Cat. No. W10E-EN-01

Software Release 2.0

CX-Supervisor

USERS MANUAL

OMRON

CX-Supervisor
User Manual

Software Release 2.0

Notice

OMRON products are manufactured for use according to proper procedures by a qualified operator and only for the purposes described in this manual.

The following conventions are used to indicate and classify precautions in this manual. Always heed the information provided in them. Failure to heed precautions can result in injury to people or damage to the product.

DANGER! Indicates information that, if not heeded, is likely to result in loss of life or serious injury.

WARNING Indicates information that, if not heeded, could possibly result in loss of life or serious injury.

Caution Indicates information that, if not heeded, could result in relatively serious or minor injury, damage to the product, or faulty operation.

OMRON Product References

All OMRON products are capitalised in this manual. The word "Unit" is also capitalised when it refers to an OMRON product, regardless of whether or not it appears in the proper name of the product.

The abbreviation "PLC" means Programmable Logic Controller and is not used as an abbreviation for anything else.

Visual Aids

The following headings appear in the left column of the manual to help you locate different types of information.

Note: Indicates information of particular interest for efficient and convenient operation of the product.

1, 2, 3… Indicates lists of one sort or another, such as procedures, checklists etc.

🔄 Represents a shortcut on the Toolbar to one of the options available on the menu of the same window.

Start Indicates a program must be started, usually by clicking the appropriate option under the standard Windows 'Start' button.

© OMRON, 2007

No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice. Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

About this Manual

This manual describes the CX-Supervisor application and its ability to create, modify and run SCADA applications.

This manual contains the following:

Chapter 1 - Graphics Editor. This chapter describes the Graphics Editor and its various tools and controls.

Chapter 2 - Pages. This chapter describes the procedures involved in the creation and amendment of pages.

Chapter 3 - Points. This chapter describes the procedures involved in the creation, amendment and removal of points. The use of PLCs with points is also described.

Chapter 4 - Objects. This chapter describes procedures involved in the creation, amendment and removal of objects.

Chapter 5 - ActiveX Objects. This chapter provides an overview of using ActiveX objects in CX-Supervisor.

Chapter 6 - Projects. This chapter describes procedures involved in the creation, amendment and removal of projects.

Chapter 7 - Graphics Library. This chapter describes the library of graphic objects and how to create and amend new libraries.

Chapter 8 - Alarms. This chapter describes the procedures involved in the creation and amendment of alarms.

Chapter 9 - Animation. This chapter describes the use of animations and how they can be applied to a CX-Supervisor project.

Chapter 10 - Recipes. This chapter describes the creation of recipes, and how they can be used.

Chapter 11 - Data Logging. This chapter describes the data logging facilities, including setting up, viewing and exporting

Chapter 12 - Databases. This chapter describes the Database facilities, including connections reading and writing data.

Chapter 13 - Multilingual Features. This chapter describes the Multilingual features, including creation of dynamically switching applications.

Chapter 14 - OPC Client. This chapter introduces OPC Server to a new user, and explains how to use CX-Supervisor as an OPC Client

Chapter 15 - Connecting to a Remote CX-Supervisor application. This chapter explains how to create a distributed solution by connecting together CX-Supervisor applications on networked PCs.

Chapter 16 - Connecting to Omron Industrial Components. This chapter details connecting CX-Supervisor to Omron's Industrial Components.

Chapter 17 – Best Practices. This chapter details the best practices to get the most from CX-Supervisor including how to plan and design your application.

Appendix A – Configuring a PC for remote connection. The appendix explains how to configure a PC for remote connection.

Appendix B – Frequently asked questions. The appendix list some FAQs, and of course their answers.

Appendix C – Troubleshooting. The appendix is a guide to troubleshooting and resolving problems.

Appendix D – CX-Server error codes. The appendix lists the error codes from the communication software, CX-Server, and their meanings.

Appendix E – Using with the Omron DyaloX. The appendix lists information when using with the Omron’s DyaloX Industrial PC.

Appendix F – Obsolete Features. The appendix details obsolete features for backwards compatibility only.

A ***Glossary of Terms*** and ***Index*** are also provided.

TABLE OF CONTENTS

Table of Contents chapter 3 continued Page

Table of Contents chapter 6 continued Page

Table of Contents chapter 8 continued Page

Table of Contents chapter 10 continued Page

Table of Contents Page

Table of Contents Appendices Page

CHAPTER 1
Graphics Editor

This chapter describes the Graphics Editor and the various tools and controls available. It also provides instructions for using these tools and controls and is supported with suitable screen displays.

About the Graphics Editor

The Graphics Editor enables a variety of objects to be created on a page. Supported objects are:

- Graphical objects.
- Control objects.
- ActiveX objects.

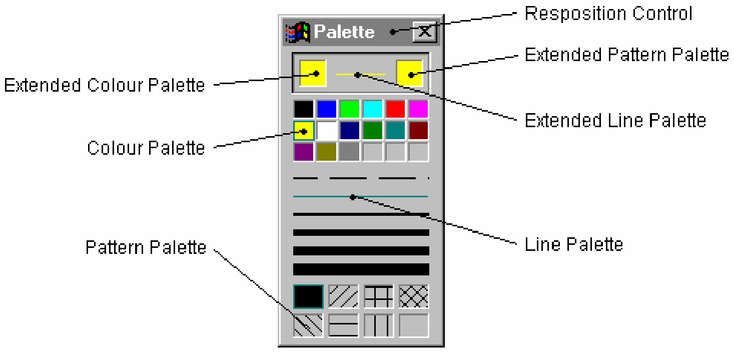
Graphical objects are geometric shapes, for example ellipses and polygons, but also include Text objects. Control objects allow information to be displayed and entered in clear way through the use of Wizards. Examples of control objects include Pushbuttons and Trend Graphs. ActiveX objects or controls are from sources external to CX-Supervisor.

Refer to chapter 4, Objects for further information regarding control objects and bitmap objects. Refer to chapter 5 for further information on ActiveX Objects.

The tools are contained on a Control Bar, and within palettes. The palettes allow all similar types of tool to be kept together, and are contained within a free floating window. The various tools and palettes are discussed in the following chapters. Status and help information is presented in a Status Bar located at the bottom of the main CX-Supervisor window.

Palette

The Palette is always displayed on top of any pages created within CX-Supervisor. If no pages are open, the Palette is not displayed, and if there are no currently active pages, the Palette is rendered inactive. An example of the Palette illustrating the position of various palettes and controls is shown below:



The Palette can be removed or re-displayed at any time by selecting *Palette* from the *View* menu.

A tick next to the name indicates the Palette is currently displayed. CX-Supervisor saves the settings when it is exited and restores them when it is next run.

Each of the palettes and controls is discussed in more detail in the following paragraphs.

Colour Palette

To create an object in a particular colour: before selecting the object's tool from the Graphic Object bar, first select the colour by clicking in the appropriate square in the Colour Palette.

Note: It is not strictly necessary to select a colour for an object before creating it, however if no colour selection is made, the object retains the same colour as the last object.

To apply a colour to a previously created object, select the object on the page and click with the left mouse button in the appropriate square in the Colour Palette.

Note: Colour may only be applied to some graphic objects. It cannot be applied to embedded objects or bitmap graphics.

Line Style Palette

The Line Style Palette is located in the Palette just below the Colour Palette.

To create an object with a particular line style: before selecting the object from the Tool Palette, first select the line style by clicking on the appropriate line in the Line Style Palette.

Note: It is not strictly necessary to select a line style for an object before creating it, however if no line style selection is made, the object retains the default style of thin solid.

To apply a line style to a previously created object, select the object on the page and click on the appropriate line in the Line Style Palette.

Note: Line styles may only be applied to some graphic objects. They cannot be applied to text, embedded objects, controls or bitmap graphics.

Fill Pattern Palette

The Fill Pattern Palette is located at the bottom of the Palette.

To create an object with a particular fill pattern: before selecting the object's tool from the Tool Palette, first select the fill pattern by clicking in the appropriate square in the Fill Pattern Palette.

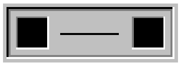
Note: It is not strictly necessary to select a fill pattern for an object before creating it, however if no fill pattern selection is made, the object retains the default of no pattern.

To apply a fill pattern to a previously created object, select the object on the page and click in the appropriate square in the Fill Pattern Palette. With a fill pattern applied to an object, clicking in the Colour Palette with the left mouse button allows the foreground colour of the chosen fill pattern to be changed. Clicking in the Colour Palette with the right mouse button allows the background colour of the chosen fill pattern to be changed.

Note: Fill patterns may only be applied to solid graphic objects. They cannot be applied to unfilled graphic objects, text objects, embedded objects, controls or bitmap graphics.

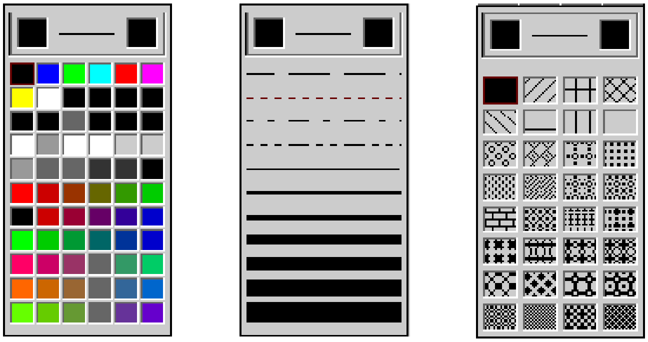
Extended Colour, Line Style and Fill Pattern Palette

Each of the Colour, Line Style and Fill Pattern Palettes may be extended to allow access to more choices. Only one palette may be extended at any one time. The Extended Selection tools are located at the top of the Palette and are illustrated as follows:



The left hand square controls the extension of the Colour Palette; the line between the two squares controls the Line Style Palette extension, and the right hand square control the Fill Pattern Palette extension.

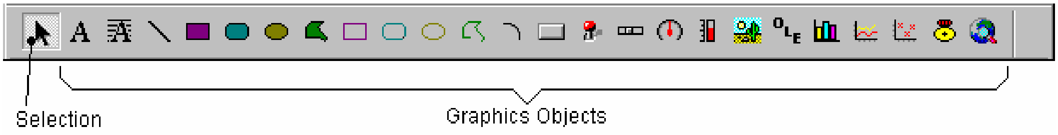
An example of the Extended Colour, Line Style, and Fill Pattern Palettes is shown below:



The operation of each of the extended palettes is identical to that of the non-extended palettes.

Graphic Object bar

The Graphic Object bar contains the tools to create the graphic objects which can be placed on pages within CX-Supervisor.



In general terms, the operation of the Graphic Object bar is simple. With a page active the Graphic Object bar is active, and operation involves merely clicking on the desired tool to select it, and then clicking and dragging to the appropriate point in the page. This is not the case with the text, polygon or polyline tools, however the operation of these tools is fully discussed in chapter 4, Objects.

Moving the mouse pointer over any of the tools causes CX-Supervisor to display a tooltip to describe the tool.

By default, the cursor returns to Selection mode (an arrow) after drawing an object. If you would prefer it to remain in its current state, clear the *Return to select mode* checkbox in the Editing Preferences dialog, which is found on the CX-Supervisor file menu.

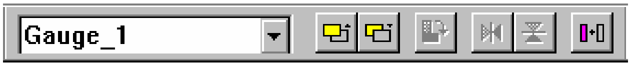
For further details concerning the tools contained within the Graphic Object bar refer to chapter 4, Objects.

Control Bar

CX-Supervisor provides a Control Bar containing formatting and object manipulation tools. The Control Bar can be activated or de-activated from the *View* menu. To activate the Control Bar, select *Control Bar* from the *View* menu; CX-Supervisor places a check mark next to it signifying its active status. To de-activate it, repeat the procedure (the check mark is removed), and the Control Bar is no longer displayed. The various tools on the control bar allow the properties of objects to be modified.

A control may be activated with more than one object selected to change a particular attribute of all of the selected objects, e.g. when all graphical objects on a page are selected and the **Rotate** button is pushed in and the desired angle specified, all selected objects are rotated.

An illustration of the Control Bar is as follows:



The Control Bar may vary slightly from this according to the display resolution currently in use - the number of buttons displayed depends on the available space - at higher resolutions more buttons are displayed.

The Control Bar can be removed from the display at any time by selecting *Control Bar* from the *View* menu.

A tick next to the name indicates that it is currently displayed. CX-Supervisor saves the settings when it is exited and restores them when it is next run.

The specific operation of the controls on the Control Bar is described in the following paragraphs.

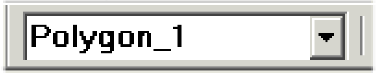
Object Identification

When an object is created, CX-Supervisor gives it a unique identifier. This identifier consists of the object type and a sequential number (starting at 1). For example, a text object could have an identifier of TEXT_1; a polygon object could have an identifier of POLYGON_3 etc. The *Object Identification* field displays a list of all current objects on a page.

The identification of an object can be changed by clicking on the entry in the Object Identification control, typing over the entry and pressing <Return>. A confirmation box is provided to double check the operation, click the **OK** pushbutton to proceed with the name change or the ***Cancel*** pushbutton to abort the operation.

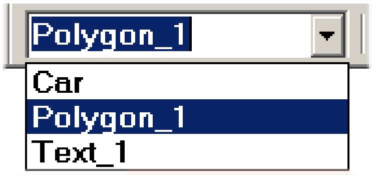
It can at times be difficult to select a particular object if it is mostly obscured by other graphic objects. The Object Identification control allows an object to be selected by simply selecting its name from the presented list. Click on the arrow button to display the list and select the desired object. On selection the object is highlighted.

The following illustration shows the Object Identification control in de-selected state:



Polygon_1

The following illustration shows the Object Identification control in selected state:



Polygon_1

Car

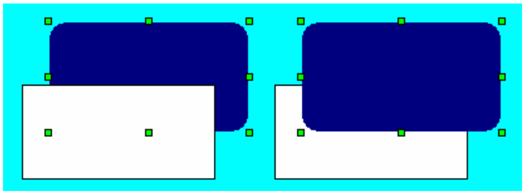
Polygon_1

Text_1

Raise Up One

The **Raise Up One** button allows an object which forms part of an overlapping group of objects to be moved nearer the top of the group, one layer at a time. With an object selected, clicking the pushbutton once moves the object up one layer. This continues until the object is at the top. Clicking the pushbutton with the object at the top of a group has no effect.

The following example illustrates the state of a group of two objects before and after clicking the **Raise Up One** button with the black round rectangle selected:

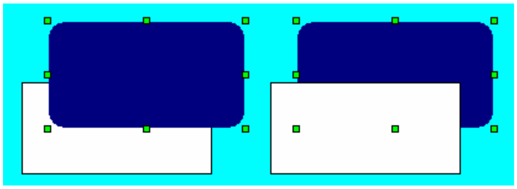


Before After

Lower Down One

The **Lower Down One** button allows an object which forms part of an overlapping group of objects to be moved nearer the bottom of the group, one layer at a time. With an object selected, clicking the button once moves the object down one layer. This continues until the object is at the bottom. Clicking the button with the object at the bottom of a group has no effect.

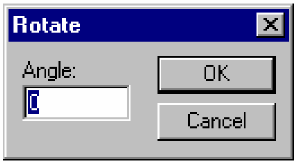
The following example illustrates the state of a group of two objects before and after clicking the **Lower Down One** button with the grey polygon selected:



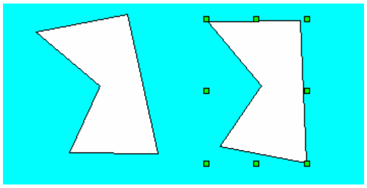
Before After

Rotate

The **Rotate** button allows graphical and bitmap objects to be rotated. With an object selected, clicking the button once opens the Rotate dialog box. Set the desired angle and click the **OK** pushbutton to rotate the object.



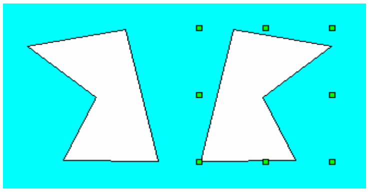
The following example illustrates the state of an object before and after clicking the **Rotate** button and defining a 10° rotation with the object selected:



Horizontal Mirror

The **Mirror Horizontal** button on the Control Bar allows graphical and bitmap objects to be mirrored in the horizontal plane. With an object selected, clicking the button once "flips" the object producing a horizontal mirror image.

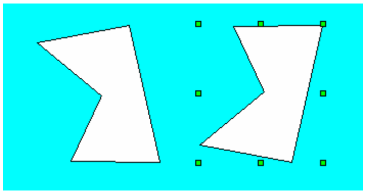
The following example illustrates the state of an object before and after clicking the **Mirror Horizontal** button with the object selected:



Vertical Mirror

The **Mirror Vertical** button allows graphical and bitmap objects to be mirrored in the vertical plane. With an object selected, clicking the button once "flips" the object producing a vertical mirror image.

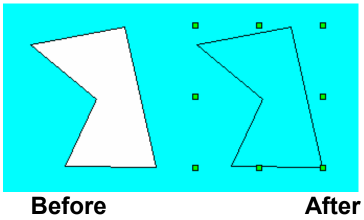
The following example illustrates the state of an object before and after clicking the **Mirror Vertical** button with the object selected:



Transparency On/Off

The **Transparency** button changes an object from solid to outline and from outline to solid. With an object selected, click the button and the object toggles from solid to outline (or vice-versa).

The following example illustrates the state of an object before and after clicking the **Transparency** button with the object selected:

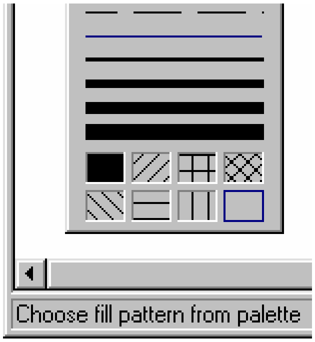


Status Bar

CX-Supervisor provides help and status information in a Status Bar located at the bottom of the main window.

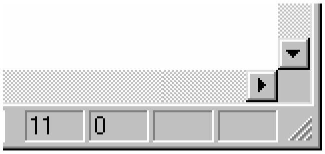
The Status Bar has two main active areas: the Help Message area, and the Cursor Co-ordinates area.

The Help Message area is used by CX-Supervisor to display helpful information concerning menu selections and controls. It is located at the left hand end of the Status Bar, and can display messages similar to the following:



Choose fill pattern from palette

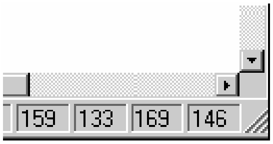
The Cursor Co-ordinates area provides position information for the current location of the mouse pointer as it tracks around the active page within the CX-Supervisor window. If an object on a page is selected, the status area displays the co-ordinates of the location of that object. An example of the system status area is illustrated as follows:



11 0

The first two display panels show the current x and y co-ordinates of the mouse pointer within the currently active page. As the mouse pointer leaves the confines of a page and moves into the client area of the main CX-Supervisor window, the last recorded co-ordinates are held in these panels.

If an object is clicked on, all four panels are used by CX-Supervisor to display the co-ordinates of that object, as follows:



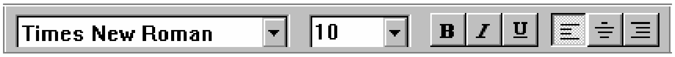
159 133 169 146

x y Width Height

The first two panels contain the distance of the top left corner of the object from the top left of the page in pixels. The second two panels show the extent of the object in pixels.

Text Bar

An illustration of the CX-Supervisor Text Bar is as follows:

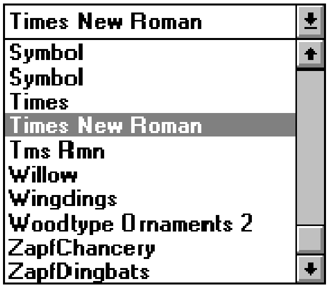


Font Name

The *Font Name* field contains a list of fonts which are available on the host version of Windows. The list is presented in alphabetical order, and only details typeface families. This does not include bold or italic variants as this attribute is set by using the appropriate button.

The following illustration shows the control in de-selected state:

The following illustration shows the control in selected state:



The *Font Name* field displays the current font attribute for the selected object(s). To change the font for a block of text, highlight it and click either: the down arrow adjacent to the edit part of the control, or into the edit part of the control itself, for the *Font Name* field. The control then displays a list of available fonts, from which the desired typeface may be selected.

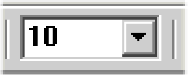
If more than one block of text is selected with each having different font attributes, the edit part of the control is empty. However, selecting a font from the supplied list still sets the font attribute for all the selected group of text blocks.

Fonts and font families are printer dependent, therefore changing the printer within the Printer Setup dialog (accessed from the *File* menu) changes the fonts which are available within the *Font Name* field.

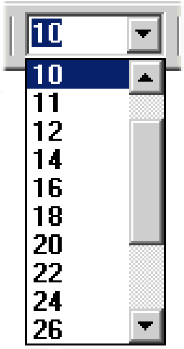
Font Size

The *Font Size* field contains a list of point sizes available on the host version of Windows for the selected font. The list is presented in numerical order.

The following illustration shows the control in de-selected state:



The following illustration shows the control in selected state:



Either click the cursor into the editable area of the control and enter a new point size, or click the down arrow and select a new point size from the presented list.

The control displays the current point size for the selected block of text. If more than one block of text is selected having different point sizes, the edit part of the control is empty. However, selecting a font from the supplied list (or typing in a new point size) still sets the point size for all the selected blocks of text.

Text Bold On/Off

The **Text Bold** button allows any selected text objects to be set to bold when the **Text Bold** button is pushed in, or normal when the **Text Bold** button is released. Text Bold only applies to text, block text, and to text on buttons, other controls and graphics cannot be emboldened.

Text Italic On/Off

The **Text Italic** button allows any selected text object(s) to be set to italic when the **Text Italic** button is pushed in, or normal when the **Text Italic** button is released. Text Italic only applies to text, block text, and to text on buttons, other controls and graphics cannot be italicised.

Text Underline On/Off

The **Text Underline** button allows any selected text object(s) to be set to underline when the **Text Underline** button is pushed in, or normal when the **Text Underline** button is released. Text Underline only applies to text, block text, and to text on buttons, other controls and graphics cannot be underlined.

Text Left Aligned

The **Text Left Aligned** button aligns any selected text object(s) to the left edge of the bounding box. Text Left Justified only applies to text, block text, and to text on buttons, other controls and graphics cannot be left justified.

Text Centred

The **Text Centred** button aligns any selected text object(s) to the centre of the bounding box. Text Centre Justified only applies to text, block text, and to text on buttons, other controls and graphics cannot be centre justified.

Text Right Aligned

The **Text Right Aligned** button aligns any selected text object(s) to the right edge of the bounding box. Text Right Justified only applies to text, block text, and to text on buttons, other controls and graphics cannot be right justified.

Grid

The use of the grid may be helpful in drawing and aligning the objects on the screen. Select the Grid button from the toolbar to enable the grid.

The grid settings are available by selecting *Grid* from the *View* menu. All the available grid sizes are shown along with a *Snap to Grid* option which forces graphical objects selected by the user to align according to the current grid setting.

Note: The toolbar buttons can be customised to remove or add favourite functions.

Tip of the Day

On startup, a Tip of the Day dialog is displayed. All tips can be reviewed using the Next and Previous buttons. This dialog can be turned off, or turned back on from the Help menu.

CHAPTER 2
Pages

This chapter explains the concept of pages. The chapter covers creating, amending, printing and saving pages.

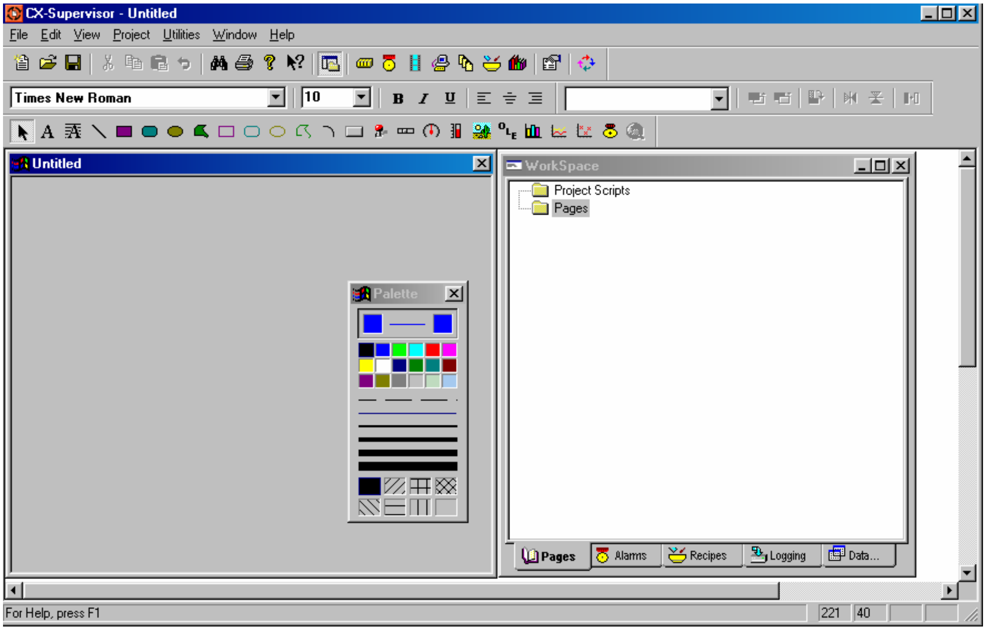
Creating a Page

A project may consist of several pages, but must contain at least one.

To create a new page, CX-Supervisor must currently have a project open. If no project is currently open, either click on the **Open** button from the toolbar to open a previously saved project, or select *New* from the *Project* menu to create a new project.

Note: When you first create a project, CX-Supervisor creates a new page for you automatically. For more details concerning projects, refer to chapter 6, Projects.

With a project open, click the **New Page** button from the toolbar to create a new blank page. An example of a new blank page is shown as follows:



Amending a Page

To amend a page, it must first be open. If no pages are open click the **Open Page** button from the toolbar. A standard File Open dialog with the caption Open Page is displayed. The layout and usage depends on your operating system so see your Microsoft documentation for details.

1, 2, 3…

1. Locate the drive and directory where the desired page is stored.
2. Select the desired page from the list presented.
3. Click the **Open** pushbutton to load the page.

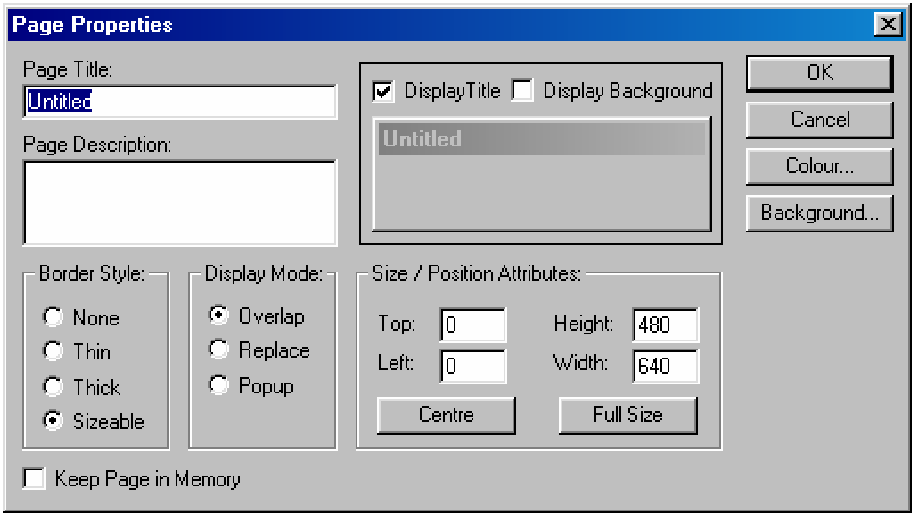
Note: Loading a page does not automatically make it part of a project. It is therefore perfectly feasible to load and edit pages from other projects. Use the Project Editor to attach a page to a project. For more details concerning projects, refer to chapter 6, Projects.

The loaded page may now be edited as required using the CX-Supervisor editing tools.

Defining the Properties of a Page

A page has certain attributes, or properties. These properties may be viewed and edited in two ways. The simplest way to access the properties of a page is to double click with the left mouse button in the background area of the page.

This causes CX-Supervisor to display the following Page Properties dialog:



Page Properties

Page Title: Untitled

Page Description:

☑ DisplayTitle ☐ Display Background

Border Style: ○ None ○ Thin ○ Thick ● Sizeable

Display Mode: ● Overlap ○ Replace ○ Popup

Size / Position Attributes:

Top: 0 Height: 480

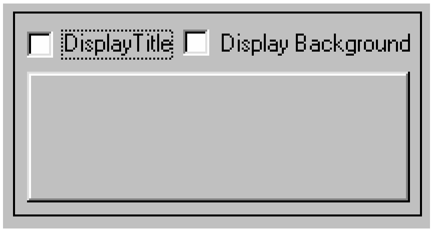
Left: 0 Width: 640

☐ Keep Page in Memory

The Page Properties dialog allows the viewing and editing of various attributes.

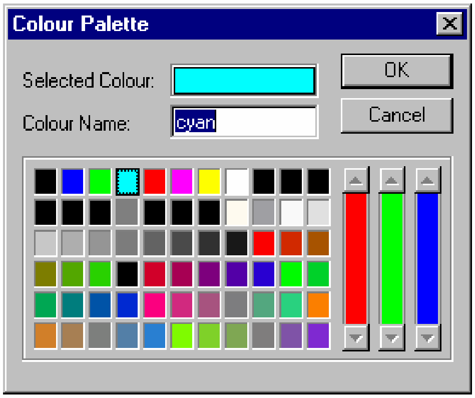
1, 2, 3…

1. Enter a title for the page in the *Page Title:* field up to a maximum of 32 characters.
2. Add a description (if required) in the *Page Description* field.
3. Enter the co-ordinates for the top-left corner of the page in the *Top* and *Left* field. This value must be in pixels, and must be a positive integer between 0 and 2000. Alternatively, click on the **Full Size** pushbutton to fill the runtime environment workspace.
4. Enter the height and width measurements for the page in the *Height* and *Width* fields. These values must be in pixels, and must be a positive integer between 0 and 2000. Click on the **Centre** pushbutton to centre the page to the graphics workspace.
5. Select the border style for the page from the *Border Style* settings. The default for this is *Sizeable*, as shown on the Page Properties dialog. Changing the selection will update the shown preview.
6. To prevent the title from being displayed, click the *Display Title* setting to remove the check mark. The following change occurs in the dialog:



☐ DisplayTitle ☐ Display Background

7. To change the display mode, select *Overlap, Replace* or *Popup* in the *Display Mode* setting. ‘Popup’ pages appear above all other page types, ‘Overlap’ pages can lie on top of other pages, and ‘Replace’ pages closes any pages that overlap.
8. To alter the colour of the page, click in the *Demo Page Window* area or click on the **Colour** pushbutton. CX-Supervisor displays the Colour Palette dialog:



Selected Colour:

Colour Name: cyan

9. Either select a colour from the palette area or click on a palette colour and edit it using the three colour sliders. When the desired colour is displayed, click the **OK** pushbutton to return to the Page Properties dialog.

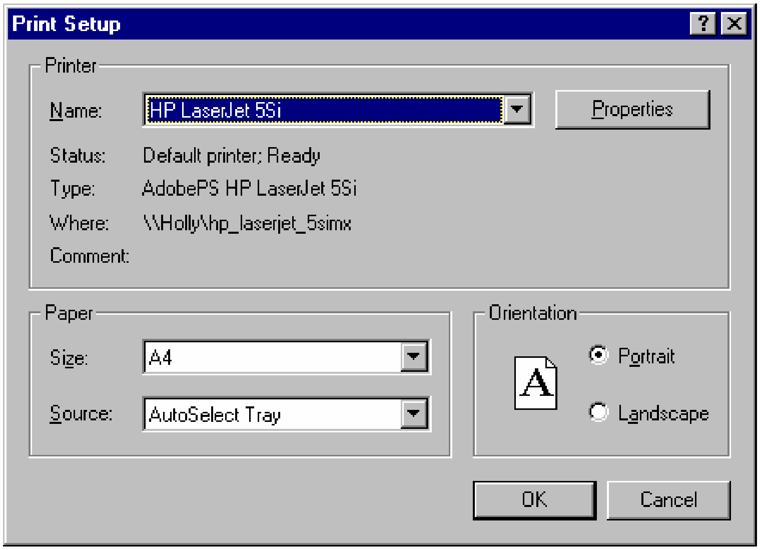
10. Alternatively, to display an image or photograph as the background click the **Background** pushbutton and select the desired file and note the **Display Background** check box is automatically ticked.

Printing a Page

Print Setup

Before printing a page, ensure that the printer has been set up correctly. To check the printer settings, select *Print Setup* from the *File* menu.

CX-Supervisor displays the Print Setup dialog in response:



Print Setup

Printer

Name: HP LaserJet 5Si

Status: Default printer; Ready

Type: AdobePS HP LaserJet 5Si

Where: \\Holly\hp_laserjet_5simx

Comment:

Paper

Size: A4

Source: AutoSelect Tray

Orientation

● Portrait

○ Landscape

The current printer selection is defined in the *Name:* control box. To alter the settings, proceed as follows:

1, 2, 3…

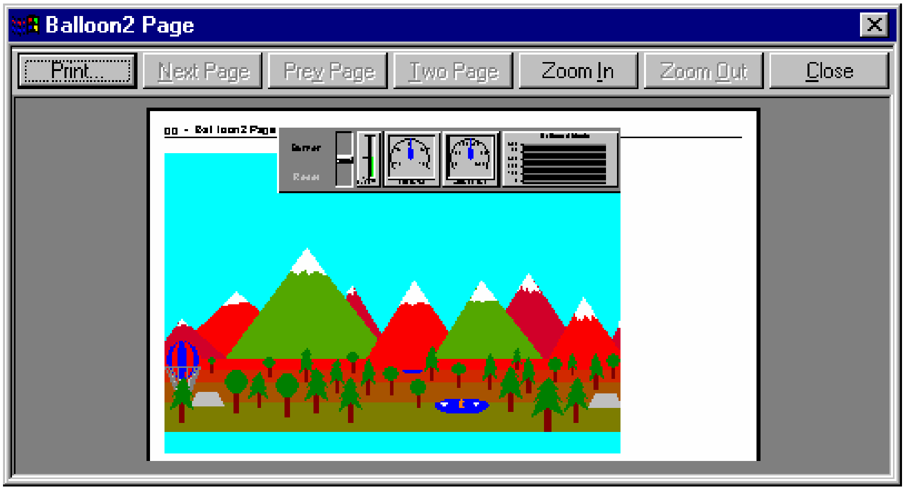
1. To change the current printer selection, click the *Name:* field and select from the list presented.
2. To change the page orientation from portrait to landscape click the *Landscape* setting, (or vice-versa).
3. To change the paper size, click the *Size:* field, and select the desired paper size from the list presented.
4. To change the paper source, click the *Source:* field, and select the desired paper source from the list presented.
5. Click the **OK** pushbutton to exit from the Print Setup dialog when the settings are correct.

Note: The **Properties** pushbutton gives access to advanced printer configuration functions for the selected printer. For details of these functions, refer to the *Windows User Manual*, *On-line Help*, or the appropriate Manufacturer's handbook.

Print Preview

To preview the page before printing, select *Print Preview* from the *File* menu.

This results in a screen display similar to the following:

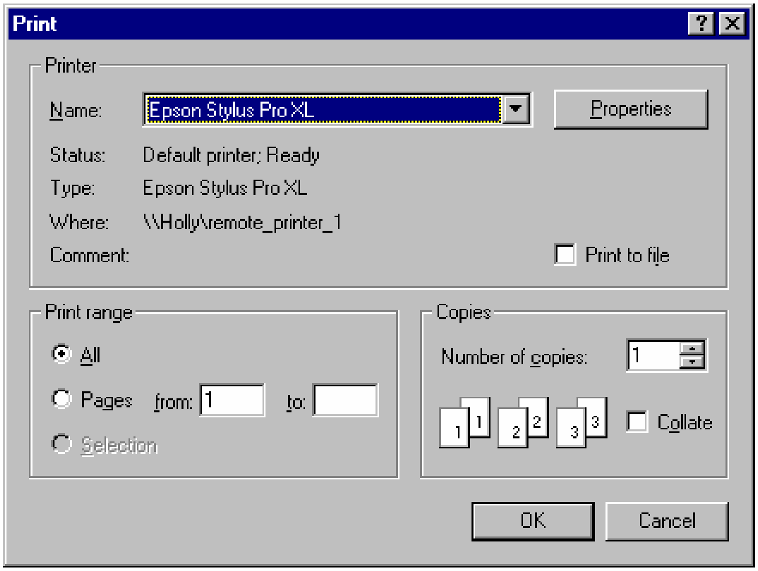


1, 2, 3…

1. To print the page, click the **Print** pushbutton, CX-Supervisor displays the Print dialog.
2. To display the next page, click the **Next Page** pushbutton.
3. To display the previous page, click the **Prev Page** pushbutton.
4. To display the current page, side-by-side with the next page, click the **Two Page** pushbutton. To return to a single page view click the same button again which now carries the legend, **One Page**.
5. Click the **Zoom In** pushbutton. Click it a second time to zoom in to the second level. This does not zoom into a specific area of the page.
6. To zoom out from a zoomed in view, click the **Zoom Out** pushbutton. Click a second time to zoom right out.
7. To close the preview screen, click the **Close** pushbutton.

Printing

To print a page, click the **Print Page** button from the toolbar. The Print dialog is displayed.



Print

Printer

Name: Epson Stylus Pro XL

Status: Default printer; Ready

Type: Epson Stylus Pro XL

Where: \\Holly\remote_printer_1

Comment:

☐ Print to file

Print range

● All

○ Pages from: 1 to:

○ Selection

Copies

Number of copies: 1

☐ Collate

1, 2, 3…

1. Set the print range by clicking one of the settings in the *Print range* settings. The *Selection* setting is only enabled when an item (or items) are selected on the page prior to starting the print process. Click the *Pages* setting and enter a range of pages in the *from* and *to* fields.
2. To change the number of copies to be printed, enter a new value in the *Copies* field.
3. If multiple copies are selected, click the *Collate* setting to instruct the printer to collate the copies as they are printed.
4. To print to file, click the *Print to file* setting.
5. Click the **OK** pushbutton to activate printing. Whilst printing is in progress, a dialog is displayed showing the status of the print job. Clicking the **Cancel** pushbutton at this point aborts the printing. Once printed, the dialog disappears. The final printed version should be similar to that shown in the Print Preview dialog, with a header describing the project, and a footer describing the page with a page number and date stamp.

Note: The Properties pushbutton on the Print dialog gives access to advanced printer configuration functions for the selected printer. For details of these functions, please consult the Windows User Manual, On-line Help, or the appropriate Manufacturer’s handbook.

Saving a Page to a Project

Once a page has been created it is wise to save it into the project in which it is planned to reside. It is good practice to ensure that pages are saved regularly, for example to minimise the loss of work in the event of a power failure.

To save a page click the **Save Page** button from the toolbar. If this is the first time the page has been saved, the Save As dialog is displayed. This is a standard Windows dialog and usage depends on your operating system. Please consult your Microsoft documentation.

1, 2, 3…

1. Move to the location where the page file is to be stored.
2. Ensure that the *Save as Type:* control is set to *CX-Supervisor Pages (*.PAG)*.
3. Enter a name in the *File Name:* field.
4. Click the **Save** pushbutton to save the file.

Subsequent saves do not cause the Save As dialog to be displayed.

After clicking the **Save** pushbutton**,** if the page is currently not assigned to a project, a confirmation dialog is displayed. Alternatively, select *Save Page* from the *File* menu (or use the short-cut key combination of <Ctrl>+S.

Save Page As

Should a copy of a page be required, (for incorporation into another project perhaps), select *Save Page As...* from the *File* menu.

CX-Supervisor displays the File Save As dialog as described previously. After entering a new name for the page, CX-Supervisor prompts to save the different page into the current project as above.

Closing a Page

To close a page after editing select *Close Page* from the *File* menu.

Alternatively, either click on the control box for the page, (located in the top left corner of the page) and select *Close* from the *Control* menu, or simply double-click on the *Control* menu and the page closes.

Should you attempt to close a page which has not been saved, CX-Supervisor displays a confirmation dialog. If you want to save the changes, click the **Yes** pushbutton. Otherwise, click the **No** pushbutton, or click the **Cancel** pushbutton to abort closing the page.

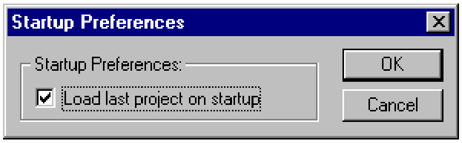
CX-Supervisor Preferences

CX-Supervisor allows a user to customise the working environment. To set or amend the CX-Supervisor setup, select *Preferences* from the *File* menu, followed by the preference to set up.

The types of customisation are described in the following paragraphs.

Startup Preferences

The Startup Preferences allows the project last accessed in the previous session of CX-Supervisor to be automatically loaded in the next session. It is accessible by selecting *Startup* from the *Preferences* menu. Click in the *Load last project on startup* setting to switch this preference as desired. Click the **OK** pushbutton to accept the change, or **Cancel** to abort.



Startup Preferences

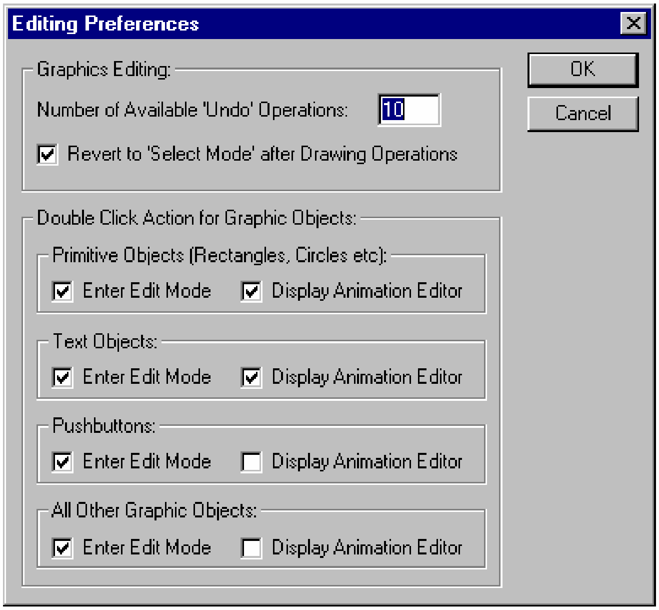
Startup Preferences:

☑ Load last project on startup

Editing Preferences

The Editing Preferences dialog provides a number of switches to enhance the use of the Graphics Editor and CX-Supervisor animation. It is accessed by selecting *Editing* from the *Preferences* menu. The number of undo operations in the Graphics Editor can be set in the *Number of Available 'Undo' Operations:* field. To make the Graphics Editor return to Select Mode after every drawing operation, click in the *Revert to 'Select Mode' after Drawing Operations* setting. The use of double-clicking on an object can be defined: when a check mark is present in the *Enter Edit Mode* field, a double-click on an object of that type causes it to enter Edit Mode.

When a check mark is present in a *Display Animation Editor* field, a double-click on an object of that type activates the Animation Editor. Click the **OK** pushbutton to accept the changes, or the **Cancel** pushbutton to abort.



Editing Preferences

Graphics Editing:

Number of Available 'Undo' Operations: 10

☑ Revert to 'Select Mode' after Drawing Operations

Double Click Action for Graphic Objects:

Primitive Objects (Rectangles, Circles etc):
☑ Enter Edit Mode ☑ Display Animation Editor

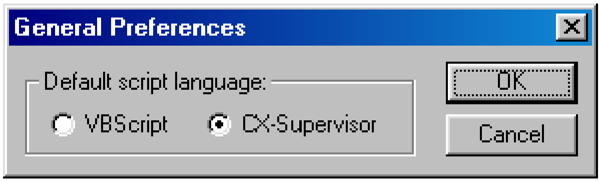
Text Objects:
☑ Enter Edit Mode ☑ Display Animation Editor

Pushbuttons:
☑ Enter Edit Mode □ Display Animation Editor

All Other Graphic Objects:
☑ Enter Edit Mode □ Display Animation Editor

General Preferences

General Preferences allow the default script language to be chosen. When new scripts are added, CX-Supervisor will default to saving scripts in the select language.



General Preferences

Default script language:
○ VBScript ● CX-Supervisor

CHAPTER 3
Points

This chapter describes CX-Supervisor points and the procedures associated with the creation, amendment and removal of points using the Points Editing facility. The application of external sources to points is also described.

What is a Point?

A point is a variable used internally by CX-Supervisor. All points within CX-Supervisor have a name, group and type associated with them.

CX-Supervisor provides a set of pre-defined System ($) Points, which are detailed in chapter 3 Points.

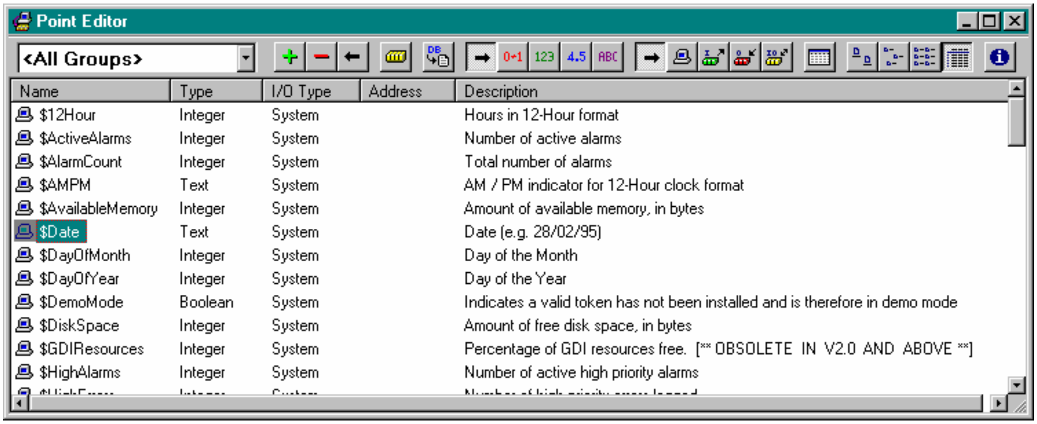
About the Point Editor

The Point Editor allows the viewing, creating, modifying and removing of points from the points database.

Note: System Points cannot be modified.

To open the Point Editor dialog, select the **Point Editor** button from the toolbar.

An example of the Point Editor dialog is as follows:



Point Editor

<All Groups>

Name	Type	I/O Type	Address	Description
$12Hour	Integer	System		Hours in 12-Hour format
$ActiveAlarms	Integer	System		Number of active alarms
$AlarmCount	Integer	System		Total number of alarms
$AMPM	Text	System		AM / PM indicator for 12-Hour clock format
$AvailableMemory	Integer	System		Amount of available memory, in bytes
$Date	Text	System		Date (e.g. 28/02/95)
$DayOfMonth	Integer	System		Day of the Month
$DayOfYear	Integer	System		Day of the Year
$DemoMode	Boolean	System		Indicates a valid token has not been installed and is therefore in demo mode
$DiskSpace	Integer	System		Amount of free disk space, in bytes
$GDIResources	Integer	System		Percentage of GDI resources free. [** OBSOLETE IN V2.0 AND ABOVE **]
$HighAlarms	Integer	System		Number of active high priority alarms
$HighErrors	Integer	System		Number of high priority errors logged

The typeface of the editor can be amended by selecting *Preferences* from the *File* menu. This is especially useful when printing.

Note: To select a range of points, click on a point to mark the start of the range and click again with the <Shift> key down to mark the end.

Note: To individually select more than one point, click whilst holding the <Ctrl> key down.

Note: The short-cut keyboard combinations for Cut, Copy and Paste operations are valid within the Point Editor. Highlight one or more points and type <Ctrl>+X to cut or <Ctrl>+C to copy; use <Ctrl>+V to paste. Since the cut and copy operations store the information in the Windows Clipboard, points may be pasted to another CX-Supervisor application. See also Chapter 3 Quick creation of many points.

Note: If the I/O type ends with '*' and a number, it is an array point with the shown number of elements.

Viewing Points in the Point Editor

The Point Editor view of the points database can be tailored by filtering or sorting the available points.

Filtering the Points in the View by Group

Points are separated into groups. To select a group, pick an entry from the *Group* field.

All points can be displayed by selecting <*All Groups*> from this list.

Filtering the Points in the View by Point Type

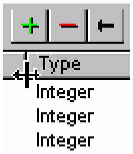
A selection of points can be displayed based on the point type. Selection of the **All Points** button from the toolbar displays points of all types.

The **Boolean Points** button only displays points of a Boolean type. The **Integer Points** button, the **Real Points** button and **Text Points** button, once selected, react in the same way.

Sorting the Points in the View by I/O Type

Points can also be filtered by I/O type. Respectively, these toolbar buttons display All Points, Memory Points, Input Points, Output Points or Input/Output Points.

Sorting the Points in the View



Individual listed points are sorted, based on a designated field type: name, type, input/output type or description. By clicking on the **Name** button, the points are sorted alphanumerically by name. The **Type** button, the **I/O Type** button, the **Address** button and **Description** button, once selected, react in the same way. In the previous Point Editor example the list is sorted by Name.

The widths of the point fields can be widened or narrowed as required using the mouse to drag the column boundaries:

A double click on the text boundaries causes the column to autosize.

Changing the Viewing Mode

Select the **View Settings** button to configure which columns of information are displayed in the Point Editor.

Select the **Large Icons** button to view details with large icons.

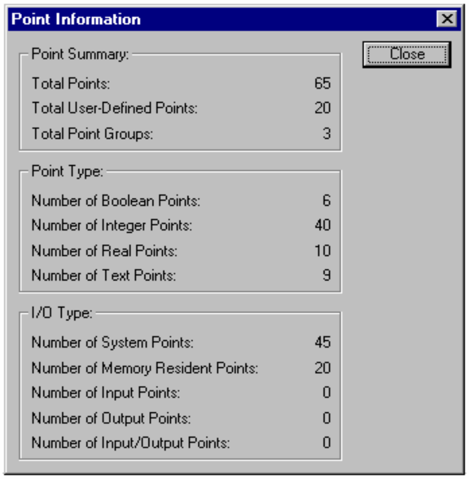
Select the **Small Icons** button to view details with normal icons.

Select the **List** button to view details as a list.

Select the **Details** button to view details as a list including name, type, I/O type, address and description information. The details can be sorted in ascending order by clicking once or in descending order by clicking twice, in the appropriate field.

Summary of Point Information

A summary of point information is available by selecting the **Display Information on Points** button from the toolbar. The resultant Point Information dialog shows an overall summary, a breakdown on the number of points per type and the input/output type. To exit the dialog, click the **Close** pushbutton. The Point Information dialog is shown as follows:



Point Information

Point Summary:

Total Points: 65
Total User-Defined Points: 20
Total Point Groups: 3

Point Type:

Number of Boolean Points: 6
Number of Integer Points: 40
Number of Real Points: 10
Number of Text Points: 9

I/O Type:

Number of System Points: 45
Number of Memory Resident Points: 20
Number of Input Points: 0
Number of Output Points: 0
Number of Input/Output Points: 0

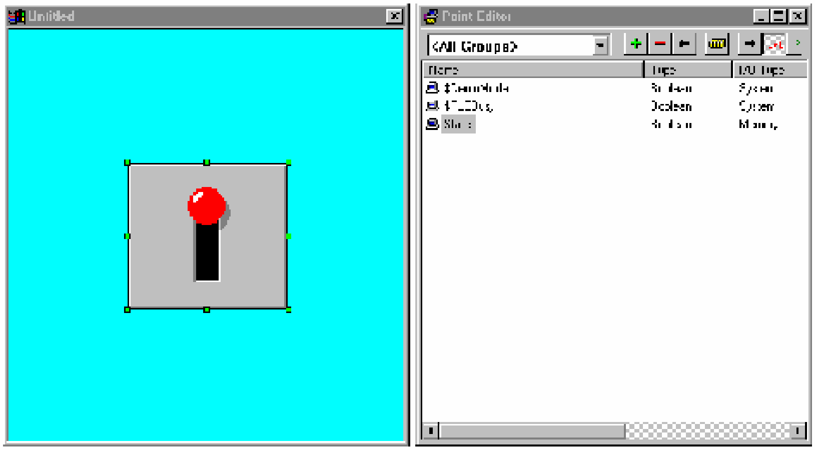
Drag and Drop of Points onto Control Objects

The Point Editor can be used to drag and drop points onto control objects. For instance, Integer points can be dragged from the Point Editor onto a Linear Gauge, Rotary Gauge, Trend Graph and Slider. Boolean points can also be applied to a Toggle.

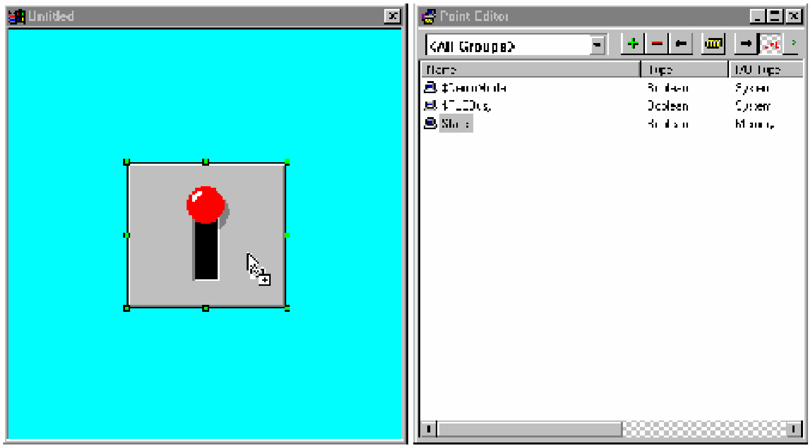
To apply a point to a control object with the Point Editor:

1, 2, 3…

1. Arrange the CX-Supervisor windows so that the point to drag and the target control object are both visible.



2. Select the point to associate and drag it onto the page.

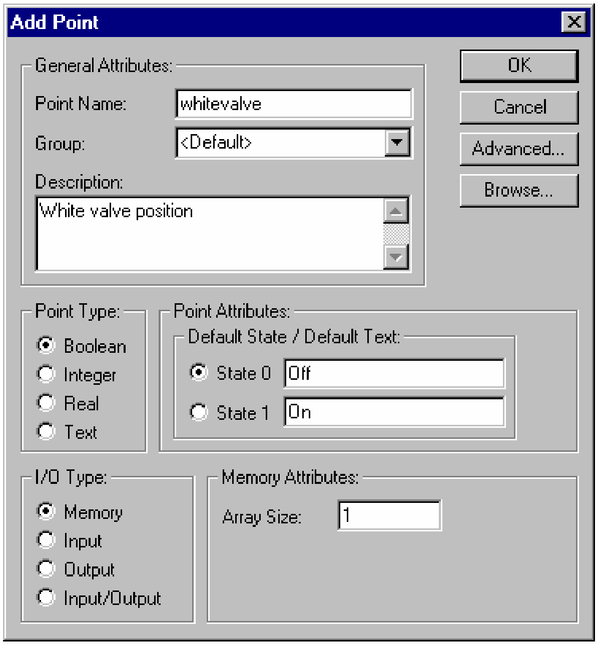


3. Drop the point at the position of the target control object. An invalid “drop” is denoted by the mouse pointer changing to a circular symbol.

For further information of the use of points with objects, refer to chapter 4, Objects.

Creating a Point

To add a new point, select the **Add Point** button from the toolbar. This results in the Add Point dialog being displayed:


Add Point

General Attributes:

Point Name: whitevalve

Group: <Default>

Description: White valve position

Point Type: ● Boolean ○ Integer ○ Real ○ Text

Point Attributes:

Default State / Default Text:

● State 0 Off

○ State 1 On

I/O Type: ● Memory ○ Input ○ Output ○ Input/Output

Memory Attributes:

Array Size: 1

Once all the information has been provided for the new point, clicking the **OK** pushbutton commits the new point to the points database, whilst the **Cancel** pushbutton aborts the add operation.

General Attributes

The name of the point is entered in the *Point Name:* field. The point name can be up to 20 alphanumeric characters, and must not begin with a digit or include mathematical operators such as "+", or be identical to a script reserved word such as "IF" or "cos". Any invalid characters (including spaces) generate an audible error, or an *"Invalid Point Name entered"* message when the **OK** pushbutton is clicked.

The group to which this point belongs is selected from the *Group:* field. A new group can be entered by typing in the *Group:* field.

A points description, detailing the use of the current point, may be inserted in the *Description:* field.

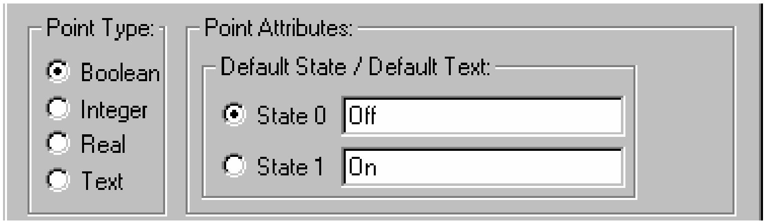
Point Type

The point type can be Boolean, Integer, Real or Text. The default type is *Boolean*.

Point Attributes

The attributes for a point vary according to the point type.

For a Boolean point, the following attributes are displayed:

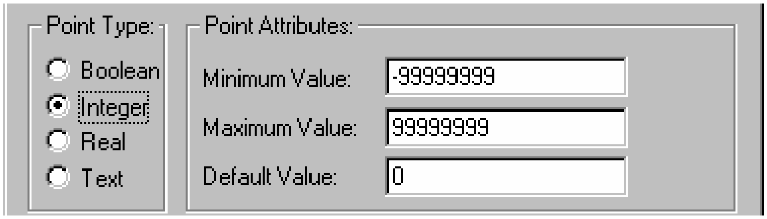


Point Type:
● Boolean
○ Integer
○ Real
○ Text

Point Attributes:
Default State / Default Text:
● State 0 Off
○ State 1 On

The default state of the point is defined by selecting either the *State 0:* or *State 1:* setting. Associative text (such as *'OFF'* or *'FALSE'* for state 0) can be applied in the related *Default Text:* fields. The default text is associated with Toggle buttons and object animations such as Display Value and Edit Point Value when they are configured to use the point.

For an Integer point, the following attributes are displayed:



Point Type:
○ Boolean
● Integer
○ Real
○ Text

Point Attributes:
Minimum Value: -99999999
Maximum Value: 99999999
Default Value: 0

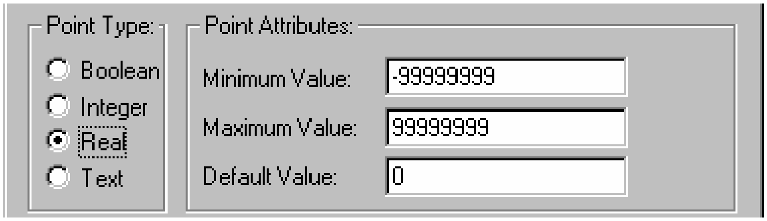
The minimum threshold for the point is inserted into the *Minimum Value:* field.

The maximum threshold for the point is inserted into the *Maximum Value:* field.

The default value is inserted in the *Default Value:* field.

Note: Integer points are limited to -2,147,483,648 to 2147483647

For a Real point, the following attributes are displayed:



Point Type:
○ Boolean
○ Integer
● Real
○ Text

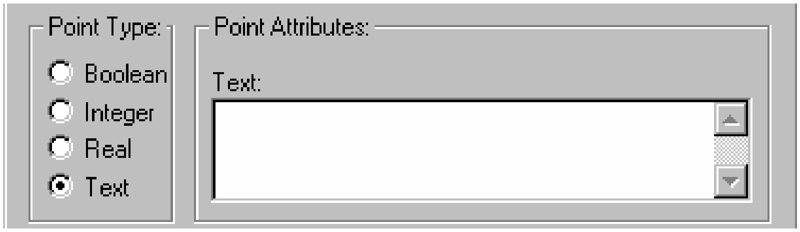
Point Attributes:
Minimum Value: -99999999
Maximum Value: 99999999
Default Value: 0

The minimum threshold for the point is inserted into the *Minimum Value:* field. The maximum threshold for the point is inserted into the *Maximum Value:* field.

The default value is inserted in the *Default Value:* field.

Note: Real points are limited to 1.7E +/- 308 to a precision of 15 digits.

For a Text point, the following attribute is displayed:



Point Type:

○ Boolean

○ Integer

○ Real

● Text

Point Attributes:

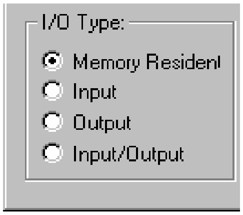
Text:

Text is entered in the *Text:* field.

Note: Memory Text points can have 32768 characters but PLC I/O Text Points are limited to 1024 characters.

I/O Type

The I/O type states the scope of the point, i.e. whether it is purely an internal variable or whether it communicates with a PLC.



I/O Type:

● Memory Resident

○ Input

○ Output

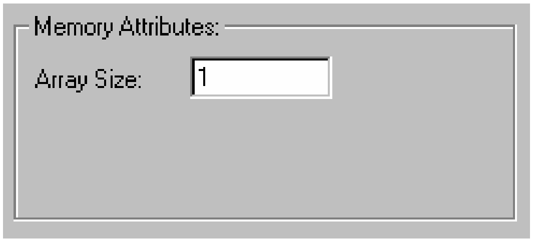
○ Input/Output

- A Memory Resident point is provided internally by CX-Supervisor.
- An Input point receives data from an external device.
- An Output point sends data to an external device.
- An Input/Output point both sends data to and receives data from an external device.

Memory Attributes

The *Array Size* field allows Memory Point arrays to be created. If an Array Size of 1 is specified, a single point is created. Specifying any other value creates an array of points of this type.

This option is only available to Memory Resident points.



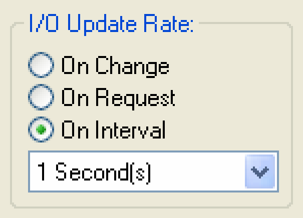
Memory Attributes:

Array Size: 1

I/O Update Rate

The I/O Update Rate specifies how and when communication with the PLC takes place.

This option is not available to Memory Resident points.



I/O Update Rate:

○ On Change
○ On Request
● On Interval

1 Second(s)

The **On Change** option specifies that communication with the PLC occurs automatically as a result of a value change.

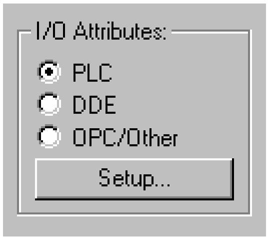
The **On Request** option specifies that data updates with the PLC will be manually requested using the scripting commands InputPoint and OutputPoint. The point is updated internally but only communicates with the PLC when requested to by the script commands.

The **On Interval** option specifies the frequency that communications occur with the PLC. When this option is selected an list box appears allowing the frequency to be entered.

I/O Attributes

The I/O Attributes option specifies the external source or destination for the point.

This option is not available to Memory Resident points.

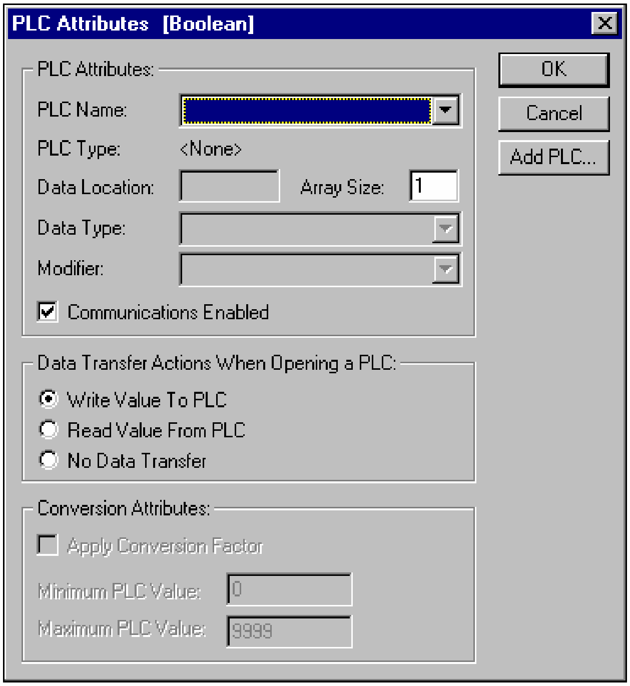


I/O Attributes:

● PLC
○ DDE
○ OPC/Other

The external source is defined by selecting the appropriate *I/O Attributes:* setting. Further configuration of the external source can be applied by clicking the **Setup** pushbutton.

On clicking the **Setup** pushbutton for a PLC external source, the PLC Attributes dialog is displayed.



PLC Attributes [Boolean]

PLC Attributes:

PLC Name:

PLC Type: <None>

Data Location: Array Size: 1

Data Type:

Modifier:

☑ Communications Enabled

Data Transfer Actions When Opening a PLC:

● Write Value To PLC

○ Read Value From PLC

○ No Data Transfer

Conversion Attributes:

□ Apply Conversion Factor

Minimum PLC Value: 0

Maximum PLC Value: 9999

The point type associated with the PLC Connection attributes is shown in the title bar. The required PLC can be selected from the *PLC:* field. If no PLCs exist in the *PLC:* field then one must be added. This is achieved by clicking the **Add PLC** pushbutton and configuring one. A point cannot be configured to have a PLC connection unless all the PLC connection attributes are correctly configured. For more information on adding and configuring PLCs and devices see chapter 6 Projects, Device Configuration.

PLC Attributes

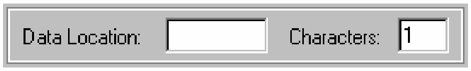
The *Data Location:* field identifies the area of memory to which this address applies and is dependent upon the type and configuration of the PLC.

The *Data Type:* field identifies the type of data held at this address. The type determines how point values are converted from a computer format into a PLC format. The field contains options applicable to the Data Location.

The *Modifier:* field indicates the command to be performed on the data at this address. An example of a modifier is for a bit: a modifier might declare that a bit is to be forced set and not to be written normally. The field contains options applicable to the Data Location.

The *Array Size:* field specifies the number of data values associated with the point. A value greater than 1 allows the point to be treated as an array. For more detail on array points and their uses see Optimisation of PLC Communications.

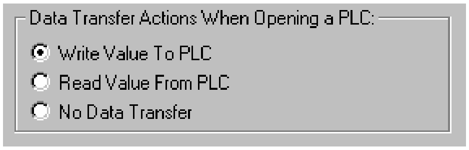
The PLC Attributes dialog takes on a slightly different appearance if the point type is text, showing the number of characters which start at the specified data location:



Data Location: Characters: 1

Data Transfer Actions When Opening a PLC

The type of data transfer action for the selected PLC can be specified by selection of the appropriate setting. Options for Input and Input/Output points are *Always Update Point Value* and *Only Update Point Value When On Display*. Options for an Output point are *Write Value, Read Value* and *No Data Transfer.*



Data Transfer Actions When Opening a PLC:

● Write Value To PLC
○ Read Value From PLC
○ No Data Transfer

Conversion Attributes

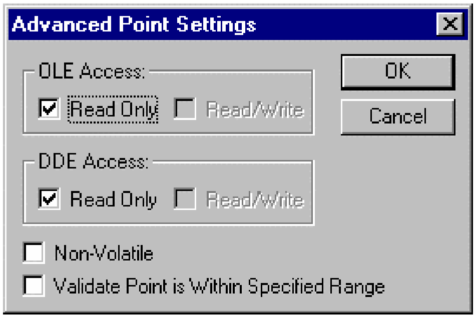
The minimum and maximum PLC value and the application of a conversion factor is specified in the *Conversion Attributes:* fields (these fields are not applicable for Boolean and text points).

Conversion Attributes can be used to convert in a linear fashion between a value in a PLC and the point range. For example, if the possible range in the PLC is 0 to 1000, and the point range is 0 to 100, then a PLC value of 500 would correspond to a point value of 50. The conversion would be performed just before the data is sent to the PLC, or immediately upon receipt of it from the PLC.

On completion of the PLC configuration connection, click the **OK** pushbutton to continue, or the **Cancel** pushbutton to abort. Click the **Add PLC** pushbutton to create a new PLC connection. Information relating to the selection of this pushbutton is described in chapter 6 Projects, Device Configuration.

Advanced Point Settings

Advanced settings can be applied to a new point, by clicking the **Advanced** pushbutton in the Add Point dialog. This results in the Advanced Point Settings dialog being displayed.



Advanced Point Settings

OLE Access:
☑ Read Only ☐ Read/Write

DDE Access:
☑ Read Only ☐ Read/Write

☐ Non-Volatile
☐ Validate Point is Within Specified Range

In order to access a CX-Supervisor point value via OLE2 Automation, it must be given OLE Read or OLE Write access. The scope of the access can be defined by checking the appropriate setting in the *OLE Access.*

The point value which is stored to disk may be Volatile or Non-volatile by selecting or deselecting the *Non-Volatile* check-box. A Non-volatile point ensures the preservation of the point's value at regular intervals. If power is lost, or CX-Supervisor is shut down for any reason, then when the application is restarted the point is initialised to the last saved value. These values are saved in a project file with .NVP extension. To prevent unexpected operation, this file is automatically deleted if further point changes are made in the Developer. In this case points revert to their Default value.

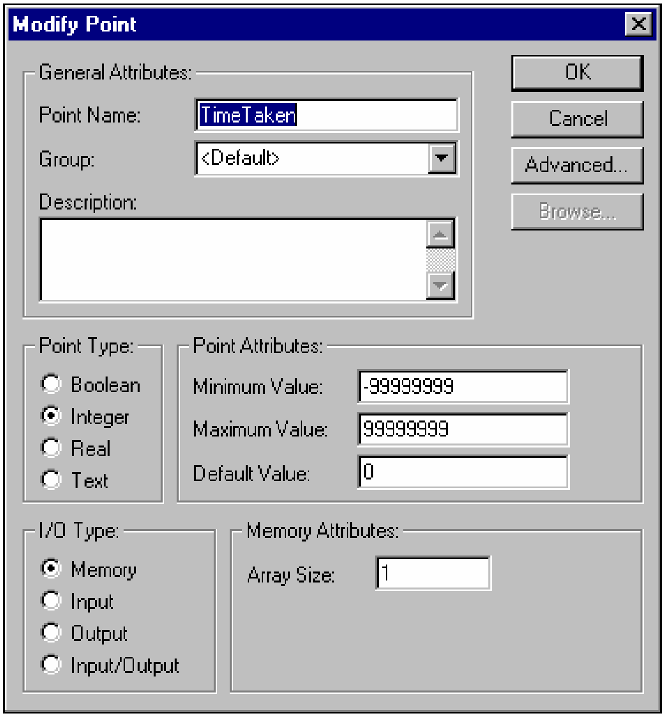
Select the *Validate Point is Within Specified Range* check-box. This option is only available for Input or I/O points of type Integer or Real. When checked, an error message is displayed in the error log if the data passed to CX-Supervisor is outside of the specified Minimum and Maximum range.

Click the **OK** pushbutton to accept the advanced settings, or the **Cancel** pushbutton to abort the operation.

Amending an Existing Point

⬅ To modify an existing point, highlight the point from the points list and click the **Modify Point** button from the toolbar.

This results in the View Point dialog being displayed as shown below, a dialog based on the Add Point dialog:



Modify Point

General Attributes:

Point Name: TimeTaken

Group: <Default>

Description:

Point Type:
- ○ Boolean
- ● Integer
- ○ Real
- ○ Text

Point Attributes:

Minimum Value: -99999999

Maximum Value: 99999999

Default Value: 0

I/O Type:
- ● Memory
- ○ Input
- ○ Output
- ○ Input/Output

Memory Attributes:

Array Size: 1

The selected point can be redefined as described in chapter 3, Creating a Point.

Note: If a point is renamed then any animations or scripts that were using the previous point name become invalid. Run the Analyse Application tool to check for References to non-existent points.

Deleting an Existing Point

To remove an existing point, highlight the point from the points list and click the **Delete Point** button from the toolbar. This results in a confirmation dialog being displayed. Click the **Yes** pushbutton to remove the point from the points database, or the **No** pushbutton to abort the delete operation.

Note: It is possible to delete more than one point by either selecting points within a range or by selecting individually several points.

To select a range click on a point to mark the start of the range and click again with the <SHIFT> key down to mark the end.

To individually select more than one point click whilst holding the <CTRL> key down.

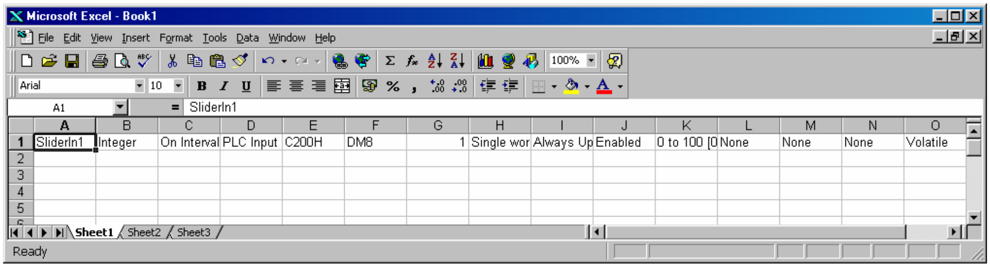
Note: If a point is deleted then any animations or scripts that were using the point become invalid. Run the Analyse Application tool to check for References to non-existent points.

Quick creation of many points

CX-Supervisor supports up to 8,000 points, which could take much time to create. Within the Point Editor you can use Copy/Paste which helpfully increments the point name automatically (BoilerTemp1 becomes BoilerTemp2, BoilerTemp3 etc) but you still have to edit each point to configure the PLC address and possibly other information. Using the Copy and Paste functions, Point data can quickly be edited and created for example in grid form using Excel. This can prove extremely quick when the design uses contiguous addresses.

1, 2, 3…

1. Open the Point Editor.
2. Select the points to edit, or a single point to act as a template and Copy to the Windows Clipboard
3. Start Excel.
4. Position the cursor in column A and paste the CX-Supervisor Points into Excel. The worksheet should resemble the following:



A1 = SliderIn1

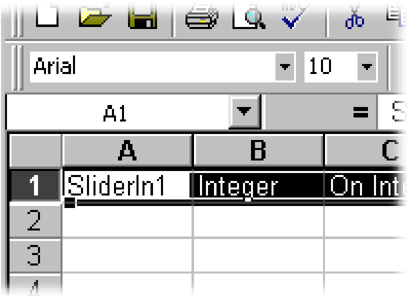
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	SliderIn1	Integer	On Interval	PLC Input	C200H	DM8	1	Single wor	Always Up	Enabled	0 to 100 [0	None	None	None	Volatile
2															
3															
4															
5															

Each row is a single point, and each column is a setting of that point.

Note: Some settings may be hidden on the right. Use the scrollbar to view.

5. Edit the point details as required.

Note: To quickly create new points with the same settings, select the whole row by clicking the row number on the left. A range can now be drawn to be filled by dragging the black square box on the bottom left of the range selection:



	A	B	C
1	SliderIn1	Integer	On Int
2			
3			

Excel will automatically increment any data ending in a number. This can be very useful for Names and PLC Addresses (columns A and F) but take care with other columns. If other columns are wrongly incremented, like PLC Name, Array size and Data Range (columns E, G and K) these can quickly be copied by selecting the correct value e.g. G1 then using the square box to highlight the column of data to fill. To see the fill options hold down the right mouse button before draging the square box. Repeat for each column.

6. When editing is completed, select the row(s) to required and select Copy

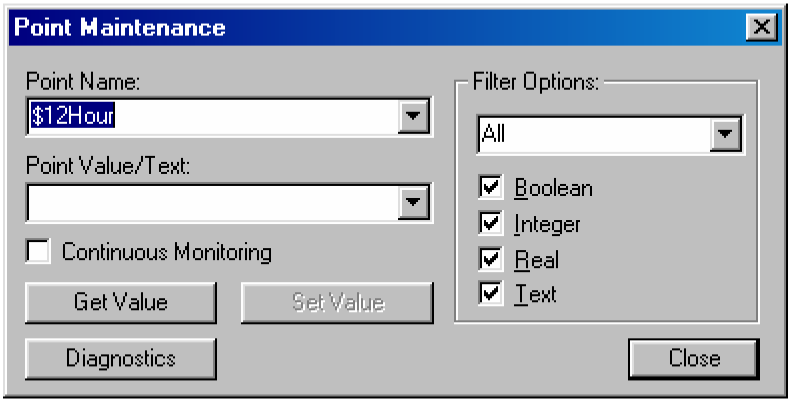
7. Switch back to the Point Editor and select Paste.

Note: If the pasted point names already exist, CX-Supervisor will automatically add or increment a number on the end to prevent overwriting. If you wish to overwrite to replace old point settings, simple delete (not Cut) the old points from the Point Editor before pasting.

Note: All pasted points will pasted into the group on view

Runtime Point Maintenance

It is possible to reconfigure points at runtime via the right mouse button floating menu option *Points Maintenance*. The Point Maintenance dialog is displayed.



Point Name: $12Hour

Point Value/Text:

☐ Continuous Monitoring

Filter Options: All

☑ Boolean
☑ Integer
☑ Real
☑ Text

To monitor and change the value of a point in runtime, select a point from the *Point Name:* field. The *Filter Options:* field and settings refines the points listed in the *Point Name:* field. Select the **Get Value** pushbutton to retrieve the current value of the selected point. Specify a new point value in the *Point Value/Text:* field and click the **Set Value** pushbutton. The **Diagnostics** button shows details including communications statistics, useful for diagnosing communication problems. Note this button is only available when a user with 'Designer' privileges is logged in. Select the **Close** pushbutton to complete the operation.

Optimisation of PLC Communications

PLC communication speeds can be increased by creating "array" points which contain several elements of the same type, rather than creating a large number of individual points. For instance, the time taken to update an array point containing 50 elements are quicker than the time taken to update 50 individual points.

Creation of an "Array" Point

An "array" point can be created by specifying a value greater than 1 in the *Elements:* field of the Setup PLC Connection dialog. The *Data Location* field specifies the memory address from which the array of data begins.

Note: Elements of an array point are located at consecutive addresses after the address specified in the Data Location field.

Accessing Elements of an "Array" Point

Direct access to array points can be achieved by applying a subscript to the pointname, e.g. pointname[index]. For more information refer to the *CX-Supervisor Script Language Reference Manual.*

Point Import

To import PLC points from other applications, click the **Import PLC Points** button from the toolbar. This results in the Import PLC Points From Another CX-Server Project dialog being displayed.

The Point Import tool can be used to import point information into the CX-Supervisor project that has already been configured. The symbol name, symbol type and PLC address is imported from the CDM file generated by other applications.

The CDM file can be generated from CX-Programmer by linking the project to the CDM file. The CDM file can also be generated by exporting from a SYSWIN project to a CDM file. Refer to the documentation supplied with the package for information on how to export or link the data to the CDM file.

It is possible to import points from the CDM file of another CX-Supervisor project but this is not recommended, as only the name, type and address are imported. A better method is to copy the points from one application to the clipboard, and paste them in the required application. This way all point information is copied.

Steps to import from another CDM file.

1) Open the Import PLC Points From Another CX-Server Project dialogue
2) Press Open Project and find the project to import from
3) Select the tab depending upon the I/O type required
4) Click the Add button and select the symbols required
5) Add these to the CX-Supervisor project by clicking the Add button
6) Press OK to return
7) Repeat for other I/O types
8) Press OK to finish

System Points

System Points are those points that are pre-defined within CX-Supervisor. They cannot be edited or deleted, but their attributes can be viewed. All system points can be selected from the System Point dialog.

System points are listed in the points list, and are denoted by a '$' symbol preceding the point name. To view system points only, select *System Points* from the *Group:* field. Once listed, the **Boolean Points, Integer Points, Real Points, Text Points** and **All Points** pushbuttons on the toolbar are unavailable for selection. To view other points, select *All Groups* from the *Group:* field.

Time Points

The following table describes system points for use with time based operations. Provisions are given to both 12 hour and 24 hour time formats.

System point	Point type	Point range	Remarks
$12Hour	Integer	0-12	Hours in 12-hour format.
$AMPM	Text	—	AM/PM indicator for 12-hour clock form.
$Hour	Integer	0-23	Hours in 24-hour format.
$Millisecond	Integer	0-999	Number of milliseconds.
$Minute	Integer	0-59	Minutes.

$Second	Integer	0-59	Seconds.
$Time	Text	—	Time (e.g. 09:46).

Date Points

The following table describes system points for use with date based operations. Provisions are given for numerical and alphanumerical formats.

System point	Point type	Point range	Remarks
$Date	Text	—	Date (e.g. 28/02/95).
$DayOfMonth	Integer	1-31	Day of the month.
$DayOfYear	Integer	1-366	Day of the year.
$Month	Integer	1-12	Month (1 - January, 12 – Dec.).
$MonthName	Text	—	Month name (e.g. February).
$ShortMonthName	Text	—	Abbreviated month name (e.g. Feb).
$ShortWeekDayName	Text	—	Abbreviated weekday name (e.g. Wed).
$ShortYear	Integer	0-99	Abbreviated year (e.g. 95).
$WeekDay	Integer	0-6	Day of the week (0 - Sun, 6 – Sat).
$WeekDayName	Text	—	Weekday name (e.g. Wednesday).
$WeekOfYear	Integer	0-51	Week number for the year.
$Year	Integer	1970-2038	Year (e.g. 1995).

Internal Points

The following table describes system points for use with interrogating current system settings, such as memory and disk space restrictions and other system resources.

System point	Point type	Point range	Remarks
$AvailableMemory	Integer	0-2,147,483,647	Amount of available memory, in bytes.
$DemoMode	Boolean	—	Indicates whether a valid USB Dongle has been installed.
$DiskSpace	Integer	0-2,147,483,647	Amount of free disk space available, in bytes.
$PCName	Text	-	Computer name of PC as seen in Explorer
$ProjectName	Text	-	Name of project file, without .SCS extension
$ProjectPath	Text	-	Path to project on disk, without project

			name.
$SpoolCount	Integer	0-2,147,483,647	Number of alarm entries currently spooled, awaiting printing to page printer.
$Version	Text	-	Version number of CX-Supervisor Runtime program

Display Points

The following table describes system points for use with the display mode.

System point	Point type	Point range	Remarks
$ScreenSizeX	Integer	0-2,147,483,647	Screen width.
$ScreenSizeY	Integer	0-2,147,483,647	Screen height.

Mouse Points

The following table describes system points for use in mouse movement and operation. They are updated on a left button click.

System point	Point type	Point range	Remarks
$MouseX	Integer	0-65535	Mouse X co-ordinates.
$MouseY	Integer	0-65535	Mouse Y co-ordinates.

Alarm Points

The following table describes system points for use with CX-Supervisor alarms.

System point	Point type	Point range	Remarks
$ActiveAlarms	Integer	0-2,147,483,647	Number of active alarms.
$AlarmCount	Integer	0-2,147,483,647	Total number of alarms.
$HighestAlarms	Integer	0-2,147,483,647	Number of active highest priority alarms
$HighAlarms	Integer	0-2,147,483,647	Number of active high priority alarms
$MediumAlarms	Integer	0-2,147,483,647	Number of active medium priority alarms
$LowAlarms	Integer	0-2,147,483,647	Number of active low priority alarms
$LowestAlarms	Integer	0-2,147,483,647	Number of active lowest priority alarms
$Unacknowledged Alarms	Integer	0-2,147,483,647	Number of alarms currently unacknowledged

Error Logger Points

The following table describes system points for use with CX-Supervisor errors.

System point	Point type	Point range	Remarks
$HighErrors	Integer	0-2,147,483,647	Number of high priority errors logged.
$LowErrors	Integer	0-2,147,483,647	Number of low priority errors logged.
$MediumErrors	Integer	0-2,147,483,647	Number of medium priority errors logged.

PLC Communications Points

The following table describes system points for use in the communication between CX-Supervisor and a PLC.

System point	Point type	Point range	Remarks
$InputsActual	Real	—	Reports average number of PLC input points read per second since application startup.
$PLCBusy	Boolean	—	Indicates if PLC communications are busy.
$PLCFailures	Integer	0-2,147,483,647	Total number of PLC failures.

Security Points

The following table describes system points for use with user login, logout, and user privileges in the runtime environment.

System point	Point type	Point range	Remarks
$SecurityLevel	Integer	0-4	Current user's security level.
$SecurityName	Text	—	Current user's security name.
$UserName	Text	—	User currently logged on.

Printing Points

Print Setup

The Point Editor can be printed in the same way pages can. Before printing, ensure that the printer has been set up correctly. To check the printer settings, refer to chapter 2, Pages.

Print Preview

To preview the page before printing, select *Print Preview* from the *File* menu.

Refer to chapter 2, Pages regarding the use of the Print Preview display.

Printing

🖨 To print the contents of the Point Editor, select the **Print** button from the toolbar.

Refer to chapter 2, Pages regarding the use of the Print dialog.

Point Substitution

Overview

As well as using point names in expressions, in many cases where text is entered this can be made dynamic by using point substitution. Part or all of the text may be a point name, enclosed in the Point Substitution characters (double round brackets by default).

Examples of the way in which text is changed:

```
The shift foreman was ((ShiftForeman))
```

becomes:

```
The shift foreman was Fred Smith
```

and

```
The shift output was ((ShiftOutput)) litres
```

becomes:

```
The shift output was 5000 litres
```

Using Format specifiers

Format specifiers can be used, similar to the Format script command:

```
%s for text string points;
%d for integer points;
%f for real floating points.
```

Where ‘MyTextpoint=“Hello”’:

```
The text of ((“My text point is %s”,Mypoint))
```

becomes:

```
The text of My text point is Hello
```

Where MyRealpoint=5467.7658:

```
To two decimal places the value of MyRealpoint is
((“%4.2f”,MyRealpoint))
```

becomes:

```
To two decimal places the value of MyRealpoint is 5467.76
```

Combinations of format specifiers are possible:

```
((%s to two decimal places the value of MyRealpoint is
%4.2f”,MyTextpoint,MyRealpoint))
```

becomes:

```
Hello to two decimal places the value of MyRealpoint is  5467.76
```

The can be used in various fields, for example:

- In Alarm Raised and Cleared Messages e.g.

  Alarm Raised: `"Boiler Temperature too high. Current value is (("%.1d", BoilerTemp))"`

- As captions for popup edit box animations. This is useful as the captions are then dynamically translated e.g.

  Caption: `"((CaptionString))"`

- As parameter to MessageBox. This is useful to help convert numbers to strings e.g.

  `MessageBox("Error ((ErrorNum)) occurred")`

- As parameter to LogError or LogEvent. This too is useful to help convert numbers to strings e.g.

  `LogError("Error ((ErrorNum)) occurred", priority)`

- Titles in Graphs and Charts (useful for translations again) e.g.

  Title: `"((TitleString))"`

- In Tooltips, for dynamic information

  Tooltip text: `"((ToolTipString))"`

- In Report templates. See *Script Reference GenerateReport()* for more information.

See *Chapter 6, Point Substitution Settings* and *FAQ – Creating Reports and HTML Reports* for related information

CHAPTER 4 Objects

This chapter describes the various objects available within CX-Supervisor. It also describes the processes for creating, editing and manipulating objects.

Objects

Pages created with CX-Supervisor are constructed from objects that are inserted and linked together to form a coherent interface. CX-Supervisor objects are divided into three groups: graphical, control and embedded.

Generally, the procedures for creating objects are identical. A pushbutton representing the desired object is clicked on the Graphic Object bar. The mouse pointer is then either clicked on the page (for a default sized object), or clicked and dragged to the appropriate point on the page for a custom sized object.

Editing Objects

Editing falls into three distinct categories:

- Re-sizing an object.
- Re-shaping an object.
- Modifying an object using a Wizard.

Re-sizing

To re-size an object, click on it with the left mouse button. This ‘selects’ it and brings up green sizing grab handles. Click and drag the handles until the object is of the desired size.

Re-shaping

Re-shaping procedures are similar for all graphical objects which can be reshaped. Not all can. Control objects can only be reshaped using Wizards.

To change the shape of a graphical object, choose *Edit Object* from the *Edit* menu to bring up the red editing grab handles. Click and drag the handles until the object is of the desired shape.

The typeface of the editor can be amended by choosing *Preferences* from the *File* menu. This is especially useful when printing.

CX-Supervisor preferences can be configured such that double-clicking on a graphical object also brings up the red editing grab handles. Setting this preference is achieved by selecting *Preferences* from the *File* menu. Refer to chapter 2, Pages for further details on preferences and CX-Supervisor configuration.

Wizards

Control objects cannot be re-shaped in the same way as a graphical object but are edited using Wizards. Wizards customise control objects to display information in an easy to understand manner.

To edit a control object and activate a Wizard, double click on the object.

Note: The shortcut keyboard combinations for Cut, Copy and Paste operations are valid within Wizards. Highlight part or all of a field and type <Ctrl>+X to cut the text or <Ctrl>+C to copy the text; move the cursor to the desired field and type <Ctrl>+V to paste the text. Since the cut and copy operations store the information in the Windows Clipboard, they may be pasted to another Wizard, dialog or application.

Creating and Editing Graphic Objects

For details on re-sizing the graphic objects described in the following paragraphs, refer to chapter 3, Editing Objects.

Arc

Arcs may be transparent or filled with a colour or pattern. To insert an Arc, select the **Arc** button from the Graphic Object bar. Click or click and drag on the page.

To edit the arc, obtain the red grab handles. Click and drag to change the angle subtended by the arc. To finish editing, click elsewhere on the page or press <Esc>.

Block Text

To insert block text, click the **Block Text** button, and then click on the page. Stretch the text object to resize it. Standard text tools from the toolbar and the keyboard (e.g. bold, italic, left-justify) can be used, and their effect applies to the whole content of the object.

To edit block text, double click on it. A Text-Editing dialog is displayed. The text can then be changed, as can the word-wrap and border options. Standard Windows cut, copy and paste facilities can be used.

Ellipse

Ellipses may be transparent or filled with a colour or pattern. To create a filled ellipse, click the **Ellipse** button.

To create a transparent ellipse, click the **Ellipse Frame** button. Alternatively, select a filled ellipse and click the **Transparency**.

Either click on the page to create a circle, or click and drag to create an ellipse. Ellipses cannot be edited but can be re-sized.

Line

To insert a line, click on the **Line** button. Click and drag on the page to draw a line of the required length.

To edit the line, obtain the red grab handles. Click and drag to change the line. To finish editing click elsewhere on the page or press <Esc>.

Polygon

Polygons may be transparent or filled with a colour or pattern. To create a polygon, click the **Polygon** button. Click on the page to draw vertices. To finish creating the polygon either click the right mouse button, press <Esc> or double click the left mouse button.

To edit the polygon obtain the red grab handles. Click on an edge to add a vertex or click and drag a vertex to move it.

To finish editing, either click elsewhere on the page or press <Esc>.

To split a straight line into two, click at the point on the line where the split is required then drag the mouse. CX-Supervisor creates a new handle which may be moved to the desired point. To remove a red handle and the vertex on which it rests, click on it with the delete key held down on the keyboard.

Polyline

To create a polyline, click the **Polyline** button. Click on the page to draw vertices. To finish creating the polyline, click the right mouse button.

To edit a polyline, obtain the red grab handles. Click on an edge to add a vertex. Press the <Delete> key whilst moving a vertex to delete it.

To finish editing, click elsewhere on the page.

Rectangle

Rectangles can be transparent or filled with a colour or pattern. To create a filled rectangle, click the **Rectangle** button.

To create a rectangle frame, click the **Rectangle Frame** button. Alternatively, select a filled rectangle and click the **Transparency** button from the toolbar.

Click on the page to draw a square, or click and drag to create a rectangle of the required size. Once created, rectangles can be edited exactly like polygons.

To finish editing, click elsewhere on the page.

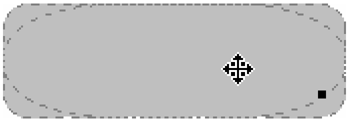
Round Rectangle

Rounded rectangles can be transparent or filled with a colour or pattern. To create a filled rounded rectangle, click the **Round Rectangle** button from the Tool Bar.

To create a rounded rectangle frame, click the **Round Rectangle Frame**. Alternatively, select a filled round rectangle and click the **Transparency** button from the Control Bar.

Click on the page to draw a rounded square, or click and drag to create a rounded rectangle of the required size.

To edit the rounded rectangle, obtain the red grab handles. The radius of curvature of the rounding can be adjusted by clicking on the single red grab handle and dragging it, as illustrated below:



Text

To insert text, click the **Text** button. Click on the page and type inside the red edit box. The cursor is moved round the text using the arrow keys. Standard text editing tools from the toolbar and the keyboard can be used, and their effect applies to the whole content of the object.

To edit text, double click on it. A box is displayed round the text.

Press <Return> to finish editing and create a new text object on the line below. To finish editing, click elsewhere on the page.

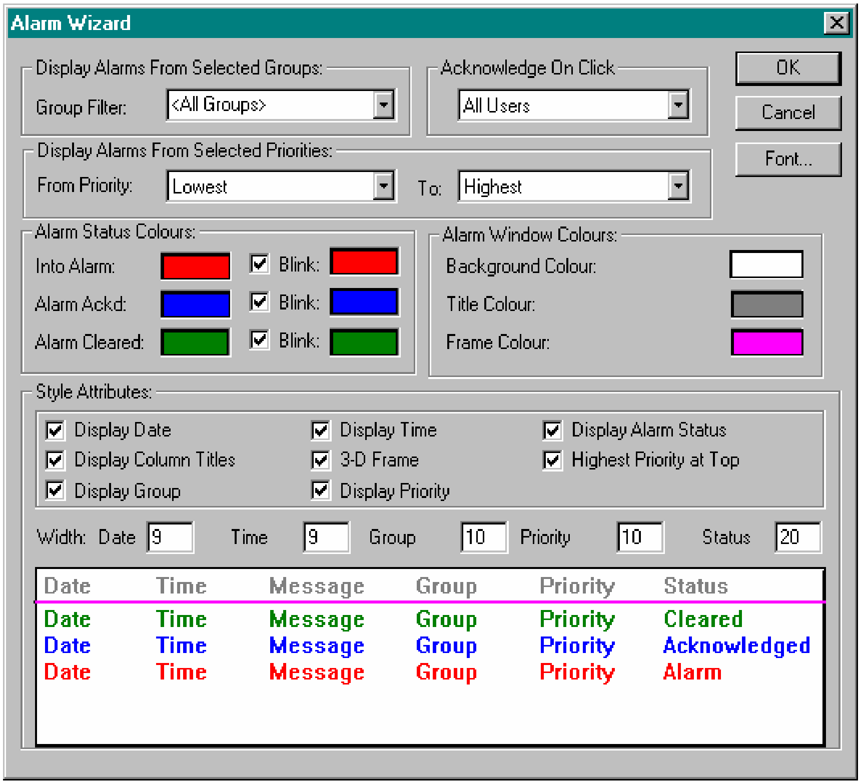
Creating and Editing Control Objects

Alarm Object

Click the **Alarm** button, then click or click and drag on the page to insert the alarm object.

The Alarm object displays alarm messages in runtime. These messages may be optionally filtered by an alarm group and can be formatted to include the date, time and status of the alarm. If there are more alarms to view than the space provided then you can scroll the alarm view.

To edit an Alarm object, double click on it. The Alarm Wizard dialog is displayed as follows:



Alarm Wizard

Display Alarms From Selected Groups:

Group Filter: <All Groups>

Acknowledge On Click

All Users

Display Alarms From Selected Priorities:

From Priority: Lowest To: Highest

Alarm Status Colours:

Into Alarm: 🟥 ☑ Blink: 🟥

Alarm Ackd: 🟦 ☑ Blink: 🟦

Alarm Cleared: 🟩 ☑ Blink: 🟩

Alarm Window Colours:

Background Colour: ⬜

Title Colour: ▨

Frame Colour: 🟪

Style Attributes:

☑ Display Date ☑ Display Time ☑ Display Alarm Status

☑ Display Column Titles ☑ 3-D Frame ☑ Highest Priority at Top

☑ Display Group ☑ Display Priority

Width: Date 9 Time 9 Group 10 Priority 10 Status 20

Date	Time	Message	Group	Priority	Status
Date	Time	Message	Group	Priority	Cleared
Date	Time	Message	Group	Priority	Acknowledged
Date	Time	Message	Group	Priority	Alarm

Alarms are defined using the Alarm Editor; refer to chapter 8, Alarms for information on using the Alarm Editor.

The Alarm Wizard allows entry of the alarm group filter, alarm status colour codes and various style attributes. The Alarm Wizard presents a preview of the alarm object, which immediately updates to show the user selections as they are made.

1, 2, 3…

1. Select the name of the alarm group by which alarm messages are to be filtered using the *Group* field to display the list of available groups. The default selection is *<All Groups>,* which displays all alarm messages.

   To display more than one group use the * wildcard character e.g. Group1* will include all groups starting Group1.

2. From the Acknowledge On Click options select the level of security that is required for acknowledgement. The default is *All Users.*

3. From the *Display Alarms from Selected Priorities,* options select the range of alarms that are to be displayed. The default selections are from the Lowest to the Highest.

   Note that the range ‘From’ must be the same or lower than the range ‘To’. Use the check box ‘Highest Priority at Top’ to reverse the alarm order.

4. In *Alarm Status Colours,* the colour codes for each of the three alarm states and blinking colours may be specified by clicking the appropriate colour box and selecting a new colour from the resultant Colour dialog.

5. The *Alarm Window Colours* options allow you to select the colours for the Alarms Windows Background, Title text and the Frame.

6. Select the font used by the alarm object using the **Font** pushbutton.

7. In *Style Attributes*, set the following options as desired:

Display Date:	shows the date of the alarm.
Display Time:	shows the time of the alarm.
Display Alarm Status:	shows the status of the alarm.
Display Column Titles:	shows the column headings.
3-D Frame:	displays object with 3-D border.
Highest Priority at Top:	shows highest priority at top.
Display Group:	shows the groups of the alarm.
Display Priority:	shows the priority of the alarm.
Date Width:	number of characters in date field.
Time Width:	number of characters in time field.
Group Width	number of characters in priority field
Status:	number of characters in status field.

8. Exit the Wizard by clicking the **OK** pushbutton to accept the new alarm object attributes or click the **Cancel** pushbutton to leave the alarm object unchanged.

Bar Chart

Click the **Chart** button, then click or click and drag on the page to insert the Chart.

To edit a Bar Chart, double click on it. The Chart Wizard dialog is displayed as follows:

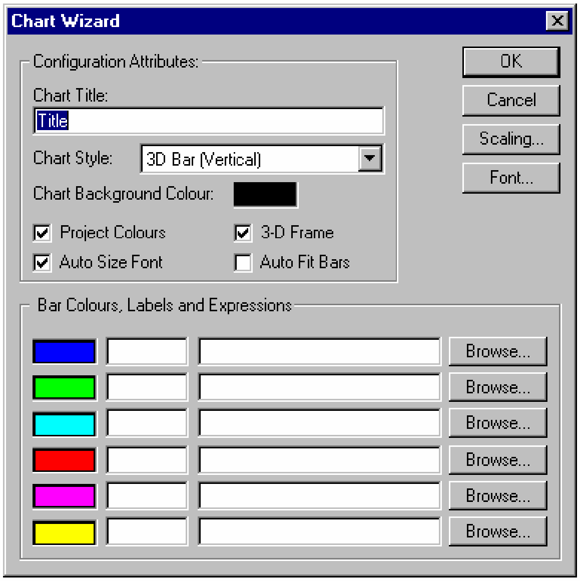


Chart Wizard

Configuration Attributes:

Chart Title: Title

Chart Style: 3D Bar (Vertical)

Chart Background Colour:

☑ Project Colours ☑ 3-D Frame

☑ Auto Size Font □ Auto Fit Bars

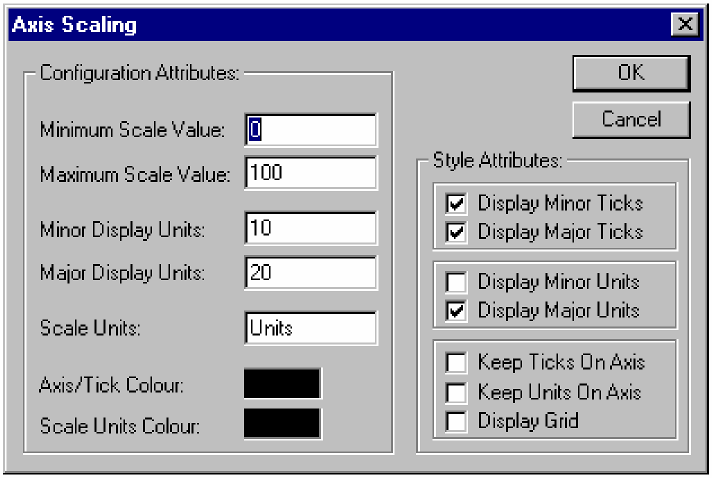
Bar Colours, Labels and Expressions

The Chart Wizard allows entry of configuration attributes and assignment of expressions.

1, 2, 3…

1. Enter a title for the chart in the *Chart Title:* field.
2. Select the chart style from the *Chart Style:* field.
3. Select the colour of the chart background.
4. Toggle the *Project Colours, 3-D Frame* and *Auto Fit Bars* fields as desired. The *Project Colours* option allows the user to change the colour of the axis using the toolbox. The *3-D Frame* option enables the chart to appear with a 3-D frame. The *Auto Fit Bars* option forces the configured bars to resize themselves to occupy all of the available chart area.
5. Change the fonts used for the chart via the Font pushbutton. The font size used for the chart can be automatically calculated by selecting the *Auto Size Font* option.
6. Enter an arithmetic trace expression for each point or select one by clicking on a **Browse** pushbutton. An existing point can also be associated with the Wizard by dragging a point directly from the Point Editor. Refer to chapter 3, Points regarding adding a new point and dragging from the Point Editor.
7. Exit the Wizard by clicking the **OK** pushbutton to accept the new chart attributes, or click the **Cancel** pushbutton to leave the chart unchanged.

To select the chart scaling, click the **Scaling** pushbutton; the Axis Scaling dialog is displayed as follows:



Axis Scaling

Configuration Attributes:

Minimum Scale Value: 0

Maximum Scale Value: 100

Minor Display Units: 10

Major Display Units: 20

Scale Units: Units

Axis/Tick Colour: ■

Scale Units Colour: ■

Style Attributes:

☑ Display Minor Ticks
☑ Display Major Ticks

☐ Display Minor Units
☑ Display Major Units

☐ Keep Ticks On Axis
☐ Keep Units On Axis
☐ Display Grid

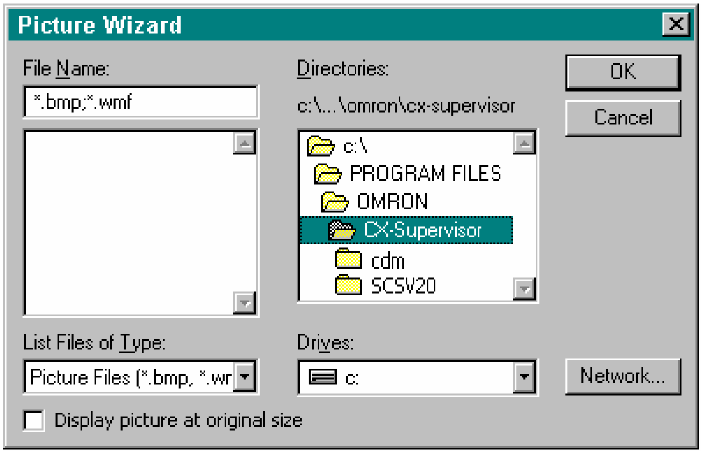
The *Configuration Attributes* fields can be altered by typing over the existing entries. The *Style Attributes* fields can be amended by clicking on the settings.

Exit the Axis Scaling dialog by clicking the **OK** pushbutton to accept the scaling attributes, or click the **Cancel** pushbutton to leave them unchanged.

Pictures

Pictures and graphics can be inserted on a page in the form of bitmaps (.bmp) or Windows metafiles (.wmf). Click the **Picture** button, then click or click and drag on the page to insert a picture placeholder.

To insert a picture, double click on the placeholder. The Picture Wizard dialog is displayed as follows:



Picture Wizard

File Name: *.bmp;*.wmf

Directories: c:\...\omron\cx-supervisor

c:\
PROGRAM FILES
OMRON
CX-Supervisor
cdm
SCSV20

List Files of Type: Picture Files (*.bmp, *.wr

Drives: c:

☐ Display picture at original size

1, 2, 3…

1. Locate the drive and directory where the desired picture is stored using the *Directories:* and *Drives:* controls.
2. Select the file type and then the picture from the list presented.
3. Use the *Display picture at original size* setting to specify whether the picture is to be displayed at original size or scaled.
4. Click the **OK** pushbutton to load the page.

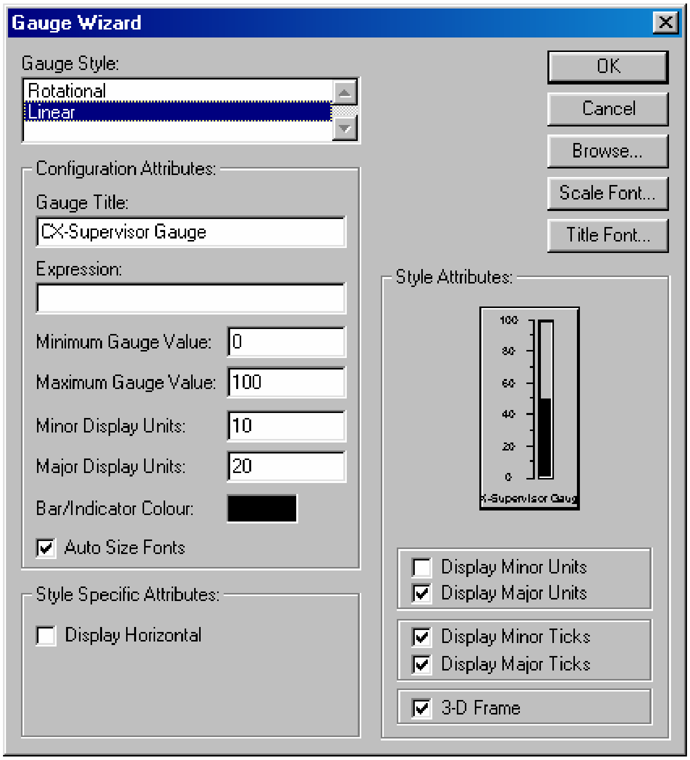
Note: If the PC installation of CX-Supervisor is on a networked machine, a Network pushbutton is added to the dialog. For further information on the function of the Network dialog, refer to the Microsoft Windows User Guide.

A bitmap or Windows metafile image can be selected and resized just like a graphical object.

Linear Gauge

A Gauge provides a display of operational values. Click the **Linear Gauge** button, then click or click and drag on the page to insert the gauge.

To edit a Linear Gauge object, double click on it. The Gauge Wizard dialog is displayed as follows:



Gauge Wizard

Gauge Style:
Rotational
Linear

Configuration Attributes:

Gauge Title: CX-Supervisor Gauge

Expression:

Minimum Gauge Value: 0

Maximum Gauge Value: 100

Minor Display Units: 10

Major Display Units: 20

Bar/Indicator Colour: ■

☑ Auto Size Fonts

Style Specific Attributes:

☐ Display Horizontal

Style Attributes:

☐ Display Minor Units
☑ Display Major Units

☑ Display Minor Ticks
☑ Display Major Ticks

☑ 3-D Frame

The Gauge Wizard dialog allows entry of the *Gauge Style:, Configuration Attributes:, Style Attributes:* and *Style Specific Attributes:* fields.

To select a style, click on an option from the appropriate field. To select an Expression Attribute, click on the **Browse** pushbutton and select a point from the displayed list. The Select Required Item dialog is displayed; click the **OK** pushbutton to accept the point or click the **Cancel** pushbutton to leave the point unselected. Clicking the **Add Point** button from the toolbar allows a new point to be created prior to association with the Gauge Wizard. An existing point can also be associated with the Gauge Wizard by dragging a point directly from the Point Editor. Refer to chapter 3, Points regarding adding a new point and dragging from the Point Editor.

To enter gauge values or display units, type over the existing field entries.

The Gauge can be displayed in linear or rotary format, either with or without ticks.

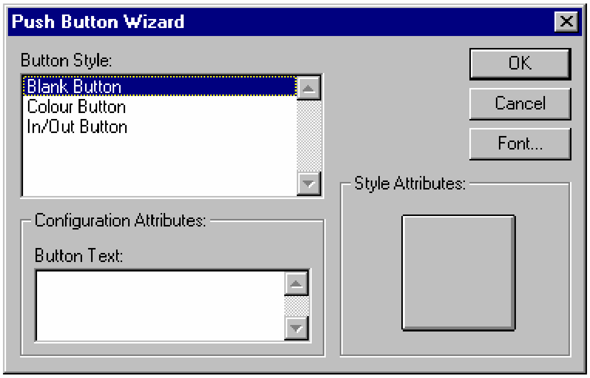
The text font used for the gauge title can be changed via the **Title Font** pushbutton. The text font used for the gauge scale can be changed via the **Scale Font** pushbutton. The font size used by the gauge can be automatically calculated for the user by selecting the *Auto Size Fonts* field.

Exit the Gauge Wizard by clicking the **OK** pushbutton to accept the new gauge attributes or click the **Cancel** pushbutton to leave the gauge unchanged.

Pushbutton

Pushbuttons provide a simple means to start a set of actions. Click the **Push Button** button, then click or click and drag on the page to insert a pushbutton.

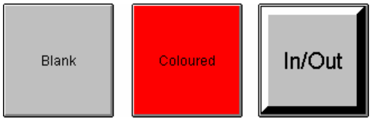
To edit the pushbutton, double click on it. The Push Button Wizard dialog is displayed as follows:



Push Button Wizard

Button Style: Blank Button

The Wizard allows a pushbutton to be assigned a style or text or both to signify its purpose. To select a style, click on an option in the *Button Style* field. To enter button text, type in the *Button Text* field (the Style Attributes: dialog is automatically updated). The text font can be changed via the **Font** pushbutton. Click the **OK** pushbutton to accept the new button attributes, or click the **Cancel** pushbutton to leave the button unchanged. Examples of the different styles of pushbutton are illustrated below:

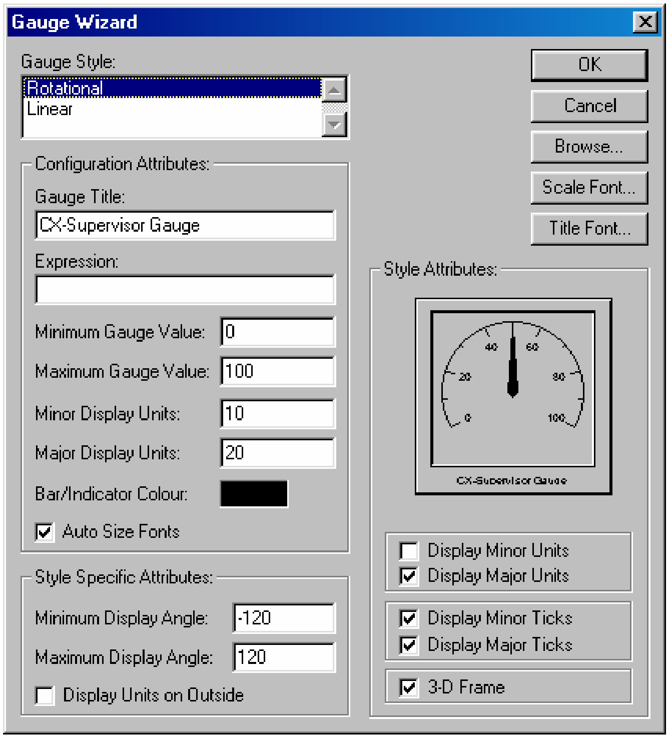


The colour of the coloured pushbutton is red by default, but can be changed by using the Palette.

Rotary Gauge

A Gauge provides a means of displaying the value of an operation or the value of a point. Click the **Rotary Gauge** button, then click or click and drag on the page to insert a gauge.

To edit the Rotary Gauge, double click on it. The Gauge Wizard dialog is displayed as follows:



Gauge Style: Rotational (selected), Linear

Configuration Attributes:

Gauge Title: CX-Supervisor Gauge

Expression:

Minimum Gauge Value: 0

Maximum Gauge Value: 100

Minor Display Units: 10

Major Display Units: 20

Bar/Indicator Colour: ■

☑ Auto Size Fonts

Style Specific Attributes:

Minimum Display Angle: -120

Maximum Display Angle: 120

□ Display Units on Outside

Style Attributes:

□ Display Minor Units
☑ Display Major Units

☑ Display Minor Ticks
☑ Display Major Ticks

☑ 3-D Frame

The Wizard allows entry in the *Gauge Style:*, *Configuration Attributes:*, *Style Attributes:* and *Style Specific Attributes:* fields.

To select a style, click on an entry in the *Gauge Style* field. To select an Expression Attribute, click on the **Browse** pushbutton and select a point from the displayed list. The Select Required Item dialog is displayed; click the **OK** pushbutton to accept the point or click the **Cancel** pushbutton to leave the point unselected. Clicking the **Add Point** button from the toolbar allows a new point to be created prior to association with the Wizard. An existing point can also be associated with the Wizard by dragging a point directly from the Point Editor. Refer to chapter 3, Points regarding adding a new point and dragging from the Point Editor.

To enter gauge values or display units, type over the existing field entries. Enter display angles by typing over the existing field entries.

Check the boxes to choose the required style attributes. The Gauge can be displayed in linear or rotary format, either with or without ticks.

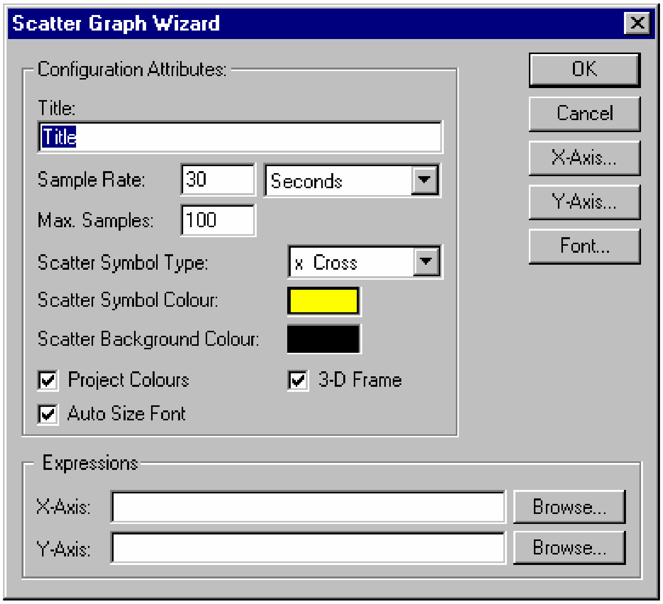
The text font used for the gauge title can be changed via the **Title Font** pushbutton. The text font used for the gauge scale can be changed via the **Scale Font** pushbutton. The font size used by the gauge can be automatically calculated for the user by selecting the *Auto Font Size* option.

Exit the Wizard by clicking the **OK** pushbutton to accept the new gauge attributes or click the **Cancel** pushbutton to leave the gauge unchanged.

Scatter Graph

Click the **Scatter Graph** button, then click or click and drag on the page to insert a graph.

To edit the graph, double click on it. The Scatter Graph Wizard dialog is displayed as follows:



Scatter Graph Wizard

Configuration Attributes:

Title: Title

Sample Rate: 30 Seconds

Max. Samples: 100

Scatter Symbol Type: x Cross

Scatter Symbol Colour: 🟨

Scatter Background Colour: ⬛

☑ Project Colours ☑ 3-D Frame

☑ Auto Size Font

Expressions

X-Axis:

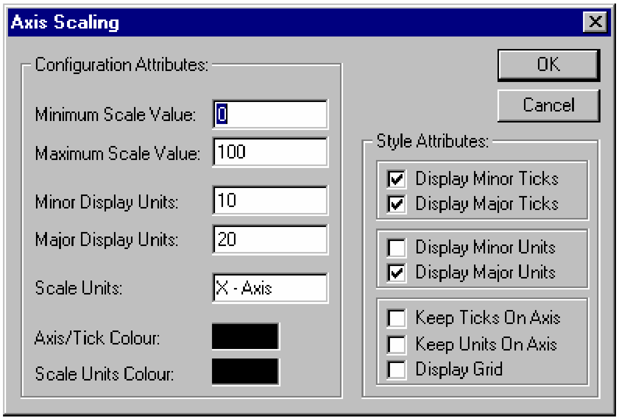
Y-Axis:

The Wizard allows entry in the *Configuration Attributes:* and *Expressions* fields.

1, 2, 3…

1. Enter the frequency of data sampling in the *Sample Rate:* field; in this example the sample rate is every thirty seconds.
2. Enter the maximum number of samples to be displayed on the scatter graph in the *Max Samples:* field.
3. Select the type of symbol to represent the data.
4. Select the colour of the symbol representing the data.
5. Select the colour of the scatter graph background.
6. Toggle the *Project Colours* and *3-D Frame* settings as desired. If set, the *Project Colours* field allows the colour of the axis to be changed using the Palette. The 3-D Frame option enables the graph to appear with a 3-D frame.
7. Either enter an arithmetic expression for the X and Y axis or select one through the **Browse** pushbutton. An expression point can also be associated with the Scatter Graph Wizard by dragging a point directly from the Point Editor. Refer to chapter 3, Points regarding adding a new point and dragging from the Point Editor.
8. Change the fonts used for the Scatter Graph via the **Font** pushbutton. The font size used by the graph can be automatically calculated for the user by selecting the *Auto Size Font* field.
9. Exit the Scatter Graph Wizard by clicking the **OK** pushbutton to accept the new scatter graph attributes or click the **Cancel** pushbutton to leave the scatter graph unchanged.

To configure the X axis, click the **X-Axis** pushbutton. To configure the Y-axis , click the **Y-Axis** pushbutton. The Axis Scaling dialog is displayed:



Axis Scaling

Configuration Attributes:

Minimum Scale Value: 0

Maximum Scale Value: 100

Minor Display Units: 10

Major Display Units: 20

Scale Units: X - Axis

Axis/Tick Colour: ■

Scale Units Colour: ■

Style Attributes:

☑ Display Minor Ticks
☑ Display Major Ticks

☐ Display Minor Units
☑ Display Major Units

☐ Keep Ticks On Axis
☐ Keep Units On Axis
☐ Display Grid

Configuration attributes can be altered by typing over the existing field entries. The style attributes can be amended by clicking on the settings.

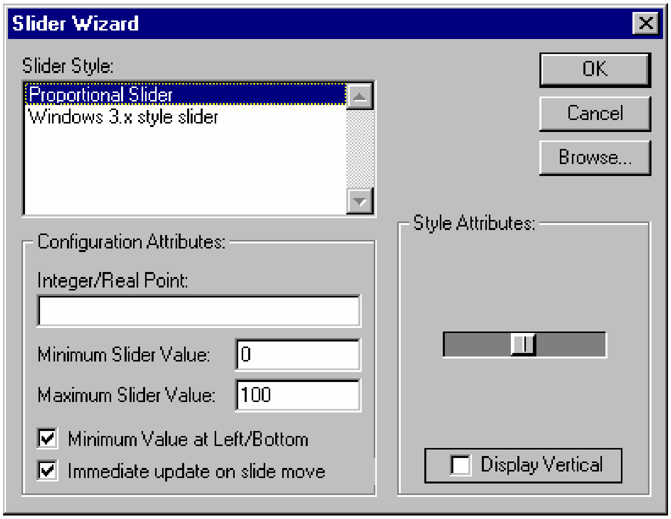
Exit the Axis Scaling dialog by clicking the **OK** pushbutton to accept the scaling attributes, or click the **Cancel** pushbutton to leave unchanged.

Slider

Sliders allow values associated with them to increase or decrease between certain limits. They can also ensure that values do not exceed previously set limits.

Click the **Slider** button, then click or click and drag on the page to insert a slider.

To edit the Slider, double click on it. The Slider Wizard dialog is displayed as follows:



Slider Style:
- Proportional Slider
- Windows 3.x style slider

Configuration Attributes:

Integer/Real Point:

Minimum Slider Value: 0

Maximum Slider Value: 100

☑ Minimum Value at Left/Bottom

☑ Immediate update on slide move

Style Attributes:

☐ Display Vertical

The Wizard allows entry of in the *Slider Style:, Style Attributes:* and *Configuration Attributes:* fields. To select a style, click on an option from the list box. The *Style Attributes:* can be either *Display Vertical* or *Display Horizontal*. To select a Boolean point, click on the **Browse** pushbutton and select a point from the displayed list. Click the **OK** pushbutton to accept the point or click the **Cancel** pushbutton to leave the point unselected. Clicking the **Add Point** button from the toolbar allows a new point to be created prior to association with the Wizard. An existing point can also be associated with the Wizard by dragging a point directly from the Point Editor. Refer to chapter 3, Points regarding adding a new point and dragging from the Point Editor.

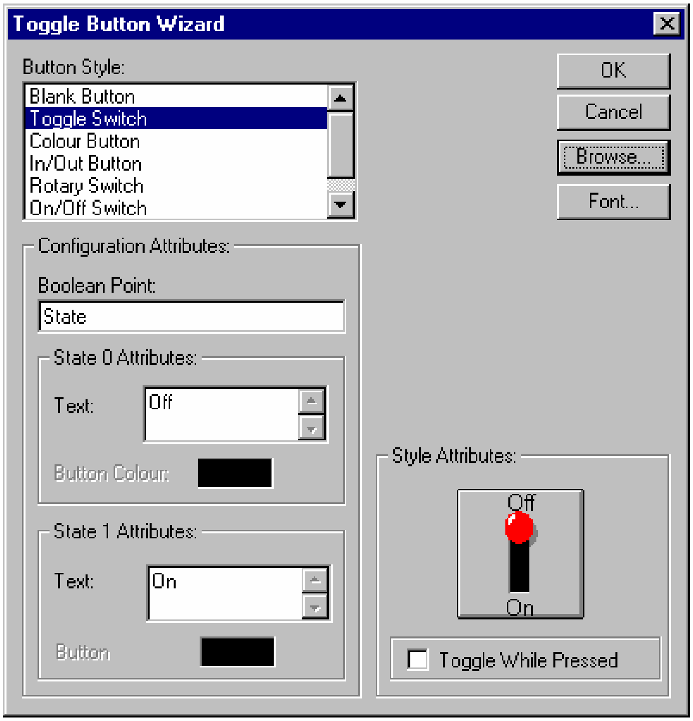
Enter the maximum and minimum slider values and check the box to display the minimum value at the left or bottom of the page.

Exit the Wizard by clicking on the **OK** pushbutton to accept the new slider attributes or click on the **Cancel** pushbutton to leave the slider unchanged.

Toggle Button

Toggle buttons are used to control and display the current value of a digital point. Click the **Toggle Button** button, then click or click and drag on the page to insert a button.

To edit the Toggle Button, double click on it. The Toggle Button Wizard dialog is displayed as follows:



Toggle Button Wizard

Button Style:
- Blank Button
- Toggle Switch
- Colour Button
- In/Out Button
- Rotary Switch
- On/Off Switch

Configuration Attributes:

Boolean Point: State

State 0 Attributes:

Text: Off

Button Colour:

State 1 Attributes:

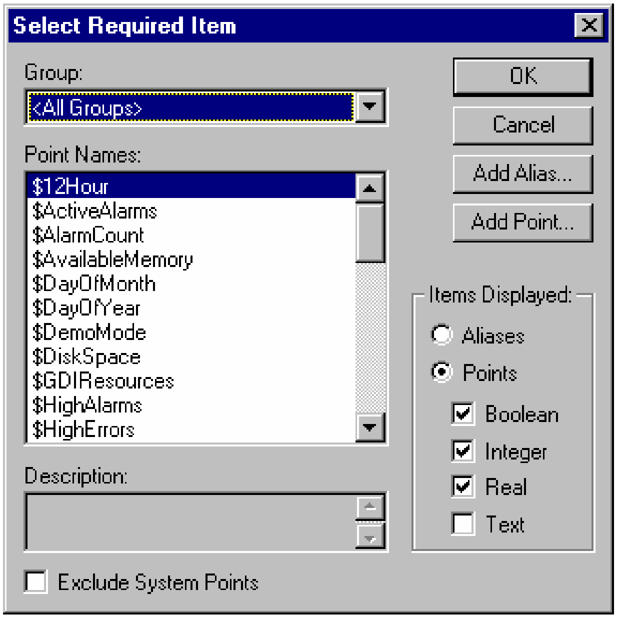
Text: On

Button

Style Attributes:

□ Toggle While Pressed

The Toggle Button Wizard allows a Toggle Button to be assigned a style, State 0 text and State 1 text to signify its purpose. It also allows selection of a Boolean point. To select a style, click on an option from the list box. To enter button text, type in the Text fields for States 0 and 1, or select *ON* and *OFF* (the Style Attributes: dialog is automatically updated). Some toggle buttons can have an On/Off colour associated with them. The text font can be changed via the **Font** pushbutton. To select a Boolean point, click on the **Browse** pushbutton and click on a point from the displayed list. The Select Required Item dialog is displayed, as illustrated below:



Select Required Item

Group: <All Groups>

Point Names:

$12Hour
$ActiveAlarms
$AlarmCount
$AvailableMemory
$DayOfMonth
$DayOfYear
$DemoMode
$DiskSpace
$GDIResources
$HighAlarms
$HighErrors

Description:

□ Exclude System Points

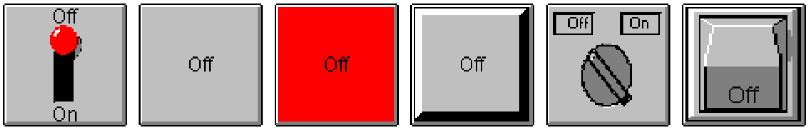
Items Displayed:

○ Aliases
● Points
☑ Boolean
☑ Integer
☑ Real
□ Text

Only viable points can be viewed from a Select Required Item dialog. The list of items in the *Point Names:* field can be refined by selecting an option from the *Group:* field. Click the **OK** pushbutton to accept the new point or click the **Cancel** pushbutton to leave the point unchanged. Clicking the **Add Point** pushbutton or **Add Alias** pushbutton allows a new point or alias to be created prior to association with the Wizard. An existing point can also be associated with the Wizard by dragging a point directly from the Point Editor. Refer to chapter 3, Points regarding adding a new point and dragging from the Point Editor.

Select the *Toggle While Pressed* field if the value should only be set while the button is pressed (i.e. the mouse button is held down when the toggle button is clicked). The value has its state toggled and then set back when the mouse button is released.

The different styles of toggle button are the same as those for pushbuttons. Examples of some of the different styles of toggle button are illustrated below:



Switch **Blank Toggle** **Coloured** **In/Out** **Rotary** **On/Off**

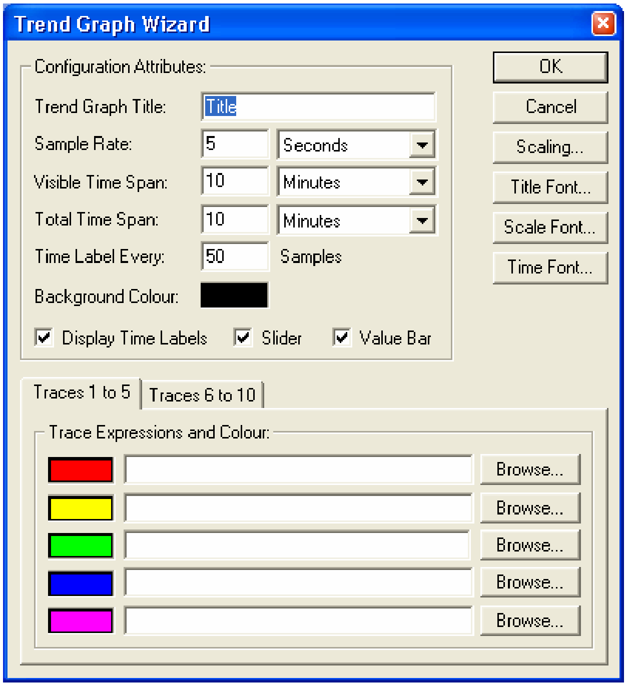
The colour of the coloured pushbutton is red by default, but can be changed by using the Palette.

Click the **OK** pushbutton to accept the new toggle button attributes or click the **Cancel** pushbutton to leave the button unchanged.

Trend Graph

Trend graphs allow the display of data over time. Click the **Trend Graph** button, then click or click and drag on the page to insert the graph.

To edit the Trend Graph, double click on it. The Trend Graph Wizard dialog is displayed as follows:



Trend Graph Wizard

Configuration Attributes:

Trend Graph Title: Title

Sample Rate: 5 Seconds

Visible Time Span: 10 Minutes

Total Time Span: 10 Minutes

Time Label Every: 50 Samples

Background Colour: ■

☑ Display Time Labels ☑ Slider ☑ Value Bar

Traces 1 to 5 | Traces 6 to 10

Trace Expressions and Colour:

🟥

🟨

🟩

🟦

🟪

The Wizard allows entry of *Configuration Attributes:* and assignment of line colours and expressions.

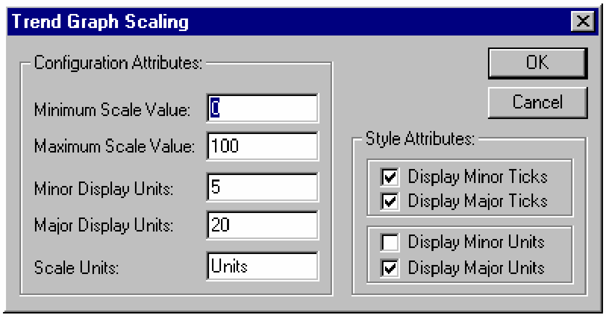
1, 2, 3…

1. Enter a title for the graph in the *Trend Graph Title:* field.
2. Enter the frequency of data sampling in the *Sample Rate:* field, in this example it is every five seconds.
3. Enter the period displayed by the graph at any one time in the *Visible Time Span:* field.
4. Enter the size of buffer for stored data of samples in the *Total Time Span:* field.
5. Enter the number of samples on the X-axis in the *Time Label Every:* field.
6. Click on the background colour and select the required colour from the palette.

7. Toggle the *Display Time Labels*, *Display Slider and Value Bar* settings as desired. Enabling the Value Bar allows the graph to be clicked during runtime to display the data value at that point.
8. Enter an arithmetic trace expression for each point or select an expression via a **Browse** pushbutton. An existing point can also be associated with the Wizard by dragging the point directly from the Point Editor. See chapter 3, Points regarding adding a new point and dragging from the Point Editor.
9. Change the fonts used for the Graph Title, Scale and Time axis via the appropriate font button. Alternatively, font sizes can be automatically calculated by selecting the Auto Font Size option.
10. Exit the Wizard by clicking the **OK** pushbutton to accept the new Trend Graph attributes, or click **Cancel** to leave the Trend Graph unchanged.

The units of measurement of time are selected from the associated field.

To select graph scaling, click the **Scaling** pushbutton; the Trend Graph Scaling dialog box is displayed as follows:


Trend Graph Scaling

Configuration Attributes:

Minimum Scale Value: 0

Maximum Scale Value: 100

Minor Display Units: 5

Major Display Units: 20

Scale Units: Units

Style Attributes:

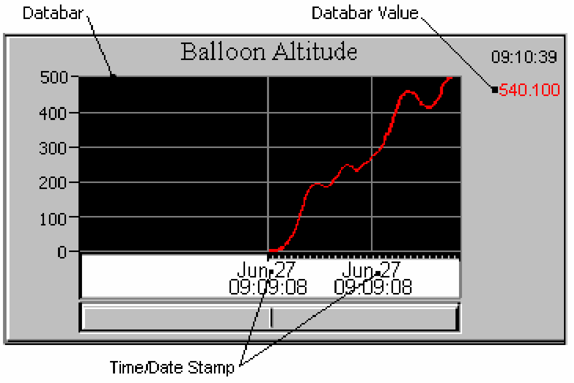
☑ Display Minor Ticks
☑ Display Major Ticks

□ Display Minor Units
☑ Display Major Units

Configuration Attributes: can be altered by typing over the existing entries. The *Style Attributes:* can be amended by clicking on the settings.

Exit the Trend Graph Scaling dialog by clicking the **OK** pushbutton to accept the scaling attributes, or click the **Cancel** push button to leave unchanged.

An example of a trend graph is illustrated as follows:



Web Browser Object

The Web Browser object allows web files, like HTML JPG or AVI files to be added to a CX-Supervisor page. These files may be stored locally, on a File Server or be distributed from any Web Server. The Web Browser object includes a Java Virtual Machine (JVM) allowing execution of Java Script and Java Applets. Double click the object to show the property page, and enter the file to be displayed either as a filename, or a fully qualified URL, for example with http: prefix.

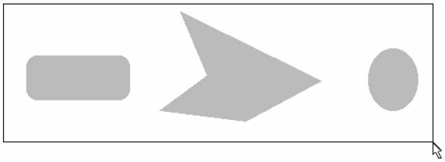
Manipulating Objects

Once inserted, objects can be manipulated to give the required results. An object must be selected before it can be manipulated.

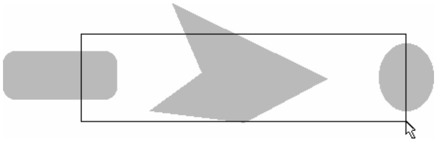
Select

To select an object, either click on it with the left mouse button or select it from the object identification control, (for further details on this control refer to chapter 1, Graphics Editor). Eight "grab handles" are displayed around the object.

To select several objects within a rectangular area, use a rubber band by clicking the left mouse button and dragging over an area, as illustrated below:



It is also possible to select all the objects a rubber band intersects by holding down the <Ctrl> key whilst rubber banding a selection, as illustrated below:



Multiple objects can be selected by holding the <Shift> key down and clicking on each object in turn. Objects can also be de-selected in similar fashion. Grab handles are displayed for each selected object.

The most recent object to be selected from the group is denoted by its green grab handles, all other grab handles are cyan. The co-ordinates of the most recent selection are displayed in the status bar.

All objects can be selected by clicking *Select All* from the *Edit* menu.

Move

To move an object or a number of objects, select them and click and hold the left mouse button within the selection, the object(s) can now be 'dragged' to their new location.

Cut

Where objects are to be moved between pages, it is often useful to cut them without having to insert a new object. CX-Supervisor has the ability to cut and paste objects. Objects which are cut and pasted retain the properties assigned to them, for example animation or alarms.

To cut an object from the current page, select it and click the **Cut** button on the toolbar.

The object is removed from the page and is held on the "clipboard" until a new object is cut or copied.

Copy

An object which is to appear on a number of pages can be copied from an original. To copy an object, select it then click the **Copy** button on the toolbar.

A copy of the object is held on the clipboard, overwriting the previously copied or cut object.

Paste

To paste an object which has been cut or copied to the clipboard, click the **Paste** button on the toolbar.

The object currently on the clipboard is pasted, either over the original if the object is to remain on the same page, or positioned in a new page. Objects can then be re-positioned by selecting and dragging using the mouse.

Delete

To delete an object or objects, select them and press the <Delete> key on the keyboard.

Undo

The **Undo** button on the toolbar allows the most recent action (or actions) to be undone. To undo the action, click the **Undo** button on the toolbar.

The number of undo operations stored by CX-Supervisor for retrieval can be amended from the *Preferences* option on the *File* menu. This is especially useful when constructing complex pages.

Mirror Image

There are various ways of mirroring objects:

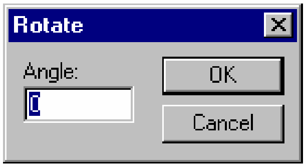
- Via the **Mirror Horizontal** and **Mirror Vertical** buttons on the toolbar.
- By selecting *Mirror* from the *Edit* menu and clicking on either *Horizontal* or *Vertical*.
- By clicking on a grab handle and dragging it across the opposite side

Some objects like Text and Control Objects cannot be mirrored. For further details of mirroring objects refer to chapter 1, Graphics Editor.

Orientation

Lines, Rectangles, Polygons and Polylines can be rotated. To rotate an object, click the **Rotate** button on the toolbar.

The Rotate Object dialog is displayed:



Enter the angle of rotation in degrees in the value entry box and click the **OK** pushbutton.

Transparency

Solid shapes, polygons and text boxes can be filled with colours and patterns from the tools palette. When they are created solid objects are filled with the colour already selected on the palette. Further details on transparency are contained in chapter 1, Graphics Editor.

Group

When there are a number of objects together they can be moved around in a group, keeping their relative position.

To group a number of objects, select them, activate the *Edit* menu and click on *Group*.

The selection is now regarded as one group with eight “grab handles” for the group rather than eight for each object; the objects can now be moved together.

Once objects have been grouped they can be ungrouped by selecting the object, activating the *Edit* menu and clicking on *Ungroup.*

Raise and Lower

Each new object inserted on a page is placed on top of the previous one. Although they may appear to be on the same level, objects can overlap, so it may be necessary to ‘raise’ an object so that it appears over the top of another object. For further details on raising and lowering objects refer to chapter 1, Graphics Editor.

Alignment

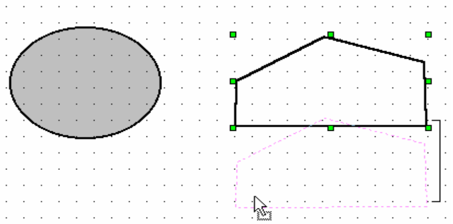
Objects inserted on a page can look messy unless they are aligned relative to each other. CX-Supervisor helps by allowing the contents of a page to align on a grid. The grid can be turned on and off as required.

There are a number of different grid sizes available: 4, 8, 16, 32 and 64 pixels. The Snap To Grid function snaps to the nearest grid.

To align page objects and view a grid, select *Grid* from the *View* menu.

Select the required grid size or click on *Snap to Grid* to align the most recently entered objects on the chosen grid. The grid can be turned off by selecting *Off*.

An example of the use of the grid is illustrated as follows:



In this example, the selected polyline is in the process of being moved down. The presence of the grid governs the movement of the object in all directions. As the object is moved, its position “snaps” an equal distance from the original position of the object against the grid. Currently, the object has been moved five grid steps down (shown by the bracket).

Alignment Toolbox

Objects on a page can be aligned in a variety of ways using the Alignment toolbar.

It is possible to:

Click the **Centre Horizontally on Page** button from the toolbar to centre objects on a page horizontally.

Click the **Centre Vertically on Page** button from the toolbar to centre objects on a page vertically.

Click the **Left Alignment** button from the toolbar to align objects along their left edge.

Click the **Right Alignment** button from the toolbar to align objects along their right edge.

Click the **Top Alignment** button from the toolbar to align objects along their top edge.

Click the **Bottom Alignment** button from the toolbar to align objects along their bottom edge.

Click the **Centre Align Horizontal** button from the toolbar to align objects on their horizontal centres.

Click the **Centre Align Vertical** button from the toolbar to align objects on their vertical centres.

Click the **Make Same Width** button from the toolbar to make objects the same width

Click the **Make Same Height** button from the toolbar to make objects the same height.

Click the **Make Same Height and Width** button from the toolbar to make objects the same width and height.

Click the **Align to Grid** button from the toolbar to align objects to the grid.

When performing alignment operations, the Master Object determines how the other objects are aligned. The Master Object, i.e. the last object clicked on, has green sizing handles.

To align objects:

1, 2, 3…

1. Select the objects to be aligned.
2. Ensure the correct master object is selected.
3. Choose the appropriate tool from the Alignment toolbar.

The alignment operations available at any one time vary according to the number of objects selected.

Zoom

It is often useful to be able to view an object in more detail. The Zoom option allows selected objects to be magnified up to four times their normal size.

To enlarge an area of the screen, choose the *View* menu and select *Zoom.* Then select the required magnification factor: *Off, 2×*or *4***.**

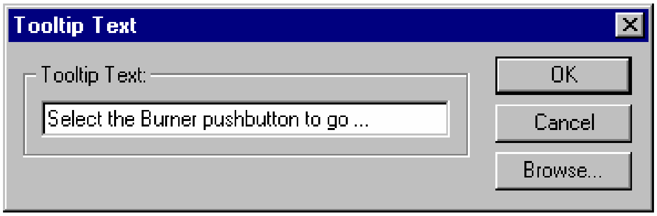
A specific object can be zoomed in on by selecting that object and then performing the procedures described above.

Applying Tooltips

In the runtime environment, it is possible to provide instant help for the object currently selected by the cursor. This is achieved by tooltips that can be applied to all objects.

1, 2, 3…

1. In the development environment, select the object to apply a tooltip.
2. Click on *Tooltip Text* from the *Edit* menu. The Tooltip Text is displayed.

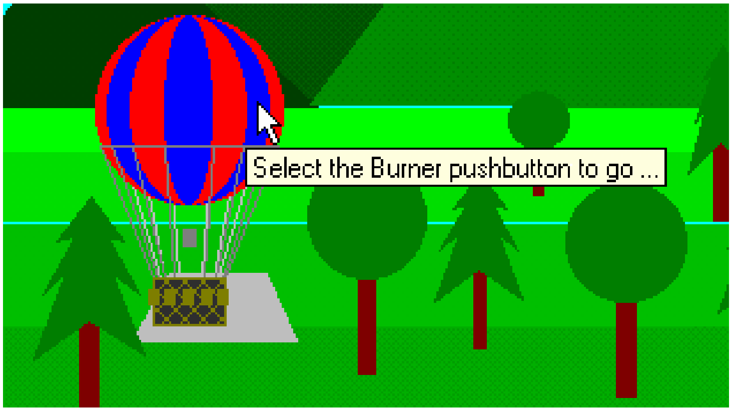


Tooltip Text

Tooltip Text: Select the Burner pushbutton to go ...

3. Type the help text in the *Tooltip text:* field or select the **Browse** pushbutton to apply the value of a point.
4. Click the **OK** pushbutton to accept the settings or the **Cancel** pushbutton to abort the operation.

An example of a tooltip in the runtime environment is as follows:



Using the Floating Menu

Clicking the right mouse button within CX-Supervisor brings up a context-sensitive menu known as the Floating menu. There are two such menus, one in the development environment and one in the runtime environment. The development environment Floating menu contains short-cuts for many operations discussed in this chapter.

CHAPTER 5
ActiveX Objects

This chapter describes the process of using ActiveX objects within CX-Supervisor applications.

Overview

The Microsoft Windows ActiveX technology allows objects called ‘components’ to be developed, and are used by inserting them into an ActiveX container, like CX-Supervisor. ActiveX components may perform many different functions, which can be graphical or non-graphical, but they follow standard rules for defining their Properties, Methods and Events. Properties are like settings, for example a control’s colour would be a property. Methods are like functions or actions that can be called for example a control might support a Redraw method. Events are actions the control may create, like OnLeftClick. Following these standard rules allows any ActiveX control from any manufacturer to work in any container.

Inserting a new object

A new ActiveX control may be inserted on the page as follows:

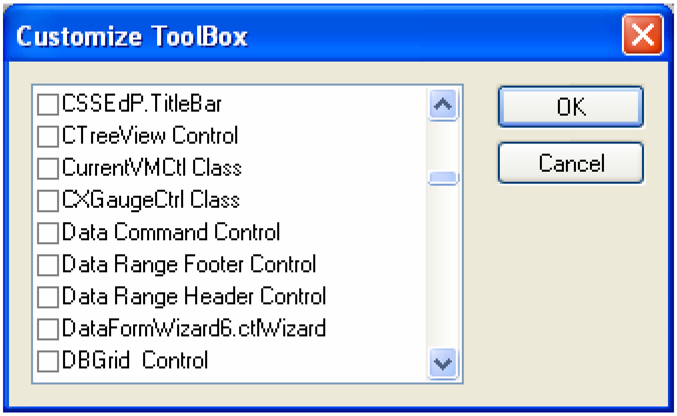
1, 2, 3… 1. Click on the page in which the object is to be inserted.

2. Turn on the ActiveX toolbar from the **View – Toolbars** menu
3. If the control you require already exists on the ActiveX toolbar:

   a) Select the required control

   or if the required control is not on the toolbar:

   b) Click Configure ActiveX Toolbar button and select the required component from the dialog:



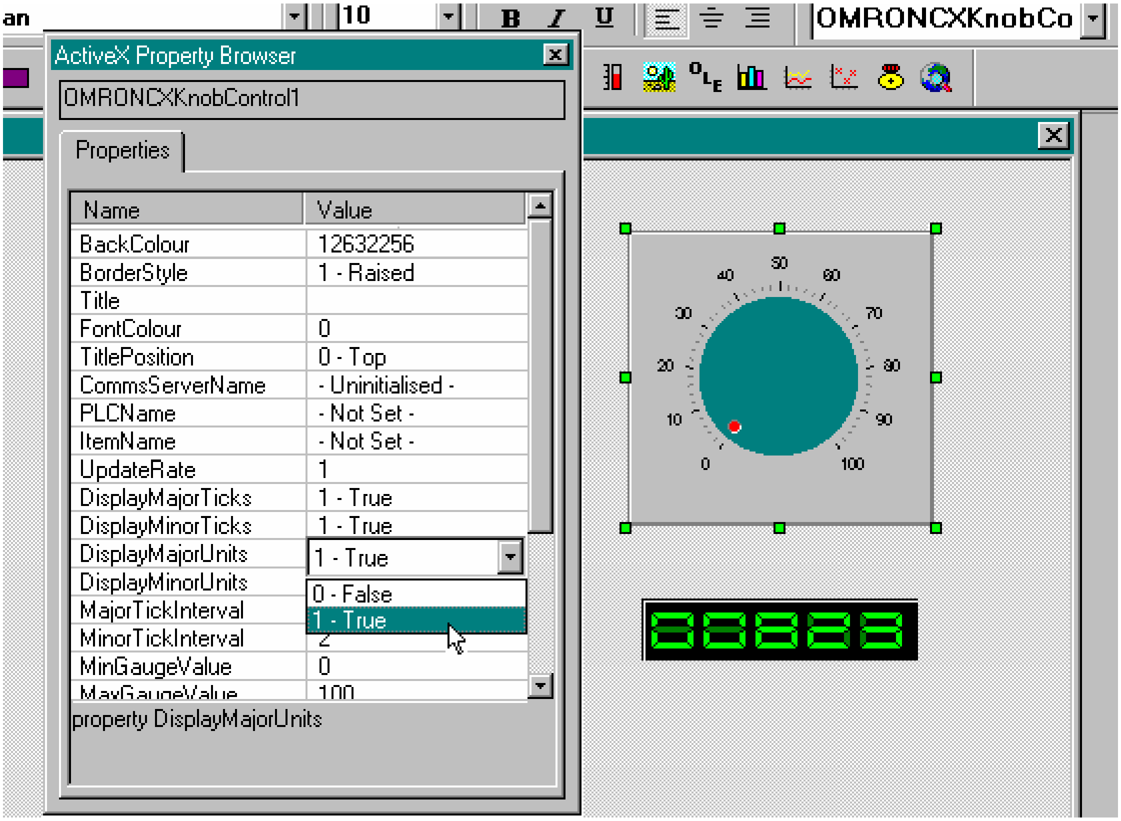
4. Draw a rectangle in the desired position

Editing Properties at Design Time

The properties of ActiveX objects can be edited during the design stage using the following procedure:

1, 2, 3…

1. Open the ActiveX Property Browser by clicking the ActiveX Property Browser button in the toolbar or by selecting the ActiveX Property Browser option from the Utilities menu.
2. With the ActiveX Property Browser displayed select the appropriate ActiveX control. This will list the full range of property names available for that control in the browser, and their values.
3. The Value of each Name can be changed as required by clicking in the value box and entering the new value. Some options require you to enter specific information, others provide a choice of entries from a drop down menu.
4. To edit the values of other ActiveX controls simply click on the control to select it. The values of the previous object will be replaced with those of the new selection.



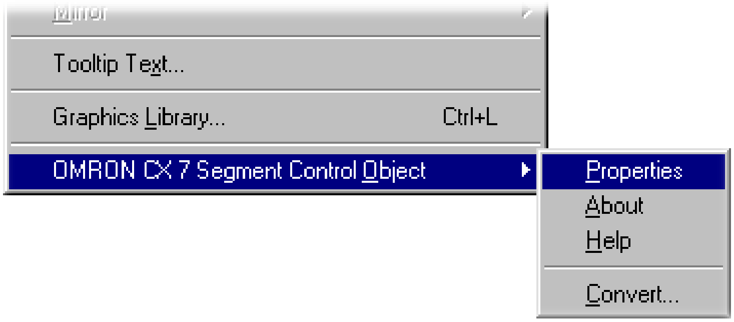
OMRONCXKnobControl1

Name	Value
BackColour	12632256
BorderStyle	1 - Raised
Title	
FontColour	0
TitlePosition	0 - Top
CommsServerName	- Uninitialised -
PLCName	- Not Set -
ItemName	- Not Set -
UpdateRate	1
DisplayMajorTicks	1 - True
DisplayMinorTicks	1 - True
DisplayMajorUnits	1 - True
DisplayMinorUnits	
MajorTickInterval	
MinorTickInterval	2
MinGaugeValue	0
MaxGaugeValue	100

0 - False
1 - True

property DisplayMajorUnits

In addition to editing properties with the Property Browser, many ActiveX controls support their own custom Property Pages. These may be accessed in design time by either double clicking the control, or right clicking the control and selecting Properties from the Object's popup menu:



Reading and Writing Properties at Run Time

ActiveX properties can be read and written at runtime, for example to change values or colours as required. This can be achieved using the CX-Supervisor script functions GetProperty and PutProperty, or alternatively in VBScript using the normal dot syntax for example

```
@VBSCRIPT
Display1.Value = 100
@ENDIF
```

For more details see the *CX-Supervisor Script Language Reference Manual*.

Calling Methods at Run Time

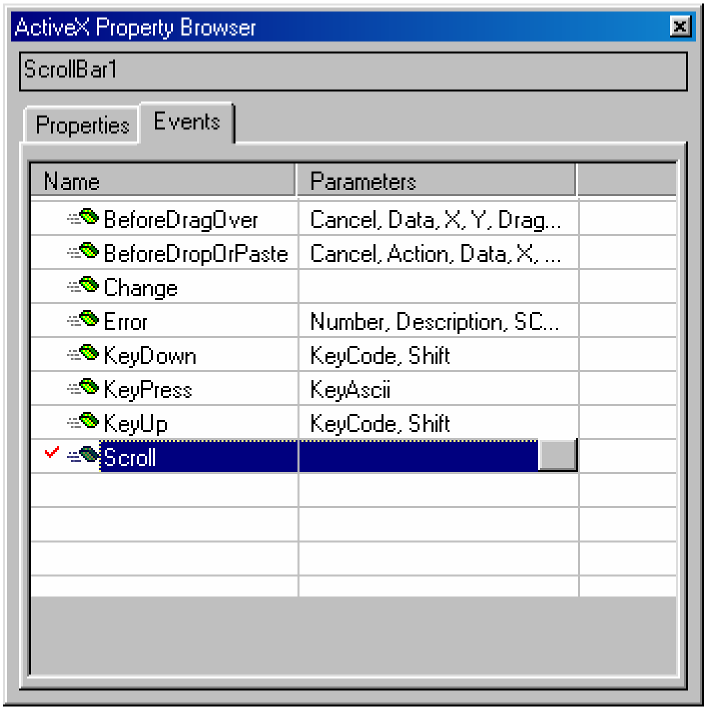
ActiveX methods can also be called at runtime. This is achieved by using the Supervisor script function Execute or alternatively in VBScript using the normal dot syntax for example

```
@VBSCRIPT
CommonDialog1.ShowOpen
@ENDIF
```

Again for more details see the *CX-Supervisor Script Language Reference Manual*.

Responding to Events

Some ActiveX components are written to generate events on certain conditions, like mouse clicking or user input or error conditions. You can write a script to execute whenever any event occurs. These scripts are defined as subroutines in the page initialisation script as they may be called any time the page is open. To easily add these subroutines, from the ActiveX property browser, click the ‘Events’ tab. This shows all the event types for this control and any parameters the event may pass, for example the code number of the key pressed. Select the event name to add or edit the script for, and click the square edit button.



ScrollBar1

Name	Parameters	
BeforeDragOver	Cancel, Data, X, Y, Drag...	
BeforeDropOrPaste	Cancel, Action, Data, X, ...	
Change		
Error	Number, Description, SC...	
KeyDown	KeyCode, Shift	
KeyPress	KeyAscii	
KeyUp	KeyCode, Shift	
✓ Scroll		

Note: In previous versions Event scripts could be added from the Animation Editor but the method above provides more efficiency as all event scripts are loaded just once on page initialisation.

CHAPTER 6
Projects

This chapter describes the concept of projects. It details procedures for creating and amending projects, and the process of associating users with projects. It also deals with the process of compiling a project.

Overview

A CX-Supervisor application consists of a number of pages linked together. The pages may contain passive or active graphics, text or animations, and may be grouped together logically to form a project. A project may consist of many pages, or simply a single page. Projects may be built and tested within the CX-Supervisor development environment, and run “stand-alone” under the CX-Supervisor run-time environment.

Only one project at a time may be open for editing within the CX-Supervisor development environment. An attempt to open a second project forces CX-Supervisor to close the current project (prompting to save changes where there is unsaved information), and open the second project.

Runtime Editions

The Development package can create applications for different Runtime platforms. The target system uses the same CX-Supervisor 'Runtime' software but needs a USB Dongle that matches the chosen target. The type of project is either chosen on creation or changed from the Target Settings dialog.

CX-Supervisor Machine Edition

For most HMI and visulisation projects. These projects only require a Machine Edition USB dongle for the Runtime.

CX-Supervisor PLUS

Where features additional to the Machine Edition are required. These projects require a PLUS USB dongle for the Runtime.

Feature	Machine Edition	PLUS
ActiveX	Yes	Yes
VBScript	Yes	Yes

Recipes	Yes	Yes
Alarms	300	5000
Animations	Yes	Yes
Max no. Devices (PLCs etc.)	15	256
OPC Connections	Yes	Yes
Max user defined Points	500	8000
Max Regular Interval Scripts	10	100
Max no. Pages	100	500
Supported Databases	MS Access	SQL, ODBC, MS Access, MS Excel, dBase, CSV

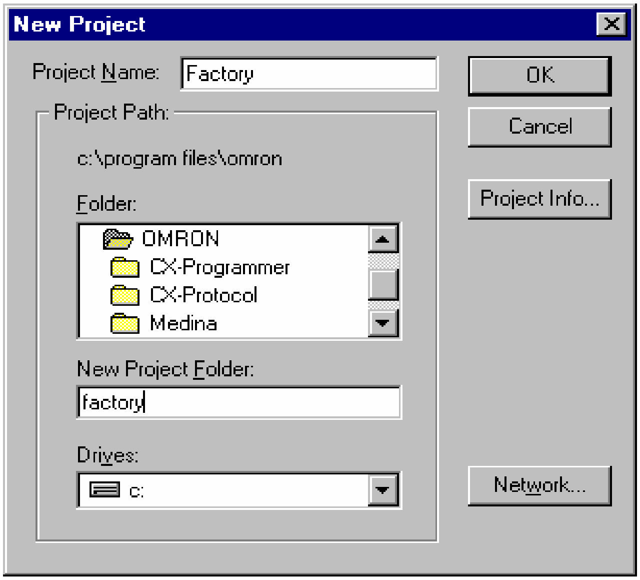
Note: Once the limits shown are reached, no more of those objects can be created.

Note: Remember that although either target can be chosen easily, you must purchase the correct USB Dongle in order to run

Note: The CX-Supervisor Runtime no longer recognises Software Tokens, Hardlock (parallel port) dongles or USB Dongles from older versions. Only USB Dongles for this version will active the Runtime

Creating a Project

To create a new project within CX-Supervisor, select *New* followed by either *Machine Edition project* or *CX-Supervisor PLUS project*. CX-Supervisor displays the following dialog:



New Project

Project Name: Factory

Project Path:

c:\program files\omron

Folder:

OMRON
CX-Programmer
CX-Protocol
Medina

New Project Folder: factory

Drives: c:

1, 2, 3…

1. Locate the parent directory in which the project directory is to reside using the *Folder:* and *Drives:* fields.
2. Enter a name for the project directory in the *New Project Folder:* field. If a name is entered, the folder is created and used; otherwise the selected folder is used. The directory name must be no more than 8 characters and may only contain characters valid for use in MS-DOS file names.
3. Enter a name for the project in the *Project Name:* field. The project name must be no more than 8 characters and may only contain characters valid for use in DOS file names.
4. Click the **OK** pushbutton on the New Project dialog to create the project.

Alternatively, click the **Project Info** pushbutton to open the Project Information dialog, and enter a title and some descriptive text for the new project. For details of using the Project Information dialog refer to chapter 6, Project Information. Click the **OK** pushbutton on the New Project dialog to create the project.

Note: If the PC installation of CX-Supervisor is on a networked machine, a Network pushbutton is added to the dialog. For further information on the function of the Network dialog, refer to the Microsoft Windows User Guide.

Amending a Project

To amend an existing project it must first be opened (assuming it is not already open). To open a project select *Open* from the *Project* menu. CX-Supervisor displays the Open dialog. This is a standard Windows dialog and usage depends on your operating system. Please consult your Microsoft documentation

1, 2, 3…

1. Locate the drive and directory where the desired project is stored.
2. Select the desired project from the list presented.
3. Click the **Open** pushbutton to load the project.

Once the project is loaded, the various editing tasks required may be carried out (such as editing page or graphics formats), as described in the other chapters of this manual.

Saving a Project

Once a project has been created it is wise to save it. It is good practice to ensure that projects are saved regularly, in case of an event such as a power failure. To save a project, select *Save* from the *Project* menu.

If this is the first time the project has been saved, the Save As dialog is displayed.

1, 2, 3…

1. Move to the location where the project is to be stored.
2. Ensure that the *Save as Type:* field is set to *CX-Supervisor Projects (*.SCS).*
3. Enter a name for the project. The name under which the project was created is offered as a default.
4. Click the **Save** pushbutton to save the project.

Note: Subsequent saves do not cause the Save Project As dialog to be displayed.

Printing a Project

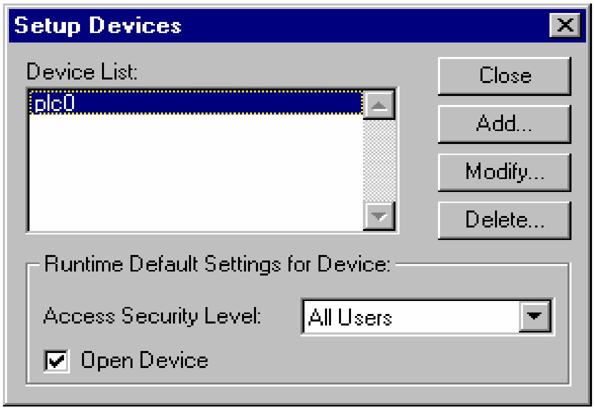
All of the printable views of the project can be printed together, including page layouts and all scripts. This can be useful as a development aid, for maintenance and for project documentation.

1, 2, 3…

1. Start CX-Supervisor and load the project to be printed.
2. From the options dialog, select the items to print and press OK.
3. Select the printer to print to and press OK.

Device Configuration

To amend the device configuration or create connections to a PLC or temperature controller, click the **Device Setup** button from the toolbar. This results in the Setup Devices dialog being displayed.



Setup Devices

Device List: plc0

Runtime Default Settings for Device:

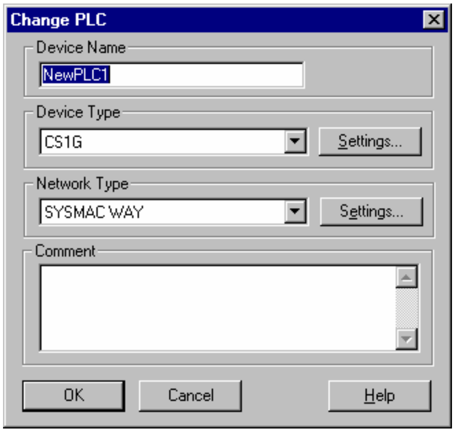
Access Security Level: All Users

☑ Open Device

Creating a PLC Connection

A new device can be added by clicking on the **Add** pushbutton on the Setup Devices dialog.

Note: CX-Supervisor calls an external application to change PLC information. The functions described in the following paragraphs may differ slightly depending upon which application and version has been invoked.



Change PLC

Device Name: NewPLC1

Device Type: CS1G

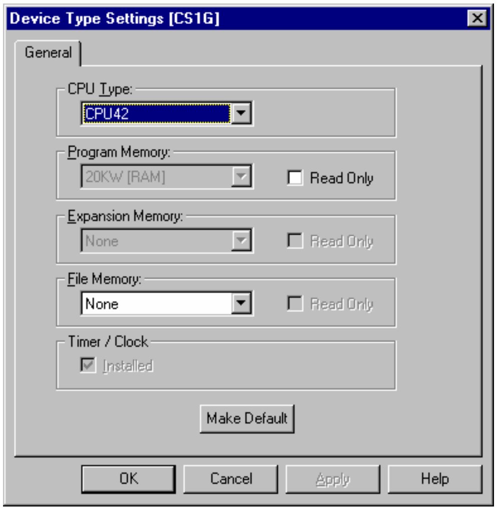
Network Type: SYSMAC WAY

Comment:

A name can be assigned for the device in the *PLC Name* field.

Select the PLC from the *Device Type:* field. To add a temperature controller, select a temperature controller from the *Device Type:* field, e.g. E5AF-AH. See also Chapter 14 for details on other models.

Clicking the **Setup** pushbutton results in the Device Type Settings dialog being displayed allowing the device type of the PLC to be configured.



Device Type Settings [CS1G]

General

CPU Type: CPU42

Program Memory: 20KW [RAM] □ Read Only

Expansion Memory: None □ Read Only

File Memory: None □ Read Only

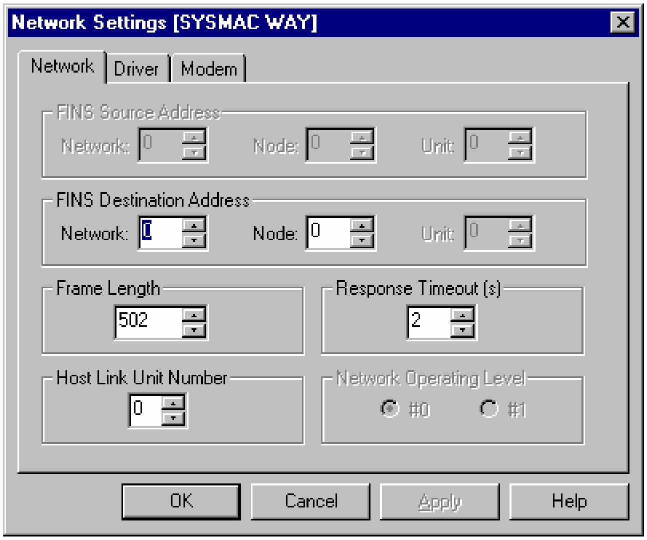
Timer / Clock ☑ Installed

Note: The possible settings for PLC configuration depend upon the type of PLC selected (this applies also to the Read Only and Timer/Clock fields).

On completion, click the **OK** pushbutton to continue, or the **Cancel** pushbutton to abort the operation. Values specified may be set as default by clicking the **Make Default** pushbutton.

A network may be specified for the selected PLC, by selecting from the *Network:* field. The networks available are dependent on the device type selected.

Clicking the ***Setup*** pushbutton results in the Network Settings dialog being displayed.



Network Settings [SYSMAC WAY]

FINS Source Address

Network: 0 Node: 0 Unit: 0

FINS Destination Address

Network: 0 Node: 0 Unit: 0

Frame Length: 502

Response Timeout (s): 2

Host Link Unit Number: 0

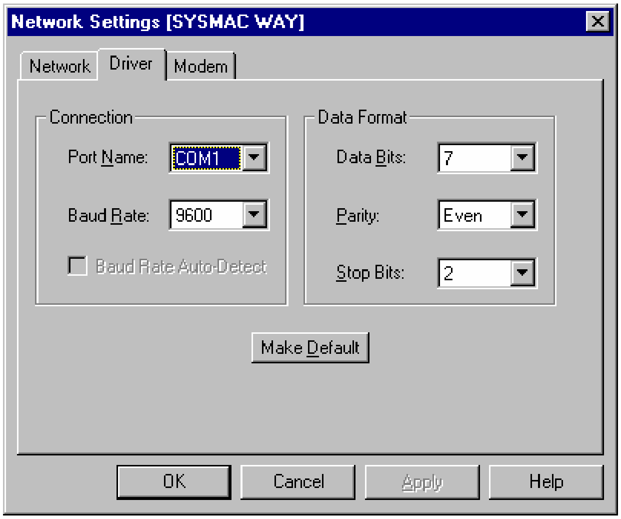
Network Operating Level: ● #0 ○ #1

The *Unit Number* is the identifier for the network being configured.

The *Destination Network Address* and *Destination Node Number* identify the connection point to the network.

A PLC can be selected to act as a gateway to the PLC being edited; this list is restricted to the PLCs contained in the current project.

Selecting the *Driver* tab results in the Driver Configuration view being displayed; this part of the Network Settings dialog helps to ensure that data is transmitted correctly over the network.



Network Settings [SYSMAC WAY]

Connection

Port Name: COM1

Baud Rate: 9600

□ Baud Rate Auto-Detect

Data Format

Data Bits: 7

Parity: Even

Stop Bits: 2

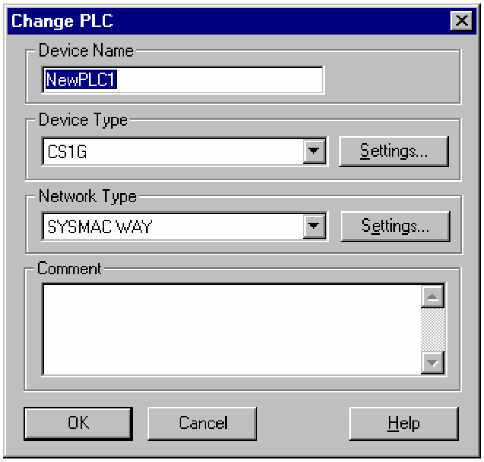
Note: The possible settings for the Baud Rate, Parity, Data Bits, and Stop Bits fields depend upon the port selected.

If a timeout occurs, the communication is not complete. Where necessary, increase the *Timeout Offset* value (in milliseconds) to ensure that the device does not cause a timeout.

Values specified may be set as default by clicking the **Make Default** pushbutton.

Modifying a PLC Connection

From the Setup Devices dialog, a PLC name may be modified by selecting the PLC name from the *Device List* on the Setup Devices dialog, and clicking the **Modify** pushbutton. This results in the Change PLC dialog being displayed.



Change PLC

Device Name: NewPLC1

Device Type: CS1G

Network Type: SYSMAC WAY

Comment

A new name can be entered in the *PLC Name:* field. If an invalid PLC name is entered, an error message is displayed on clicking the **OK** pushbutton.

Removing a PLC Connection

From the Setup Devices dialog, a PLC name may be removed from the *PLC Name:* field by selecting the PLC name from the *Device List* on the Setup Devices dialog, and clicking the **Delete** pushbutton. This results in a confirmation dialog being displayed. Click the **Yes** pushbutton to remove the PLC from the list, or the **No** pushbutton to abort the delete operation.

Note: A PLC cannot be renamed, deleted or edited if it is currently open for communications.

Accessing PLC Connection in Runtime

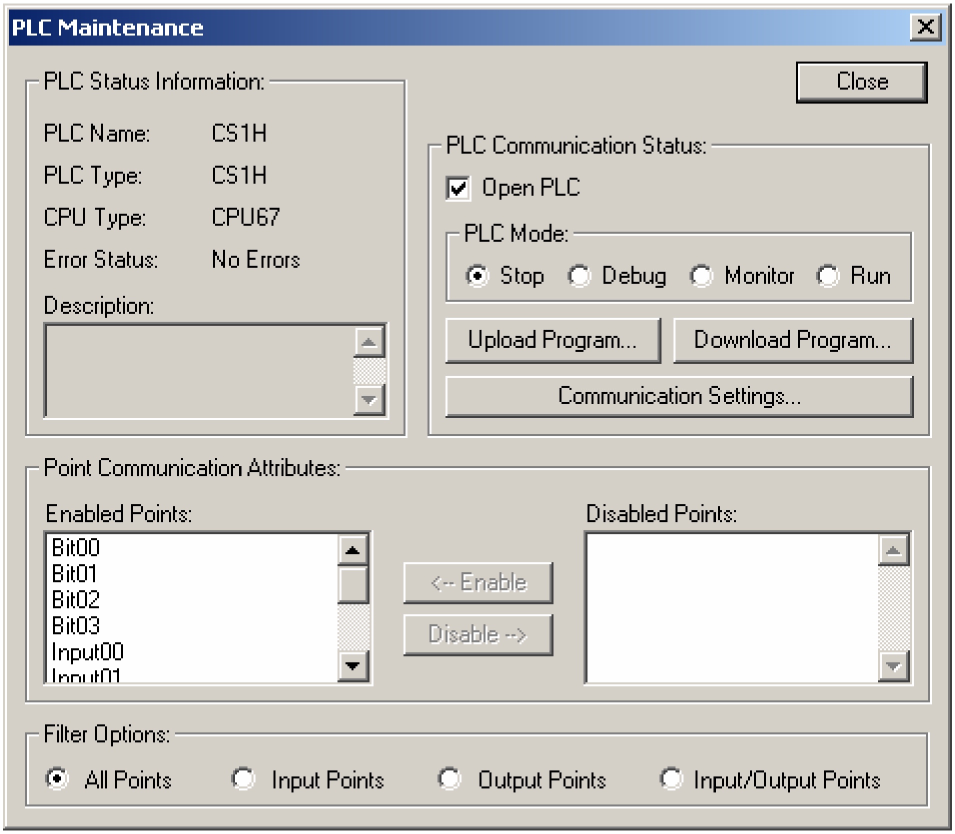
It is possible to reconfigure PLCs at runtime via the right mouse button floating menu option *PLC Maintenance List*. The list of currently configured PLCs is displayed via the PLCs in Project dialog box. On selection of a PLC, the PLC Information dialog is displayed, which allows the user to change the PLC configuration dynamically during runtime.

The *Open PLC* option provides the capability to toggle the communications status of the PLC.

The *PLC Mode* options switch the current mode of operation of the PLC between Stop, Debug, Monitor and Run.

- Stop mode halts the PLC program execution allowing the PLC to be programmed.
- Debug mode allows for the single stepping of program execution. This mode is reserved for CV-series PLCs only.
- Monitor mode operation allows normal PLC program execution and modification of data.
- Run mode operation allows normal PLC program execution. No data in any of the PLC memory areas can be changed.

It is possible to individually enable / disable point communications from the PLC Maintenance dialog.



PLC Maintenance

PLC Status Information:

PLC Name: CS1H

PLC Type: CS1H

CPU Type: CPU67

Error Status: No Errors

Description:

PLC Communication Status:

☑ Open PLC

PLC Mode: ● Stop ○ Debug ○ Monitor ○ Run

Point Communication Attributes:

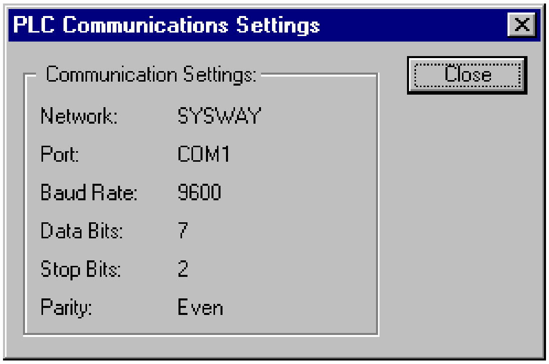
Enabled Points:

Bit00
Bit01
Bit02
Bit03
Input00
Input01

Disabled Points:

Filter Options: ● All Points ○ Input Points ○ Output Points ○ Input/Output Points

Selection of the *Communication Settings* option displays the Communication Settings dialog, showing the current settings for the PLC:



PLC Communications Settings

Communication Settings:

Network: SYSWAY

Port: COM1

Baud Rate: 9600

Data Bits: 7

Stop Bits: 2

Parity: Even

It is possible to configure default settings for the runtime via the *PLC Runtime Default Settings* fields in the Setup PLCs dialog.

Communications to the PLC can be enabled on startup via the *Open PLC* option.

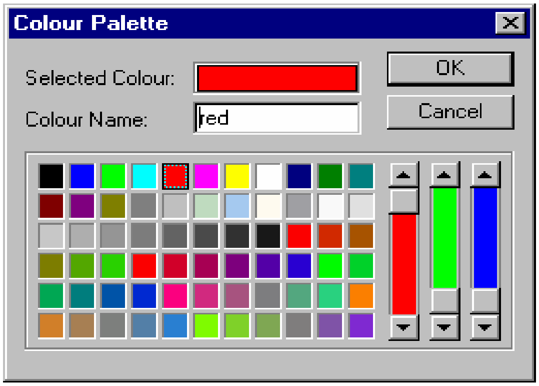
Settings

General Settings

Colour Palette

A specific set of colours may be defined for use within a project. This may include a maximum of 66 unique colours. To adjust the colours, select *General Settings* from the *Projects* menu, and select *Colour Palette* from the sub-menu.

CX-Supervisor displays the following dialog:



Colour Palette

Selected Colour:

Colour Name: red

1, 2, 3…

1. Select the coloured square which requires editing.
2. Adjust its colour values using the Red, Blue and Green sliders.
3. When the desired colour has been created, enter a name in the *Colour Name:* field.
4. Click the **OK** pushbutton when all colours requiring editing have been edited.

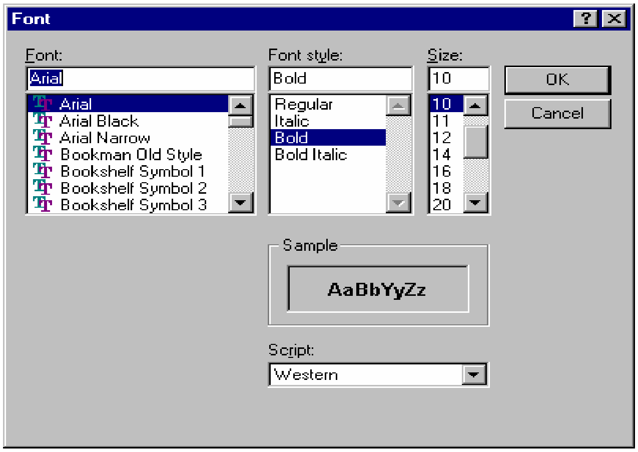
Note: The mixing of colours on screen differs from the mixing of, for example, colour paint. Under normal circumstances, the more of a colour which is added to a mix, the darker it gets. The opposite is the case for mixing colours on screen, i.e. Black consists of no Red, no Green and no Blue, whilst White consists of full Red, full Green and full Blue.

Note: The first sixteen colours cannot be mixed.

Note: Using a 16 colour-based screen resolution (consult the Microsoft Windows documentation for further information) the seventeenth colour onwards are dithered from the sixteen base colours. Higher colour-based resolutions are not dithered.

Default Button Font

The *Default Button Font* option, which can also be selected from the *General Settings* sub-menu of the *Project* menu, displays the standard font dialog:



Font: Arial

Font style: Bold

Size: 10

Sample: AaBbYyZz

Script: Western

This dialog is used to set the default font specification to be used for all text displayed on push buttons created using the graphics editor. Any button can also have its font changed from the default, using the relevant Wizard in the graphics editor. See chapter 4, Objects.

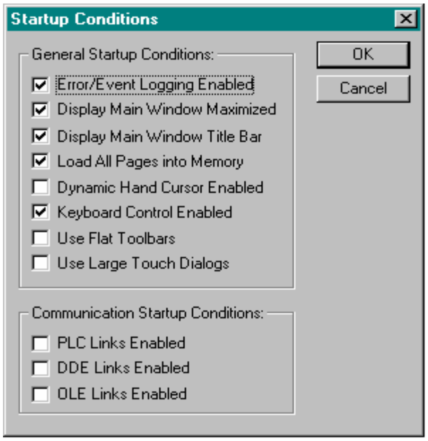
Runtime Settings

The *Runtime Settings* option is selected from the *Project* menu.

The settings discussed in the following paragraphs help to configure runtime environment applications and have no effect in the development environment. Settings that affect the development environment are discussed in chapter 2, Pages.

Startup Conditions

To open the Startup Conditions dialog, select *Startup Conditions* from the *Runtime Settings* menu. The dialog is displayed as follows:



General Startup Conditions:

☑ Error/Event Logging Enabled
☑ Display Main Window Maximized
☑ Display Main Window Title Bar
☑ Load All Pages into Memory
□ Dynamic Hand Cursor Enabled
☑ Keyboard Control Enabled
□ Use Flat Toolbars
□ Use Large Touch Dialogs

Communication Startup Conditions:

□ PLC Links Enabled
□ DDE Links Enabled
□ OLE Links Enabled

Click the settings to enable/disable *General Startup Conditions* and *Communication Startup Conditions*. Click the **OK** pushbutton to accept the settings or the **Cancel** pushbutton to abort.

When Keyboard Control is enabled, a focus rectangle becomes visible around the currently selected object. The cursor keys and <Tab> key can then be used to navigate around the selectable objects on a page. Once an object is selected certain actions can then be applied, depending on the object's type. The most common action is to simulate a left mouse button click.

The following list shows all the possible facilities with Keyboard Control enabled.

- Moving Around Selectable Objects Using Cursor Keys. The cursor keys can be used to move around the objects in the respective direction.
- With an object selected, if the right cursor key is pressed, then the closest object is selected from within an area bounded by lines drawn diagonally upwards and diagonally downwards (and to the right) from the centre of the object. If no object is found then the current object remains selected. A similar rule also applies when using the left cursor key, and the up and down cursor keys.

  Note: Selectable items must have a left mouse button event defined or have a default action.

- *Moving Around Selectable Objects Using <Tab> Key*. The <Tab> key can be used to move around all the objects in the order in which they are drawn (this can be varied using the 'raise object' and 'lower object' editing facilities). The <Shift>+<Tab> key can be used to move around the objects in the reverse order.
- To select items in a specific tab order: use the <Tab> key to move around the page items in a specific order, using the 'move to top' feature. Start at object number 1 and end with the last object, i.e. if obj1, obj2 and obj3 then use 'move to top' on obj1 first, followed by obj2 and then by obj3; this gives the tab order obj1, obj2, obj3.
- *Changing Between Pages*. The <Ctrl>+<Tab> key can be used to change between pages.

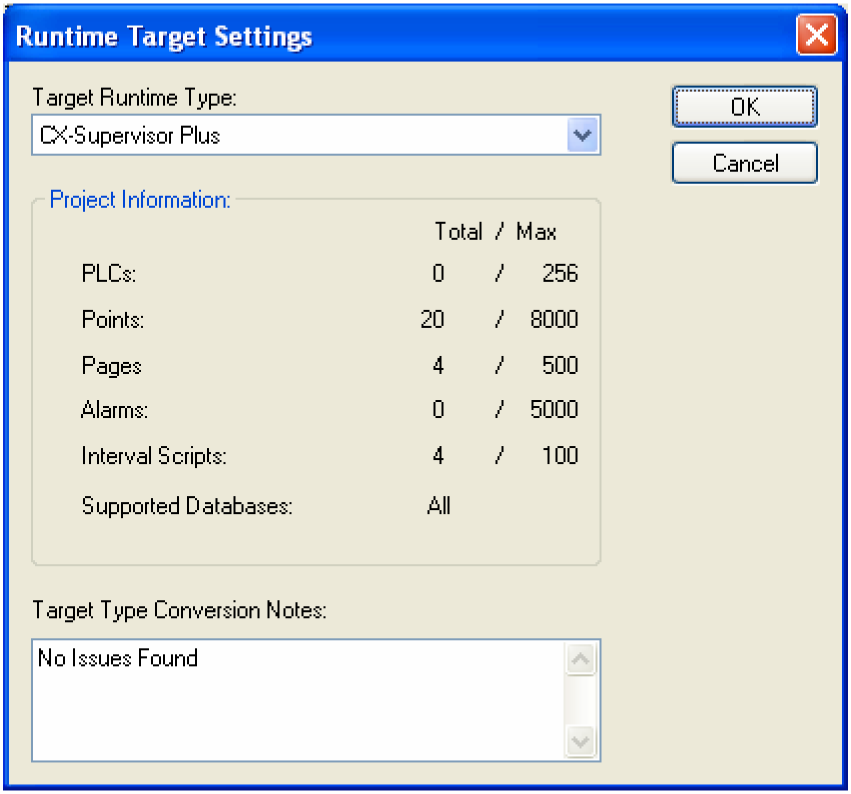
- *Simulating a Left Mouse Button Click*. Objects that have scripts attached (such as a pushbutton) or objects that perform a particular action when clicked (such as toggle buttons) can be executed by using the <Enter> key.
- *Slider Operation*. When a slider object is selected, the plus (+) and minus (-) keys can be used to increment or decrement its value respectively. This also applies to the sliders on trend graphs.
- *Obtaining the Runtime Floating Menu*. To display the Runtime Floating menu, use either the <Shift>+<F10> key combination or use the Windows right mouse button key, which is next to the <Ctrl> key on the right hand side of the keyboard.
- *Standard Windows Keys*. Some standard Windows key combinations are as follows:

<Alt>+<-> (hyphen)	Used to access the child window control box at the top left hand side of the dialog.
<Alt>+<Spacebar>	Used to access the main window control box at the top left hand side of the dialog.
<Alt>+<F4>	Used to close down the current application.

- *Using Runtime Alarm, Error and Recipe Viewer*. These can be invoked from the Runtime Floating menu (see above). To access their functionality use the <Tab> key to move from toolbar button to toolbar button, and <Enter> to press a button. The up and down cursor keys can be used to scroll the displayed list. To close them (or to move or resize them) use the standard <Alt>+<-> (hyphen) key combination to access their menus.
- *Other Notes*. In Project level scripts, it is possible to define ‘OnKeyPress’ scripts, which are attached to the cursor keys. If Keyboard Only operation is set, then it is not possible to execute a script attached to one of the cursor keys, since they are being used for navigating around the selectable objects on the page. Alternative additional keys have been added to compensate for this. They are the number pad cursor keys (i.e. 2, 4, 6, and 8). However, these can only be used when the <Num Lock> key is on.

Target Settings

The project target can be changed between Machine Edition and PLUS with this dialog:



Runtime Target Settings

Target Runtime Type: CX-Supervisor Plus

Project Information:

	Total	/	Max
PLCs:	0	/	256
Points:	20	/	8000
Pages	4	/	500
Alarms:	0	/	5000
Interval Scripts:	4	/	100
Supported Databases:	All		

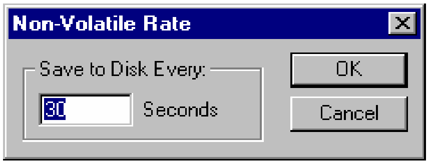
Target Type Conversion Notes:

No Issues Found

Any conversion problems are reported in the Notes at the bottom and conversion is prevented if it is not allowed.

Non-Volatile Rate

The Non-Volatile rate specifies how often the value of points flagged as ‘non-volatile’ are saved to disk, in seconds. The latest disk values are used to re-initialise the point values when CX-Supervisor is restarted. To adjust this value select *Runtime Settings* from the *Projects* menu, and select *Non-Volatile Rate* from the sub-menu. CX-Supervisor displays the following dialog:



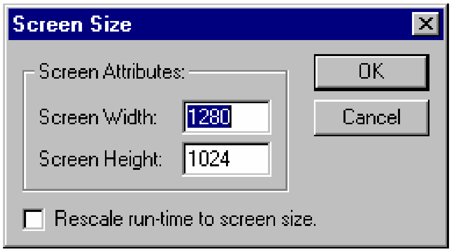
Non-Volatile Rate

Save to Disk Every: 30 Seconds

Enter a new value for the Non-Volatile Rate in the *Seconds* field and click the **OK** pushbutton.

Screen Size

To open the Screen Size dialog, select *Screen Size* from the *Runtime Settings* menu. The dialog is displayed as follows:



Screen Size

Screen Attributes:

Screen Width: 1280

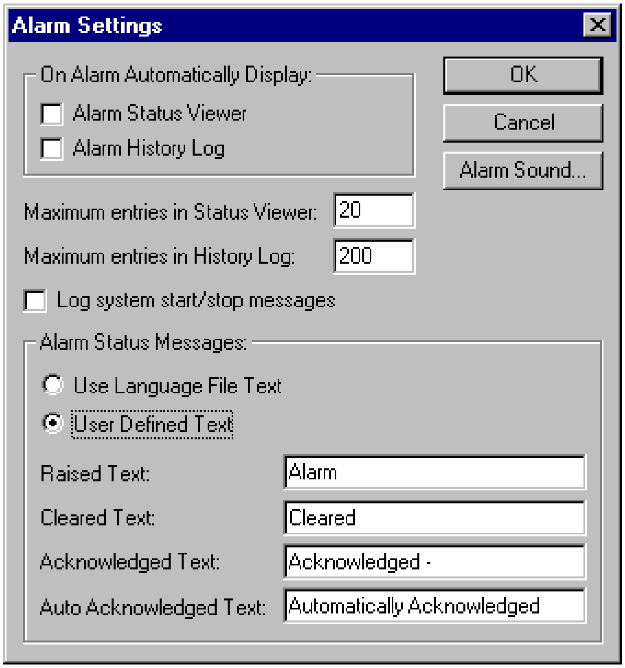
Screen Height: 1024

□ Rescale run-time to screen size.

Type valid values into the *Screen Width:* and *Screen Height:* fields. These values are used to define the screen size of the runtime application, but do not affect Maximise and Minimise dialog states. By enabling the *Rescale run-time to screen size* option the screen rescales itself to take into account the resolution of the runtime system. Click the **OK** pushbutton to accept the setting or the **Cancel** pushbutton to abort.

Alarm Settings

To open the Alarm Settings dialog select *Alarm Settings* from the *Runtime Settings* menu. The dialog is displayed as follows:



Alarm Settings

On Alarm Automatically Display:

□ Alarm Status Viewer

□ Alarm History Log

Maximum entries in Status Viewer: 20

Maximum entries in History Log: 200

□ Log system start/stop messages

Alarm Status Messages:

○ Use Language File Text

● User Defined Text

Raised Text: Alarm

Cleared Text: Cleared

Acknowledged Text: Acknowledged -

Auto Acknowledged Text: Automatically Acknowledged

In the *On Alarm Automatically Display* area set the *Alarm Status Viewer* and *Alarm History Viewer* settings as required. When these options are set, the Current Alarms viewer or Alarm History viewer (respectively) is automatically displayed in runtime when an alarm occurs.

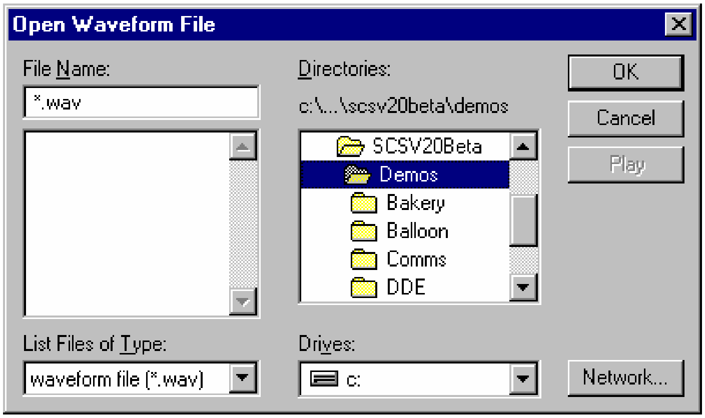
Set the *Maximum entries in Status Viewer* and *Maximum entries in History Log* fields to the desired values. The numbers specify how many messages are displayed in the respective viewer dialogs in runtime. Select the *Log system start/stop messages* setting to ON if required.

Note: If more messages exist than are requested to be displayed, the most recent messages are displayed in preference to older messages.

The values shown above are sensible defaults.

In the *Alarm Status Messages* area, if the *User Defined Text* option is set, default messages for *Raised Text, Cleared Text, Acknowledge Text,* and *Auto Acknowledge Text* can be applied. If the *Use Language File Text* option is set, the alarm status messages default to the supplied language file.

The **Alarm Sound** pushbutton allows the selection of an audible warning which may be played when an alarm occurs in runtime. The Open Waveform File dialog is shown below:



Open Waveform File

File Name: *.wav

Directories: c:\...\scsv20beta\demos

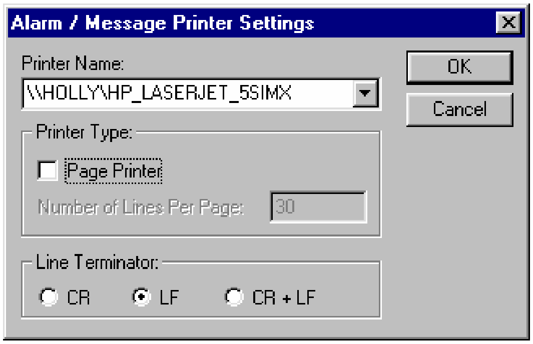
List Files of Type: waveform file (*.wav)

Drives: c:

Note: If the PC installation of CX-Supervisor is on a networked machine, a Network pushbutton is added to the dialog. For further information of the function of the Network dialog, refer to the Microsoft Windows User Guide.

Alarm/Message Printer Settings

To open the Alarm/Message Printer Settings dialog select *Alarm/Message Printer Settings* from the *Runtime Settings* menu. The dialog is displayed as follows:



Alarm / Message Printer Settings

Printer Name: \\HOLLY\HP_LASERJET_5SIMX

Printer Type:

□ Page Printer

Number of Lines Per Page: 30

Line Terminator:

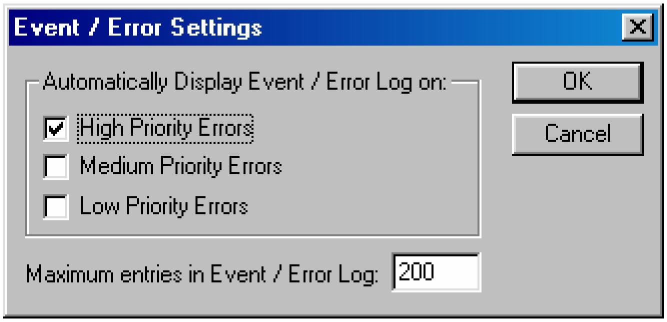
○ CR ● LF ○ CR + LF

1, 2, 3…

1. Select the target printer in the *Printer Name:* field.
2. If the printer type is a Page Printer, ensure the *Page Printer* setting is set ‘ON’ and the appropriate number of lines per page are specified in the *Number of Lines Per Page:* field.
3. Specify a *Line Terminator:* setting, either *CR* (Carriage Return), *LF* (Line Feed) or *CR + LF*.
4. Click the **OK** pushbutton to accept the changes, or the **Cancel** pushbutton to abort.

Event/Error Messages

To open the Event/Error Settings dialog select *Event/Error Settings* from the *Runtime Settings* menu. The dialog is displayed as follows:



Event / Error Settings

Automatically Display Event / Error Log on:

☑ High Priority Errors

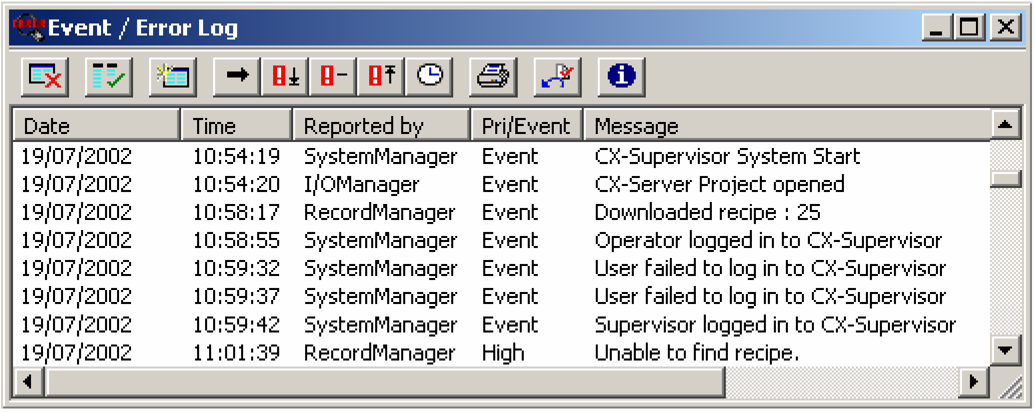
□ Medium Priority Errors

□ Low Priority Errors

Maximum entries in Event / Error Log: 200

Set the *Maximum entries in Event/Error* field and the *Automatically Display Event/Error Log on:* options as desired. Click the **OK** pushbutton to accept the changes, or the **Cancel** pushbutton to abort.

In the runtime environment, the Event/Error Log dialog can be displayed at any time.



Date	Time	Reported by	Pri/Event	Message
19/07/2002	10:54:19	SystemManager	Event	CX-Supervisor System Start
19/07/2002	10:54:20	I/OManager	Event	CX-Server Project opened
19/07/2002	10:58:17	RecordManager	Event	Downloaded recipe : 25
19/07/2002	10:58:55	SystemManager	Event	Operator logged in to CX-Supervisor
19/07/2002	10:59:32	SystemManager	Event	User failed to log in to CX-Supervisor
19/07/2002	10:59:37	SystemManager	Event	User failed to log in to CX-Supervisor
19/07/2002	10:59:42	SystemManager	Event	Supervisor logged in to CX-Supervisor
19/07/2002	11:01:39	RecordManager	High	Unable to find recipe.

The events can be sorted, based on a designated field type: date, time, reported by, priority or associated message. By selecting the *Date* field, the events are sorted alphanumerically by date. The *Time*, *Reported by*, *Pri/Event* and *Message* fields react in the same way. The Event/Error Log display shows the log listed in date order.

The widths of the alarm fields can be widened or narrowed as required using the mouse to drag the column boundaries.

There are a number of toolbar buttons that aid the use of the Event/Error Log.

The **Disable Error Updates** button, once pressed, stops further events occurring in the runtime environment being added to the log. The Event/Error Log dialog is still accessible to switch back subsequently.

The **Enable Column Sorting** button allows the format of the Error/Event Log dialog to be continually updated when subsequent errors or events are added.

The **Clear Event/Error Log** button clears all entries from the log.

The **Display All Errors/Events** button lists all events and errors irrespective of priority. The list is sorted according to the *Date*, *Time*, *Reported by*, *Pri/Event* or *Message* field.

The **Low Priority** button lists all errors designated as a low priority only. Other errors and events are not deleted from the log; they are merely not visible.

The **Medium Priority** button lists all errors designated as a medium priority only. Other errors and events are not deleted from the log; they are merely not visible.

The **High Priority** button lists all errors designated as a high priority only. Other errors and events are not deleted from the log; they are merely not visible.

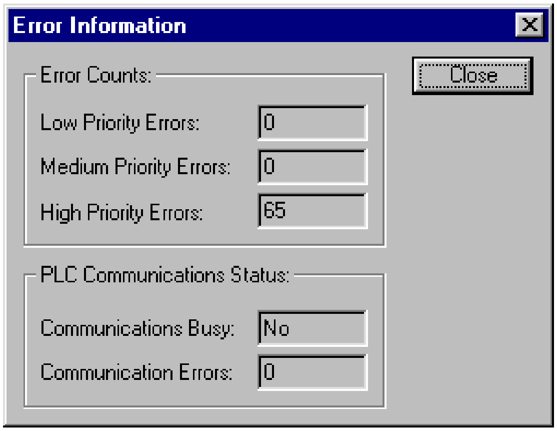
The **Events** button lists all events. Errors, of all priorities, are not deleted from the log;

they are merely not visible. Examples of events are system startup, system shutdown and user security notices.

The **Print** button prints the current contents of the error and event log. Ensure that the printer is correctly set up before printing.

The **Display Result Code Converter** launches the CX-Server error code converter, and if a CX-Server error is currently selected, displays further details about possible causes.

The **Error Information Dialog** button, once pressed, displays a summary of error information, including a detailed count of errors and PLC communication information. Click the **Close** pushbutton to remove this dialog.


Error Information

Error Counts:

Low Priority Errors: 0

Medium Priority Errors: 0

High Priority Errors: 65

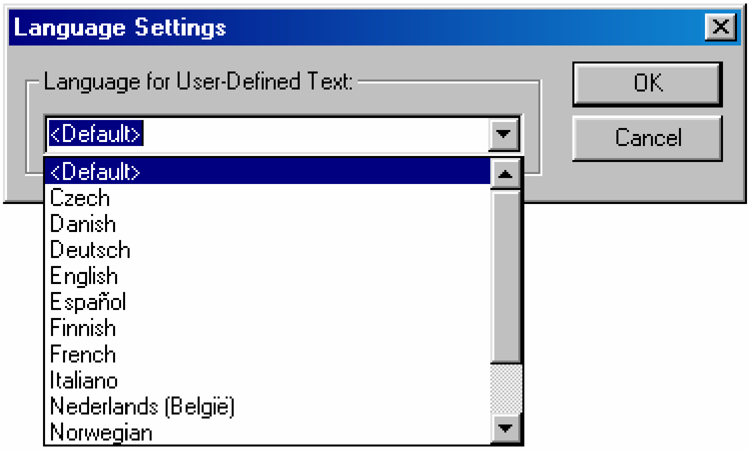
PLC Communications Status:

Communications Busy: No

Communication Errors: 0

Language Settings

The language for user-defined text can be set via the Language Settings dialog. Select *Runtime Settings* from the *Projects* menu, followed by *Language Settings* to display the Language Settings dialog.

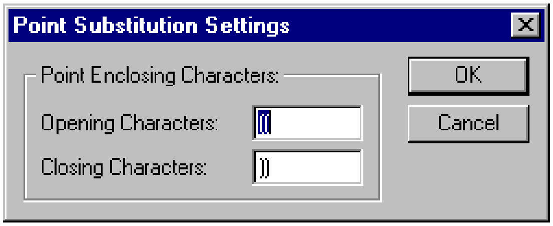


Language for User-Defined Text: <Default>

Select a language from the *Language for User-Defined Text:* field. Click the **OK** pushbutton to accept the settings, or the **Cancel** pushbutton to abort the operation.

Point Substitution Settings

The enclosing characters associated with a report can be changed via the Point Substitution Settings dialog. Once set, these characters must be fixed for all reports generated by the project. Select *Runtime Settings* from the *Projects* menu, followed by *Point Substitution Settings* to display the Point Substitution Settings dialog.



Point Enclosing Characters:

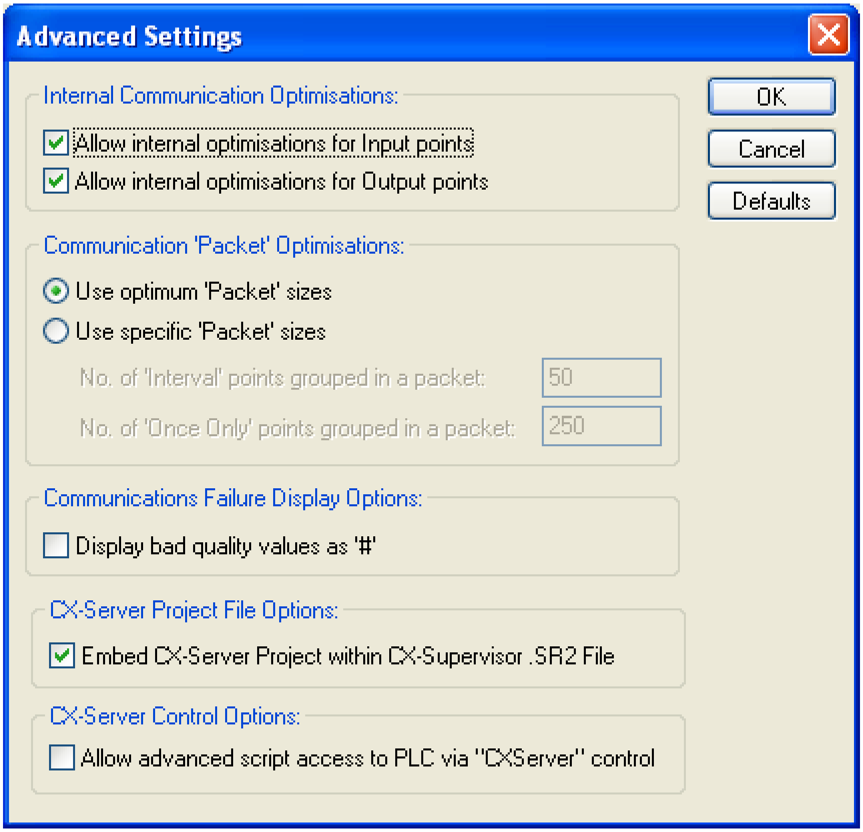
Opening Characters: [[

Closing Characters:]]

Supply enclosing characters in the *Opening Characters:* field and *Closing Characters:* field. Click the **OK** pushbutton to accept the settings, or the **Cancel** pushbutton to abort the operation.

Advanced Settings

To open the Advanced Settings dialog select *Advanced Settings* from the *Runtime Settings* menu. The dialog is displayed as follows:



Internal Communication Optimisations:

☑ Allow internal optimisations for Input points

☑ Allow internal optimisations for Output points

Communication 'Packet' Optimisations:

● Use optimum 'Packet' sizes

○ Use specific 'Packet' sizes

No. of 'Interval' points grouped in a packet: 50

No. of 'Once Only' points grouped in a packet: 250

Communications Failure Display Options:

□ Display bad quality values as '#'

CX-Server Project File Options:

☑ Embed CX-Server Project within CX-Supervisor .SR2 File

CX-Server Control Options:

□ Allow advanced script access to PLC via "CXServer" control

Click the settings to enable/disable *Internal Communication Optimisations* and *Communication 'Packet' Optimisations.*

Bad Quality values, and results of calculations using bad quality values can be shown as '#' characters. This could indicate a communication failure, or Bad Quality status in an OPC Server.

If you may need to option to edit the PLC settings in the runtime without recompiling in the developer then untick the "Embed CX-Server Project within CX-Supervisor .SR2 file" option. This is useful for example if it is possible the COM port driver number may change, remote telephone number or network node number or even the PLC CPU model. These settings can be edited by double clicking the CDM file in Explorer to edit the PLC settings without recompiling the software.

"Allow advanced script access to PLC via "CX-Server" control" adds a new ActiveX control named CXServer (no hyphen) which can be accessed from any script to perform advanced PLC functions. If you already have a graphical object called CXServer then turn this option off.

Click the **OK** pushbutton to accept the settings or the **Cancel** pushbutton to abort.

Runtime Security

Configured Users

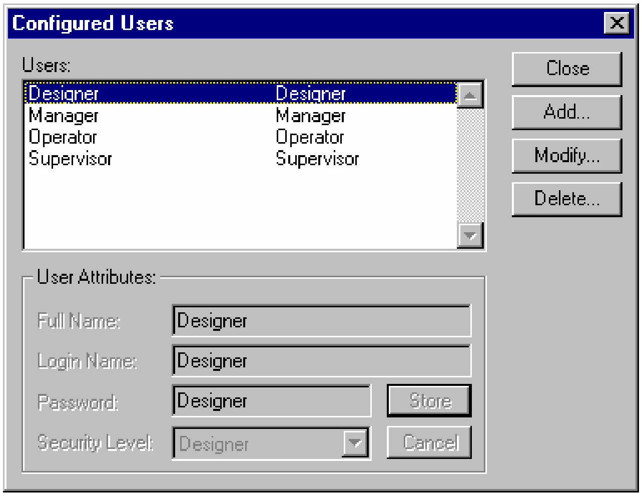
In a runtime application, it is possible to apply security measures in the development environment so that only sufficiently privileged users of the runtime application can access certain elements. It is also possible to housekeep security information within the runtime environment, with privileged users able to add, amend or remove users.

There are four levels of user privilege available in CX-Supervisor:

- Operator level privilege.
- Supervisor level privilege.
- Manager level privilege.
- Designer level privilege.

User configuration in the development environment is handled by the Configured Users dialog. Select *Configured Users* from the *Runtime Security* menu.

The Configured Users dialog is displayed as follows:



Designer	Designer
Manager	Manager
Operator	Operator
Supervisor	Supervisor

Full Name: Designer

Login Name: Designer

Password: Designer

Security Level: Designer

By default, there are four previously defined users listed in the *Configured Users:* field: *Designer* with designer-level privileges, *Manager* with manager-level privileges, *Operator* with operator-level privileges and *Supervisor* with supervisor-level privileges.

To add a new user:

1, 2, 3… 1. Click the **Add** pushbutton. The *User Attributes:* fields and buttons become enabled, and the contents of all fields are cleared, ready for a new user to be added.

2. Type the full name of the new user in the *Full Name:* field.
3. Type a login name in the *Login Name:* field.
4. Type a user password in the *Password:* field. The password must be at least four characters in length.
5. Select a level of privilege from the *Security Level:* drop down list.
6. Click the **Store** pushbutton to add the new user to the *Configured Users:* list, or the **Cancel** pushbutton to abort the operation.

To modify an existing user:

1, 2, 3…

1. Select a user from the *Configured Users:* list and click the **Modify** pushbutton. The *User Attributes:* fields and buttons become enabled, with the contents of all the fields filled with the attributes of the selected user.
2. Amend the full name, login name, password and level of privilege in the same way as adding a new user.
3. Click the **Store** pushbutton to update the user in the *Configured Users:* list, or the **Cancel** pushbutton to abort the operation.

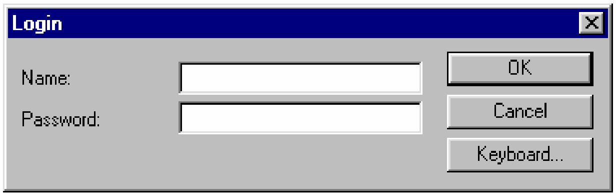
To remove a user from the *Configured Users:* list:

1, 2, 3…

1. Select a user from the *Configured Users:* list and click the **Delete** pushbutton.
2. A confirmation dialog is displayed. Click the **Yes** pushbutton to remove the selected user or the **No** pushbutton to abort the operation.

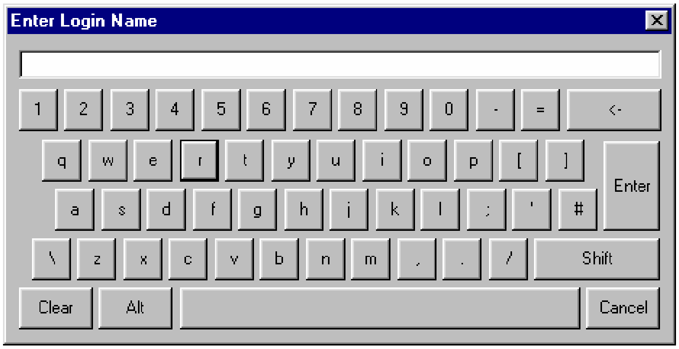
When all user amendments are complete, click the **Close** pushbutton.

In the runtime application, to make use of the designated privilege, the user must log in. Login can be accessed in a variety of ways, and so is dependent on the setup of the runtime application. Some applications may require login as soon as the application is run, others may allow login from a context-sensitive floating menu. Refer chapter 9, Animation for further information. The Login User dialog is as follows:



To log in, enter the login name (not the full name) in the *Login Name:* field and the associated password in the *Password:* field. Any characters typed in the *Password:* field are disguised by a '*' symbol for each character typed. Click the **OK** pushbutton to log in or the **Cancel** pushbutton to abort.

For users without a keyboard to enter login details, select the **Keyboard** pushbutton. The login name and password can be constructed from the subsequent dialog by clicking on each pushbutton in turn, followed by the **Enter** pushbutton to complete.

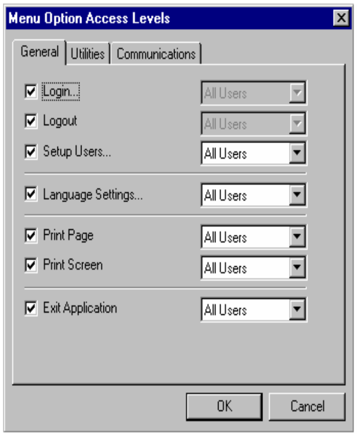


A user can detach from specialised user privileges by logging out. Access is again dependant on the application. There is no dialog associated with logout; once logout is activated privileges are immediately discontinued. Whilst one user is logged in, it is possible for a different user to log in, as prior to login, the current user is automatically logged-out by CX-Supervisor.

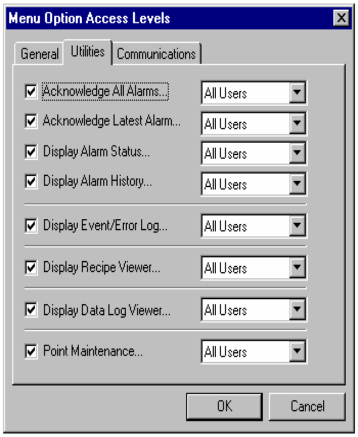
A user with the designer privilege can further amend the user configuration in the runtime environment using a similar dialog to the development environment's Configured Users dialog. As well as typing a password, verification of the password is required, and both fields are disguised by a '*' symbol. This is so that a new user can apply their own password, and that only they know the password. Verification is required to ensure the password was typed correctly in the first instance.

Menu Option Access Levels

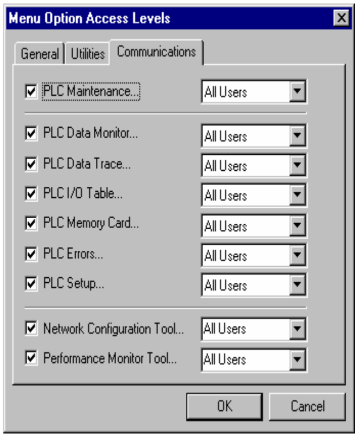
To open the Menu Option Access Levels dialog select *Menu Option Access Levels* from the *Runtime Security* menu. The dialog is displayed as follows:


Menu Option Access Levels — General | Utilities | Communications

☑ Login... All Users
☑ Logout All Users
☑ Setup Users... All Users
☑ Language Settings... All Users
☑ Print Page All Users
☑ Print Screen All Users
☑ Exit Application All Users


Menu Option Access Levels — General | Utilities | Communications

☑ Acknowledge All Alarms... All Users
☑ Acknowledge Latest Alarm... All Users
☑ Display Alarm Status... All Users
☑ Display Alarm History... All Users
☑ Display Event/Error Log... All Users
☑ Display Recipe Viewer... All Users
☑ Display Data Log Viewer... All Users
☑ Point Maintenance... All Users


Menu Option Access Levels — General | Utilities | Communications

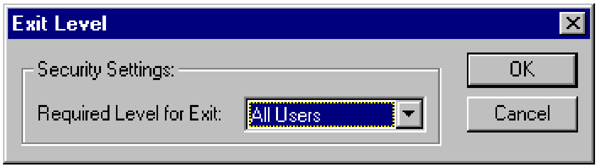
☑ PLC Maintenance... All Users
☑ PLC Data Monitor... All Users
☑ PLC Data Trace... All Users
☑ PLC I/O Table... All Users
☑ PLC Memory Card... All Users
☑ PLC Errors... All Users
☑ PLC Setup... All Users
☑ Network Configuration Tool... All Users
☑ Performance Monitor Tool... All Users

As well as a context-sensitive floating menu for the development environment, there is a floating menu for the runtime environment which is configurable in the development environment. It is possible to select the operations to be on the runtime floating menu, by clicking the relevant setting across the three tabs *General*, *Utilities* and *Communications*. It is also possible to assign a level of user privilege to the menu, so each level of user sees a different floating menu. This is achieved by selecting a user privilege from the relative drop down list. The *Communications* tab permits the activation of CX-Server components such as the PLC Data Trace and PLC Memory Card components. Refer to the *CX-Server Reference Manual* for further information on CX-Server components.

On completion, click the **OK** pushbutton to accept changes or the **Cancel** pushbutton to abort.

Exit Level

An additional security measure can be applied by selecting *Exit Level* from the *Runtime Security* menu. The Exit Level dialog is displayed as follows:



Exit Level

Security Settings:

Required Level for Exit: All Users

The dialog allows a specific privilege to be applied to exiting the runtime application. Select a user privilege from the drop down list and click the **OK** pushbutton to confirm the setting or the **Cancel** pushbutton to abort.

Compiling and Running a Project

When a project is running it cannot be edited. CX-Supervisor runs projects under the CX-Supervisor runtime environment.

To run the current project, click on the **Run** button on the toolbar.

The CX-Supervisor runtime environment starts, and automatically runs the project in a separate dialog which is given the name of the project. It allows examination of project alarm details and the run history.

Save Runtime As

Once the project has been compiled and run, the compiled version of the project may be saved into another directory location or onto another disc for issue to the target equipment. CX-Supervisor asks for a location and name for the runtime project. This is a standard Windows dialog and operation depends on your operating system. Please consult your Microsoft documentation for full details. The default file type is *.SR2.

Create Runtime Install Disk

A CX-Supervisor application can be packaged safely for authorised distribution by creating a runtime install disk. The process copies all required files, plus files to install to the specific directory to be used to distribute the runtime project. These files can be on a removable Memory Stick or later burnt to a distribution CD or even to floppy disk if the project is small enough.

Click on *Create Runtime Install Disc* from the *Project* menu. The Save Runtime As dialog is displayed ready to create an installation to the desired folder. Refer to chapter 6, Save Runtime As for further information regarding the Save Runtime As dialog.

Note: The files are not compressed. This feature does not support large projects that need to span multiple disks.

Note: The Runtime environment itself is not copied and must still be installed from a legitimate source

Note: Any ActiveX components used are not copied and should be installed on the target machine using the supplier’s instructions.

Note: Support files used by any ActiveX control cannot be detected and are not copied. These will need to be added manually. This includes:

- Omron OPC Client settings stored in .OPC file
- Omron Industrial Component communication settings, stored in .MSC file
- Any bitmaps when using Microsoft Image control
- Any file linked to by an OLE object (objects with embedded files should be OK)

Note: Existing databases or those specified by dynamic connections cannot be detected and are therefore not copied. These will need to be added manually.

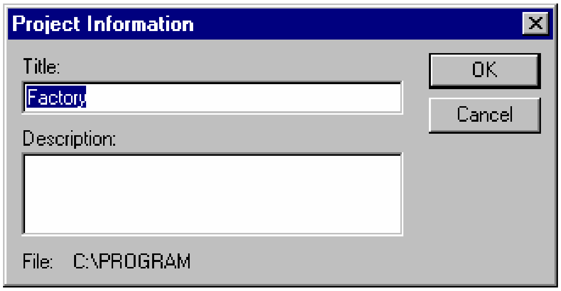
Note: Report Templates used by script cannot be detected and are therefore not copied. These will need to be added manually.

Note: Any User Settings files used by script (e.g. .CSV or .TXT files) cannot be detected and are therefore not copied. These will need to be added manually.

Project Information

Information may be stored concerning a project, by using the Project Information dialog. This dialog may be accessed, either when a project is first created, from the New Project dialog, or by selecting *Information* from the *Project* menu.

CX-Supervisor displays the following dialog:



Title: Factory

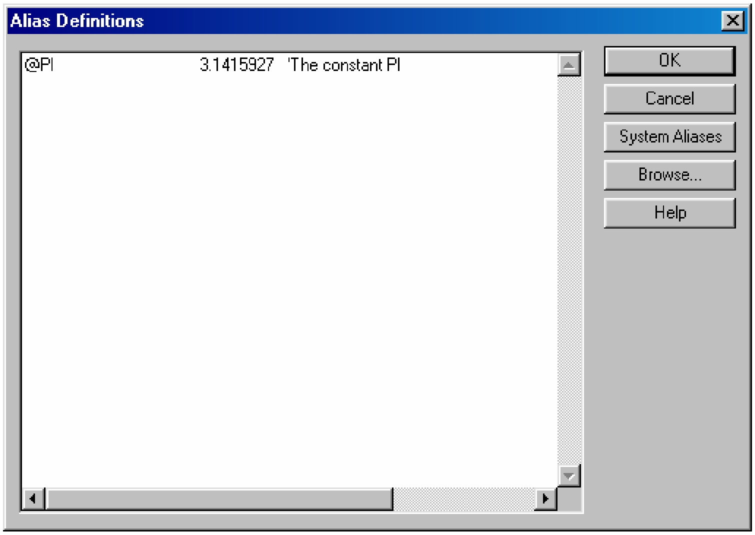
Description:

File: C:\PROGRAM

Enter a title and any relevant details concerning the project in the *Title:* and *Description:* fields, and click the **OK** pushbutton.

Alias Definitions

An alias definition can be provided to replace strings in scripts and expressions used throughout CX-Supervisor applications. An associated string replaces the alias when used in a script or expression. Select *Alias Definitions* from the *Project* menu.



@PI 3.1415927 'The constant PI

The 3 columns are seperated by tabs and are the Alias text to be used, the actual value to be used and an optional comment, (starting with ‘ character) respectively

Find

The project can be searched to find occurrences of text or a point name. It can also be used to search for text within script e.g. to find where a text message is generated from.

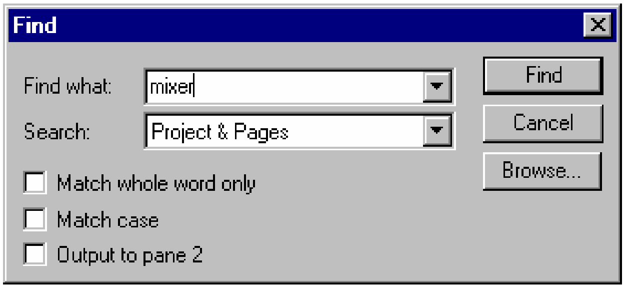
The areas which can be searched are:

- The current page.
- All pages.
- Project scripts.
- Alarms.
- Recipes.
- The project, which includes Project scripts, Alarms and Recipes.

By default, all areas are searched i.e. ‘Project and Pages’. To perform a Find operation:

1, 2, 3…

1. Select the Find button on the toolbar or *Find* from the *Edit* menu.
2. Type the text to find in the *Find What* field, or select a previous entry from the drop down list. The **Browse** pushbutton can be used to select an item.
3. Select the area to search; *Project & Pages* searches all areas.
4. Choose the required options; *Output to pane 2* results in pane 1 to be saved for future use.
5. Press the **Find** pushbutton to start the search or the **Cancel** pushbutton to abort the operation.



Find what: mixer

Search: Project & Pages

□ Match whole word only

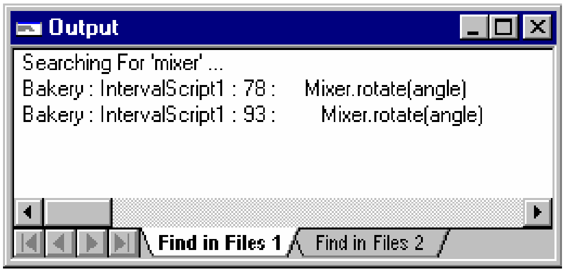
□ Match case

□ Output to pane 2

All occurrences of the text in the selected areas is shown in the Output window

Output Window

The output window shows output from the Build and Find functions.



```
Searching For 'mixer' ...
Bakery : IntervalScript1 : 78 :    Mixer.rotate(angle)
Bakery : IntervalScript1 : 93 :      Mixer.rotate(angle)
```

The window can be docked to any side of the screen or floated above other windows. To enforce floating, press <Ctrl> while moving the window.

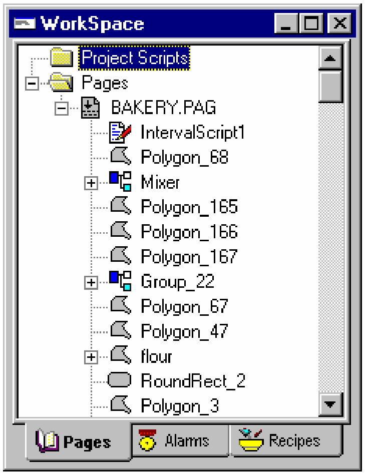
Build results are shown automatically during building and remain on screen if there are errors or warnings.

Find results show the location including page name, alarm name, recipe, script name, object, line number and animation as relevant, followed by the occurrence itself. Double clicking on any line opens the appropriate editor.

The output can be printed by selecting *Print* from the right mouse context menu.

Navigating Projects with the Workspace

The Workspace is activated by clicking the **Workspace** button on the toolbar. The Workspace dialog is displayed:



Selecting the *Pages*, *Alarms*, *Recipes*, *Data Logging* or *Databases* tab displays a list of the associated components that form part of the project.

The Workspace can be docked to any side of the screen or floated above other windows. To enforce floating, press <Ctrl> while moving the window.

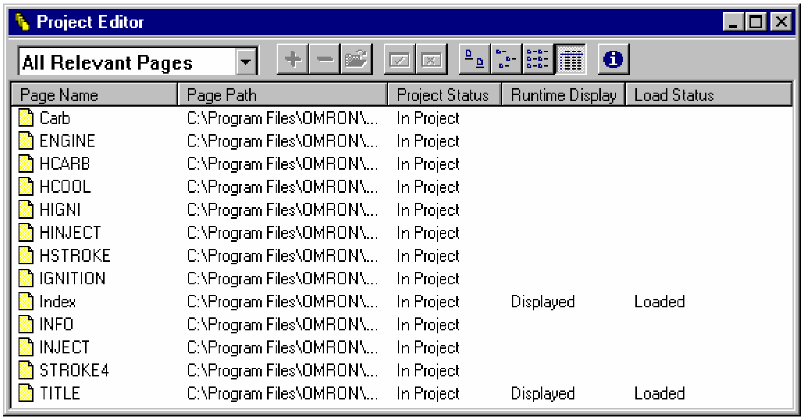
Project Editor

The Project Editor is activated by clicking the **Project Editor** button on the toolbar.

When activated, the Project Editor may be displayed minimised at the bottom of the main CX-Supervisor window. Double click to view the contents of the Project Editor.

About the Project Editor

The Project Editor provides a window into the contents of a project. It lists all the pages currently comprising a project, and allows for pages to be moved into or out of projects. An example of the Project Editor dialog is shown as follows:



All Relevant Pages

Page Name	Page Path	Project Status	Runtime Display	Load Status
Carb	C:\Program Files\OMRON\...	In Project		
ENGINE	C:\Program Files\OMRON\...	In Project		
HCARB	C:\Program Files\OMRON\...	In Project		
HCOOL	C:\Program Files\OMRON\...	In Project		
HIGNI	C:\Program Files\OMRON\...	In Project		
HINJECT	C:\Program Files\OMRON\...	In Project		
HSTROKE	C:\Program Files\OMRON\...	In Project		
IGNITION	C:\Program Files\OMRON\...	In Project		
Index	C:\Program Files\OMRON\...	In Project	Displayed	Loaded
INFO	C:\Program Files\OMRON\...	In Project		
INJECT	C:\Program Files\OMRON\...	In Project		
STROKE4	C:\Program Files\OMRON\...	In Project		
TITLE	C:\Program Files\OMRON\...	In Project	Displayed	Loaded

The Project Editor consists of a control bar and a display area. The control bar includes controls to filter the pages to be displayed, add and remove pages from a project, and open pages.

Viewing the Contents of a Project

The Project Editor is used to view the pages within a project.

Individual listed pages are sorted, based on the designated page name, page path, project status, runtime display or load status. By selecting the *Page Name* field, the pages are sorted alphanumerically by name. The *Page Type*, *Project Status*, *Runtime Display* and *Load Status* fields, once selected, react in the same way. The Project Editor display shows the projects listed in page name order.

The widths of the editor fields can be widened or narrowed as required using the mouse to drag the column boundaries.

The typeface of the editor can be amended from the *Preferences* option in the *File* menu. This is especially useful when printing.

Opening a Page via the Project Editor

To open pages via the Project Editor, click the **Open Page** button on the toolbar.

Adding Pages to a Project

When a new page is created, it is automatically registered in the Project Editor but its details are not saved. When the page is saved as a .PAG file, a message is displayed requiring confirmation to add this page to the project. Click the **Yes** pushbutton to add the page or the **No** pushbutton to save the page without adding to the project. If the page was not added to the project when saving, it can be added later by using the **Add Page** button on the toolbar.

Removing Pages From a Project

To remove pages from a project, select the page by clicking its description on the Project Editor and click the **Remove Page From Project** button on the toolbar. A message is displayed asking *'Do you want to remove (name and path of page) from the project?'*. Click the **Yes** pushbutton to delete the page, or click the **No** or **Cancel** pushbutton to keep it and return to the Project Editor.

Linking Pages in a Project

A number of pages may be linked together within a project. Linking allows a main page containing active elements (e.g. pushbuttons) to be loaded and depending upon the actions performed (i.e. which buttons are clicked) allow other pages to be loaded on demand. For further details of showing pages on demand, refer to chapter 9, Animation.

Selecting Pages for Display on Run

When a project is run the first pages to be displayed can be selected. To display a page when a project is run, open the Project Editor, select the page by clicking on its description and click the **Display Page Upon Run** button on the toolbar.

To stop a page being displayed on run, click the **Don't Display Page Upon Run** button on the Project Editor toolbar.

Changing the View Mode

Select the **Large Icons** button to view details with large icons.

Select the **Small Icons** button to view details with normal icons.

Select the **List** button to view details as a list.

Select the **Details** button to view details as a list including page name, page path, project status, runtime display and load status information. The details can be sorted in ascending order by clicking once or in descending order by clicking twice, in the appropriate field.

Viewing Project Details

To open the Project Details dialog and view the project name and description, click on the **Edit Details About Project** button on the Project Editor toolbar.

Multiple Selection

Standard multiple-selection facilities can be used within the Project Editor. To select an additional line hold the control key down while clicking; to select all lines between the anchor line (the last line clicked on) and another line hold the <Shift> key down while clicking. The buttons can then be used to apply to all the selected lines at the same time.

Printing from the Project Editor

Print Setup

The Project Editor can be printed in the same way pages can. Before printing, ensure that the printer has been set up correctly. To check the printer settings, refer to chapter 2, Pages.

Print Preview

To preview the page before printing, ensure that the Project Editor dialog is displayed and currently selected, and then select *Print Preview* from the *File* menu.

Refer to chapter 2, Pages regarding the use of the Print Preview display.

Printing

To print the contents of the Project Editor, select the **Print** button from the toolbar.

Refer to chapter 2, Pages regarding the use of the Print dialog.

Navigating Pages using Workbook mode

Open pages can be easily viewed with the page Workbook tabs shown at the bottom of the Pages area. There is a tab for each page currently open, and a single click shows the selected page. The Workbook mode can be turned off and on from the View menu

Using Full Screen mode

With many windows docked there may be times when all graphics objects or the screen layout of the whole application cannot be seen. Selecting Full Screen from the View menu shows the Pages area using the full area of the screen. The normal menu appears when the mouse is at the top of the screen, or click the “Close Full Screen” to revert to normal editing.

CHAPTER 7
Graphics Library

This chapter describes the CX-Supervisor Graphics Library. The chapter includes details of how the Graphics Library may be used for storing frequently used objects for use with a number of applications.

Overview

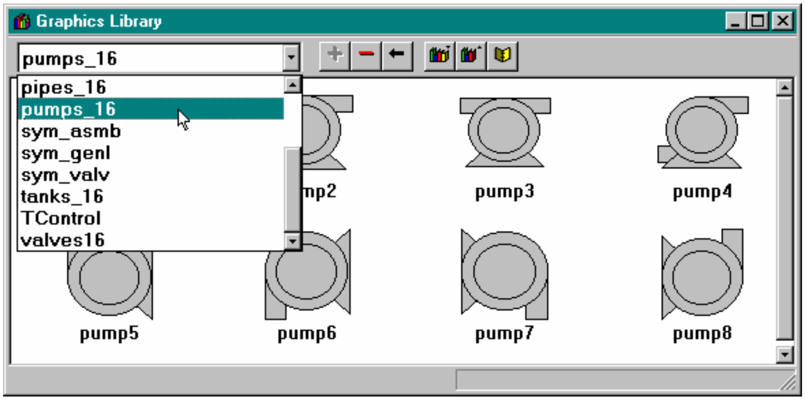
The Graphics Library is a repository for objects that are often used in CX-Supervisor pages. The Graphics Library Editor allows access to Libraries from a drop-down list box. Objects are stored in the Library and can be removed or copied, allowing quick addition of frequently used objects to pages or which appear in a project a number of times. Libraries are not project dependant, so objects can be copied from one project to another.

Graphics Library

Activating the Library

To activate the Library, click on the **Graphics Library** button on the toolbar. If the Library is already open but displayed as an icon, double click on the Library icon.

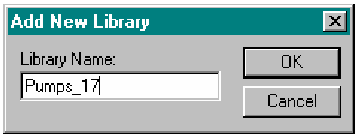
An example of the Graphics Library Editor is illustrated as follows, but note that actual library names and contents may differ from that shown in the following chapters.



Create Library

Each Library has a unique name which is entered when the Library is created.

To create a library, click on the **Add Library** button on the Graphics Library Editor toolbar. The Add New Library dialog is displayed as illustrated below:



Library Name: Pumps_17

Enter the name of the new Library file and click the **OK** pushbutton, or cancel the addition of the Library by clicking on the **Cancel** pushbutton.

If an object is dragged into the Library without a Library file open, then the Add New Library dialog is displayed. Refer to chapter 7, Manipulating Objects, for details on dragging objects into the Library.

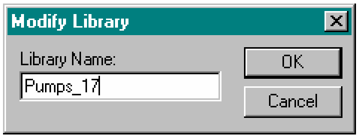
Opening a Library

The Graphics Library Editor consists of a number of Libraries which are selected from the drop down list box in the toolbar of the dialog.

Click on the Library name to display its contents in the dialog. The dialog shows the objects in the selected Library. The content of each library is provided for reference in chapter 7, Manipulating Objects.

Modify Library

The only element of the Library definition which is editable is the Library name. To change the name of a Library, open the desired Library file and click on the **Modify Library** button on the toolbar. The Modify Library dialog is displayed, an example of which is illustrated as follows:



Library Name: Pumps_17

Type over the current field entry with a new name and click the **OK** pushbutton, or cancel the operation by clicking the **Cancel** pushbutton.

Delete Library

A Library may be deleted by selecting it from the drop down list, to open it, and clicking the **Delete Library** button on the Graphics Library Editor toolbar. CX-Supervisor displays a message box to confirm the deletion of the Library.

Click the **Yes** pushbutton to remove the Library or the **No** pushbutton to cancel the operation and return to the Graphics Library. After clicking the **Yes** pushbutton, a second message box is displayed to confirm deletion of the library.

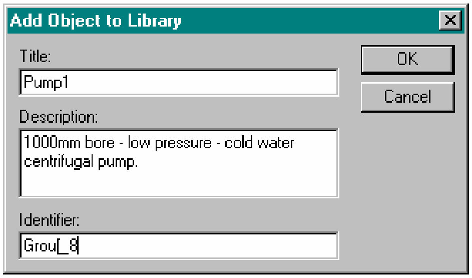
Click the **OK** pushbutton to delete the Library, or the **Cancel** pushbutton to the leave the Library unchanged.

Manipulating Objects

Add Object

Any object created on a CX-Supervisor page can be added to the Library. All attributes assigned to an object that is copied to the Library are stored, for example animation information.

To add an object, select it from a page and click the **Add Object to Library** button on the Graphics Library Editor toolbar. The Add Object To Library dialog is displayed as illustrated below:



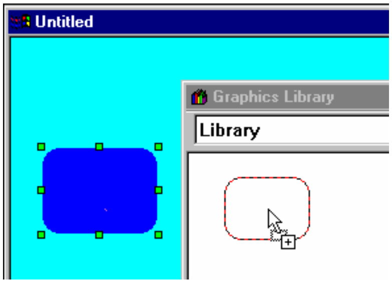
Title:
Pump1

Description:
1000mm bore - low pressure - cold water centrifugal pump.

Identifier:
Grou[_8

The name of the object used by the Object Identification control is shown in the *Title:* field. Type over this with a new name, if required (this is the name used by the Library). Enter a text description in the *Description:* field and an identifier for the object (this is used when it is inserted on a page) in the *Identifier:* field. Click the **OK** pushbutton to add the object to the Library and place the object in the dialog. Click the **Cancel** pushbutton to abort the operation. Further details on the Object Identification control are contained in chapter 1, Graphics Editor.

Alternatively, ensure that the Graphics Library Editor is open, click and hold the mouse button on the page object and drag it from the page onto the Graphics Library Editor, illustrated as follows:



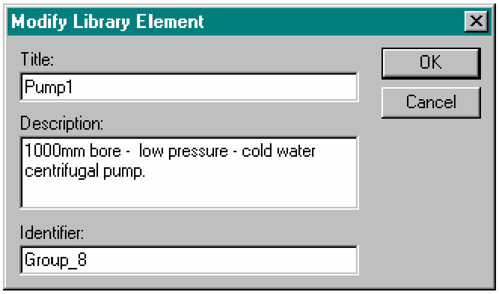
When the mouse button is released the object is placed in the Library, and the Add Object To Library dialog is displayed.

Copying an object to a page from a Library is the reverse of the click and drag procedure.

Objects can also be cut or copied and pasted into the Library, refer to chapter 4, Objects. When the object is pasted with the Graphics Library Editor open, the Add Object To Library dialog is shown.

Modify Library Element

To change the name of a Library object, click on the object in the Library (the object name is highlighted in the Graphics Library Editor), and click on the **Modify Library Element** toolbar button. The Modify Library Element dialog is displayed, as illustrated below:



Title: Pump1

Description: 1000mm bore - low pressure - cold water centrifugal pump.

Identifier: Group_8

Enter the new object title, text description and identifier. Click the **OK** pushbutton to add the new object description to the Library, or the **Cancel** pushbutton to cancel the operation.

Delete Object

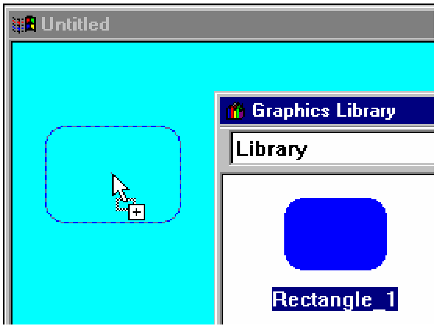
To delete an object from the Library, select the object, (the grab handles are not shown; however, the object name is highlighted in the Graphics Library Editor) and click on the **Delete Object** toolbar button.

A message box is displayed with the associated object name to remove. Click on the **Yes** pushbutton to delete the object or the **No** pushbutton to cancel the operation.

Using a Graphic Library Object in the Graphics Editor

A library object can be added to the Graphics Editor in the same way that the Graphics Editor is used to build libraries, by dragging from one dialog to another. Once a library object is added to the Graphics Editor, it can be further modified since each library object is simply a group of objects.

As an example, a page can be enhanced by using one of the gauges provided in the Gauges_1 default graphic library:



The gauge object can be stretched to suit the needs of the page, and other objects can be applied on top of, or in addition to the gauge. A library object can also be ungrouped by selecting *Ungroup* from the *Edit* menu.

The **Cut**, **Copy** and **Paste** buttons on the toolbar can be used as an alternative to dragging.

Default Graphic Library Objects

CX-Supervisor contains default, ready-to-use libraries. They cannot be added to (this is denoted by the 'READ ONLY' statement at the bottom of the Graphics Library Editor), although they can be applied to a page and tailored in the usual way.

Sharing Graphic Libraries

Graphic Libraries, including the default Graphic Libraries, can be shared by other users at other PCs running CX-Supervisor. Each library has its own *.MAT file (e.g. PANELS.MAT for the panels library). The *.MAT files reside in the same directory as the CX-Supervisor application. Using file utilities on the PC, it is possible to copy one or more *.MAT files onto a diskette or a network, and transfer them to a CX-Supervisor directory on another PC. Consult the *Microsoft Windows User Guide* for details on copying files.

Printing the Graphics Library

Print Setup

The Graphics Library can be printed in the same way pages can. Before printing, ensure that the printer has been set up correctly. To check the printer settings, refer to chapter 2, Pages.

Print Preview

To preview the page before printing, select *Print Preview* from the *File* menu.

Refer to chapter 2, Pages regarding the use of the Print Preview dialog.

Printing

To print the contents of the Graphics Library, select the **Print** button from the toolbar.

Refer to chapter 2, Pages regarding the use of the Print dialog.

CHAPTER 8
Alarms

This chapter describes CX-Supervisor alarms and the procedures associated with the creation, amendment and removal of alarms using the Alarm Editing facility. The use of alarms during the running of a project is also described.

What is an Alarm?

Alarms provide notification of a problem during the execution of an application in runtime. Alarms are defined in the development environment and monitored in the runtime environment. Alarms range from incidental to catastrophic in nature.

In runtime, the occurrence of an alarm condition, and any subsequent change of state, is recorded in a log file known as the Alarm History log.

An operator is alerted to an alarm condition by means of an Alarm Acknowledge dialog, which may also be accompanied by a warning sound. A list of current alarms is also maintained.

Individual alarms are defined in the development environment using the Alarm Editor. General alarm settings are controlled in the development environment using the Alarm Settings dialog.

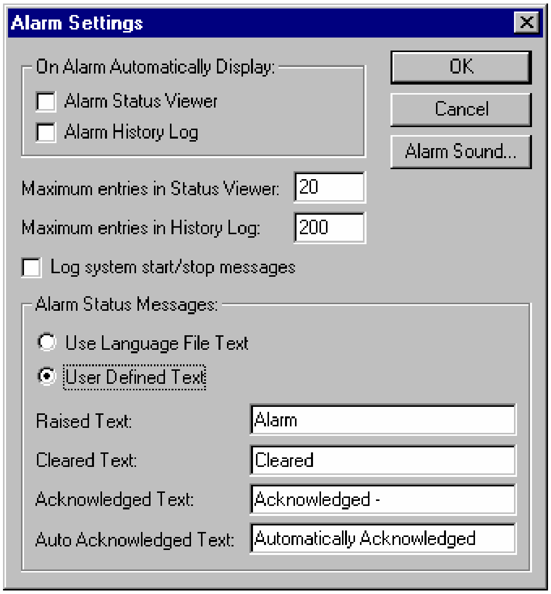
The Alarm Object (refer to chapter 4, Objects) is a graphical object which can be configured to display alarm messages for certain groups of alarms and thus provides a convenient way of filtering alarm messages.

Alarm definitions are made and modified using the Alarm Editor. To use the Alarm Editor, CX-Supervisor must currently have a project open. If no project is currently open, either select *Open Project* from the *Project* menu to open a previously saved project, or select *New Project* from the *Project* menu to create a new project.

Facilities exist to add an alarm, modify an existing alarm, copy an alarm, remove an existing alarm following confirmation, and display the alarm list in name, type or description order.

Alarm Settings

To open the Alarm Settings dialog, select the **Change General Alarm Settings** button from the Alarm Editor toolbar. The dialog is displayed as follows.



Alarm Settings

On Alarm Automatically Display:

□ Alarm Status Viewer

□ Alarm History Log

Maximum entries in Status Viewer: 20

Maximum entries in History Log: 200

□ Log system start/stop messages

Alarm Status Messages:

○ Use Language File Text

● User Defined Text

Raised Text: Alarm

Cleared Text: Cleared

Acknowledged Text: Acknowledged -

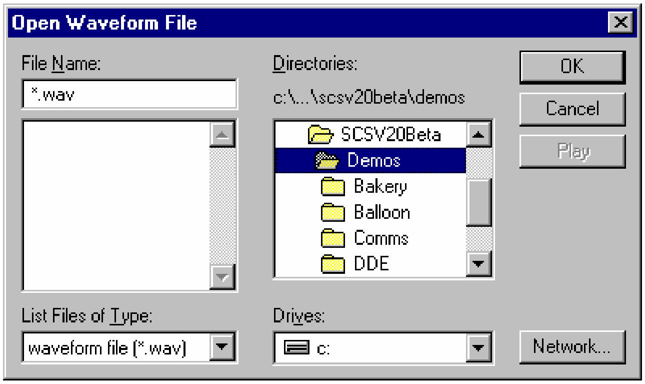
Auto Acknowledged Text: Automatically Acknowledged

In the *On Alarm Automatically Display* area, set the *Alarm Status Viewer* and *Alarm History Log* settings as required. When these options are set, the Current Alarms dialog or Alarm History dialog (respectively) displays automatically in runtime when an alarm occurs.

Set the *Maximum entries in Status* and *Maximum entries in History Log* fields to the desired values. The numbers specify how many messages are displayed in the respective viewer dialogs in runtime. Select the *Log system start/stop messages* setting to ON if required.

Note: If more messages exist than are requested to be displayed, the most recent messages are displayed in preference to older messages.

The **Alarm Sound** pushbutton allows the selection of an audible warning which are heard when an alarm is raised. The Open Waveform File dialog is shown below:



Open Waveform File

File Name: *.wav

Directories: c:\...\scsv20beta\demos

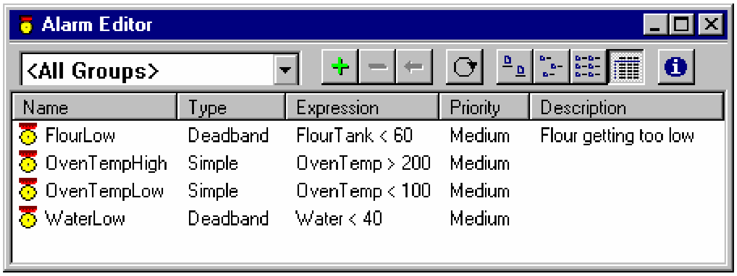
List Files of Type: waveform file (*.wav)

Drives: c:

Choosing a waveform file with this dialog does not in itself cause the warning sound to be heard when an alarm is raised. To hear the audible warning for any alarm, the *Play Sound* setting in the alarm definition must be ticked. Also, the PC on which CX-Supervisor is running must be equipped with a suitable sound card and audio capability.

Viewing the Contents of the Alarm Database

To open the Alarm Editor dialog, click the **Alarm Editor** button on the toolbar. An example of the Alarm Editor dialog is as follows:



Name	Type	Expression	Priority	Description
FlourLow	Deadband	FlourTank < 60	Medium	Flour getting too low
OvenTempHigh	Simple	OvenTemp > 200	Medium	
OvenTempLow	Simple	OvenTemp < 100	Medium	
WaterLow	Deadband	Water < 40	Medium	

Alarms may be segregated into groups. To select a group, pick an entry from the *Group* field.

All alarm definitions can be displayed by selecting <*All Groups*> from this list.

Individual listed alarms are sorted, based on a designated field type: name, type or description. By selecting the *Name* field, the alarms are sorted alphanumerically by name. The *Type* and *Description* fields, once selected, react in the same way. The Alarm Editor dialog shows the alarms in name order.

The widths of the alarm fields can be widened or narrowed as required using the mouse to drag the column boundaries.

The typeface of the editor can be amended by selecting *Preferences* from the *File* menu. This is especially useful when printing.

The **Change General Alarm Settings** button on the toolbar displays the Alarm Settings dialog, which allows global alarm settings to be modified. Refer to chapter 8, Alarm Header Information.

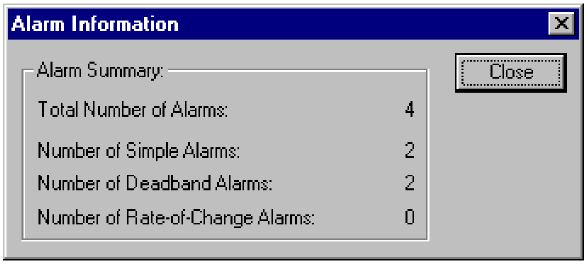
Select the **Large Icons** button to view details with large icons.

Select the **Small Icons** button to view details with normal icons.

Select the **List** button to view details as a list.

Select the **Details** button to view details as a list including name, type, expression, priority and description information. The details can be sorted in ascending order by clicking once or in descending order by clicking twice, in the appropriate field.

A summary of alarm information is available by selecting the **Display Information on Alarms** button from the toolbar. The resultant dialog shows an overall summary and a breakdown on the number of alarms per type. To exit the dialog, click the **Close** pushbutton. The Alarm Information dialog is shown as follows:

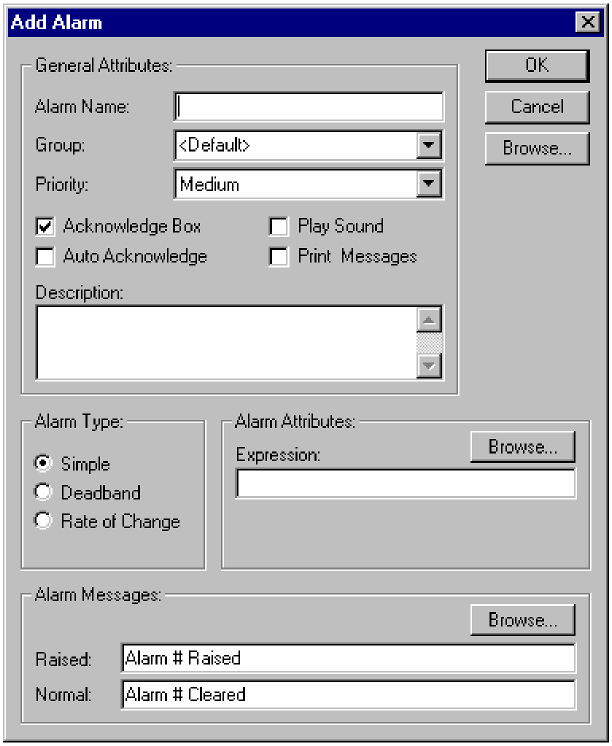


Alarm Summary:

Total Number of Alarms:	4
Number of Simple Alarms:	2
Number of Deadband Alarms:	2
Number of Rate-of-Change Alarms:	0

Creating a New Alarm

Open the Alarm Editor dialog, as described in chapter 8, Viewing the Contents of the Alarm Database. To add a new alarm, select the **Add Alarm** button from the toolbar. This results in the Add Alarm dialog being displayed.



General Attributes:

Alarm Name:

Group: <Default>

Priority: Medium

☑ Acknowledge Box □ Play Sound

□ Auto Acknowledge □ Print Messages

Description:

Alarm Type:

● Simple

○ Deadband

○ Rate of Change

Alarm Attributes:

Expression:

Alarm Messages:

Raised: Alarm # Raised

Normal: Alarm # Cleared

Once all the information has been provided for the new alarm, selecting the **OK** pushbutton commits the new alarm to the alarms database, whilst the **Cancel** pushbutton aborts this add operation.

Note: The short-cut keyboard combinations for Cut, Copy and Paste operations are valid within the Add Alarm dialog. Highlight part or all of a field and type <Ctrl>+X to cut the text or <Ctrl>+C to copy the text. Insert the cursor at the desired field and type <Ctrl>+V to paste the text. Since the cut and copy operations store the information in the Windows Clipboard, it may be pasted to another dialog or application.

Alarm Header Information

The name of the alarm is entered in the *Name:* field. The alarm name can be constructed of up to 20 alphanumeric characters. Any other characters, including spaces, generate an audible error, or an *Invalid Alarm Name entered* message. This field is visible from the Alarm Editor dialog.

The group to which this alarm is added is selected from the *Group:* field. To create a new group name, simply type the name of the new group in the *Group:* field. The use of alarm groups allows filtering of alarm messages by group name.

The priority assigned to the alarm is selected from the *Priority:* field.

The *Display Alarm Acknowledge Box* setting determines whether the Acknowledge Alarm dialog is displayed in runtime when the alarm occurs. The dialog notifies the operator of the alarm condition and allows the operator to acknowledge the alarm. The default value is *True*.

Activating the *Auto Acknowledge Alarm* setting causes these messages to be acknowledged without any operator action. Refer to chapter 8, Alarm Acknowledge.

The *Play Sound* setting determines whether an audible warning is sounded when the alarm occurs. The default value is *False*. Note that a waveform file must also have been selected for the warning sound to be heard. The waveform is selected by using the **Alarm Sound** pushbutton on the Alarm Settings dialog.

An alarm description, detailing the possibilities of how the current alarm would be raised, is inserted in the *Description:* field. To move to a new line, press <Ctrl>+<Return>. Any characters are valid, and up to 255 characters may be entered in the text field. This field is visible from the Alarm Editor dialog.

Activating the *Print Alarm Messages* setting causes alarms of this type to be printed automatically when they occur.

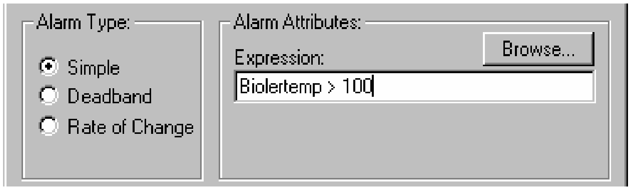
Alarm Type

The alarm type can be Simple, Deadband or Rate of Change. Select the desired type by clicking on the *Simple, Deadband* or *Rate of Change* settings. This field is visible from the Alarm Editor dialog.

Alarm Attributes

The attributes for an alarm are defined according to its type, as this defines when an alarm reacts. Since *Simple* is the default type, the Simple alarm attributes are initially visible.

For a Simple alarm, the following attributes are displayed:



Alarm Type:

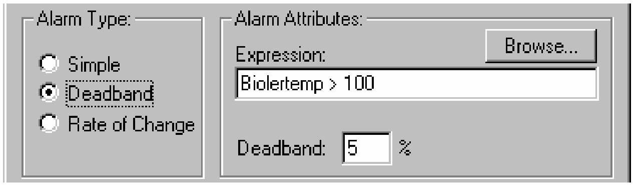
● Simple
○ Deadband
○ Rate of Change

Alarm Attributes:

Expression: Biolertemp > 100

An expression based on a point is entered in the *Expression:* field. The alarm is raised once the point meets the expression. The alarm is cleared when the point value subsequently fails to meet the expression.

For a Deadband alarm, the following attributes are displayed:



Alarm Type:

○ Simple
● Deadband
○ Rate of Change

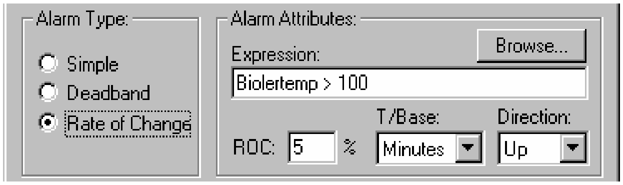
Alarm Attributes:

Expression: Biolertemp > 100

Deadband: 5 %

An expression based on a point is entered in the *Expression:* field. The alarm is raised once the point meets the expression. The alarm is cleared when the point value falls outside the expression value plus the specified deadband percentage. For the example above an alarm is raised when the boiler's temperature exceeds 100 and is cleared when the boiler temperature falls below 95 (100 minus 5%).

For a Rate of Change alarm, the following attributes are displayed:



Alarm Type:

○ Simple
○ Deadband
● Rate of Change

Alarm Attributes:

Expression: Biolertemp > 100

ROC: 5 %

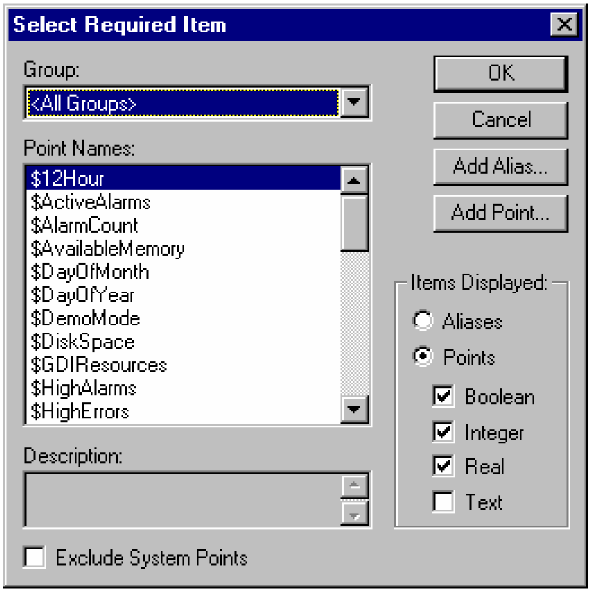
T/Base: Minutes

Direction: Up

An expression based on a point is entered in the *Expression:* field. The alarm is raised if the value exposed by the expression increases or decreases at the speed based on the values of the *ROC:* field, the *T/Base:* field, and the direction based on the *Direction:* field. The alarm is cleared when the rate-of-change is less than the critical rate.

For the above example an alarm is raised when the boiler's temperature increases by 5% or more per minute, and clears when the boiler's temperature increases by less than 5% per minute.

A point may be inserted into any of the alarm types' *Expression:* field by either typing in the point name or by selecting the **Browse** pushbutton, which results in the Select Required Item dialog being displayed, at the position where a point should be inserted.



Group: <All Groups>

Point Names:
$12Hour
$ActiveAlarms
$AlarmCount
$AvailableMemory
$DayOfMonth
$DayOfYear
$DemoMode
$DiskSpace
$GDIResources
$HighAlarms
$HighErrors

Description:

Items Displayed:
○ Aliases
● Points
☑ Boolean
☑ Integer
☑ Real
□ Text

□ Exclude System Points

Only viable points can be viewed from a Select Required Item dialog. The list of items in the *Point Names:* field can be refined by selecting an option from the *Group:* field. Click the **OK** pushbutton to accept the new point or click the **Cancel** pushbutton to leave the point unchanged. Clicking the **Add Point** pushbutton or **Add Alias** pushbutton allows a new point or alias to be created prior to association with the expression. Points are discussed in chapter 3, Points, whilst expression syntax is discussed in the *CX-Supervisor Script Language Reference Manual.*

Alarm Messages

A message pertaining to a raised alarm is entered in the *Raised:* field. The content of the field should be descriptive to provide the user with a reasonable basis for an alarm solution.

The Alarm Editor provides a default raised message. The name of the alarm is substituted for the # character in the message text when the **OK** pushbutton is clicked. The raised message may be changed at any time simply by entering the desired message in the text field.

A message associated with the alarm can also be entered in the *Normal:* field in the same manner to that above; this message is displayed once an alarm condition has been rectified and normal conditions have been resumed.

The Alarm Editor provides a default normal message. The name of the alarm is substituted for the # character in the message text when the **OK** pushbutton is clicked. The normal message may be changed at any time simply by entering the desired message in the text field.

Both alarm messages can include embedded point names via the **Browse** pushbutton, which is replaced at runtime with the value of the point:

Example

```
            Boiler temp high: ((BoilerTemp))
```

shows the point value in the alarm message.

Example

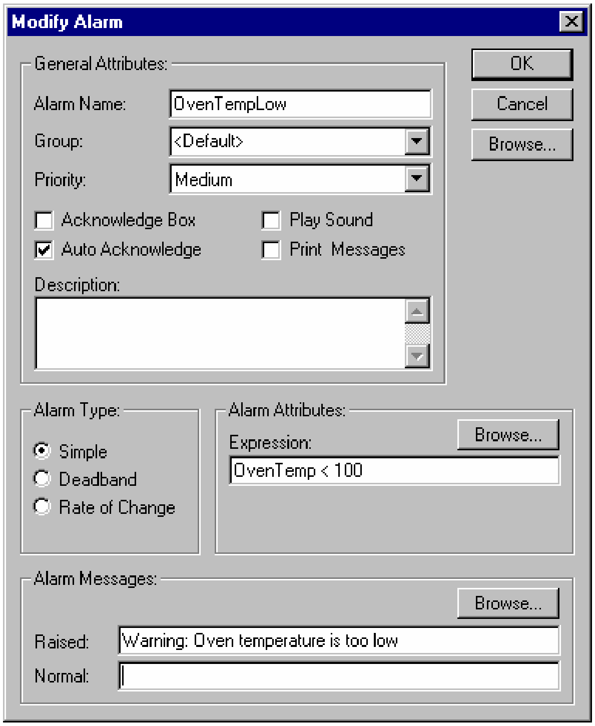
```
      Flow exceeded, Pressure = ((Press))
      ((“Temperature = %2.2f”,temp)).
```

Updating an Existing Alarm

Open the Alarm Editor dialog as described in chapter 8, Viewing the Contents of the Alarm Database.

⬅ To modify an existing alarm, highlight the alarm entry from the alarm list and select the **Modify Alarm** button from the toolbar.

This results in the Modify Alarm dialog being displayed (a dialog based on the Add Alarm dialog), as shown below:



Modify Alarm

General Attributes:

Alarm Name: OvenTempLow

Group: <Default>

Priority: Medium

☐ Acknowledge Box ☐ Play Sound

☑ Auto Acknowledge ☐ Print Messages

Description:

Alarm Type:

● Simple

○ Deadband

○ Rate of Change

Alarm Attributes:

Expression: OvenTemp < 100

Alarm Messages:

Raised: Warning: Oven temperature is too low

Normal:

The selected alarm can be redefined as described in chapter 8, Creating a New Alarm. Once all the information has been provided for the updated alarm, clicking the **OK** pushbutton commits the alarm to the alarms database, whilst the **Cancel** pushbutton aborts this modify operation.

Copying an Existing Alarm Definition

Open the Alarm Editor dialog as described in chapter 8, Viewing the Contents of the Alarm Database.

To copy an existing alarm, highlight the alarm from the alarm list and then type <Ctrl>+C. This creates a copy of the alarm definition on the Windows clipboard. To paste the alarm definition back in to the current project, type <Ctrl>+V. The Alarm Editor ensures the name of the alarm is unique by appending a number to the name. For example, if the original alarm name was “Alarm”, the name of the pasted alarm is “Alarm1”. All other properties of the pasted alarm remain the same as the original alarm definition.

It is possible to copy many alarms at once by highlighting all the desired alarms in the alarm list. Press and hold the <Ctrl> key and use the mouse to select alarm definitions one by one, or press and hold the <Shift> key and use the mouse to select blocks of alarm definitions. Once the desired alarms have been highlighted, the alarm definitions may be copied and pasted in the usual way.

Alarm definitions may be pasted into a project other than the original. Care should be exercised when attempting this procedure as the alarm expression may contain references to point names which may have different meaning in the new project.

Deleting an Existing Alarm

Open the Alarm Editor dialog as described in chapter 8, Viewing the Contents of the Alarm Database.

To remove an existing alarm, highlight the alarm from the alarm list and select the **Delete Alarm** button from the toolbar. A confirmation dialog is displayed. Click the **Yes** pushbutton to remove the alarm from the alarms database, or **No** pushbutton to abort the delete operation.

Printing Alarms

Print Setup

The Alarm Editor can be printed in the same way pages can. Before printing, ensure that the printer has been set up correctly. To check the printer settings, refer to chapter 2, Pages.

Print Preview

To preview the page before printing, select *Print Preview* from the *File* menu.

Refer to chapter 2, Pages regarding the use of the Print Preview display.

Printing

To print the contents of the Alarm Editor, click the **Print** pushbutton.

Refer to chapter 2, Pages regarding the use of the Print dialog.

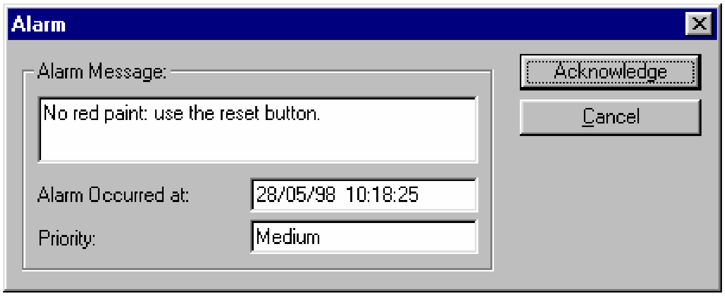
Alarm Reporting In Runtime

An alarm condition which occurs during runtime is brought to the attention of the operator. There are a variety of options available in the way alarms are reported. These options are configured in the development environment. During runtime, there are essentially four ways of examining alarm messages: the Alarm Acknowledge dialog, the Current Alarm dialog, the Alarm History dialog and the Alarm Object.

The Alarm Object is a graphical object which displays alarm messages in a similar way to the alarm status viewer. Refer to chapter 4, Objects for an explanation of how to configure an alarm object to filter alarm messages by group name.

Alarm Acknowledge

Whenever an alarm is raised during a runtime application, a confirmation dialog is optionally displayed requesting acknowledgement of the alarm. The dialog shows the alarm message, priority, and the date and time the alarm was raised. Click the ***Acknowledge*** pushbutton to close the dialog.



Alarm Message: No red paint: use the reset button.

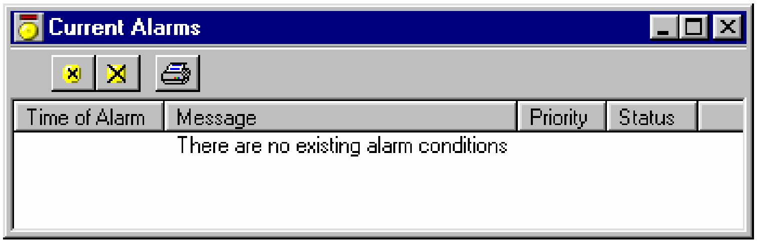
Alarm Occurred at: 28/05/98 10:18:25

Priority: Medium

Alarm messages are queued so that as each message is acknowledged, the next in the queue becomes visible. If a new alarm occurs which has higher or equal priority the details are updated to show this newer/more important alarm first. The dialog disappears when there are no further alarm messages to be reported. Each acknowledgement is logged in the alarm history log with the login name of the current user. An unacknowledged alarm does not affect a runtime application.

Current Alarms

A list of current alarms can be viewed by accessing the Current Alarms dialog. (A “current” alarm is one which has been raised but is not yet cleared and acknowledged.) The current alarm status viewer can be accessed in a variety of ways, and is dependant on the setup of the runtime application. Some applications may allow access to the dialog via the context-sensitive floating menu, whilst others may allow access via a pushbutton. Refer to chapter 6, Projects or chapter 9, Animation as appropriate. The Current Alarms dialog is as follows:


Current Alarms

Time of Alarm	Message	Priority	Status

There are no existing alarm conditions

The Current Alarm dialog always shows the alarm messages listed in order with the highest priority at the top.

The widths of the alarm fields can be widened or narrowed as required using the mouse to drag the column boundaries.

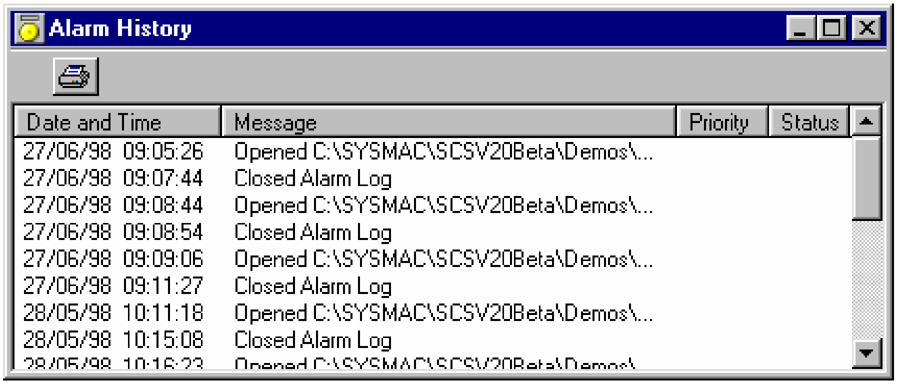
The **Acknowledge Selected Alarm** toolbar button acknowledges the alarm which is currently highlighted in the alarm status dialog. This has no effect if the alarm has already been acknowledged.

The **Acknowledge All Outstanding Alarms** toolbar button unconditionally acknowledges all outstanding alarms. Any unacknowledged alarms become acknowledged. This has no effect on alarms that are already acknowledged.

The **Print Contents of Alarm Status View** toolbar button causes the messages in the alarm status dialog to be printed. Before printing, ensure that the printer has been set up correctly.

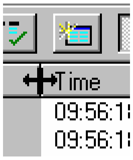
Alarm History

The occurrence of an alarm condition, and any subsequent change of state, is recorded in the alarm history log. Alarm messages recorded in the log can be displayed using the Alarm History dialog. The Alarm History dialog can be accessed in a variety of ways, and is dependant on the setup of the runtime application. Some applications may allow access to the dialog via the context-sensitive floating menu, whilst others may allow access via a pushbutton. Refer to chapter 6, Projects or chapter 9, Animation as appropriate. The Alarm History dialog is as follows:



Date and Time	Message	Priority	Status
27/06/98 09:05:26	Opened C:\SYSMAC\SCSV20Beta\Demos\...		
27/06/98 09:07:44	Closed Alarm Log		
27/06/98 09:08:44	Opened C:\SYSMAC\SCSV20Beta\Demos\...		
27/06/98 09:08:54	Closed Alarm Log		
27/06/98 09:09:06	Opened C:\SYSMAC\SCSV20Beta\Demos\...		
27/06/98 09:11:27	Closed Alarm Log		
28/05/98 10:11:18	Opened C:\SYSMAC\SCSV20Beta\Demos\...		
28/05/98 10:15:08	Closed Alarm Log		
28/05/98 10:16:23	Opened C:\SYSMAC\SCSV20Beta\Demos\		

This dialog always shows the log listed in date and time order.



The widths of the alarm fields can be widened or narrowed as required using the mouse to drag the column boundaries.

The **Print Contents of Alarm Log View** toolbar button causes the messages in the Alarm History dialog to be printed. Before printing, ensure that the printer has been set up correctly. The alarm log itself is simply a text file (*projectname*.UAL) that may be examined or printed using any of the usual utilities available under Windows, such as Microsoft Notepad. When the alarm log becomes full it is copied to the alarm backup (*projectname*.UAB) and a new alarm log file created. This switch over system uses a quick ‘append’ function for every alarm and prevents intensive write operations.

CHAPTER 9 Animation

This chapter describes CX-Supervisor animation and the procedures associated with the creation of a project animation, page animation and object animation, using the Animation Editing facility. The active use of points and how points dictate animations is also described.

Associating Points with Actions and Events

CX-Supervisor provides a facility to greatly enhance the versatility of a runtime application by the use of actions and animations to execute commands and logical expressions based on specified criteria being met. This can take many forms, for instance pressing a button, opening or closing a page or changes in the positioning of objects on a page.

CX-Supervisor achieves this enhanced functionality in two ways: by assigning an expression to a pre-defined action, or executing commands on a linear basis as a basic programming language. In addition, such functionality can be carried out on three levels: object, page, or project, with different objectives at each level.

Animations are actioned using the values or states of one or more points. For example, an integer point is assigned to an object, the colour of the object is black when the value of the point is “0”. The system causes the value of the point to be changed to the value “5”, which changes the colour of the object to red.

To demonstrate, an object’s colour can be changed in the runtime environment from black to white. This can be achieved using actions and animations in the two ways described, with the additional use of points. Firstly, using a pre-defined action, the object can be associated with the ‘Change Colour’ pre-defined action (runtime action). When a Boolean point’s value is “0” the colour is black, when the point’s value is “1” the colour is white. The same result can be achieved using the basic programming language (script). This time, a single command can change the colour from black to white. Finally, either instance can be initiated by the user in the runtime environment, e.g. the user presses a pushbutton to change the object’s colour.

This is a simple example, but underlines how quickly and smoothly a runtime application can be built. This chapter elaborates on all forms of actions and animations that can be applied to CX-Supervisor in the development environment. Also refer to the *CX-Supervisor Script Language Reference Manual* for further syntax details for the application of actions and animations.

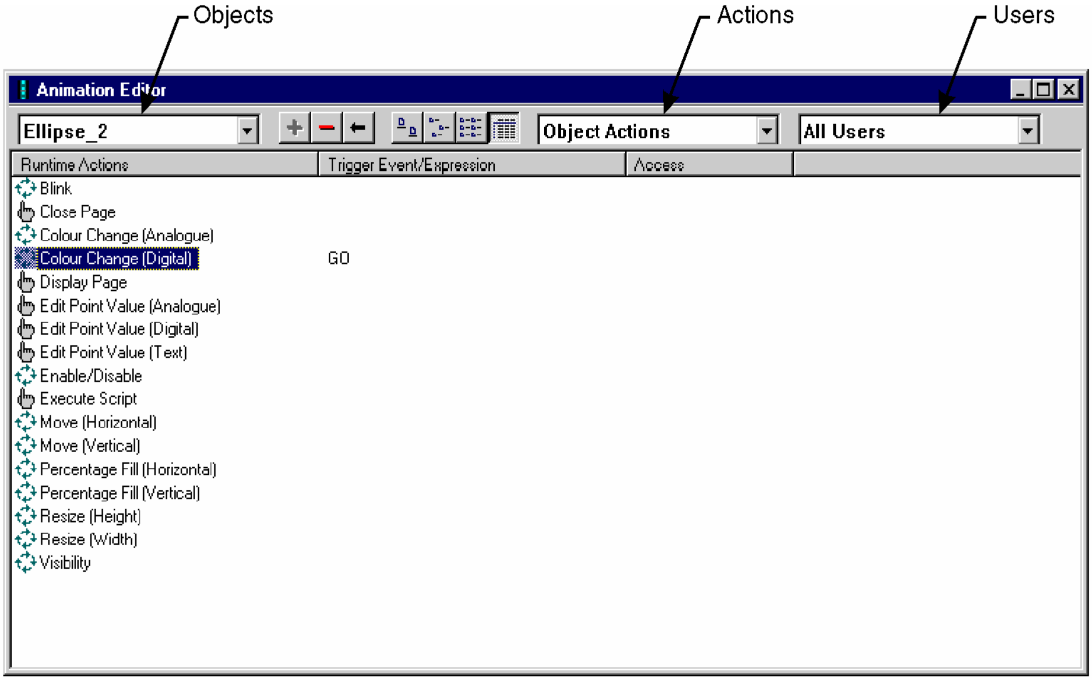
Animation Editor

Actions and animations are created on a project, page or object basis using the Animation Editor.

To use the Animation Editor, CX-Supervisor must currently have a project open. If no project is currently open, select *Open* from the *Project* menu to open a previously saved project, or select *New* from the *Project* menu to create a new project.

To access the Animation Editor, select the **Animation Editor** pushbutton from the toolbar.

The Animation Editor dialog is displayed:



Animation Editor

Ellipse_2 | Object Actions | All Users

Runtime Actions	Trigger Event/Expression	Access
Blink		
Close Page		
Colour Change (Analogue)		
Colour Change (Digital)	GO	
Display Page		
Edit Point Value (Analogue)		
Edit Point Value (Digital)		
Edit Point Value (Text)		
Enable/Disable		
Execute Script		
Move (Horizontal)		
Move (Vertical)		
Percentage Fill (Horizontal)		
Percentage Fill (Vertical)		
Resize (Height)		
Resize (Width)		
Visibility		

The Animation Editor dialog lists all possible animation actions in the *Runtime Actions* column. Attributable objects for the Runtime Actions are listed in the *Trigger Event/Expression* column.

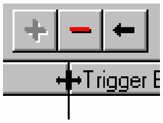
Animations can be instigated on a Project, Page or Object basis by selecting *Object Actions*, *Page Actions* or *Project Actions* from the *Actions* field. When opening the Animation Editor whilst an object within a page is selected, the *Actions* field defaults to *Object Actions*, and the selected object is shown in the Animation Editor. When accessing the Animation Editor with a page selected, this field defaults to *Page Actions*. When opening the Animation Editor whilst no selection is made, the *Actions* field defaults to *Project Actions*. The actions that may be applied to an object are dependant on the type of object selected. Only one type of action, *Execute Script,* can be applied to a Page or Project script.

The *Users* field allows selection of which users can use the selected page in the Runtime environment. This is used in conjunction with allocating users and allowing them permission to perform tasks; for further information on allocating users refer to chapter 6, Projects.

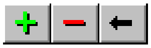
If *Object Actions* is the current selection for the *Actions* field, an object becomes active in the *Object* (left-most) field. An object within the current page can be selected from the *Object* field. The *Runtime Actions* list is dependant on the object selected from this box. The *Object* field is disabled for Page or Project actions.

Individual listed Runtime Actions are sorted, based on a designated field type, either Runtime Actions or Trigger Event/Expression. By clicking on the **Runtime Actions** pushbutton, the points are sorted alphanumerically by Runtime Actions. The **Trigger Event/Expression** pushbutton, once selected, reacts in the same way. The Animation Editor dialog shows the actions associated with ‘Polygon_1’ sorted in runtime action order.

The typeface of the editor can be amended by selecting *Preferences* from the *File* menu. This is especially useful when printing.



The widths of the animation fields can be widened or narrowed as required using the mouse to drag the column boundaries.



A new action can be added to the list of current actions by clicking the **Add Action** button in the toolbar. An existing action can be removed from the current actions by clicking the **Delete Action** button in the toolbar, whereupon a confirmation dialog is displayed. Click the **OK** pushbutton to delete the action, or click the **Cancel** pushbutton to abort the delete operation. An existing action can be modified from the current actions by clicking the **Modify Action** button in the toolbar.

Note: The short-cut keyboard combinations for Cut, Copy and Paste operations are valid within the Animation Editor dialogs. Highlight part or all of a field and type <Ctrl>+X to cut the text or <Ctrl>+C to copy the text; insert the cursor at the desired dialog field and type <Ctrl>+V to paste the text. Since the cut and copy operations store the information in the Windows Clipboard, it may be pasted to another dialog or application.

View Mode

The list of runtime actions can be viewed in a number of ways, providing simple or comprehensive details as follows:

Select the **Large Icons** button to view details with large icons.

Select the **Small Icons** button to view details with normal icons.

Select the **List** button to view details as a list.

Select the **Details** button to view details as a list including runtime actions, trigger/event expression and access. The details can be sorted in ascending order by clicking once or in descending order by clicking twice, in the appropriate field.

Project

Graphical objects have no relevance at Project level as they cannot be referenced. Scripts can be applied to a project to manipulate points. These scripts are associated with events that occur throughout the operating session. A script is made up of one or more simple statements that together make script code.

Refer to the *Script Language Manual* for explicit details regarding the construction and syntax of script code. Chapter 10, Runtime Actions describes the creation of script code using the Script Editor.

The script is defined using the *Execute Script* entry in the Animation Editor list.

Page

Page scripts are concerned with manipulating points and graphical objects that are used or included within that page. Actions cannot refer to graphical objects that appear in other pages. In other words, page scripts are used to drive a number of actions on the occurrence of a particular event. An example of page script code is as follows:

```
REM ** move the car and transport **


IF start THEN
    Position = position – speed
ENFIF


REM ** if the car is at the start then reset the **
REM ** position and reset the car colour **
IF position < 0THEN
    position = 800
    POLYGON_11.colour = dark_grey
ENDIF
REM setup the paint spray colour **
IF position < 300 || position > 400 && changepaint THEN
    IF paintblue THEN
      POLYGON_34.colour = dark_blue
  ENDIF
  IF paint green THEN
      POLYGON_34.colour = Dark_green
  ENDIF
    IF paintred THEN
      POLYGON_34.colour = red
  ENDIF
  IF paintpurple THEN
      POLYGON_34.colour = purple
  ENDIF
  IF paintyellow THEN
      POLYGON_34.colour = yellow
  ENDIF
  IF paintwhite THEN
      POLYGON_34.colour = white
  ENDIF
```

```
ENDIF
```

Refer to the *Script Language Manual* for explicit details regarding the construction and syntax of script code. 10.3 Runtime Actions describes the creation of script code using the Script Editor.

The script is defined using the *Execute Script* entry in the Animation Editor list.

Objects

One object or a selection of objects defined as a group are animated in a number of ways (refer to chapter 1, Graphics Editor for applicable objects):

- An object can be seen to be blinking.
- An object can be removed from the page.
- An object can change colour.
- A value associated with the object can be displayed.
- Text associated with the object can be displayed.
- An object can appear enabled for selection or disabled for selection.
- An object can move horizontally or vertically.
- An object can be flood-filled.
- An object can change its size, horizontally or vertically.
- An object can rotate.
- An object can have associated user interaction.
- An object can be displayed on the page.

However, it is not feasible for all the objects to be capable of all forms of animation. For instance, a toggle object can only be animated so that it is enabled or disabled, visible or invisible, whilst a text object can be resized, moved, change its colour, blink, rotate, await user interaction, be made visible or invisible, or display an associated value. A list of objects and their available animations is as follows:

	A	A	╲	□ ■	○ ●	▢ ▬	⌇	▰	◝	▭	⚲	▭▭▭	◷ ▮	Bit	OLE	📊	📈	⋰	⚙
Blink	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓						
Close page	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓				
Colour Change (Analogue)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓			✓	✓	✓	✓
Colour Change (Digital)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓			✓	✓	✓	✓
Display page	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓				
Display Value (Analog)	✓																		
Display Value (Digital)	✓																		
Display Value (Text)	✓																		
Edit point value (Analogue)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓				✓
Edit point value (Digital)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓				✓
Edit point value (Text)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓				✓
Enable/Disable	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓				
Execute script	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓				
Move (Horizontal)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Move (Vertical)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Percentage fill (Horizontal)				✓	✓	✓		✓	✓										
Resize (Height)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Resize (Width)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Rotate			✓	✓			✓	✓							✓				
Percentage fill (Vertical)				✓	✓	✓		✓	✓										
Visibility	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

A group object possesses the animation functions common to all objects within the group.

The ticks denote whether the animation function is available for the object. The blank cells denote the animation function is not available for the object. The icons within the table header denote the object created by that icon. Refer to chapter 1, Graphics Editor for further details.

In addition, an object can be animated in the form of a script. A script is made up of one or more statements. An example of object script code is as follows:

```
REM ** SET UP THE COLOUR FLAGS **
```

```
Paintblue = TRUE
paintgreen = FALSE
paintred = FALSE
paintpurple = FALSE
paintyellow = FALSE
paintwhite = FALSE

REM ** SET UP THE VALVE POSITION **

Bluevalve = TRUE
greenvalve = FALSE
redvalve = FALSE

Changepaint = TRUE
```

Refer to the *Script Language Manual* for full details regarding the construction and syntax of script code. The script is created by selecting the *Execute Script* entry in the Animation Editor list.

Chapter 10, Runtime Actions describes the application of object animations.

Printing the Animation Editor

Print Setup

The Animation Editor can be printed in the same way pages can. Before printing, ensure that the printer has been set up correctly. To check the printer settings, refer to chapter 2, Pages.

Print Preview

To preview the page before printing, select *Print Preview* from the *File* menu.

Refer to chapter 2, Pages regarding the use of the Print Preview display.

Printing

To print the contents of the Animation Editor, select the **Print** button from the toolbar.

Refer to chapter 2, Pages regarding the use of the Print dialog.

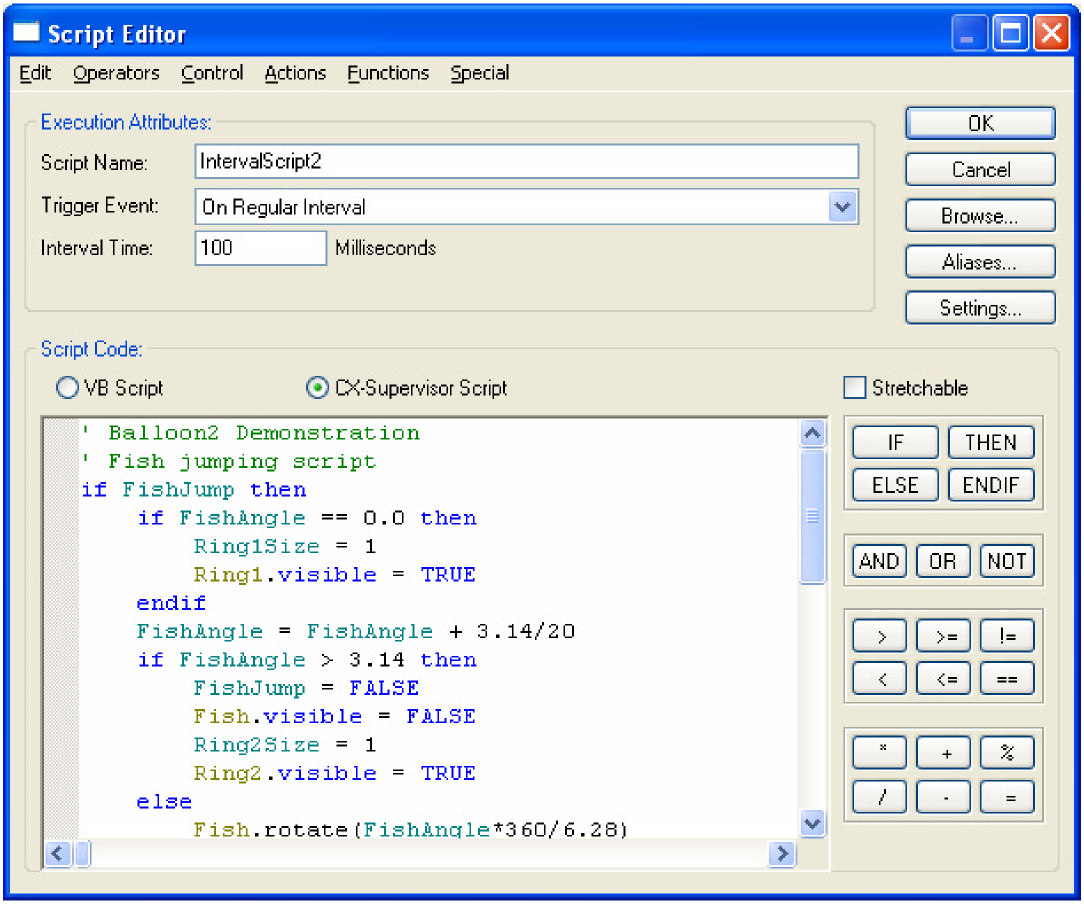
Runtime Actions

Script

A script, controlling the actions of an object, page or project can be created and updated using the Script Editor dialog.

The Script Editor allows script code to be created or amended which carries out the mechanics of the animation operation. To access the Script Editor dialog to add an action, select *Execute Script* from the animation list and click on the **Add Action** button on the toolbar. To access the Script Editor to modify an action, select *Execute Script* from the animation list and click on the **Modify Action** button on the toolbar. Refer to the Script Reference chapter 10, Objects, for a list of the objects to which an Execute Script action is applicable. Page Animations and Project Animations may have an *Execute Script* action.

On selection of the *Execute Script* action, the Script Editor dialog is displayed:



Script Name: IntervalScript2

Trigger Event: On Regular Interval

Interval Time: 100 Milliseconds

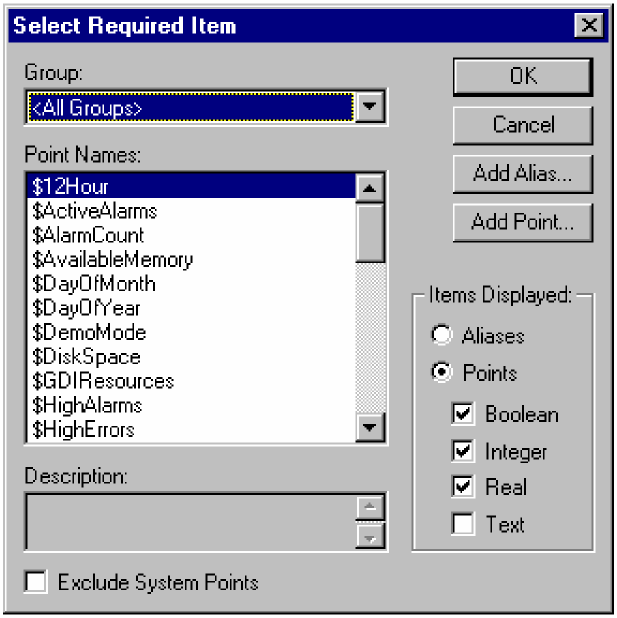
○ VB Script ● CX-Supervisor Script ☐ Stretchable

```
' Balloon2 Demonstration
' Fish jumping script
if FishJump then
    if FishAngle == 0.0 then
        Ring1Size = 1
        Ring1.visible = TRUE
    endif
    FishAngle = FishAngle + 3.14/20
    if FishAngle > 3.14 then
        FishJump = FALSE
        Fish.visible = FALSE
        Ring2Size = 1
        Ring2.visible = TRUE
    else
        Fish.rotate(FishAngle*360/6.28)
```

The script editor is colour coded to help show correct syntax with keywords and different object types are shown in different colours.

When creating the script choosing an action, function, etc. from the menu may require further information. This subsequent information is provided using some common dialogs: the Select Required Item dialog, the Simple Expression Entry dialog and the Object Selection dialog.

Use the Select Required Item dialog to 'pick' a point to associate with the current action. It is accessed by clicking the **Browse** pushbutton from the current dialog. This results in the Select Required Item dialog being displayed.



Select Required Item

Group: <All Groups>

Point Names:
$12Hour
$ActiveAlarms
$AlarmCount
$AvailableMemory
$DayOfMonth
$DayOfYear
$DemoMode
$DiskSpace
$GDIResources
$HighAlarms
$HighErrors

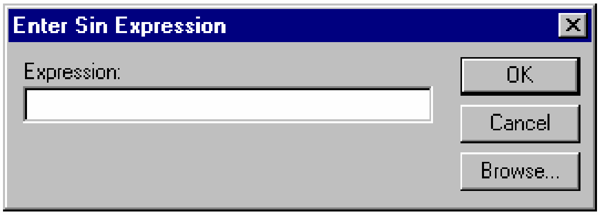
Description:

□ Exclude System Points

Items Displayed:
○ Aliases
● Points
☑ Boolean
☑ Integer
☑ Real
□ Text

Only relevant points are listed in the *Point Names:* field. Select the desired point from the *Point Names:* list, and click the **OK** pushbutton. Clicking the **Cancel** pushbutton aborts the selection. To add a new point click the **Add Point** pushbutton; for details on adding a point refer to chapter 3, Points.

Use the Simple Expression Entry dialog to assign an expression to an action. It is accessed from an entry in the script code. This results in the Simple Expression Entry dialog being displayed:

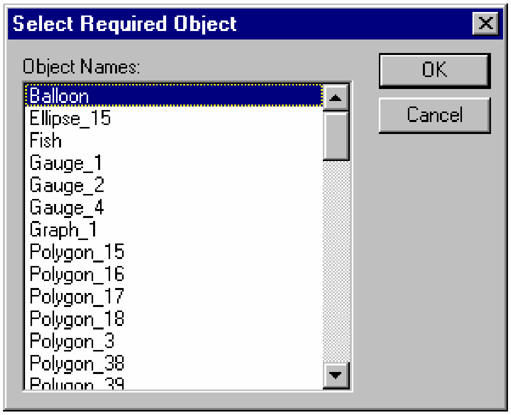


Enter Sin Expression

Expression:

Enter the expression in the *Expression:* field. The **Browse** pushbutton can be used to pick a valid point. Click the **OK** pushbutton to proceed, or the **Cancel** pushbutton to abort.

Use the Select Required Object dialog to assign an object to an action. It is accessed from an entry in the script code. This results in the Select Required Object dialog being displayed:



Select an object from the *Object Names:* field. Click the **OK** pushbutton to proceed, or the **Cancel** pushbutton to abort.

Note: The short-cut keyboard combinations for Cut, Copy and Paste operations are valid within the Script Editor. Highlight part or all of a field and type <Ctrl>+X to cut the text or <Ctrl>+C to copy the text; insert the cursor at the desired field and type <Ctrl>+V to paste the text. Since the cut and copy operations store the information in the Windows Clipboard, it may be pasted to another dialog or application.

Aliases

An alias definition can be provided to replace strings in scripts and expressions used throughout CX-Supervisor applications. An associated string replaces the alias when used in a script or expression. Select the **Aliases** pushbutton from the Script Editor dialog. The Alias Definitions dialog is displayed. Refer to chapter 6, Projects for more details on alias definitions.

Execution Attributes

The animation reacts to an event. A script can be associated with an event, depending on whether it is a project script, page script or object script.

For a project or page script, the following events trigger the execution of the script:

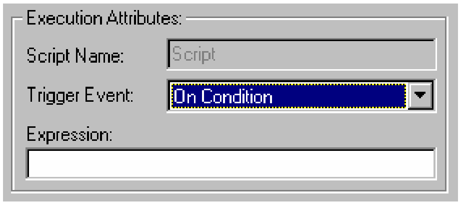
- On a condition, based on a regular expression.
- At a regular interval defined by an interval time.
- At the initialisation of the page or project.
- At the termination of the page or project.

For an object script, the following events trigger the script:

- Following the user clicking once on the left mouse button.
- On a condition, based on a regular expression.

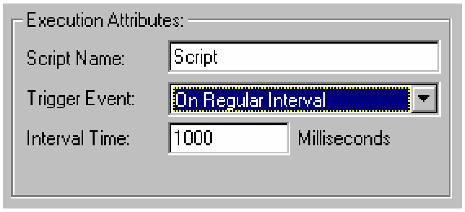
To specify a trigger for the script, select an option from the *Trigger Event:* field.

Note: If *On condition* is specified, a valid relational expression is required in the *Expression:* field. The script is executed every time a variable included in the expression changes value and the result of the expression is 'TRUE'. For example, the expression *$Second* executes every second except when *$Second* is zero; whereas *$Second >= 0* executes every second. If an invalid expression is entered, an error message is displayed when the **OK** pushbutton is clicked.



Execution Attributes:
Script Name: Script
Trigger Event: On Condition
Expression:

Note: If *On Regular Interval* is specified, a valid interval is required in the *Interval Time:* field in units of milliseconds. The script is started at the interval specified by this period. If an invalid time is entered, an error message is displayed when the **OK** pushbutton is clicked.



Execution Attributes:
Script Name: Script
Trigger Event: On Regular Interval
Interval Time: 1000 Milliseconds

Script Code

Script code is entered in the script code text field. To start a new line, press <Return>.

Refer to the *Script Language Manual* for explicit details regarding the construction and syntax of script code.

Expressions consist of operators and operands:

- Operators are relational, arithmetic, logical and include many functions.
- Operands are constants or point variables.

The script language can also contain Java and Visual Basic script. Refer to the *Script Language Manual* for details.

Scripts can be generated either by typing directly into the script code text field or by using the pull down menu commands. When using the pull down menu commands the user is assisted in the completion of the parameters to the command via a series of dialogs. The dialogs displayed vary according to the type of command being entered.

Points

A point may be inserted into the script code in a number of ways:

- Clicking the **Browse** pushbutton, picking the point off the list and clicking the **OK** pushbutton.
- Typing the name of the point

System points form part of the points list.

Undo Last Action

The last edit performed can be undone, if required, in a number of ways:

- Selecting *Undo* from the *Edit* menu.
- Pressing <Ctrl>+Z in the script code text field.

Cut to Clipboard

An area of script code can be moved to the Microsoft Windows Clipboard in a number of ways:

- Selecting the script code to cut, followed by *Cut* from the *Edit* menu.
- Selecting the script code to cut in the script code text field and pressing <Ctrl>+X.

Copy to Clipboard

An area of script code can be copied to the Clipboard, if required, in a number of ways:

- Selecting the script code to copy, followed by *Copy* from the *Edit* menu.
- Selecting the script code to copy from the script code text field and pressing <Ctrl>+C.

Paste from Clipboard

An area of script code can be copied from the Clipboard, if required, in a number of ways:

- Placing the I-beam cursor at the desired point in the script code field, followed by Paste from the Edit menu.
- Placing the I-beam cursor at the desired point in the script code field and then pressing <Ctrl>+V.

Insert Tab Code

A tab character can be inserted into the script code, if required, in a number of ways:

- Placing the I-beam cursor at the desired point in the script code field, followed by Tab from the Edit menu.

- Placing the I-beam cursor at the desired point in the script code field and then pressing <Ctrl>+<Tab>.

Finding and Replacing Text

Text can be found and replaced within the current script as follows:

- Select Find/Replace from the Edit menu and type appropriate text in the Find what and Replace with fields. Press the Find pushbutton to initiate the search and <Return> to execute text replacement.

Clear Script Code Field

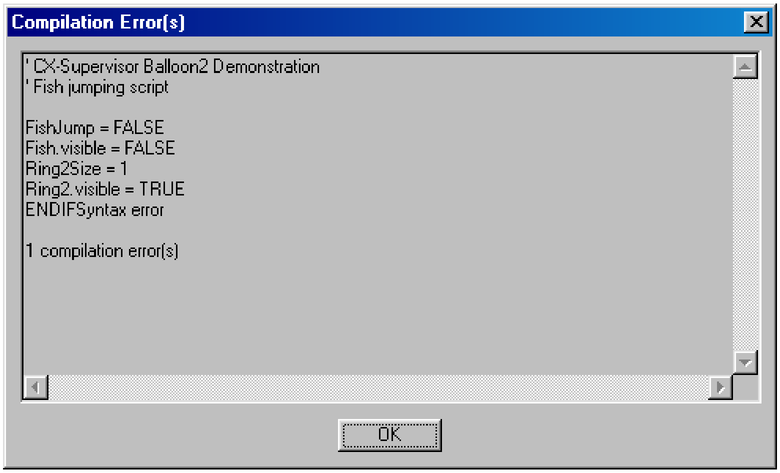
The script code field can be cleared, if required:

- Selecting Clear All from the Edit menu.

Script Completion

Once script entry or script modification is complete, click the **OK** pushbutton. To abort the Script Editor prior to completing the task, click the **Cancel** pushbutton.

If there is an error in the script, the Compilation Error(s) dialog is displayed.



```
' CX-Supervisor Balloon2 Demonstration
' Fish jumping script

FishJump = FALSE
Fish.visible = FALSE
Ring2Size = 1
Ring2.visible = TRUE
ENDIFSyntax error

1 compilation error(s)
```

In this example, the error is caused by a spurious 'ENDIF'.

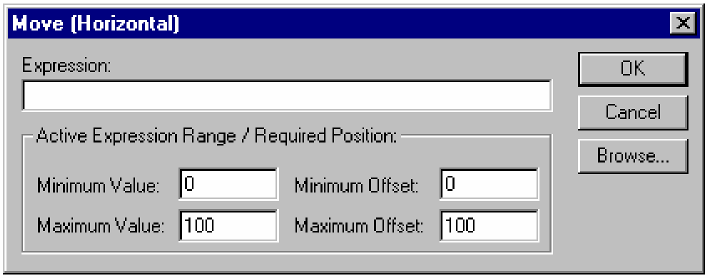
This dialog informs the user where an error has occurred and its type. If necessary, use the scroll bars to see the whole error message. Click the **OK** pushbutton to remove the Compilation Error(s) dialog and return to the Script Editor to fix the error.

Horizontal Move

Objects can be animated by moving either left or right. This is specified using the Move (Horizontal) dialog.

To access the Move (Horizontal) dialog to add an action, select *Move (Horizontal)* from the animation list and click the **Add Action** button on the toolbar. To access the Move (Horizontal) dialog to modify an action, select *Move (Horizontal)* from the animation list and click the **Modify Action** button on the toolbar. Refer to the Script Reference chapter 10, Objects, for a list of the objects to which this action is applicable.

On selection of the *Move (Horizontal)* action, the Move (Horizontal) dialog is displayed:



Move (Horizontal)

Expression:

Active Expression Range / Required Position:

Minimum Value: 0
Maximum Value: 100
Minimum Offset: 0
Maximum Offset: 100

To define the animation, an expression consisting of a number of operators and operands (which may be based on the status of a point) must be entered in the *Expression:* field. The boundaries in which the object moves are inserted into the *Active Expression Range/Required Position:* fields. The *Maximum Offset* specifies the number of pixels to the right (from its initial position) that the object moves when the result of the expression reaches the *Maximum Value*. The *Minimum Offset* specifies the number of pixels to the left (from its initial position) that the object moves, when the result of the expression reaches the *Minimum Value.* A value within the maximum and minimum values results in a proportionate movement between the left and right positions.

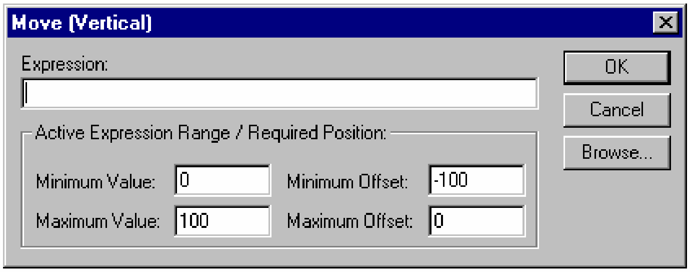
To abort the Horizontal Move definition, click the **Cancel** pushbutton. By clicking the **Browse** pushbutton, a point may be directly specified, as described in chapter 9, Runtime Actions. Once completed, click the **OK** pushbutton. If invalid data has been inserted into any field, a descriptive error message is displayed.

Vertical Move

Objects can be animated by moving either up or down. This can be specified via the Move (Vertical) dialog.

To access the Move (Vertical) dialog to add an action, select *Move (Vertical)* from the animation list and click the **Add Action** button on the toolbar. To access the Move (Vertical) dialog to modify an action, select *Move (Vertical)* from the animation list and click the **Modify Action** button on the toolbar. Refer to chapter 9, Objects, for a list of the objects to which this action is applicable.

On selection of the *Move (Vertical)* action, the Move (Vertical) dialog is displayed:



Expression:

Active Expression Range / Required Position:

Minimum Value: 0
Minimum Offset: -100
Maximum Value: 100
Maximum Offset: 0

To define the animation, an expression based on the status of a point must be entered in the *Expression:* field. The boundaries in which the object moves are inserted into the *Active Expression Range/Required Position:* fields. The *Maximum Offset* specifies the number of pixels upwards (from its initial position) that the object moves, when the result of the expression reaches the *Maximum Value.* The *Minimum Offset* specifies the number of pixels downwards (from its initial position) that the object moves, when the result of the expression reaches the *Minimum Value.* A value within the maximum and minimum values results in a proportionate movement between the top and bottom positions.

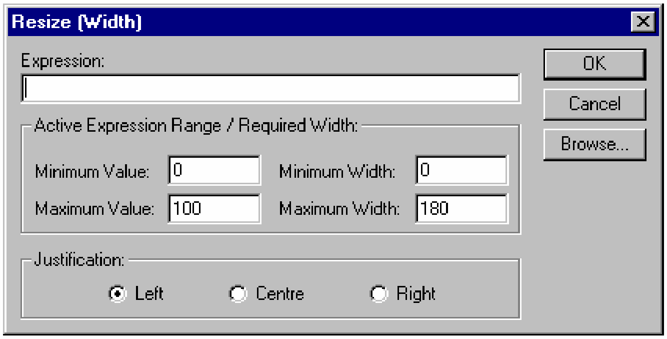
To abort the Vertical Move definition, click the **Cancel** pushbutton. By clicking the **Browse** pushbutton, a point may be directly specified, as described in chapter 9, Runtime Actions. Once completed, click the **OK** pushbutton. If invalid data has been inserted into any field, a descriptive error message is displayed.

Resize Width

Objects can be animated by shrinking or expanding horizontally. This can be specified via the Resize (Width) dialog.

To access the Resize (Width) dialog to add an action, select *Resize (Width)* from the animation list and click the **Add Action** button on the toolbar. To access the Resize (Width) dialog to modify an action, select *Resize (Width)* from the animation list and click the **Modify Action** button on the toolbar. Refer to chapter 9, Object, for a list of the objects to which this action is applicable.

On selection of the *Resize (Width)* action, the Resize (Width) dialog is displayed:



Expression:

Active Expression Range / Required Width:

Minimum Value: 0
Minimum Width: 0
Maximum Value: 100
Maximum Width: 180

Justification:
● Left ○ Centre ○ Right

To define the animation, an expression consisting of a number of operators and operands (which may be based on the status of a point) must be entered in the ***Expression:*** field.

The boundaries in which the object moves are inserted into the *Active Expression Range/Required Width:* fields. The *Maximum Width* specifies the required width in pixels of the object when the result of the expression reaches the *Maximum Value*. The *Minimum Width* specifies the required width in pixels of the object when the result of the expression reaches the *Minimum Value***.** A value within the maximum and minimum values results in a proportionate width between the minimum and maximum widths. The resizing also requires an anchor and direction in which to stretch or shrink. This can be specified as the left of the object, right of the object or centrally to the object, by selecting the appropriate *Justification:* setting.

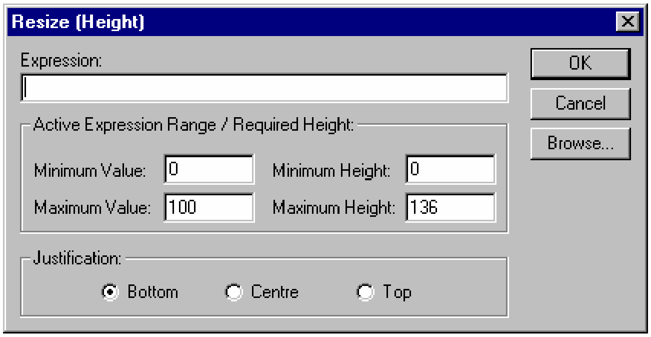
To abort the Resize Width definition, click the **Cancel** pushbutton. By clicking the **Browse** pushbutton, a point may be directly specified, as described in chapter 9, Runtime Actions. Once completed, click the **OK** pushbutton. If invalid data has been inserted into any field, a descriptive error message is displayed.

Resize Height

Objects can be animated by shrinking or expanding vertically. This can be specified via the Resize (Height) dialog.

To access the Resize (Height) dialog to add an action, select *Resize (Height)* from the animation list and click the **Add Action** button on the toolbar. To access the Resize (Height) dialog to modify an action, select *Resize (Height)* from the animation list and click the **Modify Action** button on the toolbar. Refer to chapter 9, Objects, for a list of the objects to which this action is applicable.

On selection of the *Resize (Height)* action, the Resize (Height) dialog is displayed:


Resize (Height)

Expression:

Active Expression Range / Required Height:

Minimum Value: 0
Minimum Height: 0
Maximum Value: 100
Maximum Height: 136

Justification:

● Bottom ○ Centre ○ Top

To define the animation, an expression consisting of a number of operators and operands (which may be based on the status of a point) must be entered in the *Expression:* field.

The boundaries in which the object moves are inserted into the *Active Expression Range/Required Height:* fields. The *Maximum Height* specifies the required height in pixels of the object when the result of the expression reaches the *Maximum Value*. The *Minimum Height* specifies the required height in pixels of the object when the result of the expression reaches the *Minimum Value*. A value within the maximum and minimum values results in a proportionate height between the minimum and maximum heights The resizing also requires an anchor and direction in which to stretch or shrink. This can be specified as the top of the object, bottom of the object or centrally to the object, by selecting the appropriate *Justification:* setting.

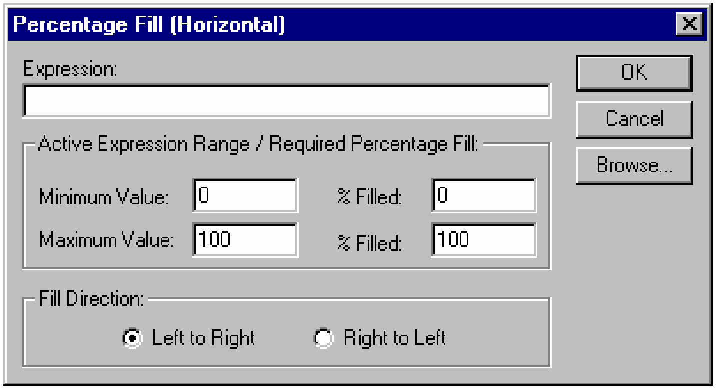
To abort the Resize Height definition, click the **Cancel** pushbutton. By clicking the **Browse** pushbutton, a point may be directly specified, as described in chapter 9, Runtime Actions. Once completed, click the **OK** pushbutton. If invalid data has been inserted into any field, a descriptive error message is displayed.

Horizontal Percentage Fill

Closed objects can be flood-filled along a horizontal axis. This can be specified via the Percentage Fill (Horizontal) dialog.

To access the Percentage Fill (Horizontal) dialog to add an action, select *Percentage Fill (Horizontal)* from the animation list and click the **Add Action** button on the toolbar. To access the Percentage Fill (Horizontal) dialog to modify an action, select *Percentage Fill (Horizontal)* from the animation list and click the **Modify Action** button on the toolbar. Refer to chapter 9, Objects, for a list of the objects to which this action is applicable.

On selection of the *Percentage Fill (Horizontal)* action, the Percentage Fill (Horizontal) dialog is displayed:



Percentage Fill (Horizontal)

Expression:

Active Expression Range / Required Percentage Fill:

Minimum Value: 0 % Filled: 0

Maximum Value: 100 % Filled: 100

Fill Direction:

● Left to Right ○ Right to Left

To define the animation, an expression consisting of a number of operators and operands (which may be based on the status of a point) must be entered in the *Expression:* field.

The boundaries associated with the percentage fill are inserted into the *Active Expression Range/Required Percentage Fill:* fields. The *% Filled* fields specify the percentage of the object to fill when the expression reaches its maximum and minimum values. The flood-fill also requires a direction. This can be specified as left to right or right to left, by selecting the appropriate *Fill Direction:* setting.

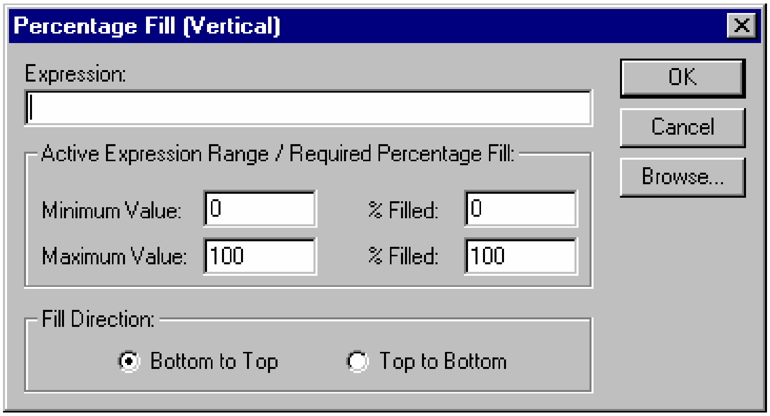
To abort the Horizontal Percentage Fill definition, click the **Cancel** pushbutton. By clicking the **Browse** pushbutton, a point may be directly specified, as described in chapter 9, Runtime Actions. Once completed, click the **OK** pushbutton. If invalid data has been inserted into any field, a descriptive error message is displayed.

Vertical Percentage Fill

Closed objects can be flood-filled along a vertical axis. This can be specified via the Percentage Fill (Vertical) dialog.

To access the Percentage Fill (Vertical) dialog to add an action, select *Percentage Fill (Vertical)* from the animation list and click the **Add Action** button on the toolbar. To access the Percentage Fill (Vertical) dialog to modify an action, select *Percentage Fill (Vertical)* from the animation list and click the **Modify Action** button on the toolbar. Refer to chapter 9, Objects, for a list of the objects to which this action is applicable.

On selection of the *Percentage Fill (Vertical)* action, the Percentage Fill (Vertical) dialog is displayed:



Percentage Fill (Vertical)

Expression:

Active Expression Range / Required Percentage Fill:

Minimum Value: 0 % Filled: 0

Maximum Value: 100 % Filled: 100

Fill Direction: ● Bottom to Top ○ Top to Bottom

To define the animation, an expression consisting of a number of operators and operands, which may be based on the status of a point, must be entered in the *Expression:* field.

The boundaries in which the object moves are inserted into the *Active Expression Range/Required Percentage Fill:* fields. The *% Filled* fields specify the percentage of the object to fill when the expression reaches its maximum and minimum values. The flood-fill also requires a direction. This can be specified as top to bottom or bottom to top, by selecting the appropriate *Fill Direction:* setting.

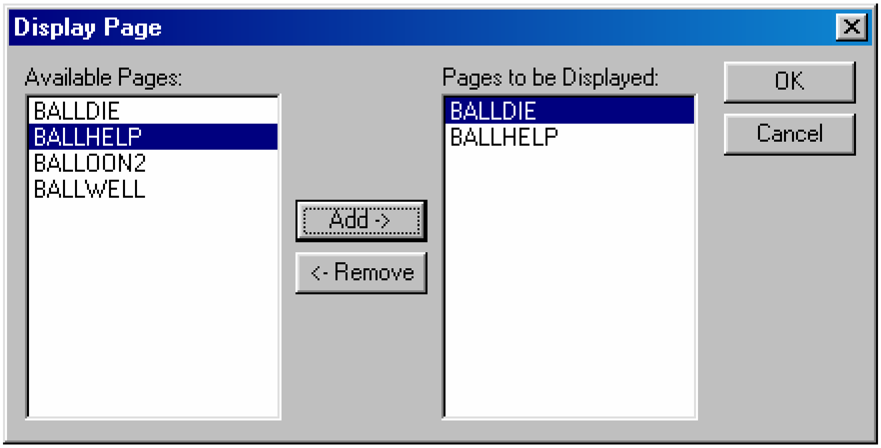
To abort the Vertical Percentage Fill definition, click the **Cancel** pushbutton. By clicking the **Browse** pushbutton, a point may be directly specified, as described in chapter 9, Runtime Actions. Once completed, click the **OK** pushbutton. If invalid data has been inserted into any field, a descriptive error message is displayed.

Display Page

CX-Supervisor allows the specification of pages within a project for display. This is set up using the Display Page dialog.

To access the Display Page dialog to add an action, select *Display Page* from the animation list and click the **Add Action** button on the toolbar. To access the Display Page dialog to modify an action, select *Display Page* from the animation list and click the **Modify Action** button on the toolbar. Refer to chapter 9, Objects, for a list of the objects to which this action is applicable.

On selection of the *Display Page* action, the Display Page dialog is displayed:



To specify a page for display, select a page from the *Available Pages:* list and click on the **Add** pushbutton. The page entry now appears in the *Pages to be Displayed:* list. More than one page can be selected for display. If a selected page is no longer required, select the entry from the *Pages to be Displayed:* list, and click on the **Remove** pushbutton.

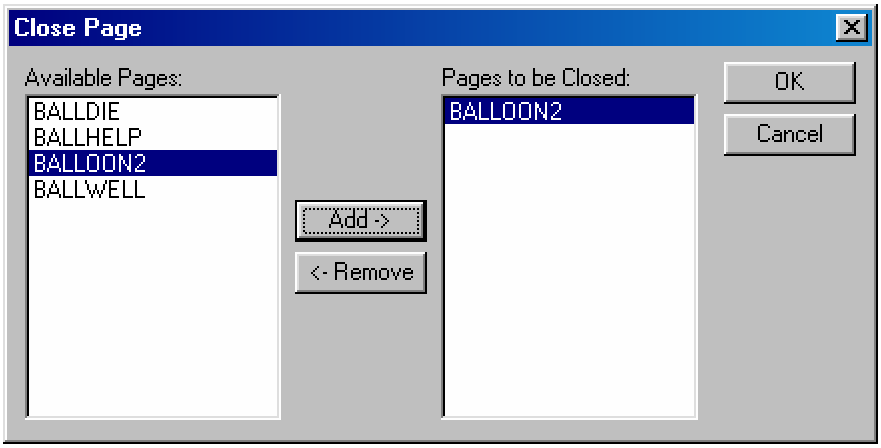
Once completed, click the **OK** pushbutton. To abort the Display Page edit, click the **Cancel** pushbutton.

Close Page

CX-Supervisor allows the specification of pages within a project to be removed from the display. This is set up using the Close Page dialog.

To access the Close Page dialog, select *Close Page* from the animation list and click the **Add Action** button on the toolbar. To access the Close Page dialog to modify an action, select *Close Page* from the animation list and click the **Modify Action** button on the toolbar. Refer to chapter 9, Objects, for a list of the objects to which this action is applicable.

On selection of the *Close Page* action, the Close Page dialog is displayed:



Available Pages:
BALLDIE
BALLHELP
BALLOON2
BALLWELL

Pages to be Closed:
BALLOON2

To specify a page for removal, select a page from the *Available Pages:* list and click on the **Add** pushbutton. The page now appears in the *Pages to be Closed:* list. Multiple pages can be selected for removal. If a selected page for removal is no longer required for that purpose, select the page from the *Pages to be Closed:* list, and click on the **Remove** pushbutton.

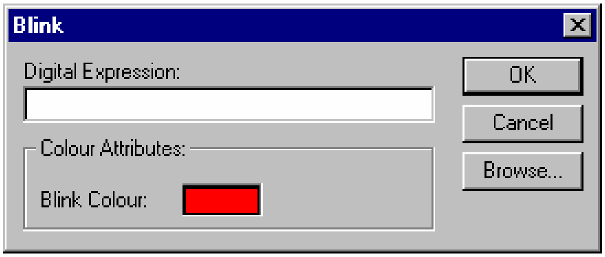
Once completed, click the **OK** pushbutton. To abort the Close Page edit, click the **Cancel** pushbutton.

Blink

Objects can be animated so that they blink. This is achieved using the Blink dialog.

To access the Blink dialog to add an action, select *Blink* from the animation list and click the **Add Action** button on the toolbar. To access Blink dialog to modify an action, select *Blink* from the animation list and click the **Modify Action** button on the toolbar. Refer to chapter 9, Objects, for a list of the objects to which this action is applicable.

On selection of the *Blink* action, the Blink dialog is displayed:



Enter a Boolean expression in the *Digital Expression:* field. Arithmetic, logical and relational expressions may also be entered as long as the result of the expression is zero or non-zero, i.e. the result equates to 'TRUE' or 'FALSE'. When the result of the expression is 'TRUE' the object(s) blinks.

To select a colour for the blink animation, click in the *Blink Colour:* field. The Colour Palette dialog is displayed so that a colour can be supplied to the object. The object blinks between its current colour and the chosen blink colour. The Colour Palette dialog is described in chapter 9, Common Colour Palette.

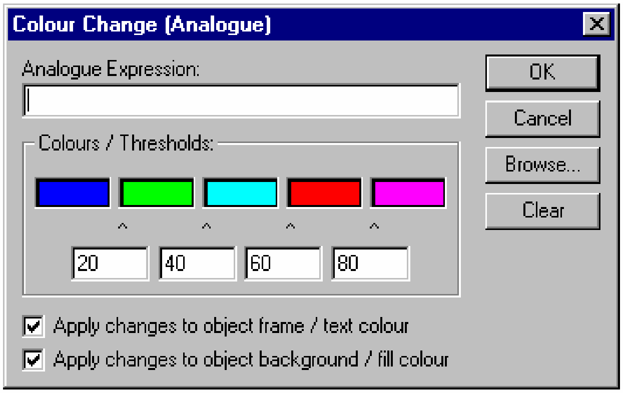
Once completed, click the **OK** pushbutton. To abort the Blink operation, click the **Cancel** pushbutton. By clicking the **Browse** pushbutton, a point may be directly specified, as described in chapter 9, Runtime Actions.

Colour Change (Analogue)

Objects can be animated so they change between a variety of colours at a variety of intervals. This is achieved using the Colour Change (Analogue) dialog.

To access the Colour Change (Analogue) dialog to add an action, select *Colour Change (Analogue)* from the animation list and click the **Add Action** button on the toolbar. To access the Colour Change (Analogue) dialog to modify an action, select *Colour Change (Analogue)* from the animation list and click the **Modify Action** button on the toolbar. Refer to chapter 9, Objects, for a list of the objects to which this action is applicable.

On selection of the *Colour Change (Analogue)* action, the Colour Change (Analogue) dialog is displayed:



Colour Change (Analogue)

Analogue Expression:

Colours / Thresholds:

20 40 60 80

☑ Apply changes to object frame / text colour

☑ Apply changes to object background / fill colour

Enter an expression in the *Analogue Expression:* field which results in a real or integer value. Only when the result of the expression reaches a value identified as a threshold does the object change colour.

The object can have one of five colours depending on the value of the expression. It changes between these colours at the thresholds. For example, choose the colour blue for the first colour, and the value 20 for the first threshold. The object then changes to blue until the value reaches 20, at which point it changes to the second colour. To apply the change to the frame of the object, select the *Apply colour changes to object frame* setting. To apply the change to the object background, select the *Apply colour changes to object background* setting.

On selection of a colour in the Colour Change (Analogue) dialog, the Colour Palette dialog is displayed. The Colour Palette dialog is described in chapter 9, Common Colour Palette.

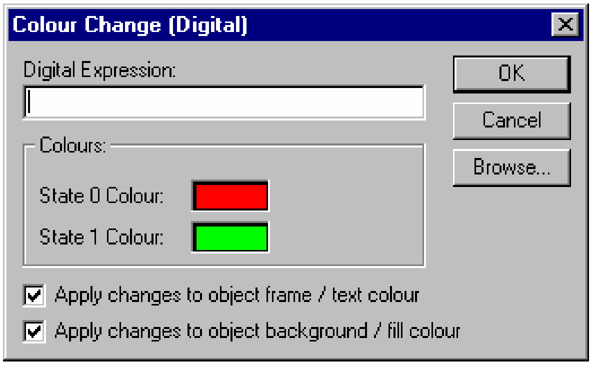
Click the **Clear** pushbutton to reset all colours and re-start. To abort the Colour Change (Analogue) edit, click the **Cancel** pushbutton. By clicking the **Browse** pushbutton, a point may be directly specified, as described in chapter 9, Runtime Actions. Once completed, click the **OK** pushbutton. If invalid data has been inserted into any field, a descriptive error message is displayed.

Colour Change (Digital)

Objects can be animated so they change between two colours. This is achieved using the Colour Change (Digital) dialog.

To access the Colour Change (Digital) dialog to add an action, select *Colour Change (Digital)* from the animation list and click the **Add Action** button on the toolbar. To access Colour Change (Digital) dialog to modify an action, select *Colour Change (Digital)* from the animation list and click the **Modify Action** button on the toolbar. For a list of the objects to which this action is applicable refer to chapter 9, Objects.

On selection of the *Colour Change (Digital)* action, the Colour Change (Digital) dialog is displayed:



To specify a change in an object's colour, enter a Boolean expression in the *Digital Expression:* field. Non-Boolean expressions may be entered as long as the result is 'TRUE' or 'FALSE'.

The colours are specified in the *Colour Attributes:* fields for Boolean State 0 and Boolean State 1. To apply the change to the frame of the object, select the *Apply colour changes to object frame* setting. To apply the change to the object background, select the *Apply colour changes to object background* setting.

On selection of a colour in the Colour Change (Digital) dialog, the Colour Palette dialog is displayed. The Colour Palette dialog is described in chapter 9, Common Colour Palette.

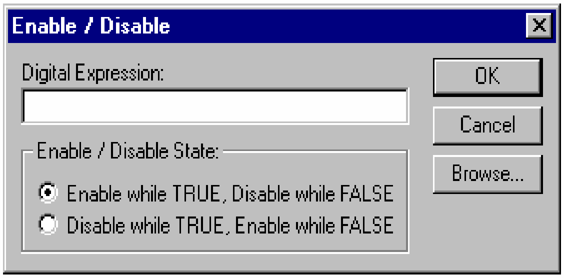
To abort the Colour Change (Digital) edit, click the **Cancel** pushbutton. By clicking the **Browse** pushbutton, a point may be directly specified, as described in chapter 9, Runtime Actions. _Once completed, click the **OK** pushbutton. If invalid data has been inserted into any field, a descriptive error message is displayed.

Enable/Disable

Objects can be enabled or disabled. This is achieved using the Enable/Disable dialog.

To access the Enable/Disable dialog to add an action, select *Enable/Disable* from the animation list and click the **Add Action** button on the toolbar. To access the Enable/Disable dialog to modify an action, select *Enable/Disable* from the animation list and click the **Modify Action** button on the toolbar. For a list of the objects to which this action is applicable refer to chapter 9, Objects.

On selection of the *Enable/Disable* action, the Enable/Disable dialog is displayed:



Enable / Disable

Digital Expression:

Enable / Disable State:

● Enable while TRUE, Disable while FALSE

○ Disable while TRUE, Enable while FALSE

To specify a change in the enabled/disabled state of an object, enter a Boolean expression in the *Digital Expression:* field. Non-Boolean expressions may be entered as long as the result is 'TRUE' or 'FALSE'. Whether the object is enabled or disabled when the expression is 'TRUE' is chosen with the **Enable/Disable State** settings.

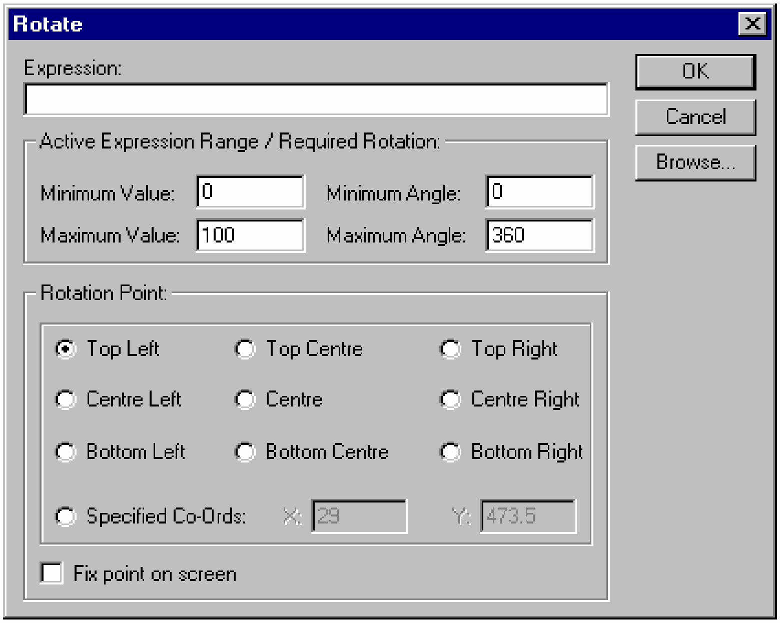
Once completed, click the **OK** pushbutton. To abort the **Enable/Disable** operation, click the **Cancel** pushbutton. If invalid data has been inserted into any field, a descriptive error message is displayed. By clicking the **Browse** pushbutton, a point may be directly specified, as described in chapter 9, Runtime Actions.

Rotate

An object can be rotated about its centre. Specify this using the Rotate dialog.

To access the Rotate dialog to add an action, select *Rotate* from the animation list and click the **Add Action** button on the toolbar. To access the Rotate dialog to modify an action, select *Rotate* from the animation list and click the **Modify Action** button on the toolbar. For a list of the objects to which this action is applicable refer to chapter 9, Objects.

On selection of the *Rotate* action, the Rotate dialog is displayed:



Rotate

Expression:

Active Expression Range / Required Rotation:

Minimum Value: 0 Minimum Angle: 0

Maximum Value: 100 Maximum Angle: 360

Rotation Point:

● Top Left ○ Top Centre ○ Top Right

○ Centre Left ○ Centre ○ Centre Right

○ Bottom Left ○ Bottom Centre ○ Bottom Right

○ Specified Co-Ords: X: 29 Y: 473.5

□ Fix point on screen

Enter an arithmetic expression in the *Expression:* field. The constraints of the rotation animation are specified in the *Active Expression Range/Required Rotation:* fields. Rotation is clockwise and the angle is specified in degrees..

Enabling the *Fix point on screen* option ensures that the rotation point remains at the same location even if the object moves.

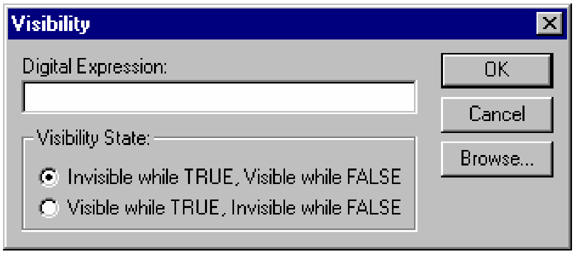
To abort the Rotate edit, click the **Cancel** pushbutton. By clicking the **Browse** pushbutton, a point may be directly specified, as described in chapter 9, Runtime Actions. completed, click the **OK** pushbutton. If invalid data has been inserted into any field, a descriptive error message is displayed.

Visibility

Objects can be rendered visible or invisible. Specify this using the Visibility dialog.

To access the Visibility dialog to add an action, select *Visibility* from the animation list and click the **Add Action** button on the toolbar. To access the Visibility dialog to modify an action, select *Visibility* from the animation list and click the **Modify Action** button on the toolbar. For a list of the objects to which this action is applicable refer to chapter 9, Objects.

On execution of the *Visibility* action, the Visibility dialog is displayed:



Digital Expression:

Visibility State:

● Invisible while TRUE, Visible while FALSE
○ Visible while TRUE, Invisible while FALSE

To specify a change in an objects' visibility, enter a Boolean expression in the *Digital Expression:* field. Non-Boolean expressions may be entered as long as the result is 'TRUE' or 'FALSE'. Whether the object is rendered visible or invisible when the expression is 'TRUE' is specified with the *Visibility State* settings.

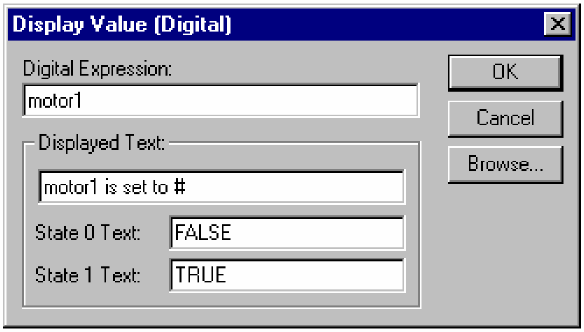
To abort the Visibility edit, click the **Cancel** pushbutton. By clicking the **Browse** pushbutton, a point may be directly specified, as described in chapter 9, Runtime Actions. Once completed, click the **OK** pushbutton. If invalid data has been inserted into any field, a descriptive error message is displayed.

Display Value (Digital)

The state of a Boolean point may be displayed using the Display Value (Digital) dialog.

To access the Display Value (Digital) dialog to add an action, select *Display Status Text* from the animation list and click the **Add Action** button on the toolbar. To access the Display Value (Digital) dialog to modify an action, select *Display Status Text* from the animation list and click the **Modify Action** button on the toolbar. For a list of the objects to which this action is applicable refer to chapter 9, Objects.

On selection of the *Display Value (Digital)* action, the Display Value (Digital) dialog is displayed:



Digital Expression: motor1

Displayed Text:

motor1 is set to #

State 0 Text: FALSE

State 1 Text: TRUE

An expression based on Boolean point must be entered. This expression is typed into the *Digital Expression:* field. A text string to coincide with the display of a Boolean value can be entered in the *Displayed Text:* field. The position of the displayed value within the text field is specified by entering a '*#'* character into the *Displayed Text:* field.

Note: To include a '#' character as part of the text, insert it as an escape character. For example, to display 'Box # 1' type 'Box \# #'. Only hashes to be displayed to the left of the # required for value display need escape characters, as the first hash without a meta character is taken to be the point at which the value should be displayed.

Text for Boolean State 0 and Boolean State 1 are entered in the *State 0 Text:* field and *State 1 Text:* field.

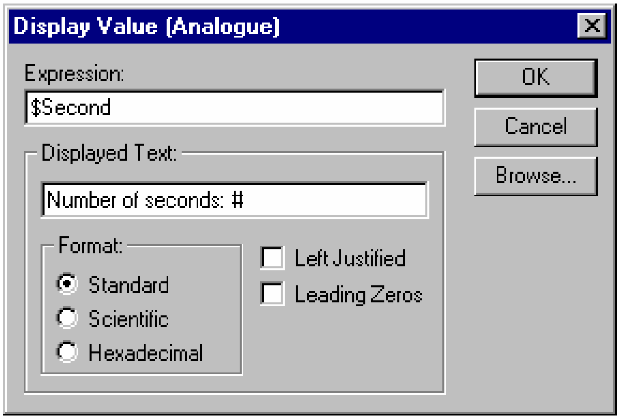
To abort the Display Status Text edit, click the **Cancel** pushbutton. By clicking the **Browse** pushbutton, a point may be directly specified, as described in chapter 9, Runtime Actions. Once completed, click the **OK** pushbutton. If invalid data has been inserted into any field, a descriptive error message is displayed.

Display Value (Analogue)

The state of a Real or Integer point may be displayed using the Display Value (Analogue) dialog.

To access the Display Value (Analogue) dialog to add an action, select *Display Value* from the animation list and click the **Add Action** button on the toolbar. To access the Display Value (Analogue) dialog to modify an action, select *Display Value* from the animation list and click the **Modify Action** button on the toolbar. For a list of the objects to which this action is applicable refer to chapter 9, Objects.

On selection of the *Display Value* action, the Display Value (Analogue) dialog is displayed:



Display Value (Analogue)

Expression: $Second

Displayed Text: Number of seconds: #

Format: ● Standard ○ Scientific ○ Hexadecimal

□ Left Justified
□ Leading Zeros

An expression based on Real or Integer point must be entered. This expression is typed into the *Expression:* field. A text string to coincide with the display of a Real or Integer value can be entered in the *Displayed Text* field (this field can be updated using the Graphics Editor). The position of the displayed value within the text field is specified by entering a *"#"* character into the *Displayed Text:* field. The value can be displayed in its decimal form, in scientific notation, or in hexadecimal, by selecting a *Format:* settings.

The number of decimal places displayed can be specified by adding a “#” character for each required place after a decimal point, e.g. #.## indicates 2 decimal places.

The text can also be left justified by setting the *Left Justified* setting.

Note: To include a ‘#’ character as part of the text, insert it as an escape character. For example, to display ‘Box # 1’ type ‘Box \# #’. Only hashes typed to the left of the # required for value display need escape characters, since the first hash without an escape character is taken to be the point at which the value should be displayed.

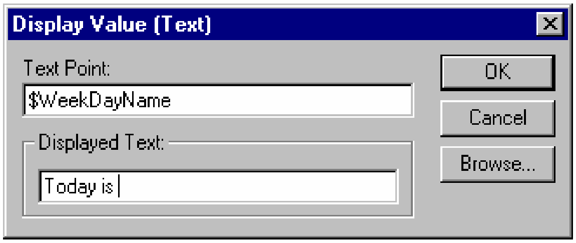
To abort the Display Value edit, click the **Cancel** pushbutton. By clicking the **Browse** pushbutton, a point may be directly specified, as described in chapter 9, Runtime Actions. Once completed, click the **OK** pushbutton. If invalid data has been inserted into any field, a descriptive error message is displayed.

Display Value (Text)

Text may be displayed using the Display Value (Text) dialog.

To access the Display Value (Text) dialog to add an action, select *Display Text Point* from the animation list and click the **Add Action** button on the toolbar. To access the Display Value (Text) dialog to modify an action, select *Display Text Point* from the animation list and click the **Modify Action** button on the toolbar. For a list of the objects to which this action is applicable refer to chapter 9, Objects.

On selection of the *Display Text Point* action, the Display Value (Text) dialog is displayed:



Text Point: $WeekDayName

Displayed Text: Today is

An expression based on a text point must be entered. This expression is typed into the *Text Point:* field. A text string to be displayed can be entered in the *Displayed Text:* field (this field can be updated using the Graphics Editor). The position of the displayed text within the text field is specified by entering a *“#”* character into the *Displayed Text:* field.

Note: To include a ‘#’ character as part of the text, insert it as an escape character. For example, to display ‘Box # 1’ type ‘Box \# #’. Only hashes typed to the left of the # required for value display need escape characters, since the first hash without an escape character is taken to be the point at which the value should be displayed.

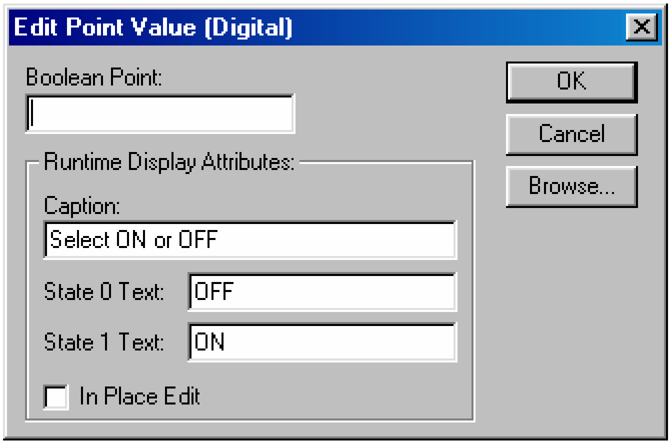
To abort the Display Text Value edit, click the **Cancel** pushbutton. By clicking the **Browse** pushbutton, a point may be directly specified, as described in chapter 9, Runtime Actions. Once completed, click the **OK** pushbutton. If invalid data has been inserted into any field, a descriptive error message is displayed.

Edit Point Value (Digital)

The value of a Boolean point may be issued to the user for amendment during runtime, defined using the User Input (Digital) dialog.

To access the **User Input (Digital)** dialog to add an action, select *Edit Point Value (Digital)* from the animation list and click the **Add Action** button on the toolbar. To access the User Input (Digital) dialog to modify an action, select *Edit Point Value (Digital)* from the animation list and click the **Modify Action** button on the toolbar. For a list of the objects to which this action is applicable refer to chapter 9, Objects.

On selection of the *Edit Point Value (Digital)* action, the dialog is displayed:



Edit Point Value (Digital)

Boolean Point:

Runtime Display Attributes:

Caption: Select ON or OFF

State 0 Text: OFF

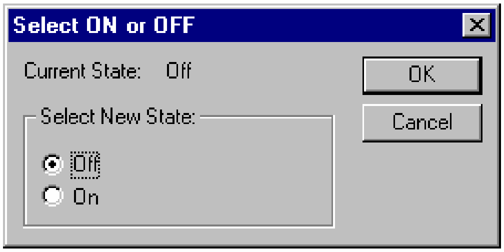
State 1 Text: ON

☐ In Place Edit

A Boolean point must be entered. This is typed into the *Boolean Point:* field. A text string to use as the caption of a user dialog can be entered in the *Caption:* field. The range of input for the value, and the representations of Boolean State 0 and Boolean State 1 are specified in the *State 0 Text:* field and *State 1 Text:* field. If *In Place Edit* is not checked a popup dialog performs the edit. If it is checked, the option is edited on the page with the options in a dropdown listbox.

To abort the Edit Point Value (Digital) edit, click the **Cancel** pushbutton. By clicking the **Browse** pushbutton, a point may be directly specified, as described in chapter 9, Runtime Actions. Once completed, click the **OK** pushbutton. If invalid data has been inserted into any field, a descriptive error message is displayed.

During runtime, selecting the user input object by clicking the left mouse button results in the digital value being edit either with the runtime User Input (Digital) dialog being displayed or the in place listbox, based on the contents of the development version.



Select ON or OFF

Current State: Off

Select New State:

● Off

○ On

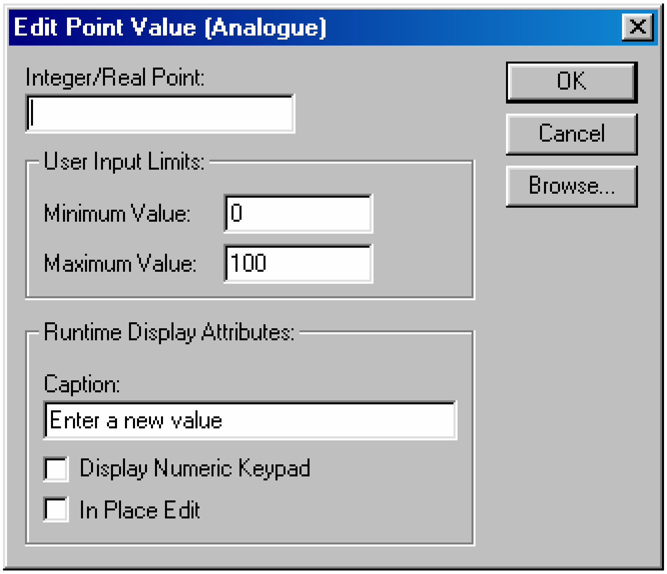
The user is provided with an instruction, based on the *Runtime Display Attributes:* field (in this example 'Select New State'), and *On* and *Off settings*, based on the *State 0 Text:* and *State 1 Text:* fields. The user clicks the **Cancel** pushbutton to abort the operation.

Edit Point Value (Analogue)

The value of a Real or Integer point may be issued to the user for amendment during runtime, defined using the User Input (Analogue) dialog.

To access the User Input (Analogue) dialog to add an action, select *Edit Point Value (Analogue)* from the animation list and click the **Add Action** button on the toolbar. To access the User Input (Analogue) dialog to modify an action, select *Edit Point Value (Analogue)* from the animation list and click the **Modify Action** button on the toolbar. For a list of the objects to which this action is applicable refer to chapter 9, Objects.

On selection of the *Edit Point Value (Analogue)* action, the dialog is displayed:



Edit Point Value (Analogue)

Integer/Real Point:

User Input Limits:

Minimum Value: 0

Maximum Value: 100

Runtime Display Attributes:

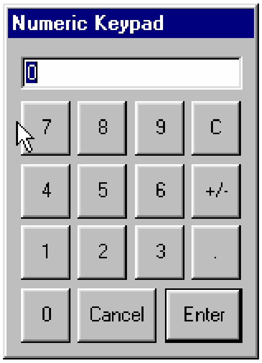
Caption: Enter a new value

□ Display Numeric Keypad

□ In Place Edit

A Real or Integer point must be entered. This is typed into the *Integer/Real Point:* field. A text string for the caption of a user dialog can be entered in the *Runtime Display Attributes:* field. The limits imposed on the user for the value of the input are specified in the *User Input Limits:* fields.

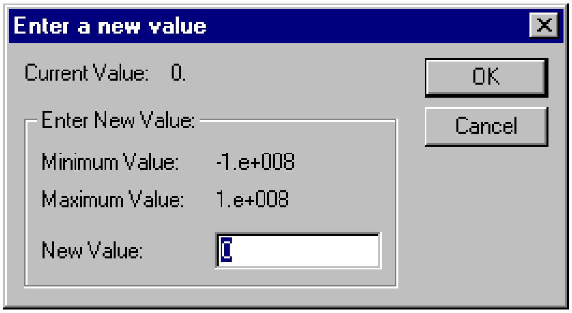
The *Display Numeric Keypad* option is useful if the runtime system does not have a keyboard, i.e. it is controlled by a touch screen or tracker ball. When a value is edited, a Numeric Keypad dialog is displayed which allows editing by clicking on screen.



If the *In Place Edit* option is checked, instead of the popup editing dialog, the value can be edited on the page in a standard Windows edit box.

To abort the Edit Point Value (Analogue) edit, click the **Cancel** pushbutton. By clicking the **Browse** pushbutton, a point may be directly specified, as described in chapter 9, Runtime Actions. Once completed, click the **OK** pushbutton. If invalid data has been inserted into any field, a descriptive error message is displayed.

During runtime, selecting the user input object by clicking the left mouse button results in the in place edit box or runtime User Input (Analogue) dialog being displayed, based on the contents of the development version.



Current Value: 0.

Enter New Value:

Minimum Value: -1.e+008

Maximum Value: 1.e+008

New Value: 0

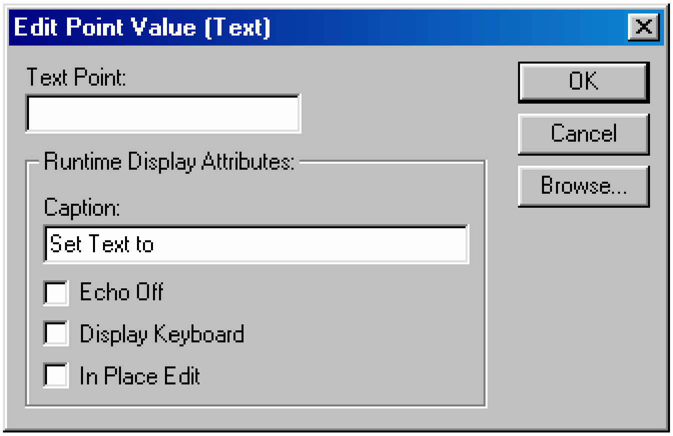
CX-Supervisor provides an instruction based on the *Runtime Display Attributes:* field, the minimum and maximum values allowed, based on the *User Input Limits:* fields, and a *New Value:* field, in which the user types the desired Real or Integer value. Click the **OK** pushbutton to accept the value, or the **Cancel** pushbutton to abort the operation. An invalid entry in the *New Value* field is exposed as an error once the **OK** pushbutton has been clicked.

Edit Point Value (Text)

A text point may be issued to the user for amendment during runtime, defined using the User Input (Text) dialog.

To access the User Input (Text) dialog to add an action, select *Edit Point Value (Text)* from the animation list and click the **Add Action** button on the toolbar. To access the User Input (Text) dialog to modify an action, select *Edit Point Value (Text)* from the animation list and click the **Modify Action** button on the toolbar. For a list of the objects to which this action is applicable refer to chapter 9, Objects.

On selection of the *Edit Point Value (Text)* action, dialog is displayed:



Edit Point Value (Text)

Text Point:

Runtime Display Attributes:

Caption: Set Text to

□ Echo Off

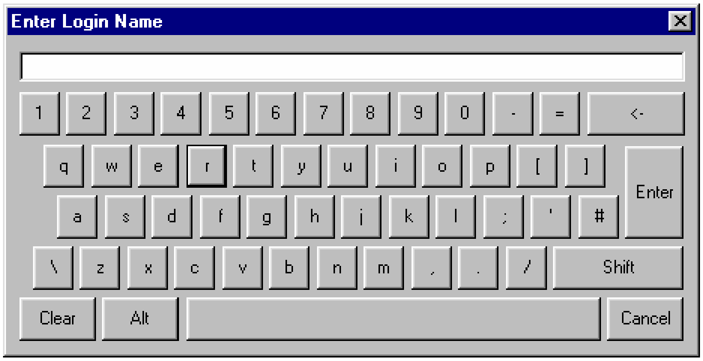
□ Display Keyboard

□ In Place Edit

A text point must be entered. This is typed into the *Text Point:* field. A text string to use as the caption of a user dialog can be entered in the *Caption:* field. Selecting the *Echo Off* setting ensures that text typed at runtime appears hidden as asterisks.

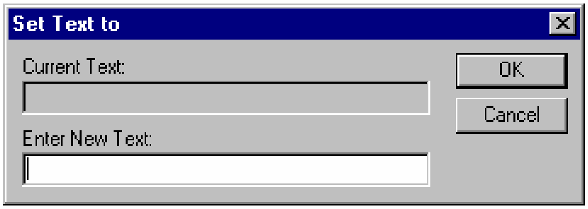
The *Display Keyboard* option is useful if the runtime system does not have a keyboard, i.e. it is controlled by a touch screen or tracker ball. When a value is edited, a Keyboard dialog is displayed which allows editing by clicking on screen.

If the *In Place Edit* option is checked, instead of the popup editing dialog, the value can be edited on the page in a standard Windows edit box.



To abort the Edit Point Value (Text) edit, click the **Cancel** pushbutton. By clicking the **Browse** pushbutton, a point may be directly specified, as described in chapter 9, Runtime Actions. Once completed, click the **OK** pushbutton. If invalid data has been inserted into any field, a descriptive error message is displayed.

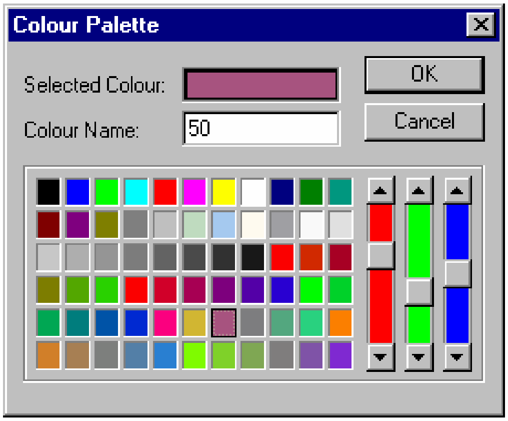
During runtime, selecting the user input object by clicking the left mouse button results in the in place edit box or runtime User Input (Text) dialog being displayed, based on the contents of the development version.



The user is provided with an instruction based on the *Runtime Display Attributes:* field, and a field in which the user enters the desired text. Click the **OK** pushbutton to accept the value, or the **Cancel** pushbutton to abort the operation. An invalid entry typed in the editable field is exposed as an error once the **OK** pushbutton has been clicked.

Common Colour Palette

If an object or action has a colour associated with it, the colour can be changed by clicking on the sample of the colour in the dialog which describes the object or action. The Colour Palette dialog is displayed so that a new colour can be specified.


Colour Palette

Selected Colour:

Colour Name: 50

A colour can be selected by clicking on one of the palette colours. This becomes the selected colour, and is denoted as such in the *Selected Colour:* field. Alternatively, a colour can be selected by specifying a colour name or number in the *Colour Name:* field. The selected colour within the palette can be mixed by using the red, blue and green scroll bars to the right of the palette.

Each colour in the colour palette is numbered from 0 to 65, number 0 located in the top-left position of the palette, with numbers reading consecutively across, then down, with colour number 65 in the bottom-right position of the palette. Additionally, colour numbers 0 to 19 are named — for instance, colour number 0 is named “black”.

Colours 0 to 15 inclusive are system colours and cannot be mixed. Although it is possible to mix new colours using the scroll bars, it is not possible to save such changes from this palette. To ensure new colours are saved, use *General Settings* from the *Project* menu.

Note: Using a 16 colour-based screen resolution (consult the Microsoft Windows documentation for further information) colours 16 to 65 are dithered from the sixteen base colours. Higher colour-based resolutions are not dithered.

CHAPTER 10 Recipes

This chapter describes CX-Supervisor recipes and the procedures associated with the creation, amendment and removal of recipes using the Recipe Editing facility. The use of recipes during the running of a project is also described.

What is a Recipe?

A recipe is a means of preparing a sequence of steps which can be repeated verbatim as and when required. A typical use for a recipe is to initialise some point data values prior to the commencement of a particular operation under the control of a CX-Supervisor project.

A CX-Supervisor recipe consists of one or more *ingredients,* each of which relates to a point. Each ingredient assigns a value to a point by means of a *target value*.

A suite of CX-Supervisor recipes may be created for any project. In runtime, the recipes are a great time-saver that eliminate mistakes that may otherwise be made were the repetitive tasks they perform left to human endeavour.

It is possible to achieve simple point initialisation using the CX-Supervisor script language. This approach works perfectly well for relatively straight-forward applications. However, further functionality is provided using the recipe system, such as the ability to modify a recipe while the CX-Supervisor project is being executed in runtime.

Recipe Components

Before proceeding any further with the description of recipes, some basic recipe terminology must be introduced:

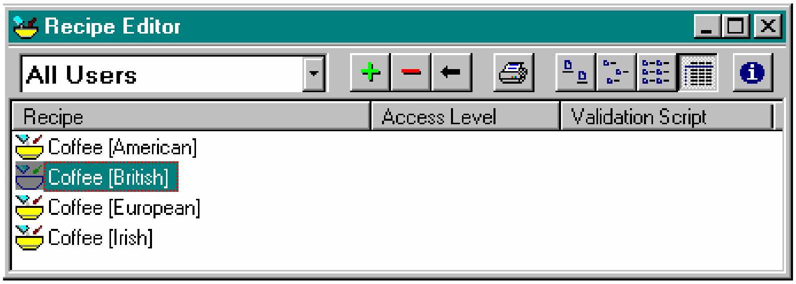
- ♦ *Recipe*. A recipe is a set of pre-defined steps used to perform a particular task. A CX-Supervisor project may contain none or many. Recipes are defined in the development environment and executed, or downloaded, in the runtime environment.
- ♦ *Ingredient*. Each recipe consists of at least one ingredient. Each ingredient must be related to an existing point.
- ♦ *Target Value*. An ingredient must specify a target value for its related point. This is the value to which the point is set in runtime when the recipe is downloaded.
- ♦ *Validation Code*. Recipe validation code is CX-Supervisor script code which is used to check point values before downloading a recipe.
- ♦ *Download*. A recipe is downloaded during runtime. This process involves identifying the appropriate recipe and executing the validation code, if any exists. The download is complete when each ingredient has set its point to the target value.

Viewing Recipes in the Recipe Editor

The Recipe Editor allows the creation, editing, copying and deletion of recipe definitions. To use the Recipe Editor, CX-Supervisor must currently have a project open. If no project is currently open, select *Open* from the *Project* menu to open a previously saved project, or select *New* from the *Project* menu to create a new project.

To open the Recipe Editor dialog, click the **Recipe Editor** button on the toolbar.

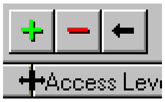
Display the Recipe Editor as described in the previous chapter. An example of the Recipe Editor dialog is shown below:



Recipe	Access Level	Validation Script
Coffee [American]		
Coffee [British]		
Coffee [European]		
Coffee [Irish]		

Recipes are displayed in three columns: *Recipe, Access Level* and *Validation Script.* Recipes are usually listed in alphabetical order by recipe name, although the ordering can be changed so that the entries are listed according to any of the three column titles.

For example, to see the recipes listed alphabetically by security access level, simply click on the *Access Level* field.



The widths of the recipe fields can be widened or narrowed as required using the mouse to drag the column boundaries.

Select the **Large Icons** button to view details with large icons.

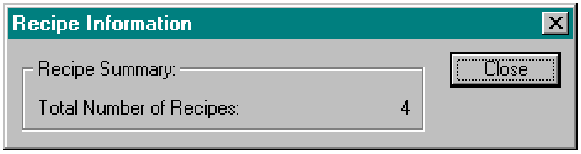
Select the **Small Icons** button to view details with normal icons.

Select the **List** button to view details as a list.

Select the **Details** button to view details as a list including recipe, access level and validation script. The details can be sorted in ascending order by clicking once or in descending order by clicking twice, in the appropriate field.

A summary of recipe information is available by selecting the **Recipe Information** button from the toolbar.

The resultant dialog shows the overall number of recipes in the project. To exit the dialog click the **Close** pushbutton. The Recipe Information dialog is shown as follows:

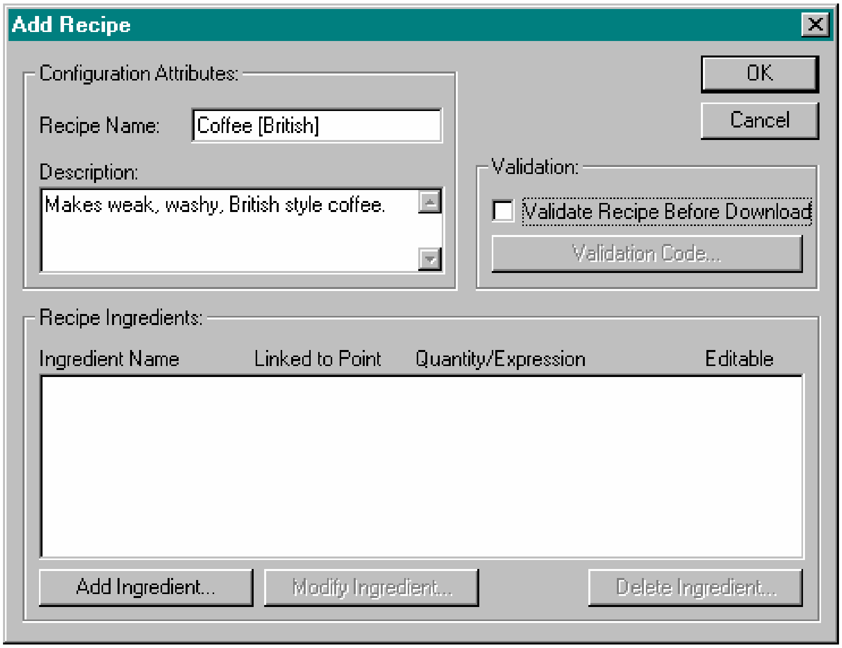

Recipe Information

Recipe Summary:

Total Number of Recipes: 4

Creating a New Recipe

Open the Recipe Editor as described in chapter 10, Viewing Recipes in the Recipe Editor. To add a new recipe, select the **Add Recipe** button from the toolbar. This results in the Add Recipe dialog being displayed.


Add Recipe

Configuration Attributes:

Recipe Name: Coffee [British]

Description: Makes weak, washy, British style coffee.

Validation:

☐ Validate Recipe Before Download

Recipe Ingredients:

Ingredient Name	Linked to Point	Quantity/Expression	Editable

Once all the information has been provided for the new recipe, selecting the **OK** pushbutton adds the new recipe definition to the project, whilst the **Cancel** pushbutton aborts this add operation.

Note: The short-cut keyboard combinations for Cut, Copy and Paste operations are valid within the Add Recipe dialog. Highlight part or all of a field and type <Ctrl>+X to cut the text or <Ctrl>+C to copy the text; insert the cursor at the desired field and type <Ctrl>+V to paste the text. Since the cut and copy operations store the information in the Windows Clipboard, they may be pasted to another dialog or application.

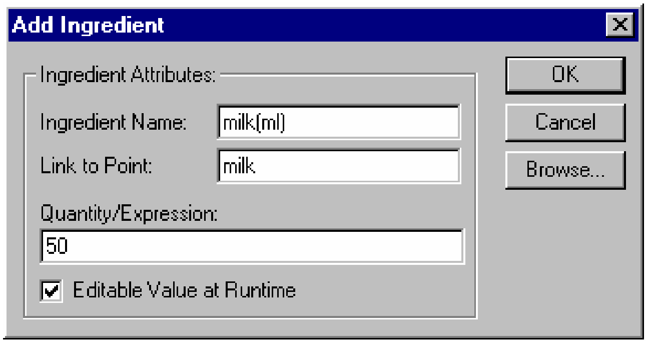
Recipe Configuration Attributes

The name of the recipe is entered in the *Recipe Name:* field. The recipe name can be constructed using any alphanumeric characters. This field is visible from the Recipe Editor dialog.

A recipe description may be inserted in the *Description:* field. To start a new line, press <Ctrl>+<Return>. Any characters are valid and up to 255 characters may be entered in the text field. This field is also visible from the Recipe Editor dialog.

Recipe Ingredients

The recipe ingredients are added using the **Add Ingredient** pushbutton in the Add Recipe dialog to display the Add Ingredient dialog, as follows:


Add Ingredient

Ingredient Attributes:

Ingredient Name: milk(ml)

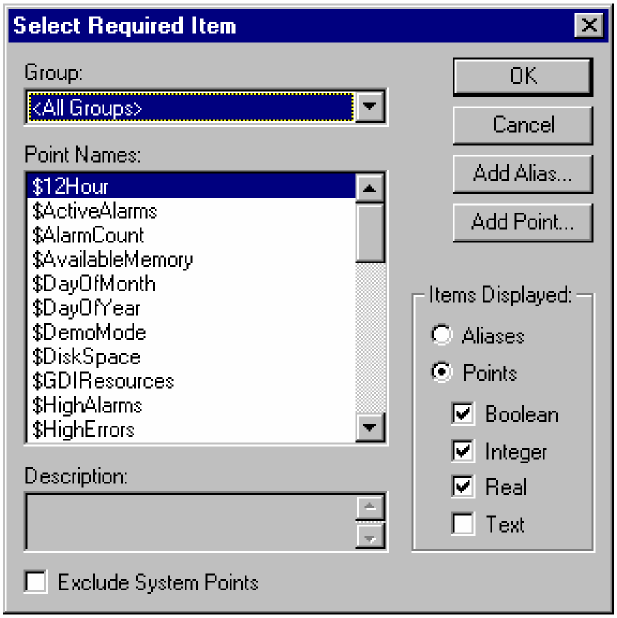
Link to Point: milk

Quantity/Expression: 50

☑ Editable Value at Runtime

Enter a suitable ingredient name in the *Ingredient Name:* field. This name should be unique and meaningful and identify the particular ingredient being added to the recipe.

Enter the name of a point which is initialised by this ingredient in the *Link to Point:* field. The adjacent **Browse** pushbutton may be used to display the Select Required Item dialog, which provides a list of points from which the selection may be made. The Select Required Item dialog also provides an **Add Point** pushbutton which allows a new point to be added. See also chapter 3, Points. The *Quantity/Expression* field is used to define the value which is assigned to the point by this ingredient. This may be a fixed value, such as 50 as in the above expression, or it may be any valid CX-Supervisor script language expression featuring one or more point names.



Select Required Item

Group: <All Groups>

Point Names:
$12Hour
$ActiveAlarms
$AlarmCount
$AvailableMemory
$DayOfMonth
$DayOfYear
$DemoMode
$DiskSpace
$GDIResources
$HighAlarms
$HighErrors

Description:

□ Exclude System Points

Items Displayed:
○ Aliases
● Points
☑ Boolean
☑ Integer
☑ Real
□ Text

The *Editable Value at Runtime* setting determines whether the target expression may be changed at runtime when the recipe is downloaded. Only expressions featuring a fixed value may be edited in this way; expressions featuring point names may not be edited at runtime.

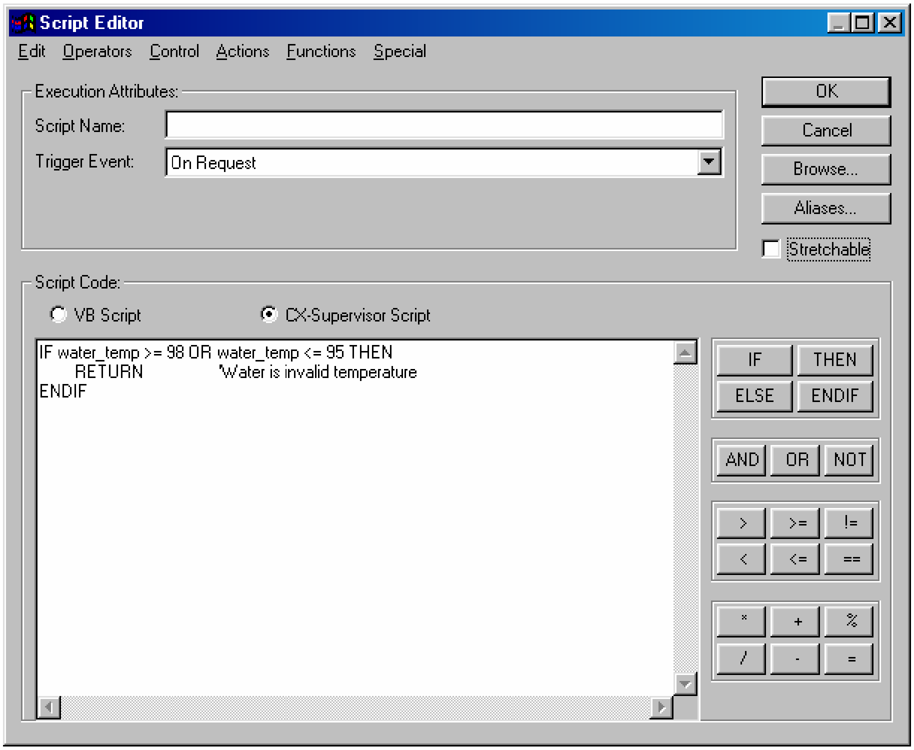
Select the **OK** pushbutton to add the ingredient to the recipe, or the **Cancel** pushbutton to abort this part of the operation.

Recipe Validation

Recipe validation is an optional safety check which may be made in runtime as the recipe is about to be downloaded. If the validation fails, the recipe is not downloaded.

If no validation is required, make sure the *Validate Recipe Before Download* setting is set to ‘OFF’.

If validation is required, check this box and then click the **Validation Code** pushbutton to display the Script Editor dialog:



Execution Attributes:

Script Name:

Trigger Event: On Request

□ Stretchable

Script Code:

○ VB Script ● CX-Supervisor Script

```
IF water_temp >= 98 OR water_temp <= 95 THEN
      RETURN              'Water is invalid temperature
ENDIF
```

Enter the recipe validation code using any standard CX-Supervisor script language constructs. The validation code should check any possible invalid point value or error condition. The point value being checked is the value of the ingredient about to be downloaded, or the current point value if the point is not an ingredient of the recipe being downloaded.

In the event of a failure, the script language should perform a premature <Return>. This has the effect of failing the recipe validation check and abort the attempt to download the recipe.

The recipe validation code should **not** contain any command which requires user input (e.g. a Message command). This is because the validation script would not have been completed (and appropriate action taken) at the time that user input is requested. Instead, if a message box is required, then a point should be used as an error status value, and set to an appropriate value. This point can then be checked in an ‘on condition’ script and the appropriate message displayed from there.

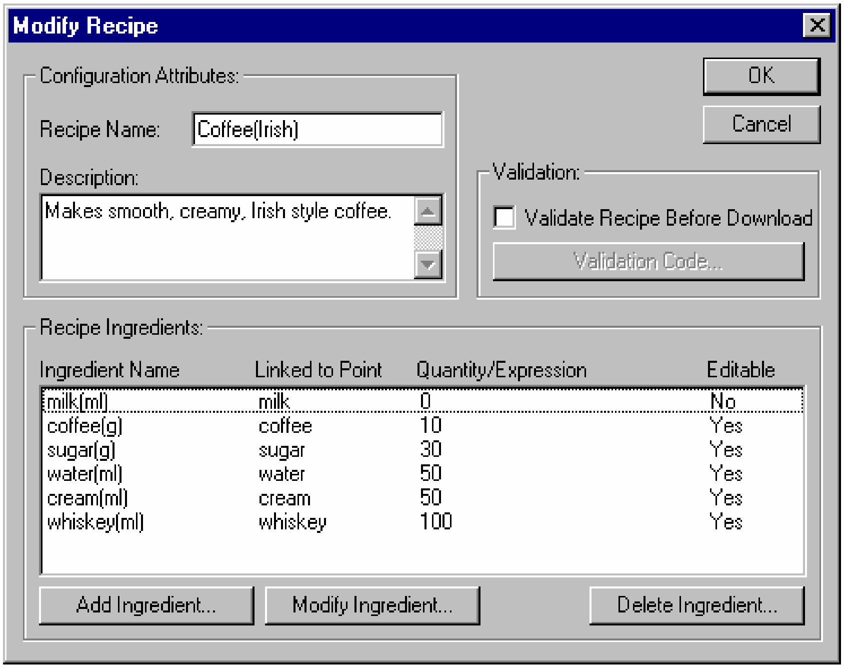
If the validation code exits normally, the recipe is downloaded in the normal way.

Updating an Existing Recipe

Open the Recipe Editor dialog as described in chapter 10, Viewing Recipes in the Recipe Editor.

To modify an existing recipe, highlight the recipe entry from the recipe list and select the **Modify Recipe** button from the toolbar.

This results in the Modify Recipe dialog being displayed, a dialog based on the Add Recipe dialog, as shown below:



Recipe Name: Coffee(Irish)

Description: Makes smooth, creamy, Irish style coffee.

☐ Validate Recipe Before Download

Ingredient Name	Linked to Point	Quantity/Expression	Editable
milk(ml)	milk	0	No
coffee(g)	coffee	10	Yes
sugar(g)	sugar	30	Yes
water(ml)	water	50	Yes
cream(ml)	cream	50	Yes
whiskey(ml)	whiskey	100	Yes

The selected recipe can be redefined as described in chapter 10, Creating a New Recipe.

The **Modify Ingredient** and **Delete Ingredient** pushbuttons on this dialog respectively allow the highlighted ingredient to be modified, or deleted (following confirmation).

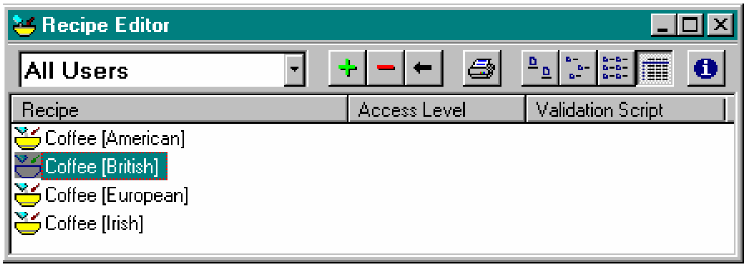
Once all the information has been provided for the updated recipe, clicking the **OK** pushbutton saves the recipe details, whilst the **Cancel** pushbutton aborts this modify operation.

Copying an Existing Recipe Definition

Open the Recipe Editor dialog, as described in chapter 10, Viewing Recipes in the Recipe Editor.

To copy an existing recipe, highlight the recipe from the recipe list and then type <Ctrl>+C. This creates a copy of the recipe definition on the clipboard. To paste the recipe definition back in to the current project, type <Ctrl>+V. The cut, copy and paste operations in the *Edit* menu may also be used. The Recipe Editor ensures the name of the recipe is unique by appending a number to the name. For example, if the original recipe name was “Recipe”, the name of the pasted recipe is “Recipe1”. All other properties of the pasted recipe remain the same as the original recipe definition.

It is possible to copy many recipes at once by highlighting all the desired recipes in the recipe list. Press and hold the <Ctrl> key and use the mouse to select recipe definitions one by one, or press and hold the <Shift> key and use the mouse to select blocks of recipe definitions. Once the desired recipes have been highlighted, the recipe definitions may be copied and pasted in the usual way.



Recipe	Access Level	Validation Script
Coffee [American]		
Coffee [British]		
Coffee [European]		
Coffee [Irish]		

Deleting an Existing Recipe

Open the **Recipe Editor** dialog, as described in chapter 10, Viewing Recipes in the Recipe Editor.

To remove an existing recipe, highlight the recipe from the recipe list and select the **Delete Recipe** button from the toolbar.

A confirmation dialog is displayed. Click the **Yes** pushbutton to remove the definition, or the **No** pushbutton to abort the delete operation.

Recipe Security Levels

A recipe may be assigned a security level governing which level of users are allowed to download the recipe in runtime. The available security levels are shown below in descending order:

- Designer;
- Manager;
- Supervisor;
- Operator;

- All Users.

To assign a security level to a recipe, display the Recipe Editor dialog. Then select the recipe to which the security level is applied. With the recipe highlighted from the list, click the *Security Level* field on the toolbar.

Select the required security level from the list. The selected security level is then applied to the recipe. To change the security level, click the *Security Level* field again and choose a different security level from the list.

Printing Recipes

Print Setup

The Recipe Editor can be printed in the same way pages can. Before printing, ensure that the printer has been set up correctly. To check the printer settings, refer to chapter 2, Pages.

Print Preview

To preview the page before printing, select *Print Preview* from the *File* menu.

Refer to chapter 2, Pages regarding the use of the Print preview display.

Printing

To print the contents of the Recipe Editor, select the **Print** button from the toolbar.

Refer to chapter 2, Pages regarding the use of the Print dialog.

It is possible to print the details of a recipe including a list of all the ingredients, points and expressions which comprise the recipe. To do this, display the Recipe Editor dialog as described earlier and then highlight the appropriate recipe from the list of recipes. Click the **Print Recipe** button on the toolbar to send a copy of the recipe to the printer. The recipe is formatted similar to the following example:

```
Recipe: Coffee(British)
Description: Makes weak, washy, British style coffee.
Access Level: All Users

Ingredient          Point          Expression        Editable
milk(ml)            milk           50                Yes
coffee(g)           coffee         3                 Yes
sugar(g)            sugar          0                 Yes
water(ml)           water          250               Yes
```

Using Recipes in Runtime

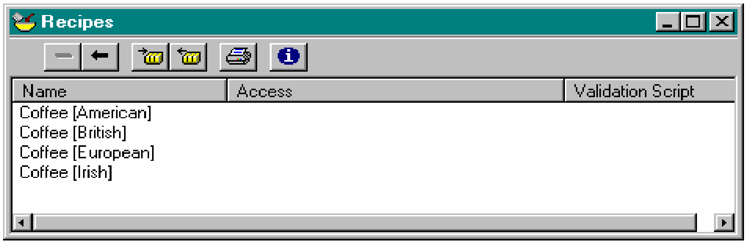
Recipes defined using the Recipe Editor in the development environment can be accessed in runtime using the Recipe Viewer. Recipes may be downloaded using the Recipe Viewer.

Recipe Viewer

The Recipe Viewer can be displayed in one of two ways in the runtime environment. It may be displayed by any CX-Supervisor script code attached, for example, to a pushbutton. The function call is simply ‘DisplayRecipes()’.

The Recipe Viewer may also be displayed using the floating menu. The floating menu needs to be configured to display the *Recipes* option. This is done in the development environment and explained in chapter 6, Projects.

The Recipes dialog is displayed as follows:

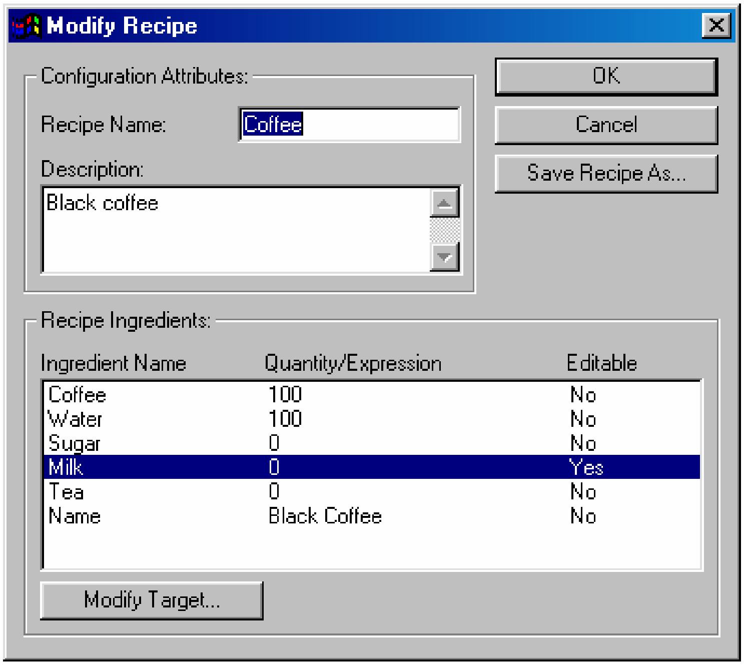


Name	Access	Validation Script
Coffee [American]		
Coffee [British]		
Coffee [European]		
Coffee [Irish]		

The Recipe Viewer is similar to the Recipe Editor, however there are some differences in functionality and appearance:

− To remove an existing recipe, highlight the recipe from the recipe list and select the **Delete Recipe** button on the toolbar. A confirmation dialog is displayed. Click the **Yes** pushbutton to remove the definition, or the **No** pushbutton to abort the delete operation. Note that recipes created in the development environment may *not* be deleted in the runtime environment. The delete operation only affects recipes copied and saved in the runtime environment.

← To modify an existing recipe, highlight the recipe from the recipe list and select the **Modify Recipe** button from the toolbar. The Modify Recipe dialog is displayed as follows:



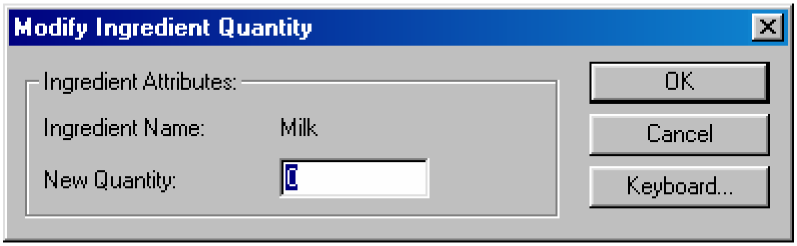
Recipe Name: Coffee

Description: Black coffee

Ingredient Name	Quantity/Expression	Editable
Coffee	100	No
Water	100	No
Sugar	0	No
Milk	0	Yes
Tea	0	No
Name	Black Coffee	No

This runtime version of the dialog is rather more limited in the power it offers than the equivalent dialog in the development environment; this is so that recipes which have been painstakingly created in the development environment cannot be accidentally overwritten or destroyed in the runtime environment. The two pushbuttons unique to the Runtime of the dialog are **Modify Target** and **Save Recipe As**.

With the appropriate ingredient selected from the list, clicking the **Modify Target** pushbutton allows the target value to be modified by means of the Modify Ingredient dialog, shown below:



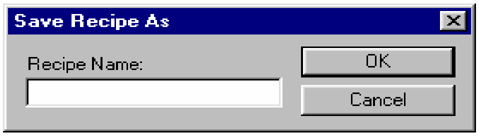
Ingredient Name: Milk

New Quantity: 0

The target value for this ingredient may be changed by entering the new value in the *New Target:* field. Selecting the **OK** pushbutton accepts the change, choosing the **Cancel** pushbutton aborts the change.

Note: The new target value is applied *only* to this recipe download. No permanent change is made to the recipe definition itself.

The **Save Recipe As** pushbutton displays a simple dialog prompting for a name to use for the new recipe.



Clicking the **OK** pushbutton creates a new recipe with the name specified. Choosing the **Cancel** pushbutton aborts the save operation. A recipe created using this dialog is added to the available list of entries displayed in the recipe dialog. The newly saved recipe is also available in the development environment using the Recipe Editor.

Note that it is possible to change the name of an existing recipe in runtime simply by typing a new name in the *Recipe Name:* field of the Modify Recipe dialog and then clicking the **OK** pushbutton. The re-named recipe appears in the available list of entries displayed in the Recipes dialog and is also available in the development environment using the Recipe Editor.

To download an existing recipe, highlight the recipe from the recipe list and select the **Download Recipe** button from the toolbar. See chapter 10, Downloading a Recipe for more information.

To upload an existing recipe, highlight the recipe from the recipe list and select the **Upload Recipe** button from the toolbar. See chapter 10, Uploading a Recipe for more information.

Click the **Print Recipe** toolbar button to send a copy of the recipe to the printer. See chapter 10, Printing Recipes for a typical example of the formatted output.

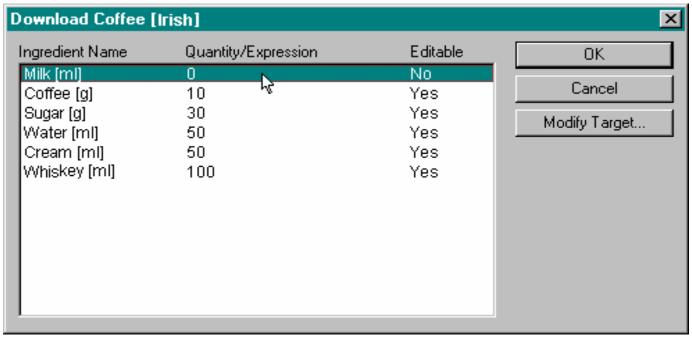
A summary of recipe information is available by selecting the **Recipe Information** button from the toolbar. The Recipe Information dialog is displayed (see chapter 10, Viewing Recipes in the Recipe Editor for an example).

Downloading a Recipe

There are two ways of downloading a recipe definition in runtime. The first method is to use a CX-Supervisor script function attached to a graphical object on a page, such as a push button. The function call is:

```
DownLoadRecipe(“<NameOfRecipe>”)
```

The other way to download a recipe, is to display the Recipe Viewer as described in the previous chapter. Highlight the desired recipe from the recipe list and click the **Download Recipe** pushbutton. The Download Recipe dialog is displayed as follows:



Ingredient Name	Quantity/Expression	Editable
Milk [ml]	0	No
Coffee [g]	10	Yes
Sugar [g]	30	Yes
Water [ml]	50	Yes
Cream [ml]	50	Yes
Whiskey [ml]	100	Yes

The **Modify Target** pushbutton allows the target value for an ingredient to be modified by means of the Modify Ingredient dialog (shown above). An ingredient must already be selected from the list of ingredients for this recipe. Any change to the target value is of a temporary nature and is not permanently changed in the recipe definition. Any number of ingredients may be modified in this way before the recipe is downloaded.

Selecting the **OK** pushbutton attempts to download the recipe. If for some reason the recipe definition can’t be found, the following error message appears: *“Unable to find recipe”*.

One recipe download must be completed before another recipe download is started. If a recipe download is already in progress when another is started, the following error message appears: “A recipe download is currently in progress”.

If any *validation code* has been entered for this recipe, the validation is conducted first. If the validation is successful, the recipe is downloaded. If the validation fails, the recipe is not downloaded and the following error message appears: “Unable to download selected recipe”.

If there is no validation code to execute, the recipe is downloaded anyway. The download process consists of setting each ingredient to its specified target value.

The download may fail at this stage if there are problems communicating with a PLC. A communications failure results in the error message *“Downloading recipe failed”*. A delay in response from the PLC of more than one minute results in the error message *“Recipe transfer timed out”*.

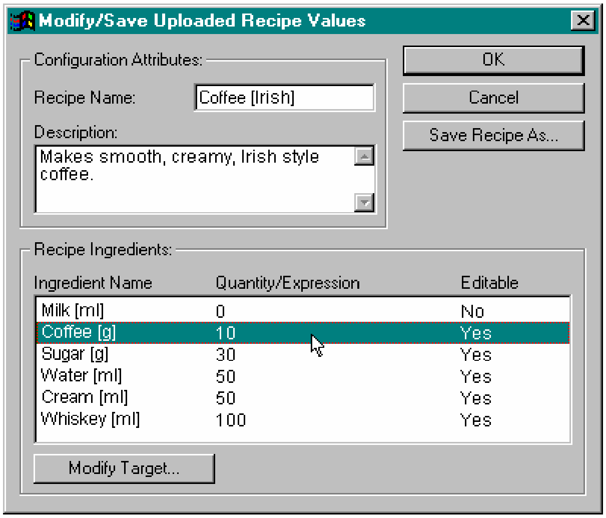
When the recipe download is complete, all the ingredients are saved and the CX-Supervisor runtime environment continues to run in the normal manner. An event is logged stating that the recipe was successfully downloaded.

Uploading a Recipe

There are two ways of uploading a recipe definition in runtime. The first method is to use a CX-Supervisor script function attached to a graphical object on a page, such as a push button. The function call is:

```
UpLoadRecipe("<NameOfRecipe>")
```

The other way to upload a recipe, is to display the Recipe Viewer as described in the previous chapter. Highlight the desired recipe from the recipe list and click the **Upload Recipe** pushbutton. The Modify/Save Upload Recipe dialog is displayed as follows:



Recipe Name: Coffee [Irish]

Description: Makes smooth, creamy, Irish style coffee.

Ingredient Name	Quantity/Expression	Editable
Milk [ml]	0	No
Coffee [g]	10	Yes
Sugar [g]	30	Yes
Water [ml]	50	Yes
Cream [ml]	50	Yes
Whiskey [ml]	100	Yes

The **Modify Target** pushbutton allows the target value for an ingredient to be modified by means of the Modify Ingredient dialog. An ingredient must already be selected from the list of ingredients for this recipe. Any change to the target value is saved in the recipe definition. Any number of ingredients may be modified in this way when the recipe is uploaded.

Selecting the **OK** pushbutton will upload the recipe. If for some reason the recipe definition can't be found, the following error message appears: *"The recipe failed its validation checks. Upload aborted"*.

Each recipe upload must be completed before another can be started. If a recipe upload is already in progress when another is started, the following error message appears: "A recipe Upload or Download is currently in progress".

The upload may fail at this stage if there are problems communicating with a PLC. A communications failure results in the error message *"Uploading recipe failed"*. A delay in response from the PLC of more than one minute results in the error message *"Recipe transfer timed out"*.

When the recipe upload is complete, all the ingredients are saved and the CX-Supervisor runtime environment continues to run in the normal manner. An event is logged stating that the recipe was successfully uploaded.

CHAPTER 11
Data Logging

This chapter describes the CX-Supervisor data logging facilities including how to configure items to be logged, the logging of data and the viewing and exporting of recorded data.

What is Data Logging

The concept of the Data Logger is the ability to define a number of points and expressions during development which may be recorded while the CX-Supervisor project is being executed in runtime. These events may be viewed at runtime and recorded for future evaluation. They may also be exported to other programs such as Microsoft Excel.

The recorded events are stored in one or more Data Sets, which can then be viewed using the Data Log Viewer. The events to be recorded are defined by the developer and any number of Data Sets may be used to record specific areas or types of events either automatically at runtime or as specified by the developer.

As an aid to viewing a number of particular events in isolation Data Sets can be subdivided into data Groups. Specific areas of related data can then be stored in these groups which can then be viewed either singly or with other groups within a Data Set.

The main elements of the CX-Supervisor Data Logger are:

- **Data Logger Editor:** used to configure items to be logged
- **Data Logging at Runtime:** the actual logging of the data
- **Data Log Viewer:** used to view the logged data
- **Remote Data Viewer:** enables viewing separately from the runtime system
- **Data Log Export facility:** enables export to other programs (e.g. MS Excel)
- **Script Functions:** gives full control over the logging process

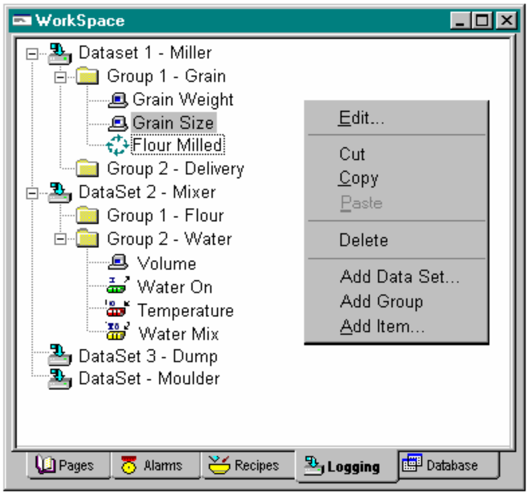
Data Log Editor

Configuring Data Sets and Logging Settings

The Data Sets, data Groups and Items to log are configured from the Logging Tab on the bottom of the Development Workspace editor. Items are edited using a right mouse-button context sensitive menu. The option available from the menu are:

- Edit: enable the parameters of the selected Data Set, Group or Item to be edited.
- Cut, Copy, and Paste: enables the selected Item to be cut or copied and then pasted into a Data Set or Group.
- Delete: will delete the selected entry.

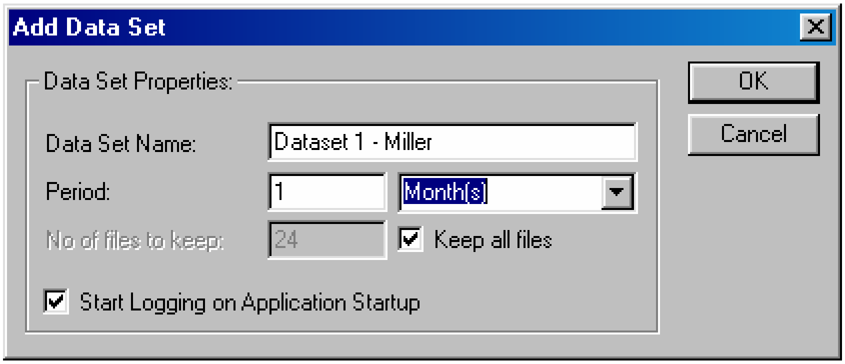
- Add DataSet, Add Group and Add Item: enables new Data Sets, Groups and Items to be added to the selected entry.



The functionality of the Items is determined by their expression. If the expression is a single point name, the item is shown as the same type as the point as configured in the point editor, i.e. memory, input, output or input/output. If the expression contains calculations, constants and/or multiple point names, the item is shown as a script calculation.

Adding/Editing Data Set Properties

The **Add/Modify Data Set** properties dialog is displayed when either the **Add Data Set** or the **Edit** (an existing Data Set) option is selected from the menu.



Data Set Properties:

Data Set Name: Dataset 1 - Miller

Period: 1 Month(s)

No of files to keep: 24 ☑ Keep all files

☑ Start Logging on Application Startup

Data Set Properties

The *Data Set Name* field enables the Data Set to be referenced with a meaningful description both in the Workspace view, and also from the script language. It also forms the prefix for the file name. A maximum of 26 characters is allowed.

The *Period* fields allow a period to be entered in Hours, Days, or Months and defines the period of data to be grouped into a single file. For example, if the Period is set to 1 month, any data logged for the current month is saved to the same data file, even when the system is stopped and restarted. When logging spans a Month boundary the data file is closed, and a new file started and given sequential file name. This allows easy archiving and backup of the logged data. The start of a new data file is always synchronised to a natural boundary, e.g. for months - the 1st of the month, for days – 24.00 hours and for hours - on the hour.

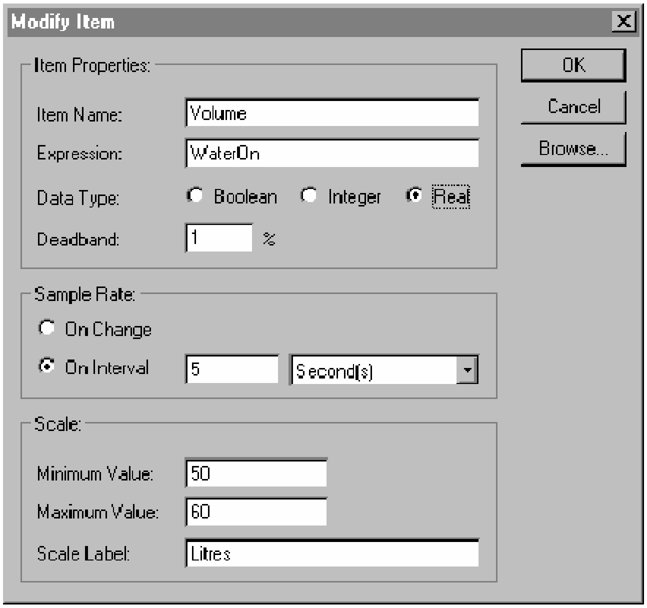
The *No. Of Files to Keep* entry determines the maximum number of data files that are to be kept. Once the number of files kept has reached the value entered, as each new file is created the oldest file is deleted. For example, to keep process data for 2 years, you could keep 24 files of 1 month period. The file names are generated automatically based on the Data Set name, plus a suffix which is incremented by one as each new file is generated.

If the *Keep all Files* option is checked the data files management is disabled and no files will be deleted.

The *Start Logging on Application Startup* checkbox controls the logging of the Data Set. When selected, all the Items in the Data Set start logging when the system is started, and stop logging when the system is stopped. If this option is not selected, then logging must be controlled by the Start/Stop logging script functions as required by the developer.

Editing Item Properties

The **Add/Modify Item** properties dialog is shown when adding a new **Item**, or editing an existing one.


Modify Item

Item Properties:

Item Name: Volume

Expression: WaterOn

Data Type: ○ Boolean ○ Integer ● Real

Deadband: 1 %

Sample Rate:

○ On Change

● On Interval 5 Second(s)

Scale:

Minimum Value: 50

Maximum Value: 60

Scale Label: Litres

Item Properties

These options enable an items name and associated expression to be entered. Its data type can be selected and if required a deadband value can be applied to the measured value.

The *Item Name* field allows the Item to be given a meaningful name. The entry should be kept short but also be logical for clarification. A maximum of 26 characters is allowed.

The *Expression* field defines the point name or expression that will be logged. The Browse button allows easy point selection. For details on how expressions or points are formulated Refer to chapter 3, Points.

The *Data Type* field determines the type of data to be logged. Select either Boolean, Integer or Real. For points, this is automatically picked up from the points database. For expressions, the required type must be selected.

The *Deadband* field enables a percentage limit or 'Deadband' to be applied to the logging expression. This allows the value of the expression to change within the percentage limit without being logged. This is particularly useful when logging analogue values which are subject to noise. Each sample is checked (when the value changes for 'On Change' sampling, or every log period for 'On Interval') against the last logged value. While the value varies within the deadband, the values are ignored. Immediately the value exceeds the deadband limits the new value will be logged. This may dramatically reduce file size, and ease analysis.

Entering a value of 0%, i.e. no deadband, will cause every sample to be logged.

Sample Rate

The sample rate options are used to determine how data will be recorded, select either on change or on interval. The default is interval.

When the *On Change* option is selected data will be logged only when the value actually changes. This ensures all transient information is logged - even if changes occur faster than maximum sample rate, and also ensures periods of inactivity do not generate duplicated data.

The *On Interval* sample rate option enables the data values to be logged at regular intervals. Enter the interval as a number and from the pick list select the form, e.g. '5 Second'. This method guarantees the value is checked and recorded at every interval. However transients occurring between sample periods are not recorded, which depending on the application may be a benefit.

Scale

The scale fields enable the graph upper and lower limits to be set and a scale label entered for the selected item.

Enter the *Minimum value* as the lower limit to be shown on the graph for this item.

Enter the *Maximum value* as the upper limit to be shown on the graph for this item.

Enter the *Scale Label* to be displayed. This is a text field the contents of which are displayed by the Y scale of the graph. The entry should be kept short but also be logical for clarification.

Editing Items

Copy, Cut, Paste Options

The Copy, Cut and Paste options follow the windows convention and enable a selected Item to be copied or cut from one Data Set Group and then pasted into another. As Item names within a Data Set must be unique, when a copied Item is pasted back into the same Data Set the name is given a numerical suffix.

New or existing Items can be renamed and have their parameters edited by high lighting the Item and then selecting the **Edit** option from the popup menu. When the Modify Item dialog appears edit the appropriate parameters in the normal way.

Adding/Renaming Groups

The adding or renaming of Data Set groups follows the windows convention. New Groups are added by selecting the Data Set to which the group is to be added and then selecting the **Add Group** option from the popup menu. The new Group is automatically placed in the correct position in the Data Set and given a default group number. New or existing Groups can be renamed by first high lighting the current group name and then selecting the **Edit** option from the popup menu. When the edit box is displayed the name can then be edited in the normal way. A maximum of 26 characters is allowed.

Unwanted groups can only be Deleted, they can not be cut and pasted.

Data Logging at Runtime

Overview

All data logging is performed on Items, which are stored in a Data Set. The System can contain many Data Sets and each Data Set can contain many Items. Each Data Set will have its own set of files when the System is run (see the chapter 11, File Management). The amount of data that is stored in a single Data Set file is determined by the specified period when the Data Set is created.

File Management

Data Logging Directory

All data log files are stored in the directory "Data Logging" separate from the CX-Supervisor files. This directory is created automatically when the project runs for the first time. A sub-directory is used as a safeguard against the accidental deletion of important files by the automatic purge mechanism and to prevent the project directory from becoming cluttered with data files.

For example, if the CX-Supervisor project is in a directory called **C:\CX-Supervisor\Project** then, when the project runs for the first time a new directory called **C:\CX-Supervisor\Project\Data Logging** will be created to hold the data log files.

Data Logging Files

Each Data Set has its own set of files. The following conventions have been adopted in order to identify the various files and add time and date information.

Data Log files are assigned the file extension .dlv "Data Log Values" and the format of the filename is:

```
<Data Set Name><Time Stamp>.dlv
```

The Time Stamp is in the format: [YYYY MM DD HH]"

where YYYY = Year in the form 1999

MM = Month in the form 01 (January)

DD = Day of the month in the form 01 – 31

HH = Hours in the form 00 –23

Daylight time saving is automatic.

For example: `Data Set 1 Miller[1999012015].dlv`

A new log file will be created automatically whenever the preceding file has expired the timestamp. The timestamp is also used to distinguish between files in the same Data Set. The time period for each file is determined by the options selected when the Data Set was created. The minimum time period for a Data Set file is one hour commencing on the hour.

Note: Do not rename any log files while they are in the Data Logging directory or the File Management System will not work correctly. The File Management System relies on the strict format of the Data Log files to determine which file to open as Live and which file(s) to delete during purging and also for performing 'previous/next' file searches in the Data Log Viewer.

Note: Additional files should never be copied into the Data Logging directory as the automatic purging mechanism may delete all the log files.

Number of files in a Data Set

When a Data Set starts logging a file will be created to hold the logged items, this file is known as a 'Live' or 'Active' log file (There is only one Live file in a Data Set file set). When the time period for a Data Set log file expires a new file is created with a more recent Timestamp, this file then becomes the 'Live' file and the previous file is now known as a 'Dead' log file. Over a period of time there can be many 'Dead' files in a Data Set but only one live one. If a limit has been set on the number of dead files to be kept when this number is reached the oldest file will be purged automatically as each new file is created.

The number of dead files to be kept for a particular Data Set is managed automatically by specifying the 'Number of Files to Keep' from the Data Set Properties Parameters when a Data Set is created. This field is only enabled when the 'Keep all files' check box is 'unchecked'.

Note: There will always be one Live file for each Data Set.

For example, if the number of files to keep is set to 24 then there the system will keep 24 Dead files and one Live file, i.e. 25 files in total.

WARNING: **No backups** are taken when a file is purged. If you wish to keep any data you must ensure that backups are taken before purging occurs.

If the 'Keep all files' check box is checked no automatic purging will take place and the System will store all the Dead files created until the disk is full.

Dead files can be copied, moved, deleted etc. as they will not be written to again by the System.

Live files may be copied (i.e. backed up to another directory) while the Log file is open but they cannot be moved or deleted as they are locked by the System. In order to delete or move a live file it must first be closed. See Opening and Closing Log Files.

Invalid Data Log Files

Data log files generated during one project sessions will continue to be used in subsequent sessions, if they are still live. For example, if a system runs during the day and is shut down at night, then a Data Set with a duration of 7 days will use the same file for the whole 7 days.

If a session is halted and Data Set Items changed, deleted or new ones added it is not valid to continue using the existing file. When the session is restarted and Data Set file opened CX-Supervisor checks to see if there are any differences between the current Data Set and the Data Set stored in the file. If any differences are detected the System will mark the existing file as invalid by altering the Timestamp brackets from "[ ]" to "{{ }}" and create a new file.

For example, if a file **`Batch1[2000032922].dlv`** is detected as being invalid it will be renamed **`Batch1{{2000032922}}.dlv.`** A new **`Batch1[2000032922].dlv`** file will then be created to log the new data.

If the file **`Batch1{{2000032922}}.dlv`** already exists then the current invalid file will be renamed **`Batch1{{2000032922}}_1.dlv`** and so on.

This is most likely to occur during the development of an application when Data items are continually being amended and tested.

Invalid files are not part of the Data Set files and can not be purged or found by the File Management 'next' or 'previous' searches. The data recorded in these files is not lost and can be viewed using the Data Log Viewer and exported from them using Export Log. Invalid files can only be deleted using Windows explorer.

Data Set Period Examples

The following examples show how the specified period (Hour(s), Day(s), Month(s)) is implemented while the System is running:

1) Data Set period = Hours

If a Data Set named CV500 starts logging at 12:30 on the 7th March 1999 with a period of 6 hours, the following files will be created:

```
CV500[1999030712].dlv
```

If the s7stem is left running the Data Logging directory will contain the following files:

```
CV500[1999030712].dlv
CV500[1999030718].dlv
CV500[1999030800].dlv
CV500[1999030806].dlv
....
....
```

The first file will only contain 5.5 hours of Data because the minimum resolution is one hour and the file was created on the half-hour, all subsequent files will contain the full six hours of data.

2) Data Set period = Days

If a Data Set named Batch1 starts logging at 23:00 on the 29th December 1999 with a period of 1 Day, the following files will be created:

```
        Batch1[1999122923].dlv
```

If the system is left running the Data Logging directory will contain the following files:

```
        Batch1[1999122923].dlv
    Batch1[1999123000].dlv
    Batch1[1999123100].dlv
    Batch1[2000010100].dlv
    ....
    ....
```

The first file will only contain 1 hour of data, all subsequent files contain the full 24 hours of data.

3) Data Set period = Months

If a Data Set named Shifts starts logging at 22:00 on 29th February 2000 with a period of 2 Months, the following file will created:

```
        Shifts1[2000032922].dlv
```

If the system is left running the Data Logging directory will contain the following files:

```
    Shifts[2000050100].dlv
    Shifts[2000070100].dlv
    ....
    ....
```

The first file will only contain 1 month and 2 hours of data, all subsequent files contain the full 2 Months of data.

Data Records

There are two types of records that are stored in a Data Log file for each Item logged these are **Events** (or breaks) and the **Actual Data**.

Event/Break Records

Event records currently consist of the following fields:

Event, Date, Time
Event = Open | Close | Clear | Start | Stop

Events have no associated data

Data Records

Data records currently consist of the following fields:

Type, Date, Time, Milliseconds, Data
Type = Normal | Error
Data = Boolean | Integer | Real

Data Log Viewer Component

Invoking the Data Log Viewer

The Data Log Viewer can be invoked by

Selecting Data Log Viewer from the popup menu (if privileged) in the runtime.

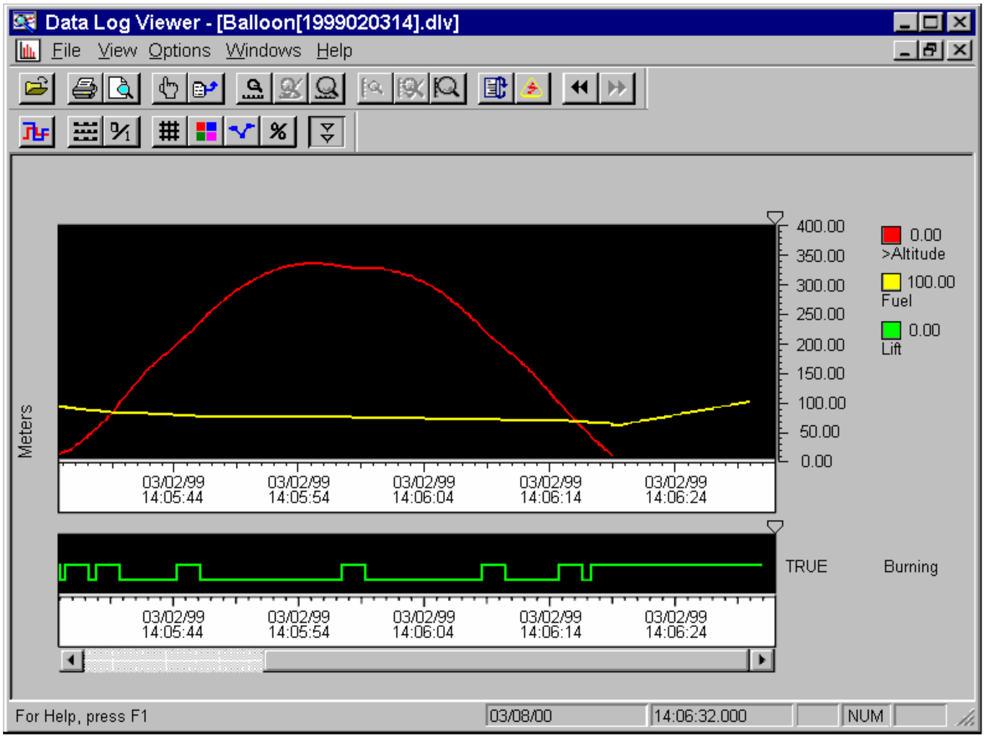
From script function ‘OpenLogView’ with the Data Set and Traces to show as arguments, and also closed by ‘CloseLogView’ script function.

From an external program.

Viewing Logged Files

The Data Log Viewer looks like this:

The window shows a graphical representation of the recorded data.



Features:

- The Data Set viewed is the name passed as an argument with the script function, or a user-selected Data Set.
- On start-up, a dialog allowing selection of traces to view from the Data Set will be shown. Up to 10 analogue traces and 50 digital traces may be shown at once. Alternatively, script parameters can be used to automatically select the traces to show.
- On display, the viewer displays:
  - the latest data file for this Data Set
  - the X axis set to the Data Set’s period
  - Y Axis set to full scale for the first trace.
- The selected analogue traces are shown overlaid.
- Boolean points are displayed in a separate view
- The X Axis shows true Date and Time, as the local time on the logging system. Remote analysis within different time zones will show the same dates and times. Logging of local time means adjustment of PC time and daylight saving are handled.
- Key shows trace colour, name, and value at the cursor. Selected trace has a “>” before the name. Clicking on other trace details selects other traces.
- Scale shows Minimum value, Maximum value and scale label of selected trace, with intermediate scales.
- A cursor allows data to be read off the graph. Pressing Ctrl and cursor keys allows moving between logged data.
- A Zoom In and Zoom Out facility (Time and Range), plus X & Y axis scrolling, are available.
- ‘Previous’ (◀◀) and ‘Next’ (▶▶) buttons load data files for the previous and next time period.
- The ‘Refresh’ button reloads the file from disk.
- A ‘Live’ option is used to show the latest data, and keep it up to date.
- Data can be exported to CSV files or to text.
- The current screen can be printed, or print previewed, in colour.
- The traces shown using trace selection dialog as shown initially can be changed using the “Select item” facility.

Remote Data Log Viewer

Data log files are not restricted to a machine that has CX-Supervisor installed on it. CX-Supervisor is supplied with a “Data Log Viewer” application that enables all the View and Export facilities (with the exception of Live updates) to be carried out on .dlv files that have been transferred to a stand alone machine.

Data Log Export Facilities

Exporting Data via the Export Dialog

All the export facilities described in the ExportLog function can be carried out from the Export dialog, which can be invoked from the CX-Supervisor right menu, script or the Data Log Viewer:

Options include the following:

`USE GENERATED FILES`

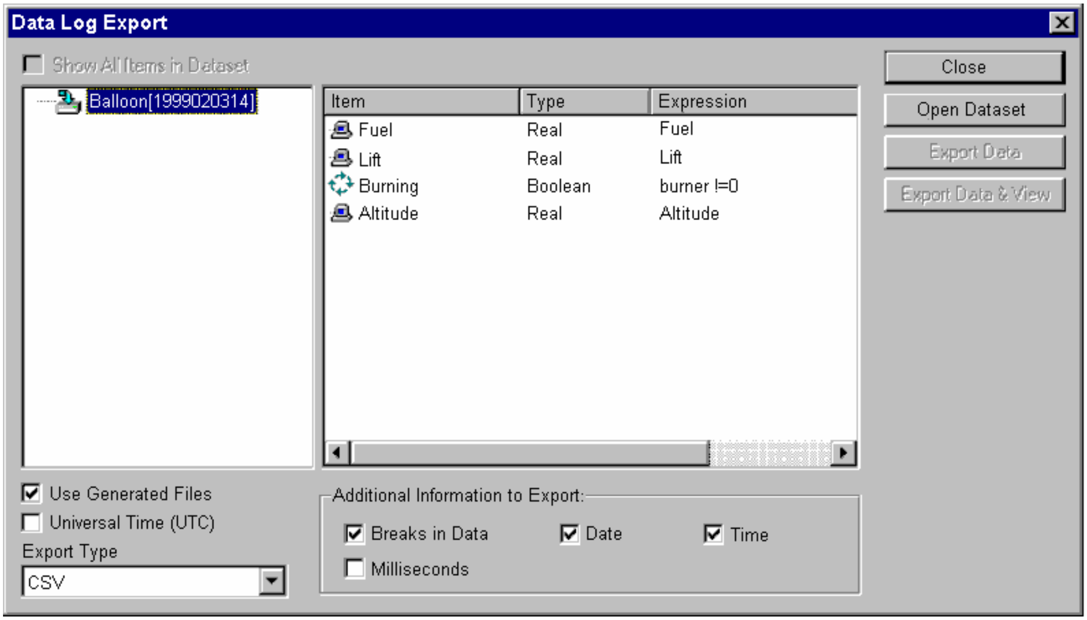
If this option is checked, then filenames will be automatically created, otherwise a "Save File As" dialog will be displayed.

`EXPORT TYPE`

Controls whether export should be as comma-separated values files or as text.

`ADDITIONAL INFORMATION TO EXPORT`

Controls whether breaks, date, time and milliseconds are exported.



□ Show All Items in Dataset

Balloon[1999020314]

Item	Type	Expression
Fuel	Real	Fuel
Lift	Real	Lift
Burning	Boolean	burner !=0
Altitude	Real	Altitude

☑ Use Generated Files

□ Universal Time (UTC)

Export Type: CSV

Additional Information to Export:

☑ Breaks in Data ☑ Date ☑ Time

□ Milliseconds

Generation of Comma Separated (CSV) Files

Both the ExportLog script function and the Export Dialog use the following conventions for automatically generating comma-separated values files for export:

All generated CSV filenames will contain at least the **<Data Set name> + <timestamp>** and a .CSV extension. CSV is a format that is recognised directly by spreadsheet packages such as Excel and can therefore be opened and viewed with this application.

```
e.g.   MyData Set1999011214.csv
```

No [ ] characters are used in the timestamp. This is for two reasons:

A. Applications like Excel seem to complain about them being there.

B. Less likely to be confused will .dlv files during searches/purges

During the generation of files then obviously duplication of names can occur. To avoid this the following convention is used:

"_nnn" will be appended to any duplicate filenames, to make them unique, where nnn = 1 to 999. (If you run out numbers the export will fail, tidying up will remedy the problem).

Generated names depend on whether a single item, or multiple items, are selected for export:

Single Selections

If a single item is selected for export i.e. "Item3" in the root directory of "MyData Set" then the following file will be generated:

```
MyData Set1999011214Item3.csv
```

Multiple Selections

On Interval Items

If a multiple selection of "on interval" items with the same interval is made then the name of the parent directory will be used e.g. if the items live in "Group1" the following file will be generated:

```
MyData Set1999011214Group1.csv
```

The file will contain a column for each item selected and they will share the Date and Time fields.

Note: The milliseconds field will only be valid for one of the Items, and should therefore be filtered out for this type of grouping.

If a multiple selection of "on interval" items, with different intervals is made then any items with the same interval will be grouped together in the same file. The same convention will be used as above except using the duplicate filenames rules described above. If an item does not share any interval with any other Item then its name will be used in the filename.

For example:

If Items "I1", "I2", "I3", "I4", "I5", "I6" are selected from Group1 of Data Set *MyData Set* and Items "I1", "I3" & "I6" share intervals and "I2" and "I5" share a different interval then the following files will be generated.

`MyData Set1999011210Group1.csv`	contains "I1", "I3" & "I6"
`MyData Set1999011210Group1_1.csv`	contains "I2" & "I5"

`MyData Set199901121OI4.csv` contains “I4”

On Change Items

All on change items will have their own filename generated regardless of any multiple selections made, because it is not possible to determine any common time interval with these type of Items.

If a multiple selection of ‘On Interval’ and ‘On Change’ Items is made then the above conventions still apply.

Generation of Text Files

Whereas CSV files only contain limited information e.g. Breaks, Date, Time, Milliseconds and Value, Text files contains all the information stored on selected Items/Groups i.e. the expression, label, deadband etc. The multiple selection rules apply as described in the previous paragraph, with the exception that On Change items can be grouped together.

Data Logging

Overview

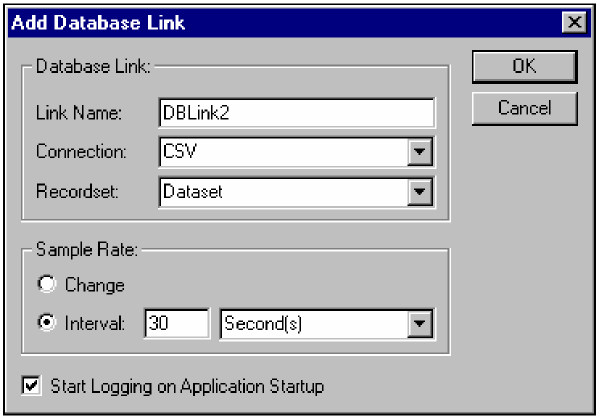
As well as the inbuilt datalogging features, it is also possible to log data directly to an existing Database. This provides direct storage of data in third party format, allowing for easy analysis using familiar tools, and there is no need for scripts to export data. However slower speed means logging to a database may be unsuitable for tasks with large data throughput or high logging speed. Also, unlike the standard Datalog files, there is no standalone viewer for instant viewing. The ADO interface used to access Data Sources does not provide any mechanism for creating Databases or Tables, therefore, it is not possible to automatically create a data source. Unpopulated data sources for use in Database Logging, must first be created using the specific software for your choice of data source e.g. "MS-Access".

To configure CX-Supervisor to log to a database:

- Create an ‘unpopulated’ data source or ‘template’ for use in Database logging.
- Create a Database connection in the Workspace Database editor to the database created above. Add a recordset with a read/write lock, and fields with 'Field Property' configured as 'Add' so records can be added.
- Create a Database Link in the Workspace Logging editor, by selecting 'Add Db Link...' from the popup menu to show the Add Database Link dialog.
- Create Field Links for each field to log by selecting 'Add Db Field...' from the popup menu to show the Add Field Link dialog.

Add Database Link Dialog

The Add Database Link dialog show below is show when the 'Add Db Link...' menu option is selected from the Data Log editor, and the identical Modify Database Link dialog when the 'Edit' menu is selected.



Add Database Link

Database Link:

Link Name: DBLink2

Connection: CSV

Recordset: Dataset

Sample Rate:

○ Change

● Interval: 30 Second(s)

☑ Start Logging on Application Startup

Link Name

For convenience, a unique Database Link name is created automatically. This can be changed to give a more meaningful description, if required.

Connection

Select the Database Connection to link to from the list showing configured Database connections.

Recordset

Select the Recordset to create a link to from the list showing Recordsets configured in the selected Connection.

Sample Rate

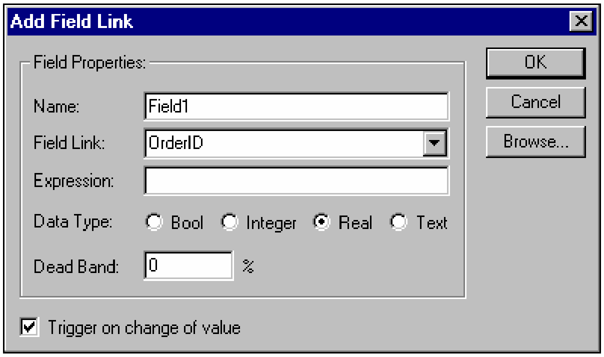
Select whether field links defined within this Database Link are logged when their expression changes or on a regular interval, for example every 30 seconds.

Start Logging on Application Startup

When unchecked, logging must be started and stopped using script commands. When checked, the logging of all associated fields starts automatically when the application is started.

Add Field Link Dialog

The Add Field Link dialog show below is show when the 'Add Field Link...' menu option is selected from the Data Log editor, and the identical Modify Field Link dialog when the 'Edit' menu is selected.



Field Properties:

Name: Field1

Field Link: OrderID

Expression:

Data Type: ○ Bool ○ Integer ● Real ○ Text

Dead Band: 0 %

☑ Trigger on change of value

Name

For convenience, a unique Field Link name is created automatically. This can be changed to give a more meaningful description, if required.

Field Link

Select the Field to link to from the list showing fields configured in the chosen Recordset.

Expression

Enter the point name or expression that will be logged. The Browse button allows easy point selection.

Data Type

Set the type of the data to be stored. For a point, this is automatically set to the same as the point type when selected using the Browse button. However, for an expression this must be specified, for example, an expression $Second / 3 will create a Real value, but the required data may be just the integer part, so the Data Type would be set to Integer.

Deadband

This allows the value of the expression to change within the percentage limit without being logged. This is particularly useful when logging analogue values that are subject to noise. Each sample is checked (when the value changes for ‘On Change’ sampling, or every log period for ‘On Interval’) against the last logged value. While the value varies within the deadband, the values are ignored. When the value exceeds the deadband limits the new value will be logged. This may dramatically reduce file size, and ease analysis.

Entering a value of 0%, i.e. no dead band, will cause every sample to be logged.

Trigger on change of value

This option is only used when the Sample Rate is set to 'On Change'. When unchecked, every time the expression changes a new record is written. If a record consists of many fields that may be read or calculated at different times, a new record is written every time any one field changes. However, the desired action may be to change several fields, and then log the new values to a single record. To achieve this action, the 'Trigger on change of value' should be checked for all fields within a record that may be read or calculated at different times. Only when all checked fields have been changed is the record logged. If however, while waiting for remaining fields to change before writing, a field is changed for a second time, the record is written with all current field values including unchanged fields, before the field value is changed for a second time.

Note: There is a risk when using fields with 'Trigger on change of value' checked that the new value of a field may not actually change its value i.e. the new value happens to be the same as the old value. Because the field has not changed, the record is not written until the value is changed again. This can produce unexpected results as fields with 'Trigger on change of value' unchecked are written with their current values at the time of writing, not the value at the time of initial change. This may also occur when a Deadband is used.

CHAPTER 12
Databases

Overview

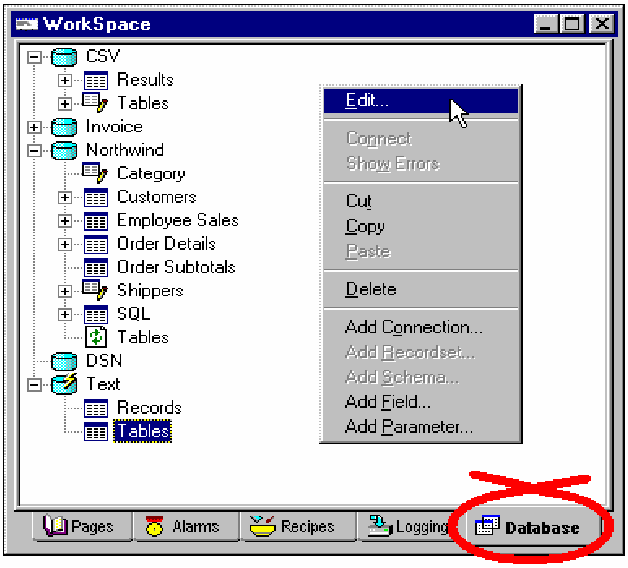
CX-Supervisor Database facilities provide fast, transparent access to many different data sources, via a database technology called ADO. Database design is not covered here, as it is a large subject explained fully in any of the many books available. It is assumed that developers using these database facilities have a working knowledge of databases. Comprehensive Database Script Functions allow complete database access.

The data sources which can be accessed depends on the Data Providers installed, but may include:

- MS-Access
- MS-Excel
- Visual FoxPro
- dBase
- ODBC for Oracle
- Paradox
- SQL Server
- Text files (in TXT and CSV format)

Database Connection Editor

The Database connection editor in the Development Workspace, enables users to create Connections, Recordsets, Field association, Parameter association and Schema objects in a familiar Tree View (hierarchical) format, using the popup menu.



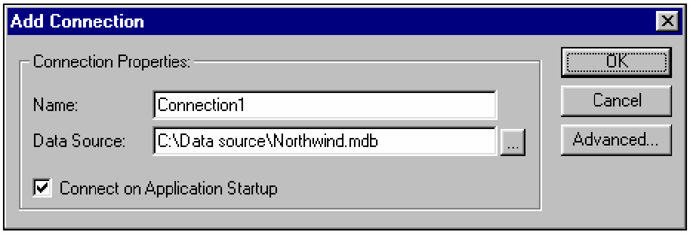
This editor is unique in CX-Supervisor, in that actual database connections can be tested online in the Development Environment. The ability to connect online also has the added benefit of providing assistance in creating objects lower down in the hierarchy. This editor has been designed to enable a large proportion of the database functionality, to be performed automatically (i.e. without the need for Script functions), although a comprehensive set of Database Script functions are available.

Configuring a Connection

Connections to existing databases are added to the Workspace by using a right mouse-button context sensitive menu option ‘Add Connection...’ which invokes the Add/Modify Database connection dialog. They can be modified later by selecting the 'Edit...' option from the menu.

Add/Modify Database connection dialog

The Add Connection dialog show below is show when the 'Add Connection...' menu option is selected from the Database connection editor, and the identical Modify Connection dialog when the 'Edit' menu is selected.


Add Connection

Connection Properties:

Name: Connection1

Data Source: C:\Data source\Northwind.mdb

☑ Connect on Application Startup

Name

For convenience, a unique Connection name is created automatically. This can be changed to give a more meaningful description of the connection, if required.

Data Source

Enter the database file to be used as a Data Source, or locate using the Browse button. The following data source file types are supported:

- MS-Access Files (*.mdb)
- MS-Excel Files (*.xls)
- Text Files (*.txt, *.csv)
- FoxPro Files (*.dbf)
- Data Source Names (*.dsn)

A Data Source Name file stores information about a database connection in a file. See Creating a Data Source Name file for more information.

Note: Connecting to CSV or Text files is slightly different from an actual Database connection. Only the ‘Directory’ that contains the required files should be supplied as a Data Source - if a file is selected, the connection will fail. The actual file to be used is specified later when configuring the Recordset. For example, if a collection of text or csv files are contained in the directory C:\Text then a valid connection ‘Data Source’ is "C:\Text\".

Note: Connections to CSV or Text files using the Provider installed with ADO version 2.0 are read only. Records can not be added or amended. To create a read/write connection to a CSV or Text file see chapter 12, Creating a Read/Write connection to CSV/Text file.

Note: Connections to Excel files using the Provider installed with ADO version 2.0 are read only. Records can not be added or amended. To create a read/write connection to an Excel file see chapter 12, Creating a Read/Write connection to an Excel file.

Connect on Application Start-up

The checkbox ‘Connect on Application Start-up’ provides the option of automatically connecting to the Database when the Runtime application is started.

Advanced

Shows the Connection String dialog, allowing the automatically generated connection string to be manually edited.

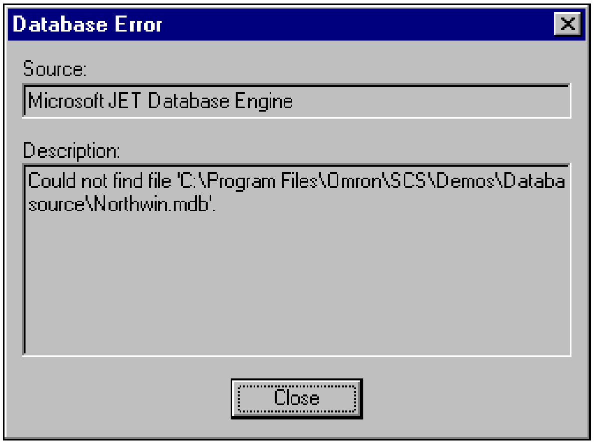
Testing Connections in the Development Environment

A connection to a Database can be made in the Development Environment by selecting the required Connection in the Tree View and then selecting the right-menu option ‘Connect’. If a Connection contains Recordsets that are set to auto open, these will also be opened by the ‘Connect’ option. If all goes well and a valid connection is made, the Database Connection Icon will be adorned with a ‘lightning bolt’. If not, then this is probably due to an error in the ‘Connection String’.

Database Errors

A detailed description of what type of error occurred (supplied by the underlying Data Provider) can be viewed, by clicking that the right-menu option ‘Show Last Error’. Whenever an error is generated by a Data Provider a description of the error and its source will be displayed in a Dialog. The ‘Show Last Error’ option is specific to each Connection.

Example: The following error was generated by the ‘Jet Database Engine’ (due to a typo in the Database name):

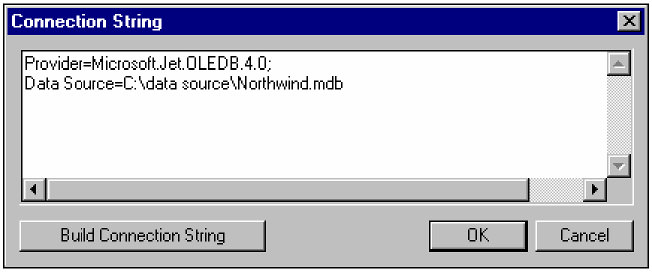


Source:
Microsoft JET Database Engine

Description:
Could not find file 'C:\Program Files\Omron\SCS\Demos\Databa source\Northwin.mdb'.

Database Connection String dialog

Connection to a Database is performed by means of a ‘Connection String’. Because different Data Providers require different information to connect you to a data store, these strings can be quite complex and cumbersome. For this reason CX-Supervisor will automatically create a valid connection string for your selected data source (if it’s supported). This string can be viewed and modified via the Connection String dialog shown below, when ‘Advanced’ is pressed on the Add/Modify Database Connection dialog.

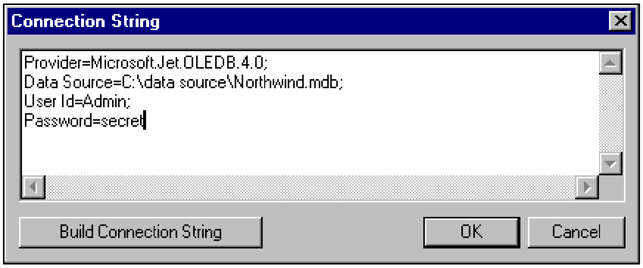


```
Provider=Microsoft.Jet.OLEDB.4.0;
Data Source=C:\data source\Northwind.mdb
```

If your data source is not supported, or you have your own drivers for a particular database, the ‘Connection String’ can be modified using this dialog (consult your database documentation for the required connection string).

User Id and Passwords

If a connection to a database requires a user id or password, this can be supplied by means of the connection string, which can be modified via the Advanced Dialog as follows:



```
Provider=Microsoft.Jet.OLEDB.4.0;
Data Source=C:\data source\Northwind.mdb;
User Id=Admin;
Password=secret
```

If you make a mistake while editing the ‘connection string’, the original string can be restored by selecting the ‘Build Connection String’ button. A new connection string will also be built automatically each time a change of Data Source is made.

Example Connection Strings

Listed below are some example connections strings for the listed providers:

Jet	"Provider=Microsoft.Jet.OLEDB.3.5.1; Data Source=c:\dbname.mdb"
SQL Server	"Provider=SQLOLEDB; Data Source=server_name; Initial Catalog=dbname; User Id=user_id; Password=user_password"
Index Server	"Provider=MSIDXS; Data Source=catalog_name"
DSN	"DSN=data_source_name"
FILEDSN	"FILEDSN=filename.dsn"
MSDASQL	"Driver={Microsoft Excel Driver (*.xls)}; DBQ=c:\Database\Invdb.xls"

Data providers installed with ADO V2.0

Type	Provider Name	Description
Jet 3.51	Microsoft.Jet.OLEDB.3.51	For Microsoft Access databases
Directory Services	ADSDSOObject	For resource data stored, such as Active Directory, this will become more important when NT5.0 is available.
Index Server	MSIDXS	For Microsoft Index Server.
ODBC Drivers	MSDASQL	For existing ODBC Drivers, this ensures that legacy data is not omitted.
Oracle	MSDAORA	Native Oracle driver simplifies access to existing Oracle data stores.
SQL Server	SQLOLEDB	For Microsoft SQL Server.
Data Shape	MSDataShape	For hierarchical recordsets, this allows the creation of master/detail type recordsets, which allow drilling down into detailed data.
Persisted Records	MSPersist	For locally saved recordsets.
Simple Provider	MSDAOSP	For creating your own providers for simple text data.

The above is just the list of standard providers supplied by Microsoft. Other vendors are actively creating their own.

Creating a Data Source Name file

A Data Source Name file (or DSN for short) stores information about a database connection in a file. The file has the extension .DSN and by default is stored in the "$\Program Files\Common Files\ODBC\Data Sources" directory. This type of file can be viewed with a suitable text editor e.g. "Notepad". One advantage of using a DSN file over specifying the full path of the database is that the DSN file remains unchanged while its contents can be re-configured to reflect any changes in directory or database file name etc.

Creating a New DSN

- For Windows 98, ME and NT: From your Windows ‘Control Panel’, select the ODBC Data Sources icon. This will show the ODBC Data Source Administrator dialog box. For Windows 2000 and XP: From the ‘Programs’ folder, select ‘Administrative tools’ and ‘Data Sources (ODBC) icon (note this is only available if the logged in user has administrative rights, and the Taskbar properties “Display Administrative Tools’ option is checked). This will show the ODBC Data Source Administrator dialog box.
- Click on the 'File DSN' tab. Any Data Source Names already defined will be listed.

- Click on 'Add' to create a new Data Source Name file. This will invoke the Create New Data Source dialog box with a list of available drivers (only drivers that are installed on your machine will be shown).
- Choose the driver to access the data source and select ‘Next >’.
- You will then be prompted to name your Data Source. Type a suitable name and select ‘Next >’
- Verify the information shown and select 'Finish' to complete this part of the operation.
- Depending on the driver selected, you may be prompted for details of the database you wish to connect to.
- A new DSN file will now exist which can be used by CX-Supervisor to Create a Connection.

Creating a Read/Write connection to an Excel file

Connections to Excel files may result in the error “Operation must use an updateable query”. This is because the Provider installed with ADO version 2.0 is read only by default hence Records can not be added or amended. To override this function you must manually change the Read Only property in the Connection String. On the “Modify Connection” dialog click “Advanced…” and set the connection property as shown in the last line below:

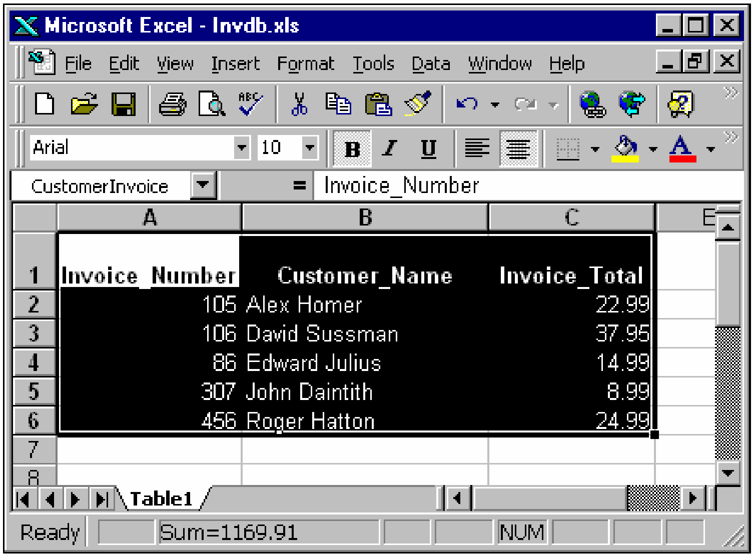
```
Driver={Microsoft Excel Driver (*.xls)};
DBQ=C:\WorkBook.xls;
ReadOnly=False;
```

Excel files can also be updated by accessing the file via the ODBC DSN driver. This is achieved by carrying out the following steps:

- Create a File DSN for the required Excel file with the following options (see Chapter 12, Creating a Data Source Name file)
- Select the Microsoft Excel Driver (*.xls). If this option does not exist, you will need to install the Microsoft ODBC driver for Excel from the Excel setup.
- Ensure that the "Read Only" check box is clear.
- Load the Excel spreadsheet and create a table to access the data by creating a Named Range as follows:
- Highlight the row(s) and column(s) area where your data resides (including the header row).
- On the ‘Insert’ menu, point to ‘Name’, click ‘Define’ and enter a name for your range.
- Create a connection in the CX-Supervisor Workspace specifying the File DSN as its source.

- Add a Recordset to the connection and select the Named Range (which will appear in the list of available tables, if the connection is live) as the Table name. Records in this table can now be added or modified as with any other database table (Note: If records are added to this type of table the Named Range will increase in size accordingly automatically).

The example below demonstrates a valid range selection named: "CustomerInvoice" (note the name displayed just above column A):


Microsoft Excel - Invdb.xls
File Edit View Insert Format Tools Data Window Help
Arial
10
Table1
Ready
NUM

CustomerInvoice = Invoice_Number

	A	B	C
1	Invoice_Number	Customer_Name	Invoice_Total
2	105	Alex Homer	22.99
3	106	David Sussman	37.95
4	86	Edward Julius	14.99
5	307	John Daintith	8.99
6	456	Roger Hatton	24.99
7			
8			

Sum=1169.91

Note: The first row of the range is assumed to contain the Column Headings.

Note: When updating files the column headings cannot contain numbers or spaces, for example "Column1" or "Invoice Total" is invalid. Also some words are reserved and can not be used, for example a column heading of "Number" will cause an error. These restrictions do not apply when only reading these files.

Note: Make sure that all the cells in a column are of the same data type. The Excel ODBC driver cannot correctly interpret which data type the column should be if a column is not of the same type, or you have types mixed between "text" and "general".

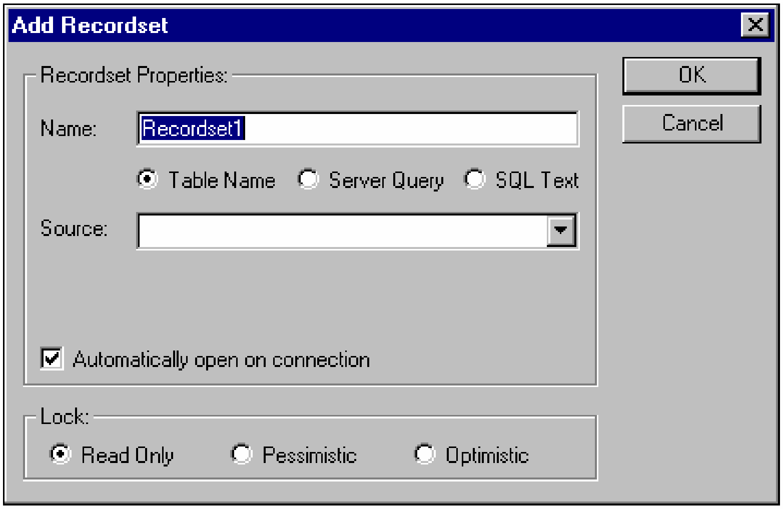
Note: This type of querying and updating information in an Excel Spreadsheet does not support multi-user concurrent access.

Creating a Read/Write connection to CSV/Text file

Connections to CSV or Text files using the Provider installed with ADO version 2.0, like Excel files, are also read only hence Records can not be added or amended. However, CSV and Text files can be updated by converting the data into an Excel spreadsheet and accessing the file via the ODBC DSN driver. This is achieved by carrying out the steps as above using the CSV/Text file instead. Note the Excel provider is still used, and a named range in Excel must still be created.

Configuring Recordsets

The Recordset is the heart of the Database facility, it contains all of the columns and rows returned from a specific action. The Recordset is used to navigate a collection of records, and update, add, delete or modify records. Once a Connection has been added to the Workspace, the right menu option 'Add Recordset...' will be enabled. Selecting this option will invoke the following dialog:



Add Recordset

Recordset Properties:

Name: Recordset1

● Table Name ○ Server Query ○ SQL Text

Source:

☑ Automatically open on connection

Lock:

● Read Only ○ Pessimistic ○ Optimistic

Name

A unique Recordset name will be automatically provided. This can be modified to provide a more meaningful name if required.

Recordset Type

The Recordset can be 1 of 3 types:

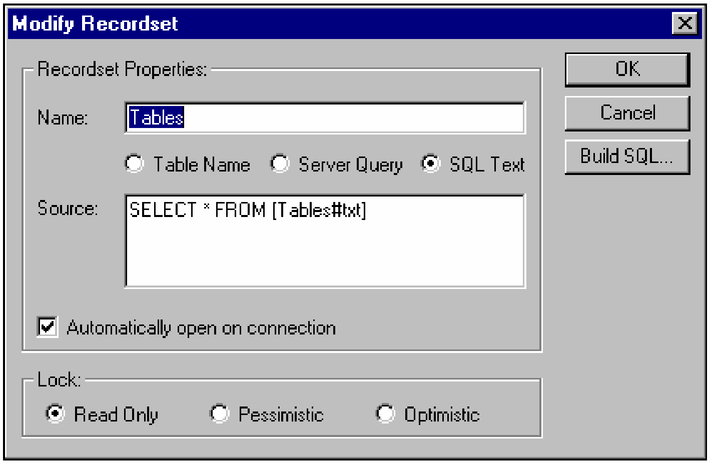
Table Name The Recordset is the name of an actual table in the Database.

Server Query The Recordset is the results of a pre-defined Server Query stored in the database.

SQL Text The Recordset is the results of an SQL query executed when the Recordset is opened.

Note: It is more efficient to run a Server Query than an SQL query.

Note: For Database connections all three of the above options are available, but for Text or CSV connections only one option is available, namely ‘SQL Text’. For convenience, a facility is provided for automatically building the required SQL Text for this type of connection. This facility is invoked from the ‘Build SQL...’ button shown below:



Name: Tables

○ Table Name ○ Server Query ● SQL Text

Source: SELECT * FROM [Tables#txt]

☑ Automatically open on connection

Lock: ● Read Only ○ Pessimistic ○ Optimistic

This will bring up a dialog with a list of all valid files in the ‘Directory’ specified for the Connection. After choosing a file and exiting from the ‘Build SQL’ dialog the required SQL Text is built. In the above example, the file ‘Tables.txt’ was chosen, but this will be written as Tables#txt in the SQL Text as most Providers will not accept the ‘.’ character, because it is used as a delimiter.

Source

The source field shows the Table name, Server Query or SQL text as selected above which the Recordset is linked to.

Automatically open on connection

If this checkbox is ticked the Recordset will be automatically opened when the Connection is opened. If this is unchecked, the Recordset must be open with a script command.

Lock

The lock option enables the Recordset to be opened in either read only or read/write modes, there are two type of read/write locks as defined below:

Read Only The default lock is read only i.e. data cannot be changed.

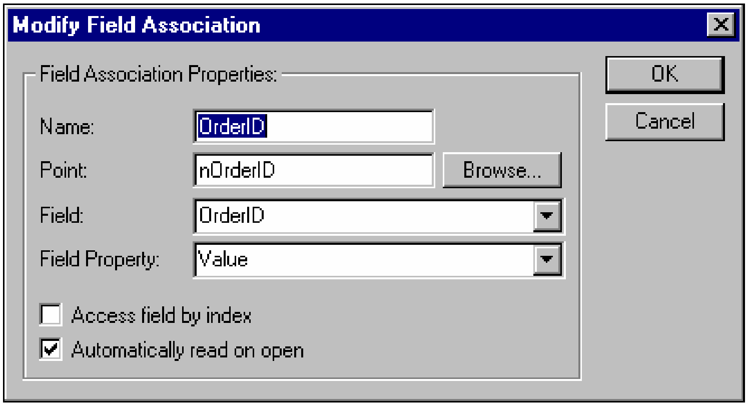
Pessimistic Locks records when you start editing and releases the lock when Update() (or Cancel()) is called, no need to worry about a conflict with other users, but can cause records to be locked for long periods of time preventing other users from accessing the same records.

Optimistic Locked only when the Update() method is called, therefore changes can be made to records without creating a lock, conflicts have to be catered for because someone else might have changed the record between the time you started editing and the time you called Update().

Note: If the Connection is open when a Recordset is added the Combo boxes for 'Table Name' and 'Server Query' will be automatically populated with valid entries for the selected Database. When the 'Add Recordset...' dialog is closed an attempt will be made to open the newly configured Recordset.

Configuring Field Associations

Field associations provide a means of connecting CX-Supervisor Points with fields (i.e. columns of data) in a Recordset, thus enabling data transfers to be made between Points and Records. By creating a Field Association for each field in a record, data can easily be read from a record in the database to its associated points, and written from the points to the current record in the database. Once a Recordset has been added to a Connection in the Workspace, the right menu option 'Add Field...' will be enabled. Selecting this option will invoke the following dialog:



Modify Field Association

Field Association Properties:

Name: OrderID

Point: nOrderID

Field: OrderID

Field Property: Value

☐ Access field by index

☑ Automatically read on open

Name

A unique Field name will be automatically provided. This can be modified to provide a more meaningful name if required.

Point

The name of the point that will be used in data transfers. The Browse button may be used to select a current point, or add a new one.

Field

The name of the Recordset field to be associated with the above point. If the Recordset is open, this list will automatically show all available fields.

Field Property

The type of information from the field to be transferred, the following options are available:

Value	default - the assigned value of the field
Name	the name of the field / column title
Type	the fields Data Type
Size	the maximum width of the field
Add	used to add new fields to a record

Note: The Name, Type and Size properties are fixed for all entries of the column, whereas the field value depends on the current position of the Recordset.

Note: The 'Add' property is specifically designed to enable fields to be added together to create new records. They are not involved in any read operations, as with the other field property types. For this reason, the 'Automatically read on open' checkbox is disabled when this type is applied. When creating configurations to add new records you will need to create a 'Add' association for every field required to 'create' a valid record i.e. primary keys, non-null values etc. need to be catered for. See DBAddNew() for more details.

Access field by index

When checked, a numeric index is used to identify a particular field instead of its name. This is useful if you want to configure generic field associations.

Automatically read on open

When checked, the data is transferred from the Recordset field to the associated point, when the Recordset is opened.

Field Paging

You can work with a single record at a time by associating points with the required fields, and process the data one record at a time. However, if the point specified is an Array point the whole array will be processed i.e. multiple fields will be read, written or added. This concept is called Paging. Paging is supported by the Database script functions, enabling you to manipulate and navigate the database a page at a time. CX-Supervisor determines the page size, by using the number of elements in the Array point used in Field Associations, i.e. if an array point with 10 elements is used then a page size of 10 will be used. In order for paging to work sensibly, you should ensure that all array points used in multiple field associations for a particular Recordset are of the same size. If arrays, of differing length are used, the smallest array size will be adopted as the page size.

Note: Paging only operates on Field Associations that have the Property Type ‘Value’ selected, this enables you to have Field Associations with a Property Type of ‘Name’ or ‘Add’ associated with single points in the same Recordset, without effecting the page size determined by the array points.

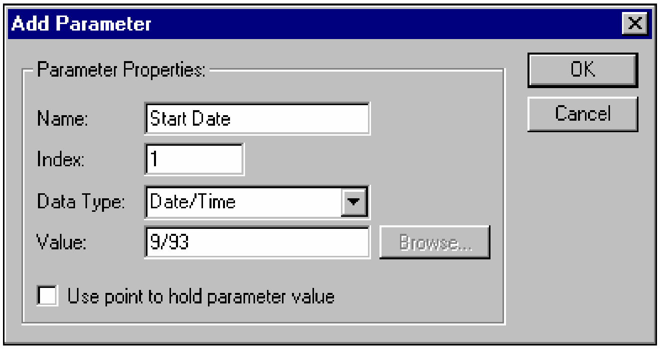
Note: Paging is designed to operate at the Recordset level (the concept of levels is explained in the chapter on DB Script functions). If you perform a Read operation on a recordset that has paging in force, then a ‘page’ of records will be read into all the Field Associations connected to the Recordset. In contrast to performing a read operation at the Field level which will override the page size and use the individual fields length.

Configuring Parameter Associations

When a Recordset which is defined as a Server Query is opened, the query is executed, and the set of records produced. The Server Query may be defined in the database as requiring parameters to be passed which allow criteria to be passed to the query. Recordsets defined in CX-Supervisor as Server Queries may have Parameter Associations added to the recordset. Recordsets defined as a Table Name or SQL Text do not use parameters, hence any Parameter Associations are ignored.

Parameter associations provide a means of supplying values to parameters whenever a Server Query is run. Each required parameter defined within the query is associated with a point or constant value. At the time the Query is run, the current value of the point (or the constant value) is passed to the query. Under the Development Environment the points default value is used.

Parameter Associations are added by right-clicking the Recordset and selecting 'Add Parameter...' from the popup menu. The following dialog is shown:



Add Parameter

Parameter Properties:

Name: Start Date

Index: 1

Data Type: Date/Time

Value: 9/93

□ Use point to hold parameter value

Name

A unique Parameter Association name will be automatically provided. This can be modified to provide a more meaningful name if required. By convention, this is made the same as the name of the parameter defined within the database but this is not essential.

Index

The index is used to determine which parameter in the Query to associate the value with. The index is automatically incremented for each parameter that is added to the Recordset.

Data Type

The Data Type list will be populated with a selection of available data types. The correct data type for the parameter being configured must be selected, otherwise the Recordset will fail to open.

Use point to hold parameter value

When unchecked, the constant Value specified is passed as the parameter to the query. When checked, the current value of the point is passed as the parameter.

Point

The name of the point to be used. The Browse button may be used to select a current point, or add a new one.

Value

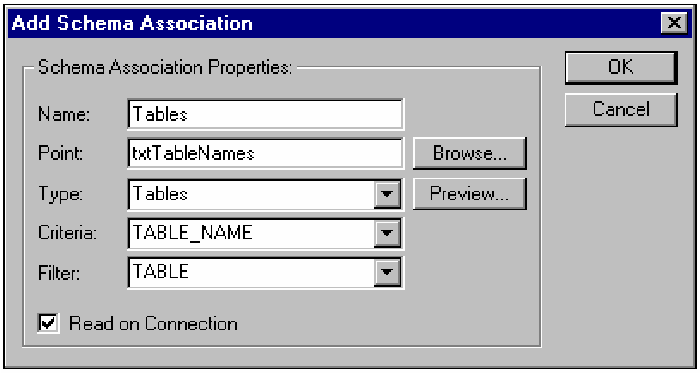
The constant value to be used.

Configuring Schemas

Schemas enable information about a Database to be obtained from a Provider. There are a large number of Schema Types available. The most useful feature of schemas is the ability to obtain Table and Query names from the Database, in fact schemas are used by the Development Environment to populate the Combo boxes when working with 'live' connections.

The information to be returned from the Provider is determined by specifying a Schema Type and a Criteria. The Criteria argument is an array of values that can be used to limit the results of a schema query. Each Schema type has a different set of Criteria that it supports.

A Schema is configured by selecting the desired Connection and choosing the right menu option 'Add Schema...' to invoke the following dialog:



Add Schema Association

Schema Association Properties:

Name: Tables

Point: txtTableNames

Type: Tables

Criteria: TABLE_NAME

Filter: TABLE

☑ Read on Connection

Name

A unique Schema name will be automatically provided. This can be modified to provide a more meaningful name if required.

Point

The name of an array point which will hold the results of the schema request. The Browse button may be used to select a current point, or add a new one.

Type

The Type list will be populated with a selection of available Schema Types.

Criteria

The Criteria list shows the available Criteria for the given Schema Type.

Filter

The Filter list is used with certain Schema types to reduce the information returned.

Read on Connection

If checked, the Schema results are automatically obtained when successful connection to the database is achieved.

Preview

If the Connection is live, then the Preview button will be enabled on the dialog, which allows you to view the results of the configured Schema.

The Schema 'Type', 'Criteria' and 'Filter' values can be modified at Runtime via the DBSchema() function.

Database Schema Types

The Database Schema types supported in ADO are:

Schema Type values	Criteria Values
Schema Asserts	CONSTRAINT_CATALOG CONSTRAINT_SCHEMA CONSTRAINT_NAME
Schema Catalogs	CATALOG_NAME
Schema Character Sets	CHARACTER_SET_CATALOG CHARACTER_SET_SCHEMA CHARACTER_SET_NAME

Schema Type values	Criteria Values
Schema Check Constraints	CONSTRAINT_CATALOG CONSTRAINT_SCHEMA CONSTRAINT_NAME
Schema Collations	COLLATION_CATALOG COLLATION_SCHEMA COLLATION_NAME
Schema Column Domain Usage	DOMAIN_CATALOG DOMAIN_SCHEMA DOMAIN_NAME COLUMN_NAME
Schema Column Privileges	TABLE_CATALOG TABLE_SCHEMA TABLE_NAME COLUMN_NAME GRANTOR

	GRANTEE
Schema Columns	TABLE_CATALOG TABLE_SCHEMA TABLE_NAME COLUMN_NAME
Schema Constraint Column Usage	TABLE_CATALOG TABLE_SCHEMA TABLE_NAME COLUMN_NAME
Schema Constraint Table Usage	TABLE_CATALOG TABLE_SCHEMA TABLE_NAME
Schema Foreign Keys	PK_TABLE_CATALOG PK_TABLE_SCHEMA PK_TABLE_NAMEFK_TABLE_CATALOG FK_TABLE_SCHEMA FK_TABLE_NAME
Schema Indexes	TABLE_CATALOG TABLE_SCHEMA INDEX_NAME TYPE TABLE_NAME
Schema Key Column Usage	CONSTRAINT_CATALOG CONSTRAINT_SCHEMA CONSTRAINT_NAME TABLE_CATALOG TABLE_SCHEMA TABLE_NAME COLUMN_NAME

Schema Type values	Criteria Values
Schema Primary Keys	PK_TABLE_CATALOG PK_TABLE_SCHEMA PK_TABLE_NAME
Schema Procedure Columns	PROCEDURE_CATALOG PROCEDURE_SCHEMA PROCEDURE_NAME COLUMN_NAME
Schema Procedure Parameters	PROCEDURE_CATALOG PROCEDURE_SCHEMA PROCEDURE_NAME PARAMTER_NAME
Schema Procedures	PROCEDURE_CATALOG PROCEDURE_SCHEMA

	PROCEDURE_NAME PROCEDURE_TYPE
Schema Provider Specific	See Remarks
Schema Provider Types	DATA_TYPE BEST_MATCH
Schema Referential Constraints	CONSTRAINT_CATALOG CONSTRAINT_SCHEMA CONSTRAINT_NAME
Schema Schemata	CATALOG_NAME SCHEMA_NAME SCHEMA_OWNER
Schema SQL Languages	<none>
Schema Table Constraints	CONSTRAINT_CATALOG CONSTRAINT_SCHEMA CONSTRAINT_NAME TABLE_CATALOG TABLE_SCHEMA TABLE_NAME CONSTRAINT_TYPE
Schema Table Privileges	TABLE_CATALOG TABLE_SCHEMA TABLE_NAME GRANTOR GRANTEE
Schema Tables	TABLE_CATALOG TABLE_SCHEMA TABLE_NAME TABLE_TYPE

Schema Type values	Criteria Values
Schema Translations	TRANSLATION_CATALOG TRANSLATION_SCHEMA TRANSLATION_NAME
Schema Usage Privileges	OBJECT_CATALOG OBJECT_SCHEMA OBJECT_NAME OBJECT_TYPE GRANTOR GRANTEE
Schema View Column Usage	VIEW_CATALOG VIEW_SCHEMA VIEW_NAME
Schema View Table Usage	VIEW_CATALOG VIEW_SCHEMA

	VIEW_NAME
Schema Views	TABLE_CATALOG TABLE_SCHEMA TABLE_NAME

Note: The actual schemas are defined by the OLE DB specification and Providers are not required to support all of the OLE DB standard schema queries. Specifically, only 'Schema Tables', 'Schema Columns', and 'Schema Provider Types' are required by the OLE DB specification. However, the provider is not required to support the Criteria constraints listed above for those schema queries.

Using Transactions

Modifications of data in the database i.e. adding, modifying and deleting records, can be grouped so that all modifications take place at the same time. This group of modifications is called a Transaction. A Transaction includes any modifications to data in a connection, regardless of the Recordset. A transaction is started by calling DBExecute with the 'BeginTrans' command. From that point, any request to add, modify or delete records are stored instead of being immediately processed. The Transaction is completed either by calling DBExecute with the 'CommitTrans' command, which processes all the stored requests in one go, or by calling DBExecute with the 'RollbackTrans' command, which discards all the stored requests leaving the database as it was when the Transaction started.

Nested Transactions

Transactions may be nested, that is a new transaction may be started before the preceding transaction has been completed. In this case, any 'CommitTrans' or 'RollbackTrans' commands relate to the most recently started transaction, and any further 'CommitTrans or 'RollbackTrans' commands relate to the transaction begun before it.

Note: Care should be taken to ensure that each 'BeginTrans' is matched with a 'CommitTrans' or 'RollbackTrans' to ensure that your work is saved or discarded as required. If there are any pending transactions when a connection is closed, the user will be prompted to either commit or rollback these outstanding transactions. A DBExecute command 'TransCount' is available which returns the number of pending transactions.

Note: Not all Providers support use of Transactions.

Saving Recordsets as XML

Using the DBExecute() 'Save' command any Recordset may be saved as an XML file, to be imported in to another application. XML resembles and complements HTML. XML describes data, such as city name, temperature and barometric pressure, and HTML defines tags that describe how the data should be displayed, such as with a bulleted list or a table. XML, however, allows developers to define an unlimited set of tags, bringing great flexibility to authors, who can decide which data to use and determine its appropriate standard or custom tags.

Example: XML is used to describe an Employees phone list:

```
<EmployeeList>
        <Entry>
                <Employee>John Jones</Employee>
                <Phone>555-1213</Phone>
                <Type>Mobile</Type>
        </Entry>
        <Entry>
                <Employee>Sally Mae</Employee>
                <Phone>555-1217</Phone>
                <Type>Business Fax</Type>
        </Entry>
</EmployeeList>
```

You can use an application with a built in XML parser, such as Microsoft® Internet Explorer 5 to view XML documents in the browser just as you would view HTML pages.

Datashaping

The ADO SHAPE command can be used to produce hierarchical recordsets i.e. where a value within the current record is used to access another Recordset. For example, a record showing Customer Details might contain a field for Customer ID. Every time the record position is changed, a child Recordset showing Orders for that Customer ID could be recreated.

Hierarchical recordsets present an alternative to using JOIN syntax when accessing parent-child data. Hierarchical recordsets differ from a JOIN in that with a JOIN, both the parent table fields and child table fields are represented in the same recordset. With a hierarchical recordset, the recordset contains only fields from the parent table. In addition, the recordset contains an extra field that represents the related child data, which you can assign to a second recordset variable and traverse.

Hierarchical recordsets are made available via the MSDataShape provider, which is implemented by the client cursor engine.

A new clause, SHAPE, is provided to relate SELECT statements in a hierarchical fashion. The syntax is summarized below:

```
SHAPE {parent-command} [[AS] name]
APPEND ({child-command} [[AS] name] RELATE parent-field TO child-
field)
[,({child2-command} ...)]
```

Note: By default, the child recordsets in the parent recordset will be called Chapter 1, Chapter 2, etc., unless you use the optional [[AS] name] clause to name the child recordset.

Note: You can nest the SHAPE command. The {parent-command} and/or {child-command} can contain another SHAPE statement.

Note: The {parent-command} and {child-command} do not have to be SQL SELECT statements. They can use whatever syntax is supported by data provider.

Note: A child recordset will be automatically opened/closed whenever its Parent recordset is opened/closed. A child recordset is effectively a field of its parent recordset therefore whenever a new record is selected in the parent a new child recordset will be generated.

Creating a Datashape connection.

- ♦ Create DSN file specifying the required database as the Data Source
- ♦ Configure a Connection to the DSN file. In the connection string type
  Provider=MSDataShape;
  FILEDSN=<your file name>.dsn
- ♦ Configure a Recordset as SQL Text and enter the required shape command as the Source. (See Datashape Source examples).
- ♦ After successfully adding a Datashape Recordset it is now possible to add a Child Recordset to the existing Recordset by selecting the right menu option ‘Add Recordset’ which will now be enabled. If the Connection is ‘Live’ a list of valid child recordset names will be entered in the Source ComboBox
- ♦ Field associations can be added to Child recordsets in the normal manner.

Child recordsets can be accessed via Script command in a similar manner to normal recordsets

```
bResult = DBState( "DataShape.Customers.Orders.Details", "Open" )
```

Note: Child recordsets are not supported in the Database function dialog

Simple Relation Hierarchy example:

```
SHAPE  {select * from customers}
APPEND ({select * from orders} AS rsOrders
     RELATE customerid TO customerid)
```

The parent recordset contains all fields from the Customers table and a field called rsOrders. rsOrders provides a reference to the child recordset, and contains all the fields from the Orders table. The other examples use a similar notation.

Compound Relation Hierarchy example:

This sample illustrates a three-level hierarchy of customers, orders, and order details:

```
SHAPE  {SELECT * from customers}
APPEND ((SHAPE  {select * from orders}
     APPEND ({select * from [order details]} AS rsDetails
         RELATE orderid TO orderid)) AS rsOrders
     RELATE customerid TO customerid)
```

In addition to the Simple Relation Hierarchy example, the Order ID is now used to create a recordset containing the order details for the shown order.

Hierarchy with Aggregate example:

```
SHAPE  (select * from orders}
APPEND ({select od.orderid, od.UnitPrice * od.quantity as
ExtendedPrice
   from [order details] As od}
      RELATE orderid TO orderid) As rsDetails,
      SUM(ExtendedPrice) AS OrderTotal
```

This example creates a Recordset of all the orders and their details, and adds a field called ExtendedPrice to store the total order value for each record, and sums all the ExtendedPrice values which is stored in Order Total.

Group Hierarchy example:

```
SHAPE  {select customers.customerid AS cust_id, orders.*
   from customers inner join orders on customers.customerid =
   orders.customerid} AS rsOrders
COMPUTE rsOrders BY cust_id
```

Group Hierarchy with Aggregate example:

```
SHAPE
   (SHAPE {select customers.*, orders.orderid, orders.orderdate from
      customers inner join orders on customers.customerid =
      orders.customerid}
   APPEND  ({select od.orderid,
    od.unitprice * od.quantity as ExtendedPrice from [order details]
    as od} AS rsDetails
   RELATE orderid TO orderid),
    SUM(rsDetails.ExtendedPrice) AS OrderTotal) AS rsOrders
COMPUTE  rsOrders,
        SUM(rsOrders.OrderTotal) AS CustTotal,
        ANY(rsOrders.contactname) AS Contact
BY  customerid
```

Note: The inner SHAPE clause in this example is identical to the statement used in the Hierarchy with Aggregate example.

Multiple Groupings example:

```
SHAPE
    (SHAPE  {select customers.*,
        od.unitprice * od.quantity as ExtendedPrice from (customers
        inner join orders
    on customers.customerid = orders.customerid) inner join
         [order details] as od on orders.orderid = od.orderid}
             AS rsDetail
        COMPUTE ANY(rsDetail.contactname) AS Contact,
                ANY(rsDetail.region) AS Region,
                SUM(rsDetail.ExtendedPrice) AS CustTotal,
                rsDetail
        BY customerid) AS rsCustSummary
COMPUTE rsCustSummary
BY      Region
```

Grand Total example:

```
SHAPE
  (SHAPE  {select customers.*,
   od.unitprice * od.quantity as ExtendedPrice from (customers inner
   join orders on customers.customerid = orders.customerid) inner
join
         [order details] as od on orders.orderid = od.orderid}
            AS rsDetail
   COMPUTE ANY(rsDetail.contactname) AS Contact,
          SUM(rsDetail.ExtendedPrice) AS CustTotal,
                rsDetail
        BY customerid) AS rsCustSummary
COMPUTE SUM(rsCustSummary.CustTotal) As GrandTotal,
        rsCustSummary
```

Note: The missing BY clause in the outer summary. This defines the Grand Total because the parent rowset contains a single record with the grand total and a pointer to the child recordset.

Grouped Parent Related to Grouped Child example:

```
SHAPE
  (SHAPE  {select * from customers}
     APPEND ((SHAPE {select orders.*, year(orderdate) as OrderYear,
        month(orderdate) as OrderMonth from orders} AS rsOrders
            COMPUTE rsOrders
              BY customerid, OrderYear, OrderMonth)
               RELATE customerid TO customerid) AS rsOrdByMonth )
        AS rsCustomers
COMPUTE rsCustomers
BY  region
```

Chapter 13
Multilingual Features

This chapter introduces CX-Supervisor's Multilingual features to a new user, and explains how to develop in, and create applications in languages other than the default English, for example:

- How a non-English designer can run the development tools in their native language, even if they are creating an application in another language
- How non-English runtime applications can be developed
- How applications can be developed where the end users with different nationalities can switch between 2 languages (i.e. Bilingual) or more i.e. Multilingual
- How applications can be developed for export, where the developers can work in their native language and switch language before export, and visiting maintenance engineers onsite can switch back to the development language

Note: Projects created in CX-Supervisor versions 1.25 and earlier use Microsoft's Multi Byte Character Sets (known as MBCS, or sometimes DBCS for 'Double Byte Character Sets') for multilingual features. The formatting of character tables in MBCS is *not* the same as Unicode. However, old projects are automatically converted to Unicode when they are loaded, and will then be saved in Unicode. See the section ***Loading old projects*** for important information before you start.

Multilingual issues can be very complex, but like most areas CX-Supervisor provides this functionality in an easy to use way, with a few simple dialogs.

The CX-Supervisor Multilingual features can easily be divided into two clear sections: Development and Runtime. Users of each section have different needs and so these sections function completely separately. For example, it is possible to *develop* an application in any language, but this does not affect which language the user must use at *runtime* in any way.

Note: The distinction between the development program and runtime program and their respective target users is very important to understand, and will help you understand the following sections.

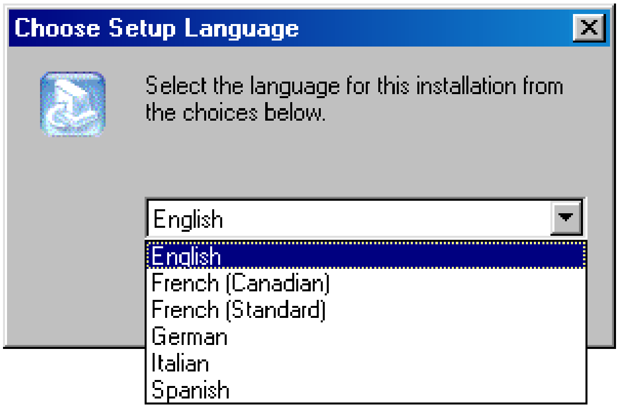
Note: CX-Supervisor is designed and tested for use with European versions of Microsoft Windows. Every language permutation is not tested and no testing is performed on other versions like Russian, Japanese or Chinese etc., as this is simply standard Microsoft functionality. However you should ensure your translated application can be correctly deployed before starting full development, for example, by creating a small test application.

Note: The formatting of character tables used by CX-Supervisor for multilingual features is the worldwide standard Unicode format. This product therefore only runs on Unicode compliant Operating Systems, i.e. Windows 2000, Windows XP and later products.

Development Features

For the benefit of the application developer the CX-Supervisor Development application may be run in any major European language i.e. English, Spanish, Italian, German and French. This means all menus, dialogs and error messages displayed by the Development program are shown in this language which provides a more comfortable and more efficient development experience.

To choose between languages, at the beginning of the installation select the required language. Note the installation program runs in your selected language, and also the Development application resources are installed in this language. In the unlikely event you ever need to change your language selection, simply uninstall CX-Supervisor in the normal manner, and then reinstall, selecting the desired language.



Choose Setup Language

Select the language for this installation from the choices below.

English

Of course the designer enters all application data, like page names, on screen text, button captions etc, and these all form part of the runtime application. Translation of the runtime application is covered in the next section.

Note: Remember the choice of development language is entirely separate to the language the end user will use and has absolutely no impact on the runtime application.

Runtime Language Features

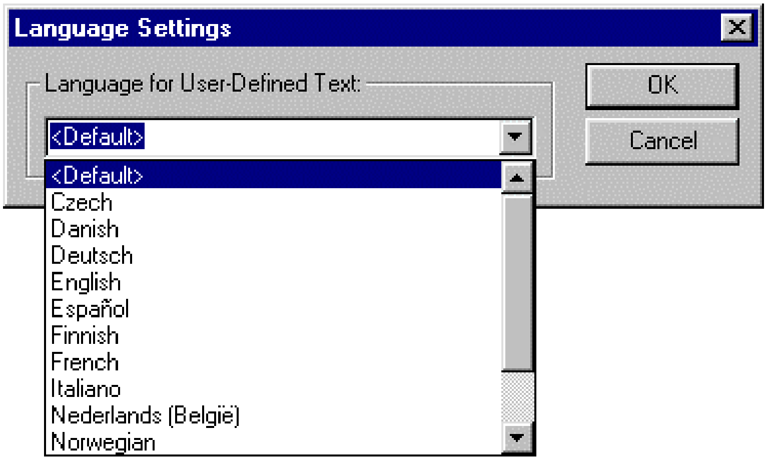
The developer will be required to create the runtime application to run in at least 1 language. This is often English but does not have to be. It could be the same language as the developer or even a completely different language for the end user. All translations for all system text, like dialogs and menus, are installed automatically. User defined text entered at development time is normally entered in the target language (see Runtime Multilingual Features if this is not suitable). The Default language can be set so that at runtime the translated system text automatically loaded and displayed.

To support the diversity of end users, the Runtime features come translated in 16 languages (including English) and can easily be extended by the developer (see Adding unsupported Runtime languages).

Note: Remember that the choice of runtime language is in no way connected to the language of the installed Developer application.

Setting the Default language

The Runtime application first starts up in the preferred default language. Set this language during development from the Language Settings dialog (select menu Project|Runtime Settings…|Language Settings):



Language Settings

Language for User-Defined Text: <Default>

After setting, when the application is next run it will start in the selected language

Note: In this context for this dialog only, "<Default>" means no language file will be loaded so the default text compiled in the executable, i.e. English, will be used. This is the initial setting, and can be useful if no Language features are being used.

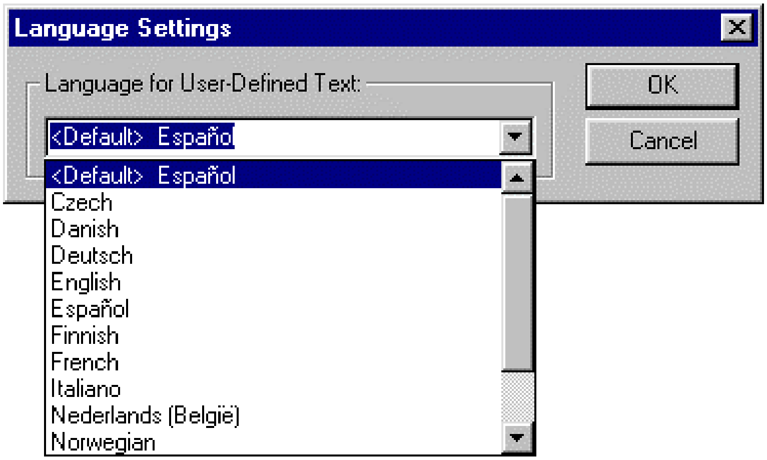
Runtime Multilingual Features

Applications supporting more than one language can be very useful, for example for general use in countries with more than one national language, or where a plant has a specific mix of personnel. This can also prove valuable for international installations when the developers and on-site maintenance engineers native language is different to the end customers. The system can be switched to a comfortable language during work, and easily switched back to the customer's requirement prior to hand over.

All translations for all system text (like dialogs, menus and errors etc) are installed automatically. User defined text entered at development time (like button captions, static text like "Value =", graph titles etc.) can be exported to a text file for easy translation into as many additional languages as required. At runtime the user can then change languages dynamically i.e. without having to exit and restart the application. The translated system text and user defined text is automatically loaded and displayed.

Changing language at Runtime

The user (if permitted by security settings) may change the language at runtime. The user may right-click the application and select the "Language Settings…" option (see Chapter 6 Projects Menu Option Access Levels for details of the right-click menu and setting access permissions).



Language Settings

Language for User-Defined Text: <Default> Español

Note: In this context, "<Default>" means the language selected as the Default startup language during development, in the above case Español (Spanish).

In addition, the developer may call the "SetLanguage" script function (see the *CX-Supervisor Script Language Reference Manual* for details), for example behind a button to allow the user to automatically switch languages.

User Defined text

While the Runtime system is shipped with full translations for all system text, any "user defined text" added by the developer must be translated and these translations incorporated into the application. Examples of user-defined text are:

- Button captions, like "Open Page"
- Text in Text objects, like "Value ="
- Default values of Text Points
- Chart and Graph titles

The system always loads user defined text from files with the file name exactly the same as the language selection, and extension ".UDT". The file names for installed languages are: English, Czech, Danish, Deutsch, Español, Finnish, French, Italiano, Nederlands (België), Norwegian, Polish, Português, Russian, Slovenija, Srpski and Swedish.

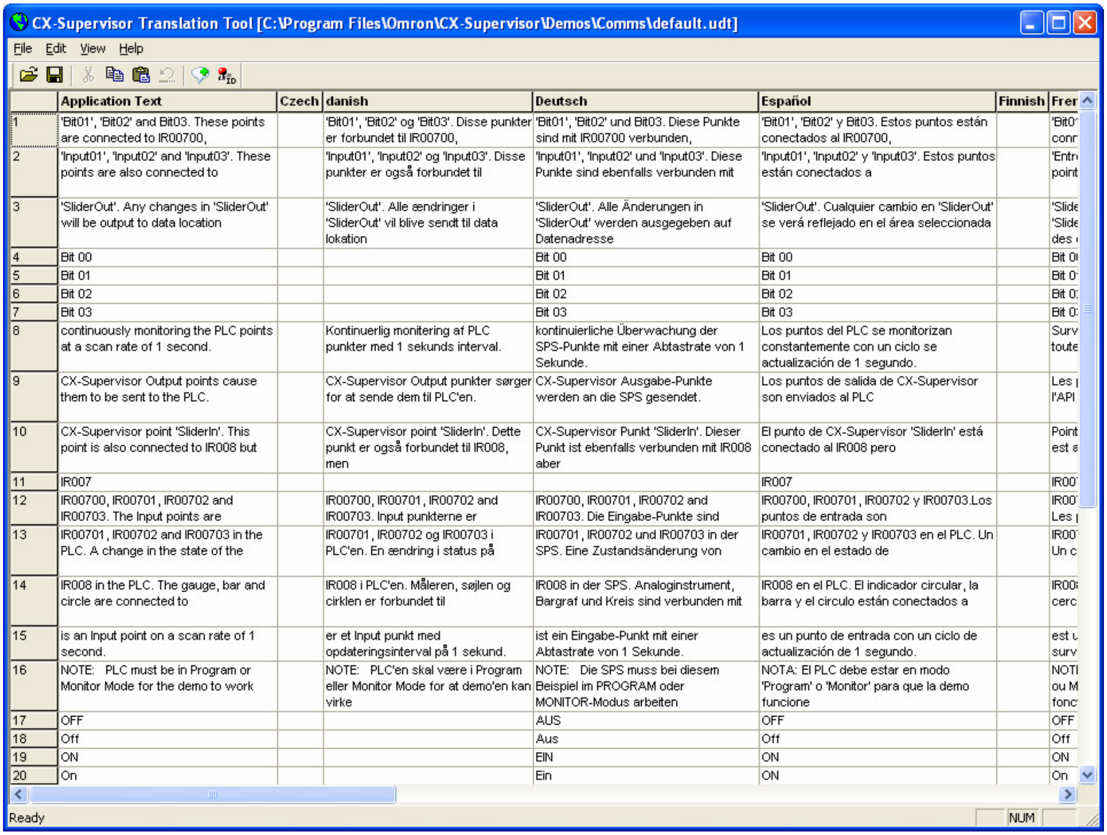
Note: User defined text is specific to each application, and therefore is located in the application project directory i.e. in the same folder as the <project>.SCS file.

Note: In CX-Supervisor version 1.25 and earlier the user defined text was stored in files with a .USL extension. These files are in MBCS format. For backwards compatibility, if these files still exist they will be loaded as MBCS and converted to Unicode temporarily for the duration of the runtime. For this conversion to succeed, the operating system must be configured to support displaying and converting the original language. To permanently convert a USL translation file to a Unicode format UDT file simply open the USL file in Notepad, select Save As and choose Encoding "Unicode" and change the extension to UDT.

Note: If both a UDT and old style USL file are present the Runtime will load and use translations from the newer UDT file.

Translating User Defined text with the Translation Tool

At any time during development, select **Translation Tool** from the Tools menu. A dialog similar to below will be shown:



CX-Supervisor Translation Tool [C:\Program Files\Omron\CX-Supervisor\Demos\Comms\default.udt]

	Application Text	Czech	danish	Deutsch	Español	Finnish	Fren
1	'Bit01', 'Bit02' and Bit03. These points are connected to IR00700,		'Bit01', 'Bit02' og 'Bit03'. Disse punkter er forbundet til IR00700,	'Bit01', 'Bit02' und Bit03. Diese Punkte sind mit IR00700 verbunden,	'Bit01', 'Bit02' y Bit03. Estos puntos están conectados al IR00700,		'Bit0 conr
2	'Input01', 'Input02' and 'Input03'. These points are also connected to		'Input01', 'Input02' og 'Input03'. Disse punkter er også forbundet til	'Input01', 'Input02' und 'Input03'. Diese Punkte sind ebenfalls verbunden mit	'Input01', 'Input02' y 'Input03'. Estos puntos están conectados a		'Entr point
3	'SliderOut'. Any changes in 'SliderOut' will be output to data location		'SliderOut'. Alle ændringer i 'SliderOut' vil blive sendt til data lokation	'SliderOut'. Alle Änderungen in 'SliderOut' werden ausgegeben auf Datenadresse	'SliderOut'. Cualquier cambio en 'SliderOut' se verá reflejado en el área seleccionada		'Slide 'Slide des
4	Bit 00			Bit 00	Bit 00		Bit 0
5	Bit 01			Bit 01	Bit 01		Bit 0
6	Bit 02			Bit 02	Bit 02		Bit 0
7	Bit 03			Bit 03	Bit 03		Bit 0
8	continuously monitoring the PLC points at a scan rate of 1 second.		Kontinuerlig monitering af PLC punkter med 1 sekunds interval.	kontinuierliche Überwachung der SPS-Punkte mit einer Abtastrate von 1 Sekunde.	Los puntos del PLC se monitorizan constantemente con un ciclo se actualización de 1 segundo.		Surv toute
9	CX-Supervisor Output points cause them to be sent to the PLC.		CX-Supervisor Output punkter sørger for at sende dem til PLC'en.	CX-Supervisor Ausgabe-Punkte werden an die SPS gesendet.	Los puntos de salida de CX-Supervisor son enviados al PLC		Les l'API
10	CX-Supervisor point 'SliderIn'. This point is also connected to IR008 but		CX-Supervisor point 'SliderIn'. Dette punkt er også forbundet til IR008, men	CX-Supervisor Punkt 'SliderIn'. Dieser Punkt ist ebenfalls verbunden mit IR008 aber	El punto de CX-Supervisor 'SliderIn' está conectado al IR008 pero		Point est a
11	IR007				IR007		IR00
12	IR00700, IR00701 , IR00702 and IR00703. The Input points are		IR00700, IR00701 , IR00702 and IR00703. Input punkterne er	IR00700, IR00701 , IR00702 and IR00703. Die Eingabe-Punkte sind	IR00700, IR00701 , IR00702 y IR00703.Los puntos de entrada son		IR00 Les
13	IR00701 , IR00702 and IR00703 in the PLC. A change in the state of the		IR00701 , IR00702 og IR00703 i PLC'en. En ændring i status på	IR00701 , IR00702 und IR00703 in der SPS. Eine Zustandsänderung von	IR00701 , IR00702 y IR00703 en el PLC. Un cambio en el estado de		IR00 Un c
14	IR008 in the PLC. The gauge, bar and circle are connected to		IR008 i PLC'en. Måleren, søjlen og cirklen er forbundet til	IR008 in der SPS. Analoginstrument, Bargraf und Kreis sind verbunden mit	IR008 en el PLC. El indicador circular, la barra y el circulo están conectados a		IR00 cerc
15	is an Input point on a scan rate of 1 second.		er et Input punkt med opdateringsinterval på 1 sekund.	ist ein Eingabe-Punkt mit einer Abtastrate von 1 Sekunde.	es un punto de entrada con un ciclo de actualización de 1 segundo.		est u surv
16	NOTE: PLC must be in Program or Monitor Mode for the demo to work		NOTE: PLC'en skal være i Program eller Monitor Mode for at demo'en kan virke	NOTE: Die SPS muss bei diesem Beispiel im PROGRAM oder MONITOR-Modus arbeiten	NOTA: El PLC debe estar en modo 'Program' o 'Monitor' para que la demo funcione		NOTI ou M fonc
17	OFF			AUS	OFF		OFF
18	Off			Aus	Off		Off
19	ON			EIN	ON		ON
20	On			Ein	ON		On

Note: The Translation Tool can also be launched manually from the Start button, and the “Default.UDT” for the required project loaded.

Note: The Translation Tool can be distributed to translators licence free, and an installation package can be found on the original CD in the folder “Translation Tool”

The User Defined Text for the application is shown in the first column, sorted alphabetically. A column is shown for each existing translated UDT file. To add new translations, click the **Add New Language** toolbar button and select the required language. Now simply select each cell and type the required translation. Non translatable text (like numbers, product names or formatting e.g. “1.0”, “Microsoft Windows” or “###.###”) can be left blank to use the original translation, or Copy and Pasted from other columns.

When completed, exit the Translation Tool and save the changes. The User Defined Text will now be loaded when you change languages at Runtime.

Note: If some development text with translations is changed or corrected the translations will be shown in Red as below, as if the Application text is missing.

Note: Where there are translations for text that is not in the Application Text, the whole row is shown in Red. To correct this problem either copy the old translations to the correct row, or just delete the unwanted translation. When all columns are corrected you can save and close the Translation Tool. When reopened the highlighted rows have been deleted. Alternatively you can delete an entire highlighted row by right clicking on the row header (with the row number) and selecting *Delete Selected Row*.

Note: Where older .USL files exist, they will be loaded if no corresponding UDT file exists, and converted to Unicode. When saving, new Unicode UDT files are always written.

Note: If you need translations for the same text in different contexts, *Show* the **Object ID** column and select the **Add new object ID string** button. Select the application text for the new translation and the object context for this special case. This is in the format PageName followed by a dot followed by ObjectName for example MyPage.Push_1

Note: Any text that includes a quote " character cannot be stored so is automatically converted to single quote ' character. Translations must also include single quote characters.

Note: To enter a ‘newline’ in the edit field hold the <Ctrl> key and press the Return key.

Translating User Defined Text manually

To get the user defined text translated and incorporated follow these steps:

1, 2, 3…

1. Develop your application completely. It is easier (and more effective) to perform these steps just once. However if further development is required, repeat these steps and use a development tool like WinGrep, Visual Sourcesafe or other version control software which should easily show the changes that have been made

2. Export the user defined text by selecting menu Project|Create User Language

File. This will create a file in the project directory called “default.udt”

3. Copy this file for as many translations as are required. Name each copy <language>.udt using exactly the same text (particularly the accents), as appears in the Language Settings dialog. If the filename prefix is not exactly the same the file will not be loaded. The format of these files is:

   “development text”,”translated text”,optional object ID

   The object ID (e.g. “Text_1”) is optional but can be supplied if specific text requires different translations for different objects. For example “Run” in English could be used as a verb on a button, and as static text for a run number. In this case different translations can be supplied depending on the object.

4. Get these files translated. An editor like Notepad can be used to enter the translated text in the second column i.e. in-between the quotes after the first comma.

5. Copy the translated files back to the application directory. The runtime should now automatically use these files.

Note: Titles of pages cannot be translated. If necessary the page title can be hidden in the Page Properties dialog, and custom title using a static text object added which does support automatic translation.

Note: Any text that includes a quote " character cannot be stored so is automatically converted to single quote ' character. Translations must also include single quote characters.

Note: Any newline characters will be converted to “\n” so that each translation always appears on one line in the file. Translations should also include the “\n” characters and never include carriage returns part way through the translation.

Configuring Windows for language support

Windows XP

Being the most modern operating system, language support provided by Windows XP is the best. If you are considering a Multilingual application this operating system offers the best solution. It is possible to install language support for multiple languages, multiple character sets and fonts, multiple keyboard layouts for input and to switch between them. To achieve this, follow these steps:

1. Start Windows in the normal way
2. Open Control Panel and select **Regional and Language options**
3. The **Regional Options** tab settings do not affect the language support, but can be set as required
4. On the Languages tab, click **Details**

5. Add all the Input Locales required and set the preferred default. These affect how the keyboard is mapped for typing letters and can be changed as CX-Supervisor is running by the box in the task bar
6. OK all dialogs and install new OS files from CD if requested. Reboot if required

Note: If some extended characters in standard Windows controls (like listboxes, editboxes etc) appear as solid boxes it may be necessary to select the required language on the Advanced tab for Language for non-Unicode programs. Microsoft does not document the reason for this.

Note: In some circumstances (like Russian or Greek support) it has been noted "Install files for complex script" needs to be selected on the Languages tab. Microsoft does not document the reason for this.

Windows 2000

Windows 2000 language support is reasonable. It is possible to install language support for multiple languages, multiple character sets and fonts, multiple keyboard layouts for input and to switch between them. To achieve this, follow these steps:

1. Start Windows in the normal way
2. Open Control Panel and select **Regional Options**
3. The **Numbers**, **Currency**, **Time** and **Date** tabs settings do not affect the language support, but can be set as required
4. On the **General** tab, select ALL the languages that may be required in the **Language settings for the system** section

   Note: If some extended characters in standard Windows controls (like listboxes, editboxes etc) appear as solid boxes it may be necessary to change the default System Locale. Click the **Set Default…** button and select the required language.

   Note: In some circumstances (like Russian or Greek support) it has been noted "Thai" language also needs to be selected. Microsoft does not document the reason for this.
5. Add all the required languages in the **Input Locales** tab and set the preferred default. These affect how the keyboard is mapped for typing letters and can be changed as CX-Supervisor is running by the box in the task bar
6. OK all dialogs and install new OS files from CD if requested. Reboot if required

Loading old projects

Projects and pages created in CX-Supervisor 1.25 and earlier use Microsoft's Double Byte Character Sets (known as DBCS, or sometimes MBCS) for multilingual features. The formatting of character tables in MBCS is *not* the same as Unicode. However, old projects and pages are automatically converted to Unicode when they are loaded, and will then be saved in Unicode. This works without problem if the project is English, or contains no extended characters. However there can be problems converting old MBCS that use extended German, Spanish, Finnish and especially MBCS Russian characters. Follow this section to load and convert these types of projects properly.

Note: As always, when updating the format of the project file to a new file version always keep a copy of the old project for backup purposes.

Note: For the conversion process to work properly, and all extended characters to be correctly converted it is vital that the conversion takes place while the computer is configured to correctly display these characters.

Failure to follow these steps will result in extended characters being substituted for the nearest suitable character, which is likely to be incorrect and cause errors. Once saved, the conversion error cannot be undone so any occurrences will need to be edited manually. This can be time consuming and should be avoided.

These settings are only required temporarily during the conversion process. Once converted to Unicode the Operating System can be reconfigured and the Unicode characters will be displayed correctly.

Windows XP

To configure Windows XP to correctly display extended characters, to allow conversion to Unicode:

1. Open Control Panel and select **Regional and Language options**
2. Click on the **Advanced** tab.
3. From the 'Language for non-Unicode programs' select the language that the application was written for. Remember that CX-Supervisor 1.25 and earlier was a non-Unicode program.

Note: In some circumstances (like Russian or Greek support) it has been noted "Install files for complex script" needs to be selected on the Languages tab. Microsoft does not document the reason for this.

4. OK all dialogs and install new OS files from CD if requested. Reboot if required

Windows 2000

To configure Windows 2000 to correctly display extended characters, to allow conversion to Unicode:

1. Open Control Panel and select **Regional Options**
2. On the **General** tab, select the original application language in the **Language settings for the system** section

   Note: In some circumstances (like Russian or Greek support) it has been noted "Thai" language also needs to be selected. Microsoft does not document the reason for this.

3. Change the default System Locale to the original application language. Click the Set Default… button and select the required language.
4. OK all dialogs and install new OS files from CD if requested. Reboot if required

Once configured, follow these steps to convert the application:

1. It can be worth checking the language configuration, by running the old application under the old CX-Supervisor version and confirming all characters are correctly shown
2. Install the new CX-Supervisor version and load the old project.
3. Open each page and select “File menu|Save Page” to force the pages to be written in the new Unicode format
4. Select “Project menu|Save” to force the project to be written in the new Unicode format and then run the project to recompile the application
5. Test the application fully in all supported languages, and by entering text from different Locales.

Data Log Viewer

The Data Log Viewer is also supplied with translations for the 15 runtime languages. If launched from the Runtime application, it automatically starts in the same language as the runtime application. If it is started from the "Start" button, it always starts in English. The language can be selected from the standard dialog from the menu "Options|Language Settings".

Adding unsupported runtime languages

Although an impressive16 runtime languages are supported, you may require a language not listed. A developer can easily add support for further languages:

1, 2, 3…

1. Copy the file "English.lng" in the installed application folder (default is "C:\Program Files\Omron\CX-Supervisor") and rename the copy with the language name, e.g. "American.lng".
2. Translate the text between quotes. Note that formatting characters like %s or %d indicate a position for inserted text or numbers so should be left in the relevant place. Also the '&' character used in menus signifies the keyboard shortcut for the item so select an appropriate character in the translated language
3. The Runtime will now automatically show any new files with .LNG extension in the "Language Settings" dialog.
4. Repeat steps 1 and 2 above for the same file with the .LND extension. This provides the translations for the Data Log Viewer.
5. Create a new language file for the user defined text as described in the section User Defined Text above.

To deploy this language to a new machine, just install CX-Supervisor as normal, and then copy the .LNG and .LND file to the installed application folder. These languages will then be available for all applications run on this machine. When copying the application files also include the newly translated UDT files.

Popup keyboard layout

The layout of characters shown on the Alphanumeric popup keyboard can be set for existing or new languages. The layout is updated when a new language is selected.

All keyboard layouts are saved in files with .KBD extension and have the same name as the selected language.

Note: If there is no valid keyboard layout for the selected language then the default layout in SCSDEFS.KBD is used. This already caters for all the western languages supported.

To create a new keyboard layout:

1, 2, 3…

1. Copy the file “scsdefs.kbd” in the installed application folder (default is “C:\Program Files\Omron\CX-Supervisor”) and rename the copy with the language name, e.g. “American.kbd”.
2. Follow the instructions at the top of the file to add each character for each key
3. Save the new keyboard layout
4. The Runtime will now automatically update the layout when the new language is selected.

Chapter 14
Using CX-Supervisor as an OPC Client

This chapter introduces OPC Server to a new user, and explains how to use CX-Supervisor as an OPC Client.

An Overview of OPC

What is OPC?

OPC stands for **O**LE for **P**rocess **C**ontrol, and is a standard published by the OPC Foundation. The basic aim of the OPC standard is to allow hardware vendors to produce software drivers (called OPC Servers) and for software vendors to produce applications (called OPC Clients) which use a standard method for data interchange. This allows software and hardware from different vendors to be used together.

Based on Microsoft's OLE (now ActiveX), COM (component object model) and DCOM (distributed component object model) technologies, OPC consists of a standard set of interfaces, properties, and methods for use in process-control and manufacturing-automation applications. The ActiveX/COM technologies define how individual software components can interact and share data. Backed by Microsoft's NT technology, OPC provides a common interface for communicating with diverse process-control devices, regardless of the controlling software or devices in the process. The goal of the standard is Plug-and-Play, a concept developed by Microsoft and a number of other companies a few years ago. By using a standard way of configuring computer hardware (and software interfaces) automatically, a device will easily connect to another and immediately work without the need for lengthy installation procedures or complex configuration. Instead of having to learn how to use 100 or more custom toolkits, users will only have to learn one set of tools, because all OPC drivers will work the same way. OPC's purpose is to compel the automation industry suppliers to push all device drivers toward a standard form. Essentially, OPC defines a common interface that permits interface development work to be performed once and then easily reused. The OPC standard requires hardware suppliers to provide front-line data collection and distribution. They are the most familiar with how to access the device's internal data efficiently. These devices then become OPC servers, providing data to OPC client applications consistently. Application developers can then write code in any language deemed appropriate.

The latest version of the OPC standard is version 2. This replaces the earlier version 1 standard.

CX-Supervisor uses an OPC version 2 interface to connect to an OPC Server which is version 2 compliant. Note that the OPC version 1 interface has been superseded, and is not supported.

Users of CX-Supervisor may need some basic understanding of OPC. For more information on OPC, see the OPC Foundation web site at www.opcfoundation.org.

For instructions on configuring your DCOM settings for connection to remote PCs, see Appendix A.

A Brief History of OPC Data Access

In the early 1990s a group of people from several important companies involved in the creation of SCADA (Supervisory Control and Data Acquisition) systems began meeting at Microsoft's headquarters in Redmond. Their interests focused on the use of the Windows operating system within the factory automation environment, in particular for the acquisition of real time data. Microsoft at the time were working on the development of OLE 2.0 (Object Linking and Embedding) and it was apparent that this new technology would replace that of DDE (Dynamic Data Exchange) which up until that point had been used extensively for data exchange within SCADA systems designed for Windows. The new OLE technology was more flexible, robust and efficient than DDE.

OLE provided an opportunity to create a standard interface between the SCADA core and the device drivers responsible for reading and writing data to various automation devices such as PLCs. Such a standard interface would benefit both the SCADA vendors and equipment suppliers as the SCADA vendors would not need to invest costly effort in developing software drivers, while the equipment manufacturers could provide just one driver that would work with all Windows software – in the same way that printer manufacturers already could.

The first draft of the OPC (OLE for Process Control) specification v1.0 was released in December 1995. The following year, the group of companies involved in the definition of the standard decided that an independent body must be set up to manage the OPC specification. This decision resulted in the formation of the OPC Foundation that has continued to develop the philosophy of standardised interfaces for SCADA.

In 1998 the Data Access 2.0 specification for OPC was released. This addressed several deficiencies and ambiguities in the original standard and included specifications for both the Automation interfaces (typically used by VB programmers) and Custom interfaces (typically used by C++ programmers).

In 2000, using these DA 2.0 specifications Omron created the first version of CX-Server OPC, which provides Client and Server software for the CX-Automation Suite software range.

The current version of Omron's CX-Server OPC is compliant with version 2.05 of the Data Access specification.

Other OPC Specifications

Since the first OPC Data Access specification was produced in 1995 the OPC Foundation have addressed a number of additional areas of control and automation normally associated with SCADA. The original specification for Data Access is now just part of a whole series of specifications that include such areas as Alarms/Events and Batch control, although the DA interface is still far more commonly supported than the others.

The figure below shows some of the current areas covered by OPC Interface Specifications:

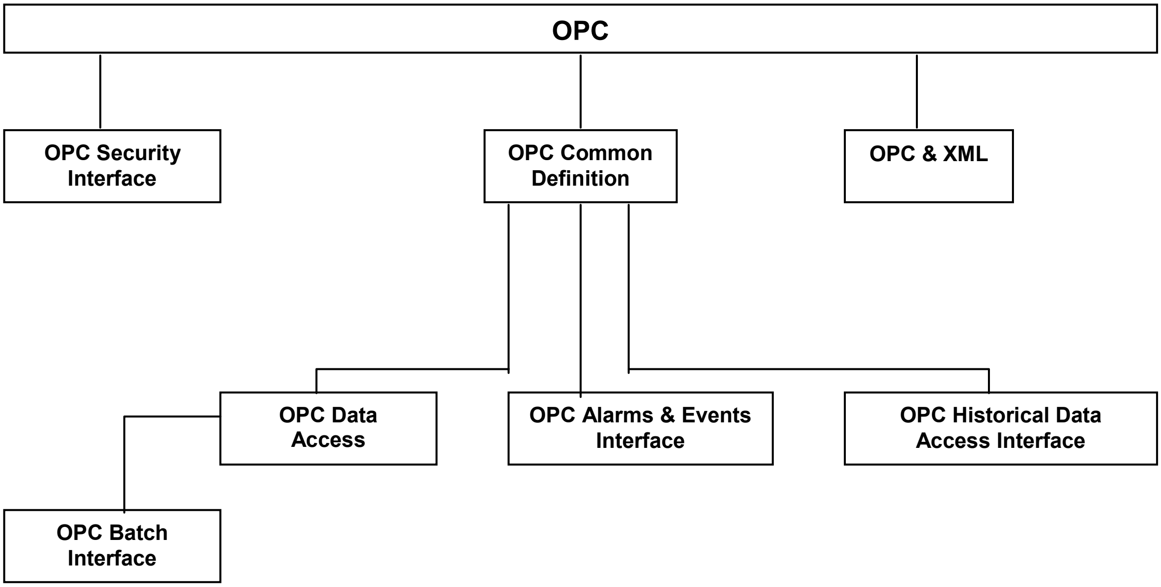


Figure 1: **OPC Interface Specifications**

Key Technologies used by OPC

This section provides, for convenience, a brief introduction to some key technologies that are used by, or that form part of, OPC. Some of these are described in more detail in the appendices; plentiful reference books are available elsewhere describing all of them in any required level of detail.

A Brief Introduction to DCOM

Microsoft describes DCOM as:

> “The Distributed Component Object Model (DCOM) is a protocol that enables software components to communicate directly over a network in a reliable, secure, and efficient manner. Previously called "Network OLE," DCOM is designed for use across multiple network transports, including Internet protocols such as HTTP. DCOM is based on the Open Software Foundation's DCE-RPC spec and will work with both Java applets and ActiveX® components through its use of the Component Object Model (COM).”

In other words, DCOM is an object-programming model for the implementation of distributed applications using a client server pattern. A client can use several servers at the same time and a server can provide functionality to multiple clients simultaneously.

Translating that into plainer English, COM basically allows software components to be written in such a way that they can be used by all COM-aware applications (e.g. C++, later versions of Visual Basic) without those applications needing to know anything about the "internals" of the object. DCOM is simply the distributed version of COM – i.e. the objects can be spread across a network. It is a very powerful system, although some machine-level and security configuration may be required to allow it to work correctly and reliably.

Central to the capability of a DCOM object are its interfaces. **All** communication with a DCOM object occurs through its interfaces – an interface is said to provide a "contract" (a full and unchanging description) for the functionality provided by that object. Each interface has a unique ID and describes a group of related methods. The description of the interface defines the syntax and the semantics of the services provided by that interface – the internal implementation of those services doesn't matter to the calling applications.

In the case of OPC these interfaces for each of the DCOM objects are defined within the relevant specifications. This guide only deals with the OPC Data Access specification Custom (used by C++) and Automation (used by script languages and VB) interfaces.

What are the benefits of OPC?

Asking what the business benefits of OPC are is like asking what the benefits of plug-and-play technology are to the computer industry. More choices, better access to process data, ease of plug-and-play operation, and efficient utilization of development resources are the main benefits of OPC technology. OPC brings the value that comes with the use of standards, including reduced training costs, reduced custom development costs, and lower long-term maintenance costs. By design, OPC-compliant products work seamlessly with one another. With this plug-and-play approach, off-the-shelf components can be brought together efficiently to solve immediate requirements. In addition, long-term maintenance and upgrading can be achieved by removing and replacing individual components in a system without any work needed to "wire up" the new pieces. To illustrate the savings, imagine the increase in cost if every household appliance had its own type of wall plug. Eliminating customisation drastically reduces the cost of an automation system by saving money during acquisition, installation, and maintenance.

Freedom of choice

With the introduction of OPC-compliant manufacturing automation products, users are provided their due right to select and implement systems comprised of best-in-class components without the pain of custom interfaces. This user benefit is sometimes referred to as “freedom of choice.” For example, both Netscape and Internet Explorer can browse the web equally well, but people use the browser they like best. As a result of this freedom of choice, vendors will need to become more competitive and offer superior products and solutions to maintain their customers. Besides freedom of choice, the user also has vendor independence, or “freedom from a proprietary lock.” If the implemented control system is comprised of modules with proprietary interfaces, any customer who desires to upgrade any component function of the integrated whole is entirely dependent on the vendor. With OPC components, only the module of interest must be upgraded and not the entire system. The requirement to use the original vendor is eliminated. High-priced proprietary solutions (and their expensive after-sale support contracts) will yield to lower cost OPC-enabled alternatives.

Time Reduction through Lower System Integration Costs

OPC eliminates the need for costly custom software integration. OPC provides plug-and-play software and hardware components from a variety of automation software, device, and system suppliers. Process and manufacturing companies can easily integrate applications into corporate-wide automation and business systems, something that has been virtually unachievable in the past. OPC-compatible components greatly reduce system integration costs because all software and hardware components adhere to a single, standard interface that’s being adopted around the world. Automation suppliers are providing hardware devices with integrated OPC servers that are replacing proprietary device-driver software. The driver connection between hardware and software from different vendors has historically been the number one headache in system integration. OPC offers the opportunity to ease the pain and shorten the application development cycle. This gets automation projects up faster, which saves time for new projects and brings the benefit of automation to the process sooner.

Who should care about OPC?

You should care about OPC if your applications are largely run in personal computers and you are involved with solving plant integration problems. As a critical mass of servers and OPC-enabled applications become available, OPC is likely to become an important part of your plant integration tool set. The OPC specification promises a future without proprietary interfaces that will greatly benefit both manufacturing customers and automation suppliers.

How is OPC going to improve my bottom line?

From a business perspective, the use of OPC for connectivity promises to reduce the cost of automation, control, and integration solutions. By using OPC compliant products, significant savings can be achieved through shorter development efforts and a wide choice of vendor hardware and software solutions. For every automation system installed today, a significant amount of time and money is spent ensuring that the system can share information with other systems and devices. OPC will save the customer time and money by eliminating a lot of the system integration problems caused by lack of open standards that exist between automation devices, systems, and manufacturing software. After an automation system is installed, OPC will not improve business bottom line directly; however, OPC will provide a common method to access real-time information. The real key to improving the bottom line is to distribute and use the information throughout the business' value chain.

Using CX-Supervisor with OPC Servers

The exact procedure for connecting CX-Supervisor to Third Party OPC Servers will depend on the server being used. Consult your server documentation for full details. However, the following is a basic overview:

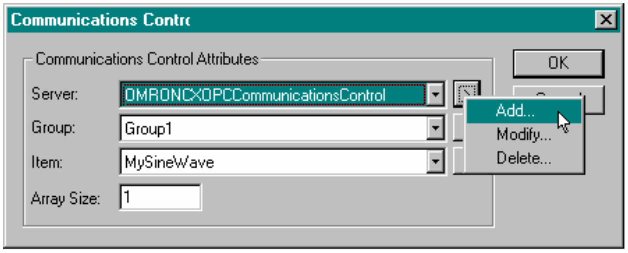
1, 2, 3… Start

Start the OPC Server and configure any settings and workspace items as described in the vendor's manual.

Start

1. Start CX-Supervisor and open your application.
2. Open the Point Editor by selecting the Point Editor option from the Utilities menu or by clicking the toolbar button.
3. In the Point Editor dialog click the Add Point button in the toolbar to open the Add Point dialog.
4. In the Add Point dialog move to the **I/O Type:** options and select the Input, Output or Input / Output option as appropriate. Note that the **I/O Update Rate:** and **I/O Attributes:** options and displayed.
5. In the **I/O Attributes:** options select **OPC/Other** and click the Setup button. This will open the Communications Control Attributes dialog.
6. In the Communications Control Attributes dialog enter the appropriate parameters as follows:

Communications Control Attributes



Server: OMRONCXOPCCommunicationsControl

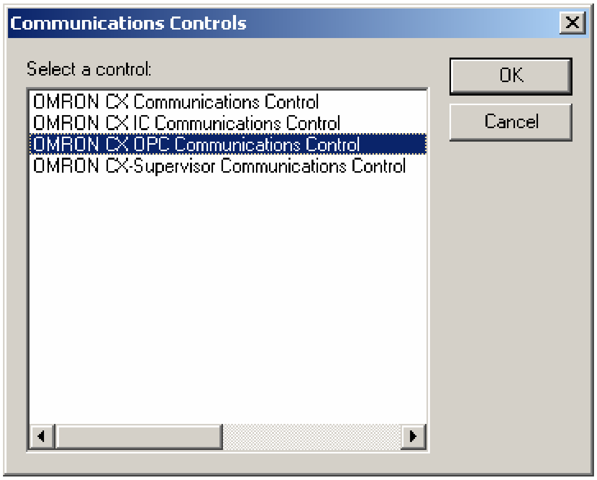
Group: Group1

Item: MySineWave

Array Size: 1

Server: This shows the name of the communications object to connect to the server i.e. OMRONCXOPCCommunicationsControl. If you wish to use a different object make your selection from the drop down list. To Add, Modify or Delete an object click the Info button. You will only need 1 communications object per OPC server but note you can connect to more 1 server at a time if desired using this setting.

This field is initially blank and an OPC control should be added. Select the Add option, which will open the Communications Control dialog. From the list of control objects select the one to be added and click the OK button.



Select a control:

- OMRON CX Communications Control
- OMRON CX IC Communications Control
- OMRON CX OPC Communications Control
- OMRON CX-Supervisor Communications Control

Note: On adding, the Communication Control Properties dialog is automatically shown as detailed below. To edit these properties later select Modify from the Info menu.

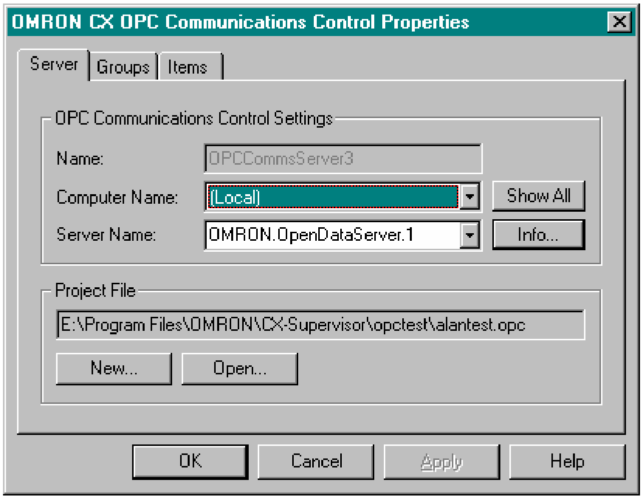
Group: Select the Group containing the required item from the dropdown list. To Add, Modify or Delete a Group click the Info button.

Items: The Items within the selected Group are listed. The Item Attributes dialog can be used to add or edit the items. To Add, Modify or Delete an Item click the Info button. Selecting the Add or Modify option will open the Items Attributes window from where the item details can be added or updated.

Array Size: Enter the number of array elements in the CX-Supervisor point that this item should have. If not an array, enter 1.

1) **OPC COMMUNICATIONS CONTROL PROPERTIES**

a) **Server Tab** The options in this window allow you to select the correct name of the computer with the OPC Server and select the appropriate server.



OMRON CX OPC Communications Control Properties

Server | Groups | Items

OPC Communications Control Settings

Name: OPCCommsServer3

Computer Name: (Local)

Server Name: OMRON.OpenDataServer.1

Project File

E:\Program Files\OMRON\CX-Supervisor\opctest\alantest.opc

OPC Communication Control Settings

Name: – This is the name of the selected communications control. The default name for the first control is OMRONCXOPCCommunicationsControl. For each additional control added the number increments by one.

Computer Name: – This shows the system name for the computer with the OPC server. If the server is on the same computer as CX-Supervisor, set to '(Local)'. If the OPC server is on a remote computer click the Show All button and select the correct name from the list. If you cannot identify the PC in the list contact your system administrator.

Server Name: – This shows the names of the OPC Version 2 compliant servers installed on the specified computer. Make your selection from the drop down list. Clicking the Info button opens the Server Status Information dialog for the selected server, check the information is correct and click the OK button.

Note: If the server is not shown in the drop down list, the server may not be fully OPC version 2 compliant. Check the server documentation.

Project File: – This is the name of the current Project file which stores the setup of the OPC items. If no name is shown or it is incorrect click the New or Open button as appropriate.

Clicking the **Open…** button opens the Open Project dialog from where you can navigate to the project file you wish to open. If you want to create a new file click the New… button to open the Create Project dialog. In the Create Project dialog navigate to the directory in which you want to create your new file and enter your File Name. Select the appropriate file type (Save as type – i.e. File Name.opc) and click the Save button.

Note: A Project File name can not be entered from the keyboard. File names can only be entered by using the New or Open buttons.

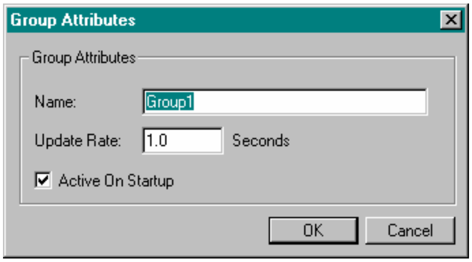
b) **Groups Tab** The groups within the selected project file are listed. A Group is made up of one or more items (data points) that share a common update, rate enabling them to be controlled as a single entity. For example you could create any number of groups each containing Items that have the same update rate.

There is no limit to the number of groups you can have in a Project File and each group can contain any number or type of Items (data points) provided they have the same update rate.

Note: This dialog cannot be opened until a valid project file exists in which to create the groups.

Groups: – Use the Add, Edit and Delete buttons to show the Group Attributes dialog and update the list of groups. For identification purposes groups should be given logical names. The defaults are Group1, Group2 etc.

i) GROUP ATTRIBUTES



Group Attributes

Group Attributes

Name: Group1

Update Rate: 1.0 Seconds

☑ Active On Startup

Name: – This is the name of the selected group. If you are editing an existing group name or creating a new one the current or default name is overwritten. The default name for new groups are Group1, Group2 etc.

Update Rate: – This is the rate at which the Items (data points) in the group will be interrogated and new data entered if it is a readout, or read from if it's a control. Enter the rate as a numerical value and select the units from the drop down list. The minimum value is 100msec; the maximum is 99999Sec i.e. over 27 hours.

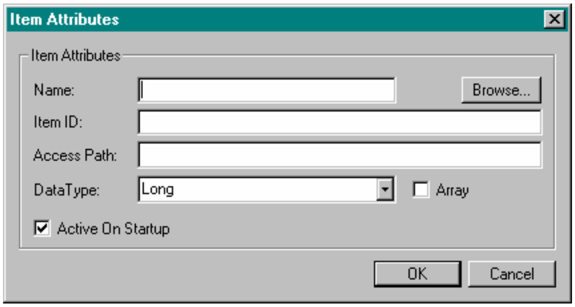
Active On Startup: – Checking this option will make the Group active when the OPC server is started. This means the items contained within will be able to read and write data. The default is checked.

c) **Items Tab** The Items within the selected group are listed. Items are data points of information stored in the OPC Server. The Item Attributes dialog can be used to add or edit the items.

Note: This dialog cannot be opened until at least one group is added.

i) ITEM ATTRIBUTES

The Item Attributes dialog enables you to define the attributes of each Item (data point) in the selected group.



Item Attributes

Item Attributes

Name:

Item ID:

Access Path:

DataType: Long

□ Array

☑ Active On Startup

Name: – This is the client name for an Item i.e. a point of data.

Item ID: – This is the OPC Server's name for the data. If the server supports OPC item browsing facilities, click Browse to browse the OPC Server to list the namespace groups and items to create the OPC Item ID. Otherwise, type the OPC Item ID in the format specified in the server documentation.

Access Path: – This is the full path name for the data point. Consult the documentation for the server to determine the correct format.

Data Type: – From the drop down menu select the format used to record the data point information.

Active On Startup: – Checking this option will ensure the Item is active when the OPC server is started. The default is checked.

8. Having checked that all the parameters are correct bring control back to the Point Editor window by closing any attributes window that may be open.
9. The application can now be run by clicking on the **Run Project** button in the toolbar. The CX-Supervisor Point will now be connected to the OPC Server and can be used like all other CX-Supervisor points, i.e. to drive animations, in alarm expressed, in recipes, for data logging etc.

Using with Omron's CX-Server OPC

1. Note that the Omron OPC server discussed is supplied as part of the CX-Server OPC product.
2. Omron's OPC Server configuration is described in the CX-Server OPC manual
3. While launched, the CX logo is displayed in the system tray.
4. For step **7-1-a-i**, the correct Server Name is "Omron.OpenDataServer.1"
5. For step **7-1-c-i**, the Item ID should be the name of the point in the CX-Server project (.CDM) file configured in the server. The Access Path is not used, and may be left blank.
6. During step 9, the Omron OPC Server logo is displayed momentarily as the server starts.

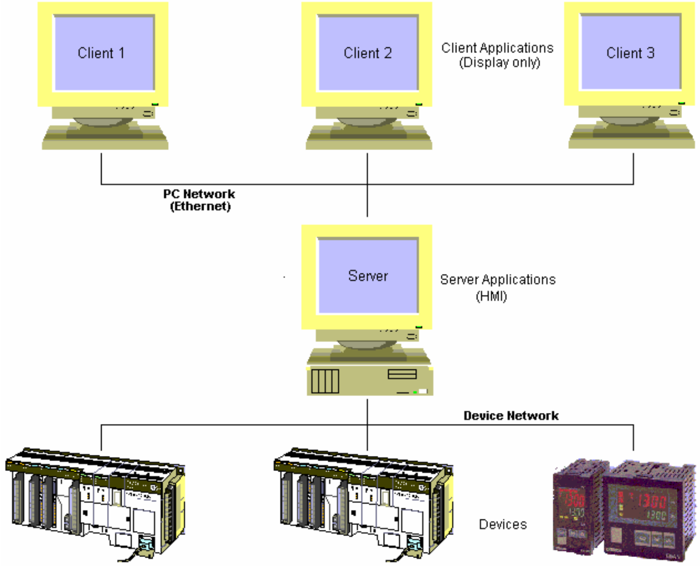
Chapter 15
Connecting to a remote CX-Supervisor application

This chapter explains how to connect multiple CX-Supervisor applications together to form a distributed solution. Point data from one application can be 'shared' directly with other CX-Supervisor applications across the PC network. Depending on the requirements, there are many reasons and topographies for interconnecting CX-Supervisor applications.

Two Tier, Client - Server or Master - Slave

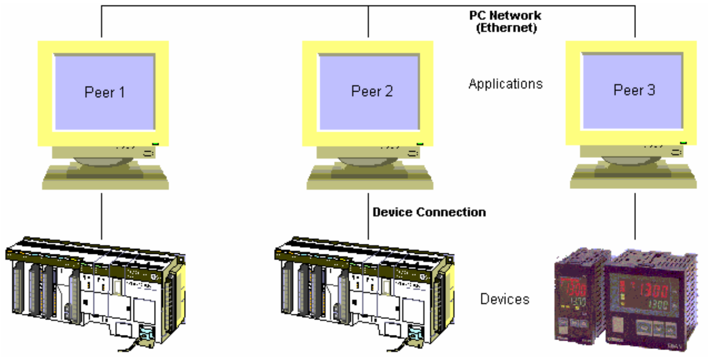
One application is configured as a *Server* application or *Master*. The *Server* is often the main station, and is permanently switched on, and often used locally for HMI for example but could have no graphical element, which is termed a *Blind Server*. It has direct access to the Devices or Device network and is responsible for collecting the data from the devices.

Other applications may connect to the *Server* to read and even control values in the Server. These applications are called *Client* applications or *Slaves*. *Client*s are often used remotely and therefore are often configured 'display-only' applications, transferring the data from the Server using the corporate PC network. The *Clients* do not talk directly to the devices, and often only connect when required.



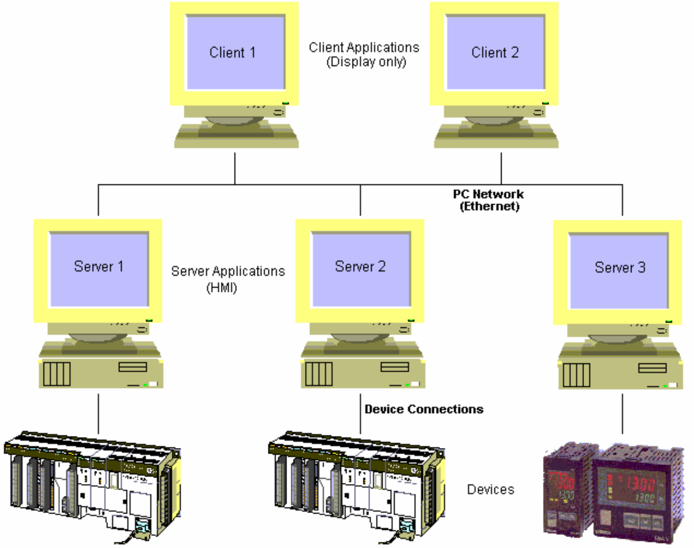
Peer to Peer

Several applications are written to share data with each other. Actually, an application is written as a Server to connect directly to a device, but is also a Client using other Servers to connect to other devices through the corporate network.



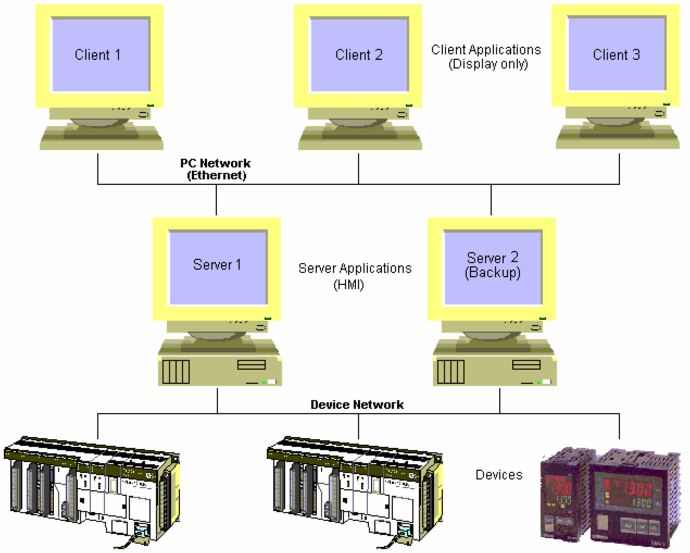
Distributed Server

Several applications are configured as Servers for direct connection, so the data for the system is 'distributed' across several machines. One or more Client applications collect the data from the distributed servers. This can be useful to help performance, by distributing the communication on the device connections, and server processing. It can also provide different security for different clients and offers limited protect against failures, as remaining servers still function.



Redundant Server

Several applications are configured as Servers for direct connection to the same Devices or Device network. The same data is collected by all Servers. One or more clients can collect the data from any single Server, and in the event of a Server failure, can be written to switch to data from an alternative Server.



A distributed solution may be any of the above, or a combination of the ideas. Once your topography is defined, the steps to connect each Client or Server are the same.

Creating a CX-Supervisor Server application

A Server application must collect device data, and allow clients to access it. It may also have other elements like graphics, control or logging. To create your application:

- Create Device points for every data value required by the Server application and any required by any client. See Chapter 3 - Points for more details.
- Configure DCOM on the server machine, to allow access from the client machines. See Appendix A for further details.
- Add any other elements required by the Server application e.g. Graphics, Control, Logging, Alarms etc.

Be sure to note the computer name of the server machine.

Note: A quick way to display the computer name is to right-click the 'Network Neighbourhood' icon on the Windows Desktop and select 'Properties'.

Creating a CX-Supervisor Client application

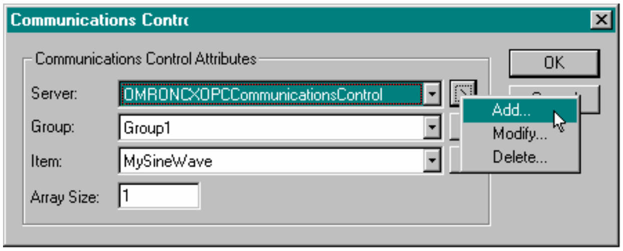
The client must retrieve the data, and process it. To create your application:

- Add any elements required by the Client application e.g. Graphics, Control, Logging, Alarms etc.
- Configure DCOM on the client machine, to allow access from the server machines. See Appendix A for further details.
- Create points for every data value required by the Client application following these steps:

1, 2, 3…

1. Start CX-Supervisor Development and open your application, or start a new application.
2. Open the Point Editor by selecting the Point Editor option from the Utilities menu or by clicking the toolbar button.
3. In the Point Editor dialog click the Add Point button in the toolbar to open the Add Point dialog.
4. In the Add Point dialog move to the **I/O Type:** options and select the Input, Output or Input / Output option as appropriate. Note that the **I/O Update Rate:** and **I/O Attributes:** options and displayed.
5. In the **I/O Attributes:** options select **OPC/Other** and click the Setup button. This will open the Communications Control Attributes dialog.
6. In the Communications Control Attributes dialog enter the appropriate parameters as follows:

Communications Control Attributes



Communications Contrc

Communications Control Attributes

Server: OMRONCXOPCCommunicationsControl

Group: Group1

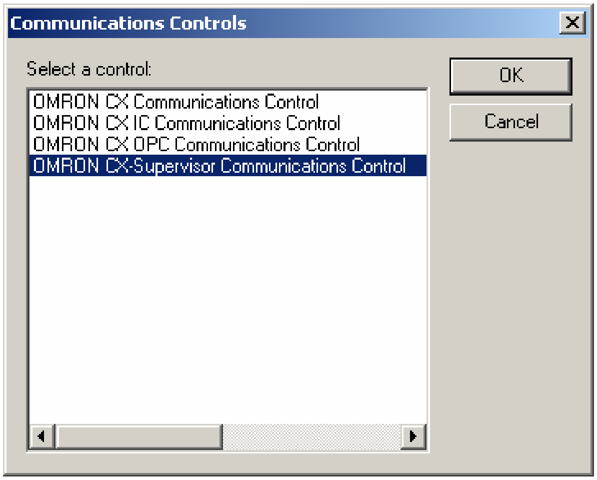
Item: MySineWave

Array Size: 1

Server: This shows the name of the communications object to connect to the server i.e. OMRONCXSupervisorCommunicationsControl. If you wish to use a different object make your selection from the drop down list. To Add, Modify or Delete an object click the Info button.

Selecting the Add option will open the Communications Control dialog. From the list of control objects select 'OMRON CX-Supervisor Communications

Control' and click the OK button.


Communications Controls
Select a control:
OMRON CX Communications Control
OMRON CX IC Communications Control
OMRON CX OPC Communications Control
OMRON CX-Supervisor Communications Control
OK
Cancel

Clicking the OK button in the Communications Controls dialog will open the Communications Control Properties dialog from where the server details can be added or updated. Click 'Show All' and select the computer name of the server machine from the list. To edit these properties later select Modify… from the Info menu.

Group: Select the Group containing the required item from the dropdown list. To Add, Modify or Delete a Group click the Info button.

Items: The Items within the selected Group are listed. The Item Attributes dialog can be used to add or edit the items. To Add, Modify or Delete an Item click the Info button. Selecting the Add or Modify option will open the Items Attributes window from where the item details can be added or updated.

Array Size: Enter the number of array elements in the CX-Supervisor point that this item should have. If not an array, enter 1.

7. Having checked that all the parameters are correct return to the Point Editor window by closing any attributes window that may be open.

The application can now be run by clicking on the **Run Project** button in the toolbar. The CX-Supervisor Point will now be connected to Remote CX-Supervisor application and can be used like all other CX-Supervisor points, i.e. to drive animations, in alarm expressed, in recipes, for data logging etc.

CHAPTER 16
Connecting to Omron Industrial Components

This chapter details connecting CX-Supervisor to Omron's Industrial Components, like Temperature Controllers, Digital Panel Meters and Timer/Counters. Any process value or parameter from these devices can be monitored or written. See the device documentation for available parameters and their description. This is achieved in CX-Supervisor by creating a point and linking it to the required parameter. Device ranges supported are following models that support communication (where '*' indicated a wildcard):

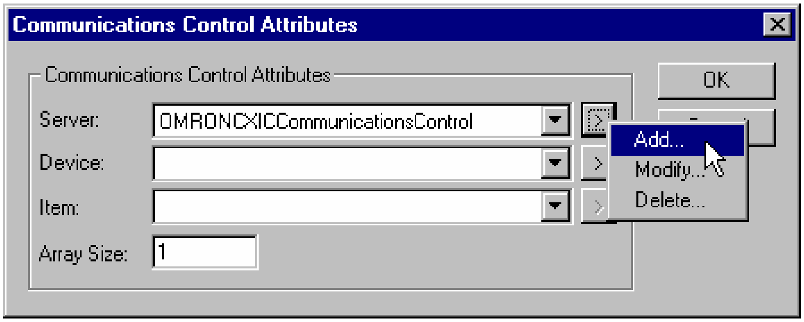
- K3GN
- E5*N
- E5*J
- E5ZE
- E5*K
- H8GN
- K3N*
- E5ZN

Adding a point linked to a parameter

1, 2, 3…

1. Start CX-Supervisor Development and open your project.
2. Open the Point Editor by selecting the Point Editor option from the Utilities menu or by clicking the toolbar button.
3. In the Point Editor dialog click the Add Point button in the toolbar to open the Add Point dialog.
4. In the Add Point dialog move to the **I/O Type:** options and select the Input, Output or Input / Output option as appropriate. Note that the **I/O Update Rate:** and **I/O Attributes:** options and displayed.
5. In the **I/O Attributes:** options select **OPC/Other** and click the Setup button. This will open the Communications Control Attributes dialog.
6. In the Communications Control Attributes dialog enter the appropriate parameters as follows:

Communications Control Attributes



Communications Control Attributes

Communications Control Attributes

Server: OMRONCXICCommunicationsControl

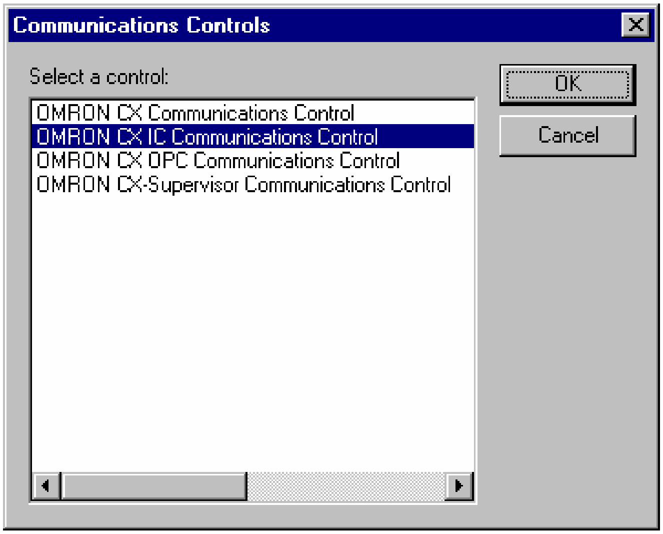
Device:

Item:

Array Size: 1

Server: This shows the name of the communications object to connect to the server i.e. OMRONCXICCommunicationsControl. If you wish to use a different object make your selection from the drop down list. To Add, Modify or Delete an object click the Info button.

Selecting the Add option will open the Communications Control dialog. From the list of control objects select the one to be added and click the OK button.



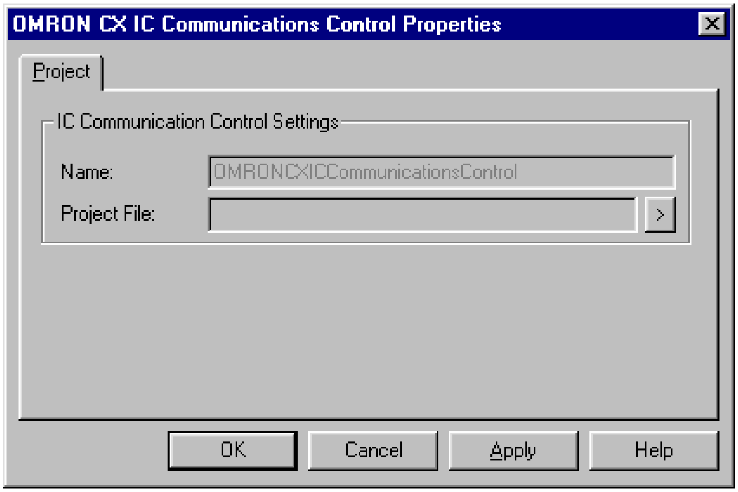
Communications Controls

Select a control:

- OMRON CX Communications Control
- OMRON CX IC Communications Control
- OMRON CX OPC Communications Control
- OMRON CX-Supervisor Communications Control

Clicking the OK button in the Communications Controls dialog will open the Communications Control Properties dialog from where the details can be added or updated. To edit these properties later select Modify from the Info menu.

CX IC Communications Control Properties



IC Communication Control Settings

Name: OMRONCXICCommunicationsControl

Project File:

Name: – This is the name of the selected communications control. The default name for the first control is OMRONCXICCommunicationsControl. For each additional control added the number increments by one.

Project File: - This is the name of the current Project file that stores the setup of the items. If no name is shown or it is incorrect click the Info button and select **Create…** or **Open…** as appropriate.

Clicking the **Open…** button opens the Open Project dialog from where you can navigate to the project file you wish to open. If you want to create a new file click the Create… option to open the Create Project dialog. In the Create Project dialog navigate to the directory in which you want to create your new file and enter your File Name. Select the appropriate file type (Save as type – i.e. File Name.msc) and click the Save button.

Note that a Project File name can not be entered from the keyboard, files names can only be entered by using the Create or Open options.

Device: Select the Device containing the required item from the dropdown list. To Add, Modify or Delete a Device click the Info button.

Items: The Items within the selected Device are listed. The Item Attributes dialog can be used to add or edit the items. To Add, Modify or Delete an Item click the Info button. Selecting the Add or Modify option will open the Items Attributes window from where the item details can be added or updated.

Array Size: Enter the number of array elements in the CX-Supervisor point that this item should have. If not an array, enter 1.

7. Having checked that all the parameters are correct bring return to the Point Editor window by closing any attributes window that may be open.

The CX-Supervisor Point will now be connected to Device parameter and can be used like all other CX-Supervisor points, i.e. to drive animations, in alarm expressions, in recipes, for data logging etc. The application can now be run by clicking on the **Run Project** button in the toolbar.

CHAPTER 17
Best Practices

This chapter details the best practices to get the most from CX-Supervisor including how to plan and design your application.

Design

This section covers some good design practices. There is no doubt, as with everything in life that a good design is essential and creating a good CX-Supervisor application is no different. Poorly designed applications may run slower, provide slower user feedback, be difficult to maintain – increasing costs and in some cases not fulfil all User Requirements resulting in the application being rewritten. Never fall into the trap "code first, think later".

- **Design your page layouts and navigation flow.**

Traditional design works well for most applications. Providing an application with a clear structure can make applications very easy to navigate even if they have many pages.

- Create a Main overview / navigation page. This can include basic information like company logo and date/time. If the application will be using security levels provide buttons to login.
- Add links from the main page to detailed system areas with either buttons or schematic diagram e.g. Process 1, Process 2.
- Add links from the main page, if required, to pages dedicated to Alarms, Diagnostics, Reports, Data Logging, Statistics etc
- If the system has repeated functions e.g. 5 similar machines, instead of creating 5 pages think about creating 1 page that can show data for any machine. This can be achieved using indirection and/or keeping data in arrays, and just setting the index to display the machine you require. Minor differences (e.g. a button for Process 1 only) can be shown or hidden depending on the index. This will speed up development and also help when the application is maintained.

Organise the pages sensibly on screen. Make sure that only the pages that are necessary are open. When a page is no longer needed close it. CX-Supervisor will update all pages that are open, therefore if lots of pages are open then it may be doing a lot of unnecessary work that will slow down the application. For pages that are displayed in the same place, set the 'Display Mode' property to 'Replace'. This allows pages to automatically be closed when a new page is loaded.

- **Use Logical Point names instead of physical addresses**

Logical point names (like 'Boiler_3_Pressure') provide more flexibility than physical addresses (like 'D8319') allowing the PLC program and addresses to be changed, without impacting the SCADA. They also make it much easier read during development, and more importantly, during maintenance.

Tip: If the Logical symbols are created in CX-Programmer, it can link them to a CX-Server file which CX-Supervisor can then 'Import' into the Points Editor. This is a very quick method of creating all the required points.

- **Perform all data conversion in PLC ladder**

Although today's PCs have excellent maths performance consider the format of the PLC data and how it will be used. It can be temping to scale a point value when read e.g. to change the units of measurement, or always perform a calculation e.g. calculate a running average or to format it e.g. make "1700" become "17:00". Instead consider writing this conversion as ladder in the PLC hardware rather than in CX-Supervisor software. This is particularly true if this conversion has to occur on several pages where the value is used or before data logging. This ensures the converted value is always available, and results in a simplified solution.

- **Perform all timing functions and pulse generation from PLC ladder**

The PLC hardware has far superior timing and accuracy than Windows can provide. When timing a process or event always use the PLC. The stored result can then be read by the SCADA at its leisure. If creating a pulse of fixed duration never attempt to design the SCADA to set a bit, wait, and clear the bit. Instead, allow the SCADA to set the bit but use the PLC program to clear the bit after the required time.

- **Operation should continue if the SCADA is unplugged**

In a well designed system normal operation can continue even if the SCADA is unplugged. Some functions like data logging, batch control and control obviously will not be available but the system should be able to continue with its task until completion. That is, the process should not rely on co-operation between the SCADA and PLC because of the possibility of failure e.g. of the power supply, PC components or operating system.

- **Ensure the PLC has hardware failsafes**

In situations where the system can cause damage e.g. to itself, the materials or people it is essential to ensure the PLC has suitable safeguards and failsafes to prevent such damage. Given the possibility of SCADA failure it is not acceptable to implement such failsafes in software. Care should also be paid to the design of software 'control' to allow and cope with the possible refusal of operation from the PLC.

Performance

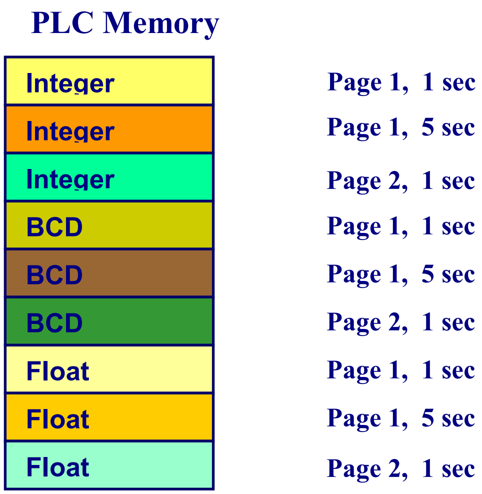
This section covers some good practices to ensure excellent performance.

- **Organise the PLC memory properly**

This is probably the single most important practice to provide maximum performance and the best responsiveness.

It is important to design the PLC program and SCADA application together. This will naturally help create arrays of information, and optimise communications allowing CX-Server to collect data in the most efficient manner.

Consider the examples in the following 2 figures:



PLC Memory

Integer	Page 1, 1 sec
Integer	Page 1, 5 sec
Integer	Page 2, 1 sec
BCD	Page 1, 1 sec
BCD	Page 1, 5 sec
BCD	Page 2, 1 sec
Float	Page 1, 1 sec
Float	Page 1, 5 sec
Float	Page 2, 1 sec

Figure 1, Bad grouping example

In Figure (1) we see the PLC Programmer has arbitrarily chosen to group data by its format: Integers, then BCD then Floats (or even worse not at all!). When the SCADA is written, this data is used by different pages and different update rates. The different colours are to indicate that each block must be read individually, totalling 9 communication requests, which could be for as few as 9 memory addresses.

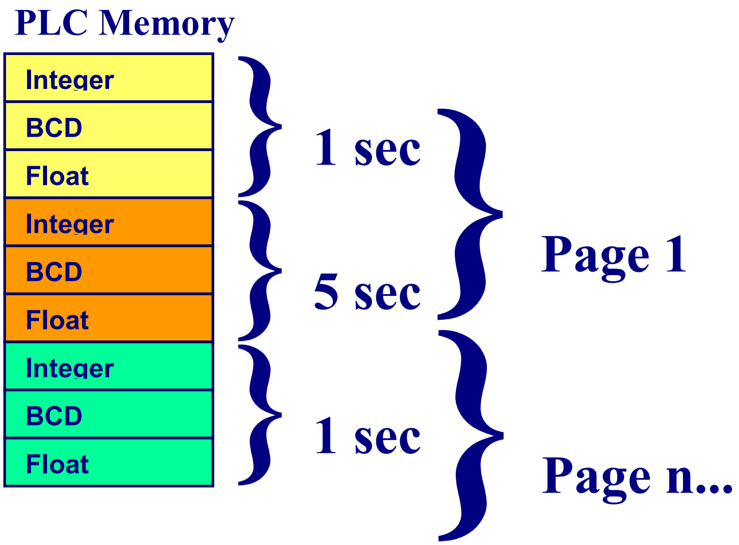


Figure 2, Good grouping example

However in Figure (2) we see the PLC Programmer and SCADA developer have reorganised the memory now there are only 3 blocks, which would be the same for up to 3000 memory addresses. This is clearly far better than just 9 memory addresses with 9 communication requests. To achieve this, design your application using the following rules:

1. Group together points that will be configured as "Always Update" in contiguous PLC memory addresses
2. Group together points that will be configured as "Update when On Display", grouping for each page
3. Within groups from steps 1 & 2, create sub-groups for each required update rate
4. Within update rate sub-groups, sort and group data by data format (BCD, Float etc). These points can be defined as 1 or more arrays.

- **Use Fins event memory**

If your communications is using a Fins driver (e.g. Controller Link) it is possible to create a responsive application giving the impression of high performance by linking the SCADA to the event memory, which the network updates automatically. Of course reads and writes actually access the local cache very quickly although this will lag behind actual device values.

- **Control communications**

Use the 'On Request' update type to give you complete control over when the value of a point is read from the PLC or written to the PLC. The 'InputPoint' script function is used to input the value of a specified point and the 'OutputPoint' script function is used to output the value of a specified point.

Points

This section covers some good practices with the Points for quicker development and smoother running.

- **Define as few points as possible.**

Keep the Points database as slim and manageable as possible by:

  - Using arrays where possible especially for PLC points
  - If memory points are defined for scripting try to reuse general purpose points like ‘nLoopCounter’, ‘bReturnValue’, ‘nTemp’
  - Use expressions in animations instead of calculating new values e.g. using “TemperatureK + 273” rather than creating a new “TemperatureCelsius” memory point.
  - For memory areas that are bit and word addressable, access as words. If you want the bits use expressions with & and | operators e.g. Display Digital animation expression ‘MyW100 & 0x80’ and ‘MyW100 & 0x40’ instead of ‘MyW100bit8’ etc.

- **Reduce the number of active points**

Aim to reduce the number of active points, that are currently communicating where possible:

  - Automatically by setting “Update when on Display” option
  - Manually using script (EnablePoint(), DisablePoint(), OpenPLC(), ClosePLC() etc).

- **Have a minimum of Input/Output points**

Double-check the use of all Input/Output points. Don’t forget that ‘Output’ points can actually have ‘Data Transfer’ set to input the value only once on start up.

- **Use 'Standard' update rates.**

CX-Supervisor only allows update rates in standard groups, which allows a greater chance for CX-Server to optimise reads together. Old projects may still be using non-standard update rate and are best changed to the standard rates.

- **Use few update rates.**

Even when using the standard update rates, use as few different update rates as possible, ideally 5 or less different rates. Try to use the same update rates for the same areas of PLC memory so optimisations can occur.

- **Use sensible update rates.**

Aim to keep update rates low. 5 seconds or 1 second updates should suffice. If faster than 1 second updates are needed always question "Why?". If fast update rates are required pay special attention to consider if they can be disabled when not required.

- **Use Copy/Paste to Excel for bulk editing**

During development, the point editor supports Copying and Pasting in textual format so Excel's powerful editing / formulas can be used to create definitions for points. This can allow thousands of consecutive points to be created in seconds.

- **Use <Shift> and <Ctrl> to select multiple points for editing**

During development, using <Shift> and <Ctrl> while selecting can be useful to apply the same changes to many points.

- **Use Hungarian notation**

Improve readability and maintenance by defining and using a standard naming convention. For example Hungarian notation where variable names are prefixed by a type character e.g. bMyBooleanPoint, nMyIntegerNumber, txtMyTextPoint or rMyReal.

- **Use default optimisations**

Leave the default communications optimisations turned on. In nearly all cases they really do provide the best solution. Seek advice if you are considering turning them off.

- **Delete unused points**

After a development phase and prior to deployment, delete any unused points. These are listed in the Point Editor under the group "<Unused Points>".

Drawing

This section covers some good practices when drawing the screens. These can greatly speed up application development and help produce a professional looking system.

- **Start new pages with ‘Snap to Grid’**

During the creative phase of any page this helps ensure objects are automatically aligned and sized, saving time tidying up later.

- **Use cursor keys for fine adjustments**

The cursor keys move selected objects by 1 pixel, or snaps to grid if <Ctrl> held, and vice-versa when ‘Snap to grid’ is active. This is a lot easier than using the mouse.

- **Group objects**

To make Workspace navigation of objects easy, reduce the number of top level objects by collecting objects together into Groups. Grouped objects can be nested on other groups in a natural hierarchy e.g. a machine has several panels that have keypads that have buttons.

- **Create object libraries**

Keep a common look and feel between pages and speed up development by creating libraries of commonly used objects e.g. displays or buttons. These can also be reused in later projects.

- **Use Copy and Paste**

Making good use of Copy and Paste to duplicate similar objects can reduce development time.

- **Use Alignment functions**

Create professional looking applications by prevent unsightly errors quickly using ‘Align Top/Left’ and ‘Make Same Width/Height’.

- **Draw perfect circles**

Quickly draw perfect circles by holding <Ctrl> while drawing an Ellipse.

- **Use meaningful object names**

Instead of accepting the default object and group names rename them to something more helpful e.g. Conveyor1, InstrumentPanel, StartButton etc

Scripts

This section covers some good practices for writing scripts.

- **Never use ‘On Regular Interval’ scripts**

You should (almost!) never need to use ‘On Regular Interval’ scripts. Always question:

  - If it is used for logging - use the logging facilities instead
  - If it is used for timing - use PLC ladder instead as this is much more reliable.
  - When they start with (or contain) “IF <condition> THEN …” you should probably be using an “On Condition” script using <condition> from the IF statement.
  - When they perform calculations on PLC or memory points (e.g. “Z = X * Y), use “On Condition” to recalculate when the source data changes e.g. with a condition X || Y || TRUE (which forces execution even if value changes to value 0). This will *guarantee* the calculated is up to date with the latest source information, plus stress the system much less while the value doesn’t change.

- **Never use Sleep() command**

You should (almost!) never need to use the Sleep command. It can also create confusing applications as it can create re-entrant scripts and asynchronous PLC communications. Always question:

  - If it is used for timing - use PLC ladder instead as this is much more reliable.
  - Consider if the code following the Sleep should in fact be executed on an event e.g. after completion flag returned from PLC instruction or new data value etc

- **Use Animations if possible**

Always choose direct object animations in preference to page/object scripts: blink, colour, disable, height, width, horizontal%fill, vertical%fill, move, rotate, display and close. These are both quicker and clearer.

- **Keep scripts short**

Although the script editor can manage hundreds of lines this would suggest a serious design problem. Consider 30 lines as a practical maximum

Data Logging

This section covers some data logging related good practices.

- **Utilise Data Logging performance**

Always use the inbuilt Data Logging in preference to trend graphs or file I/O functions, as the performance is far superior.

- **Always use “On Change”**

Always log data “On Change”. There is no sampling error and CPU, disk activity and storage required are almost certainly reduced.

- **Use deadband option**

Consider using the deadband option, especially for noisy analogue signals to reduce the actual logging and storage requirements.

- **Don’t “Keep all files”**

The Dataset “Keep all files” checkbox defaults ‘On’ for complete data retention but this should be cleared to stop the Hard Disk filling up.

APPENDIX A

Configuring a PC for remote connection

The OPC interface and CX-Supervisor Communications Control use a Microsoft technology called DCOM. This allows the Clients and Servers to be seamlessly 'Distributed' over a PC network. The Server should be running on the PC with direct connection to the Device or Device network. However, the Client, or indeed multiple Clients, can be run on different networked PCs and will automatically read and write data over the PC network. To do this, the PC running the Server must be correctly configured. If necessary refer to your OPC server documentation. For full details of DCOM configuration and security issues see your Microsoft documentation. The following is a quick guide:

Note: Although DCOM connections can be achieved using Windows 98 and ME they are more difficult to configure, and offer less functionality. For best results always use Windows NT or Windows 2000.

Configuring a Client PC running Windows XP Service Pack 2

Windows XP Service Pack 2 includes a communications 'Firewall' to protect your computer from malicious communications. By default, this firewall is turned on, and so will block all OPC and DCOM communications.

1, 2, 3… To allow OPC or DCOM communications through the firewall *either*:

a) Completely disable the firewall as follows. Note can leave your computer vulnerable to attack so consult your computer documentation or IT administrator if you are unsure:

1. Open **Control Panel** followed by **Computers**.
2. Right click **My Computer** and select **Security Center**
3. Select **Windows Firewall**
4. On the General tab choose option **Off**

b) Alternatively, on the Exceptions tab you can configure specific programs to permit communications

All other required settings are the same as for running standard Windows XP as explained below.

Configuring a Client PC running Windows XP

1, 2, 3…

Start

1. Start **Component Services** e.g. by running DCOMCNFG.EXE by selecting RUN from the start button. The default location is C:\WINDOWS\SYSTEM.
2. Select **Component Services** followed by **Computers**.
3. Right click **My Computer** and select **Properties**
4. View the **Default Properties** tab. Ensure that the **Enable Distributed COM on this computer** is checked.

Configuring a Client PC running Windows NT or 2000

1, 2, 3…

Start

1. Start DCOMCNFG.EXE e.g. by selecting RUN from the start button. The default location is C:\WINDOWS\SYSTEM.
2. View the **Default Properties** tab. Ensure that the **Enable Distributed COM on this computer** is checked.

Configuring a Server PC running Windows XP Service Pack 2

Windows XP Service Pack 2 includes a communications 'Firewall' to protect your computer from malicious communications. By default, this firewall is turned on, and so will block all OPC and DCOM communications.

1, 2, 3… To allow OPC or DCOM communications through the firewall *either*:

a) Completely disable the firewall as follows. Note this can leave your computer vulnerable to attack so consult your computer documentation or IT administrator if you are unsure:

Start

   1. Open **Control Panel** followed by **Computers**.
   2. Right click **My Computer** and select **Security Center**
   3. Select **Windows Firewall**
   4. On the General tab choose option **Off**

b) Alternatively, on the Exceptions tab you can configure specific programs to permit communications

All other required settings are the same as for running standard Windows XP as explained below.

Configuring a Server PC running Windows XP

1, 2, 3…

Start

1. Start **Component Services** e.g. by running DCOMCNFG.EXE by selecting RUN from the start button. The default location is C:\WINDOWS\SYSTEM.
2. Select **Component Services** followed by **Computers**.
3. Right click **My Computer** and select **Properties**.
4. View the **Default Properties** tab. Ensure that the **Enable Distributed COM on this computer** is checked.
5. From the **Default Properties** tab, configure the **Default Authentication Level** to **Connect** and the **Default Impersonation Level** to **Identify**. Setup the access permissions by either:
   a) On the **Default COM Security** tab, adding the user to the Access, Launch and Configuration lists by clicking the **Edit Default…** button in each case. The user added should have Administrator rights on the local PC. If not, it may be necessary to add user groups 'INTERACTIVE' and 'NETWORK' as well.
   b) From the **My Computer** item in **Component Services**, select **DCOM Config** and configure the properties for required applications e.g. **SCS**, **OpenDataServer** and **OpcEnum**. On the **Security** tab, add the required users to each of the **Custom Permissions**. The users added should have Administrator rights on the local PC. If not, it may be necessary to add user groups 'INTERACTIVE' and 'NETWORK' as well.

Configuring a Server PC running Windows NT or 2000

1, 2, 3…

Start

1. Start DCOMCNFG.EXE e.g. by selecting **RUN** from the Start button. The default location is C:\WINDOWS\SYSTEM.
2. View the **Default Properties** tab. Ensure that the **Enable Distributed COM on this computer** is checked.

3. From the **Default Properties** tab, configure the **Default Authentication Level** to **Connect** and the **Default Impersonation Level** to **Identify**. Setup the access permissions by either:

   a) On the **Default Security** tab, adding the user to the Access, Launch and Configuration lists by clicking the **Edit Default…** button in each case. The user added should have Administrator rights on the local PC. If not, it may be necessary to add user groups 'INTERACTIVE' and 'NETWORK' as well.

   b) From the **Applications** tab, configure the properties for required applications e.g. **SCS**, **OpenDataServer** and **OpcEnum**. On the **Security** tab, add the required users to each of the **Custom Permissions**. The users added should have Administrator rights on the local PC. If not, it may be necessary to add user groups 'INTERACTIVE' and 'NETWORK' as well.

APPENDIX B
Frequently Asked Questions

The appendix list some FAQs, and of course their answers.

Q What is the maximum limit of number of points?

For Machine Edition projects it is possible to create an application with up to 500 user defined points. In theory each of these could be an array of 1024 elements totalling 512,000 addresses. For PLUS edition there is a limit of 8000 user defined points, totalling over 8 million addresses if using arrays !!

In practice the communications medium, number of active points and their frequency of updates determine the maximum limit.

See “What is the maximum limit of communication updates?”

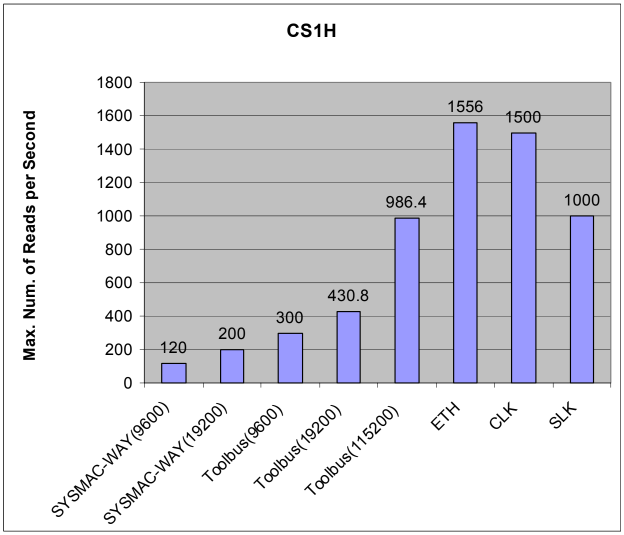
Q What is the maximum limit of communication updates?

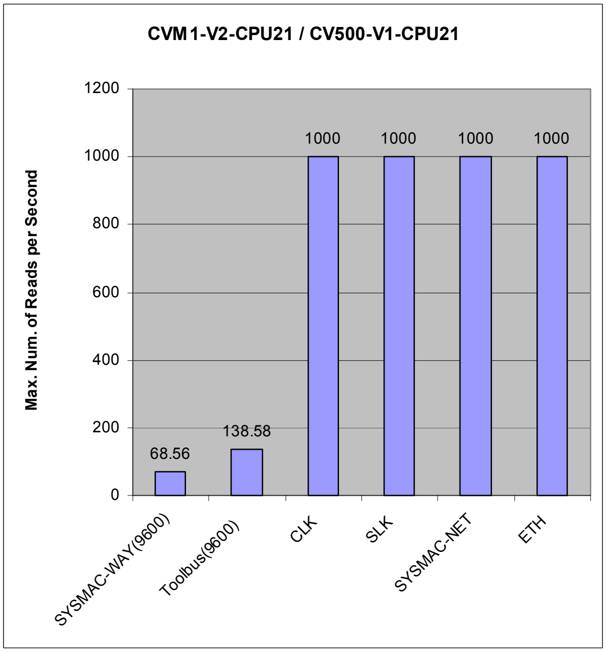
The charts below show the number of simultaneous reads per second depending on each PLC and Network used. The maximum number of CX-Supervisor points depends on the update rates and element sizes. For example, 1000 reads per second could be 1000 points of 1 element, with a 1 second update rate, or 200 array points with 5 elements with a 1 second update rate, or 20 array points with 5 elements with a 100 millisecond update rate, or any combination of variations.

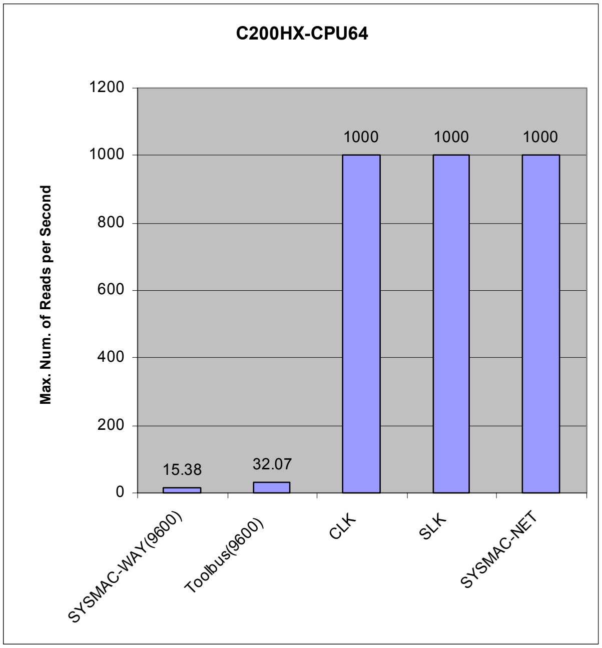
The figures should be self-explanatory, but its worth noting that Toolbus is better than twice as good as SYSMAC WAY at the same baud rate, and depending on the PLC supports higher baud rates. This is a very cheap way to improve performance.

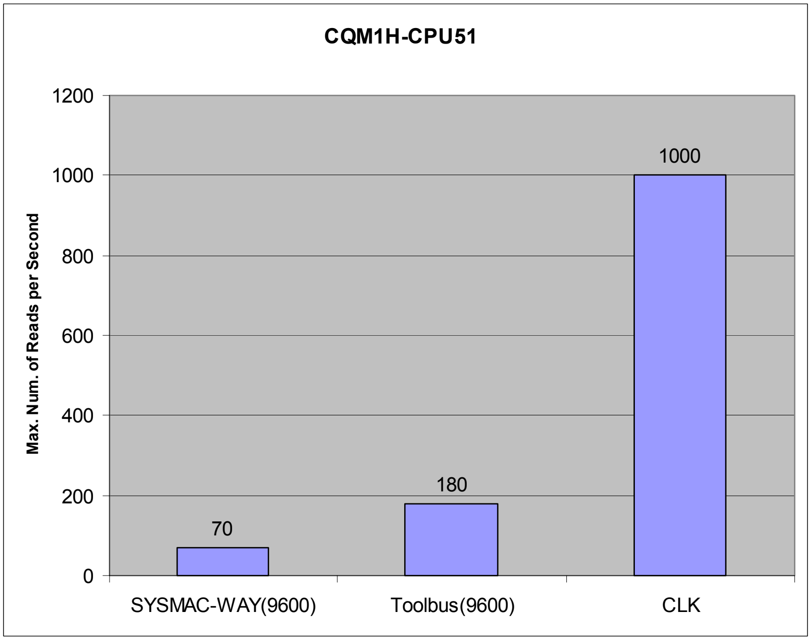
Note 1: These figures assume best-case scenario, where data is in contiguous addresses. Lesser performance will be achieved as memory areas are fragmented.

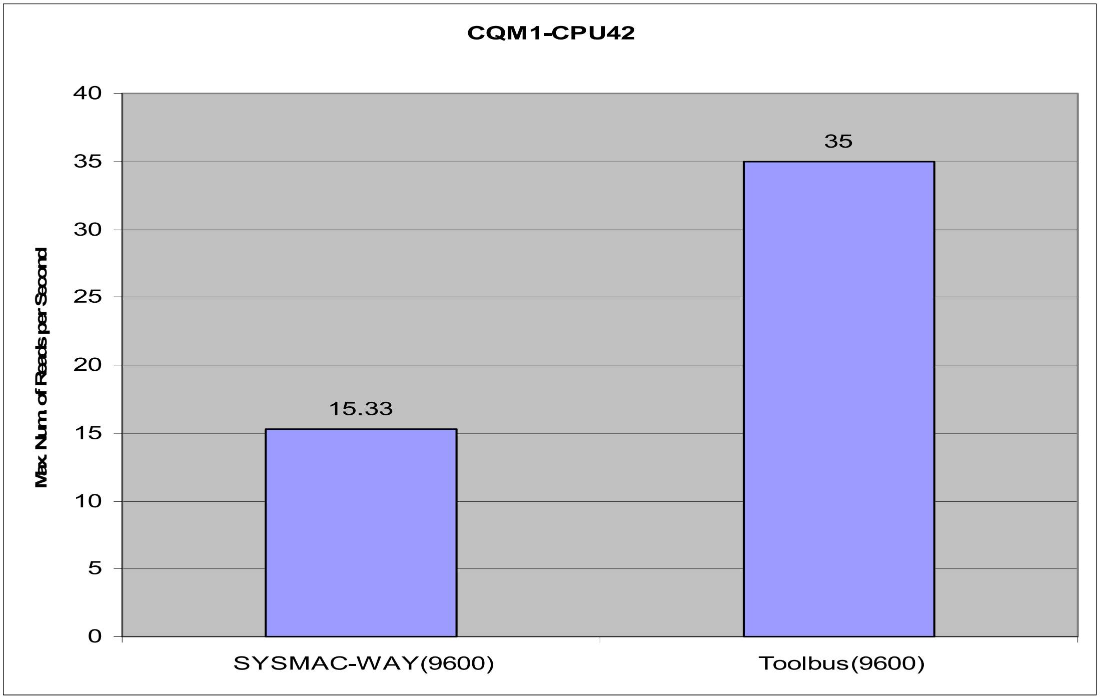
Note 2: These tests stopped at 1000 updates so actual limits are beyond this figure, except for CS1H Ethernet and Controller Link Networks, which has been tested for 10,000 updates per second.


CS1H
Max. Num. of Reads per Second
0
200
400
600
800
1000
1200
1400
1600
1800
120
200
300
430.8
986.4
1556
1500
1000
SYSMAC-WAY(9600)
SYSMAC-WAY(19200)
Toolbus(9600)
Toolbus(19200)
Toolbus(115200)
ETH
CLK
SLK


CVM1-V2-CPU21 / CV500-V1-CPU21
Max. Num. of Reads per Second
0
200
400
600
800
1000
1200
68.56
138.58
1000
1000
1000
1000
SYSMAC-WAY(9600)
Toolbus(9600)
CLK
SLK
SYSMAC-NET
ETH


C200HX-CPU64
Max. Num. of Reads per Second
1200
1000
800
600
400
200
0
15.38
32.07
1000
1000
1000
SYSMAC-WAY(9600)
Toolbus(9600)
CLK
SLK
SYSMAC-NET


CQM1H-CPU51
Max. Num. of Reads per Second
1200
1000
800
600
400
200
0
70
180
1000
SYSMAC-WAY(9600)
Toolbus(9600)
CLK



See -> "How does the Operating System affect performance?"

See -> "What is the maximum limit of number of points?"

See also Chapter 16 Best Practices, Performance and Chapter 16 Best Practices, Points

Q How can I see if the communication network is overloaded?

Use "CX-Supervisor Performance Monitor Tool" to see if the network is overloaded

This tool is available in the CX-Supervisor run-time by right clicking and selecting **Performance…** (if security privileges allow). It can be used to see how much of the available bandwidth is in use. You can also see how many points are waiting to be processed and the latency of actual communications. To improve responsiveness try to reduce the bandwidth used, the latency and the number of active messages.

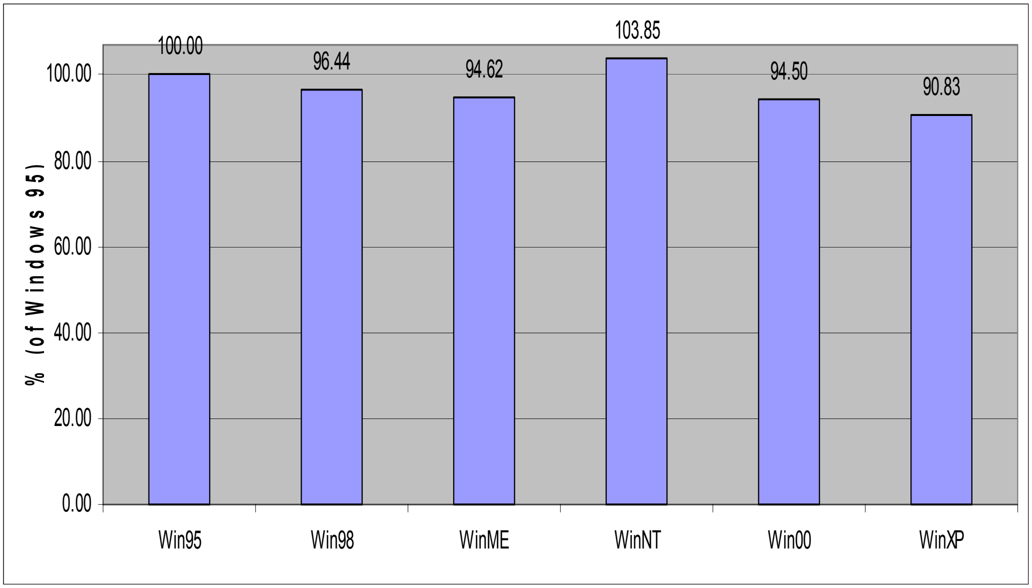
Note that performance begins to reduce well before 100% saturation is achieved. For example it is rare to achieve more than 70-80% for a serial connection. This is analogous to a motorway where cars slow down long before they are touching bumpers, and might only achieve a maximum of 50% capacity (i.e. each car has a car length space behind it).

Q How does the Operating System affect performance?

On the same specification machine Windows NT provides the best overall performance. It is understandable that newer, larger Operating Systems run slower on the same specification machine. This is not usually a problem as newer operating systems are usually delivered on newer specification machines.

FINS Gateway drivers for particular networks operate consistently across all operating systems.

OMRON recommend Windows 2000 and Windows XP Professional for the following reasons:
- Better security
- Better stability
- Better multitasking
- Better DCOM configuration

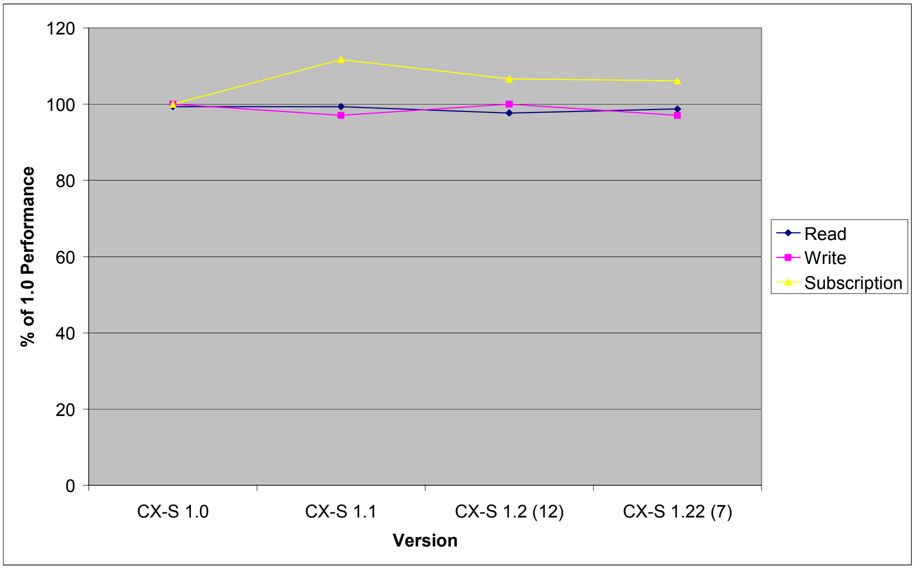


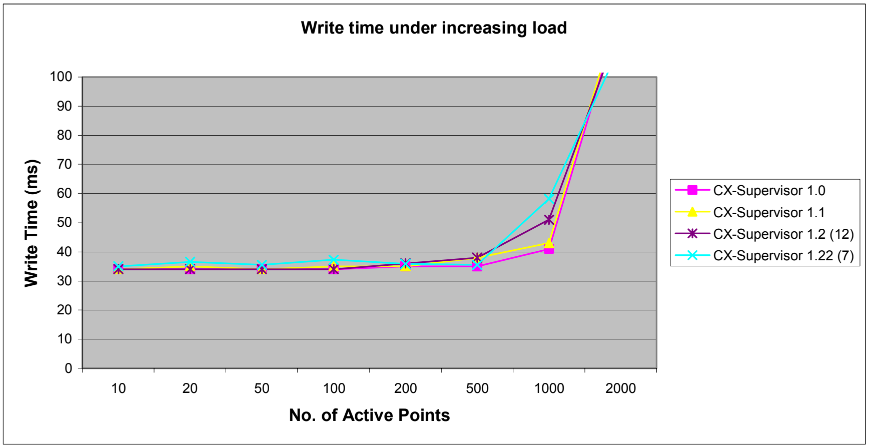
Q How does upgrading CX-Supervisor affect communications performance?

The graphs below show that this does not have an impact itself, which is not surprising as a communications functions, and therefore performance, are governed by the communication middleware: CX-Server. The graphs show:

- Upgrading CX-Supervisor does not affect the total number of Subscription points that can be simultaneously read, which has remained more than 1000.
- Performance of instantaneous read and write is not significantly affected by the version of CX-Supervisor.
- Updating CX-Supervisor does not affect how performance degrades when the system gets busy.

Note: CX-Supervisor 1.2 and later *do* include automatic optimisations for writing array elements that can yield substantial results, particularly with serial communications. These performance improvements have been excluded from these results rather than distort the true figures.





Note: lower write time indicates greater performance

See -> "How does upgrading CX-Server affect communications performance?"

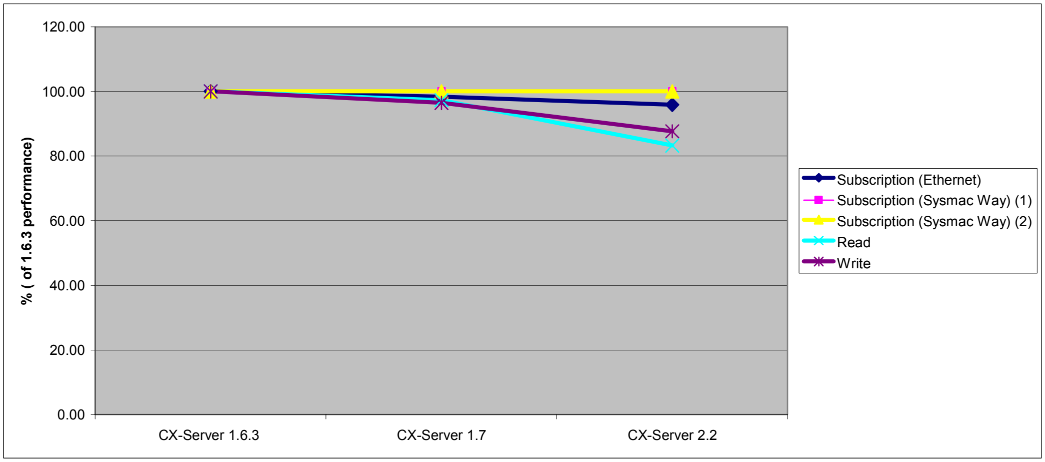
Q How does upgrading CX-Server affect communications performance?

The graphs below should show that this does not have a significant impact. The graphs show:

- Time to perform instantaneous read and writes are slightly slower on average, although this is in the order of a few milliseconds so is not ordinarily visible.
- Maximum number of Subscription points remains the same across all versions. This is confirmed with a variety of protocols

  See -> "How does my PC specification affect communications performance?"

  See -> "What is the maximum number of communication updates?"



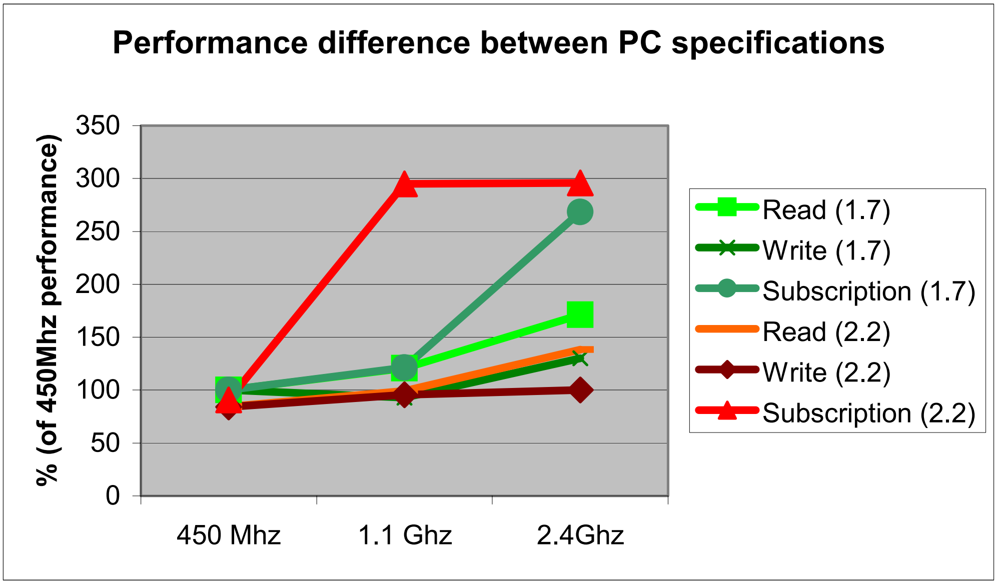
Q How does my PC specification affect communications performance?

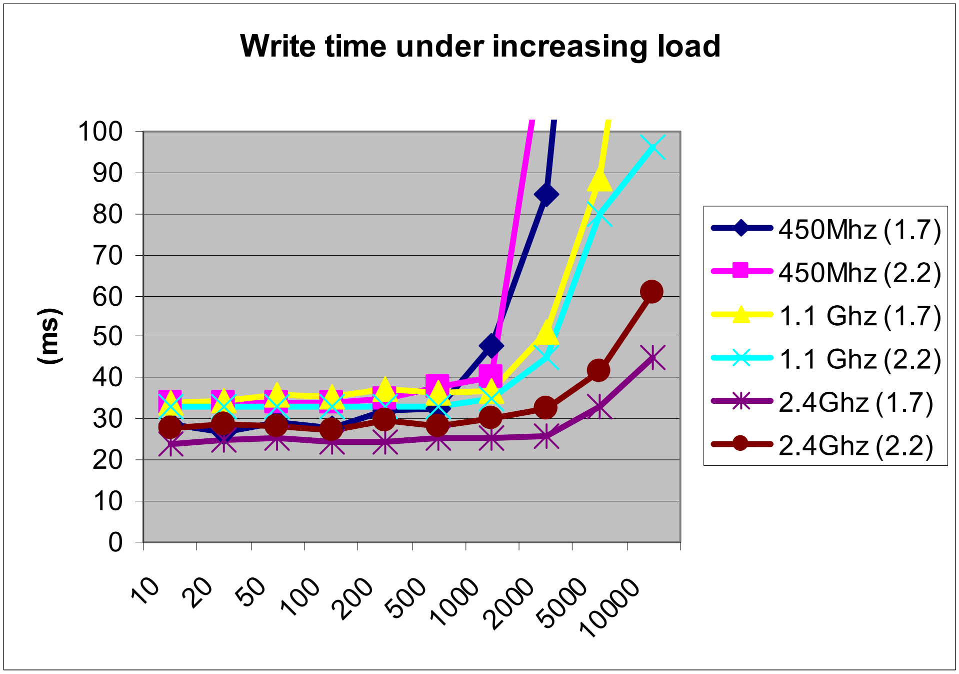
The graphs below show that processor speed has a significant impact on performance, especially for large systems with high quantities of On Interval Subscription points. The graphs show:

- Upgrading from a 450Mhz system to a 2.4Ghz can *quadruple* the maximum number of Subscriptions that can be processed

  See -> "What is the maximum number of communication updates?"

- Read, Write times and performance under load improve with processor speed
- CX-Server DI 2.2 performs in very similar manner to previous versions and better on a mid spec machine.


Performance difference between PC specifications
% (of 450Mhz performance)
350
300
250
200
150
100
50
0
450 Mhz
1.1 Ghz
2.4Ghz
Read (1.7)
Write (1.7)
Subscription (1.7)
Read (2.2)
Write (2.2)
Subscription (2.2)


Write time under increasing load
(ms)
100
90
80
70
60
50
40
30
20
10
0
10
20
50
100
200
500
1000
2000
5000
10000
450Mhz (1.7)
450Mhz (2.2)
1.1 Ghz (1.7)
1.1 Ghz (2.2)
2.4Ghz (1.7)
2.4Ghz (2.2)

Q Can I run CX-Supervisor Runtime as a restricted Windows User?

CX-Supervisor (and CX-Server) require Administrator rights to be installed on a computer, but other users can then run the software. The runtime will run without problems as a Standard User (member of Power Users group) and will run as a Restricted User (member of Users Group) although will show errors when trying to update the registry.

To avoid these errors specific registry access can be provided by an Administrator to the following registry keys:

- HKEY_CLASSES_ROOT\CLSID\{0002D780-0000-0000-C000-000000000046}
- HKEY_CLASSES_ROOT\CLSID\{0002D781-0000-0000-C000-000000000046}
- HKEY_CLASSES_ROOT\CX-Supervisor.Application
- HKEY_CLASSES_ROOT\CX-Supervisor.Project

Q Can I use CX-Simulator with CX-Supervisor?

Yes Omron’s CX-Simulator virtual PLC can be easily used with CX-Supervisor. To achieve this follow these steps:

1. Start CX-Simulator from the windows Start menu.
2. Select 'Create New PLC'
3. Select the required PLC Type
4. No need to register I/O ---> Next
5. Keep default Virtual Comms Unit ---> Next
6. Keep default Comm Port ---> Next
7. Create the data folder ---> Finish
8. In the work CX-Simulator dialogue, connect to the virtual PLC.
9. Start the virtual PLC. It must be started first for CX-Supervisor to be able to open the device.
10. In CX-Supervisor project, set the PLC type as per CX-Simulator or FinsGateway.
11. Set the communication type as per CX-Simulator setup e.g. default is Controller Link, Network 1 Node 10 Unit 0

When the CX-Supervisor project is run, communication with the virtual PLC will be established. All PLC aspects of CX-Supervisor will work with CX-Simulator as if it were a real PLC

Q Why do I get "Invalid Point" error?

If this occurs on startup of the developer, it can be due to a corrupt point definition. Usually, editing the Point and pressing OK is enough to correct the corruption. If there are too many points, select a large group and use the 'Multiple Edit' function to assign some common attribute to them all e.g. OLE Access is Read Only. This in effect edits each point, also correcting the corruption.

If this error occurs in the Runtime is means the point has been deleted or renamed but a script or animation is still using the Point. Run the Analyse Application tool to highlight and fix References to non-existent points.

Q How do I access "CX-Supervisor Local Points" from Omron Graphical components?

Omron ActiveX Graphical components, like the Thumbwheel shipped with CX-Supervisor or those shipped with CX-Server Lite and CX-Server OPC (7 Segment, Display, Knob Toggle etc) can be placed on CX-Supervisor pages and linked directly to CX-Supervisor points by selecting a data source of "CX-Supervisor Local Points".

However, note that the default setting for points is to allow reading, but to protect against writing from external sources like these controls. If this is attempted you will get the following error message "OLE Point <pointname> has no write access".

Note however, that as a low priority message the default project configuration will not cause the error log to automatically open and you will have to manually view the error log viewer.

To prevent this error and to allow write access change the 'OLE Access' to 'Read/Write' on the Advanced Point Settings dialog.

Note: Copy Protection must be installed for this function.

Q Which OPC Servers have been tested with CX-Supervisor?

The following third party OPC Server products have been tested with CX-Supervisor:

Vendor	Server Name	Product Version	Result
4CE Industry	OPCSysDiag	1.0	Pass
4CE Industry	Modbus Server	2.0.4	Pass
Alstom Technology	FBSOPC	1.30a	Pass (although noted not OPC compliant)
Amersham Biosciences	UNICORN	Zappa build 313	Pass
Applicom International	applicom IO	2.1 (reported as 2.1 (1000))	Pass
AXEDA Systems	OPC DA Server	1.0.0.2	Pass
CimQuest	OPC Server for Allen Bradley	2.25.00.01 (reported as 2.25 (1))	Pass
CimQuest	OPC Server for Omron	2.25.00.01	Pass
Cyberlogic Technologies, Inc.	MBX OPC Server	5.00.02	Pass (although noted not OPC compliant)
KEPware	KEPServerEX	4.12.135	Pass
KW Software GmbH	KW Software-ProConOS OPC Server	2.0	Pass
ICONICS Inc	Modbus OPC Server	3.07	Pass
ICONICS Inc	DataStore OPC Server 3.10	3.10	Pass
INAT GmbH	INAT OPC-Server TCPIPH1 S5 and S7	2.05.37	Pass
INAT GmbH	OPC-Server MPI/PPI	2.05.37	Pass
Invensys Systems / Wonderware	DASS7	1.1	Pass
Matrikon	OPC Server for Simulation & Test	1.1.3.230	Pass

Vendor	Server Name	Product Version	Result
National Instruments		4.5.3 (beta)	Pass
Northern Dynamic	OPC Server Toolkit	2.01	Pass
OSI Software Inc	OPC Data Access & Historical Data Access Server for the PI System	1.0.0.14	Pass (although noted not OPC compliant)
Phoenix Contact GmbH & Co. KG	Interbus OPC Server	2.12	Pass
Rockwell Software	RSLinx	2.30.00 (beta)	Pass
SMAR	OPC & Conf. Server for DFI302	3.3.0.0	Pass
SST	DH+OPC Server	1.6.16	Pass
SST	ControlNet OPC Server	1.3.0	Pass
Siemens AG	WinCC	5.0	Pass
Siemens AG	Simatic WinCC	V6.0 SP2 Beta	Pass (although noted not OPC compliant)
Siemens AG	Simatic Net OPC Servers	6.1	Pass
Siemens AG	IndustrialDataBridge	6.0.3007	Pass
Siemens Moore	APACS+OPC Device Server	1.00 K (beta)	Pass
Triconex Corp.	TS1131	3.1	Pass
Wonderware	InControl	7.11.1614	Pass
Wonderware	Virtual PLC Server	1.0 (beta)	Pass

The following products have not passed:

Vendor	Server Name	Product Version	Comment
Klinkmann	Omron driver	?	Ignores 'Read from Device' option. Seen to lose device values, potential control risk.
OPC Labs	Time Monitor	2.01	Fail. Group inactive, item active still results in a callback
Rockwell Software	RSView32	6.2	V1.0a Server only – not supported
Siemens AG	WinCC	4.1	V1.0a Server only – not supported. Contact Siemens to upgrade to V5.0
Siemens Building Technologies Inc	MK8000 MP1.30	MP1.30	Could not get the simplest of operations to succeed
Softing	OPC Toolkit	?	Some support issues reported
USDATA Corp	Factorylink	7.0.1 (beta)	Single test failed to connect.

Q What is the maximum limit of commands in a script?

There is no set limit, and some applications have had many hundreds of lines. However it is not practical to have too many lines and a limit of 30 lines should be considered, not as a system limit but as a warning that the application is not correct.

Q What is the maximum limit of ingredients in a recipe?

There is no limit other than the PC memory.

Q What is the maximum limit of number of entries in the Alarm Log?

The number of lines in the alarm log and the error log is set in the Project / Runtime Settings / Alarm and Event settings dialogs. The maximum limit is currently 2000.

Q What is the maximum limit of number of Alarms?

Theoretically, the maximum number of alarm types that can be defined is 10,000.

Q What is the maximum limit of number of serial ports supported?

It is the maximum that the Windows OS supports, and are correctly installed on the PC. We have successfully tested eight.

Q How do I create Reports and HTML reports?

CX-Supervisor provides a simple yet powerful report generation facility. Using the same 'Points Substitution' format, a text based template file may be used to generate a report. Good formats are .TXT, .RTF and .HTML. Binary formats like Word .DOC can be problematic although RTF can usually be used in these cases.

These generated files may then be viewed, or distributed using a file server or web server. Note that although information may be textual or graphical in nature these reports are intended for non real-time output of data only, and not for monitoring or control.

See documentation for "GenerateReport" script function for more information, and the "Coffee Tutorial" Step 12 in the CX-Supervisor Getting Started Manual.

See -> "How do I make an HTML report update and refresh automatically?"

Q How do I make an HTML report update and refresh automatically?

CX-Supervisor supports the generation of dynamic text and HTML reports from a static template - see "GenerateReport" script function for details.

However, Web Browsers views all pages as static pages. There is a very simple HTML trick that will force the browser to reload or "Refresh" the HTML report that would then show any updates, assuming GenerateReport had been called again e.g. On Condition when the data changes.

To an existing HTML page add the following line. This must be inserted between the `<head>` and `</head>` markers:

```
<meta http-equiv="refresh" content="5; url=Report.htm">
```

This will force the page to be reloaded, every 5 seconds. The value after the content= (e.g. "5") is the refresh rate in seconds. Be aware that 5 seconds may practically be the fastest update, as some pages may take several seconds to download. Also the value after the url= is the name of the page to refresh.

Q Can CX-Supervisor be used with Industrial Touchscreen PCs (IPCs)?

Absolutely. CX-Supervisor is designed for use with Omron's DyaloX Touchscreen IPC but can be used with other manufacturers. It has special features especially for Touchscreen PCs:

- Edit animations have option to enter data from a popup keyboard to allow typing direct on the screen.
- CX-Supervisor Login dialog has a popup keyboard.
- Project setting for 'Large dialogs' to increase font size on high resolution touchscreens giving a larger contact area.

Projects can be built on a development machine and deployed using "Create Runtime Install Disk" to copy required files to a shared drive or USB memory stick.

It is also commonplace to copy a shortcut to the SR2 application to the 'Startup' folder. If the keyboard is to be detached see also "How To Enable Automatic Login in Windows"

Q How do you enable automatic login in Windows?

The initial Windows security screen can be bypassed which may be useful on systems with no keyboards, or systems that should start automatically on boot up. Information on how to achieve this is published by Microsoft on its Web site - see Q253370 How to Enable Automatic Logon in Windows 2000 and Q315231 How to Enable Automatic Logon in Windows XP.

Please note the cautions included on registry editing and security.

Q What is the functionality of "On Condition" scripts?

ON CONDITION scripts will be executed whenever any of the points mentioned in the expression change value (or are re-evaluated as a result of being input) AND the result of the expression is TRUE.

For example, if the expression was $Second, then the script would be executed 59 times a minute. The expression would be evaluated every second when the value changes, but when $Second was 0, the result of the expression would be FALSE.

Q What is the scan time of "On Condition" scripts?

There is no scan rate directly associated with the script processing itself. In fact the execution is instant, as soon as the value has changed. The script expression will be checked whenever any of the points are updated (a value is input, or the value is changed in some way).

CX-Supervisor is an "event-driven, object-oriented" system meaning is that it is actually the points changing that force the script code to re-evaluate the expression. They do this whenever a point changes value, or is re-evaluated (e.g. a value is received from the PLC at the point's usual scan-rate). This removes any sampling error associated with polling and also makes CX-Supervisor very efficient.

Q How do I display Real Numbers and Decimal Places?

In graphics editor select text tool from palette and type text e.g.

Temperature = ##.### C

The # after the decimal point indicate the precision of the real number display, e.g. to 3 decimal places. Finally select animation editor and animate with "Display Value (Analogue)" and assign the point name e.g. TEMP

Q Which Operating system should I use?

Windows XP Professional.

Windows NT, Windows 2000 and Windows XP Professional all offer the best stability, reliability, multitasking, security and DCOM support.

On the same spec machine, Windows NT is actually up to 10% faster, probably as it has the least developed code although Windows 2000 and XP are faster on their appropriate minimum specification machines.

Windows 98 and Windows ME are currently tested although not recommended for industrial solutions.

Windows 95 is no longer supported.

Q How do I optimise the use of Files?

When closing a file using the 'CloseFile' script function, if TRUE is passed as a parameter e.g. CloseFile(TRUE) then the blank spaces at the end of each line will be removed, thus reducing the size of the file. Care should be taken if the file is being used by multiple systems over a network drive.

Q What is the limit of points in a group?

There is no imposed limit but it is recommended to limit about 1000 points per group. There are two main reasons for this:-

- There is a physical limit on how many lines of text the actual list can display. This is dependent on the height of the text. With large text you may even find that 1000 is too many.
- When a group is selected the point list is refreshed and re-sorted. The sort algorithm takes longer if it has a large list to sort. This is normally quick but if the list is large and the points are already in the correctly sorted order in the database, then it can take a while because of the way the sorting algorithm works.

Q Why does setting slider value result in erratic movement of slider action?

The movement of the slider bar can appear to return to its previous setting once it has been re-set. After setting the slider to a value, it will read the actual value from the PLC, this value is then shown in the slider, resulting in the slider moving from the re-set value to the old value, the slider then starts to move towards the re-set value. This actually better reflects the current value in the PLC, but can be undesirable.

In this case de-select the "Immediate update on slide move" option in the slider wizard to prevent it.

Q How do optimise my use of animations?

Animation actions (from the Animation Editor) can be applied directly to the objects being animated. This means that you don't have to define lots of memory points to hold intermediate values. Just link the appropriate action to the object (using the Animation Editor) and enter the required expression. For example using expression “TemperatureK + 273” rather than creating a new “TemperatureCelsius” memory point.

Boolean expressions can easily written to operate on bits within a word e.g. nFlags & 0x02 will be TRUE if bit 2 is set or FALSE if it not. This eliminates the need for additional memory points and processing.

Q Can I use token disks with USB Floppy drives or in high capacity LS120 Disk Drives?

The token copy protection is no longer supported. The software tokens have proven to be incompatible with some USB floppy drives and the high capacity LS120 Disk Drives. If you need to remove a software token use the network to conduct the token transfer, by sharing a floppy drive from another machine.

Q Can I print accented characters on alarm printers?

There is a limitation with Windows, such that it is not possible to print accented characters to a line printer a line at a time. The only alternatives are:

- To print using English only characters
- To set the printing to page mode, which will print a page when a specified number of alarms have occurred. This can be set up under Runtime settings, Alarm/Message printer settings.

Q Why do I get Installation error -115?

Error -115 during installation means the specified file could not be copied. This is most often caused by the file already being in use or locked. Reboot the machine, and attempt installing before running any Omron software.

This error can also appear when attempting to install on a operating system that does not support extended European characters, e.g. Japanese or Chinese Windows. During installation, on the 'Select Components' dialog, select 'CX-Supervisor package' and click 'Change...'. Scroll to the bottom of the list of sub-components, and deselect 'Language Files'. This will prevent language files, including those using extended European characters being installed.

Q How do I address extended memory banks in PLCs?

In the point editor where you configure the PLC address, enter a hyphen and the bank number for example:

E32000-3

Or (depending on PLC type)

EM32000-3

This will address memory location 32000 in bank 3.

Q Why can't I see all my installed controls in the Insert Object dialog?

Under Windows 2000 and Windows XP the Insert Object dialog is not able to see all the controls installed on the PC. These include the MS Forms 2.0 controls that are installed as part of MS Office.

This is by Microsoft's design for these operating systems, however it is possible to force the old operation, depending on which controls you need to expose for use in CX-Supervisor. Contact your supplier to receive details of the required Registry changes.

Q How does Sleep() work?

The CX-Supervisor Sleep function is designed to create a short pause in script execution. The parameter specifies the duration of the pause in milliseconds. For example:

```
<statements A>

<statements B>

Sleep(100)

<statements C>
```

In this example 'statements B' will begin executing immediately when 'statements A' complete, though 'statements C' will not begin execution until 100ms after 'statements B' have completed.

Note 1: In a well designed, truly event driven system use of the Sleep() statement should never be required. Always consider if the statements after the Sleep should be in their own script, executed when a Condition occurs.

Note 2: The Granularity (or intervals) differs between Operating Systems. In Windows NT (and 2000) expiration is checked every 10ms, so 'Sleep(100)' actually pauses for 100 to 109.99 milliseconds depending on when it was started. For Windows 98 and ME the granularity is 55ms so 'Sleep(100)' actually pauses for 110 (2 times 55) to 164.99 milliseconds (nearly 3 times 55). For this reason, Sleep statements can act differently on different Operating Systems making the application OS dependant.

Note 3: Sleep should never be used as a delay for timing processes, for the following reasons:

- The actual time delay depends on the OS
- There is always an error of 0 to 1 granularity, depending on when the action is started.
- The frequency cannot be guaranteed as the OS may be busy, or handling other processes.

Note 4: Because of Note 2 and Note 3 above, Sleep should not be used for creating output or logic pulses e.g. setting a bit on, sleeping, then setting the bit off. Instead, just use the PC to set the bit and allow the PLC to clear the bit, either after a duration timed by the PLC real time clock or when the triggered process is complete.

The requirement for Sleep is quite simple but the implementation is actually very complex. In the example above, the application must continue to run during the pause i.e. read PLC data, log data, check alarms, animate graphics etc. For this reason, any script with a Sleep statement must be executed in parallel with the system. This can result in unexpected results when a Sleep is added, and previously functioning script commands are now running in parallel. Redesign the script considering parallel processing.

This can also result in unexpected behaviour when a Sleep is added to a Project initialisation or Page initialisation script as some parts of the system or objects may not be loaded at time of execution.

Sleep can cause problems during shutdown of the runtime, if a sleeping script is still running.

Q How do I use System points in Visual Basic 'VBScript'?

Access CX-Supervisor points from Visual Basic style scripting called VBScript couldn't be simpler – just use the point name normally, e.g. MsgBox(MyPoint).

However note that '$' is a reserved character so System Points can be accessed with 's_' prefix instead e.g. MsgBox(s_Second) for $Second system point.

For up to date information and all help on scripting and the Windows Scripting Host see http:\\www.microsoft.com/scripting

Q What are the syntax differences between VBScript and CX-Supervisor script?

VBScript has some minor syntax differences to CX-Supervisor script. The most important are:

- Boolean equality test == is only = in VBScript.
- Boolean inequality test != is <> in VBScript.
- Endif becomes End If in VBScript.
- Array indexing using ( ) uses square brackets in VBScript [ ]

Note that CX-Supervisor interprets { and } characters as string delimiters, even in VBScript so the following code does not work:

```
MsgBox("Press {Enter}")          'does not work
```

Instead use the following to replace these characters:

```
MsgBox("Press " + chr(123) + "Enter" + chr(125))
```

Q Why are ActiveX events shown differently in the Animation Editor and Property Browser?

New in CX-Supervisor version 1.2 the ActiveX Property Browser has an Events Tab, which allows the object's events to be linked to scripts. These scripts are defined as VBScript subroutines defined in the Page Initialisation script.

It is more efficient to load these subroutines once during page initialisation, than every time the event occurs as per previous versions using the Animation Editor. However, this method is still supported to allow editing of old projects. This difference is by design and is correct.

Q Why does CX-Supervisor show different ActiveX properties to Visual Basic or Excel?

Properties of ActiveX controls shown in the CX-Supervisor "ActiveX Property Browser" may differ from those seen in the Visual Basic or Excel property browser. This can be for one or more of the following reasons:

- Both Visual Basic and Excel show their own container properties in this window, as well as the object properties. CX-Supervisor does not show its container properties. Examples of VBA container properties are:
  - CausesValidation
  - DataSource
  - DragIcon
  - DragMode
  - Height
  - HelpContextID
  - Index
  - Left
  - TabIndex
  - TabStop
  - Tag
  - ToolTipText
  - Top
  - Visible
  - WhatsThisHelpID
  - Width
- CX-Supervisor only shows properties with data types it can edit for example Boolean, Integer, Real, String, UserDefined, and Colours. Other data types i.e. Font type cannot be edited so are not displayed. Also properties with multiple parameters are not listed in CX-Supervisor as they cannot be edited.
- CX-Supervisor will show additional private object properties, which Visual Basic and Excel choose not to display.

Q Is it possible to connect an external PLC to CX-Supervisor via a modem?

Yes, you can connect from CX-Supervisor via CX-Server to a PLC via a modem. In the PLC configuration 'Network Settings' dialog, select the 'Modem' tab. CX-Server uses standard TAPI build into the OS. The modem must be installed properly to be shown in the Modem list.

Q What is the CX-Supervisor and SYSMAC-SCS Tokens compatibility?

CX-Supervisor 2.0 does not operate with old software tokens from older versions. You need to purchase a version 2.0 USB Dongle. Older product versions are illustrated below:

Product	**SCS V1.0 & V1.1** **(CC V1.67)** **Token**	**SCS V2.0 - V2.2** **(CC V1.72)** **Token**	**CX-Supervisor V1.0** **(CC V1.72)** **Token**	**CX-Supervisor V1.1** **(CC V1.72)** **Token**	**CX-Supervisor V1.2** **(CC V1.72)** **Token**	**CX-Supervisor V1.3** **(CC V1.72)** **Token**
SYSMAC-SCS 1.0	Yes					
SYSMAC-SCS 1.1	Yes					
SYSMAC-SCS 2.0	Yes	Yes				
SYSMAC-SCS 2.1	Yes	Yes				
SYSMAC-SCS 2.2	Yes	Yes				
CX-Supervisor V1.0	Yes	Yes	Yes	Yes	Yes	Yes
CX-Supervisor V1.1	Yes	Yes	Yes	Yes	Yes	Yes
CX-Supervisor V1.2	Yes	Yes	Yes	Yes	Yes	Yes
CX-Supervisor V1.3	Yes	Yes	Yes	Yes	Yes	Yes
CX-Supervisor V2.0	**No**	**No**	**No**	**No**	**No**	**No**

Different versions of SYSMAC-SCS and CX-Supervisor have used different versions of the copy protection software.

The Product versions listed above vertically, will work with the token versions listed horizontally which have a ‘Yes’. If using CCMOVE, the token disk, and the token being moved must all be the same version. Doing it from a native DOS prompt is recommended.

Q What is the CX-Supervisor and SYSMAC-SCS Dongles compatibility?

CX-Supervisor 2.0 does not operate with USB Dongles from version 1.3 or old parallel port dongles from older versions. You need to purchase a version 2.0 USB Dongle. Older product versions are illustrated below:

Product	**Hardlock Parallel port dongle**	**CX-Supervisor 1.3 USB Dongle**	**CX-Supervisor 2.0 USB Dongle**	**CX-Supervisor 2.0 USB Dongle**
CX-Supervisor V1.2 and ealier	Yes	N/A	N/A	N/A
CX-Supervisor V1.3	Yes	Yes	N/A	N/A
CX-Supervisor V2.0	**No**	**No**	**Yes**	**Yes**

Q What is the maximum limit of number of PLCs or Devices?

Machine edition projects are capable of communication with up to 15 devices through CX-Server, and the PLUS projects with up to 256 devices.

Q What is the maximum limit of size of array points?

Each array point may contain up to 1024 elements.

Q What is the maximum limit of number of pages?

Machine edition projects can configure up to 100 pages, and the PLUS projects with up to 500.

Q What is the maximum limit of number of objects on a page?

There is no theoretical limit to the number of objects on a page, but 1,000 should be considered the limit for normal PC configurations. Since each object consumes resources, Operating Systems like Windows NT, Windows 2000 and Windows XP will support larger applications than Win98 or ME.

Q What is the maximum limit of number of scripts on a page?

There is no theoretical limit to the number of scripts associated with a page, but performance of the application will be reduced the more scripts there are.

Q What is the maximum limit of number of recipes?

There is no theoretical limit to the number of recipes, except limits imposed by the size of memory of the PC. In practice, no more than 10,000 should be created on a standard PC configuration.

Q What is the maximum limit of number of event/error log entries?

The maximum number of entries in the event/error log is 500.

Q What is the maximum limit of number of Users?

The maximum number of Users that can be configured is 500.

Q Why do I get the message Failed To Install MSVCRT.DLL: Access Denied error message?

During installation, the following error can occur:

```
Failed to install MSVCRT.DLL:
Access Denied
```

This file is locked by the system so must be configured to install on the next reboot. However, the current user priviledges prevent access to the system registry to do this. This problem has only been seen on Windows NT

To correct this problem and allow full installation, you must log on to your computer with administrator rights. Either log onto your network as the network Administrator, or log on to your local domain as Administrator (or user with local administrator privileges). See your Systems Administrator for further details.

Q Why do I get the message ODBC DriverError - : 'The query is not updateable because it contains no searchable columns to use as a hopeful key'?

When attempting to pass information to an Excel spreadsheet, it the following ODBC Driver Error can occur: 'The query is not updateable because it contains no searchable columns to use as a hopeful key'. The Excel ODBC driver cannot correctly interpret the data type of a column, if the column in the spreadsheet contains both text and numbers.

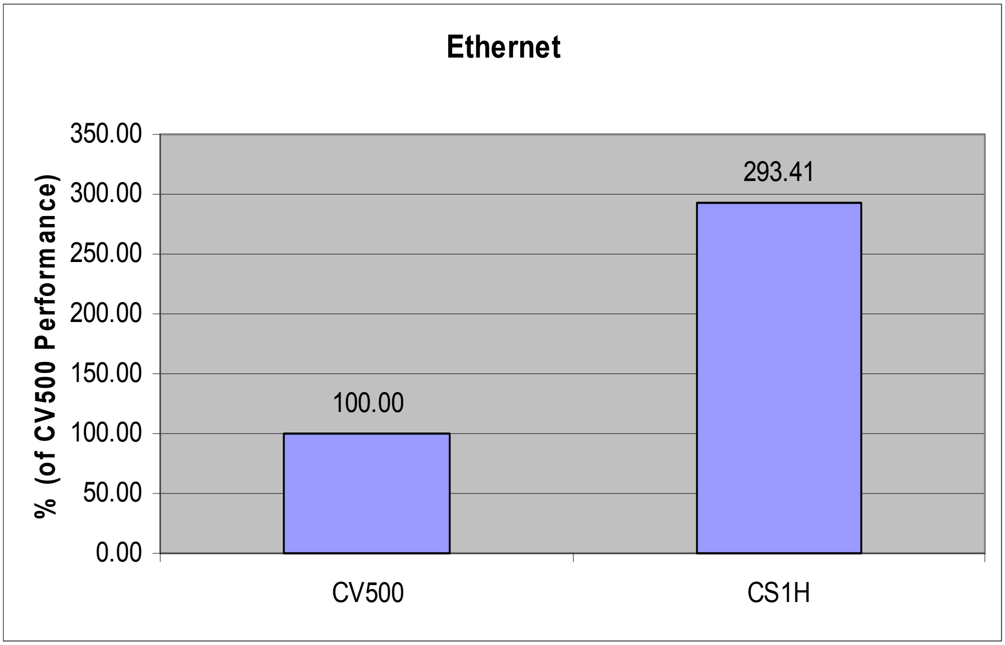
If a spreadsheet has been set up with data directly inputted in Excel and an attempting to write new data to the spreadsheet from CX-Supervisor this error will occur. This is because the ODBC Driver sees different formats in a column.

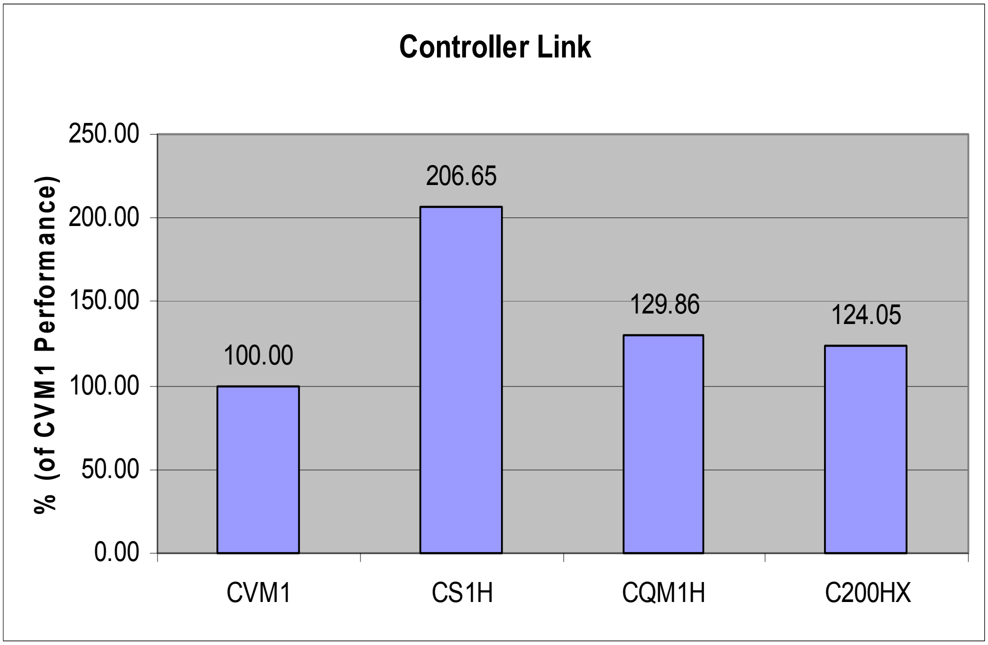
To overcome this set up the spreadsheet the data area definition to contain only the column headings (Insert | Name | Define…).

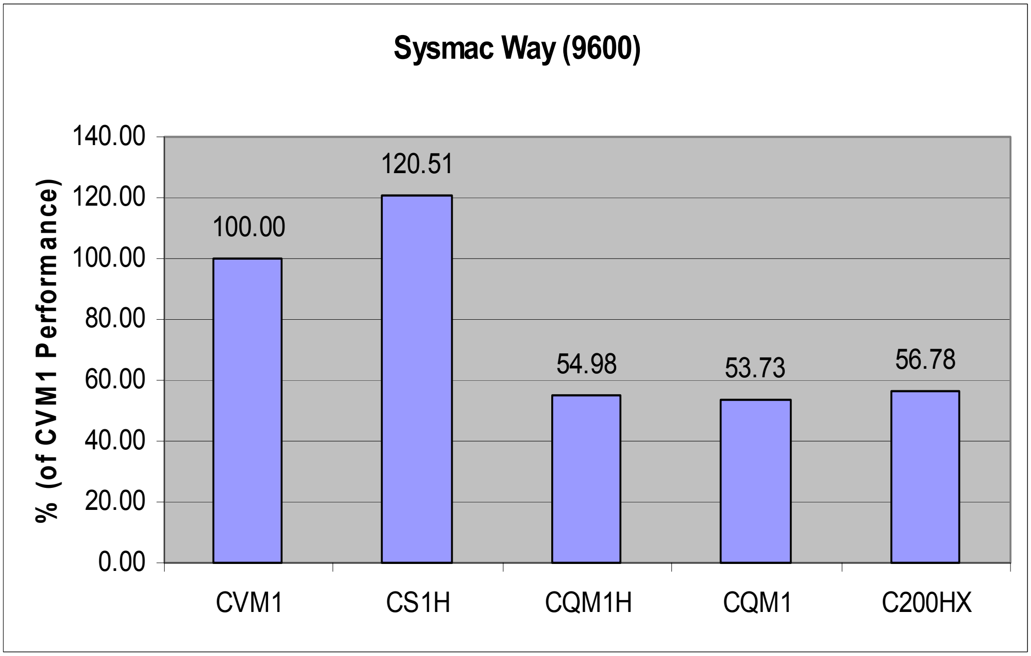
Another problem in this area has been that the format of the data can change from ‘Number’ to ‘Text’ (i.e. ‘333) formats in the cell of the spreadsheet. The exact cause of this error is unknown, but can be resolved by clearing the cell formats (Edit | Clear | Formats) of the spreadsheet that will be receiving the data.

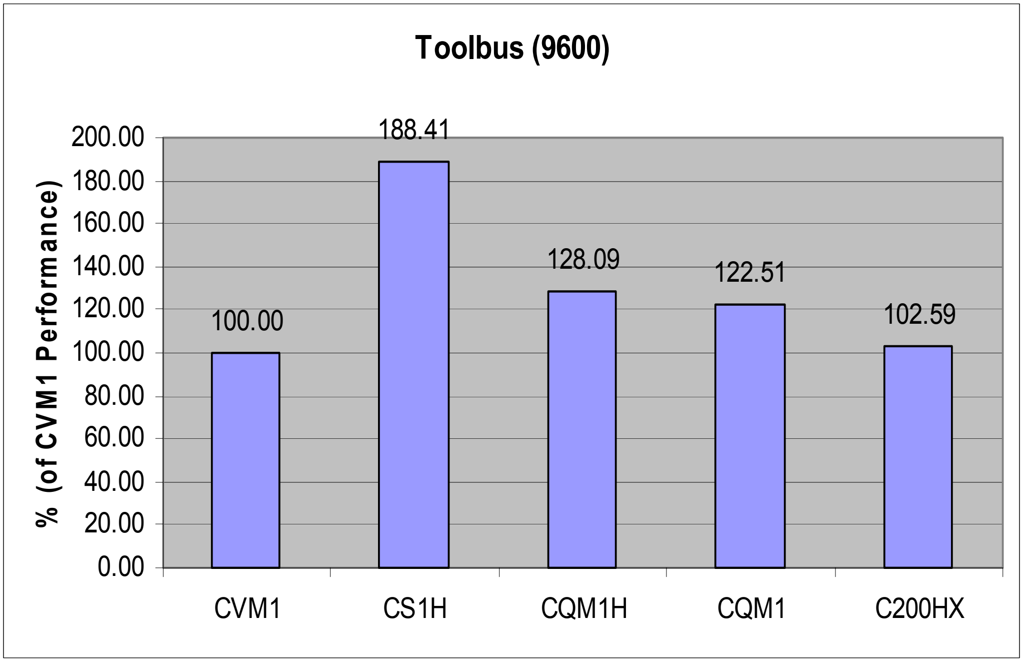
Q How does the PLC affect performance for my network type?

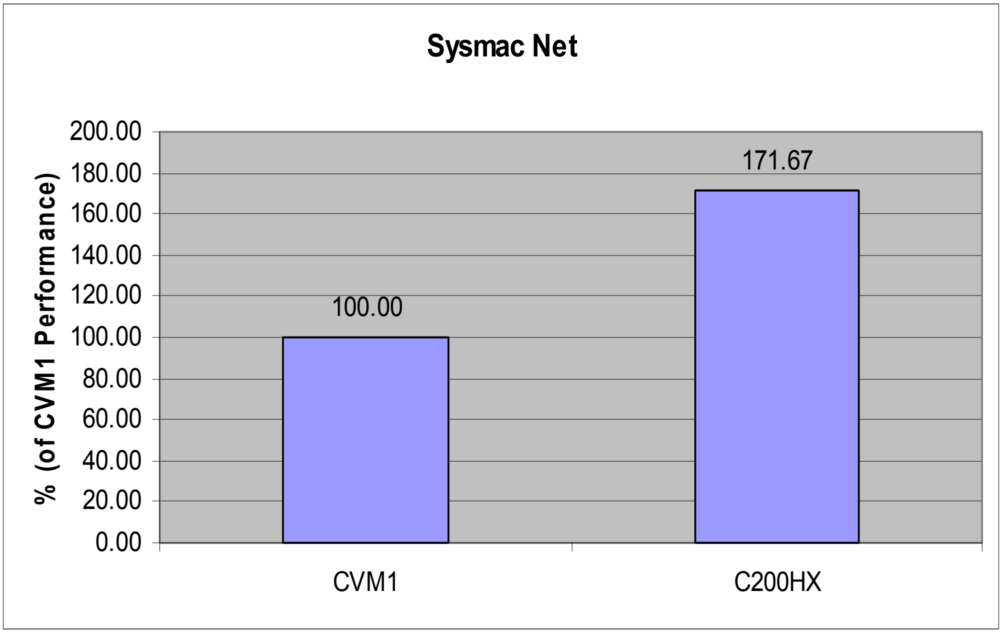
The charts below show how the network type affect the performance depending on the PLC used.

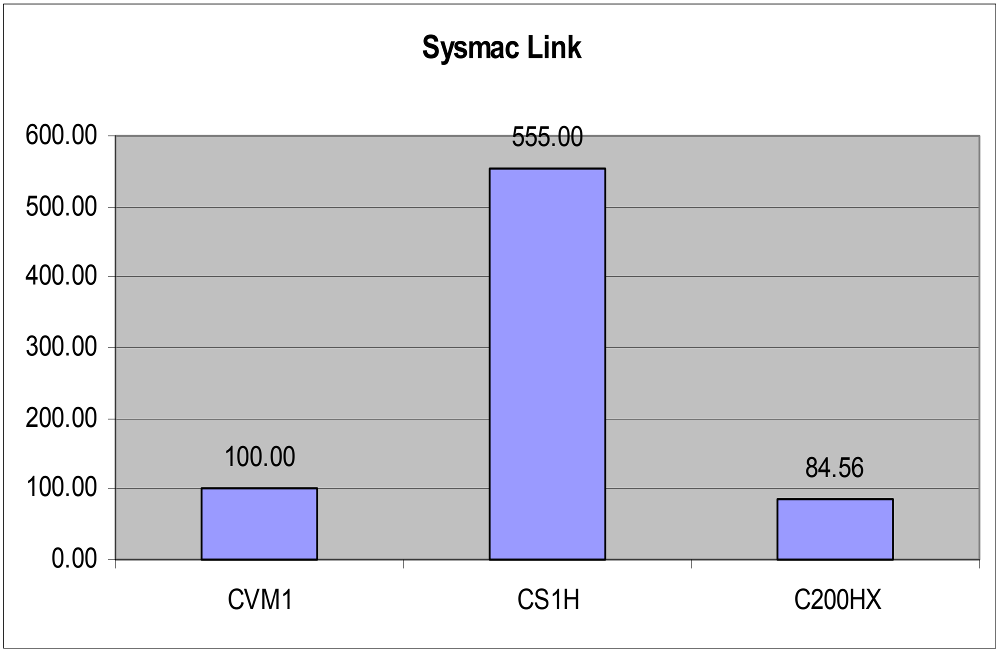



Controller Link
% (of CVM1 Performance)
250.00
200.00
150.00
100.00
50.00
0.00
100.00
206.65
129.86
124.05
CVM1
CS1H
CQM1H
C200HX


Sysmac Way (9600)
% (of CVM1 Performance)
140.00
120.00
100.00
80.00
60.00
40.00
20.00
0.00
100.00
120.51
54.98
53.73
56.78
CVM1
CS1H
CQM1H
CQM1
C200HX


Toolbus (9600)
% (of CVM1 Performance)
200.00
180.00
160.00
140.00
120.00
100.00
80.00
60.00
40.00
20.00
0.00
100.00
188.41
128.09
122.51
102.59
CVM1
CS1H
CQM1H
CQM1
C200HX


Sysmac Net
% (of CVM1 Performance)
200.00
180.00
160.00
140.00
120.00
100.00
80.00
60.00
40.00
20.00
0.00
100.00
171.67
CVM1
C200HX


Sysmac Link
600.00
500.00
400.00
300.00
200.00
100.00
0.00
100.00
555.00
84.56
CVM1
CS1H
C200HX

APPENDIX C
Troubleshooting

The appendix is a guide to troubleshooting and resolving problems. Several tools and techniques are discussed to assist troubleshooting, and in the case of no success, the information required to request support.

How to diagnose problems

Here are some tips to help diagnose problems:

- Where you think a problem exists try to create steps to reproduce it (see below)
- Be scientific:
  – Decide what the problem *could* be, and perform a test to prove or disprove the theory
  – Perform one test at a time
- Check the data values are correct in the PLC using the PLC Data Monitor, and are correct in CX-Supervisor using the Point Maintenance tool
- Prove that scripts are executed by popping up message boxes
- For speed or responsiveness issues use the Performance Monitor to check the system is not overloaded

Point Maintenance

The Point Maintenance dialog is very useful during troubleshooting for displaying, and setting point values. Launch by right clicking the runtime and selecting Point Maintenance, providing that you have sufficient security.

When an expected animation or script execution has not occurred, the Point Maintenance can quickly show if the data is not the expected value. If the data is the correct value then the fault must lay with the animation or script.

Note: You can not set the value of points defined as Input only. In this case use the PLC Data Monitor.

PLC Data Monitor

The PLC Data Monitor is similar to the Point Maintenance dialog, except that it is a CX-Server tool that operates at a lower level on the PLC. It can be used to read and write logical and physical addresses, and using a variety of PLC data formats. It is launched by right clicking the runtime and selecting Communication Utilities|PLC Data Monitor

Most useful is the Address monitor: select the Address tab and double click Monitor. Any CX-Supervisor PLC ‘Symbol’ (point) can be selected and monitored and set, including Input only points.

CX-Supervisor Performance monitor

The CX-Supervisor Performance Monitor tool displays application and communication information and can be very helpful in diagnosing performance problems. It is launched by right clicking the runtime and selecting **Performance…** (if security privileges permit). There are statistics and a performance trend for each PLC, each network, the CPU time and a Summary and Overview page.

Overview

Shows an overview of the headline performance of all other components.

Summary

Provides a Summary view, totalising and averaging property values from all PLCs on all networks.

CPU Time

Lists specific CPU and process information including processing times for all scripts and all callbacks.

Network

Totalises properties for all PLCs on each network.

PLC

The ‘PLC Average Latency (ms)’ field shows the physical delay of the network and PLC response time. Depending on the PLC, setup and network this should be 9-30ms. If this is drastically higher it could be the cause of performance problems. Try some of the following steps to reduce it:

The ‘PLC Average Latency (ms)’ field shows the physical delay of the network and PLC response time. Depending on the PLC, setup and network this should be 9-30ms. If this is drastically higher it could be the cause of performance problems. Try some of the following steps to reduce it:

- Test on a dedicated network, or if not possible with other nodes disabled. This is to ensure it is not due to network loading or other external factors.
- Try with the PLC in ‘STOP’ mode. If this has an impact double-check the PLC settings. This can vary depending on PLC type, but some require a longer scan rate, so more free CPU time is available to service the communications, whereas some require a shorter scan rate so the communications are service more frequently at the end of each scan. Some other settings may also impact the CPUs ability to service communications.
- Try to reduce the number of active messages (see below)

- Try to reduce to % usage (see below)

On the Performance tab, the quantity of ‘Active Messages’ is shown. Each Active Message is a single communication request although the internal optimisations mean that many continuously addressed points can be read in one message. This depends on the frame size, which in turn depends on the network type. This is why use of arrays and good memory layout are essential to performance. To reduce the Active Messages see Chapter 16 Best Practices, Performance and Chapter 16 Best Practices, Points.

On the Performance tab, the current and historic usage is shown. 100% usage is rarely seen and the system may be running at capacity well before this. This is analogous to a motorway where cars slow down long before they are touching bumpers, and might only achieve 50% of capacity (each car has a car length space behind it). In practical terms, for serial connections consider 70-80% the limit. For Ethernet packet collisions start occurring above 30% and are automatically corrected, but 40-50% is the practical limit. For Controller Link, which has a vast bandwidth, any value above 10% signals a performance issue. To reduce the % usage see Chapter 16 Best Practices, Performance and Chapter 16 Best Practices, Points.

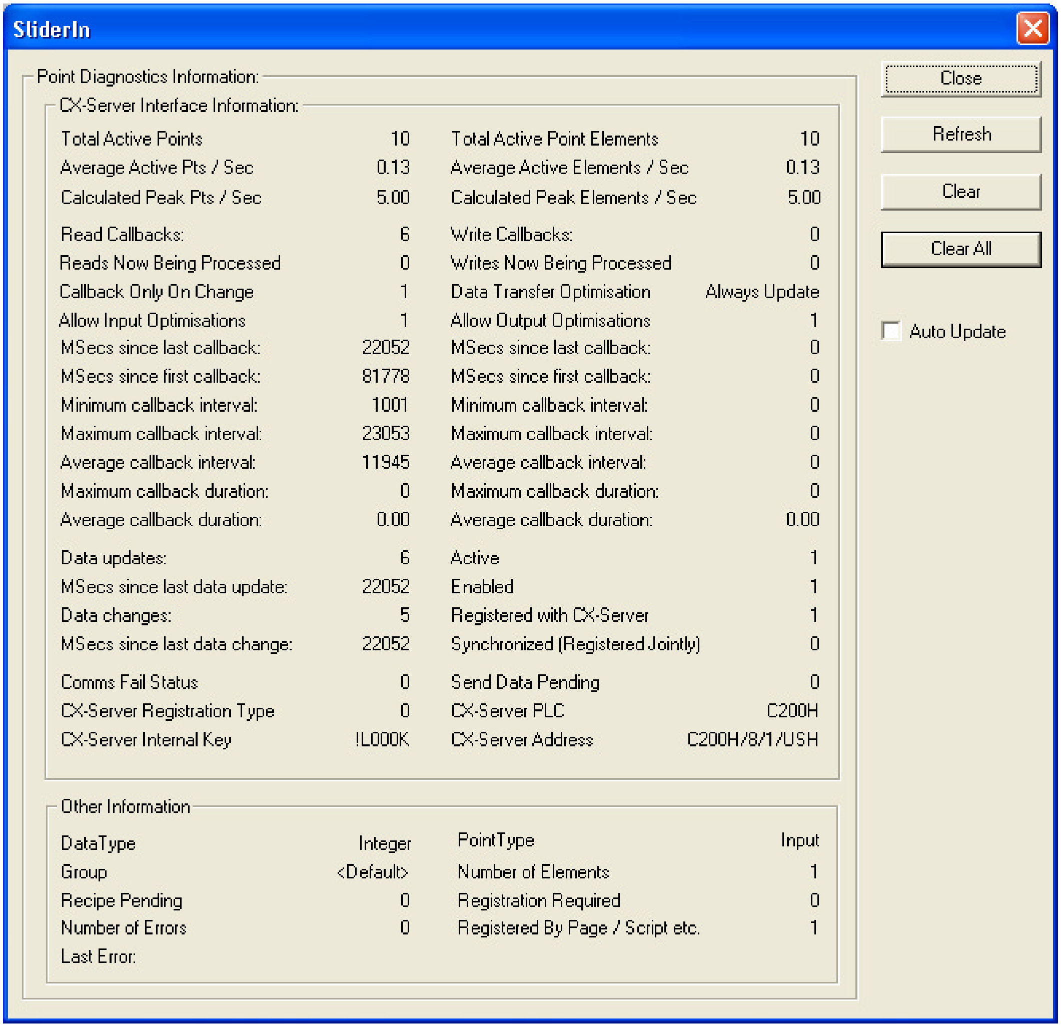
Diagnostics dialog

The Runtime has a communications diagnostics window. This will only normally be used under guidance of Technical Support to assist diagnosing specific communication issues. To view the dialog:

a) Log in as a user with ‘Designer’ privileges

b) Open the Point Maintenance dialog and select the PLC point to diagnose

c) Press ‘Diagnostics…’. The following screen is shown:



□ Auto Update

Point Diagnostics Information:

CX-Server Interface Information:

Total Active Points	10	Total Active Point Elements	10
Average Active Pts / Sec	0.13	Average Active Elements / Sec	0.13
Calculated Peak Pts / Sec	5.00	Calculated Peak Elements / Sec	5.00
Read Callbacks:	6	Write Callbacks:	0
Reads Now Being Processed	0	Writes Now Being Processed	0
Callback Only On Change	1	Data Transfer Optimisation	Always Update
Allow Input Optimisations	1	Allow Output Optimisations	1
MSecs since last callback:	22052	MSecs since last callback:	0
MSecs since first callback:	81778	MSecs since first callback:	0
Minimum callback interval:	1001	Minimum callback interval:	0
Maximum callback interval:	23053	Maximum callback interval:	0
Average callback interval:	11945	Average callback interval:	0
Maximum callback duration:	0	Maximum callback duration:	0
Average callback duration:	0.00	Average callback duration:	0.00
Data updates:	6	Active	1
MSecs since last data update:	22052	Enabled	1
Data changes:	5	Registered with CX-Server	1
MSecs since last data change:	22052	Synchronized (Registered Jointly)	0
Comms Fail Status	0	Send Data Pending	0
CX-Server Registration Type	0	CX-Server PLC	C200H
CX-Server Internal Key	!L000K	CX-Server Address	C200H/8/1/USH

Other Information

DataType	Integer	PointType	Input
Group	<Default>	Number of Elements	1
Recipe Pending	0	Registration Required	0
Number of Errors	0	Registered By Page / Script etc.	1
Last Error:			

The most useful readings on this screen are:

- "Auto Update" refreshes the data twice a second.
- Total Active Points: Number of CX-Server points (both input and output) *currently* active.
- Average Active Pts/Sec: Measured number of CX-Server Inputs per second, since application started (or since "Clear" pressed). With 'Input On Change' optimisation this can be very small. Also $InputsActual System Point
- Calculated Peak Pts/Sec: Calculated number of CX-Server Input points currently active, multiplied by update rate.

- '…Elements': Same as above except multiplied by number of array elements per point. Ratio between Total Active Points and Total Active Points Elements shows how well optimised (or badly like above) with use of arrays.
- Read Callbacks: Number of data postings from CX-Server.
- MSecs since last callback: Milliseconds waited since last data.
- Min, Max Average Callback interval: Calculation using above values.
- Write …: Same as above but for the writing part of output (and I/O) points.
- Data Changes: Number of inputs causing change of value.

Scripting errors

The scripting engines are very complex which provides many opportunities for errors. This can be categorised as follows:

VBScript Syntax errors

These are seen during runtime as errors in the Error Log, often giving the line and character of the error. Referring back to the original script should easily display the problem. Note that if the @VBScript syntax is used within CX-Supervisor script the line number reported refers just to the lines of VBScript, with the @VBScript being line 1.

To troubleshoot further:

- For specific help on VBScript language syntax see Microsoft's web site at http://msdn.microsoft.com/scripting
- Double check capitalisation, and objects names
- See Appendix B FAQ, "What are the syntax differences between VBScript and CX-Supervisor script?"
- To confirm the line causing the problem, temporarily comment it out and see if the syntax error is no longer reported
- As a test, try to simplify complex script lines into 1 or more lines with single instruction to identify which instruction is causing the error
- As a test, simplify complex math formulae into several lines with single operation to identify which operation is causing the error
- As a test, use hardcoded values for parameters using known valid values
- Display values using 'MsgBox' command or using the LogEvent command
- Confirm path of execution using 'MsgBox' in all conditional branches

CX-Supervisor Syntax errors

These are seen during development when OK is pressed on the Script dialog when the script type is ‘CX-Supervisor script’.

To troubleshoot further:

- Refer to the comprehensive CX-Supervisor Script Reference manual or on-line help for detailed information including examples on specific script instructions
- Double check capitalisation, and objects names
- To confirm the line causing the problem, temporarily comment it out and see if the syntax error is no longer reported
- As a test, try to simplify complex script lines into 1 or more lines with single instruction to identify which instruction is causing the error
- As a test, simplify complex math formulae into several lines with single operation to identify which operation is causing the error
- As a test, use hardcoded values for parameters using known valid values

Runtime errors

If any script has correct syntax but generates an error at runtime e.g. trying to open a file that does not exist, the error is recorded in the Error Log.

To troubleshoot further:

- To confirm the line causing the problem, temporarily comment it out and see if the error is no longer reported.
- Display values using ‘MsgBox’/’Message’ command or using the LogEvent command.

Design errors

The only remaining errors are when the script compiles and runs correctly, but do not produce the expected results. These are typically errors in the design of the script.

To troubleshoot further:

- Confirm path of execution using ‘MsgBox’/’Message’ in all conditional branches.
- Display values using ‘MsgBox’/’Message’ command.

PLC Maintenance dialog

The PLC Maintenance dialog is launched by right clicking and selecting PLC Maintenance. After choosing the PLC you can:

- Open and close the PLC
- See if there are errors
- Change the PLC Mode
- See the effect of optimisations as points are 'Disabled'

Database errors

The database connectivity can sometimes be problematic to configure correctly. To troubleshoot CX-Supervisor database functionality:

- Try viewing and writing the data from another data source, for example Microsoft Access application for mdb files etc.
- 'Connect' to the database from the Developer to check Table and Field names can be correctly read. Remember to disconnect before running if the provider doesn't support multiuser connections (like the Excel provider)
- Note that every database function returns an error code to confirm if the operation was successful. Always assign this to a temporary point and after the call to check the value
- When database errors occur, try calling DBGetLastError(). This will give a textual description of the problem. As these errors are standard there is often information about the cause of these errors available on the Internet – just search for the exact text of the error

How to create steps to reproduce

The first step in troubleshooting is creating steps to reproduce the problem. Not only does this help understand the problem but is a requirement so that your potential fixes can be tested with confidence. Computers and software are guaranteed to perform the same way with the given data and so there are always steps to reproduce although sometimes these can be difficult to find. It can be timing dependant ("you have to do this exactly 1.342 seconds later"!) or just very difficult to see a pattern. However in most cases problems are quite repeatable.

Once a problem can be reproduced, the next step is to refine the steps to determine the exact cause. The process should be repeated, attempting to leave out one step at a time to confirm if that step is required. Where scripts are executed, each line should be removed to determine just the required steps to reproduce the problem.

Information necessary to send to Support

If all these diagnostic suggestions fail and it becomes necessary to contact support please have the following information:

- Your software licence key
- The software version and build number e.g. 1.2 (12). Also state if the project was upgraded from a previous version
- The EXACT wording and spelling of any error message
- If an application is being sent it should be possible to remove unnecessary pages, objects, scripts, points, alarms, recipes, datalogging and database connections
- Clear description of the function that the Manual says the software should provide
- Clear, precise, repeatable steps to recreate the issue on a new application

APPENDIX D
CX-Server Error Codes

When errors occur in the communication package, CX-Server, an error code may be displayed in the CX-Supervisor Error Log. To explain the cause of this error:

In the CX-Supervisor Error log, press the “Display Code Converter” toolbar button to launch the Error converter and show further details. See Chapter 6 Projects, Events / Error messages for more details.

Start

Or launch the Error Code converter from the Start menu, under the CX-Supervisor menu and type in the code displayed.

Alternatively, the full list of causes is below. Note that the cause is only the last 2 digits of the error code for example, if the error shown is “0x8a1b” then ignore the “8a” and just look up “0xnn1b”:

Code	Cause
0xnn00	Normal Operation.
0xnn01	Cancel Button was pressed.
0xnn02	Normal Operation.
0xnn03	The executing function was stopped in the applications callback handler.
0xnn04	The return specified buffer is not large enough to hold all the data requested.
0xnn05	The specified name is not unique within the open project file.
0xnn06	An unspecified error occurred in CX-Server that is not handled by any other error code.
0xnn07	CX-Server has not been initialised by the application.
0xnn08	The specified address is invalid for the selected or open device. This error may be generated by the Communications Manager during address validation, or the device during a read / write request.
0xnn09	The CDMPLCCommand function executed is not supported on the selected or open device.
0xnn0A	The File or filename specified is invalid or could be opened / created.
0xnn0B	The device's operating mode is invalid for the requested operation, or the device was opened in an mode not compatible with the requested operation e.g attempt to execute CDMGetData when the device was open in Read Only mode.
0xnn0C	The specified name is invalid, it name contains invalid characters.
0xnn0D	One or more of the specified parameters are invalid.

0xnn0E	The open PLC handle specified is invalid.
0xnn0F	The open Project handle specified is invalid.
0xnn10	The open User handle specified is invalid.
0xnn11	The Key specified is invalid or cannot be found.
0xnn12	The device or function is locked by another user or application.
0xnn13	The name specified does not exist or cannot be found.
0xnn14	The specified PLC name does not exist.
0xnn15	The specified Project cannot be found or created.
0xnn16	The operation cannot be performed as the correct access permissions have not be obtained. This usually indicates another host currently has the PLC access rights.
0xnn17	The function or CDMPLCCommand is not supported in this version of CX-Server.
0xnn18	The PLC could not be opened or the operation failed because the PLC is open for communications. Check the communication settings and that another application is not currently using the connection method e.g Mouse configured and using COM1 port.
0xnn19	The Point could not be opened or the operation failed because the Point is open for communications.
0xnn1A	Attempt to delete a currently selected User.
0xnn1B	Communications to the device could not be established or the connection has been broken. Check the communication settings and connection method to the device.
0xnn1C	An executing command was aborted by the execution of another command or action.
0xnn1D	CX-Server is currently processing the maximum number of requests. This error indicates the application is thrashing the maximum possible communications throughput of the device's connection.
0xnn1E	CX-Server has reached the maximum limitation for the specified function.
0xnn1F	Communications Error occurred because of a Network Routing Table problem. Check the routing tables in all devices are correctly configured. In Alpha C series PLC's ensure the 'Allow Routing Table' flag is set.
0xnn20	The specified Point name or key is invalid.
0xnn21	Device setup information or settings configuration is incorrect.
0xnn22	Device configuration information or data is incorrect.
0xnn23	The Unit number specified is invalid or does not exist.

0xnn24	The data cannot be converted to BCD as it contains Hexadecimal values between 0xA and 0xF

APPENDIX E
Using with the Omron DyaloX

The CX-Supervisor Runtime can be installed on an Omron DyaloX Industrial PC which can also be purchased with the Runtime software preinstalled. The preinstalled bundle will run “Machine Edition” projects without a USB Dongle, but a PLUS dongle must be purchased to run PLUS projects.

If installing the Runtime manually, or other application software, make sure there will be enough disk space. If not, the software can be installed on an additional compact Flash drive within the IPC.

General Use

The DyaloX IPC is designed to be a run time, end user platform and is ideal for running the “CX-Supervisor Runtime Only” package. Runtime applications can be created on a workstation with the “CX-Supervisor Full Package” installed, and then saved to a network drive, solid-state ‘flash’ disk, or USB key and transferred to the IPC.

Note: Right mouse clicks can be simulated using the “Event Selector” (mouse icon) on the System Tray. When clicked, the next touch screen press is a right click. This is sometimes required for accessing context menus.

Note: Windows Explorer has ‘Folder Options’ to allow a single icon click to open the item. This can enhance usability.

Note: The touch screen confirmation sound can be controlled and turned off from the Advanced tab on the ‘Pointer Devices’ settings in the Control Panel.

Note: Running the “CX-Supervisor Runtime Only” package on the DyaloX IPC is recommended. CX-Supervisor Developer Package can be run although some features are easier to use with an external keyboard and mouse be fitted. The Full developer package also requires more disk space.

Installing CX-Supervisor

To install CX-Supervisor Runtime only package manually:

1. First install CX-Server manually by launching Setup.exe from the CX-Server folder on the installation CD. If required for space reasons, change the install folder from the default e.g. to drive D.

2. Install CX-Supervisor Runtime Only package. If required for space reasons, change the install folder from the default e.g. to drive D, and choose NOT to install CX-Server as part of the installation as it has already been done in step 1 above.

3. Install any ActiveX controls used by the application. This includes any use of Microsoft Forms (FORMS20.DLL) which is not included on the DyaloX IPC by default.

4. If you are running a “Machine Edition” project you will not need a USB Dongle. However if you are running a PLUS project install the USB Dongle copy protection.

5. Copy the desired runtime application.

Note: Remember that drive C is for the operating system, and space may become limited. Consider installing other software to other drives (like additional flash cards or USB keys). However, note that typically, all software automatically installs some files to Windows system folders, or Common folders on the C drive even if the program is installed to another drive.

Communication Settings

The USB port can be used to communicate with compatible Omron hardware. The CX-Server USB drivers for the USB PLC must be installed manually if required, and can be found in the CX-Server installation directory.

Note: At runtime, the DyaloX IPC can only use the COM ports that are configured in the operating system. The Development platform therefore must have the same COM ports available so they can be selected at development time, and the same port available for use at runtime. If required, it is possible to reconfigure the COM port for a configured device once the application is compiled if the “Embed CX-Server Project within CX-Supervisor .SR2 file” setting is turned off so the devices settings can be editing on the target platform.

APPENDIX F
Obsolete Features

This appendix provides a summary of features that are obsolete and have been removed from the standard documentation. Details are included here to assist maintaining old projects still using these features. These features should not be used in development of new solutions as it is likely support for the following features may and will be removed from the next or future releases.

Windows 98, Windows ME, Windows NT

This product no longer installs on Windows 98, ME or NT. It is recommended to upgrade to Windows 2000 or Windows XP Professional.

Configuring a OPC/DCOM Client PC running Windows 98 or Me

1, 2, 3…

1. Ensure **File and Printer sharing** is enabled by selecting Network from the Control Panel. Add a service and click either “File and print sharing for Microsoft Networks” or “File and print sharing for Netware Networks” as appropriate.

2. Start DCOMCNFG.EXE e.g. by selecting **RUN** from the Start button. The default location is C:\WINDOWS\SYSTEM.
3. View the **Default Properties** tab. Ensure that the **Enable Distributed COM on this computer** is checked.

A CX-Supervisor Client running on Windows 98 or ME also requires the **Microsoft Remote Registry** network service to be installed with the operating system and correctly configured on <u>both</u> the server and client machine. To check: start the Control Panel and view the Network settings. In the list of network components, look for **Microsoft Remote Registry**. If it does not exist, follow these steps to add it.

1, 2, 3…

1. In the Network settings, ensure **User-level access control** is selected on the **Access Control** tab.
2. From the **Configuration** tab, click **Add** to add a Network component. Choose **Service** from the type list and click **Add**.
3. Click **Have Disk…** and browse your Windows CD. Select the path (**\Admin\Nettods\remotReg**) for Win95 CD or (**\Tools\ResKit\NetAdmin\RemotReg**) for Win98 and select **regsrv.inf**.
4. Follow the screen prompts to complete installation and reboot if necessary.
5. On the server machine, select **Passwords** from the Control Panel.
6. Ensure the **Enable remote administration of this server** option is checked
7. Add all required user ids to the Administrators list by clicking **Add…**

Note: Windows ME no longer includes the **Microsoft Remote Registry** network service on the product CD but this can still be installed and used from any Windows 95 or Windows 98 CD ROM using the above steps.

Configuring a DCOM / OPC Server PC running Windows 98 or Me

1, 2, 3…

1. Ensure **File and Printer sharing** is enabled by selecting Network from the Control Panel. Add a service and click either "File and print sharing for Microsoft Networks" or "File and print sharing for Netware Networks" as appropriate.

2. Start DCOMCNFG.EXE e.g. by selecting **RUN** from the Start button. The default location is C:\WINDOWS\SYSTEM.
3. View the **Default Properties** tab. Ensure that the **Enable Distributed COM on this computer** is checked.
4. View the **Default Security** tab and check the **Enable remote connection** check box.
5. From the **Default Properties** tab, configure the **Default Authentication Level** to **Connect** and the **Default Impersonation Level** to **Identify**. Setup the access permissions by either:

   a) On the **Default Security** tab, adding the user to the Access list by clicking the **Edit Default…** button in each case. The user added should have Administrator rights on the local PC. If not, it may be necessary to add user groups 'INTERACTIVE' and 'NETWORK' as well.

   b) From the **Applications** tab, configure the properties for **OPC Server** and **OPC ServerList Class**. On the **Security** tab, add the required users to each of the **Custom Permissions**. The users added should have Administrator rights on the local PC. If not, it may be necessary to add user groups 'INTERACTIVE' and 'NETWORK' as well.

Third party servers and clients running on Windows 98 or ME may require the **Microsoft Remote Registry** network service to be installed with the operating system and correctly configured on <u>both</u> the server and client machine. To check: start the Control Panel and view the Network settings. In the list of network components, look for **Microsoft Remote Registry**. If it does not exist, follow these steps to add it.

1, 2, 3…

1. In the Network settings, ensure **User-level access control** is selected on the **Access Control** tab.
2. From the **Configuration** tab, click **Add** to add a Network component. Choose **Service** from the type list and click **Add**.
3. Click **Have Disk…** and browse your Windows CD. Select the path (**\Admin\Nettods\remotReg**) for Win95 CD or (**\Tools\ResKit\NetAdmin\RemotReg**) for Win98 and select **regsrv.inf**.

4. Follow screen prompts to complete installation and reboot if necessary.
5. On the server machine, select **Passwords** from the Control Panel.
6. Ensure the **Enable remote administration of this server** option is checked.
7. Add all required user ids to the Administrators list by clicking **Add…**.

Note: Windows ME no longer includes the **Microsoft Remote Registry** network service on the product CD but this can still be installed and used from any Windows 95 or Windows 98 CD ROM using the above steps.

Windows 95

This product is no longer supports Windows 95. It is recommended to upgrade to Windows 2000 or Windows XP Professional.

System Points

The following System Points are obsolete and are no longer updated. Scripts and expressions will currently still compile, but their value at Runtime will always be 0.

$GDIResources	Integer	0-100	Percentage of GDI resources free.
$SystemResources	Integer	0-100	Percentage of system resources free.
$UserResources	Integer	0-100	Percentage of user resources free.

DDE

Overview

CX-Supervisor supports Dynamic Data Exchange (DDE), which is a method of communication between Windows programs. DDE uses messages to exchange data between applications and a protocol to synchronise the passing of data. DDE applications fall into four categories *client, server, client/server* and *monitor*. A client application requests data or services from a server application. A server application responds to a client applications requests for data or services. Monitor applications can intercept DDE messages but cannot act on them, they are therefore useful for debugging purposes. CX-Supervisor is a client/server application, which is both a client application and a server application, thus requesting and providing information.

All CX-Supervisor DDE data transfers are carried out on points and are asynchronous transfers. There are two types of DDE points that can be created, namely ‘DDE Client Points’ and ‘DDE Server Points’.

With ‘DDE Client Points’ all data transfers or conversations are initiated by CX-Supervisor either sending data to or requesting data from external DDE Server Application(s). For example, a CX-Supervisor point could be linked to update a cell on a Microsoft Excel worksheet.

With ‘DDE Server Points’ all data transfers are initiated by external DDE Client Application(s) either requesting or sending point values. For example, a value could be entered into a cell in Microsoft Excel which would update a CX-Supervisor point.

DDE Client Points

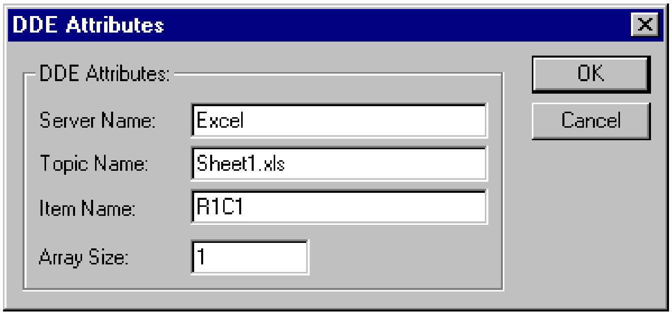
A DDE client point sends data to or requests data from an external server application. This chapter explains how to create points that make use of the CX-Supervisor DDE Client capabilities. In order for data to be transferred between a point and a server application the point must uniquely identify the application and the data that is to be used in the transfer. DDE applications use a three-tiered identification system to distinguish themselves from other DDE applications. An *application name* is at the top of the hierarchy, the application name refers to a server application e.g. “EXCEL”. A *topic name* further defines a server application e.g. for Microsoft Excel the topic would define the worksheet to be used e.g. “SHEET1.XLS”, a server can support one or more topics. Finally each topic can have one or more *item names,* which uniquely identifies a data item within a topic, i.e. “R1C1” or a cell name reference identifies a single cell within a Microsoft Excel worksheet.

Example

A DDE Client point named “DDE1” that connects to Microsoft Excel, worksheet – “Sheet1.xls” and cell “R1C1”, is created.

1, 2, 3…

1. Select the Add Point button from the Point Editor toolbar. The Add Point dialog is displayed.
2. Enter “DDE1” in the *Point Name:* field.
3. Set the *I/O Attributes* setting to *DDE* and click on the **Setup** pushbutton. The DDE attributes dialog is displayed.



DDE Attributes

DDE Attributes:

Server Name: Excel

Topic Name: Sheet1.xls

Item Name: R1C1

Array Size: 1

4. Enter “Excel” in the *Server Name:* field. This is the name of the external DDE server application.

5. Enter “Sheet1.xls” in the *Topic Name:* field. This is the required topic, in this case it is a Microsoft Excel worksheet named ‘Sheet1.xls’. It is possible to specify a specific topic: for instance, in Microsoft Excel, to specify a sheet within a book a colon is used to delimit the information, e.g. ‘Book1:sheet3.xls’.

6. Enter “R1C1” in the *Item Name:* field. This refers to the item name.

7. Enter “1” in the *Array Size:* field.

8. Click the **OK** pushbutton to accept the settings in both the DDE Attributes dialog and the Add Point dialog.

Note: It is not necessary to give ‘DDE Client Points’ DDE access via the Advanced dialog - this field is only used in the creation of ‘DDE Server Points’. If the DDE Access Read/Write setting is set ‘ON’, this point’s value would then be ‘exposed’ to change by external DDE server application(s) which may not always be desirable.

This process is repeated for any further DDE data transfers that are required.

DDE Server Points

A DDE server point receives data from or receives a request for data from an external Client application. This chapter explains how to create “DDE Server Points”, these points are exposed to DDE Client applications either for “Read/Only” or “Read/Write” access. In order for a point to take part in a conversation with a DDE Client, it must be given ‘DDE Access’. The following dialog demonstrates how to give a Real memory point, DDE Read/Write access:

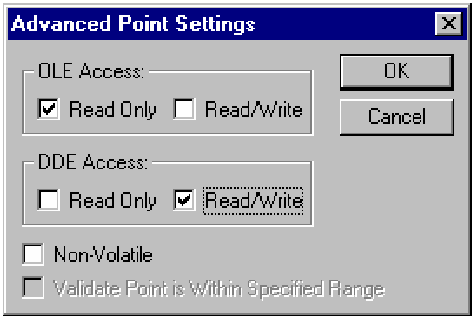
A DDE Server point is created as follows:

1, 2, 3…

1. Select the Add Point button from the Point Editor toolbar. The Add Point dialog is displayed.

2. Enter a meaningful name for the DDE Server point in the *Point Name:* field.

3. Click on the **Advanced** pushbutton. The Advanced Point Settings dialog is displayed.



OLE Access:
☑ Read Only □ Read/Write

DDE Access:
□ Read Only ☑ Read/Write

□ Non-Volatile

□ Validate Point is Within Specified Range

4. Ensure the DDE Access Read/Write setting is set to ‘ON.
5. Click the **OK** pushbutton to accept the settings in both the Advanced Point Settings dialog and the Add Point dialog.

Note: The DDE Access group’s Read/Write box in the Advanced Point Settings dialog is checked, this is the only action required to expose “DDESVR3” to a DDE Client application for read/write access.

Note: Any CX-Supervisor point can be given DDE Access, even DDE Client points.

DDE Array Points

CX-Supervisor supports arrays in DDE, for both ‘Client’ and ‘Server’ transactions. DDE Client data transfers are initiated by CX-Supervisor, either sending or requesting data from external DDE Applications (such as Microsoft Excel). DDE Server data transfers are initiated by external DDE Applications either sending or requesting data to or from CX-Supervisor.

Refer to the *CX-Supervisor Script Language Reference Manual* for further details.

DDE Client Array Points

This chapter explains how to create DDE Client Array points. This is similar to the way normal DDE Client points are created, except extra information is required to specify the range of the array and also whether it is to be stored as a row or column in the Server application.

1, 2, 3…

+

1. Select the Add Point button from the Point Editor toolbar. The Add Point dialog is displayed.
2. Enter a meaningful name in the *Point Name:* field.
3. Set the *I/O Attributes* setting to *DDE* and click on the **Setup** pushbutton. The DDE attributes dialog is displayed.
4. Enter an application in the *Server Name:* field.
5. Enter a topic in the *Topic Name:* field.
6. Enter the item in the *Item Name:* field. It is also necessary to specify how the array is to be transferred. For instance, in Microsoft Excel, this is either as a ‘[row]’ or ‘[column]’, the default is as a row which can be omitted. The following are valid array item names: “R1C1:R3C1[col]”, “first:last[c]”, “R5C2:R5C5”, “r1c1:r3c1[column]”.
7. Enter the number of elements in the array in the *Array Size:* field. This must match with the specified range in the Item Name field

In the CX-Supervisor Runtime environment, the above DDE Client array transactions are initiated by means of script in the same way as single DDE Client points. The CX-Supervisor script below gives examples of the facilities available with DDE Client array points:

1) Example CX-Supervisor Script demonstrating client array points

```
chan = DDEInitiate(“Excel”, “Book1.xls”)
IF chan > 0 THEN

  ‘Establish data transfers between point ‘DDEArray’ and Excel
  ‘the initial values of DDEArray is sent to Excel
  DDEOpenLinks(chan)

  ‘The array is initialised with the value 5 and sent to Excel in
one
  operation.
  InitArray(DDEArray, 5)

  ‘The contents of ‘MemoryArray’ are copied into ‘DDEArray’ and the
  ‘array are sent to Excel in one operation.
  MemoryArray[0] = 100
  MemoryArray[1] = 34
  MemoryArray[2] = 89 * 6
  CopyArray(MemoryArray, DDEArray)

  ‘Sets element 2 of the array to 6 and sends the whole array to
Excel
  ‘Note: Use ‘On Request’ option and ‘OutputPoint’ to send the array
  ‘after ‘setting several elements of a large array.
  DDEArray[2] = 6
ENDIF
```

DDE Script Functions

The existing DDE Script functions DDEPoke() and DDERequest() can be used with any array points as the following example CX-Supervisor Scripts show:

1) Example CX-Supervisor Script demonstrating DDEPoke() with arrays

```
chan = DDEInitiate(“Excel”, “Book1:Sheet2”)
IF chan > 0 THEN

  ‘Send element 1 of ‘RealArray’ to Excel
  DDEPoke(chan, "R16C1", RealArray[1])

  ‘Send IntegerArray to Excel as a column (note: range and column
must
  ‘be ‘specified)
  DDEPoke(chan, "R1C1:R3C1[column]", IntegerArray)

  ‘Send TextArray to Excel as a row (default only range required)
  DDEPoke(chan, "R1C2:R1C4", TextArray)
ENDIF
```

2) Example CX-Supervisor Script demonstrating DDERequest() with arrays

```
chan = DDEInitiate(“Excel”, “Book1:Sheet3”)
IF chan > 0 THEN

  ‘Request a row of cells from Excel and copy into ‘Array1
  Array1 = DDERequest(chan, "R1C1:R1C3")

  ‘Request a column of cells from Excel and copy into Array2, using
  ‘the return ‘flag
  Array2 = DDERequest(chan, "R1C2:R3C2", bReturnFlag)

  ‘Request a cell value from Excel and copy into element 2 of
‘Array1’
  Array1[2] = DDERequest(chan, "R3C2")
ENDIF
```

Note: All the above points must have DDE Read/Write access set.

DDE Server Array Points

The value of an array point named ‘ddearray’ in a CX-Supervisor project called ‘ddetest.srt’ can be read from a Microsoft Excel worksheet by entering the following formula format into a cell.

```
=<Server>|<Topic>!<item>.<index>
```

Example

```
=SCS|Point!ddearray.3        ‘access ddearray[3] using ‘Point’ topic
```

or

```
=SCS|ddetest.srt|ddearray.0  ‘access ddearray[0] using ‘Project’
topic
```

Note: Microsoft Excel accepts the square brackets ‘[ ]’ used in CX-Supervisor to reference an array index in a formula; use ‘.’ instead.

Note: The index must always be specified if an individual array element is required.

Note: CX-Supervisor supports ‘Hot’ DDE links, if Microsoft Excel has the link option automatic set, then the value in Microsoft Excel are updated whenever ddearray[index] changes.

The above example is the simplest way to access/read single elements of an array from Microsoft Excel, to read/write whole arrays, it is necessary to use macros (Microsoft Excel Visual Basic scripts).

The example scripts that follow have all been created using Microsoft Excel and are working examples. They contain the minimum amount of information required to demonstrate the particular facility being described; i.e. they do not contain any error-checking code.

Sending Arrays to CX-Supervisor via DDEPoke()

In order to write to an CX-Supervisor array point using the DDEPoke() function, it must first be given DDE Read/Write access, via the Advanced Point Settings dialog when adding or modifying the point. The following script shows how to send arrays of values from Microsoft Excel to CX-Supervisor via DDEPoke().

1) Example sending array values from Microsoft Excel to CX-Supervisor

```
Sub SendArrayValues()
  Dim chan As Integer
  chan = DDEInitiate(“SCS”, “Point”)
  If chan <> 0 Then

    ‘Send a row of data to an array point named “Array1”
    DDEPoke chan, “Array1”, Range(Cells(1,1), Cells(1,3))

    ‘Send a column of data to an array point named “Array2”
    DDEPoke chan, “Array2”, Range(Cells(2,1), Cells(4,1))

    ‘Send individual array element values to “Array3”
    ‘The ‘[ ]’ or ‘.’ format can be used to delimit the array index
    DDEPoke chan, “Array3[0]”, Cells(1,1)
    DDEPoke chan, “Array3.1”, Cells(1,2)
    DDEPoke chan, “Array3[2]”, Cells(1,3)
  End If
End Sub
```

Requesting Arrays from CX-Supervisor via DDERequest()

Requesting arrays from CX-Supervisor and storing them in Microsoft Excel is a little more complicated than sending, in that both CX-Supervisor and Microsoft Excel need to know if the array is to be stored in rows or columns. CX-Supervisor is informed of the row/col requirement by specifying either ‘row’ or ‘column’ after the array points name. The default is ‘row’ if nothing is specified. The following are all valid examples of specifying names for “Array1” in a DDERequest():

```
“Array1”, “Array1:Row”, Array1:r”        ‘Valid ways to specify a row
“Array1:Column”, “Array1:col”, Array1:C” ‘Valid ways to specify a
                                         ‘column
```

Microsoft Excel is informed of the row/col requirement by specifying a Range of cells in either row or column format. Both the Microsoft Excel and CX-Supervisor specifications must match in order for the call to be successful. It is not necessary to specify rows or columns if a single element of an array is required.

Note: DDE Requests are one-shot request, i.e. they are not part of a ‘Hot’ link.

1) Example requesting CX-Supervisor array values from Microsoft Excel

```
Sub RequestingArrayValues()
  Dim chan As Integer
  chan = DDEInitiate("SCS", "Point")
  If chan <> 0 Then

   'Request "Array1" from CX-Supervisor and store in a row starting
at R1C1
   Range(Cells(1,1), Cells(1,3)) = DDERequest(chan, "Array1")

   'Request "Array2" from CX-Supervisor and store in a column
starting at R2C1
   Range(Cells(2,1), Cells(4,1)) = DDERequest(chan, "Array2:col")

   'Request elements [2] and [3] from "Array2" and store in R3C1 &
    'R3C2
    'The '[ ]' or '.' format can be used to delimit the array index
    Cells(3,1) = DDERequest(chan, "Array2[2]")
    Cells(3,2) = DDERequest(chan, "Array2.3")
  End If
End Sub
```

OLE Automation

OLE Automation provides a mechanism whereby one application can control another. In order to access a CX-Supervisor points value via OLE Automation, it must be given either OLE Read or OLE Write access. This can be achieved by checking the appropriate box in the Advanced Point Settings dialog when the point is created or modified.

Note: All CX-Supervisor System points are given OLE2 read access by default.

CX-Supervisor exposes the following Runtime functions via OLE2 Automation:

- *SetValue(stringPointName,Value)*. Enables any point with OLE Read/Write access to be modified (value may be Boolean, integer, long or string).
- *GetValue(stringPointName,&Value)*. Enables any point with OLE Read access to be monitored (value may be Boolean, integer, long or string).
- *QueryCount()*. Returns the total number of points in the CX-Supervisor database.
- *QueryId(stringPointName)*, Returns the WORD id of a given point name. The id is an integer in the range between 1 and the total number of points in the CX-Supervisor database.
- *QueryOLE(wordID)*. Returns the Read/Write access rights of a point as an integer. 0 represents no access type; 1 represents Read Only; 2 represents Read/Write; 3 represents an invalid access type.
- *QueryType(wordID)*. Returns the points data type as an integer from a given id. 0 represents a void datatype; 1 represents a digital datatype; 2 represents an integer datatype; 3 represents a real datatype; 4 represents a text datatype; 5 represents an invalid datatype.

QueryName(wordID). Returns the points name string from a given identity. CX-Supervisor provides OLE Automation helper routines and code examples in the OLE2AUTO subdirectory. Refer to the file OLE2AUTO.WRI in that subdirectory for more information.

OLE Overview

The Microsoft Windows operating environment allows the transfer and sharing of information between applications by using a technique known as Object Linking & Embedding, or OLE (pronounced olé). Embedding allows drawings, sounds or almost any objects to be created within an application, and then inserted (embedded or linked) within another file or document. The embedded object can then be edited (or activated) by merely double clicking on it. Linking allows an object to be shared by several documents by forging links between the destination documents and the source object, e.g. a company logo may appear in several word processor documents by linking to a graphics object rather than embedding.

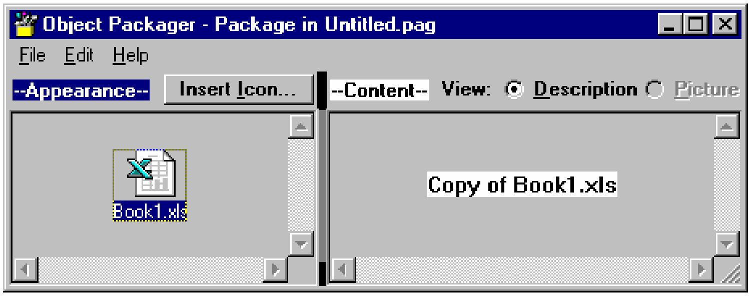
Embedding and Linking Objects within the CX-Supervisor application can either be achieved by using the Object Packager, or by embedding or linking objects directly within a page.

Note: Packages may only be embedded or linked into CX-Supervisor pages if they are created using OLE-compliant applications. This applies to objects either directly linked or embedded within CX-Supervisor, and those inserted using the Object Packager.

Object Packager

The Object Packager is a Windows application which enables objects to be "wrapped" for insertion into documents. It can be used to either create a link to a file held on disk or to embed an object into a page. Once embedded or linked, the object can be moved and positioned in a similar way to any other object created within CX-Supervisor.

To use the Object Packager, select *Package* from the list of objects displayed in the Insert Object dialog shown later in this chapter. An example of a typical Object Packager window is as follows:

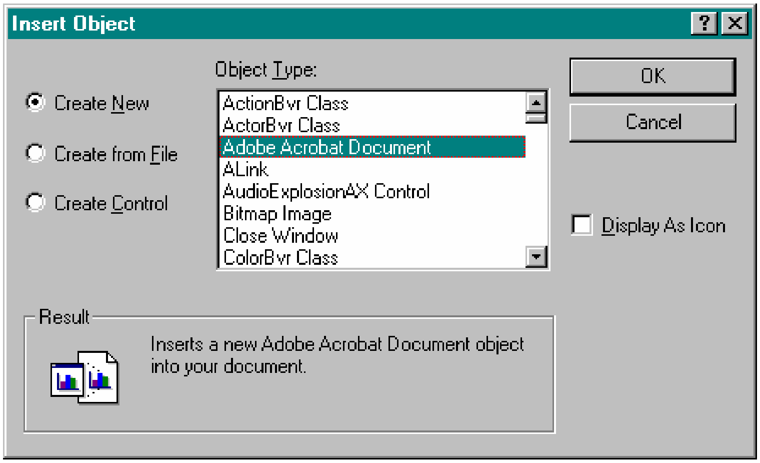


For the remainder of this chapter only the direct insertion of objects using the CX-Supervisor built-in functions are detailed. For further details of using the Object Packager refer to the *Windows User Guide*, or the *Object Packager On-line Help*.

Whether using the Windows Object Packager or embedding or linking objects directly, proceed as follows:

1, 2, 3…

1. Click on the page in which the object is to be inserted.
2. Click the **Insert OLE Object** button. The Insert Object dialog is displayed:



Insert Object

● Create New

○ Create from File

○ Create Control

Object Type:

ActionBvr Class
ActorBvr Class
Adobe Acrobat Document
ALink
AudioExplosionAX Control
Bitmap Image
Close Window
ColorBvr Class

□ Display As Icon

Result

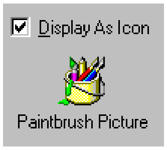
Inserts a new Adobe Acrobat Document object into your document.

Creating an Object From New

To create an object from new, proceed as follows:

1, 2, 3…

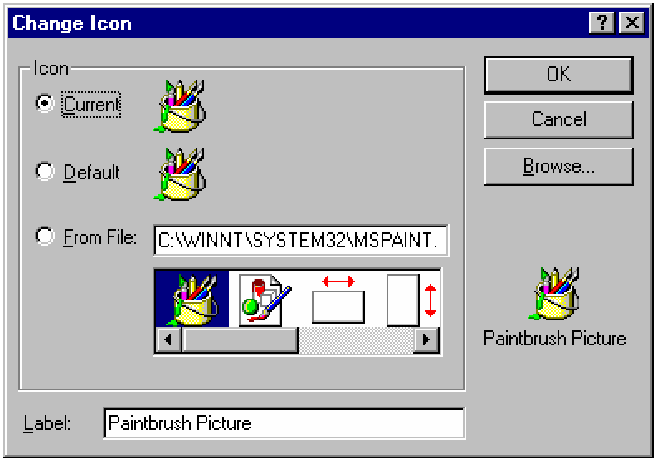
1. Scroll through the list of object types presented in the list box until the desired type is highlighted.
2. Click the **OK** pushbutton to insert the object into the current page and display it as it would look from within the application in which the object was created.



☑ Display As Icon

To display the object as an icon, click on the *Display As Icon* setting.

The icon displayed is the first one which is stored in the icon resource table for the application which created the object (in the case of the example shown, a Paintbrush icon). If another icon is required, clicking the **Change Icon** pushbutton displays the following dialog:



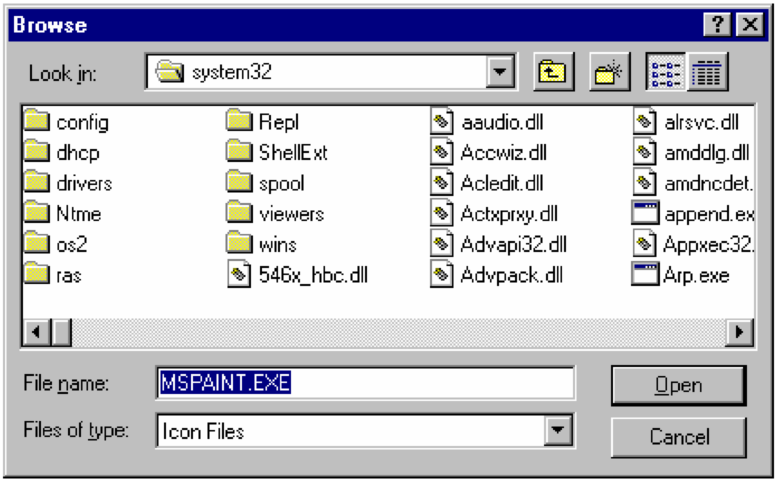
● Current
○ Default
○ From File: C:\WINNT\SYSTEM32\MSPAINT.
Label: Paintbrush Picture

1, 2, 3…

1. Select either the current or default icon by clicking either the *Current:* or *Default:* setting; and click the **OK** pushbutton to return to the Insert Object dialog. Should a different label be required to display beneath the inserted icon, delete the contents of the *Label:* field and type in a new label (the default label offered is the filename of the file containing the inserted object).

2. Type the full path of the target application or DLL ending with the application or DLL name, into the *From File:* field. To locate an existing file name or a different path, click the **Browse** pushbutton to display the Browse dialog.

An example of the Browse dialog is shown as follows:



Look in: system32
File name: MSPAINT.EXE
Files of type: Icon Files

Both programs (extension .EXE) or Dynamic Link Libraries (extension .DLL) are listed in the *Files of Type* field.

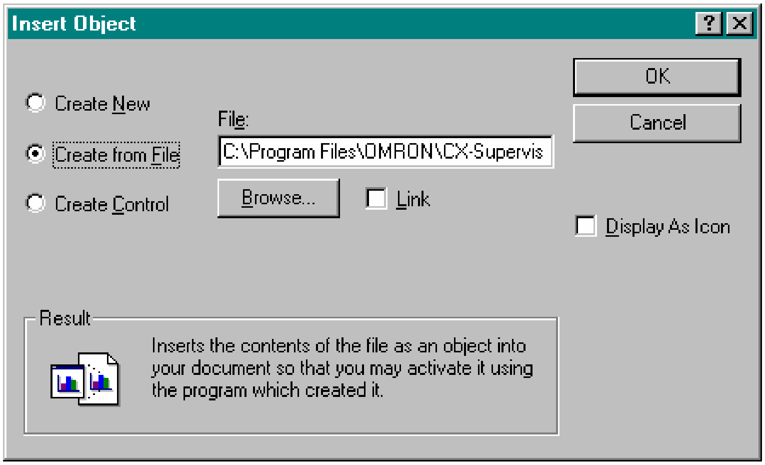
The Browse dialog functions identically to the File Open dialog described in chapter 6, Projects except the file list offered shows icons as opposed to pages.

1, 2, 3…

1. On return to the Change Icon dialog, select the required icon from those presented.
2. Change the icon's label (if required).
3. Click the **Open** pushbutton to return to the Insert Object dialog.
4. On return to the Insert Object dialog click the **OK** pushbutton to return to the current page and embed the selected object into it at the current insertion point.

Creating an Object From a File

Creating an object from a file allows linking or embedding of objects created at sometime in the past. As before, select *Insert New Object* from the *Edit* menu, and the Insert Object dialog is displayed. Click the *Create From File:* setting and the dialog changes to a format similar to that of the following:



Insert Object

○ Create New

● Create from File

○ Create Control

File: C:\Program Files\OMRON\CX-Supervis

□ Link

□ Display As Icon

Result

Inserts the contents of the file as an object into your document so that you may activate it using the program which created it.

Type the full path of the file containing the object to be inserted, ending with the full file name, into the *Create from File:* field. To locate an existing file name or a different path, click the **Browse** pushbutton to display the Browse dialog.

A Browse dialog similar to that used for changing an object's icon is displayed, the difference being that the *List Files of Type* field has a single entry of *All Files (extension .*).*

1, 2, 3…

1. Use the *Look in:* field to navigate to the appropriate location.

2. Select the file from the list offered and click the **Open** pushbutton to return to the Insert Object dialog.

3. On return to the Insert Object dialog click the **OK** pushbutton to return to the current page and embed the selected object into it at the current insertion point.

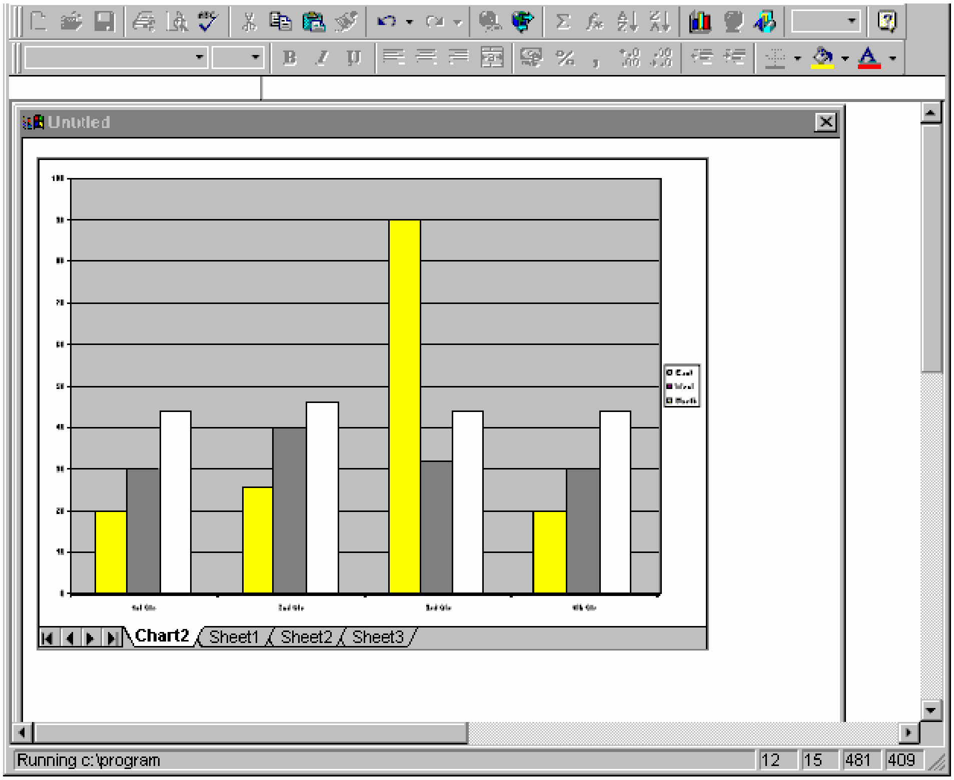
To display the object as an icon, follow the procedure detailed in the previous chapter and click the **OK** pushbutton to return to the current page and embed the selected object, in icon form, into it at the current insertion point.

It is often more appropriate to link to an object rather than embedding it in the current page. By linking to an object rather than embedding it CX-Supervisor always has access to the most up-to-date copy of the object every time the page containing it is opened.

Placing a check mark in the *Link* field forges a link between the page and the file containing the object.

All other activities are carried out in an identical manner to those explained previously.

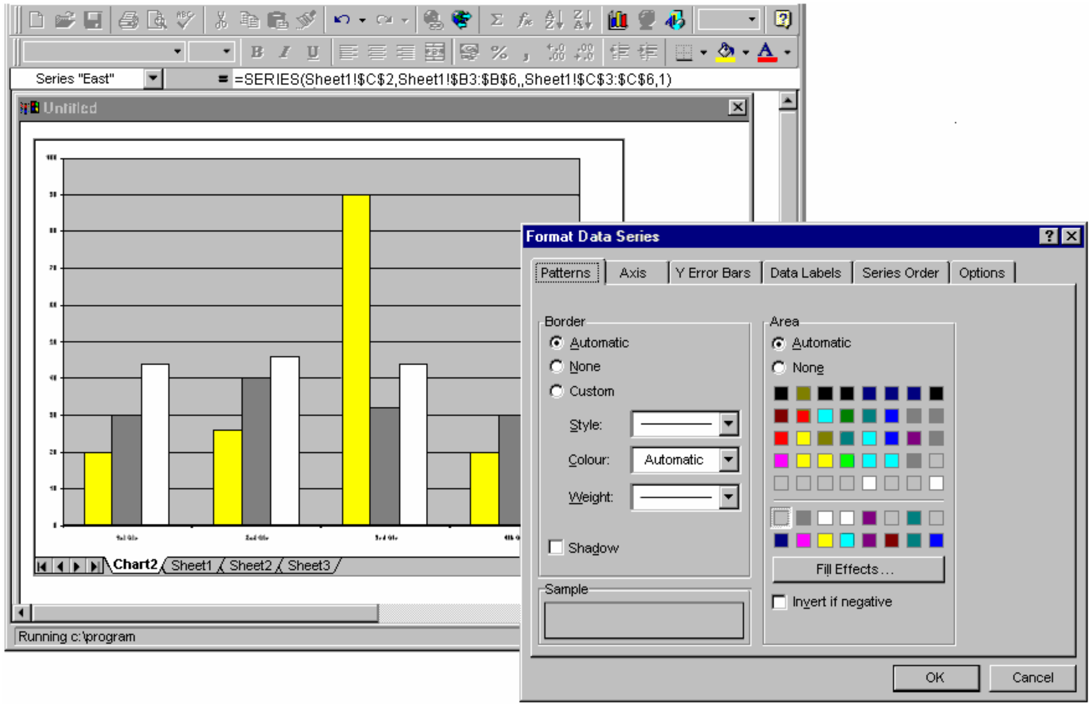
The following illustrates a Microsoft Excel Chart object which has been inserted in a page:



Activating an Object

To activate an embedded or linked object, merely double click using the left mouse button on the object's graphical representation on the page (either a true representation of the object or an icon). If the object is a document or image, Windows activates the application that created the object and places it, ready to be edited.

In the following example, the bitmap image when activated launches the Format Chart Area application to allow editing:



Series "East" =SERIES(Sheet1!$C$2,Sheet1!$B3:$B$6,,Sheet1!$C$3:$C$6,1)

Running c:\program

Format Data Series

Border: ● Automatic ○ None ○ Custom

Colour: Automatic

□ Shadow

Area: ● Automatic ○ None

□ Invert if negative

When editing is complete, select *Update and Exit* from the application's *File* menu to return to CX-Supervisor.

Note: The Update and Exit command varies from application to application.

If the object is an animation or sound file, when activated it is played through until it ends, at which point control is returned to CX-Supervisor.

Note: If the application which created the object no longer be available, CX-Supervisor reports an error.

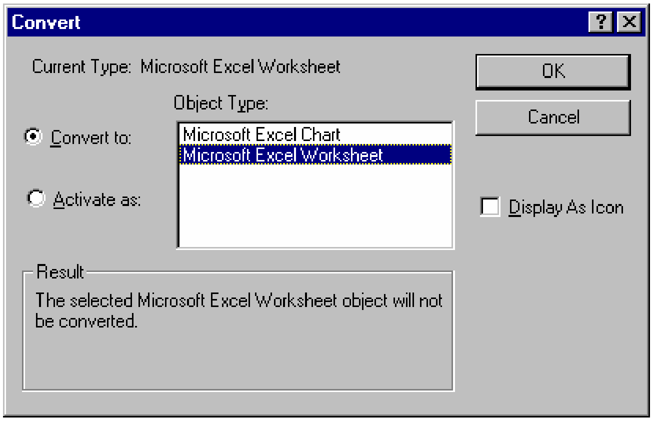
Converting a Package Object

Certain types of object may be converted from one form to another. To initiate conversion:

1, 2, 3…

1. Select the object on the page.
2. Activate the *Edit* menu and select *Package Object (*displayed as the last item on the *Edit* menu).
3. Select *Convert* from the sub-menu displayed.

Note: The menu item name varies from application to application but always references the inserted object.



Convert

Current Type: Microsoft Excel Worksheet

Object Type:

● Convert to:

○ Activate as:

Microsoft Excel Chart
Microsoft Excel Worksheet

□ Display As Icon

Result

The selected Microsoft Excel Worksheet object will not be converted.

GLOSSARY OF TERMS

ADO
ADO stands for Active Data Objects and is data access technology which uses OLE-DB to access data sources in a uniform way e.g. MS-Access databases, MS-Excel spreadsheets and Comma Separated Variable files.

AND
A logic operator used to interrogate Boolean type points. AND returns 'TRUE' if all arguments are 'TRUE'. An example of AND is that if *a* is a statement and *b* is a statement, AND returns 'TRUE' if both *a* and *b* are 'TRUE'. If one or both statements return 'FALSE' then AND returns 'FALSE'.

Application
A software program that accomplishes a specific task. Examples of applications are CX-Supervisor, CX-Server and Microsoft Excel. CX-Supervisor and its development environment allows the creation and testing of new applications through a Graphical User Interface (GUI).

Arguments
Words, phrases, or numbers that can be entered on the same line as a command or statement to expand or modify the command or statement within the CX-Supervisor script language. The command acts on the argument. In essence the command is a *verb*, and the argument is the *object of the verb*. An example of an argument in CX-Supervisor is "`DDETerminate(channel)`" where `DDETerminate` is a command within the script language, and `channel` is the argument upon which the command will act.

ASCII
An old standard, defining a set of characters. Officially using only 7 bits allows definitions for only 127 characters, and does not include any accented characters.

Bitmap
The representation of an image stored in a computer's memory. Each picture element (pixel) is represented by bits stored in the memory. In CX-Supervisor a bitmap image can be installed as a single object.

Boolean type
A type of point where the value of the point can be one of two states. Essentially the two states are '0' and '1', but these states can be assigned a meaningful designation. Examples are:

State	Example	Example	Example	Example
0	'OFF'	'FALSE'	'OUT'	'CLOSED'
1	'ON'	'TRUE'	'IN'	'OPEN'

See also: AND, NOT and OR.

COM
COM is a Microsoft technology that allows components used to interact.

Communications Driver
The relevant communications management system for OMRON PLCs in conjunction with Microsoft Windows, providing facilities for other SYSMAC software to maintain PLC device and address information and to communicate with OMRON PLCs and their supported network types.

Constant
Within CX-Supervisor, a constant is a point within the script language that takes only one specific value.

Control Object
In CX-Supervisor, a control object is applied in the development environment and can be a pushbutton, a toggle button, a slider, a trend graph, a rotational gauge or a linear gauge. Essentially a control object can be a complex graphic object consisting of a number of primitive graphic objects, which provides user interaction.

CX-Server
An advanced communications management system for OMRON PLCs providing facilities for software to maintain PLC device and address information and to communicate with OMRON PLCs and their supported network types. CX-Server supports CS-Series PLCs.

Database connection
A Database connection (or Connection for short) contains the details used to access a data source. This can either be via Data Source Name (DSN), filename or directory.

Database Connection Level
A Database Connection Level is a string which determines what level in the database tree hierarchy is to be operated on. Some examples are listed below:

"Northwind"	Connectionlevel
"CSV.Result"	Recordset level
"Northwind.Order Details.OrderID"	Field level
"Invoice.Data Types"	Schema level

Database Recordset
A Database recordset (or Recordset for short) is a set of records. This could either be an actual Table in the database, or a table that has been generated as a consequence of running a Query.

Database Schema
A Database Schema (or Schema for short) obtains database schema information from a Provider.

Database Server Query
A Database Server Query (or Server Query for short) is a query that is stored in the actual Database. They are pre-defined and added by the database designer which means they are 'fixed' for the duration of a project. Server Queries may have pre-defined 'Parameters', which allow criteria to be passed to the query at runtime e.g. values to filter, allowing one query to be used to produce different results. Each pre-defined parameter must have a Parameter Association defined. Because these queries are stored in a compiled and tested form they are more efficient and therefore preferential to running a SQL Query.

Database SQL Query
A Database SQL Query (or SQL Query for short) is interpreted dynamically at runtime. The SQL Text can be modified at runtime, enabling different Queries to be run for varying situations however, the SQL Text has to be compiled on the fly every time it is executed and consequently is less efficient than a Server Query.

DBCS

DBCS stands for Double Byte Character Set and is a Microsoft extension of ASCII which uses 2 bytes (16 bits) to define character codes. With this larger range it can include accented characters, extended ASCII characters, Nordic characters and symbols.

DCOM

DCOM is a distributed version of COM that allows components on different PCs to interact over a network.

DDE

Dynamic Data Exchange. A channel through which correctly prepared programs can actively exchange data and controls other applications within Microsoft Windows. DDE technology was notoriously unstable and was replaced with OLE technology.

See also Item, Server, server application and Topic.

Development Environment

SCADA applications are created and tested using the development environment within CX-Supervisor. On completion, the finished application can be delivered as a final customer application to be run by the run-time environment.

DLL

Dynamic Link Library. A program file that although cannot be run stand-alone as an executable, can be utilised by one or more applications or programs as a common service. DLL files have a *.DLL extension. DLL's comprise a number of stand-alone functions. In CX-Supervisor, a DLL containing icons can be accessed to represent the display part of an OLE object. One such DLL, 'MORICONS.DLL', is provided in the standard Microsoft Windows installation.

Download

A recipe is *downloaded* during runtime. This process involves identifying the appropriate recipe and executing the validation code, if any exists. The download is complete when each ingredient has set its point to the target value.

Executable

A file that contains programs or commands of an application that can be executed by a user or another application. Executable files have a *.EXE file extension. CX-Supervisor provides two executable files, one for the development environment (CXSUPERVISORDEV.EXE), and one for the run-time environment (SCS.EXE).

Expressions

In the CX-Supervisor script language, expressions are a construct for computing a value from one or more operands. For instance, in the example "`lift = height + rate`", the expression is "`height + rate`" where the result yielded from the expression is used for the value of "`lift`".

Outside of the script language, expressions consisting of operators and operands can be used to control objects, through actions.

Field association

A field association enables a link to be made between a CX-Supervisor Point and a particular field (i.e. column) within a recordset.

Graphic Object In CX-Supervisor, a graphic object is created in the development environment, and can be a line, an arc, a polygon (including a square and rectangle), a round rectangle, an ellipse (including a circle), or a polyline. A complex object can exist as a combination of two or more graphic objects.

GUI Graphical User Interface. Part of a program that interacts with the user and takes full advantage of the graphics displays of computers. A GUI employs pull-down menus and dialog boxes for ease of use. Like all Microsoft Windows based applications, CX-Supervisor has a GUI.

I/O type Input/Output type. An attribute of a point that defines the origin and destination of the data for that point. The data for a point can originate (be *input* from) and is destined (is *output* to) to the internal computer memory or PLC.

Icon Pictorial representations of computer resources and functions. The CX-Supervisor development environment and run-time environment are run from icons.

Ingredient Each recipe consists of at least one ingredient. Each ingredient must be related to an existing point.

Integer type A type of point where the value of the point can only be a whole positive or negative number.

Item Within the CX-Supervisor script language, Item is a generic term for a point, OPC item or Temperature Controller item.

JScript A Java style scripting language supported by Microsoft's Windows Scripting Host.

JVM Java Virtual Machine.

Microsoft Excel A spreadsheet application.

Microsoft Windows A windowing environment that is noted for its GUI, and for features such as multiple typefaces, desk accessories (such as a clock, calculator, calendar and notepad), and the capability of moving text and graphics from one application to another via a clipboard.

CX-Supervisor will run only under Microsoft Windows. DDE functions communicating with other applications supported by CX-Supervisor use Microsoft Windows as a basis.

Microsoft Word for Windows A word processing application.

Nesting To incorporate one or more IF THEN ELSE/ELSEIF ENDIF statements inside a structure of the same kind.

Network

1. Part of the PLC configuration, based on the device type. The number of Networks available is dependant on the device type.

2. A number of computers linked together with a central processing point known as a Server which is accessible to all computers. Networks affect CX-Supervisor in that further Network associated options are available if the computer is Network connected.

Non-Volatile
A point that is designated as 'non-volatile' is a point whose value is saved on disk and automatically reloaded when CX-Supervisor resumes execution.

NOT
A logic operator used to interrogate Boolean type points which produces the Boolean inverse of the supplied argument. An example of NOT is that if *a* is a statement and is 'FALSE', then NOT returns 'TRUE'. If *a* is a statement and is 'TRUE', then NOT returns 'FALSE'.

Object
In CX-Supervisor, an object can be text, graphics, a control, a bitmap, or ActiveX object as created in the development environment. A complex object can exist as a combination of two or more objects of any of the above types. Specifically, graphical objects can be categorised as a line, an arc, a polygon (including a square and rectangle), a round rectangle, an ellipse (including a circle), or a polyline. A control is essentially a complex graphic object and is specifically either a pushbutton, a toggle button, a slider, a trend graph, a rotational gauge or a linear gauge.

OLE-DB
OLE-DB is the underlying database technology, on which ADO relies. OLE-BD is designed to be the successor to ODBC.

Operand
The term used for constants or point variables.

Operator
A symbol used as a function, with infix syntax if it has two arguments (e.g. "+") or prefix syntax if it has only one argument (e.g. NOT). The CX-Supervisor script language uses operators for built-in functions such as arithmetic and logic.

OR
A logic operator used to interrogate Boolean type points. OR returns 'TRUE' if any of the supplied arguments are 'TRUE'. An example of OR is that if *a* is a statement and *b* is a statement, OR will return 'TRUE' if either *a* and *b* are 'TRUE'. If both statements return 'FALSE' then OR will return 'FALSE'.

Pages
The combination and manipulation of pages containing objects within projects forms the basis of CX-Supervisor. More than one page can exist for each project. The pages in a project provide the visual aspect of CX-Supervisor corresponding to a display with the objects contained in each page providing a graphical representation of the system being monitored.

Parameter Association
A Parameter Association enables values, either constant or stored in a point, to be passed to a Server Query.

Pixel
A single displayable point on the screen from which a displayed image is constructed. The screen resolution of the computer's Visual Display Unit (VDU) is defined by the number of pixels across and the number of pixels down (e.g. 1024 x 768).

See also SVGA mode and VGA mode.

PLC
Programmable Logic Controller.

Point variable
A point within the CX-Supervisor script language that stores a value or string assigned to that point.

Point
A point is used to hold a value of a predefined type - Boolean, Integer, Text, etc. The contents of a point may be controlled by an object or I/O mechanism such as PLC communication. The contents of a point may control the action or appearance of an object, or be used for output via an I/O mechanism.

See also Boolean type, Integer type, point variable, Real type and Text type.

Project
A CX-Supervisor application will consist of one or a number of pages linked together. The pages may contain passive or active graphics, text or animations, and may be grouped together logically to form a project. A project may consist of many pages, or simply a single page. Projects may be built and tested within the CX-Supervisor development environment, and run stand-alone under the CX-Supervisor run-time environment.

Only one project at a time may be open for editing within the CX-Supervisor development environment.

Real type
A type of point where the value of the point can be any number, including those containing a decimal point.

Recipe
A recipe is a set of pre-defined steps used to perform a particular task. A CX-Supervisor project may contain zero or more number of recipes. Recipes are defined in the development environment and executed, or downloaded, in the run-time environment.

Run-Time Environment
SCADA applications are run using the run-time environment of CX-Supervisor, following creation of the application in the CX-Supervisor development environment.

SCADA
Supervisory Control and Data Acquisition.

Server
A Server is the central processing point of a Network that is accessible to all computers. Networks affect CX-Supervisor in that further associated options are available if the computer Network is connected.

Server Application
An application that can be used to view or interact with, whilst currently within CX-Supervisor.

Statement
Within the CX-Supervisor script language, a statement is a command understood by the run-time environment. Statements are constructed of commands and arguments, which when combined, help to formulate a finished application to be used in the run-time environment.

String
The contents of a Text type point that can only contain literal alphanumeric characters. A string starts following an opening quotation mark, and ends before a closing question mark; in the example “`name = "spot"`”, the point “`name`” holds the string `spot`.

SVGA mode A mode of video display that provides 800 × 600 **pixel** resolution (or higher) with 16 or more colours and is supported on Super Video Graphics Adapter systems.

CX-Supervisor A SCADA software application which creates and maintains graphical user interfaces and communicates with PLCs and other I/O mechanisms.

Target Value An ingredient must specify a target value for its related point. This is the value to which the point will be set in runtime when the recipe is downloaded.

Taskbar An integral part of Microsoft Windows which allows Microsoft Windows based applications to be started. CX-Supervisor is run from the Taskbar.

Text Object In CX-Supervisor, a text object is a string on a page. Attributes such as typeface, point size, embolden, italicise, underline, left justify, flush right, and centre can be applied to enhance its presentation.

Text Type A type of point that holds a string.

Unicode A Multi-Byte Character Set, which not only includes European Characters like DBCS, but can also include global support including for Japanese, Chinese and Cyrillic fonts. However, Unicode is not supported on all Windows platforms.

Validation Code Recipe validation code is CX-Supervisor script language which is used to check point values before downloading a recipe.

VBScript A Visual Basic style scripting language supported by Microsoft's Windows Scripting Host.

VGA mode A mode of video display that provides 640 × 480 pixel resolution with 16 colours and is supported on Video Graphics Adapter systems.

Windows Desktop An integral part of Microsoft Windows which allows Microsoft Windows based applications to be started from icons and for all applications to be organised. CX-Supervisor can be run from Windows Desktop.

Windows Scripting Host A scripting engine supplied by Microsoft to run VBScript or JScript. See http://msdn.microsoft.com/scripting

Wizard Wizards are dialogs used by the CX-Supervisor development environment to take the user through complex operations in a simplified step-by-step process.

INDEX