

**CX-Server Lite
User Manual
Getting Started
Version 1.0**

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.



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

Note: Indicates procedures that are specific to Visual Basic.



© OMRON, 2000

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of OMRON.

All copyright and trademarks acknowledged.

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-Server Lite application and its ability to interface with OMRON CS, CV and C PLCs. It does not provide detailed information concerning the PLCs themselves, for this information the commercial manual for the device must be consulted.

This manual contains the following information:

- **Getting Started with CX-Server Lite:** This describes the CX-Server Lite software in general terms
- **Tutorial:** This is a quick tutorial for Excel and Visual Basic host applications.
- **Appendix A Component Properties:** This appendix summarises the available properties for the ActiveX components.
- **Appendix B Script Interface:** The Visual Basic script interface for the CX-Server communications control.

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

Warning:	Failure to read and understand the information provided in this manual may result in personal injury or death, damage to the product, or product failure. Please read each chapter in its entirety and be sure you understand the information provided in the chapter and related chapters before attempting any of the procedures or operations given.
-----------------	---

Table of Contents

CX-Server Lite	Page
<u>Getting Started with CX-Server Lite</u>	6
Welcome to CX-Server Lite	6
About this Manual	6
System Requirements	7
Installing/Uninstalling CX-Server Lite	8
The Help System, and How to Access it	8
Technical Support	10
Objects Overview	11
<u>Tutorial</u>	13
Step 1: Viewing PLC Data using Omron Graphical Controls	13
Step 2: Inserting PLC Data in Cells	16
Step 3: Adding Third Party Controls	16
Other Features	17
<u>Appendix A – Component Properties</u>	21
<u>Appendix B - Script Interface</u>	24
Functions	24
PLC Memory Functions	29
<u>Glossary of Terms</u>	30
<u>Index</u>	32

Getting Started with CX-Server Lite

This book introduces the CX-Server Lite application to a new user.

Welcome to CX-Server Lite

CX-Server Lite allows PLC data collected by CX-Server to be accessed from Microsoft Excel (97 and later) and Visual Basic (5.0 and later). It allows existing process data to be collected and analysed, plus it includes some graphical components allowing easy creation of simple MMI applications.

Included with CX-Server Lite is the CX-Server Runtime, plus a range of ActiveX components that can be dragged and dropped onto your Workbook or Form.

About this Manual

This manual helps a new user get started with CX-Server Lite, by describing the software installation and computer configuration, and by leading the user through the basics of CX-Server Lite. For the most up to date information see the On Line help or the release notes in the installed directory.

Separate OMRON manuals describe the related products; CX-Server, CX-Programmer and SYSMAC-SCS.

CX-Server Lite comes with a comprehensive on-line help system, which is designed to complement this manual, and provide a quick reference at any point in the CX-Server Lite application when the manual is not to hand. This general help system uses a fast 'hypertext' system which allows progressively more information to be obtained about any topic by selecting keywords within the descriptive text.

Throughout this manual, it is assumed that a working knowledge of Microsoft Windows is obtained, and that the user can:

- ◆ Use the keyboard and mouse.
- ◆ Select options from Windows menus.
- ◆ Operate dialog boxes.
- ◆ Locate, open and save data files.
- ◆ Edit, cut and paste text.
- ◆ Drag and drop.
- ◆ Start programs from the "START" button.

If Windows has not been used before, it is recommended that some time working with the Microsoft documentation is spent before using CX-Server Lite.

This introductory section deals with several important aspects of installing CX-Server Lite and setting it up for use. It is recommended that this entire section be read before installing the software.

System Requirements

CX-Server Lite operates on IBM compatible personal computers with 200MHz Pentium central processor. It is designed to run in the Microsoft Windows 95 / 98 and Windows NT V4.0 environment.

Note: CX-Server Lite is not guaranteed to be compatible with computers running Windows emulation (e.g. Apple Macintosh).

Hardware Requirements

The following configuration is the **minimum** system requirements for running CX-Server Lite

- ◆ IBM PC compatible 200MHz Pentium processor:
- ◆ 64 Mbytes of RAM,
- ◆ 30 Mbytes available hard disk space,
- ◆ 800 x 600 Super VGA display.

The **Recommended** minimum is:

- ◆ IBM PC compatible 400MHz Pentium II processor,
- ◆ 64 Mbytes of RAM,
- ◆ 50 Mbytes available hard disk space,
- ◆ 1024 x 768 Super VGA display.

Operating Systems and Environments

The operating systems on which this software will run are:

- ◆ Microsoft Windows 95 / 98,
- ◆ Microsoft Windows NT 4.0 (Service Pack 4 and later),

Containers in which this software will run are:

- ◆ Microsoft Excel 97 and later,
- ◆ Microsoft Visual Basic version 5.0 and later.

Interfaces to Hardware - PLC Communications

Interfaces to PLC hardware are via the CX-Server runtime system. Note that Temperature Controllers are not supported in this version.

Interfaces to Network Service Boards (NSBs) are achieved using Fins Gateway, which is supplied and supported as part of the CX-Server product.

Interfaces to Hardware - Peripherals

Interfaces to PC hardware (printers, graphics, keyboard, mouse, Ethernet etc.) are supported by drivers installed and supported by Windows.

Installing/Uninstalling CX-Server Lite

The CX-Server Lite software is supplied on CD-ROM and is installed easily from within Windows.

To install CX-Server Lite

- 1, 2, 3... 1. Close all programs.
2. Insert the CD labelled **CX-Server Lite** into your CD-ROM drive. If Autorun is enabled on your system, the installation starts automatically, otherwise see the README.TXT on the CD-ROM for instructions to launch manually.
3. Follow the instructions on the screen.

By default, CX-Server Lite is installed in **C:\Program Files\Omron**. Additional Omron applications will be installed in a subfolder under Omron. Common components are installed into **C:\Program Files\Common Files\Omron\Components**.

To uninstall CX-Server Lite

When you uninstall, your CX-Server Lite project data remains intact – uninstall only removes program files. It is recommended, however, that you copy and move any projects you saved in the CX-Server Lite folder. Save a copy of your projects in another location on your hard drive (such as your My Documents folder) before uninstalling CX-Server Lite.

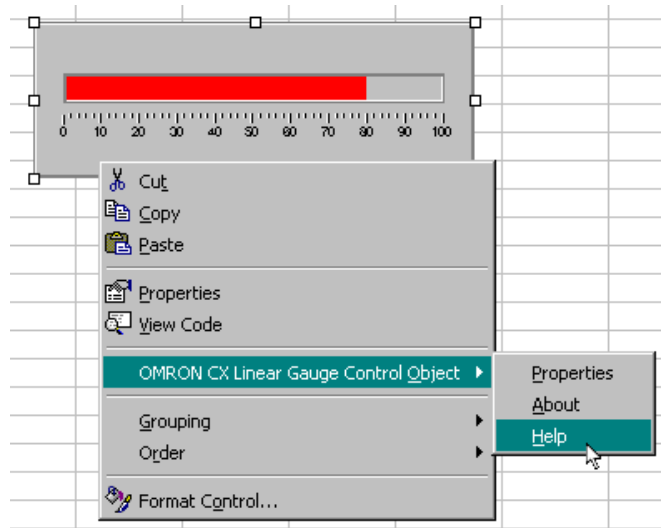
- 1, 2, 3... 1. From the **Start** menu, select **Settings – Control Panel**.
2. Double-click the **Add/Remove Programs**.
3. Click the **Install/Uninstall** tab.
4. From the list programs that you can remove, select **CX-Server Lite**.
5. **Click Add/Remove**.
6. At the prompt, select the **Remove** item and click **Next** and follow any further prompts.
7. Wait until the uninstall program indicates that the process is complete.

The Help system, and How to Access it

CX-Server Lite comes with a detailed help system. At any time while using the software, it is possible to get help on a particular point that is currently being worked on, or on general aspects of CX-Server Lite. This system is intended to complement the manual, by providing on-line reference to specific software functions and how to use them. This manual is designed to provide tutorial information and discuss the various facilities offered by CX-Server Lite.

Help Topics

Select the *Help* option on the *Object Properties* menu, which can be accessed by clicking on an object using the right mouse button and then selecting the option which shows the object name.



The help system provides a standard look-up dialog under the Contents tab showing the contents of the CX-Server Lite Help file. Double-click on an item to read the associated information.

Index

Use the following procedure to retrieve on-line help from the *Index* tab of the Help dialog.

- 1, 2, 3... 1. Select the *Help* option from the Object Properties Menu.
2. Select the *Index* tab.
3. Enter a text query into the first step field. The second step field is refreshed according to the query entered in the first step field.
4. Select an entry in the second step field and select *Display* pushbutton, or double-click on the index entry.
5. If an entry is linked to two or more topics, the names of the topics are displayed in the Topics Found dialog. Select a topic and choose the *Display* pushbutton or double-click in the topic.

Find

Use the following procedure to retrieve on-line help from the *Find* tab of the Help Topics dialog.

- 1, 2, 3... 1. Select the *Help* option from the Object Properties Menu.
2. Select the *Find* tab.
3. Enter a text query into the first step field. The second step field is refreshed according to the query entered in the first step field. Previous text queries can be retrieved by selecting from the drop down list in the first step field.

4. Select a word that matches the query – some words may be automatically selected. More than one word can be selected by pressing Shift and selecting another word to extend the selection or by pressing Ctrl and selecting another word to add to the selection. The third step field is refreshed according to the word or words selected. The number of topics found is shown at the bottom of the dialog.
5. Select a topic from the third step field and select the *Display* pushbutton, or double-click on the topic from the third step field. Select the *Clear* pushbutton to restart the find operation.

The Find operation can be enhanced by the use of the *Options* pushbutton and *Rebuild* pushbutton. Refer to *Microsoft Windows documentation* for further information.

About CX-Server Lite

CX-Server Lite Components include an About dialog accessible from the properties menu. The About dialog supplies technical reference information about the application such as version and copyright information. It also contains essential version number information that is required for obtaining technical support. The CX-Server Communications Control also includes details of the version of CX-Server installed.

In addition, a brief description of CX-Server Lite and the CX-Automation Suite can be accessed from the main help contents dialog.

Technical Support

If the installation instructions for this application have been followed, no difficulties should be encountered.

If a problem occurs, check that it does not relate to a fault outside CX-Server Lite, for instance, with external components. Check the following:

- ◆ The PC is working correctly,
- ◆ The external system or application is working correctly,
- ◆ The communications system is set up correctly,
- ◆ Any errors are cleared in the associated PLCs.

When Customer Services need to be contacted, keep the following details to hand. A clear and concise description of the problem is required, together with the exact text of any error messages.

Note: Use the About dialog to obtain the version number of the application.

Objects Overview

CX-Server Communications Control

This control provides a seamless interface between the CX-Server Lite host application (Excel, Visual Basic) and Omron's communication software, CX-Server. Note that the control is only visible when the host application is in the Design mode.

The Properties dialog enables you to select the project you wish to open. Clicking the **Open...** button will open the Open Project dialog allowing you to navigate to the appropriate file.

Clicking the Edit Project button opens the CX-Server Project Editor dialog. This dialog is supported by CX-Server and follows the standard Windows Explorer format. The left hand pane shows the tree structure for the project. By expanding the tree the associated PLCs and Points etc. can be reviewed and edited as necessary. New PLCs and points can be added by right clicking in the right hand pane and selecting **New** from the menu. Consult the associated help file for more detailed information on editing.

7 Segment

The 7 segment displays a numerical value in Binary, Decimal or Hexadecimal formats. Leading zeros and unused segments hidden. The colour of the segments and the display background can be set independently. The 7 Segment can not be used to set a value.

Display

The Display displays an analogue or text value. The Display only displays a value i.e. you can not set a value using this display.

LED Indicator

The LED functions as a coloured on/off indicator. The colour of the indicator and the display background can be set independently while its shape can be round or square. In the off state, the chosen indicator colour is dimmed.

Linear Gauge

The Linear Gauge displays an analogue value by filling a rectangle to represent the actual value as a proportion of its expected maximum. The rectangle can be filled from bottom to top (like a thermometer) or from left to right (like a progress complete bar). There is also a configurable scale, enabling intermediate values to be estimated. The Linear gauge will only display a value, you can not set a value with this gauge.

Rotational Gauge

The Rotational Gauge displays an analogue value, similar to a speedometer. An indicator needle rotates according to the value. There is a configurable scale, enabling intermediate values to be estimated. The Rotational gauge will only displays a value, you can not set a value with this gauge.

Rotary Knob

The Rotary Knob allows you to set an analogue value, similar to a volume knob. You can rotate the knob, e.g. by clicking and dragging the mouse, to set the pointer to a new position. There is a configurable scale, enabling intermediate values to be estimated. The pointer always reflects the current value e.g. on start-up, and will change position in response to an external influence.

Toggle

The Toggle allows you to toggle a Boolean bit between its 'On' and 'Off' state. This is as a switch that can be clicked to change its state. The current state is shown by the position of the switch. The switch position also reflects the current value e.g. on start-up, and will change position in response to an external influence.

Timer

The timer enables you to run a set of instructions repeatedly at regular intervals.

Tutorial

The following sections take you through the steps required to open your selected application, i.e. Excel or Visual Basic and create a working area. Using the short tutorial you can then continue and load a number of ActiveX objects, link them together and run a simulation.

As you become more practised in using CX-Server Lite you will find there is usually more than one way to perform an operation. The following procedures may not always be the quickest but have been written to show how the application works using the basic features.

If the ActiveX objects are not visible in the Visual Basic Toolbox they can be added as follows:





- 1, 2, 3... 1. Right click in the Toolbox and select the **Components...** option. This will open the Components dialog.
2. Find the CX-Server Lite controls in the list, all of which all start with **OMRON CX**, and tick each box.
3. Click the **OK** button. The objects are now displayed in the Toolbox.

Step 1: Viewing PLC Data using Omron Graphical Controls

Adding the Communications Control

Before the Graphical Controls objects of CX-Server Lite can communicate with a PLC the correct data source connections have to be set up for it. This is not necessary if the Graphical Control will be used stand alone and driven from script.

To add a Communications Control:

- 1, 2, 3...  1. Start the host application e.g. Microsoft Excel 97.
Tip: If you intend having a large number of components on your desktop it is recommended you run Excel in full screen mode.
2. Ensure the **Control Toolbox** and **CX-Server Lite** toolbars are shown by selecting them from the **View, Toolbar** menu.
Note: In Visual Basic, ensure the **Toolbox** is shown by selecting **Toolbox** from the **View** menu. 
-  3. Ensure the host application is in design mode for example, in Excel by clicking the Design Mode button in the **Control Toolbox**.
-  4. In the **CX-Server Lite** toolbar click the **Add CX-Server Communications Control** button. The Communications Control object is drawn in the default position - top left hand corner of the work area.

Note: In Visual Basic, select the required component from the **Toolbox** and draw a rectangle at the desired position.

Tip: Double clicking the toolbox button inserts the component with a default size.

5. Using Drag and Drop the object can now be repositioned in the work area. Note that the object will not be visible in run mode.

Connecting the Communications Control to a PLC

The first step is to create a CX-Server project file (.CDM file) or select one which has been created by another application (e.g. CX-Programmer or SYSMAC-SCS). This file contains PLC configuration data and symbolic definitions.

The following procedure takes you through the steps required to load an existing .CDM file or create a new one.

1, 2, 3...



1. Right click on the CX-Server communications control. In the popup menu, select the **OMRON CX Communications Control Object** option.

Note: In Visual Basic, the menu option is called **Properties**.



2. In the Communication Control Properties dialog 'Project' field enter or select the following.

- i. **To open an existing CX-Server project (.CDM) file:**

- ◆ Click the **Open...** button and in the Open Project dialog navigate to the file you wish to open.
- ◆ When you click the **Open** button, the full path name of the selected file will be entered into the Project field.

Caution: When sharing a CDM file with another applications it is important to realise that any changes that are made to the CDM file may affect the other applications.

- ii. **To create a new CX-Server project (.CDM) file:**

- ◆ Click the **New...** button and in the Create Project dialog navigate to the directory in which you wish to create the new file.
- ◆ In the File Name field, enter the desired file name. When you click the **Save** button, the full path name of the new file will be entered into the Project field.

3. Projects can be edited by clicking the **Edit Project...** button and then making the required changes from the Project Edit Dialog. This is a CX-Server runtime utility – invoke help from within it for details of how to add and configure PLCs and Points.
4. Click the **OK** button to complete the configuration.

The communications control is now ready to connect to CX-Server, and retrieve data from PLCs. This data can be accessed using script commands (see Appendix B), or by adding a Graphical Component.

Adding a 7 Segment Display

1, 2, 3...



1. With the host application in design mode, add a 7 Segment control.

2. Right click on the graphical component and from the popup menu select the **OMRON CX 7 Segment Control Object** option.

Note: In Visual Basic, the menu option is called **Properties**.



3. In the component properties dialog select the **Data Source** tab and enter the following information:

- ◆ **Server:** - Select the name of the communications control to be used. If only one has been added, it is selected automatically. If the list is empty then you need to add one first.
- ◆ **PLC:** - Select the required PLC. If the appropriate PLC is not in the list click the > button and select **Add PLC....** The Add PLC dialog is part of the CX-Server runtime. For further information on adding PLCs refer to the CX-Server User Manual or the online help.
- ◆ **Item:** - Select the point Item. If the appropriate item is not in the list click the > button and select **Add Item.....** The Point Editor dialog is part of the CX-Server runtime. For further information on adding Points refer to the CX-Server User Manual or the online help. The Item field will also accept physical addresses e.g. "DM100" instead of defining logical addresses.
- ◆ **Update Rate:** - Enter the rate, in seconds, at which the data is updated.



Note: This value should be chosen carefully. If the update rate is set low it will increase the volume of data being transferred, and may cause the system to slow down or stop responding.

4. Click **OK** to complete the connection.

Running the Application



Click the Mode button to change to 'Run' mode. The communications control will disappear and will connect to the PLC. After a short delay, depending on the PLC communications medium, the 7 Segment display will show the PLC value.

Note: In Visual Basic, click  to switch to run mode, and  to return to design mode.

Step 2: Inserting PLC Data in Cells

Step 1 shows PLC data in a graphical control, but the PLC data can also be inserted directly into cells within Excel. This could be useful for further numerical analysis, like averaging or statistical control.

Note In Visual Basic, there is no concept of cells, but the same technique could be used to set a Visual Basic variable with a PLC value.



Assuming Step 1 above has been completed:

- 1, 2, 3...**
1. Decide when the data should be updated:
 - i. For the user to control the data update, add a standard **Command Button** from the **Control Toolbox**.



- ii. For the value to constantly update on a regular interval, add the Omron timer control.



2. Double click the added object to access the script and add the line:


```
Cells(1, 1) = Comms1.Value("MyPointName")
```

where MyPointName is the logical name of the item to read.



3. Close the Visual Basic editor, and run the application as shown in Step 1. The cell A1 (that is row 1, column 1) will show the required data.

Step 3: Adding Third Party ActiveX Controls

Step 1 shows PLC data in an Omron graphical control, but the PLC data can be used by other ActiveX controls, like Graphical Control Libraries supplied by other manufacturers, or controls like Charts or Scroll Bars supplied with Microsoft products.

Assuming Step 1 above has been completed, the following steps show connecting the data to a standard Scroll Bar:

- 1, 2, 3...**
1. Add a standard **Scroll Bar** from the **Control Toolbox**.



2. Select and resize the buttons as required then drag and drop the buttons in the desired position.
3. Double-click on the scroll bar buttons. This will reposition the cursor in the code sheet at the following entry.

```
Private Sub Scrollbar1_Change()  
|  
End Sub
```

Note: In Visual Basic, the default name for scroll bars is **HScroll1** or **VScroll1**.



4. Add the following syntax. The additional command instructs any new Scroll Bar value to be written to the PLC.

```
Private Sub Scrollbar1_Change()  
    Comms1.Value("MyPointName") = Scrollbar1.Value  
End Sub
```



Run the application as shown in Step 1. The communications control will disappear and the 7 Segment display will show the PLC value. When you click on the scroll bar buttons the value is sent to the PLC. This new value is then shown on the 7 segment display. Note that the maximum value is limited to 100. This is the default value of the scroll bar buttons.

This example shows a control setting a PLC value. Third Party controls can also display PLC values. For this, the syntax would be (depending on actual control):

```
ControlName.Value = Comms1.Value("MyPointName")
```

This script could also be added to a button or Timer control, as explained in Step 2. Script functions are described further in Appendix B.

Other Features

The following sections provide a brief overview of some of the more advanced features available in CX-Server Lite.

Event Driven Routines

Many of the script examples in this manual use asynchronous communications, that is communications are carried out on demand without synchronisation with the rest of the system. Asynchronous communications can be easily used to quickly create solutions that are easy to understand. As a solution grows however, asynchronous communications can prove inefficient and produce unpredictable updating, which is difficult to debug because multiple scripts may be demanding the same data at the same time.

The CX-Server Communications Control provides facilities for synchronous communications, that is communications and data updating are synchronised. The **GetData** and **StopData** script commands (see Appendix B for full details) control the generation of **OnData** events on regular intervals, which can be used to efficiently drive multiple controls, and is easier to debug.

Example:

- 1, 2, 3... 1. Add an **OMRON CX-Server Communications Control** and two standard **Command Buttons**.
2. Double click the communications control to add the following script. Note the script is in the Event **OnData**:

```
Private Sub Comms1_OnData(ByVal PLC As String, ByVal  
Point As String, ByVal Value As Variant, ByVal  
BadQuality As Boolean)  
    If (Point = "MyPointName") Then  
        'Data is from this point  
        Cells(1, 1) = Value  
    End If  
End Sub
```

Every time **OnData** is called with data from the point **MyPointName** the value is written to the cell A1.

3. Double click the Command Buttons to add the following script:

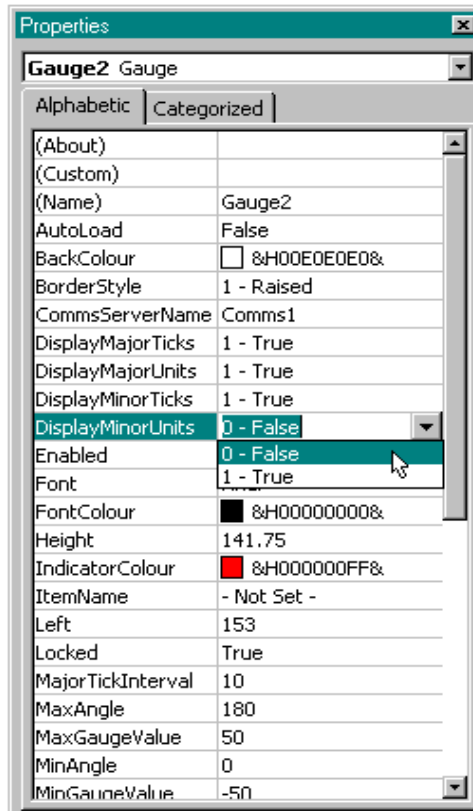
```
Private Sub CommandButton1_Click()  
    Comms1.GetData("MyPLC", "MyPointName", 1.0, 0)  
End Sub  
  
Private Sub CommandButton2_Click()  
    Comms1.StopData("MyPLC", "MyPointName")  
End Sub
```

Run the application. Click CommandButton1 to start creating OnData events every second. Note the cell A1 updating. Click CommandButton2 to stop the updating.

Advanced Properties

When working with Visual Basic the advanced properties dialog is normally displayed on the right of the work form, although it can be docked in any position. In Excel it is opened by right clicking on an object and selecting the Properties option from the pop up menu. The dialog allows you to scan through all the available options. Some options require you to enter specific information, others provide a choice of entries from a drop down menu.

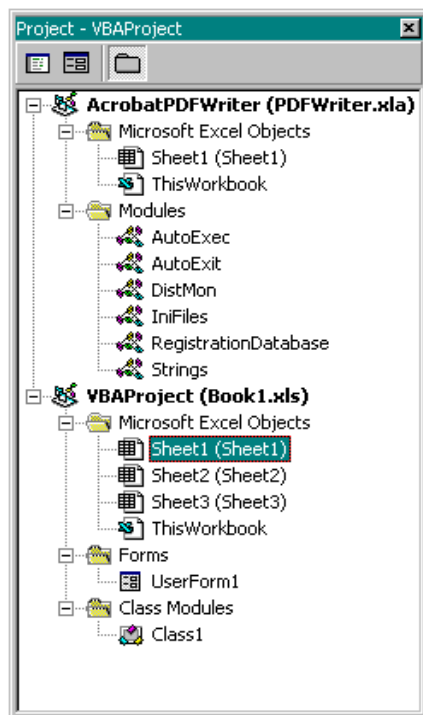
From the drop down menu at the top of the dialog select the object to be edited. This will display the full range of options available for that object, which can then be viewed either Alphabetically or Categorized. A full list of the options and their settings and ranges for the CX-Server Lite objects can be found in Appendix A – Component Properties.



Project Tree

Like Explorer the Project Tree provides a graphical representation of your application. In Visual Basic it is displayed to the right of the work form while in Excel it is shown on the left of the code sheet.

By expanding the tree you can see all your associated files and work sheets. It is possible to open any number of work sheets by simply double-clicking on them. Having multiple work sheets open in this way enables you to copy and paste between them saving you valuable time rewriting sections of code that already exist, and more importantly are known to work.



Controlling ActiveX Objects

A number of objects can be grouped together such as the 7 segment display and the spin buttons by selecting the objects you want to group while holding down the 'Shift' key as you select each object. When you have selected all the objects, right click on an object and select the 'Group' option from the drop down menu. Note however that objects must be ungrouped before their parameters can be edited.

Other drawing commands such as Bring to Front, Send to Back, Cut, Copy, Past etc. follow the standard windows conventions and are selected from the toolbar and/or drop down menus.

Appendix A

Component Properties

This appendix gives a list of the available properties. Each component supports a selection of these properties which can be set in design mode by using the properties dialog, or in the run time by using a Visual Basic script command – for example: - `Object1.Value = 10`

Property Title	Example Values	Description
About	None	Description of the object.
(Custom)	None	Opens the properties dialog for the object.
(Name)	Object Name	This is the system generated name for the object.
Autoload	True False	Switches the Autoload function On or Off. When set to On the ActiveX component value will autoload. True = On, False = Off.
AutoSize	0 to 1	Switches the fonts auto size option on or off. 0 = Off, 1 = On.
BackColor	■ &H00E0E0E0&	The code determines the background colour of the object. Click the browse button to display the colour palette.
BorderStyle	0 to 3	The value determines the visual appearance of the border that surrounds the object. 0 = None, 1 = Raised, 2 = Sunken, 3 = Single_Line
ButtonStyle	0 to 6	The value determines the button style. 0 = Toggle Switch, 1 = Colour Button, 2 = In/Out Button, 3 – Rotary Switch 4 = Rocker Switch, 5 = Indicator Button, 6 = Blank Button.
CommsServerName	Comms1 to n	This is the name of associated communications control.
DecimalPlaces	2	Number of decimal places for 7 Segment. Only applies when NumberBase = 10
DisplayFont	Arial	This is the system name of the font used for the display. Use the browse button to display the font dialog. Arial is the default.
DisplayFontColour	■ &H00000000&	The code determines the colour of the display font. Click the browse button to display the colour palette.
DisplayFormat	0 – Dec 1 – Hex 2 – Scientific	Determines the format used to display the analogue display information. 0 = Decimal, 1 = Hexadecimal, 2 = Scientific.
DisplayMajorTicks	False True	Shows or hides the major tick marks of the scale. True = Show, False = Hide.
DisplayMajorUnits	False True	Shows or hides the major units markers of the scale. True = Show, False = Hide.
DisplayMinorTicks	False True	Shows or hides the minor tick marks of the scale. True = Show, False = Hide.
DisplayMinorUnits	False True	Shows or hides the minor units markers of the scale. True = Show, False = Hide.
DisplayType	0 – Analogue 1 – Digital 2 – Text	Determines the type of display. 0 – Analogue: as a numeric value, 1 – Digital: as textual value, 2 – Text: as text.
Enabled	True False	Switches the object on or off. True = On, False = Off.
Font	Arial	This is the system name of the font used for the object title. Use the browse button to display the font dialog. Arial is the default.
FontColour	■ &H00000000&	The code determines the colour of the title font. Click the browse button to display the colour palette.
Height	141	Sets the overall height of the object in pixels.

Property Title	Example Values	Description
IndicatorColour	■ &H00FF0000&	The code determines the colour of the indicator. Click the browse button to display the colour palette.
ItemName	Output Flow No1	This is the reference name given to the object.
KnobBorderPercentage	25	This value determines the size of the knob border as a percentage of the overall size.
KnobColour	■ &H02C4723E&	The code determines the colour of the knob. Click the browse button to display the colour palette.
KnobStyle	0 – Integer 1 – Real	Determines how the numerical value of the knob is interpreted 0 = as an integer or 1 = as a real number.
LeadingZeros	False True	Turns On or Off the leading zero's of the display. False = On, True = Off.
LEDColour	■ &H00FF0000&	The code determines the default or inactivated colour of the LED indicator. Click the browse button to display the colour palette.
Left	153	Determines the position of the object from the left hand edge of the work area in pixels.
Locked	True False	Locks the actions of the ActiveX object. True = Locked, False = Unlocked.
MajorTickInterval	10	Sets the number of major tick intervals.
MaxAngle	180	Determines the length of the arc for rotary gauges. Range $\pm 360^\circ$.
MaxCharacters	6	Determines the number of digits to be included in the display.
MaxDecimalPlaces	2	Determines the position of the decimal point in the display.
MaxGaugeValue	50	Sets the maximum value of the gauge.
MaxValue	999.99	Sets the maximum value that can be shown by a display.
MinAngle	0	Determines the length of the arc for rotary gauges. Range $\pm 360^\circ$.
MinGaugeAngle	50	Sets the minimum angle of the gauge.
MinGaugeValue	10	Sets the minimum value displayed on the gauge.
MinorTickInterval	2	Sets the number of tick intervals between each major tick interval.
MinValue	-999.99	Sets the minimum value that can be shown by a display.
NumberBase	10	Number Base for 7 Segment e.g. 2 for Binary, 10 for Decimal, 16 for Hexadecimal.
NumberOfDigits	3	Determines the number of digits to be included in the display.
Orientation	0 or 1	Sets the orientation of the gauge – Horizontal or Vertical. 0 = Vertical, 1 = Horizontal.
PLCName	PLC1	This is the name given to the PLC, not its type.
PrintObject	True False	Determines if this ActiveX object will be shown on a printout of the work sheet. True = Printed, False = Not printed.
Round	0 or 1	Determines the shape of the LED indicator. 0 = Square, 1 = Round.
ScaleFont	Arial	This is the system name of the font used for the scale. Use the browse button to display the font dialog. Arial is the default.
ScaleFontColour	■ &H00000040&	The code determines the colour of the scale font. Click the browse button to display the colour palette.
SegmentColour	■ &H0000C000&	The code determines the colour of the display segments. Click the browse button to display the colour palette.
Shadow	True False	Adds or removes a shadow from the object giving it a raised appearance. True = Shadow On, False = Shadow Off.
State0Colour	■ &H000000FF&	The code determines the colour of the toggle switch in the 0 state. Click the browse button to display the colour palette.
State0Text	Text	This is the text indicating the 0 state of the switch, i.e. Off, Stopped, Halt etc.
State1Colour	■ &H0000FF00&	The code determines the colour of the toggle switch in the 1 state. Click the browse button to display the colour palette.

Property Title	Example Values	Description
State1Text	Text	This is the text indicating the 1 state of the switch, i.e. On, Running, Start etc.
StateFont	Arial	This is the system name of the font used to display the state of the switch on button. Use the browse button to display the font dialog. Arial is the default.
StateFontColour	■ &H00000000&	The code determines the colour of the title font. Click the browse button to display the colour palette.
Title	Text	This is the title that appears in the object.
TitlePosition	0 – Top 1 – Bottom	Determines where title will be placed in the object. Above or below the scale.
Top	230	Determines the position of the object from the top of the work area in pixels.
UnitPosition	0 – Inside 1 – Outside	Positions the dial units inside or outside the scale.
UpdateRate	10	The value sets the frequency, in seconds, at which the data passing to or from the object is updated.
Value	0	Determines the default value of the ActiveX object.
Visible	True False	Switches the object between visible and invisible in run mode. True = Visible, False = Invisible.
Width	98.25	Determines the overall width of the object in pixels.

Appendix B

Script Interface

The Script Interface defines the Visual Basic script interface for the CX-Server communications control.

Functions

Value	Function for getting and setting an area of memory in a PLC. This function allows logical names to be used. If an array is used, the first element is returned.
Values	Function for getting and setting an area of memory in a PLC. This function allows logical names to be used. If an array is used then a SAFEARRAY is returned with all values.
SetDefaultPLC	Function for setting the default PLC. This is primarily used when a project contains multiple PLCs.
OpenPLC	Opens the specific PLC for communications.
ClosePLC	Closes the specific PLC.
ReadArea	Function for reading a block of memory from the PLC.
WriteArea	Function for writing a block of memory to the PLC.
GetData	Function for starting OnData Events.
StopData	Function for stopping OnData Events.
OnData	Event for receiving notification of a change in data.
RunMode	Function for reading the current mode of the PLC.
TypeName	Function for reading the PLC type (e.g. CQM1H).
IsPointValid	Checks a point name is valid
PLC Memory Functions	A, AR, C, CIO, D, DM, DR, E, EM, G, GR, H, IR, LR, SR, ST, T, TC, TK, W. Functions for getting and setting the memory areas in the PLC.

Value

Reads the value of an address from a PLC, or writes a value to an address in a PLC. This function allows logical names.

Example 1 – Reading a value from the PLC using a logical name.

```
intVal = Comms1.Value("BoilerTemp")
or
intVal = Comms1("BoilerTemp")
```

In these examples, the PLC address associated with 'BoilerTemp' will be read from the PLC and stored in 'intVal'. "Value" is the default property for a CX-Server communications control and does not have to be specified.

Example 2 – Writing a value to the PLC using a logical name.

```
Comms1.Value("BoilerTemp") = 50  
or  
Comms1("BoilerTemp") = 50
```

In these examples, the value 50 will be written to the PLC address associated with 'BoilerTemp'. "Value" is the default property for a CX-Server communications control and does not have to be specified.

Further examples

```
Comms1("PLCName/DMO") = 1000  
Comms1("DMO") = 1000  
IntVal = Comms1("DM100")
```

These examples are physical addresses. A PLC name can optionally be specified otherwise the current default PLC will be assumed.

Values

Reads an array of values from a PLC, or writes an array of values to a PLC. This function allows logical names. If an array is used then a SAFEARRAY is returned with all values.

Example 1 – Reading an array of values from the PLC using a logical name.

```
SomeArray = Comms1.Value("BoilerTemps")
```

Example 2 – Writing an array of values to the PLC using a logical name.

```
Comms1.Values("BoilerTemps") = SomeArray
```

SetDefaultPLC

The 'SetDefaultPLC' function can be used to inform the script parser that a particular PLC is has been set as the default. Once a default PLC has been set, then it is not necessary (with some functions) to specify a PLC name. For example,

```
Comms1.SetDefaultPLC("MyPLC")  
intVal = Comms1.Value("BoilerTemp1")  
Comms1.Value("BoilerTemp1") = 75  
intVal = Comms1.Value("DM50")
```

Each 'Value' function above will access data in the PLC called 'MyPLC'.

Note: If there is only 1 PLC in the project then it is not necessary to call the 'SetDefaultPLC' function.

The first PLC in a project will automatically be set as the default PLC.

OpenPLC

Opens a PLC for communications. If no PLC is specified then the default PLC is opened.

Example 1:

```
Comms1.SetDefaultPLC("MyPLC")
Comms1.OpenPLC()
Comms1.DM(100) = 10
Comms1.DM(50) = 10
```

Example 2:

```
Comms1.OpenPLC("MyPLC")
Comms1.DM(100) = 10
```

ClosePLC

Closes a previously opened PLC. If no PLC is specified then the default PLC is closed.

Example:

```
Comms1.ClosePLC("MyPLC")
```

ReadArea

Reads a specified block of memory from a PLC.

Examples of synchronous read:

```
MyVariant = Comms1.ReadArea("MyPLC/DM0", 12, vbString, WaitUntilComplete )
MyVariant = Comms1.ReadArea("BoilerTemp", 10, vbInteger, WaitUntilComplete )
MyVariant = Comms1.ReadArea("BoilerTemp", 20)
```

In the first example, DM0 to DM11 will be read as characters (part of a string) from 'MyPLC' and will be stored in 'MyVariant'. The second example demonstrates that it is also possible to use a logical name for the start address, and that any VB variant types (such as vbInteger) can be used. The third example shows that the VB Variant type parameter is optional – if none is specified then vbInteger is assumed. In all these examples, WaitUntilComplete is specified (or assumed) so the script will wait for the read operation to complete before continuing to execute the next line.

Example of asynchronous read:

```
MyVariant = Comms1.ReadArea("BoilerTemp", 10, vbInteger, NoWaiting )
```

In this example, the Point 'BoilerTemp' will be read. The script will continue to execute the next line immediately, and when the data is read, it generates an OnData event.

WriteArea

Writes a block of memory to a specified area in a PLC.

Examples of synchronous write:

```
MyString = "TestString"  
Comms1.WriteArea "MyPLC/DM50", 10, MyString, WaitUntilComplete  
Dim newValue(2) As Long  
newValue(1) = 0  
newValue(2) = 1  
Comms1.WriteArea "BoilerTemp",2,newValue, WaitUntilComplete
```

In the first example, the contents of 'MyString' will be written into DM50 to DM54. Any additional data in 'MyString' will be ignored (i.e. if 'MyString' is 15 characters in length then the first 10 characters will be written to DM50 to DM54 and the remaining 5 characters will be ignored – {Note: each PLC address holds 2 characters}). The second example shows that a logical name can be used. In all these examples, WaitUntilComplete is specified (or assumed) so the script will wait for the write operation to complete before continuing to execute the next line.

Example of asynchronous write:

```
Comms1.WriteArea "Boiler Temp", 10, newValue, NoWaiting
```

In this example, the contents of 'newValue' will be written to the Point 'BoilerTemp'. The script will continue to execute the next line immediately, and the data write operation will continue in the background.

GetData

Starts asynchronous data reading of the specified point at the requested update rate.

Example

```
Comms1.GetData "MyPLC", "MyPoint", nUpdateRate
```

In this example, MyPoint in MyPLC would be read at the rate of nUpdateRate (seconds). Data is then sent to the OnData routine.

StopData

Stops asynchronous data reading of the specified point.

Example

```
Comms1.StopData "MyPLC", "MyPoint"
```

In this example, the reading of MyPoint in MyPLC would be stopped.

OnData

This event is sent back to the container (e.g. Excel) when GetData has been called and new data is available.

Example

```
Private Sub Comms1_OnData(ByVal PLC As String, ByVal Point As String,  
                          ByVal Value As Variant, ByVal BadQuality as Boolean)  
    TextBox1 = Point  
    Segment1 = Value  
End Sub
```

In this example, the CX-Server 7 Segment component is set to the value of the point and a text box is set to display the current point.

If BadQuality is set to True then the value may be inaccurate e.g. from a device which has been disconnected.

The OnData routine can be enhanced to include logical expressions on the incoming Point name and then update the correct graphical object etc. for example:

```
Private Sub Comms1_OnData(ByVal PLC As String, ByVal Point As String,  
                          ByVal Value As Variant, ByVal BadQuality as Boolean)  
    If Point = "MyPoint" then  
        Segment1 = Value  
    Else if Point = "MyOtherPoint" then  
        Cells(1,1) = Value  
    End if  
End Sub
```

RunMode

Reads the current operating mode of a PLC (Stop/Program, Debug, Monitor, Run), where 0=Stop/Program mode, 1=Debug mode, 2=Monitor mode and 4=Run mode.

Example

```
intMode = Comms1.RunMode("MyPLC")
```

In this example, the operating mode would be read from 'MyPLC' and stored in 'intMode'. If 'MyPLC' was in 'Monitor' mode then 'intMode' would be set to the value 2.

TypeName

Reads the PLC model name of a PLC (e.g. C200H, CQM1H, CVM1 etc).

Example

```
strPLCType = Comms1.TypeName("MyPLC")
```

In this example, the PLC model type will be read from 'MyPLC' and will be stored in 'strPLCType'.

IsPointValid

Checks if a Point name has been defined in the CX-Server project file.

Examples

```
bValid = Comms1.IsPointValid("MyPoint")
```

```
bValid = Comms1.IsPointValid "MyPoint", "MyPLC"
```

In both examples, the boolean variable *bValid* is set True if the point "MyPoint" has been defined.

PLC Memory Functions

(A, AR, C, CIO, D, DM, DR, E, EM, - G, GR, H, IR, LR, SR, ST, T, TC, TK, W)

All PLC memory functions (e.g. A, AR, D, DM etc.) work in exactly the same way. The following examples use the DM function to get and set the value of a DM address in a PLC.

Example 1

```
intVal = Comms1.DM(100)
```

In this example, the contents of DM100 will be read from the PLC and stored in 'intVal'.

Note: These examples assume there is only 1 PLC in the CX-Server project file, or that the 'SetDefaultPLC' function has been used to select the required PLC. Refer to the 'SetDefaultPLC' function for details about using script with multiple PLCs in the project.

Example 2

```
Comms1.DM(100) = 75
```

In this example, the value 75 will be written to DM100 in the PLC.

Bit addressing, that is accessing data from individual memory bits, is also supported by these memory areas: IR, AR, HR and CIO.

Example 3

```
intVal = Comms1.IR("100.2")
```

In this example, the status of bit IR100.2 (i.e. bit 2 of IR100) will be read from the PLC and stored in 'intVal' (e.g. 'value' will be set to TRUE or FALSE).

Example 4

```
Comms1.IR("100.2") = True
```

In this example, bit IR100.2 (i.e. bit 2 of IR100) in the PLC will be set to True. Note that use of the quotes is optional, but is required to differentiate between 100.1 and 100.10

Glossary of Terms

ActiveX	A component technology developed by Microsoft allowing components to communicate with applications .
Application	A software program that accomplishes a specific task. Examples of applications are SYSMAC-SCS , SYSMAC-CDM , Microsoft Word and Microsoft Excel .
Bitmap	The representation of an image stored in a computer's memory. Each picture element (pixel) is represented by bits stored in the memory.
Communications Driver	The relevant communications management system for OMRON PLCs in conjunction with Microsoft Windows , providing facilities for other software to maintain PLC device and address information and to communicate with OMRON PLCs and their supported network types.
Event	User action, e.g. mouse click or System action, e.g. timer tick which may cause a script to execute.
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.
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 <i>input</i> from) and is destined (is <i>output</i> to) to the internal computer memory, a PLC or a target application.
Icon	Pictorial representations of computer resources and functions.
Microsoft Excel	A spread sheet application .
Microsoft Windows	A windowing environment for MS-DOS computers, 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-Server Lite will run only under Microsoft Windows.
Microsoft Word	A word processing application .
OLE	Object Linking and Embedding. Used to transfer and share information between Microsoft Windows based applications and accessories.
PC	Abbreviation for Personal Computer.
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. 1024x768).
PLC	Abbreviation for Programmable Logic Controller.

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 DDE . The contents of a point may control the action or appearance of an object, or be used for output via an I/O mechanism.
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.
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.

Index

A

About CX-Server Lite · 10
About this Manual · 6
ActiveX Objects
 7 Segment · 11
 Display · 11
 LED Indicator · 11
 Linear Gauge · 11
 Rotary Knob · 12
 Rotational Gauge · 11
 Time · 12
 Toggle · 12
Adding a 7 Segment Display · 15
Adding the Communication Control · 13
Adding Third Party ActiveX Controls · 16
Advanced Properties · 19
Available Properties · 21

C

Component Properties · 21
Connecting CX-Server Lite to a PLC · 14
Controlling ActiveX Objects · 20

E

Event Driven Routines · 17

G

Getting Started with CX-Server Lite · 6
Glossary of Terms · 30

H

Hardware Requirements · 7
Help Topics · 9

I

Inserting PLC Data in Cells · 16
Installing/Uninstalling CX-Server Lite · 8
Interfaces to Hardware - Peripherals · 7
Interfaces to Hardware – PLC Communications · 7

O

Objects Overview · 11
Operation Systems and Environments · 7
Other Features · 17
 Advanced Properties · 19
 Controlling ActiveX Objects · 20
 Event Driven Routines · 17
 The Project Tree · 20

P

Project Tree · 20

R

Running an Application · 15

S

Script Interface · 24
 Functions · 24
 PLC Memory Functions · 28
Script Interface Functions
 GetData · 27
 OnData · 27
 ReadArea · 26
 RunMode · 28
 SetDefaultPLC · 25
 StopData · 27
 TypeName · 28
 Value · 24
 Values · 25
 WriteArea · 26
System Requirements · 7

Hardware Requirements · 7
Interfaces to Hardware · 7
Operating Systems and Environments · 7
Peripherals · 7
PLC Communications · 7

T

Technical Support · 10
The Help System, and How to access it · 8
Tutorial · 13
 Step 1
 Viewing PLC Data using Omron Graphical Control
 · 13
 Step 2
 Inserting PLC Data in Cells · 16
 Step 3
 Adding Third Party ActiveX Controls · 16

V

Viewing PLC Data using Omron Graphical Control
 · 13

W

Welcome to CX-Server Lite · 6