 516)*	Open the CIMPLICITY Process Control window.
Step 2 *(page 518)*	Connect to a Project in CIMPLICITY Process Control.
Step 3 *(page 519)*	Determine the CIMPLICITY process status.

Step 4 (page 520)	Determine the correct startup/shutdown order.
Step 5 (page 521)	Start/stop processes.

Step 1. Open the CIMPLICITY Process Control Window

CIMPLICITY provides several methods to open the CIMPLICITY Process Control Window.

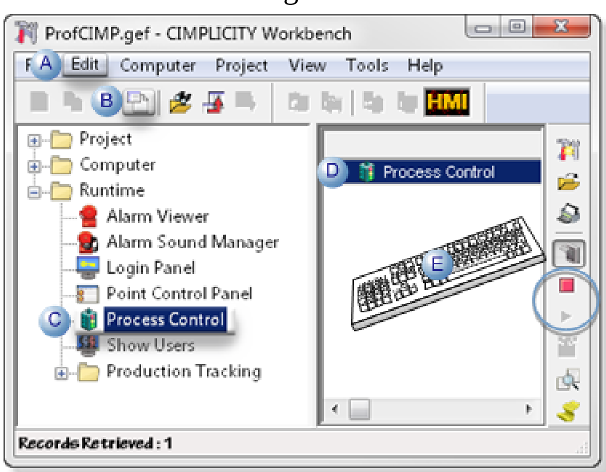
- Workbench
- Start menu

Workbench

1. Select **Project>Runtime>Process Control** in the Workbench left pane.

2. Select Process Control 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:	
	Either	Or
	Double click **Process Control**.	a. Right-click **Process Control**. b. Select Properties on the Popup menu.
D	In the Workbench right pane:	
	Either	Or

	Double click **Process Control**.	a. Right-click **Process Control**. b. Select Properties on the Popup menu.
E	Press Alt+Enter on the keyboard.	

Start Menu

4. Right-click **Process Control**.

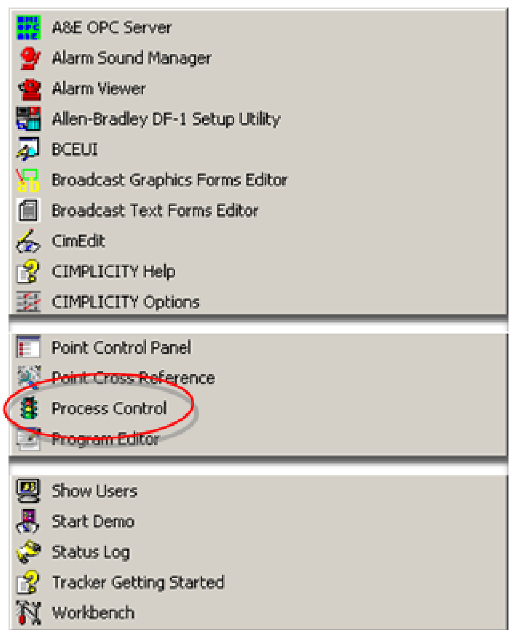
5. Select Properties on the Popup menu.

6. Right-click **Process Control**.

7. Select Properties on the Popup menu.

8. Click Start on the Windows task bar.

9. Select (All) Programs>Proficy HMI SCADA CIMPLICITY version>Process Control.



Result:

- If no project is running:

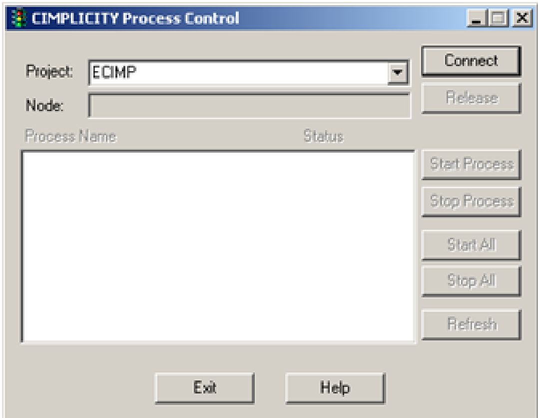
The CIMPLICITY® Project dialog box opens.

Select a CIMPLICITY project to run.

The CIMPLICITY Process Control dialog box opens.

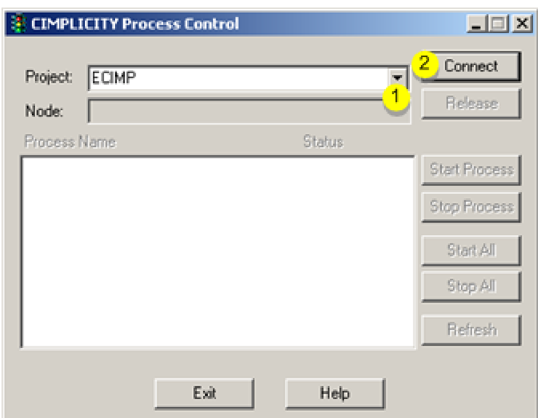
- When a project is running:

The CIMPLICITY Process Control dialog box opens.



Step 2. Connect to a Project in CIMPLICITY Process Control

Connect to a project as follows.

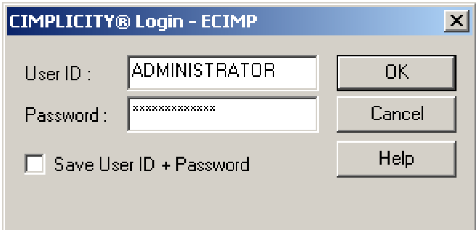


1	Select the project to connect to. 1. Click the drop-down list button to the right of the **Project** field. 2. Select the project you want to connect to.
2	Click Connect.

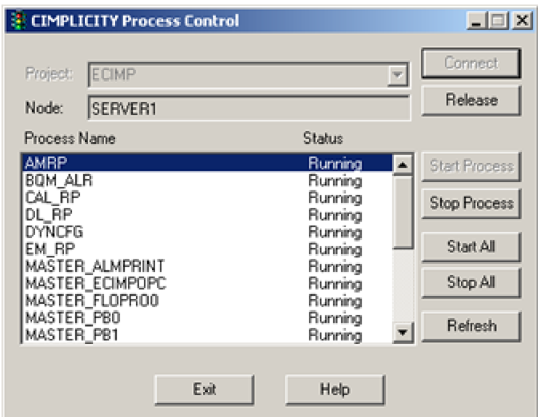
Note: If you are not currently logged in to the project, a CIMPLICITY® Login dialog box opens.

1. Enter a valid **User ID** and **Password**.

2. Click OK.



When CIMPLICITY processes a saved login or the login just entered, the CIMPLICITY Process Control dialog box displays with the list of process names and statuses for the project you selected.



Important: This is not the recommended way to start or stop CIMPLICITY software. You should only use CIMPLICITY Program Control to start and stop processes if you have been instructed to do so by GE Intelligent Platforms support personnel, or if you are testing an API application. In the latter case, you should only be starting and stopping your API application.

Step 3. Determine the CIMPLICITY Process Status

CIMPLICITY processes display in the left column of the CIMPLICITY Process Control dialog box. The status appears on the row, to the right of the process.

A process status can be:

- Halted or
- Running.

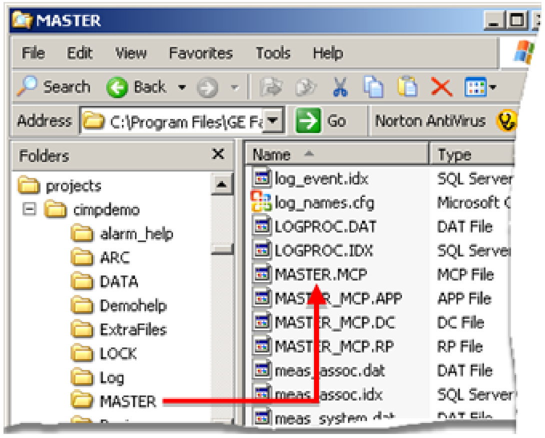
The following table lists background Process IDs for all base system and product options for CIMPLICITY processes. The set of options running on a node depends on the communication protocols, printers, and product options you have installed.

AMRP	Alarm Management Resident Process
CAL_RP	Action Calendar
CNCALM	CNC Alarm Process

DL_RP	Data Logger Resident Process
DYNCFG	Dynamic Configuration Process
EM_RP	Event Manager Resident Process
MTCPSI_RP	Modbus TCP Client Interface
MRTUSI_RP	Modbus RTU Client Interface
MASTER_<port>	Device Communications Interfaces
MASTER_<printer_name>	Alarm Manager Line Printer Process
MASTER_PTDP_RP	Point Management Virtual Point Resident Process
MASTER_PTM<n>_RP	Point Management Resident Process
PASH	SOA resident process creates the SOA host component and launches CIMPLICITY Service Provider.
PDC_DS	Tracker - Decision Control Data Server
PROF_RP	Profile Trending
PRT_CNT	Tracker–Tracking Attribute Counts
PRT_CLNT	Tracker–Client (communicates from one project to another)
PRT_DC	Tracker–Data Collector
PRT_DS	Tracker–Tracking Data Server
PRT_GRD	Tracker–Tracking Graphic Display
PTDL_RP	Point Data Logger Resident Process
PTX_RP	Point Translation Resident Process
<rcositename>	Tracker–Tracker Routing Control Objects Site
RCODB_RP	RCO Configuration Provider to RCO Runtime User Interface
SPC_<group file name>	Statistical Process Control
UR_RP	User Registration Resident Process

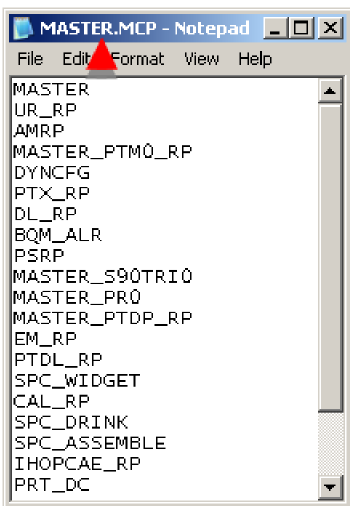
Step 4. Determine the Correct Startup and Shutdown Order

The order in which some processes should be started and stopped is critical to their operation. If you are attempting to start or stop individual processes, and you want to make sure you are doing so in an orderly manner, you can check the master.mcp file in your project's Master directory. This file lists, in correct order, the files that are started up for your project.



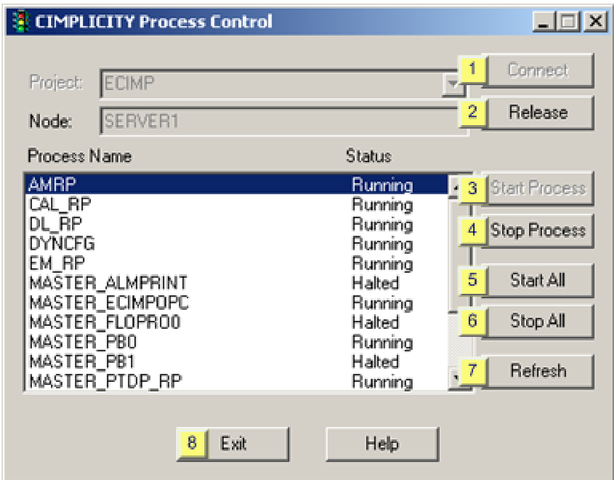
Example

The master.mcp file for a project is on a computer whose Node ID is Master.



Step 5. Start/Stop Processes

Options for starting and stopping processes in the CIMPLICITY Process Control window include the following.



rect 101, 258, 186, 287 [(page 525)]
rect 285, 216, 370, 245 [(page 524)]
rect 286, 183, 371, 212 [(page 524)]
rect 286, 52, 371, 81 [(page 523)]
rect 284, 125, 369, 154 [(page 523)]
rect 286, 154, 371, 183 [(page 524)]
rect 285, 97, 370, 126 [(page 523)]
rect 286, 25, 371, 54 [(page 523)]

1 (page 523)	Connect
2 (page 523)	Release
3 (page 523)	Start Process
4 (page 523)	Stop Process
5 (page 524)	Start All
6 (page 524)	Stop All
7 (page 524)	Refresh
8 (page 525)	Exit

1	Connect

Connects *(page 518)* the Process Control to a project.

2	Release

Click Release to releases the Process Control from the connected project.

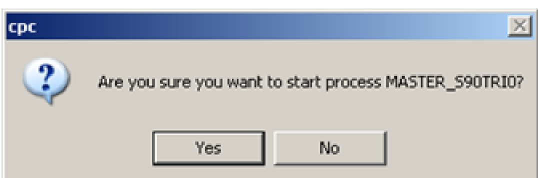
Result: When the project is released the Process Control window is emptied. You can either select another project to connect *(page 518)* to or exit the Process Control.

3	Start Process

1. Select a process that is halted.

1. Click Start Process.

A message displays to confirm the start.



1. Click Yes.

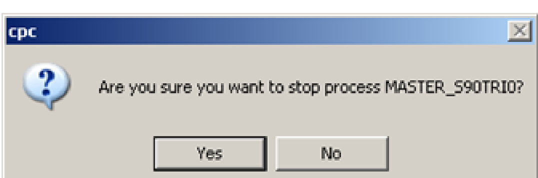
Result: If the process starts successfully the process status changes to Running.

4	Stop Process

1. Select a process that is running.

1. Click Stop Process.

A message displays to confirm the stop.



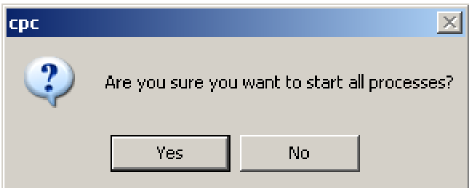
1. Click Yes.

Result: If the process stops successfully the process status changes to Halted.

5	Start All

1. Click Start All.

A message displays to confirm starting all processes.



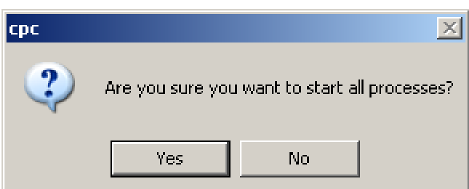
1. Click Yes.

Result: All halted processes are started; the status for all processes that start successfully changes to Running.

6	Stop All

1. Click Stop All.

A message displays to confirm stopping all processes.



1. Click Yes.

Result: All halted processes are stopped; the status for all processes that stop successfully changes to Halted.

Note: Even though all of the processes are halted, Process Control continues to control the project. You must click Release in order to release the project.

7	Refresh

The process list displays the process statuses at the time you connected to the project. These statuses are updated automatically only when you perform a startup or shutdown operation.

Click Refresh.

Result: CIMPLICITY refreshes the list to display the status of all processes on the list.

8	Exit

Click Exit to close the Process Control window.

Results: Process Control releases the project if it has not already been released; the Process Control window closes.

OEM Key

About the OEM Key

OEM Key: Use

The OEM Key:

- Is used for systems that do not have development servers and, more often than not, do not have development viewers.
- Enables OEMs to:

1. Remotely connect to a CIMPLICITY system or get directly onto the system.
2. Perform whatever configuration changes are needed.

   OEM KEY: Limitations

   The OEM Key:

   - Automatically ends development mode on the runtime server in two hours.
   - Requires stopping any running projects twice.

   Any projects that are running must be stopped:

3. Before the OEM Key can be activated.
4. When the OEM Key terminates (automatically or if it is exited before the two hours).

   Note: Projects are stopped automatically when the OEM Key terminates or is exited.

   - Makes the runtime server a stand-alone machine; it cannot communicate at all with its viewers while the OEM Key is active.

OEM Key Activation

OEM Key Activation

1. Open the Explorer or a Command Prompt window
2. Locate the **CIMPLICITY exe** directory.

   (If you accepted the default location for CIMPLICITY software during installation, it is **C:\Program Files\Proficy\Proficy CIMPLICITY**.)

3. Run **oemkey.exe**.

   The CIMPLICITY License Agreement dialog box opens.

4. Click **Yes** to accept all the terms of the license agreement.

The next dialog box that opens depends on your computer status:

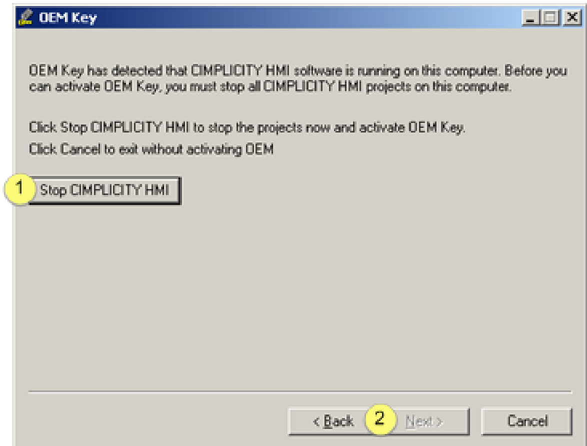
- If any projects are running, the OEM Key dialog box opens.
- If no projects are running, the CIMPLICITY OEM Key dialog box opens.

Note: If you click **No**, OEM Key terminates.

Projects Stopped for OEM Key Activation

If OEM Key finds that CIMPLICITY projects are running on your computer, the OEM Key dialog box opens. This dialog box gives you the opportunity to exit OEM Key or terminate the projects and activate OEM Key.

Do the following.



1	Click Stop CIMPLICITY HMI.
2	Click Next, which is enabled when all CIMPLICITY projects stop running.

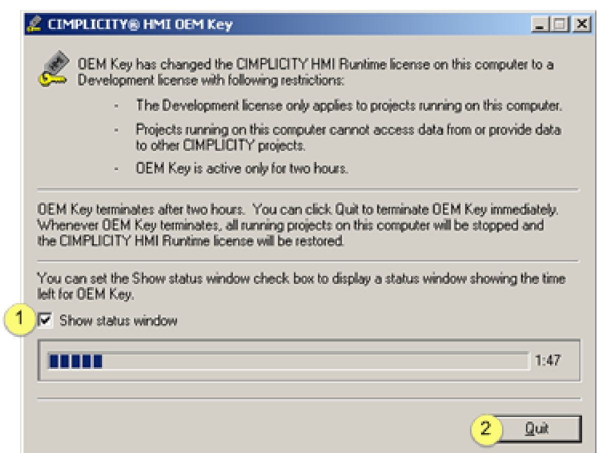
Result: The CIMPLICITY OEM Key dialog box opens.

Note: Click Cancel to exit OEM Key without stopping the running projects.

OEM Key Status

The CIMPLICITY OEM Key dialog box shows you the time left for the OEM Key.

You can optionally do the following.



1	Check the Show status window check box to display the time left. This status window displays on top of all the windows on your screen. 
2	Click Quit to cancel installation.

Note: If you exit the dialog box or click Quit, the OEM Key terminates immediately.

OEM Key Termination

OEM Key terminates at the end of two hours. You can terminate it earlier by exiting the CIMPLICITY OEM Key dialog box. When OEM Key terminates it:

- Displays a dialog box warning you that it will stop all running projects.
- Stops all running projects.
- Converts your temporary Development System License back to your Runtime System License.

You can now restart your projects.