



Your PDF Guides

You can read the recommendations in the user guide, the technical guide or the installation guide for OMRON CX-SUPERVISOR. You'll find the answers to all your questions on the OMRON CX-SUPERVISOR 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
User guide OMRON CX-SUPERVISOR
Operating instructions OMRON CX-SUPERVISOR
Instructions for use OMRON CX-SUPERVISOR
Instruction manual OMRON CX-SUPERVISOR

Cat. No. W9E EN01

Software Release 2.0

CX-Supervisor

SCRIPT REFERENCE

OMRON



[You're reading an excerpt. Click here to read official OMRON CX-SUPERVISOR user guide](http://yourpdfguides.com/dref/2886838)

<http://yourpdfguides.com/dref/2886838>

.....

.....

.....

.....

.....

.....

.....

.....

..... 3 Chapter 3 Scripts .

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.. 7 Object

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

...7 Page ..

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

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

.....
.....
..7 Project...
.....

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

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

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

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

.....
...7 Chapter 4 CX-Supervisor Script Language..
.....

.....
.....

.....
.....

.....
... 9 Points

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

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

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

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

.....
.....
...10 Logic and Arithmetic..
.....

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

.....
.....

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

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

.....
.....

...12 Control Statements .

.....

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

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

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

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

...15 Subroutines

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

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

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

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

.....22 Punctuation .

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

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

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

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

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

.....23 Indirection within Script Commands and Expressions

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

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

.....26 Point Arrays within Script Commands and Expressions

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

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

.....
.....27 Using Aliases...

.....
.....

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

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

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

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

.....28 Chapter 5 VBScript Language reference

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

..... 31 List of features ..

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

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

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

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

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

...31 Chapter 6 Functions and Methods.

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

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

.. 35 Object Commands ...

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

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

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

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

..39 Page Commands ...

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

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

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

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

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

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

.... 143 Script Interface

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

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

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

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

.. 143 Functions.....

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

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

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

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

.....
.....

.... 143 Appendix B CX-Server Communications Control .

.....

.... 145 Component Properties.

.....
.....

.....
.....

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

145 Appendix C JScript Features.....

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

151 Page (viii) Revision 2.0 OMRON CX-Supervisor Script Language Appendix D Obsolete Features

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

..... 153 Glossary of Terms ..

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

.... 161 Index.

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

.. 169 Revision 2.0 Page (ix) OMRON CHAPTER 1 - Introduction CHAPTER 1 Introduction This reference manual describes the script language syntax as a supplement to the CX-Supervisor User Manual. It provides detailed definition of the syntax of CX-Supervisor scripts that drive project, page, object actions and CX-Supervisor expressions as used by objects and scripts. Typographic conventions used in the examples in this reference manual are as follows: Script commands and reserved words are shown in the preferred case, which may be either lower-, upper- or mixed-case. Points are shown in lower-case. Objects are shown in upper-case. The following terms are used in this reference manual: Application. A set of files, containing an executable file, that carry out certain tasks.

This reference manual refers to the ion has an inferred "==" TRUE". Sometimes Boolean expressions can be difficult to read e.g. "bMyFlagPoint" or "BitMask & 0x80. It can help maintenance if this "==" TRUE" is explicitly specified e.g. "bMyFlagPoint == TRUE" or "BitMask & 0x80 == TRUE". When using Boolean operators (e.g. ==, !=, &&, ||, !) never mix tests for Boolean and non Boolean operands.

For example never use "bMyFlagPoint == 1" or "bMyFlagPoint == 0".



[You're reading an excerpt. Click here to read official OMRON CX-SUPERVISOR user guide](http://yourpdfguides.com/dref/2886838)
<http://yourpdfguides.com/dref/2886838>

Instead always test using the correct Boolean constant i.e. "TRUE" or "FALSE" for CXSupervisor scripts, or "True" and "False" when using VBScript. On Condition scripts are only executed when the expression is TRUE. Sometimes this leads to peculiar results, for example using \$Second as it will be executed when \$Second changes to 59, and to 1 but not when it changes to 0. To execute a condition script any time a point changes, force the expression to always evaluate to TRUE for example "\$Second || TRUE". This works because the \$Second forces the expression to be tested when the point changes, but the || TRUE means the test will return TRUE regardless of the value of the point. Use array points in On Condition expressions with caution. The expression "MyArray[3] == 1" does not mean "execute every time the third element changes to 1".

It means execute when any element of MyArray changes and the third element happens to be 1 Using an array point without any index is the same as specifying element 0 i.e. MyArray actually means MyArray[0] == 1 Note: Note: Note: Note: Revision 2.0 Page 5 CHAPTER 2 - Expressions OMRON Page 6 Revision 2.0 OMRON CHAPTER 3 - Scripts CHAPTER 3 Scripts A CX-Supervisor script is a simple programming language used to manipulate points.

Scripts can be created at different levels, at object level, page level or project level. Although the script code can be applied to all levels of script, there are subtle differences, described in the following paragraphs. Object If a script is executed as a runtime action of an object, then the script can affect the object of the action, or any other, depending on the actual content of the script. Page Page scripts are concerned with manipulating points and graphical objects that are used or included within that page. In other words page scripts are used to drive a number of actions on the occurrence of a particular event.

These actions may manipulate several graphical objects on one page. Project Scripts can be applied to a project to manipulate points. These scripts are associated with events that occur throughout the whole operating session Revision 2.0 Page 7 CHAPTER 3 - Scripts OMRON Page 8 Revision 2.0 OMRON CHAPTER 4 CX-Supervisor Script Language CHAPTER 4 CX-Supervisor Script Language This chapter describes the CX-Supervisor script language syntax.

It provides a detailed definition of the syntax of CX-Supervisor scripts that drive project, page and object actions, and CX-Supervisor expressions as used by objects and scripts. In conjunction with the script functions and methods described in Chapter 6, the CX-Supervisor script language provides a very powerful, compiled, fast and full featured programming language. The following table describes the script language syntax at a glance. Function Name &, !, ^, <<, >>

(objects) (points) +, -, *, /, %, =, ++, -<, >, <=, >=, ==, != AND CALL DO LOOP WHILE UNTIL EXIT DO FALSE FOR TO STEP NEXT EXIT FOR IFTHEN ELSE\ELSEIF ENDIF OR NOT REM RETURN SELECT CASE/END SELECT TRUE Function Type bitwise operators statement statement arithmetic operators relational operators logical operators statement statement Boolean state statement statement logical operators logical operators statement statement statement Boolean state Type All OP All All All All All Scr Scr Scr Scr All All Scr Scr Scr Scr Remarks Applies bitwise expressions Specifies an object name for modification or test. Specifies a point name for modification or test.

Applies arithmetic expressions. Applies relational expressions. Applies logical expressions. Call a subroutine Script segment to be repeated Applies Boolean expression. Script segment to be repeated Applies a test to a script. Applies logical expressions. Applies logical expressions. Remarks on line or lines of script. Stops sequential execution of script. Applied to complex tests.

Applies Boolean expression. The 'Type' column refers to the types of script and expression the function can be applied to. 'All' refers to both expressions and scripts. 'Scr' refers to scripts only. 'OP' refers to Object and Page scripts only. Revision 2.0 Page 9 CHAPTER 4 CX-Supervisor Script Language OMRON Points Basic Point Assignment Syntax pointname = expression Remarks Argument pointname expression Description The point name to be assigned a value. The value to be assigned to pointname. The expression may be of type Boolean, Integer, Real or Text. Typical Examples count = 100 The Integer or Real point 'count' is assigned the value 100.

result = TRUE The Boolean point 'result' is assigned the state 'TRUE'. name = "Valve position" The Text point 'name' is assigned the associated text, contained within quotation marks. Note: When assigning Real (floating point) values to an Integer point the assignment uses the 'Symmetrical Rounding Down' (towards 0) standard. This means a value of 4.1 would be assign a value 4.

A value of -4.1 would assign a value of -4. References Refer to chapter 4, Punctuation for details of the use of quotation marks. Page 10 Revision 2.0 OMRON CHAPTER 4 CX-Supervisor Script Language Further Point Assignment Syntax pointname = expression Remarks Argument pointname expression Description The point name to be assigned a value.

The value to be assigned to pointname. The expression may be of type Boolean, Integer or Real and can include other points, logical or arithmetical expressions. Mathematical precedence is applied as follows: · Parenthesis (highest). · Unary minus and NOT logical operator. · Multiplication, division and modulus. · Addition and subtraction. · Greater than, less than, greater than or equal to, and less than or equal to relational operators. · Shift Left (SHL) and Shift Right (SHR). · Equal to and not equal to relational operators. · Bitwise AND, XOR, OR.

· AND logical operator, OR logical operator (lowest). Typical Examples lift = height + rate/5.0 The Integer or Real point 'lift' is assigned the value calculated by the value of point 'rate' divided by 5, plus the value of point 'height'. Precedence can be changed by the introduction of parenthesis. lift = lift - 0.2 The Integer or Real point 'lift' is assigned the value calculated by the current value of point 'lift' minus 0.2. distance = distance * time The Integer or Real point 'distance' is assigned the value calculated by the current value of point 'distance' multiplied by point 'time'. References Refer to chapter 4, Logic and Arithmetic for details of the use of arithmetic and logic functions. Refer to chapter 4, Punctuation for details of the use of parenthesis.

Revision 2.0 Page 11 CHAPTER 4 CX-Supervisor Script Language OMRON Logic and Arithmetic Arithmetic Operators Syntax pointname = expression Remarks Argument pointname expression Description The point name to be assigned a value based on an arithmetical expression.



[You're reading an excerpt. Click here to read official OMRON CX-SUPERVISOR user guide](http://yourpdfguides.com/dref/2886838)
<http://yourpdfguides.com/dref/2886838>

The value to be assigned to pointname. The expression may include the following operators with points and constants: · Addition '+'. · Subtraction '-'.
· Multiplication '*'. · Division '/'. · Modulus '%'. · Increment '++'. · Decrement '--'.

Typical Examples result = 60 + 20/5 The Integer or Real point 'result' is assigned the value calculated by the value of 20 divided by 5, plus 60. lift = height + rate/5.0 The Integer or Real point 'lift' is assigned the value calculated by the value of point 'rate' divided by 5, plus the value of point 'height'. Precedence can be changed by the introduction of parenthesis. References Refer to chapter 4, Punctuation for details of the use of parenthesis. Bitwise Operators Syntax pointname = expression or IF expression or DO WHILE expression Page 12 Revision 2.0 OMRON or DO UNTIL expression CHAPTER 4 CX-Supervisor Script Language Remarks Argument pointname expression Description The pointname to be assigned a value based on the bitwise operation. The value to be assigned to pointname, or to be evaluated as a Boolean expression. The expression can include the following operators with points and constants: · Bitwise AND, 'BITAND' or '&'. · Bitwise OR, 'BITOR' or '|'.
· Bitwise XOR, 'XOR' or '^'. · Bitwise Shift Left, 'SHL' or '<<'. · Bitwise Shift Right, 'SHR' or '>>'. Typical Examples MSB = value & 128 The Boolean point 'MSB' is set 'TRUE' if the binary representation of 'value' has the bit set which is worth 128. Pattern = value << 2 The binary representation of 'value' is shifted left twice, and stored in 'pattern'. Each Shift Left operation has the effect of doubling the value, so two shifts quadruple the value. Logical Operators Syntax pointname = expression or IF expression or DO WHILE expression or DO UNTIL expression Revision 2.0 Page 13 CHAPTER 4 CX-Supervisor Script Language Remarks Argument Pointname Expression Description OMRON The point name to be assigned a value based on a logical expression. The Boolean value to be assigned to pointname or the Boolean value forming a conditional statement. The expression includes the following operators with points and constants: · And 'AND'.
· Or 'OR'. · Not 'NOT'. Typical Examples flag = temp AND speed The Boolean point 'flag' is assigned a value based on the logic of point 'temp' AND point 'speed'. If 'temp' and 'speed' are both not zero, 'flag' is set to 1, or 'TRUE'. A value of zero in either 'temp' or 'speed' supplies 'FALSE' or 0 to 'flag'. IF flag AND temp AND speed THEN flag = FALSE ENDIF The Boolean point 'flag' is assigned 'FALSE', on the condition that 'flag' AND point 'temp' AND point 'speed' are all not zero. If the condition fails, then 'flag' is not assigned 'FALSE'. References Refer to chapter 4, Control Statements for details of the use of the IF THEN ELSE/ELSEIF ENDIF statements. Relational Operators Syntax IF expression or DO WHILE expression or DO UNTIL expression Page 14 Revision 2.0 OMRON Remarks Argument Expression CHAPTER 4 CX-Supervisor Script Language Description The value forming a conditional statement. The expression may include the following operators with points and constants: · Greater than '>'. · Less than '<'. · Greater than or equal to '>='. · Less than or equal to '<='. · Not equal to '!='. · Equal to '=='. Typical Example IF fuel < 0 THEN fuel = 0 ENDIF The point 'fuel' is assigned the value 0 on the condition that currently, 'fuel' is less than 0. If 'fuel' is not less than 0, then it is not assigned the new value. References Refer to chapter 4, Control Statements for details of the use of the IF THEN ELSE/ELSEIF ENDIF statements. Control Statements Simple Conditional Statements Syntax IF condition THEN statementblock1 ENDIF or IF condition THEN statementblock1 ELSE statementblock2 ENDIF Revision 2.

0 Page 15 CHAPTER 4 CX-Supervisor Script Language Remarks Argument Condition Description OMRON Statementblock1 Statementblock2 The condition is made up of points and constants, using relational, logical or arithmetical notation as a test. The condition can evaluate Boolean state 'TRUE' and 'FALSE', Integer or Real numbers, or a text string. One or more statements which are performed if the condition is met. One or more statements which are performed if the condition is not met. Typical Examples IF fuel < 0 THEN fuel = 0 ENDIF Provided Integer point 'fuel' is less than 0, then it is assigned the value 0. IF burner THEN fuel = fuel - rate ENDIF Provided Boolean point 'burner' is 'TRUE', then Integer point 'fuel' is assigned a new value. It is also possible to apply IF burner == TRUE THEN as the first line, with identical results. IF distance > 630 AND distance < 660 AND lift >= -3 THEN winner = TRUE burner = FALSE ENDIF Provided that Integer point 'distance' is greater in value than 630 AND 'distance' is less in value than 660 (i.e. 'distance' is a value between 630 and 660) AND point 'lift' is greater than or equal to -3, then Boolean points 'winner' and 'burner' are assigned new values. IF burner AND fuel > 0 AND rate > 0 THEN fuel = fuel - rate ELSE lift = 0 altitude = 0 ENDIF Provided that Boolean point 'burner' is 'TRUE' AND points 'fuel' and 'rate' are greater in value than 0, then 'fuel' is assigned a new value. Otherwise points 'lift' and 'altitude' are assigned a new value. References Refer to chapter 4, Punctuation, Indentation for details on the layout of code. Page 16 Revision 2.0 OMRON CHAPTER 4 CX-Supervisor Script Language Nested Conditional Statements Syntax IF conditionA THEN statementblock1 IF conditionB THEN statementblock3 ENDIF ELSE statementblock2 ENDIF or IF conditionA THEN statementblock1 IF conditionB THEN statementblock3 ELSE statementblock4 ENDIF ELSE statementblock2 ENDIF or IF conditionA THEN statementblock1 ELSEIF conditionB THEN statementblock3 ENDIF or IF conditionA THEN statementblock1 ELSE statementblock2 IF conditionB THEN statementblock3 ELSE statementblock4 ENDIF ENDIF Revision 2.

0 Page 17 CHAPTER 4 CX-Supervisor Script Language Remarks Argument conditionA Description OMRON conditionB statementblock1 statementblock2 statementblock3 statementblock4 The condition is made up of points and constants, using relational, logical or arithmetical notation as a test.



[You're reading an excerpt. Click here to read official OMRON CX-SUPERVISOR user guide](http://yourpdfguides.com/dref/2886838)

<http://yourpdfguides.com/dref/2886838>

The condition can evaluate Boolean state 'TRUE' and 'FALSE', Integer or Real numbers, or a text string. This condition is nested in the first condition, either on a successful or unsuccessful evaluation of conditionA. The condition is made up of points and constants, using relational, logical or arithmetical notation as a test. The condition can evaluate Boolean state 'TRUE' and 'FALSE', Integer or Real numbers, or a text string.

There is no limit to the number of nested conditional statements. One or more statements which are performed if conditionA is met. One or more statements which are performed if conditionA is not met. One or more statements which are performed if conditionB is met. One or more statements which are performed if conditionB is not met. Typical Examples IF burner AND fuel > 0 AND rate > 0 THEN lift = lift + rate/5 ELSE count = 1 IF altitude > 140 THEN lift = lift - 0.2 ENDIF ENDIF Provided a successful evaluation has been made to points 'burner' AND 'fuel' AND 'rate', point 'lift' is updated with the current value of rate divided by 5 plus 'lift'. Otherwise, a further evaluation is required on point 'altitude'. If 'altitude' is currently greater than 140, then 'lift' is decremented by 0.2.

IF burner AND fuel > 0 AND rate > 0 THEN lift = lift + rate/5 ELSE IF altitude > 140 THEN lift = lift - 0.2 ENDIF ENDIF IF burner AND fuel > 0 AND rate > 0 THEN lift = lift + rate/5 ELSEIF altitude > 140 THEN lift = lift - 0.2 ENDIF Page 18 Revision 2.0 OMRON CHAPTER 4 CX-Supervisor Script Language These two examples are identical. The use of the ELSEIF statement combines the ELSE statement and the IF/ENDIF statements for brevity. It is acceptable to have more than one ELSEIF statement in an IF THEN ELSE/ELSEIF ENDIF construct. References Refer to chapter 4, Punctuation for details of the use of indentation. Case Select Syntax SELECT CASE expression CASE expression statementblock1 CASE expression statementblock2 CASE expression statementblock3 END SELECT or SELECT CASE expression CASE expression statementblock1 CASE expression statementblock2 CASE ELSE statementblock3 END SELECT Remarks Argument expression statementblock1 statementblock2 statementblock3 Description The expression may be a point, or a calculation of constants and/or points that produces a result. One or more statements that are only performed if the preceding CASE expression is met. One or more statements that are only performed if the preceding CASE expression is met.

One or more statements that are only performed if the preceding CASE expression is met. Revision 2.0 Page 19 CHAPTER 4 CX-Supervisor Script Language Typical Examples SELECT CASE colourvalue CASE 1 colour (blue) CASE 2 colour (green) CASE 3 colour (cyan) CASE ELSE colour (0) END SELECT OMRON This example shows the assignment of a colour according to the value of a point. The value of Integer point 'colourvalue' is evaluated and compared with each case until a match is found. When a match is found, the sequence of actions associated with the CASE statement is performed.

When 'colourvalue' is 1, the colour given to the current object is blue, when 'colourvalue' is 2, the colour given to the current object is green, when 'colourvalue' is 3, the colour given to the current object is cyan. If 'colourvalue' falls outside the integer range 1--3, then the colour given is 0 (black). Like ELSE and ELSEIF, the CASE ELSE statement is optional. SELECT CASE TRUE CASE temperature > 0 AND temperature <= 10 colour (blue) CASE temperature > 10 AND temperature <= 20 colour (green) CASE temperature > 20 AND temperature <= 30 colour (red) CASE ELSE colour (white) ENDSELECT In this example, instead of using a point as the condition as with the previous example, the value is the condition -- in this case Boolean state 'TRUE' -- with the integer point 'temperature' being tested at each case. If it is 'TRUE' that 'temperature' is between 0 and 10, then the current object is set to blue, or if it is 'TRUE' that 'temperature' is between 11 and 20, then the current object is set to green, or if it is 'TRUE' that 'temperature' is between 21 and 30, then the current object is set to red.

If none of these CASE statements are met, then the current object is set to white. Like ELSE and ELSEIF, the CASE ELSE statement is optional. References Refer to chapter 6, Object Commands for details of applying attributes to an object and for the use of the Colour object command. Refer to chapter 8, Colour Palette for details of the Colour Palette colour designation. Page 20 Revision 2.0 OMRON CHAPTER 4 CX-Supervisor Script Language FOR... NEXT Loop Syntax FOR pointname = startpt TO endpt STEP steppt statementblock1 NEXT Remarks Argument pointname startpt endpt steppt Description The pointname to be used as the loop counter. The initial setting of pointname, and the first value to be used through the loop.

The last value to be used. The loop ends when pointname exceeds this value. Amount to increase pointname by every pass of the loop. Steppt can be negative to count backwards providing startpt is larger than endpt. The STEP keyword and variable may be omitted in which case pointname is incremented at each pass of the loop (identical to adding STEP 1). Typical Examples FOR loopcount = 0 TO 100 Ellipse_1.vertical%fill = loopcount NEXT In this example, Ellipse_1' is gradually filled 100 times. FOR loopcount = 100 TO 0 STEP -5 Ellipse_1.vertical%fill = loopcount NEXT In this example, the fill for Ellipse_1' is gradually removed 20 times (100 times/-5). Note: Loop statements should be used with caution, as they consume processor time while they are running and some other parts of the system may not be updated.

DO WHILE/UNTIL Loop Syntax DO WHILE expression statementblock LOOP or DO statementblock LOOP WHILE expression Revision 2.0 Page 21 CHAPTER 4 CX-Supervisor Script Language or DO UNTIL expression statementblock LOOP OMRON or DO statementblock LOOP UNTIL expression Remarks Argument expression statementblock Description The expression may be a point, or a calculation of constants and/or points that produces a result. One or more statements to be executed multiple times depending on expression. Typical Example DO WHILE dooropen == TRUE Message ("You must shut the door before continuing") LOOP DO nextchar = Mid(Mystring, position, 1) position = position + 1 LOOP UNTIL nextchar = "A" Note: Loop statements should be used with caution, as they consume processor time while they are running and some other parts of the system may not be updated. Subroutines Call Syntax CALL subroutine (arguments) Page 22 Revision 2.

0 OMRON Remarks Argument subroutine arguments CHAPTER 4 CX-Supervisor Script Language Description The name of the subroutine defined at project level.



[You're reading an excerpt. Click here to read official OMRON CX-SUPERVISOR user guide](http://yourpdfguides.com/dref/2886838)
<http://yourpdfguides.com/dref/2886838>

The list of arguments required by the subroutine separated by commas. Each argument may be a pointname, constant, arithmetical or logical expression or any valid combination. Typical Example CALL MySub (\$Second, "Default", 2 + Int1) Return Syntax RETURN Typical Example IF limit > 1000 THEN RETURN ELSE value = limit ENDIF REM final part of script POLYGON_1.COLOUR = red ELLIPSE_5.

WIDTH = value The integer point 'limit' is tested for its value. If its value exceeds 1000, then the condition is met, and the RETURN command is executed. All statements after the RETURN command are ignored. If the value of integer point 'limit' does not exceed 1000, then the RETURN command is not executed, and statements after the RETURN command are performed. References Refer to the CX-Supervisor User Manual for the use of the RETURN statement for Recipe validation. Punctuation Command String Delimiters Description Alternative string delimiters allowing string to contain quote " characters. Revision 2.0 Page 23 CHAPTER 4 CX-Supervisor Script Language Syntax {Some "string" text} OMRON Typical Example Message({Error: "Invalid Function" occurred}) The { and } braces inserted around the whole strings allows the actual text in the string to contain quotes which will be displayed normally. They can be used in any situation where quotes can be used whether or not embedded quotes are required. However, for clarity the quote characters should be used by preference.

Indentation Typical Examples IF burner AND lift = lift + ELSE IF altitude > lift = lift ENDIF ENDIF fuel > 0 AND rate > 0 THEN rate/5 140 THEN 0.2 IF burner AND fuel > 0 AND rate > 0 THEN lift = lift + rate/5 ELSE IF altitude > 140 THEN lift = lift - 0.2 ENDIF ENDIF Both examples provide identical functionality, but the use of indentation, either spaces or tabs to show the construction of the statements aids readability. The use of the ELSEIF statement in this example was omitted for clarity. Multiple Commands Typical Examples count = 75 result = log(count) count = 75 : result = log(count) Both examples provide identical functionality, but the use of the colon between statements allows both to reside on the same line. Page 24 Revision 2.0 OMRON CHAPTER 4 CX-Supervisor Script Language Parenthesis Typical Examples result = 20 + 30 * 40 The result is 1220. result = (20 + 30) * 40 The values in parenthesis are calculated first. The result is 2000. References Refer to chapter 4, Logic and Arithmetic, Arithmetic Operations for further details.

Quotation Marks Typical Examples name = "Valve position" The Text point 'name' is assigned associated text, contained within quotation marks. Quotation marks must be used in this instance. Message("This text to be displayed as a message.") Passing static text as arguments to functions. BlueCarsAck = IsAlarmAcknowledged("BLUEPAINT") The point 'BlueCarsAck' is assigned a Boolean state based on the alarm 'BLUEPAINT'.

Quotation marks must be used for an alarm name. Remarks Syntax REM \ rem comment or `comment Remarks Argument Comment Type --Descriptive text. Description Typical Examples REM The following statement adds two numbers result = 45 + 754 Revision 2.0 Page 25 CHAPTER 4 CX-Supervisor Script Language result = 45 + 754 `add two numbers OMRON Indirection within Script Commands and Expressions It is possible to use text points directly or indirectly in place of literal string arguments within scripts and expressions. For instance, each of the following commands has the same effect: Using a string literal; PlayOLE("ole_1", 0) Using a textpoint directly; textpoint = "ole_1" PlayOLE(textpoint, 0) Using a textpoint indirectly via the '^' notation. text = "ole_1" textpoint = "text" PlayOLE(^textpoint, 0) It is possible to use text points indirectly in place of point name arguments within script commands.

For instance, each of the following commands has the same effect: Using a point name directly; verbnumber = 0 PlayOLE("ole_1", verbnumber) Using a textpoint indirectly via the '^' notation. verbnumber = 0 textpoint = "verbnumber" PlayOLE("ole_1", ^textpoint) An example using Indirection The value of point indirection can be seen in a situation where it is necessary to dynamically change the pointname that an object is linked to. In the following example a toggle button is configured to control the Boolean state of one of four points: The four Boolean points to be controlled are called 'motor1', 'motor2', 'motor3' and 'motor4'. The text point 'textpoint' is used to store the name of the Boolean point to be controlled. The text point 'text' is used to store the string value of the integer point 'index'. The integer point 'index' (which has a range 1-4) is used to dynamically change the point being controlled. Access to any of the four Boolean points 'motor1', 'motor2', 'motor3', 'motor4' can be achieved by applying indirection to 'textpoint' using the '^' notation and changing the contents of 'textpoint'. Page 26 Revision 2.0 OMRON CHAPTER 4 CX-Supervisor Script Language For instance, in order to dynamically change the Boolean point a toggle button is linked to follow these steps. 1, 2, 3.

.. 1. 2. Link the toggle button to a textpoint using indirection e.g. ^textpoint. Link the following script code to run as required. e.g. on clicking a button. · 3. Text = ValueToText(index) · TextPoint = "motor" + text The ValueToText function converts the integer value of the point 'index' into a string held in the textpoint 'text'. Therefore the point 'text' contains either '1', '2', '3' or '4'. The expression 'motor' + text appends the contents of the point 'text' to the literal string 'motor'.

Therefore 'textpoint' contains either 'motor1', 'motor2', 'motor3' or 'motor4' dependant on the value of 'index'. Change the value of the 'index' to determine which Boolean point to control. e.g. via the Edit Point Value (Analogue) animation.

Point Arrays within Script Commands and Expressions It is possible to access the elements of a point array directly or indirectly from within scripts or expressions. Setting the value of an array point directly; arraypoint[2] = 30 Getting the value of an array point directly; value = arraypoint[2] Setting the value of an array point using indirection; textpoint = "arraypoint" ^textpoint[2] = 30 Getting the value of an array point using indirection; textpoint = "arraypoint" value = ^textpoint[2] An example using Point Arrays The value of array points can be seen in a situation where it is necessary to dynamically change the pointname that an object is linked to.



[You're reading an excerpt. Click here to read official OMRON CX-SUPERVISOR user guide](http://yourpdfguides.com/dref/2886838)

<http://yourpdfguides.com/dref/2886838>

In the following example a toggle button is configured to control the Boolean state of one of four elements of an array point. The Boolean array point 'motor' is configured to contain 4 elements. The integer point 'index' (which has a range 0-3) is used to dynamically change the element of the point being controlled. Revision 2.0 Page 27 CHAPTER 4 CX-Supervisor Script Language OMRON In order to dynamically change the element of a Boolean point that a toggle button is linked to follow these steps. 1, 2, 3...

1. 2. Link the toggle button to an array point. e.g. 'motor[index]'. Change the value of the 'index' to determine which element of the Boolean point to control. e.g. via the Edit Point Value (Analogue) animation.

Using Aliases This facility is used to declare an alias - that is, to define a text string that can be used in place of another text string or a number within any script or expression. The Alias Definitions dialog is displayed by selecting the "Alias Definition..." option from the Project menu. It can also be displayed if "Aliases..." is selected from the script editor. The dialog displays either the User defined aliases or the preset System aliases and is toggled between these two displays by pressing the User/System Alias button.

The following illustration shows the Alias Definitions dialog displaying a number of User defined aliases. The System aliases are pre-defined and can not be edited or added to. Syntax: @AliasName Alias definition 'optional comment Page 28 Revision 2.0 OMRON Remarks: Argument @AliasName Alias definition 'comment Type string string string CHAPTER 4 CX-Supervisor Script Language Description The string name of the alias This is a string representing the actual text or expression of the expanded alias. This is an optional comment. The @ symbol at the beginning of each line initiates each alias command. For example, the text string @SomePoint could be used to represent any sequence of characters in a script or expression e.g. it could be defined as: @SomePoint = InArray[1] or even @SomePoint = Inarray[1] + Inarray[2] / 2 This is an easy way of identifying the individual members of array points. It can also be used to associate names with numbers, for example, @SecondsPerDay = 86400 Alias definitions are stored in a simple text file in the project directory, called <project name>.

pre. The format of the file consists of any number of lines such as: @Test1 = InArray[12] * 10 i.e. an @ symbol followed by the name of the alias, then an equals sign (or space), followed by the definition of the alias. Anything that follows the last apostrophe (') symbol on a line is interpreted as a comment. Any line which does not start with the @ symbol is also assumed to be a comment. Typical Examples Declare boiler temperatures @BoilerTemp1 = InArray[0] ' for boiler room 1 @BoilerTemp2 = InArray[1] ' for boiler room 2 @SecondsPerMinute = 60 ' sets duration Aliases may also be used to create a complicated expression such as @HYPOTENUSE sqrt(Opposite * Opposite + Adjacent * Adjacent) 'Calculates length of Hypotenuse This can be used in a script in the following way: Opposite = 8.45 Adjacent = 9.756 length = @HYPOTENUSE where Opposite, Adjacent and length are all REAL points. Revision 2.0 Page 29 CHAPTER 4 CX-Supervisor Script Language Note: OMRON Changing an alias definition after it has been used in an expression or script will not automatically change the result in the script. The appropriate script or expression where that alias is used must be accessed and recompiled by pressing the OK button in order to apply the changes. Page 30 Revision 2.0 OMRON CHAPTER 5 VBScript Language Reference CHAPTER 5 VBScript Language Reference This chapter is a reference for the syntax of Microsoft Visual Basic scripting language called VBScript. These features are provided by the Windows Scripting Host, included by default with Windows 2000 and Windows XP.

For a full User Guide, Language reference and details of the latest versions and support contact Microsoft at <http://msdn.microsoft.com/scripting> List of Features: Category Array handling Keyword / Feature Array Dim, Private, Public, ReDim IsArray Erase LBound, UBound Set Comments using ` or Rem Empty Nothing Null True, False Do... Loop For...Next For Each...Next If...Then.

..Else Select Case While...Wend With Abs Asc, AscB, AscW Chr, ChrB, ChrW CBool, CByte CCur, Cdate CDBl, CInt CLng, CSng, CStr DataSerial, DateValue Hex, Oct Fix, Int Sgn TimeSerial, TimeValue Date, Time DateAdd, DateDiff, DatePart DateSerial, DateValue Day, Month, MonthName Weekday, weekdayName, Year Hour, Minute, Second Assignments Comments Constants/Literals Control flow Conversions Date / Times Revision 2.0 Page 31 CHAPTER 5 VBScript Language Reference Category Keyword / Feature Now TimeSerial, TimeValue Class Const Dim, Private, Public, ReDim Function, Sub Property Get, Property Let, Property Set On Error Err Eval Execute RegExp Replace Test FormatCurrency FormatDateTime FormatNumber FormatPercent InputBox LoadPicture MsgBox Empty False Nothing Null True Atn, Cos, Sin, Tan Exp, Log, Sqr Randomize, Rnd Eval Function Execute Statement RGB Function CreateObject Err Object GetObject RegExp Addition (+), Subtraction (-) Exponentiation (^) Modulus arithmetic (Mod) Multiplication (*), Division (/) Integer Division (\) Negation (-) String concatenation (&) Equality (=), Inequality (<>) Less Than (<), LessThan or Equal(<+) Greater Than (>) Greater Than or Equal To (>=) Is And, Or, Xor Eqv, Imp OMRON Declarations Error Handling Expressions Formatting Strings Input / Output Literals Math Miscellaneous Objects Operators Page 32 Revision 2.0 OMRON Category Options Procedures CHAPTER 5 VBScript Language Reference Keyword / Feature Option Explicit Call Function, Sub Property Get, Property Let, Property Set Abs Int, Fix, Round Sgn ScriptEngine ScriptEngineBuildVersion ScriptEngineMajorVersion ScriptEngineMinorVersion Asc, AscB, AscW Chr, ChrB, ChrW Filter, InStr, InStrB InStrRev Join Len, LenB LCase, UCase Left, LeftB Mid, MidB Right, RightB Replace Space Split StrComp String StrReverse LTrim, RTrim, Trim IsArray IsDate IsEmpty IsNull IsNumeric IsObject TypeName VarType Rounding Script Engine ID Strings Variants Revision 2.0 Page 33 CHAPTER 6 Functions and Methods OMRON CHAPTER 6 Functions and Methods This chapter describes the Functions and Methods available to the scripting language. In most cases, this can be CX-Supervisor script, VBScript or JScript.

The following table describes the Functions and Methods at a glance. Function Name AcknowledgeAlarm AcknowledgeAllAlarms AcknowledgeLatestAlarm Acos Asin Atan CancelForce Chr ClearAlarmHistory ClearErrorLog ClearLogFile ClearSpoolQueue close CloseAlarmHistory CloseAlarmStatus CloseComponent CloseErrorLog CloseFile CloseLogFile CloseLogView ClosePLC colour CopyArray CopyFile cos DeleteFile disable DisableGroup DisablePoint display Function Type alarm command alarm command alarm command unary function unary function unary function point command text command alarm command event/error commands Data Logging command printer command object command alarm command alarm command comms command error command file command Data Logging command Data Logging command PLC command object command point command file command unary function file command object command point command point command object command Type Scr Scr Scr All All All Scr All All All Scr All Scr All Scr All Scr Scr Scr Scr OP All Scr All Scr OP All Scr Scr Remarks Acknowledges an alarm.



[You're reading an excerpt. Click here to read official OMRON CX-SUPERVISOR user guide](http://yourpdfguides.com/dref/2886838)
<http://yourpdfguides.com/dref/2886838>

Acknowledges all alarms. Acknowledge the latest alarm. Applies unary expression.

Applies unary expression. Applies unary expression. Removes the forcing of values on a point. Displays a character based on the ASCII character set. Clears the alarm history.

Clears the error log. Clears a data log file Discards any queued messages or alarms. Closes a specified page. Closes the current alarm history. Closes the current alarm status. Closes a component for a PLC (e.g. CX-Server components). Closes the currently open Error Log. Closes the open file.

Closes a data log file Closes the log viewer Close communications with a PLC. Specifies a colour to an object. Copies the content of an array. Copies a specified file. Applies unary expression. Deletes the specified file. Disables an object. Prevents a group of points to be read or written. Disables communications to a point. Displays a specified page.

Page 34 Revision 2.0 OMRON Function Name DisplayAlarmHistory DisplayAlarmStatus DisplayErrorLog DisplayPicture DisplayRecipes DownloadPLCProgram DownloadRecipe EditFile EnableAlarms EnableErrorLogging EnableGroup EnableOLE EnablePLC EnablePrinting ExportAndViewLog ExportLog FileExists Force ForceReset ForceSet FormatText GenerateReport GetBit GetPerformanceInfo GetPLCMode GetTextLength height horizontal%fill InputPoint IsAlarmAcknowledged IsAlarmActive Left CHAPTER 6 Functions and Methods Function Type alarm command alarm command event command general command recipe command PLC command recipe command file command alarm command error command point command comms command comms command point command printer command Data Logging command Data Logging command file command point command point command point command text command report command point command general command PLC command text command object command object command point command alarm command alarm command statement Type Scr Scr Scr Scr Scr All Scr All All Scr All Scr Scr Scr All Scr Scr All Scr Scr Scr All All All All All All OP OP Scr Scr Scr Scr Remarks Displays the current alarm history. Displays the alarm status of all current alarms. Displays the current Error Log. Reload an image for a picture object View the current recipes in the project.

Downloads specified files to the PLC. Downloads a specified recipe. Edits a specified file. Enables alarm functions. All actions become subject to Error Logging.

Permits a group of points to be read or written. Allows use of OLE functions. Allows use of PLC functions. Enables communications to a point. Permits printing of Alarms or messages. Exports data log and views Exports data log Specifies the existence of a file. Locks the value of a point. Sets a point value to 0. Sets a point value to 1. Inserts text with standard 'C' formatting characters.

Produces a report based on a report template. Retrieves a bit from a point. Retrieves internal performance and diagnostic values. Retrieves the mode of a PLC. Specifies the number of characters in a text point. Specifies the height of an object. Specifies the horizontal fill of an object. Reads a value from a point. Tests if a specified alarm has been acknowledged. Tests if a specified alarm is currently active.

Extracts characters from the left of a string Revision 2.0 Page 35 CHAPTER 6 Functions and Methods Function Name log log10 LogError LogEvent Login Logout Message Mid move MoveFile OpenComponent OpenFile OpenLogFile OpenLogView OpenPLC OutputPoint PlayOLE PlaySound PLCCommsFailed PLCMonitor PointExists PrintActivePage PrintFile PrintMessage PrintPage PrintReport PrintScreen PrintSpoolQueue Rand Read ReadMessage Right rotate OMRON Type All All Scr Scr Scr Scr Scr Scr OP Scr All Scr Scr Scr Scr Scr Scr All Scr All Scr Scr All Scr All Scr All Scr Scr All Scr OP Function Type unary function unary function error command error command security command security command text command text command object command file command comms command file command Data Logging command Data Logging command PLC command point command gen. command gen. command PLC command report command gen. command printer command gen. command file command file command text command gen.

command report command gen. command printer command gen. command file command file command text command object command Remarks Calculates the natural logarithm on a number. Calculates the base-10 logarithm on a number. Logs an error message with the error logger. Logs an event message with the error logger. Logs a user into a run-time application. Logs a user out of a run-time application. Outputs a string in a message box. Extracts a substring from a string. Moves an object. Moves the specified file. Opens a component for a PLC (e.g. CXServer components).

Opens the specified file. Opens a data log file Opens the Data Log Viewer Opens communications with a PLC. Displays the current value of a point. Plays an OLE object. Plays a sound file. Specifies if the PLC communications have failed. Monitors a PLC. Specifies the existence of a point. Prints the currently active page. Prints the specified file.

Prints messages to the configured 'Alarm/message printer'. Prints the specified page. Prints a report Prints the current display screen. Prints all queued alarms or messages. Calculates a random number.

Reads data from an open file into a point. Reads text from an external file. Extracts characters from the right of a string. Rotates an object. Page 36 Revision 2.

0 OMRON Function Name RunApplication RunHelp SelectFile SetBit SetPLCMode SetPLCPhoneNumber SetupUsers ShutDown sin sqrt StartLogging StopLogging tan TCAutoTune TCB backupMode TCGetStatusParameter TCRemoteLocal TCRequestStatus TCReset TCRspLsp TCRunStop TCSaveData TCSettingLevelI TextToValue UploadPLCProgram ValueToText vertical%fill ViewReport visible width Write CHAPTER 6 Functions and Methods Function Type gen. command gen. command file command point command PLC command PLC command security command gen. command unary function unary function Data Logging command Data Logging command unary function temp. controller command temp. controller command temp. controller command temp. controller command temp. controller command temp. controller command temp. controller command temp. controller command temp.

controller command temp. controller command temp. controller command temp. controller command text command PLC command text command object command report command object command object command file command Type Scr Scr All All All All Scr Scr All All Scr Scr All All All All All All All All All All All All Scr All Scr OP All OP OP Scr Remarks Runs the specified application. Runs the specified help file. Specifies a file name and path. Sets a specific bit from a point. Sets the mode of a PLC. Sets a phone number to a PLC. Defines users and passwords for Login.

Terminates CX-Supervisor. Applies unary expression. Applies unary expression. Starts a data set logging. Stops a data set logging.

Applies unary expression. Starts or stops a temperature controller auto-tune operation. Defines how a temperature controller stores internal variables. Retrieves the temperature controller status parameter.



[You're reading an excerpt. Click here to read official OMRON CX-SUPERVISOR user guide](http://yourpdfguides.com/dref/2886838)

<http://yourpdfguides.com/dref/2886838>

Defines the operational mode of a temperature controller.

Retrieves the temperature controller status. Resets the temperature controller. Defines the setpoint mode used by the temperature controller. Defines either auto-output mode shift or manual output mode shift. Saves data associated with the temperature controller. Performs a setting level function for the temperature controller. Converts a string to a numerical point value. Uploads programs in the PLC to specified files. Converts a numerical value into a text point. Specifies the vertical fill of an object.

Displays a report Toggles the visibility of an object. Specifies the width of an object. Writes a value to an open file. Revision 2.0 Page 37 CHAPTER 6 Functions and Methods Function Name WriteMessage OMRON Type All Function Type file command Remarks Writes text to an external file. The 'Type' column refers to the types of script and expression the function can be applied to. 'All' refers to both expressions and scripts. 'Scr' refers to scripts only. 'OP' refers to Object and Page scripts only. Object Commands Object commands control native CX-Supervisor graphical objects, like rectangles or lines. Note: Objects are native to CX-Supervisor and therefore cannot be accessed or commands issued from external script languages, like VBScript or Jscript. Current Object Syntax objectcommand Remarks Argument objectcommand · Description The expression can be made up of the following commands, which are also described in chapter 6, Object Commands: · Colour command. · Disable command. · Visible command. · Move command. · Rotate command. · Vertical fill command. · Horizontal fill command. · Height command. · Width command.

The content of the commands are made up of arithmetical or logical expressions, x and y co-ordinates, or references, varying between commands. The colour command requires a colour identifier. Typical Example colour (red) The current object is specified as red in colour. Page 38 Revision 2.0 OMRON References Refer to: CHAPTER 6 Functions and Methods Chapter 6, Blink for use of the blink command. Chapter 6, Colour for use of the colour command. Chapter 6, Disable for use of the disable command. Chapter 6, Height for use of the height command. Chapter 6, Horizontal Fill for use of the horizontal fill command. Chapter 6, Move for use of the move command.

Chapter 6, Rotate for use of the rotate command. Chapter 6, Vertical Fill for use of the vertical fill command. Chapter 6, Visible for use of the visible command. Chapter 6, Width for use of the width command. The CX-Supervisor User Manual for details of the Animation Editor. Other Objects Syntax objectname.objectcommand pagename.objectname.objectcommand Remarks Argument objectname Description This is the name of the object. The object is provided with a generic name on creation, which can be amended later to something more meaningful.

The script is automatically updated following any amendment to the object name. This can be made up of the following commands, which are described in chapter 6, Object Commands: · Blink command · Colour command. · Disable command. · Visible command. · Move command. · Rotate command. · Vertical fill command. · Horizontal fill command. · Height command. · Width command.

The content of the commands are made up of arithmetical or logical expressions, x and y co-ordinates, or references, varying between commands. The colour command requires a colour identifier. objectcommand Revision 2.0 Page 39 CHAPTER 6 Functions and Methods Typical Examples POLYGON_1.colour (red) POLYGON_1.colour = red OMRON The specified object, 'POLYGON_1' is set to be red in colour. References Refer to: CX-Supervisor User Manual for details of object names. Chapter 6, Blink for use of the blink command. Chapter 6, Colour for use of the colour command. Chapter 6, Disable for use of the disable command.

Chapter 6, Height for use of the height command. Chapter 6, Horizontal Fill for use of the horizontal fill command. Chapter 6, Move for use of the move command. Chapter 6, Rotate for use of the rotate command. Chapter 6, Vertical Fill for use of the vertical fill command. Chapter 6, Visible for use of the visible command. Chapter 6, Width for use of the width command. Blink Syntax objectname.blink (colour, status) Remarks Argument objectname colour

Description This is the name of the object. Where a script is directly attached to an object, objectname is not required. Colour to blink to. Some colour values within the colour palette have a meaningful colourID. This takes the form of the colour name, e.g., 'black' or 'yellow'. Alternatively, an integer value of 0x1000000 can be added to a number 0-65 to select a palette entry. This argument may be omitted. May be on of: TRUE turn blinking On. FALSE turn blinking Off. If omitted, TRUE is assumed.

status Page 40 Revision 2.0 OMRON Typical Examples blink (red, TRUE) CHAPTER 6 Functions and Methods Start blinking red.

LINE_1.blink(0xFFFF00, status) The object LINE_1 starts or stops blinking yellow depending on value of Boolean point 'status'. Colour Syntax objectname.colour (expression, context) colour (expression, context) or objectname.colour (colourID, context) colour (colourID, context) An equals sign may be used as an alternative to brackets: objectname.colour = expression colour = expression or objectname.colour = colourID colour = expression Either spelling 'colour' or 'color' is acceptable. Note: An equals sign may also be used for most other object commands, even if it is not directly specified in this manual.

Remarks Argument objectname expression Description This is the name of the object. Where a script is directly attached to an object, objectname is not required. The expression may be an Integer point, or a calculation of constants and/or points that produce an Integer value between 0 and 16777215. This is the desired colour's RGB value. (format is 0xBBGGRR). Some colour values within the colour palette have a meaningful colourID. This takes the form of the colour name, e.g., 'black' or 'yellow'. Alternatively, an integer value of 0x1000000 can be added to a number 0-65 to select a palette entry.

This argument is optional and may be omitted. It defines which part of the object has its colour changed. May be one or more of: @FILL change fill colour @FRAME changes frame colour If omitted both are changed. Equivalent to @FILL | @FRAME colourID context Revision 2.0 Page 41 CHAPTER 6

Functions and Methods Typical Examples TEXT_3.

colour (blue) OMRON or TEXT_3.colour = blue The object 'TEXT_3' is set to blue. BALL.colour (35 + 0x1000000) The object 'BALL' is set to colour 35 from the colour palette.



[You're reading an excerpt. Click here to read official OMRON CX-SUPERVISOR user guide](http://yourpdfguides.com/dref/2886838)
<http://yourpdfguides.com/dref/2886838>

BALL.

colour (0xFF0000, @FILL) The object 'BALL' is set to blue. shade = tint1 + tint2 IF shade > 65 OR shade < 0 THEN shade = 0 ENDIF ELLIPSE_1.colour (shade + 0x1000000) The point 'shade' is set to a value based on 'tint1' and 'tint2', and is tested first to ensure that it is a value between 0 and 65. If 'shade' falls outside this range, then it cannot be applied as a colour to an object, and is therefore reset to 0 (or black). ELLIPSE_1' is set to the palette colour of the value of shade. References Refer to chapter 6, Colour Palette for details of colour names and colour numbers. Disable Syntax objectname.disable (expression) Remarks Argument objectname expression Description This is the name of the selectable object. Where a script is directly attached to an object, objectname is not required. The expression can be made up of points resulting in 'TRUE' or 'FALSE'.

Typical Examples disable (TRUE) The current pushbutton object to which this example applies is disabled. PUSH_8.disable (count AND flag) Page 42 Revision 2.0 OMRON CHAPTER 6 Functions and Methods The selectable object 'PUSH_8' is disabled provided Integer point 'count' AND Boolean point 'flag' return 'TRUE'. Height Syntax objectname.height (expression, context) objectname.height = expression Remarks Argument objectname expression context Description This is the name of the object, following any amendment to the object name. Where a script is directly attached to an object, objectname is not required. This is a value, point or an arithmetic expression returning a new height value in pixels. This argument is optional and may be omitted. It defines which part of the object is the datum, and remains static. May be one of: @TOP uses object top as datum. @CENTRE uses object centre as datum @BOTTOM uses object bottom as datum If omitted @CENTRE is assumed Typical Examples height (100) or height = 100 The height of the current object is set to 100. LINE_1.height (stretch/offset, @top) The height of object 'LINE_1' is changed to the value calculated by points 'stretch' and 'offset', keeping the top where it is.

Horizontal Fill Syntax objectname.horizontal%fill (expression, context) Revision 2.0 Page 43 CHAPTER 6 Functions and Methods Remarks Argument objectname expression context Description OMRON This is the name of the object. Where a script is directly attached to an object, objectname is not required. This is an arithmetic expression that must return a value between 0 and 100.

On return of a valid result, the fill commences from left to right. This argument is optional and may be omitted. It defines which side of the object is filled from. May be one of: @LEFT fill from the left @RIGHT fill from the right If omitted, @LEFT is assumed Typical Examples horizontal%fill (50) The current object to which this example applies is filled by 50%. ELLIPSE_1.horizontal%fill (GAS_LEVEL, @RIGHT) The object 'ELLIPSE_1' is filled from the right, provided the point 'GAS_LEVEL' returns a valid result, between 0 and 100. Move Syntax objectname.move (x co-ordinate, y co-ordinate) Remarks Argument objectname x co-ordinate y co-ordinate Description This is the name of the object, following any amendment to the object name. Where a script is directly attached to an object, objectname is not required.

The x and y co-ordinates of the origin of the object at its resultant position in pixels are specified in the form (x, y). Points alone or as part of an arithmetic expression may be used as a basis for this expression. Typical Examples move (100, 200) The current object to which this example applies is moved to the specified position. POLYGON_1.move (xpos, ypos/5) The object 'POLYGON_1' is moved to the position specified by points 'xpos' and 'ypos' divided by 5.

Page 44 Revision 2.0 OMRON CHAPTER 6 Functions and Methods Rotate Syntax objectname.rotate (angle, context, fixed, xcoord, ycoord) Remarks Argument objectname angle context Description This is the name of the object. Where a script is directly attached to an object, objectname is not required. The angle of rotation can range between 0 to 360 in a clockwise direction.

Points alone, or as part of an arithmetic expression may be used as an angle. This argument is not required and may be omitted. May be one of: @TOPLEFT rotate around top left of object @TOPCENTRE rotate around top centre of object @TOPRIGHT rotate around top right of object @CENTRELEFT rotate around centre left of object @CENTRE rotate around centre of object @CENTRERIGHT rotate around centre right of object @BOTTOMLEFT rotate around bottom left of object @BOTTEMCENTRE rotate around bottom centre of object @ BOTTOMRIGHT rotate around bottom right of object @USERDEFINED user defined point specified in xcoord and ycoord. This argument may be omitted. If this boolean value is true, the rotation origin is fixed to the screen, even if the object is moved.

*Otherwise, the rotation origin is relative to object position. Only required if @USERDEFINED is specified. These integer variables specify the rotation origin in pixels fixed xcoord ycoord Typical Examples rotate (45) The current object to which this example applies is rotated by 45°. RECTANGLE_1.rotate(tilt, @USERDEFINED, 0, -100, 10) The object 'RECTANGLE_1' is rotated by the value of 'tilt', about a point 100, 10 relative to the objects current position. rotate (a * sin(b)) The current object is rotated based on the result of an arithmetic expression involving points named 'a' and 'b'. Revision 2.0 Page 45 CHAPTER 6 Functions and Methods OMRON Vertical Fill Syntax objectname.vertical%fill (expression, context) Remarks Argument objectname expression context Description This is the name of the object. Where a script is directly attached to an object, objectname is not required. This is an arithmetic expression that must return a value between 0 and 100. On return of a valid result, the fill commences from bottom to top. This argument may be omitted. May be one of: @DOWN Fill object downwards @UP Fill object upwards If omitted, @UP is assumed Typical Examples vertical%fill (50) The current object to which this example applies is filled by 50%. ELLIPSE_1.*

vertical%fill (OIL_QUANTITY, @DOWN) The object 'ELLIPSE_1' is filled provided the point 'OIL QUANTITY' returns a valid result, between 0 and 100. Visible Syntax objectname.visible (expression) Remarks Argument objectname expression Description This is the name of the object. Where a script is directly attached to an object, objectname is not required. The expression can be made up of points resulting in 'TRUE' or 'FALSE'. Typical Examples visible (TRUE) The current object to which this example applies becomes visible.



[You're reading an excerpt. Click here to read official OMRON CX-SUPERVISOR user guide](http://yourpdfguides.com/dref/2886838)

<http://yourpdfguides.com/dref/2886838>