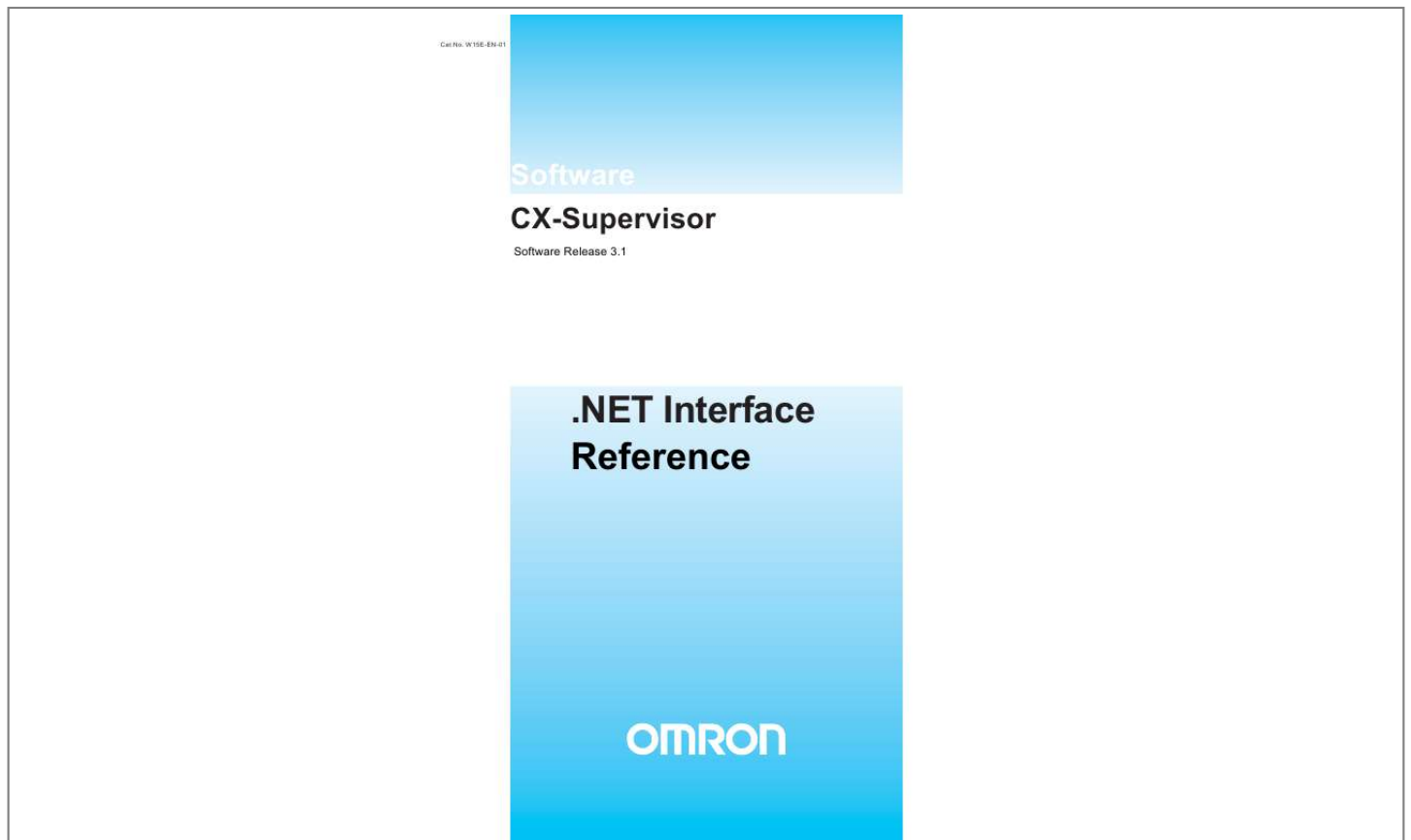




Your PDF Guides

You can read the recommendations in the user guide, the technical guide or the installation guide for OMRON CX-SUPERVISOR 3.1. You'll find the answers to all your questions on the OMRON CX-SUPERVISOR 3.1 in the user manual (information, specifications, safety advice, size, accessories, etc.). Detailed instructions for use are in the User's Guide.

User manual OMRON CX-SUPERVISOR 3.1
User guide OMRON CX-SUPERVISOR 3.1
Operating instructions OMRON CX-SUPERVISOR 3.1
Instructions for use OMRON CX-SUPERVISOR 3.1
Instruction manual OMRON CX-SUPERVISOR 3.1



[You're reading an excerpt. Click here to read official OMRON CX-SUPERVISOR 3.1 user guide](http://yourpdfguides.com/dref/3692078)
<http://yourpdfguides.com/dref/3692078>

Manual abstract:

W15E-EN-01 Software CX-Supervisor Software Release 3.1 .NET Interface Reference Trademarks and copyrights Notice Notice OMRON products are manufactured for use by a trained operator and only for the purposes described in this manual. The following conventions are used to classify and explain the precautions in this manual. Always heed the information provided with them. Note: Indicates information of particular interest for efficient and convenient operation of the product. Caution: Indicates information that, if not heeded, could possibly result in minor or relatively serious injury, damage to the product, or faulty operation. Warning: Indicates information that, if not heeded, could possibly result in serious injury or loss of life. Trademarks and copyrights CX-Supervisor is a registered trademark of OMRON. All other product names, company names, logos or other designations mentioned herein are trademarks of their respective owners.

Copyright Copyright © 2010 OMRON 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. 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. @@@@7 Referencing the DLL

.....

.....

....

....

.....

.....

....

....

.....

.....

....

....

.....

.....

....

....

7 SECTION 2 Examples

.....

....

....

.....

.....

....

....

.....

.....

....

....

9 2-1 2-2 2-3 VB .NET

.....

....

....

.....

.....

....

....

.....

.....

....

....

.....

.....

.....
.....
.....

.....
... *9 C#.NET* ..

.....
.....

.....
.....
.....
.....

.....
.....
.....
.....

.....
.....
.....
.....

.....
.....
.....
.....

..... *9 VB .NET and ASP.NET* ..

.....
.....
.....

.....
.....
.....
.....

.....
.....
.....
.....

.....
.....
.....
.....

..... *9 SECTION 3 Class Reference* ...

.....
.....
.....

.....
.....
.....
.....

.....
.....
.....

..... *11 3-1 PointMngt*

.....
.....
.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

... *11 3-1-1 ListGroups..*

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

... *11 3-1-2 ListPoints ..*

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.... *11 3-1-3 SetValue*

.....

.....

.....

.....

.....
.....
.....
.....

.....
.....
.....
.....

.....
.....
.....
.....

. 11 3-1-4 GetValue

.....
.....
.....

.....
.....
.....
.....

.....
.....
.....
.....

.....
.....
.....
.....

..... 11 3-1-5 IsValidPoint ..

.....
.....
.....

.....
.....
.....
.....

.....
.....
.....
.....

.....
.....
.....
.....

. 12 3-1-6 IsValidGroup

.....
.....
.....

.....
.....
.....
.....

.....
.....

.....
.....
.....
.....
.....
.....

.. 12 3-1-7 *GetPointData* ...

.....
.....

.....
.....
.....
.....

.....
.....
.....
.....

.....
.....
.....
.....

.. 12 3-1-8 *BrowsePoints* ...

.....
.....

.....
.....
.....
.....

.....
.....
.....
.....

.....
.....
.....
.....

.. 12 *ApplicationMngt...*

.....
.....

.....
.....
.....
.....

.....
.....
.....
.....

.....
.....
.....
.....

.....
.....
.....
.. 13 3-2-1 GetErrorString.....

.....
.....
.....

.....
.....
.....

.....
.....
.....

.....
.....
.....

. 13 3-2-2 Restart

.....
.....
.....

.....
.....
.....

.....
.....
.....

.....
.....
.....

. 13 3-2-3 GetProjectName

.....
.....

.....
.....
.....

.....
.....
.....

.....
.....
.....

..... @@@@The interface can be divided in to the following areas of functionality: Point monitoring and data gathering Application/System management Alarm monitoring and control Script execution Error management The reference section of this document provides full details of the classes and methods which implement this functionality. 1-3 Referencing the

DLL The interface is exposed through the file SCSRUNLib.dll which is located in the CX-Supervisor installation directory. In order to use the interface classes you must first add a reference to it in your project and then insert an appropriate using or Imports declaration in your class.

7 Referencing the DLL SECTION 1 Introduction 8 VB .NET SECTION 2 Examples SECTION 2 Examples These code snippets are provided as an example of how to use the CXSupervisor .NET Interface. 2-1 VB .NET ' declare variables CErrorMngtClass errorMngr = new CErrorMngtClass(); object list; ' get list of errors errorMngr.GetErrorLog(out list); ' convert list to string array string[] errors = (string[])list; ' use error log... 2-2 C# .NET // declare variables CPointMngtClass pointMngr = new CPointMngtClass(); object list; // get list of points pointMngr.

ListPoints(out list); // convert list to string array string[] points = (string[])list; // use points list... 2-3 VB .NET and ASP.

NET The source code for the CX-Supervisor Standard Web Pages is installed along with CX-Supervisor. It can be opened and run using Visual Web Developer 2008. The Express edition of this software is available as a free download from Microsoft. 9 VB .NET and ASP.

NET SECTION 2 Examples 10 PointMngt SECTION 3 Class Reference SECTION 3 Class Reference 3-1 PointMngt Allows the client to manage the acquisition of point information and the reading and writing of point data. 3-1-1 ListGroups Retrieves a list of all the point groups contained within a CX-Supervisor application. ListGroups(ByRef pGroups As Object) Parameters Description pGroups An array of strings representing the names of all the groups 3-1-2 ListPoints Retrieves a list of all the points that are part of the given group. ListPoints(ByVal Object) szName As String, ByRef pPoints As Parameters Description szName pPoints Name of Group Receives an array of strings representing the names of all the points contained in the group 3-1-3 SetValue This method sets a point to a specified value. In circumstances where the point can not be set to a value (i.e. More than allowed maximum) the value returned in the retVal parameter will differ from that specified and represent the actual value the point was set to. SetValue(ByVal szName As String, ByVal varValue As Object, ByRef retVal As Object) Parameters Description szName varValue retVal Name of Point Value the point is to be set to The actual value the point was set to 3-1-4 GetValue Reads the current value of a point. GetValue(ByVal szName As String, ByRef retVal As Object) Parameters Description szName retVal Name of Point Receives the current value of point 11 PointMngt SECTION 3 Class Reference 3-1-5 IsValidPoint Determines whether the point is valid. IsValidPoint(ByVal szName As String, String, ByRef retVal As Integer) Parameters Description szName szGroup retVal Name of Point Name of Group Receives true or false ByVal szGroup As 3-1-6 IsValidGroup Determines whether the group is valid.

IsValidPoint(ByVal Integer) Parameters Description szName retVal Name of Group Receives true or false szName As String, ByRef retVal As 3-1-7 GetPointData Retrieves the metadata of a specific point GetPointData(ByVal szName As String, ByRef vartype As UShort, ByRef pDescription As Object, ByRef bReadOnly As Integer, ByRef iArraySize As Integer, ByRef Value As Orings representing the names of all the alarms 3-3-3 GetAlarmData Retrieves the metadata of a specific alarm. GetAlarmData(ByVal szName As String, ByRef pType Object, ByRef pAuto As Integer, ByRef pDescription Object, ByRef pPriority As Object, ByRef pStatus Object, ByRef pDateTime As Object, ByRef pMessage Object) Parameters Description szName pType pAuto Name of the Alarm The type of Alarm "simple", "deadband", "rateofchange" Indicates Automatically acknowledged As As As pDescription The description of the alarm pPriority pStatus pDateTime pMessage The priority of the alarm "Highest", "High", "Medium", "Low", "Lowest" The current status of the alarm "Inactive", "Active", "Acknowledged" The time and date the alarm entered it's current state The message shown 15 AlarmMngt SECTION 3 Class Reference 3-3-4 AcknowledgeAlarm Acknowledge an alarm. AcknowledgeAlarm(ByVal szName As String, ByVal szUser As String) Parameters Description szName szUser Name of Alarm User who acknowledged alarm 3-3-5 AcknowledgeAllAlarms Acknowledge all un-acknowledged, active alarms.



[You're reading an excerpt. Click here to read official OMRON CX-SUPERVISOR 3.1 user guide](http://yourpdfguides.com/dref/3692078)
<http://yourpdfguides.com/dref/3692078>

AcknowledgeAllAlarms(ByVal *szUser* As String) Parameters Description *szName* *szUser* Name of Alarm User who acknowledged alarms 3-3-6

BrowseAlarms This method retrieves a filtered list of the alarms contained within a CXSupervisor application. *BrowseAlarms*(ByVal *szFilter* As String, ByVal *szPriorityFilter* As String, ByRef *pAlarms* As Object) Parameters Description *szFilter* Free format filter eg. A* *szPriorityFilt* Alarm priority filter. Eg. "High". If string empty all types are er returned *pAlarms* Receives an array of strings representing the names of all the alarms 3-3-7

GetAlarmLog This method provides the ability to get a list of all the alarm log entries. The returned array strings each delimited by tabs will provide time, message and status information.

The list matches the order of entries in the log. @@@@GetScriptParameters(ByVal *pParamList* As Object) Parameters Description *szName* *pParamList* The name of the script Receives an array of strings representing the data types of the parameters *szName* As String, ByRef 3-5 3-5-1

ErrorMngt Allows the client to access to the error log. *GetErrorLog* Get a list of all the error log entries. *GetErrorLog*(ByRef *pErrors* As Object) Parameters Description *pErrors* Receives and array of strings representing the all of the error log entries (date/time and message) 17

ErrorMngt SECTION 3 Class Reference 18

Revision history Revision history A manual revision code appears as a suffix to the catalog number on the front cover of the manual. Cat. No. W15E-EN-01 The following table lists the changes made to the manual during each revision. The page numbers of a revision refer to the previous version.

Revision Date code 01 Sept. 2010 Revised content First version in the standard Omron format.

19 Revision history 20 Authorized Distributor: Cat. No. W15E-EN-01 Note: Specif cations subject to change without notice. Printed in Europe .



[You're reading an excerpt. Click here to read official OMRON CX-SUPERVISOR 3.1 user guide](http://yourpdfguides.com/dref/3692078)
<http://yourpdfguides.com/dref/3692078>