



# CREW

MANUAL



# CREW Manual



#### **ESA elettronica S.p.A.**

Via Padre Masciadri 4/a  
22066 Mariano Comense (CO) -Italia  
Tel. +39 031 757400  
Fax. +39 031 751777

#### **ESA elettronica S.p.A.**

Unità locale di Bentivoglio  
Via Monari Sardè 3  
40010 Bentivoglio (BO) Italy  
Tel. +39-051-6640464  
Fax +39-051-6640784

#### **ESA Europa S.L.U.**

Passeig del Ferrocarril, 335  
08860 Castelldefels (Barcelona) - España  
Tel. +34 936455014  
Fax. +34 936455013

意萨电子科技（上海）有限公司  
中国上海市宜山路889号齐来工业城4号楼6层D1  
**ESA Electronic Technology (Shanghai) Co. Ltd**  
Unit D1, 6F, Bldg. 4#, No. 889 Yishan Road  
Shanghai 200233 - P.R.China  
Tel. +86 21 6090 7250  
Fax +86 21 6090 7258

#### **ESA Technology Inc.**

780 NW York Drive Suite 202  
Bend, OR 97703 U.S.A.  
Tel. +1 707 5447300  
Fax. +1 541 7492208

#### **ESA energy S.r.l.**

Via Fortunato Zeni 8  
38068 Rovereto (TN) - Italia  
Tel. +39 0464 443272  
Fax. +39 0464 443273

#### **ESA elettronica S.p.A.**

Unità locale di Pontedera  
Via Molise,1 - Z.I. Gello  
56025 Pontedera (PI) - ITALY  
Tel. +39 0587 296014  
Fax. +39 0587 294240

#### **ESA Elettronica GmbH**

Carl-Zeiss-Str. 35  
D-63322 Rödermark  
Tel : +49 6074 486 45 22  
Fax: +49 6074 486 45 66

#### **ESA Software & Automation India Pvt. Ltd**

Ist Floor, 2nd Main, HRBR Layout,  
3 rd Block, Kalyan Nagar Post,  
Bangalore 560 043 - India  
Tel. +91 80 25435656

#### **ESAElektronik Technology Ticaret Limited Şirketi**

Şerifali Mah., Çetin Cad. Kible Sk.  
No: 6 Of Plaza Kat: 5 D.: 7  
Ümraniye/İstanbul - Türkiye  
Tel. +90 216 466 70 33  
Fax. +90 216 466 70 99



**MOTORS**

# CREW Manual

---

## Contents

<b>What is Crew</b>	<b>14</b>
<b>Main features of Crew</b>	<b>15</b>
<b>Installation of Crew</b>	<b>16</b>
Product registration .....	25
Minimum requirements.....	29
Recommended requirements.....	30
<b>Start page</b>	<b>31</b>
Menu layout .....	32
Main bar of the menu .....	33
START Menu.....	38
Create new project .....	39
Open an existing project .....	46
Save project .....	47
Save project as.....	47
Close project.....	48
Project properties .....	49
Upload panel image .....	50
Options .....	56
General.....	57
Update .....	61
Canvas.....	63
About.....	65
PROJECT Menu .....	66
Panel .....	67
System alarms .....	75
Languages .....	90
Fonts.....	103
Dictionary.....	104
Groups.....	114
User group - Administrators - .....	115
User group - Users - .....	116
Users.....	128
Notifications.....	135
Adding a serial protocol .....	167
Adding an ethernet protocol .....	169
Convert device.....	172
Shared devices.....	175
Project validation .....	184
Project simulation .....	186
Online simulation.....	189
Project download.....	192
Project download - Ethernet TCP IP - .....	197
Download project TCP/IP - UDP - .....	215

# CREW Manual

Project download - USB.....	220
TOOLS Menu .....	225
Create archive and send .....	226
Backup (offline) .....	230
Online Tools .....	235
GRAPHICS Menu .....	245
Align to the left .....	250
Align in the centre.....	252
Align to the right .....	254
Align with the top .....	256
Align in the middle.....	258
Align with the bottom .....	260
Distribute horizontally .....	262
Submenu Preview .....	266
Rotation Submenu .....	269
Rotate 90° left.....	270
Rotate 90° right.....	272
Move to 0° .....	274
Size Submenu.....	276
Same size .....	277
Same length .....	279
Same height .....	281
Grouping Submenu.....	283
Grouping .....	284
Separate .....	287
Order Submenu.....	289
Place in foreground .....	290
Up .....	292
Down .....	295
Move back.....	298
Layer Submenu .....	300
Move to first .....	301
Up one level .....	306
Down one level.....	311
Move to last.....	316
Selection tool.....	321
Rectangle .....	323
Rectangle Properties .....	325
Ellipse .....	328
Ellipse Properties.....	329
Circular sector .....	332
Circular sector Properties.....	333
Arch .....	336
Arch Properties .....	337
Polygon.....	340
Polygon Properties .....	341
Regular polygon .....	344
Regular Polygon Properties.....	345
Line.....	348
Line Properties .....	349
Polygonal chain.....	352

# CREW Manual

Polygonal chain Properties .....	353
Pipeline .....	356
Pipeline Properties.....	357
Text - Label - .....	360
Text - Label - Properties.....	361
Text list .....	364
Text list Properties.....	366
Image.....	369
Image Properties .....	371
Image list.....	374
Image List Properties.....	379
Edit Box - Numerical field - .....	382
Edit Box Properties.....	383
Sensitive area .....	386
Sensitive Area Properties .....	387
Sensitive area - Ellipse - .....	390
Sensitive area - Ellipse - Properties .....	391
Switch.....	394
Switch Properties.....	395
Button .....	398
Button Properties.....	399
Text Button .....	402
Text Button Properties.....	403
Image Button.....	406
Image Button Properties .....	407
Selector.....	410
Selector Properties .....	411
Bar .....	414
Bar Properties.....	415
Indicator .....	418
Indicator Properties.....	419
Data Log.....	422
Data Log Properties .....	423
Trend .....	433
Trend Properties.....	434
Trend XY .....	442
TrendXY Properties .....	443
Touch Trend.....	446
Active Alarms .....	453
Active Alarms Properties.....	454
Alarm Log .....	472
Alarm Log Properties.....	473
Recipe .....	491
Recipe Properties.....	492
Recipe List .....	508
Recipe List Properties.....	509
Users.....	528
User Properties .....	529
Details on Bézier Curves.....	532
Constructing Bézier curves.....	533
FDA 21 CFR Part 11 .....	539

# CREW Manual

<b>Editor Properties</b>	<b>541</b>
Properties.....	546
Dynamic assignment of values to the properties.....	549
Thresholds Management Feature .....	550
Thresholds Management Operation .....	550
Event Editor .....	556
Page Events .....	561
Pop-up page Events.....	565
Variable Events .....	565
Alarm Events.....	566
Datalog Events.....	567
Recipe Events.....	568
Timer Events .....	569
Layers.....	569
Operations on Layers.....	571
Add one or more Layers.....	572
Delete the selected Layer.....	573
Move the selected Layer up one level.....	574
Move the selected Layer down one level.....	575
Merge Layers .....	577
Edit a Layer .....	581
Select a Layer .....	582
Show-Hide Layers .....	582
Block-Unblock Layers.....	588
Libraries.....	590
Buttons .....	591
Led lights.....	599
Indicators .....	600
Bars .....	601
Selectors.....	602
Animations.....	603
Images.....	604
Images (Industrial).....	605
Images (Logos).....	606
Images (Flags).....	607
Templates .....	609
<b>Explore Project</b>	<b>612</b>
Pages .....	612
Page management.....	620
Page properties .....	623
Popup .....	626
Pop-up page management .....	630
Pop-up page properties.....	635
Sequences.....	638
Tags.....	647
Tags Grid.....	651
Tags Editor .....	652
Internal Tag .....	655

# CREW Manual

Device Tag.....	661
Shared Device Tag.....	668
System Tag.....	682
Network Tag .....	690
Panel network.....	691
Validation .....	701
Transformations.....	712
Limits .....	717
Events - Tags - .....	722
Export - Import Tags.....	731
Import Tags.....	732
Alarms .....	738
Alarms Grid.....	744
Alarms Editor .....	766
Export - Import Alarms .....	780
Export Alarms.....	784
Properties - Alarms - .....	787
Events - Alarms - .....	790
Alarm Signals .....	791
Datalogs.....	798
Recipes.....	812
SyncMode - Synchronised transfer .....	820
Status Area.....	820
Command Area .....	822
Scripts.....	831
Creating a new script:.....	832
Key of Variable Types and premises on syntax .....	838
Notes .....	839
ESAHMI .....	841
Sub-Objects .....	841
Shared properties .....	841
Methods - ESAHMI .....	842
Properties.....	851
ESAALARMMGR .....	853
Methods - ESAALARMMGR .....	853
Methods - ESACOM( [index] ) .....	857
ESACTRL.....	861
ESACTRL ( page, controlname ) .....	861
Methods - ESACTRL.....	862
ESAETH .....	869
ESAETH.TCPCLIENT .....	870
ESAETH.TCPSERVER.....	874
ESAETH.UDP.....	881
ESAFILE.....	886
Stream Methods - ESAFILE .....	891
Properties - ESAFILE .....	900
ESAPAGEMGR.....	901
Methods - ESAPAGEMGR.....	902
ESAPRN.....	910
Methods - ESAPRN .....	911
Properties - ESAPRN.....	913

# CREW Manual

ESARECIPEMGR .....	914
Methods - ESARECIPEMGR .....	915
Properties - ESARECIPEMGR .....	925
ESASAMPLEMGR .....	926
Methods - ESASAMPLEMGR .....	927
ESATAG .....	931
Methods - ESATAG .....	932
ESATIMER .....	939
Methods - ESATIMER .....	940
ESUSERMGR .....	942
Methods - ESUSERMGR .....	943
<b>Predefined functions</b>	<b>954</b>
Alarm Functions .....	955
Hardware functions .....	956
Message Functions .....	956
Page Functions .....	957
Project Functions .....	958
Recipe Functions .....	959
Datalog - Sample Functions .....	960
System Functions .....	961
Variables - Tag Functions .....	962
Timers Functions .....	963
Users Functions .....	964
<b>Tutorial</b>	<b>965</b>
Example Crew+CODESYS project .....	966
Adding a project page .....	967
Creating a variable or Tag .....	968
Creating a Switch Button and associating it to a tag .....	970
CODESYS settings .....	977
Opening the Standard Project .....	979
Installing ESA-LIBRARIES (XML-file) .....	980
Symbol Configuration .....	981
Creating Tag + PRG program .....	982
Build up SYMBOL .....	983
Add ESA I/O (Plug Device) .....	984
Associate the program tags to the I/O signals .....	985
Create the application boot file .....	986
Data Log Configuration - Crew side .....	994
Data Log Configuration - EW Terminal Side .....	1008
Recipe Configuration - Crew Side .....	1014
Recipe Configuration - EW Terminal Side .....	1028
Examples of how Script is used .....	1043
Example 1 - Analysis of variables and launching events .....	1043
Example 2 - Page access based on user-level .....	1046
Example 3 - Exporting Alarms to user-selected file .....	1048
Example 4 - Storing Recipe on memory support .....	1049
Example 5 - Deleting all of the recipes on EW .....	1052



# CREW Manual

Example 6: Print recipe list in paper .....	1054
Everyware .....	1056
What is Everyware.....	1056
Main characteristics of Everyware .....	1057
What can be done with Everyware.....	1057
Types of licences.....	1058
Creating domains.....	1059
Log in.....	1062
Local Connection .....	1063
Proxy Settings.....	1065
Adding a panel.....	1066
Adding a user .....	1069
Features of Everyware .....	1073
Remote desktop .....	1074
Remote desktop (Local connection).....	1077
Chat service.....	1082
VPN (Virtual Port Network).....	1087
Passthrough .....	1091
Explore files.....	1092
Settings on panel side .....	1094
Everyware Configuration.....	1095
“Network” settings.....	1100
Communication Set Up - Examples .....	1103
Allen Bradley -Rockwell ControlLogix Ethernet.....	1103
Parameters Set up - Allen Bradley -Rockwell.....	1113
Exporting Tags from RSLogix.....	1117
Importing RSLogix Tags .....	1120
CODESYS SoftPLC - For generic PLCs.....	1128
Emerson - Control Techniques (CT Modbus protocol Unidrive M) .....	1140
Set up Parameters - Emerson Control Techniques .....	1150
Modbus RTU Master Ethernet.....	1154
Set up Parameters - Modbus RTU Master Ethernet .....	1164
Siemens S7-1200 1500 Ethernet .....	1172
Parameters Set up - Siemens S7-1200 1500 Ethernet.....	1182
Exporting Tags from TIA PORTAL .....	1189
Importing TIA PORTAL Tags.....	1196
Connection cables.....	1211
Connection of the cable shield .....	1211
Cables for PLC Allen Bradley - Rockwell (RS232) .....	1214
Cable for PLC Omron (RS232) .....	1219
Cables for PLC Mitsubishi (RS232).....	1221
Cable for PLC Panasonic (RS232).....	1223
RS485 connection cables.....	1224
Cable for Emerson - Control Techniques.....	1224
SP, SK, Uni M200 – 700 (RS485).....	1224
Cable for PLC SIEMENS S7 (RS485).....	1225
Backup and Restore - Example .....	1226

# CREW Manual

List of serial Drivers :.....	1245
List of Ethernet Drivers :.....	1246
List of CAN Open Drivers :.....	1247
List of ProfiBUS Drivers :.....	1248
<b>Serial Drivers</b>	<b>1249</b>
Mitsubishi - FX2-3 series.....	1250
Mitsubishi - FX5 MC binary type protocol.....	1252
Modbus Master RTU.....	1255
Omron FINS protocol.....	1257
Omron Host Link series.....	1259
Siemens - PPI protocol Slave mode.....	1262
<b>Ethernet Drivers</b>	<b>1263</b>
Allen Bradley - Ethernet-IP for ControlLogix.....	1264
Control Techniques - CT Modbus -.....	1267
ESA - Codesys SoftPLC -.....	1269
Fanuc Robotics - Focas2 CNC protocol Tcp IP.....	1270
Fanuc Robotics SNP-X protocol - Tcp IP -.....	1275
GeFanuc - SRTP protocol -.....	1277
Mitsubishi MC protocol FX2 - 3 Tcp.....	1279
Mitsubishi - FX5 SLMP binary type protocol.....	1281
Modbus Master RTU Tcp.....	1283
Modbus RTU Slave Tcp IP.....	1285
Omron FINS protocol Tcp.....	1287
SAIA - Ether S-BUS.....	1289
Schneider TelemecaniqueTSX series with ETZ module (Tcp)..	1291
Siemens Industrial Ethernet for LOGO.....	1293
Siemens Industrial Ethernet.....	1296
Wago - 750-8xx -.....	1300
<b>Drivers CAN Open</b>	<b>1302</b>
ESA - CanOpen Master -.....	1303
ESA - CanOpen Slave -.....	1308
<b>Drivers ProfiBUS</b>	<b>1311</b>
ESA - ProfiBUS standard -.....	1312
SIEMENS - ProfiBUS -.....	1314
<b>Esaware: the products in the line</b>	<b>1319</b>
HMI.....	1321
<b>EW104xxxxx</b>	<b>1325</b>
Dimensions - Drilling.....	1326
Rear.....	1327
Front.....	1330

# CREW Manual

CAN line termination .....	1331
Changing the battery .....	1333
EW Installation .....	1334
Power supply .....	1335
Electromagnetic compatibility.....	1335
Communication ports .....	1336
<b>EW107xxxxx</b>	<b>1337</b>
Dimensions - Drilling .....	1338
Rear .....	1339
Front .....	1341
CAN line termination .....	1342
Changing the battery .....	1344
EW Installation .....	1345
Power supply .....	1346
Electromagnetic compatibility.....	1346
Communication ports .....	1347
<b>EW112xxxxx</b>	<b>1348</b>
Dimensions - Drilling .....	1349
Rear .....	1350
Front .....	1352
CAN line termination .....	1353
Changing the battery .....	1355
EW Installation .....	1356
Power supply .....	1357
Electromagnetic compatibility.....	1357
Communication ports .....	1358
<b>EW115xxxxx</b>	<b>1359</b>
Dimensions - Drilling .....	1360
Rear .....	1361
Front .....	1363
CAN line termination .....	1364
Changing the battery .....	1367
EW Installation .....	1368
Power supply .....	1369
Electromagnetic compatibility.....	1369
Communication ports .....	1370
<b>EW6xxxxxx</b>	<b>1371</b>
Description.....	1372
Performance.....	1373
Features.....	1374
Technical specifications .....	1376
<b>EW600Axxxxx</b>	<b>1380</b>

# CREW Manual

Dimensions .....	1381
EW600 assembly on EW terminal.....	1382
EW600 removal from EW terminal.....	1383
Electrical connections.....	1384
<b>EW600Bxxxxx</b>	<b>1388</b>
Dimensions .....	1389
EW600 assembly on EW terminal.....	1390
EW600 removal from EW terminal.....	1391
Electrical connections.....	1392
<b>EW600Cxxxxx</b>	<b>1395</b>
Dimensions .....	1396
EW600 assembly on EW terminal.....	1397
EW600 removal from EW terminal.....	1398
Electrical connections.....	1399
<b>EW600Dxxxxx</b>	<b>1405</b>
Dimensions .....	1406
EW600 assembly on EW terminal.....	1407
EW600 removal from EW terminal.....	1408
Electrical connections.....	1409
<b>EW600Exxxxx</b>	<b>1412</b>
Dimensions .....	1413
EW600 assembly on EW terminal.....	1414
EW600 removal from EW terminal.....	1415
Electrical connections.....	1416
<b>EW600NOOCxx</b>	<b>1419</b>
Dimensions .....	1420
EW600 assembly on EW terminal.....	1421
EW600 removal from EW terminal.....	1422
Electrical connections.....	1423
<b>EW600NOOExx</b>	<b>1426</b>
Dimensions .....	1427
EW600 assembly on EW terminal.....	1428
EW600 removal from EW terminal.....	1429
Electrical connections.....	1430
<b>Service Page</b>	<b>1433</b>
<b>Control Panel</b>	<b>1435</b>
HMI Info:.....	1436

# CREW Manual

Everyware: .....	1437
Brightness: .....	1445
Date / Time:.....	1446
SNTP: .....	1447
Stylus: .....	1449
Owner: .....	1454
Password: .....	1455
Network: .....	1456
IP Config: .....	1460
Backup: .....	1462
Reset: .....	1463
FTP: .....	1465
Reboot:.....	1466
CODESYS3:.....	1467

# CREW Manual

## What is Crew

The Crew platform has been designed to satisfy the four main requirements of the user: efficiency, ease of learning, simple storing of the commands and satisfaction of the end user.

Crew's powerful editor uses an extreme operating logic flexibility to manage high amounts of data at a high speed.

Its sensitivity in responding to the user's gestures allows Crew Runtime to quickly adapt and obey commands. Fast response and maximum fluidity in web surfing are perfect for multi-touch applications and guarantee optimal operation with touch screen, both capacitive and resistive.

Now, creating, modifying and controlling objects is very simple. The new icons and templates offer a ready-interactivity that allows the user to create easy-to-use applications quickly, with excellent graphic design. And that's not all.

Crew offers a vast library of objects divided by type of use, all customisable by the user in a simple and intuitive manner. The graphic design is completely vectorial with the possibility of having transparencies and gradients as well.

Lastly, Crew's versatility guarantees the operator considerable savings in time.

# CREW Manual

## Main features of Crew

Advanced user management.

Powerful tools for the storage of any type of data.

Crew is also an App: now it is easy to connect apps from any mobile device, such as a smartphone or a tablet with Android, Apple IOS or Windows Phone operating systems.

The details of the project can be viewed and edited with common commands such as pinch, scroll and swipe, which allow us to scroll the page, zoom the screen quickly and interact with the project commands.

The Esaware applications have been designed to control the plant from your mobile device with a single hand, to make the use of any mobile device, tablet or smartphone simpler and safer.

The dual method of READ-ONLY/READ-WRITE interaction offers users a secure method of interaction in supervision with read only or a complete interaction during the data editing phase.

# CREW Manual

## Installation of Crew

This section provides the information necessary to proceed with the first step of accessing Crew: INSTALLING it.

We describe the requirements that a machine needs so that the app functions correctly and the crucial steps of the installation phase will be described.



# CREW Manual

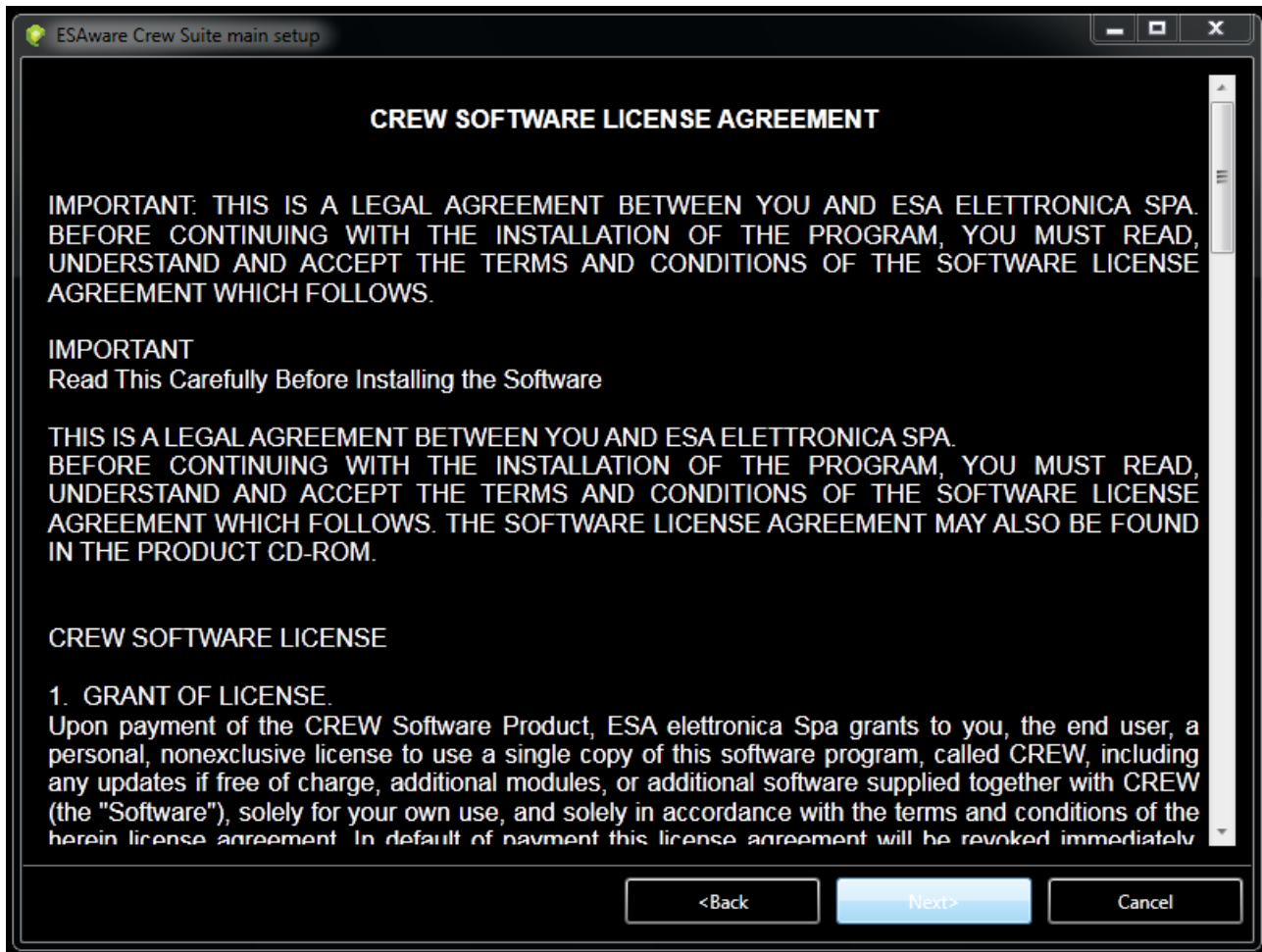
## Installation

Run the executable file contained on the DVD, click "Next" on the window that appears.



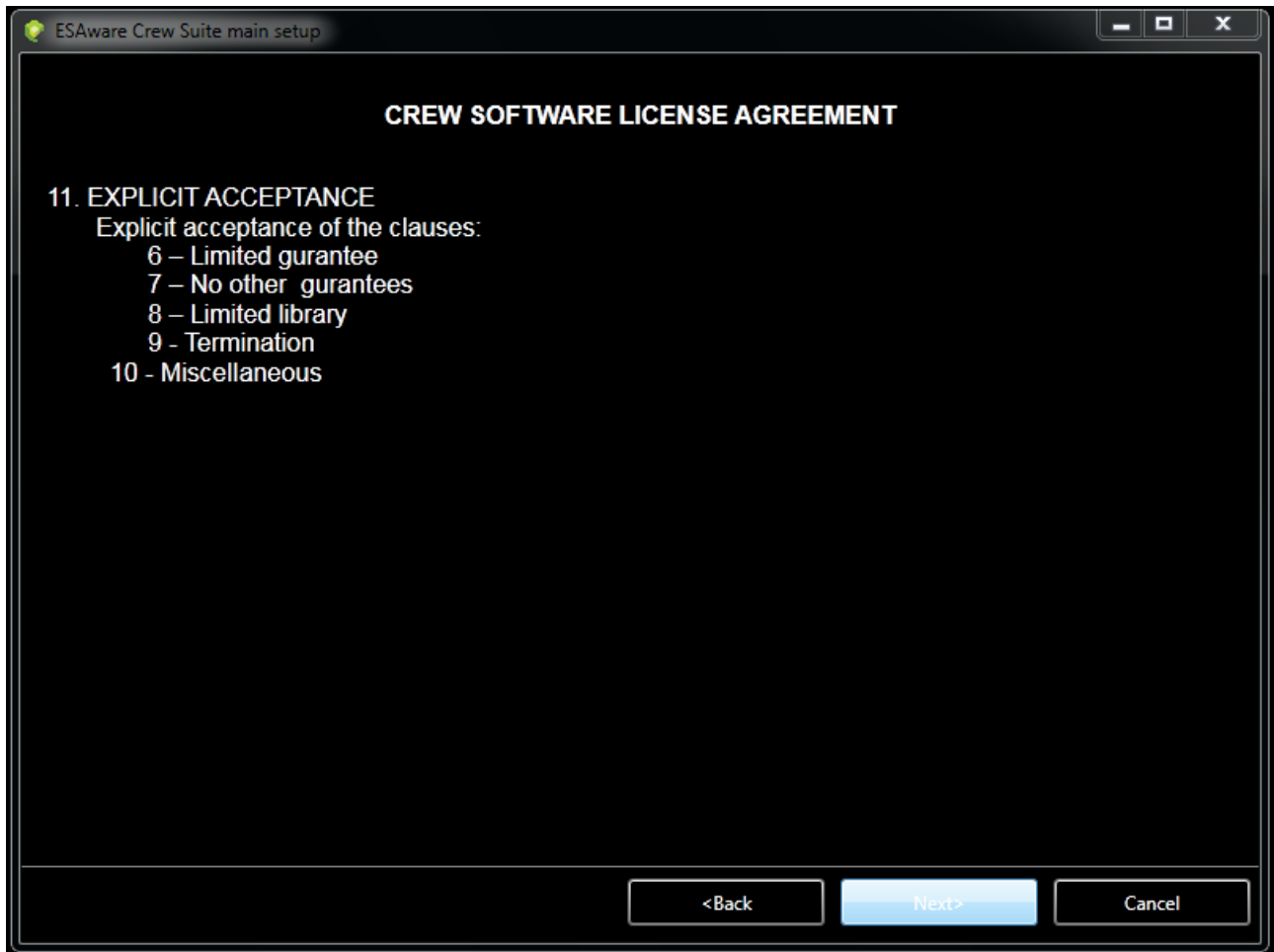
# CREW Manual

Read the licence contract, then click "Next".



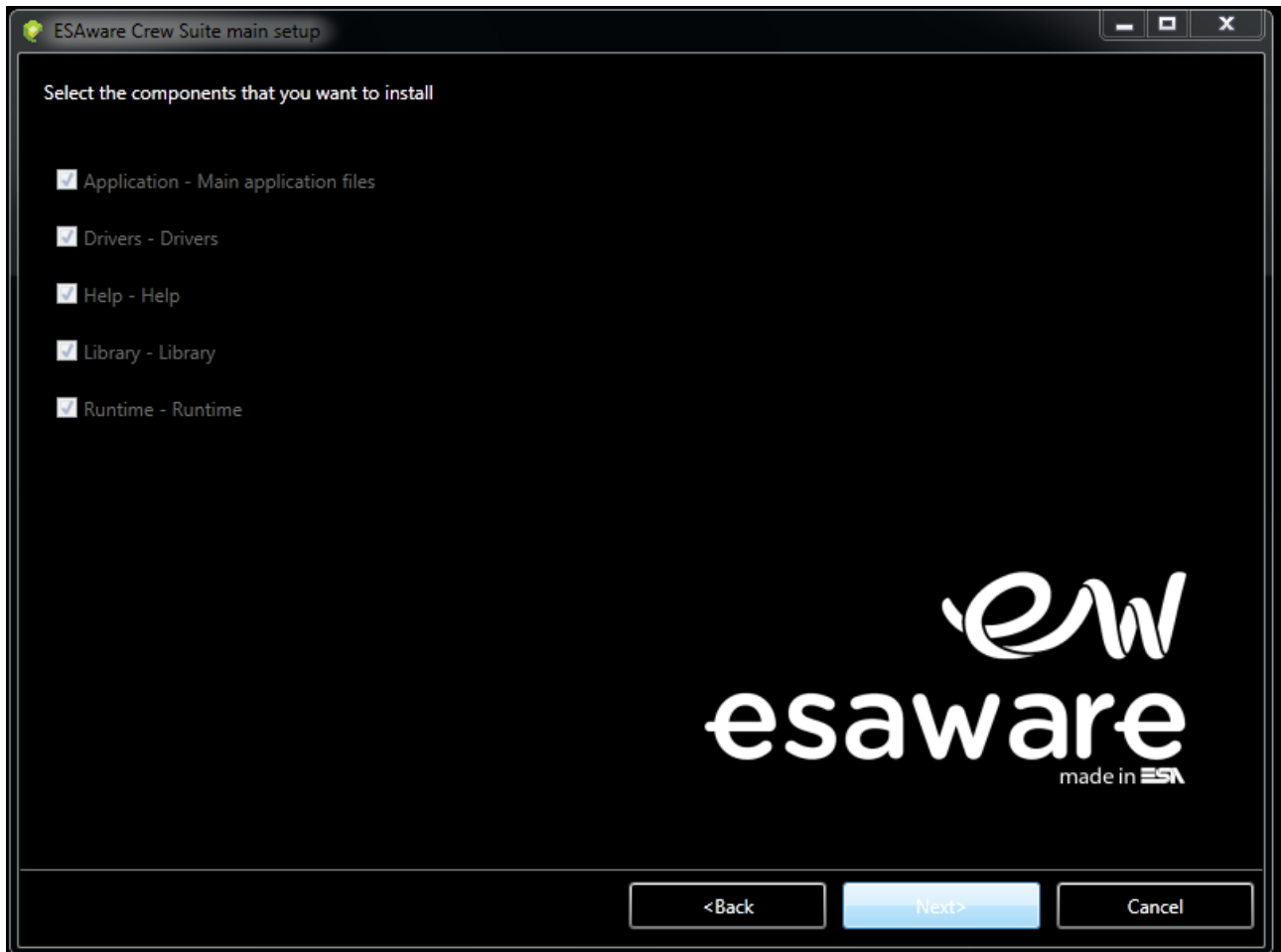
# CREW Manual

Click "Next".



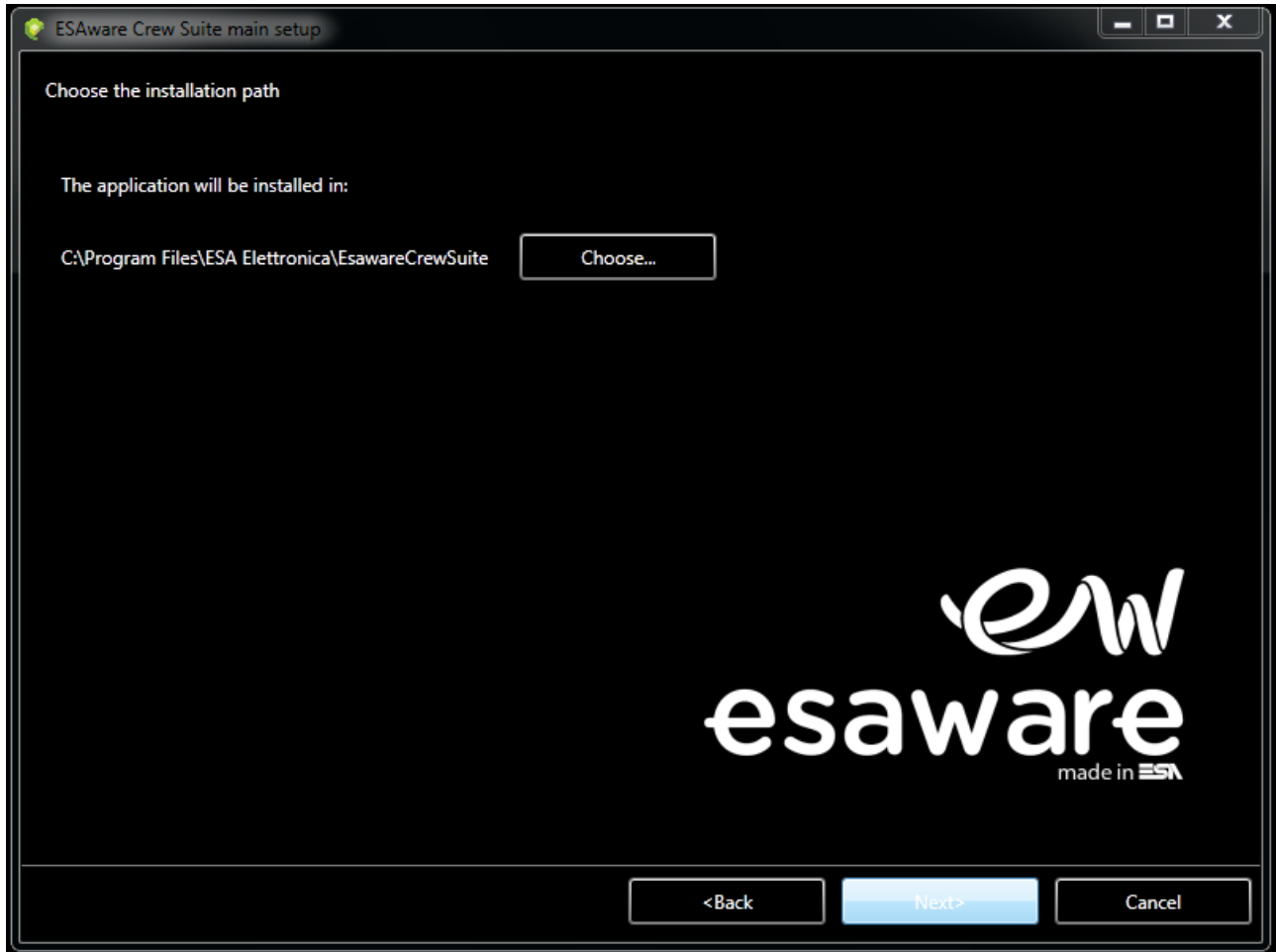
# CREW Manual

Click "Next".



# CREW Manual

Click "Next" or select the target path where to install the software.



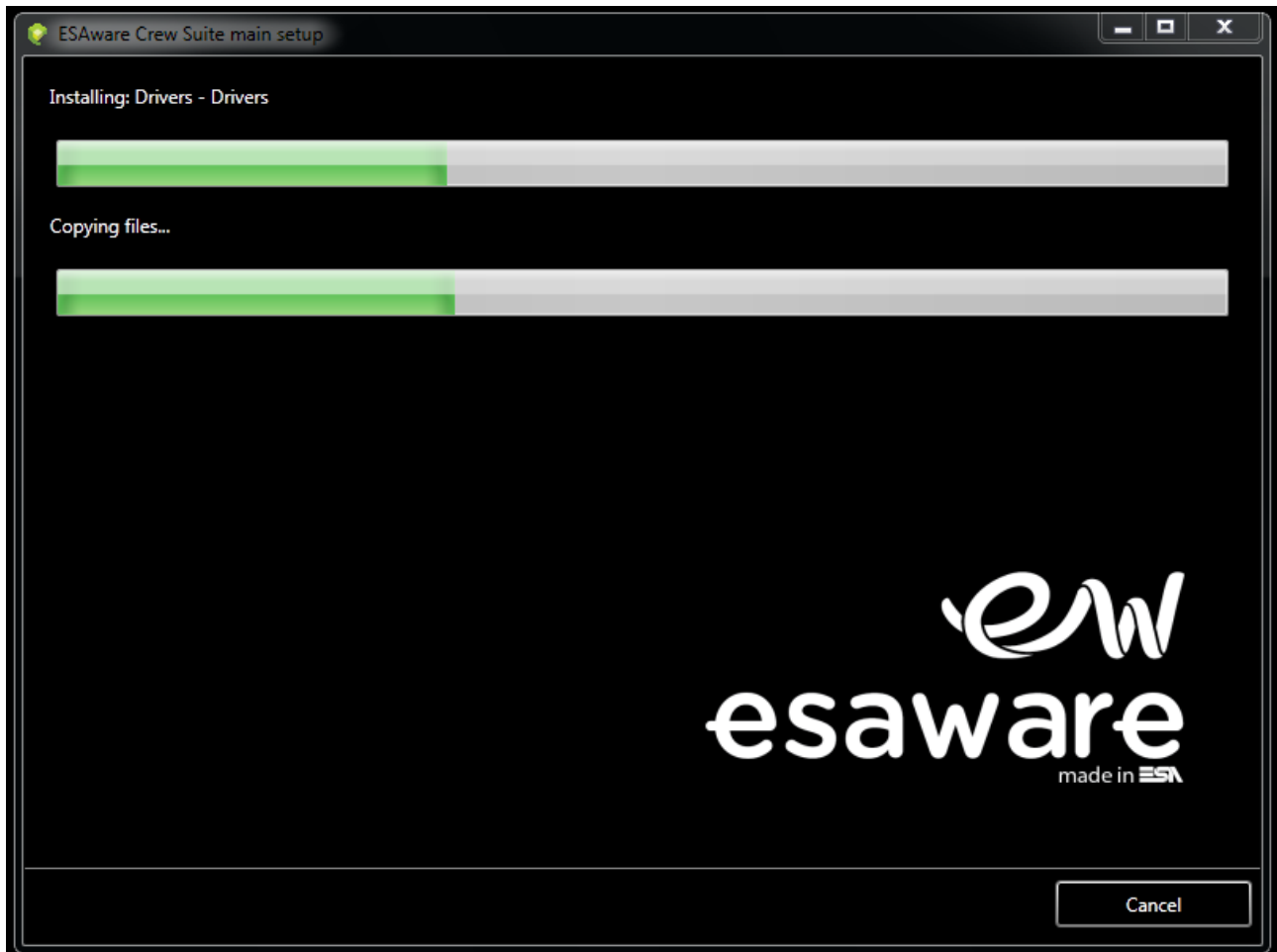
# CREW Manual

Click "Start installation".



# CREW Manual

Now the app installation and copy files bars will appear.



# CREW Manual

At the end click "Close".

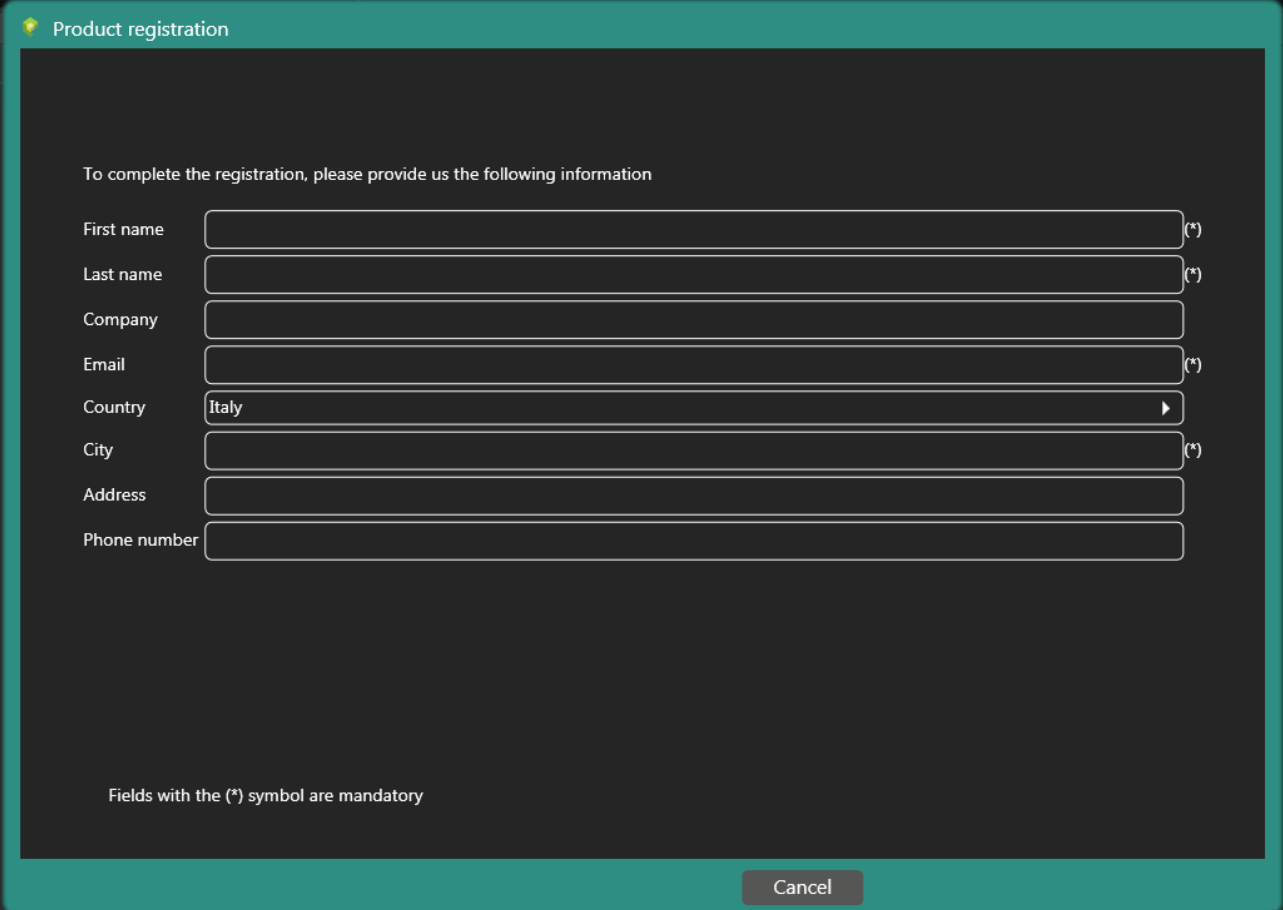




# CREW Manual

## Product registration

Launch Crew to make the following image appear automatically.



The screenshot shows a 'Product registration' dialog box with a teal header and a dark grey background. The form contains the following fields:

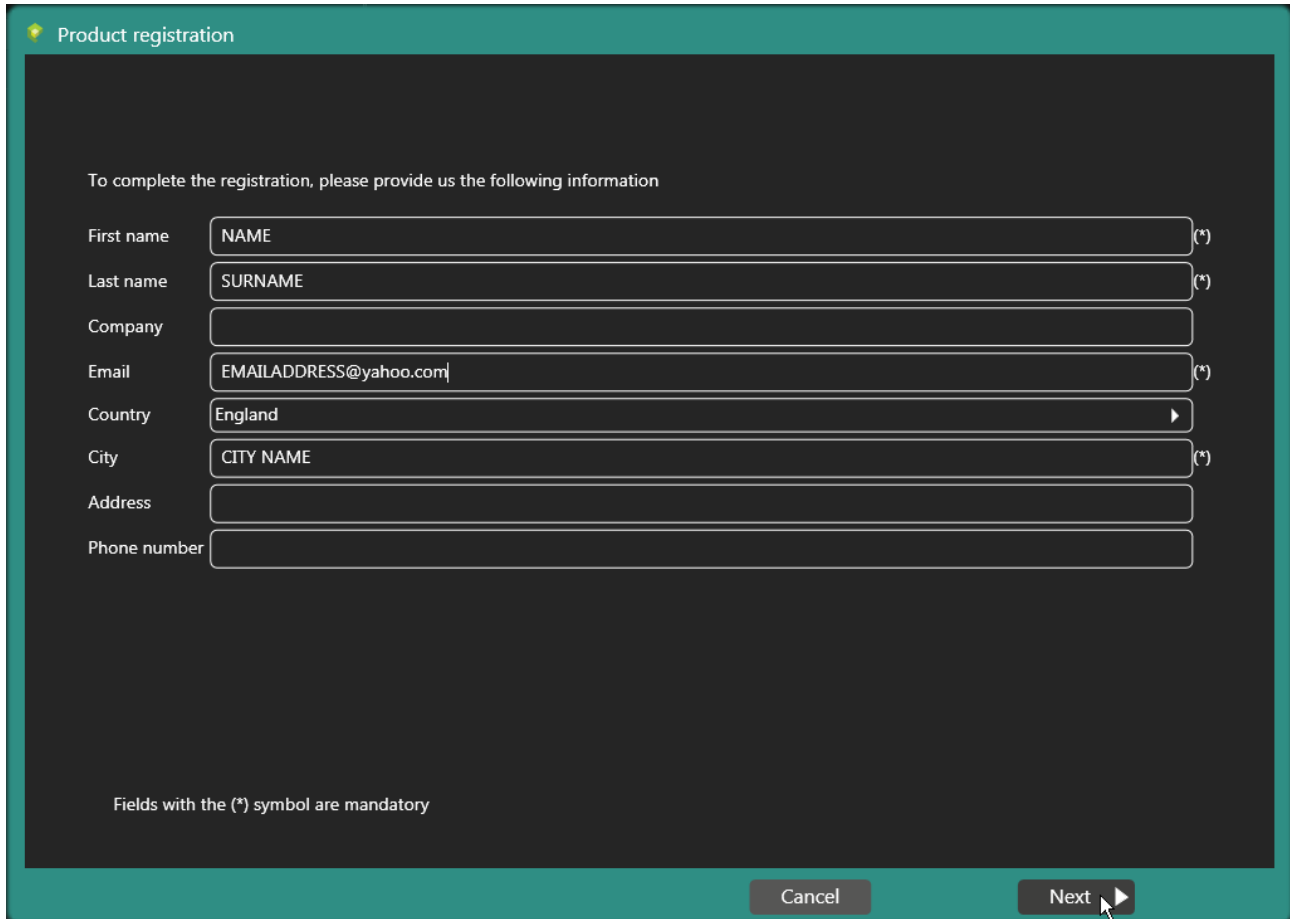
- First name (\*)
- Last name (\*)
- Company
- Email (\*)
- Country (Dropdown menu with 'Italy' selected)
- City (\*)
- Address
- Phone number

Fields with the (\*) symbol are mandatory.

Cancel

# CREW Manual

Fill in the mask with all of the required data and then click “Next”.



Product registration

To complete the registration, please provide us the following information

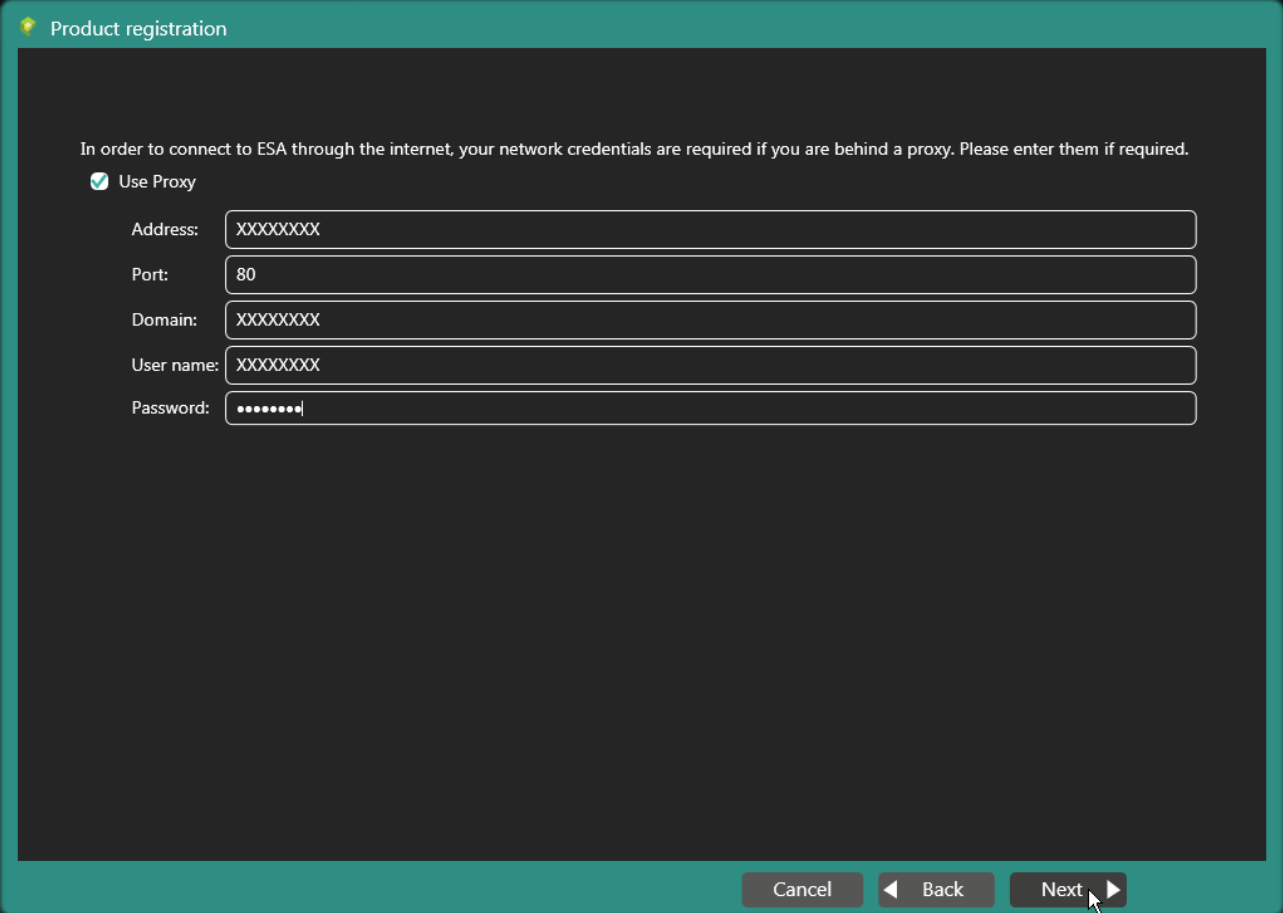
First name	<input type="text" value="NAME"/>	(*)
Last name	<input type="text" value="SURNAME"/>	(*)
Company	<input type="text"/>	
Email	<input type="text" value="EMAILADDRESS@yahoo.com"/>	(*)
Country	<input type="text" value="England"/>	
City	<input type="text" value="CITY NAME"/>	(*)
Address	<input type="text"/>	
Phone number	<input type="text"/>	

Fields with the (\*) symbol are mandatory

Cancel Next

# CREW Manual

Enable the “check box” if you are using a Proxy and enter the networks credentials. Then click “Next”.



Product registration

In order to connect to ESA through the internet, your network credentials are required if you are behind a proxy. Please enter them if required.

Use Proxy

Address:

Port:

Domain:

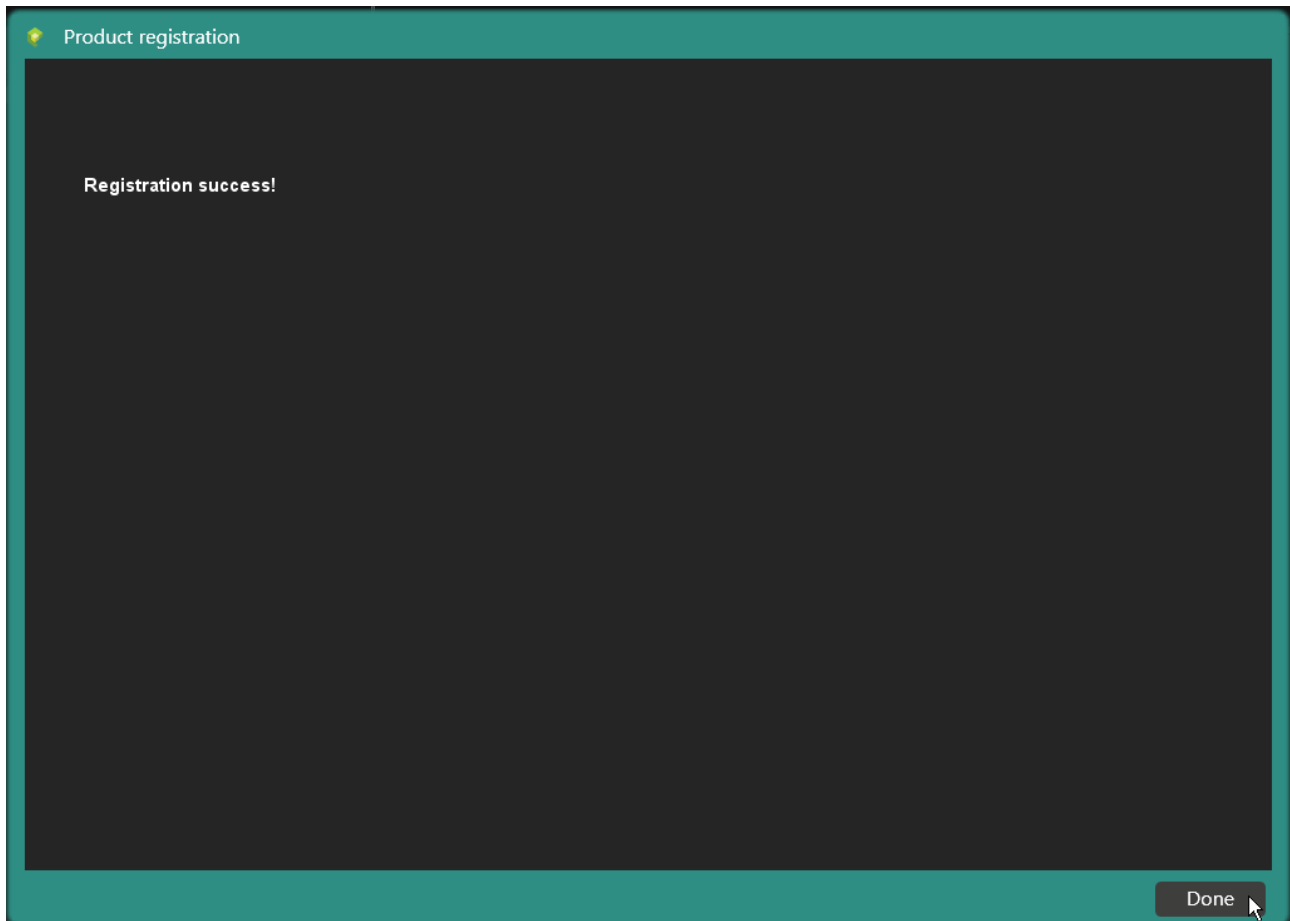
User name:

Password:

Cancel ◀ Back Next ▶

# CREW Manual

At the end click “Done”.



# CREW Manual

---

## Minimum requirements

The minimum requirements necessary to use Crew on your machine are as follows.

Type	Requirement
Operating System	Windows® XP with Service Pack 3
RAM Memory	1 GB RAM
Processor	Pentium IV or equivalent
Screen Resolution	1280*768
Hard Disk Space	3 GB

# CREW Manual

---

## Recommended requirements

The recommended requirements necessary to use Crew on your machine are as follows.

Type	Requirement
Operating System	Windows <sup>®</sup> 7 Windows <sup>®</sup> 8
RAM Memory	2 GB RAM or higher
Processor	Pentium IV or higher
Screen Resolution	1280*768
Hard Disk Space	3 GB or higher

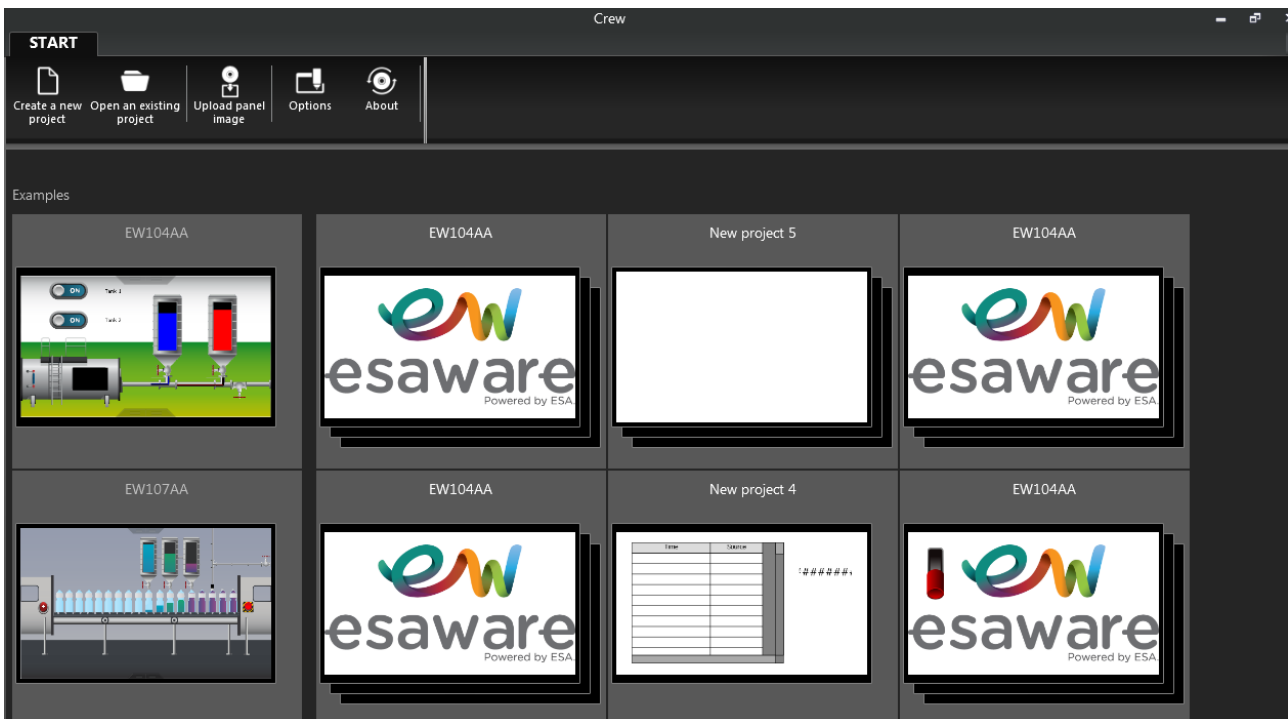
# CREW Manual

## Start page

When you start Crew the home page appears illustrating recently opened projects and where it is possible to choose from the following menus:

- 1 Create a new project
- 2 Open an existing project
- 3 Update the “CE” image on the panel
- 4 Options
- 5 About (information on the installed software)

The image is the following:

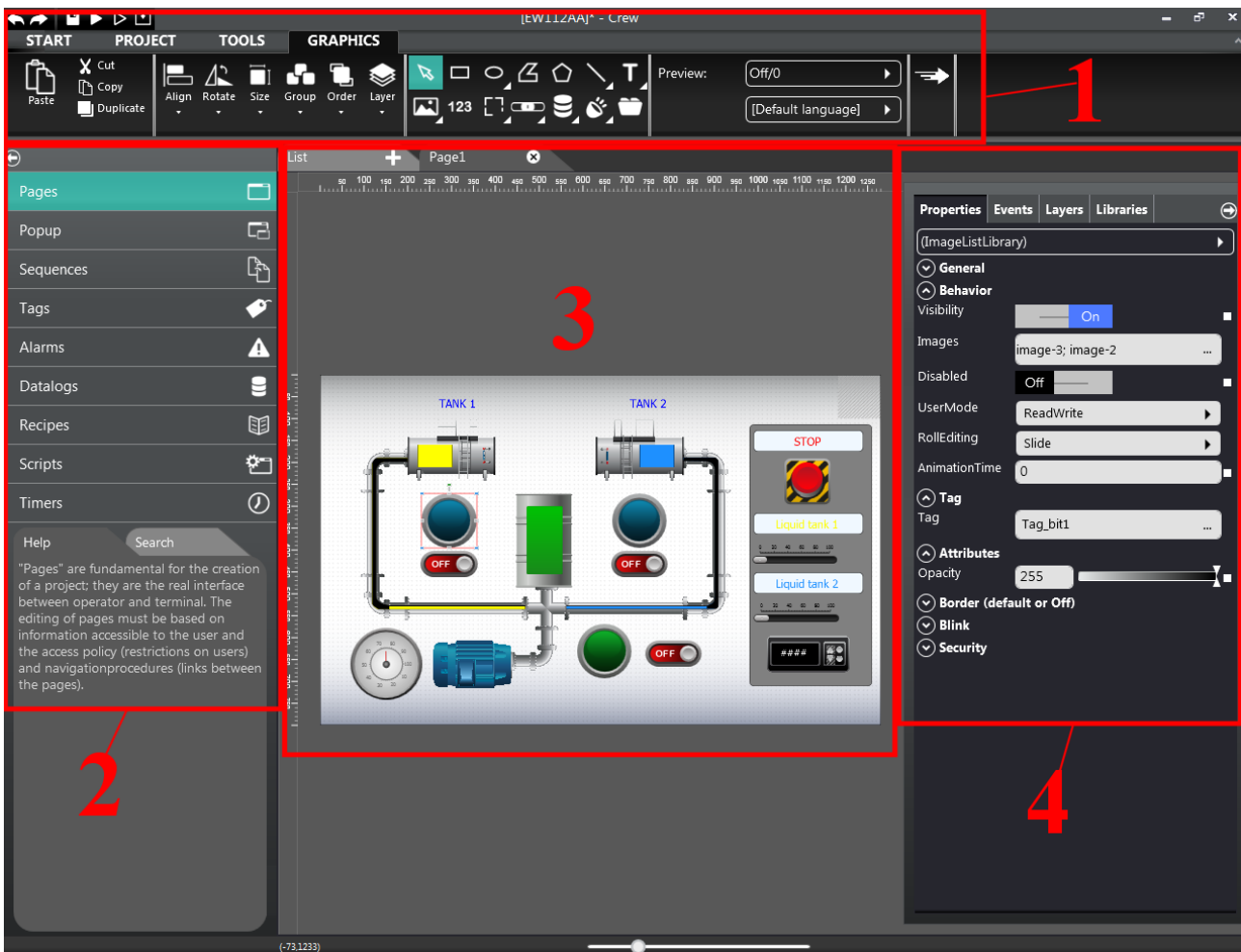


# CREW Manual

## Menu layout

The layout of the application is made up of the following areas:

- Main bar of the menus (1)
- Explore project (2)
- Work area (3)
- Event Editor / Property Editor / Layer / Libraries (4)





# CREW Manual

---

## Main bar of the menu



The main bar is composed of the following menus:

- START Menu
- PROJECT Menu
- TOOLS Menu
- GRAPHICS Menu (this appears when at least one page has been entered in the project)

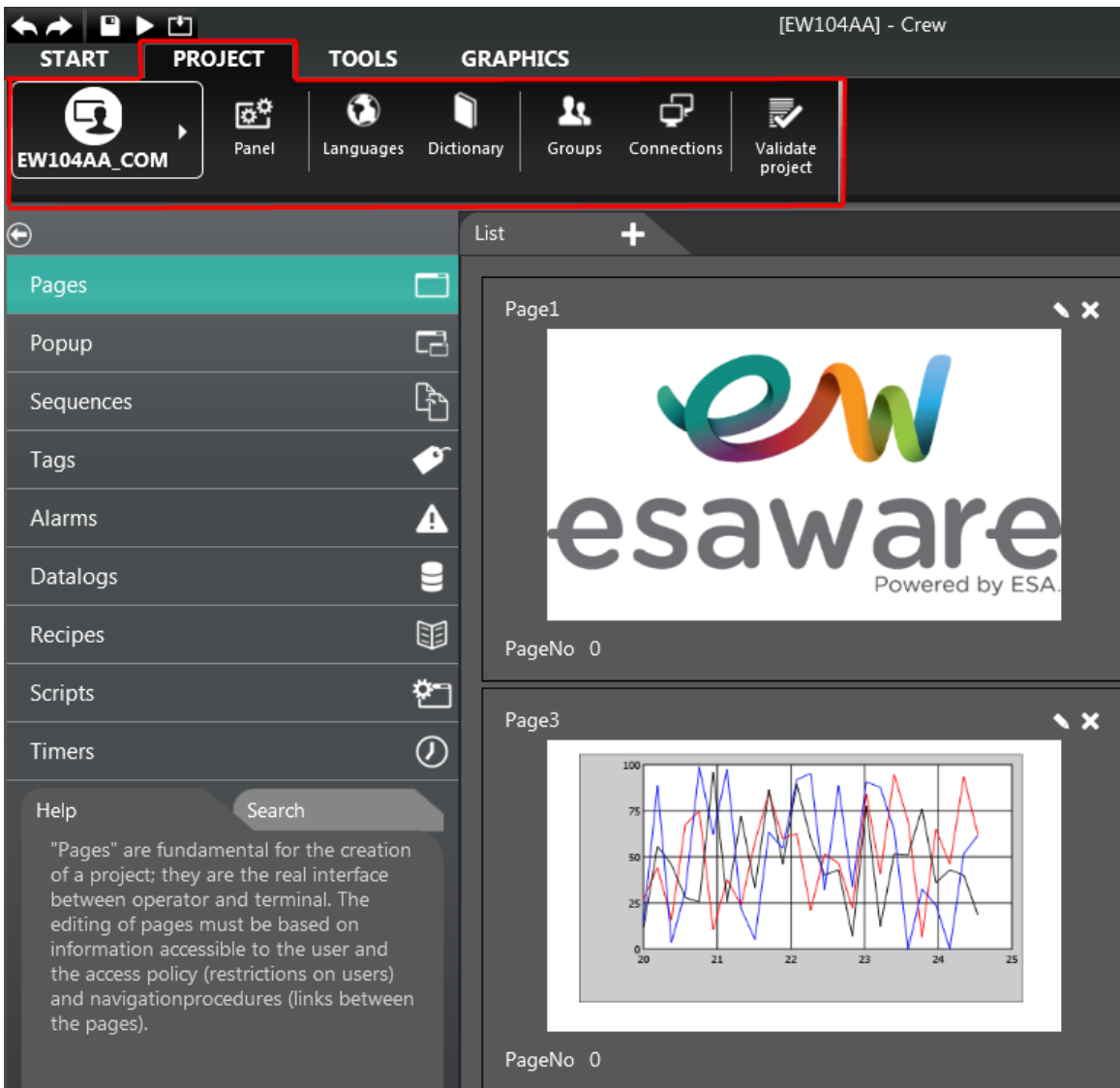
# CREW Manual

## START Menu



# CREW Manual

## PROJECT Menu



The screenshot displays the CREW software interface. At the top, the title bar shows "[EW104AA] - Crew". Below it, a menu bar contains "START", "PROJECT", "TOOLS", and "GRAPHICS". The "PROJECT" menu is highlighted with a red box and contains the following items: "EW104AA\_COM" (with a user icon), "Panel" (with a gear icon), "Languages" (with a globe icon), "Dictionary" (with a book icon), "Groups" (with a group of people icon), "Connections" (with a network icon), and "Validate project" (with a checkmark icon).

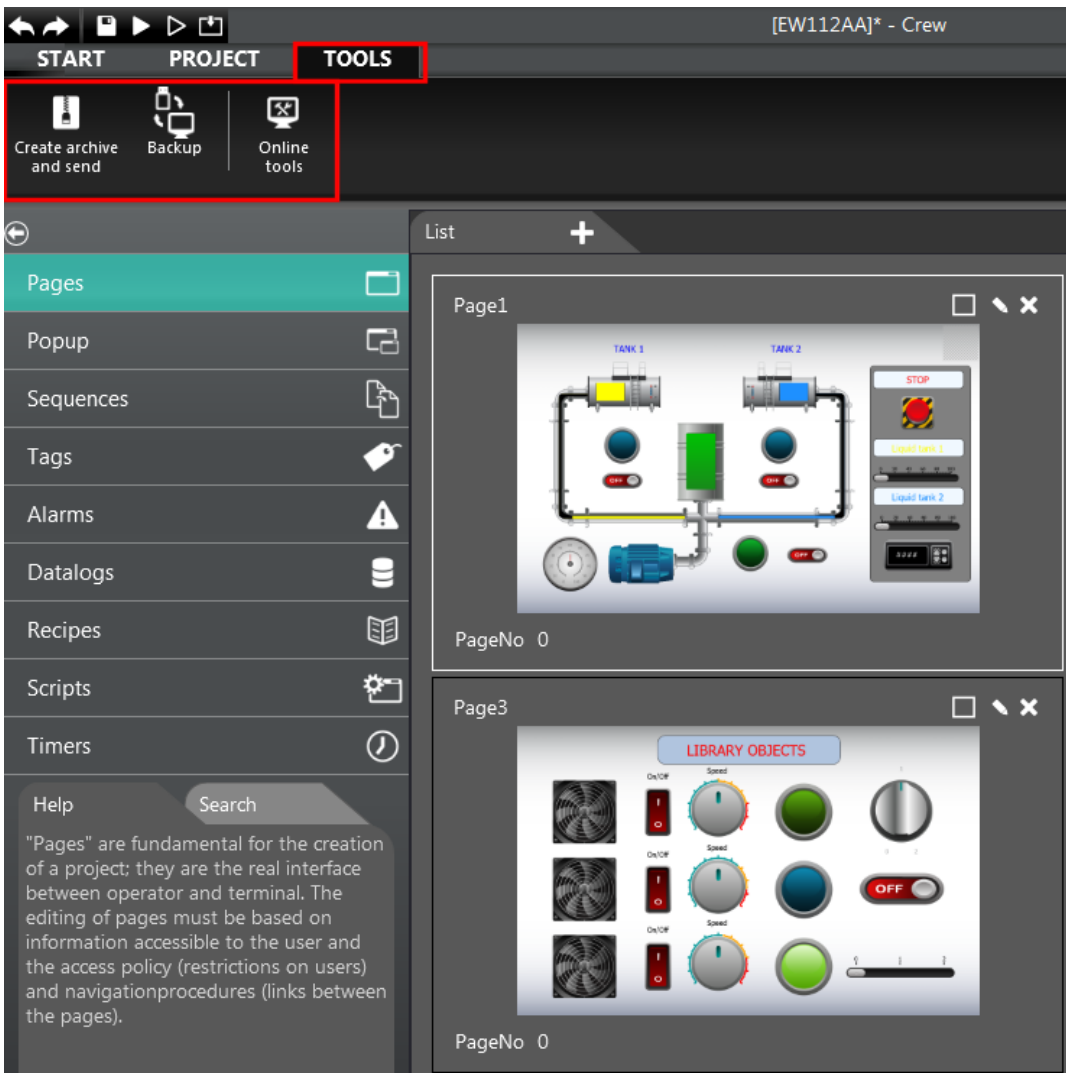
Below the menu bar, a "List" panel is visible, showing a list of project pages. The "Pages" item is selected and highlighted in green. The list includes: Pages, Popup, Sequences, Tags, Alarms, Datalogs, Recipes, Scripts, and Timers. A "Help" section is also visible, with a search bar and a text block explaining the role of pages:

Help Search  
"Pages" are fundamental for the creation of a project; they are the real interface between operator and terminal. The editing of pages must be based on information accessible to the user and the access policy (restrictions on users) and navigation procedures (links between the pages).

The main workspace displays two pages. "Page1" shows the "esaware" logo with the text "Powered by ESA." below it. "Page3" shows a line graph with a y-axis ranging from 0 to 100 and an x-axis ranging from 20 to 25. The graph displays multiple data series in various colors (blue, red, black, green) showing fluctuating values over time.

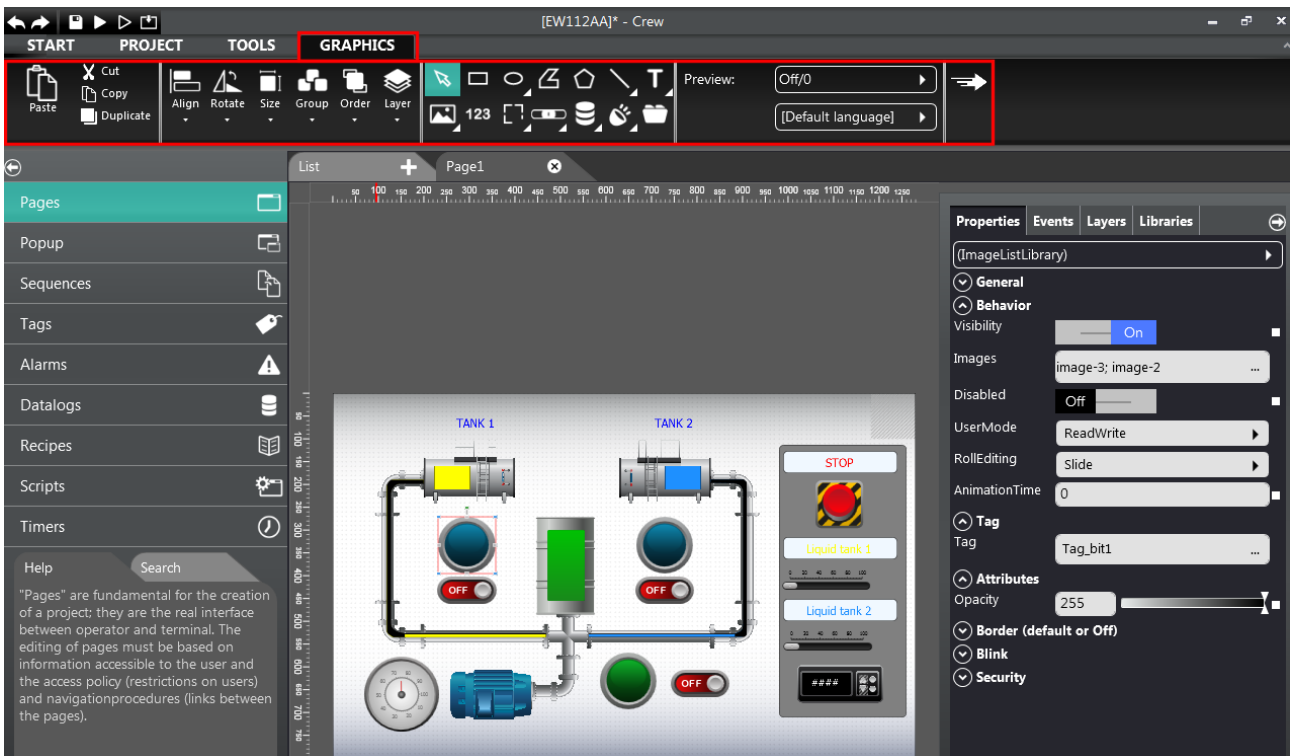
# CREW Manual

## TOOLS Menu



# CREW Manual

## GRAPHICS Menu



# CREW Manual

## START Menu

The START menu contains the following sub-menus.



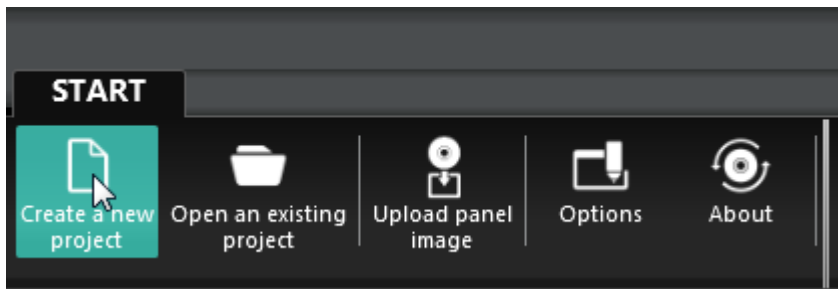
- Create a new project: to create a project and choose all of the necessary settings.
- Open an existing project: to open an existing project to edit it.
- Save project: to save the project currently in execution.
- Save project as: to save the project and choose the target path.
- Close project: to close the currently open project.

# CREW Manual

- Project properties: to view the project details, such as Name, Description, Author, Version and Folder it belongs to.
- Upload panel image: to update the “CE” image on the panel.
- Options: to view the options such as Language and default Folders (save, import/export, images etc.).
- About: to obtain general information on Crew, such as software, driver and Runtime version.

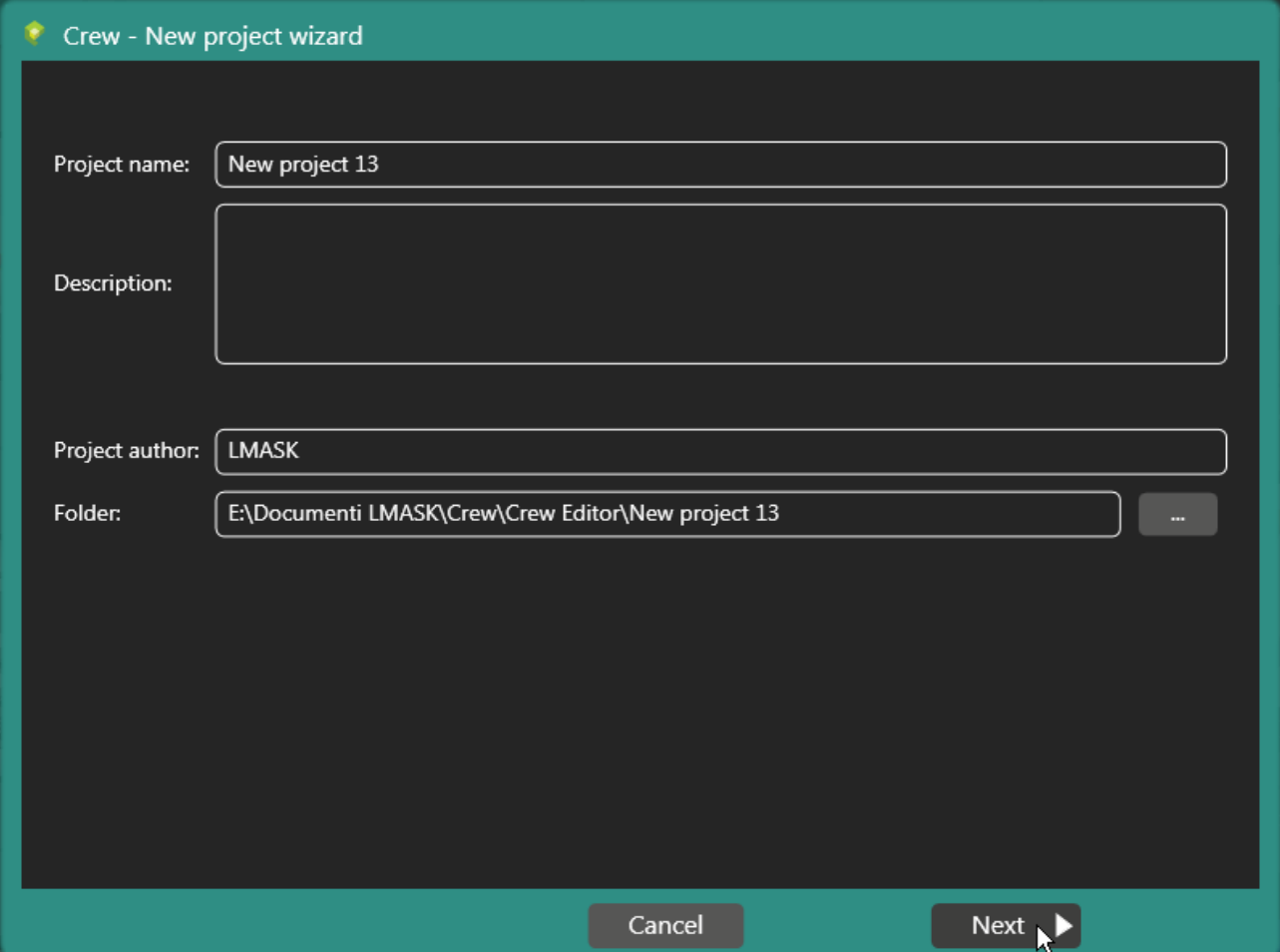
## Create new project

Click the “Create a new project” icon.



# CREW Manual

In the window that appears, enter the project data (name, any description, author, target folder) and click “Next”.



Crew - New project wizard

Project name: New project 13

Description:

Project author: LMASK

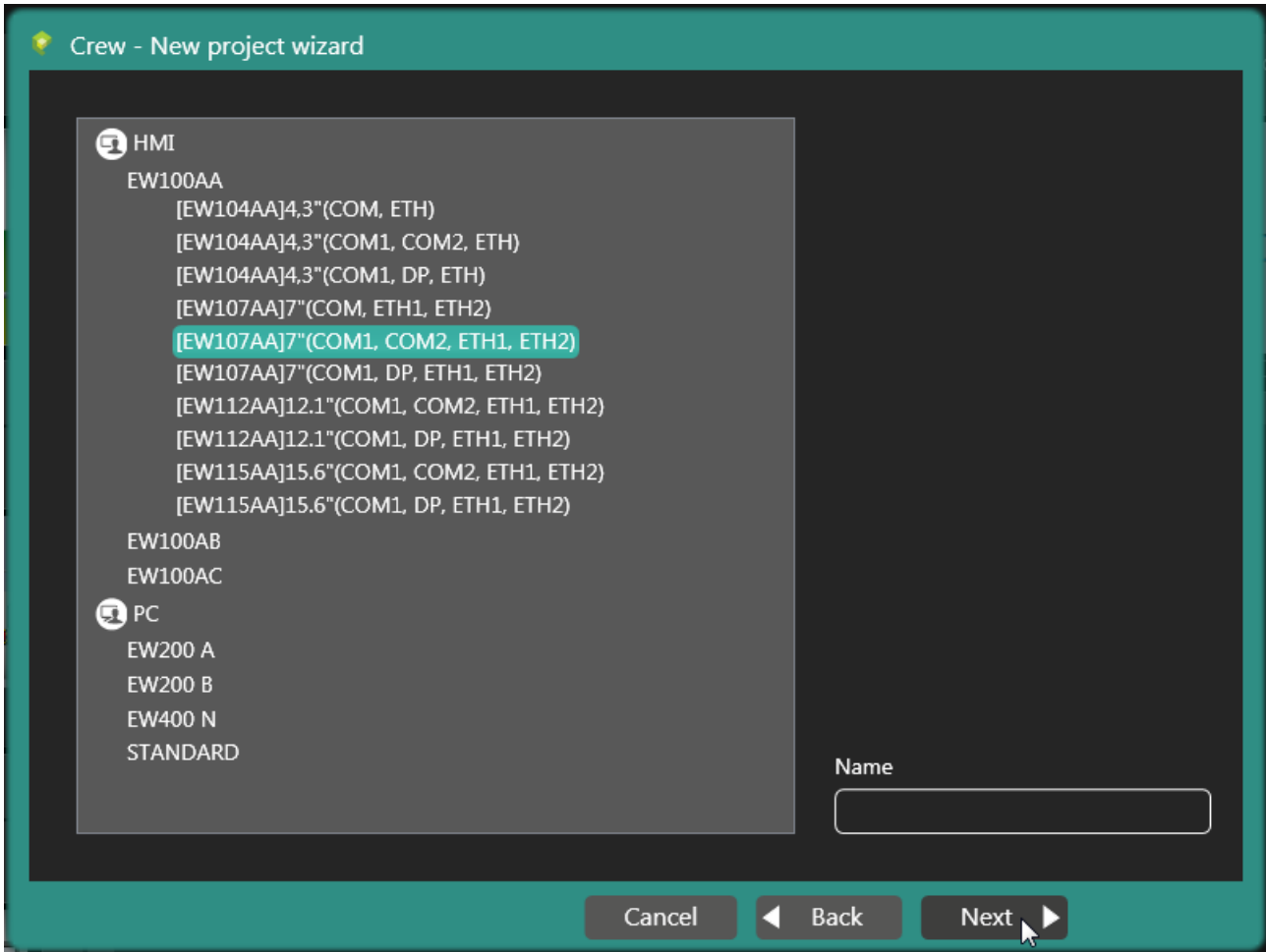
Folder: E:\Documenti LMASK\Crew\Crew Editor\New project 13

Cancel Next



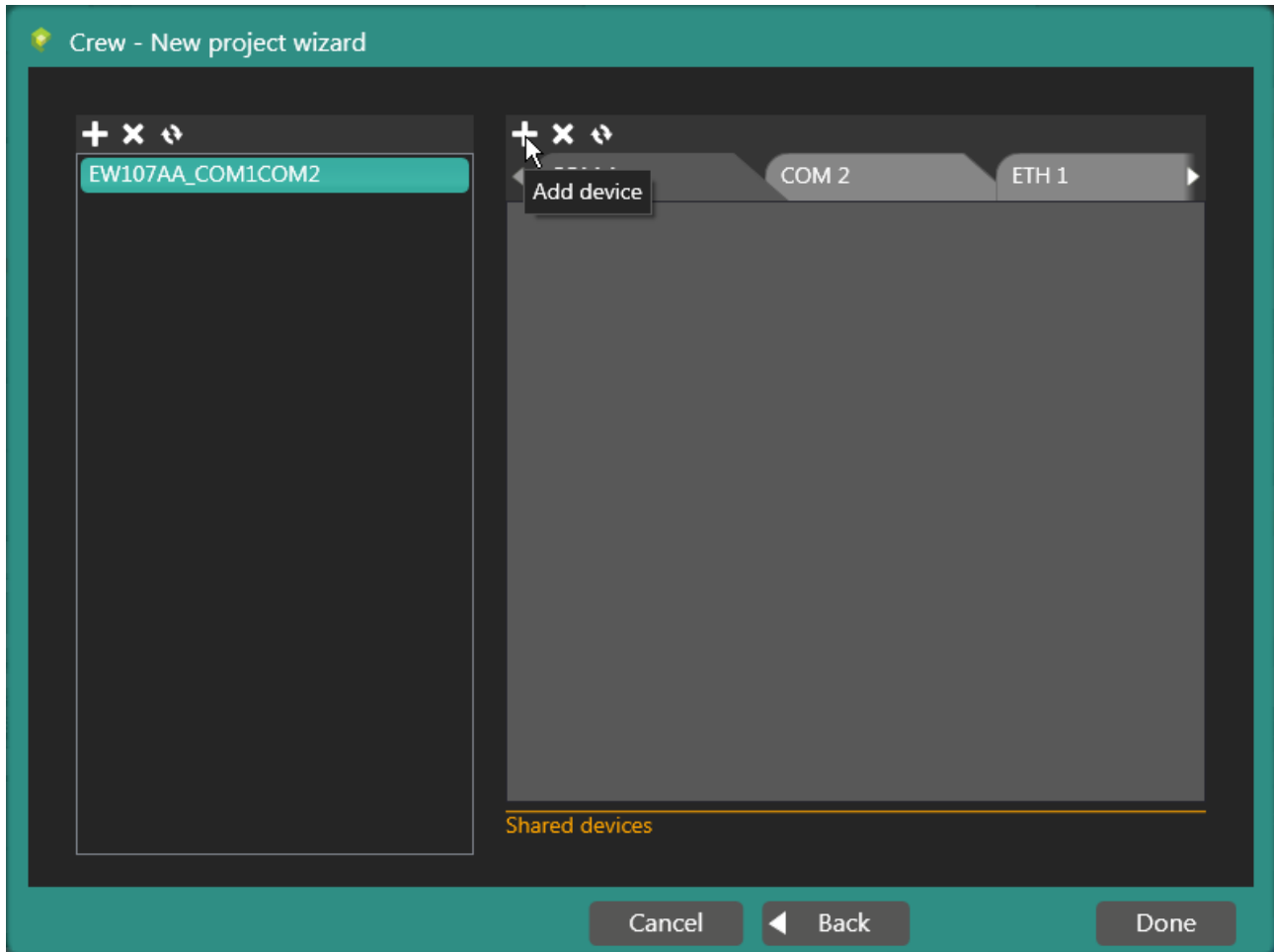
# CREW Manual

Choose which panel to enter into the project, then click “Next”.



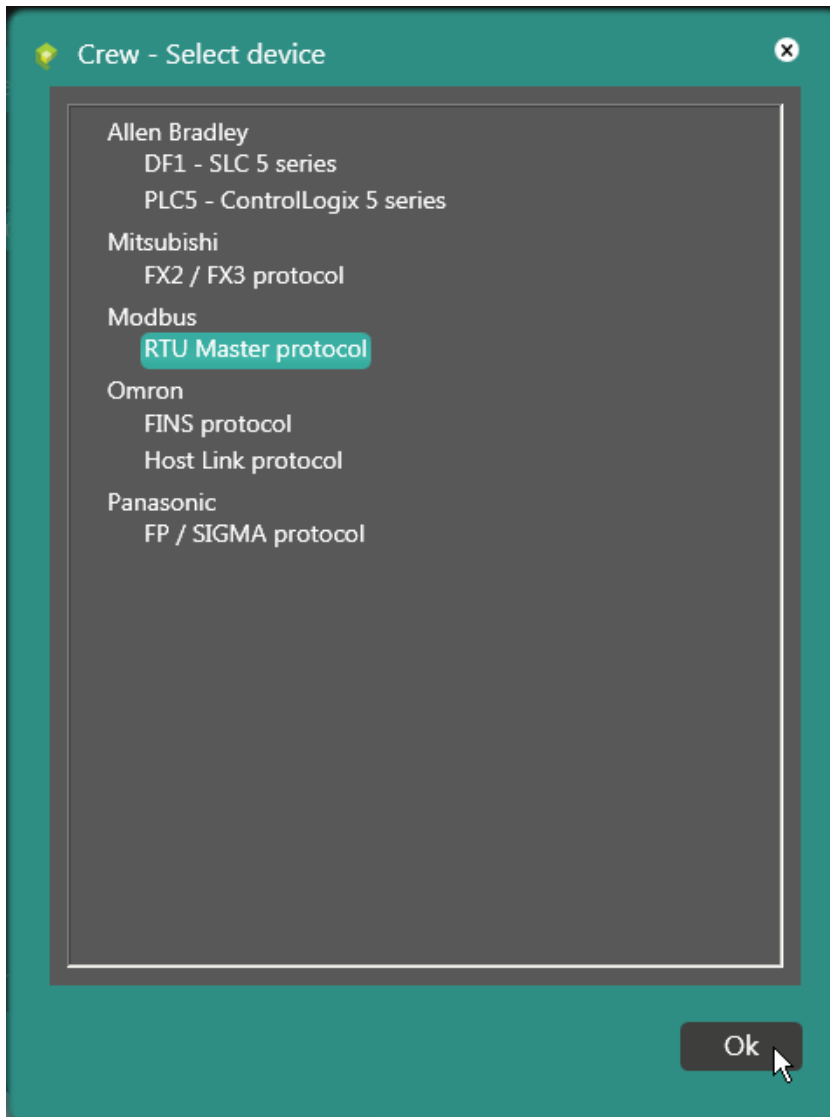
# CREW Manual

Click the relative key to add the protocol to the panel port (in our example we decide to add a protocol to the Ethernet port).



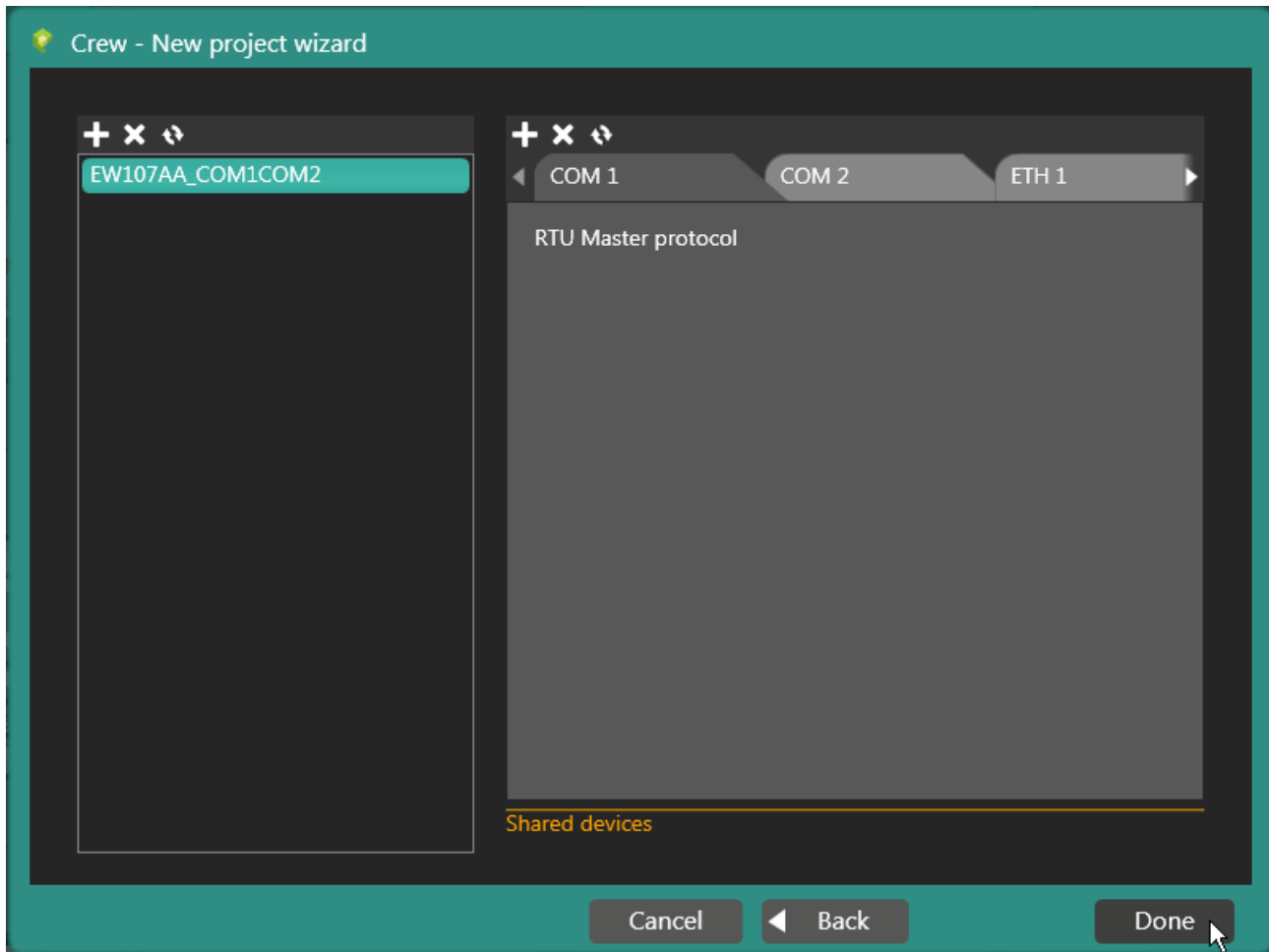
# CREW Manual

Then click “Ok”.



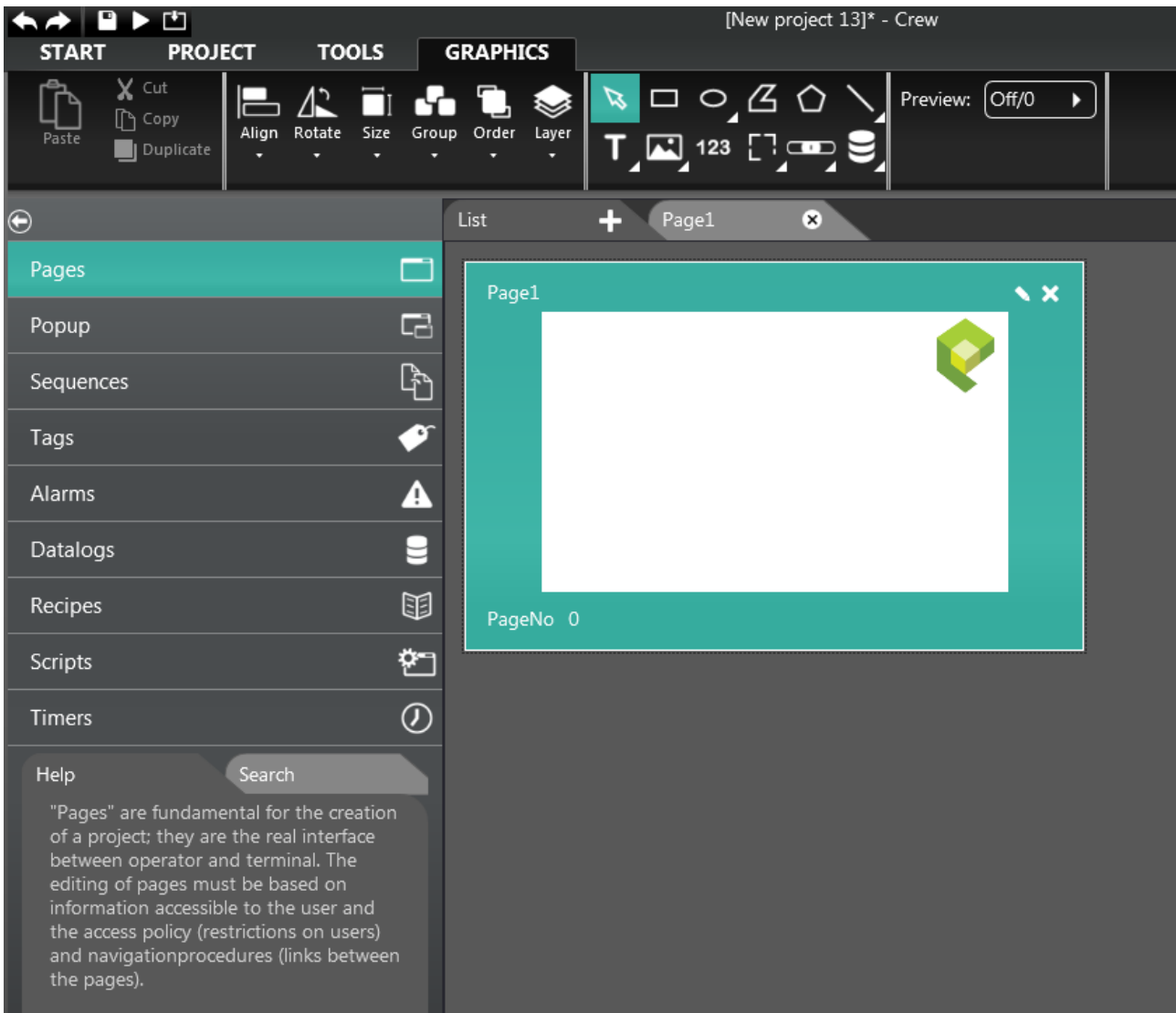
# CREW Manual

Then click "End" to confirm.



# CREW Manual

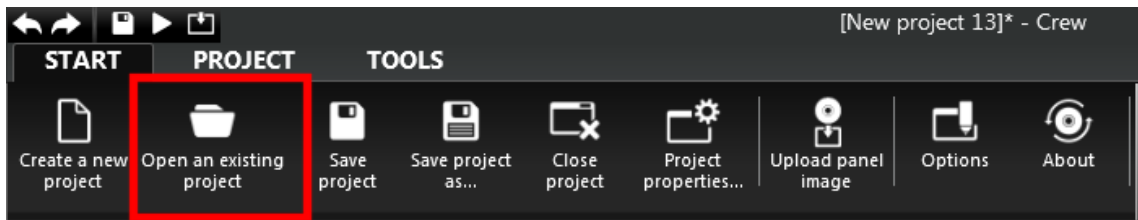
Once it is confirmed, Crew opens the work area where it is possible to start the editing the project (entering pages, variables etc.).



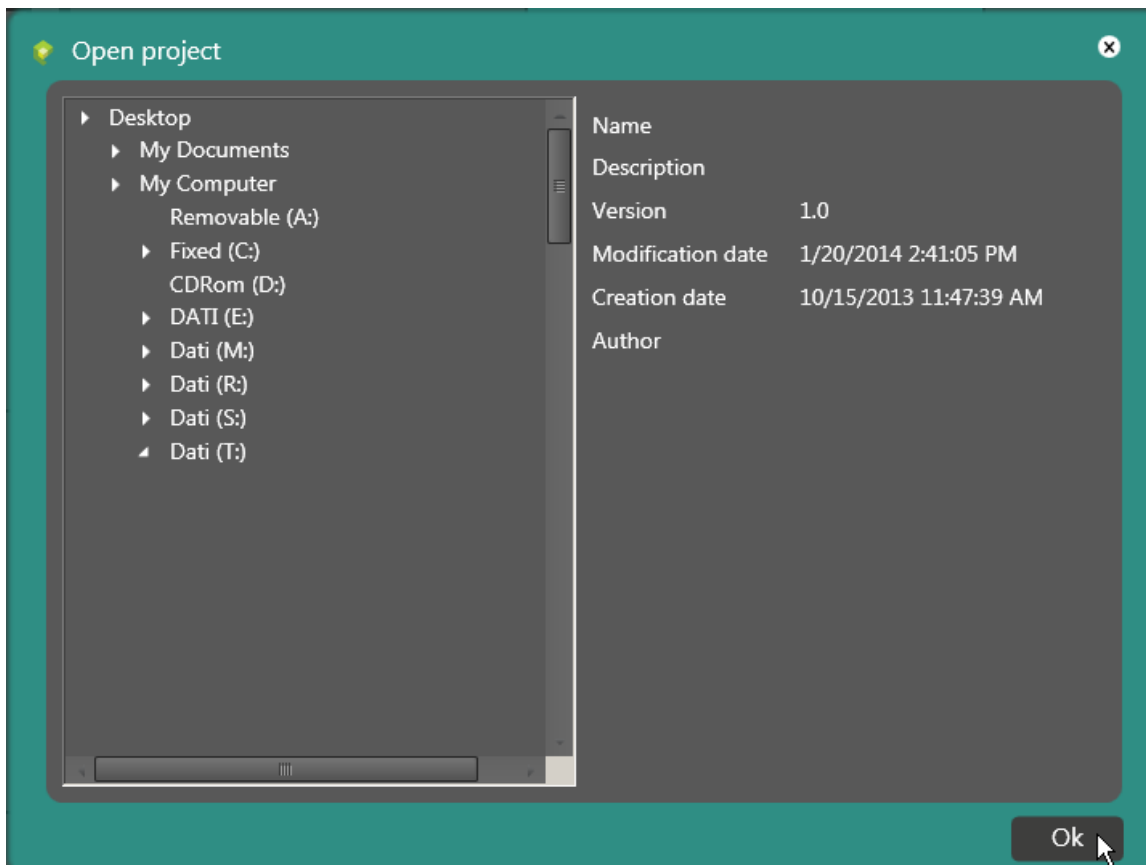
# CREW Manual

## Open an existing project

Click the “Open an existing project” icon.

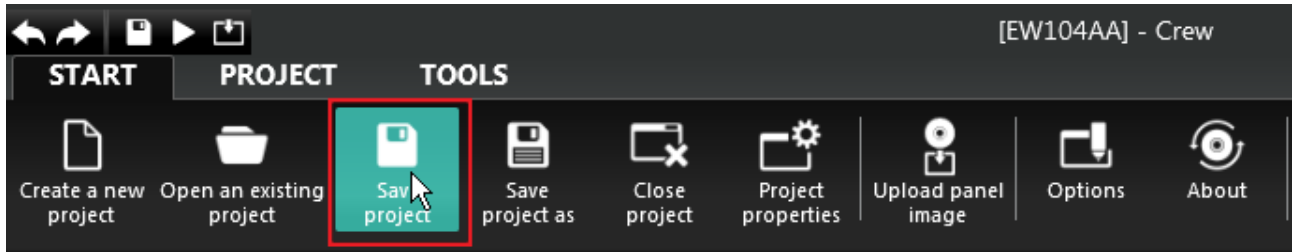


Select the project you wish to open and click “Ok”.



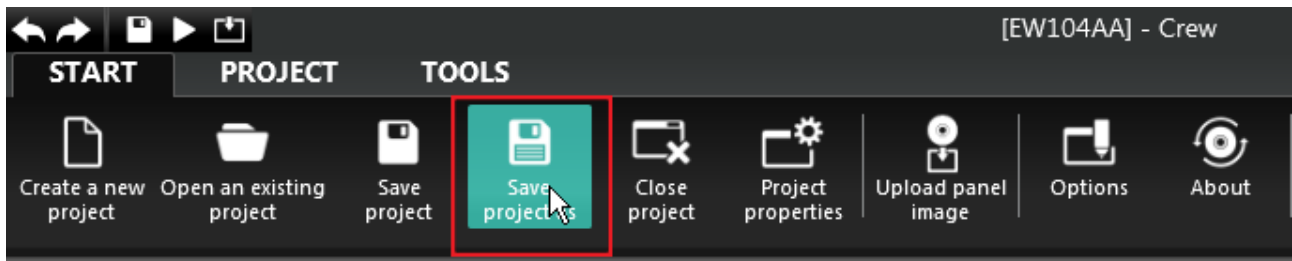
# CREW Manual

## Save project



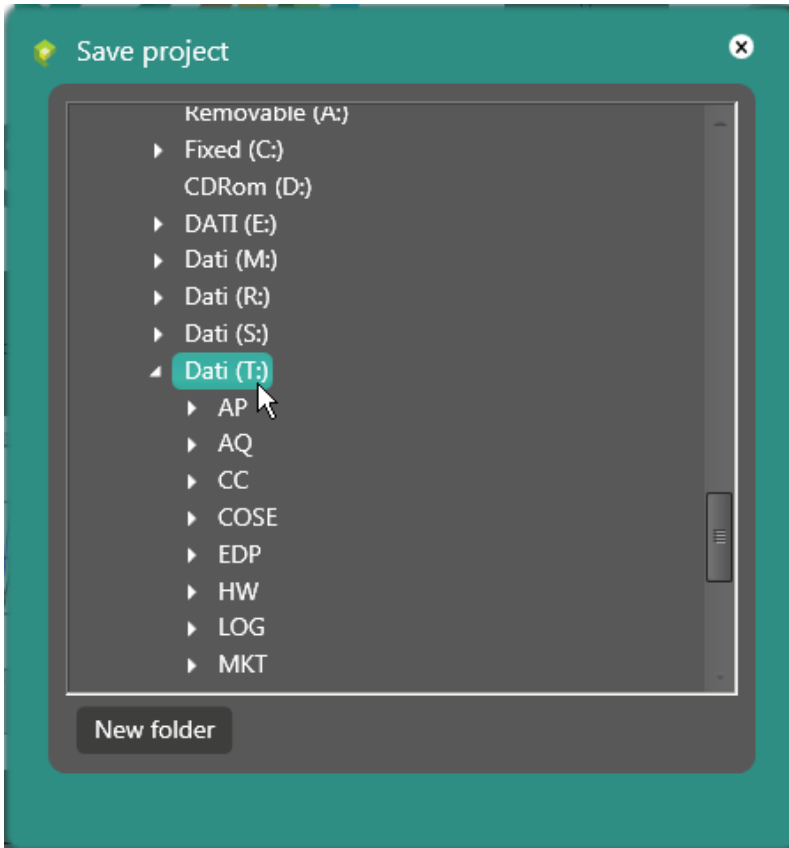
To save the project currently in execution.

## Save project as

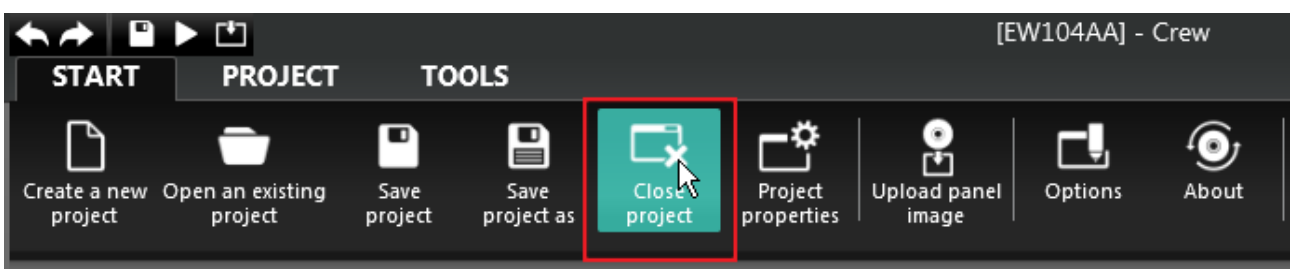


# CREW Manual

To save the project and choose the target path.



## Close project

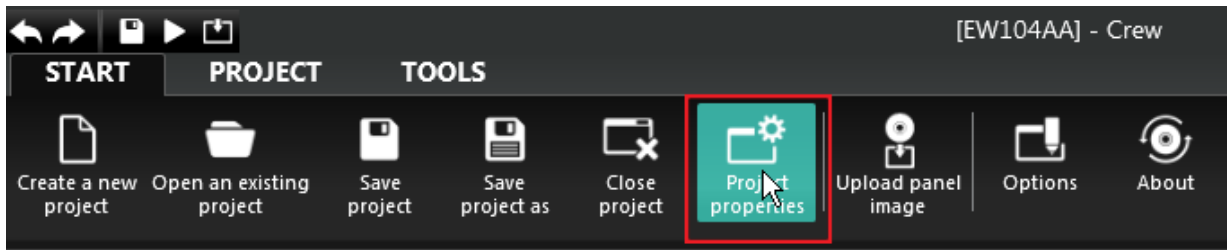


To close the currently open project.

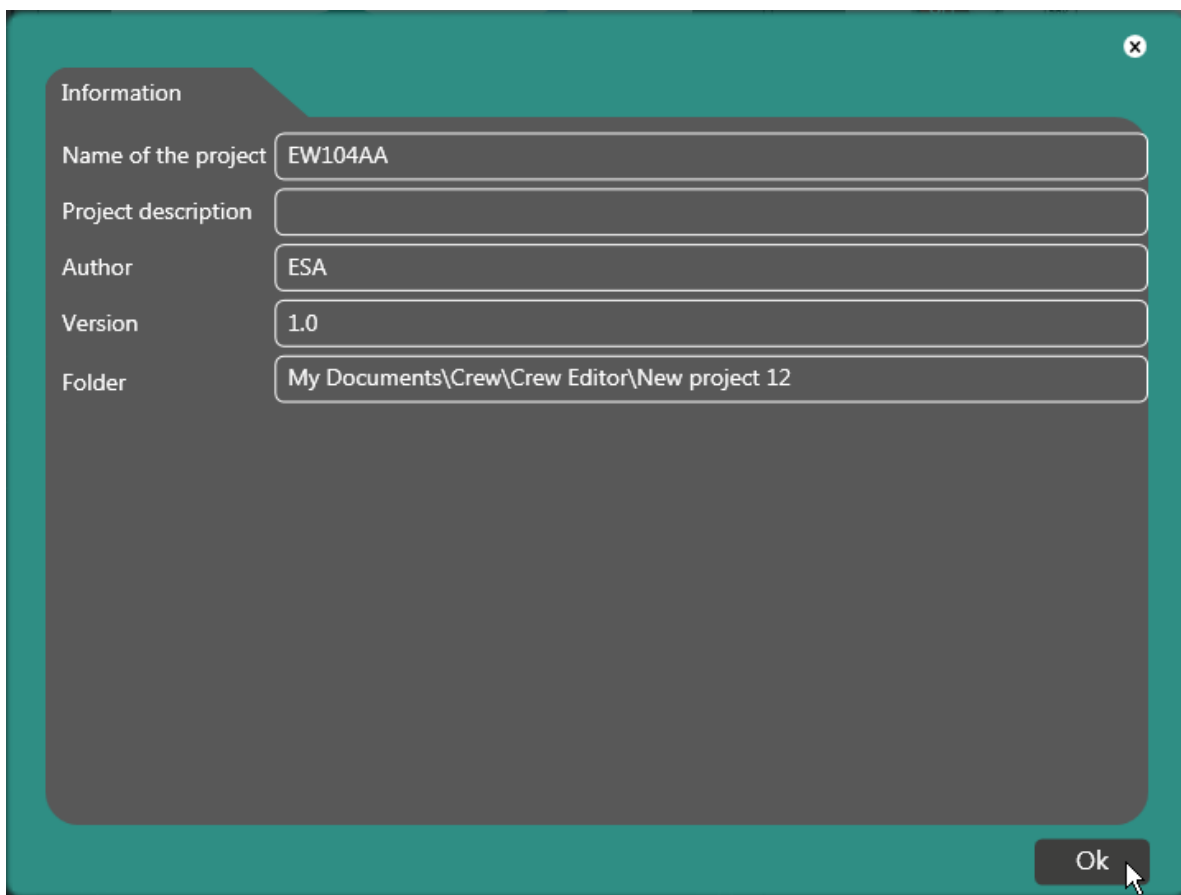


# CREW Manual

## Project properties



To view project details, such as Name, Description, Author, Version and Folder it belongs to.



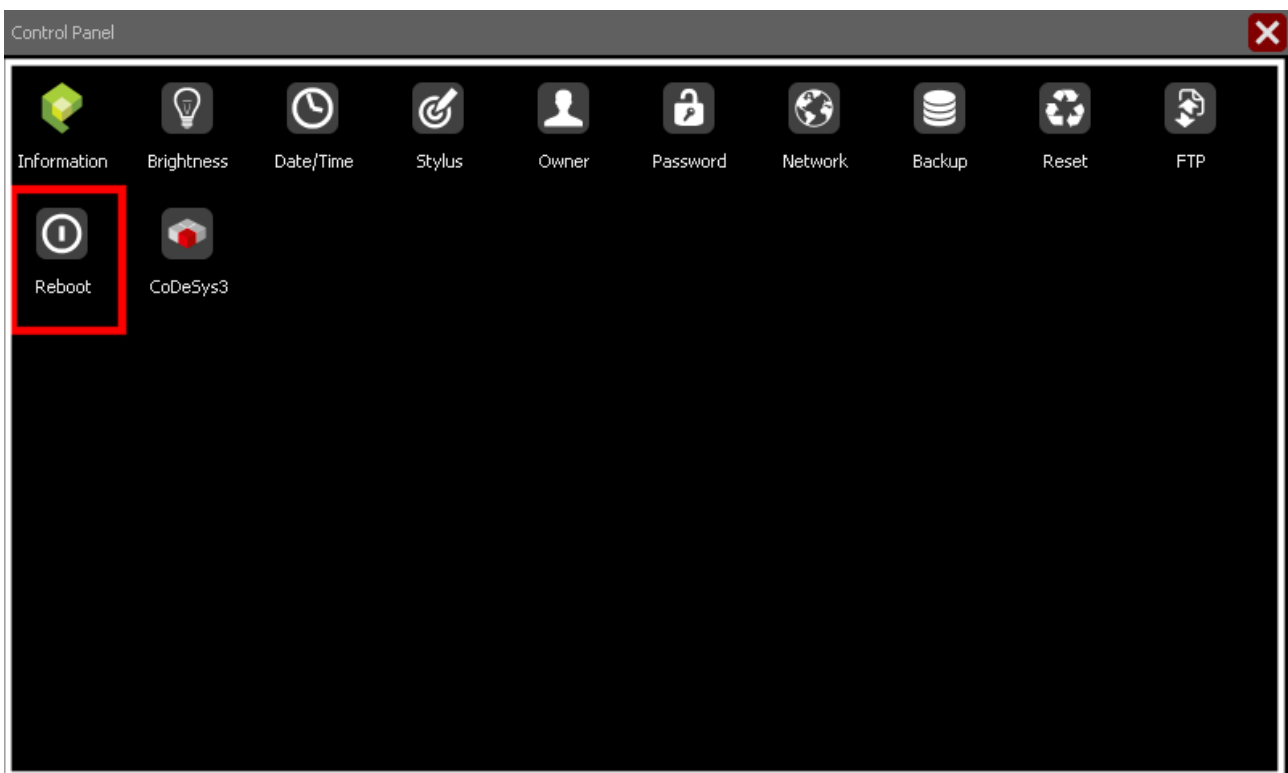
# CREW Manual

## Upload panel image

To update the “CE” image on the panel.

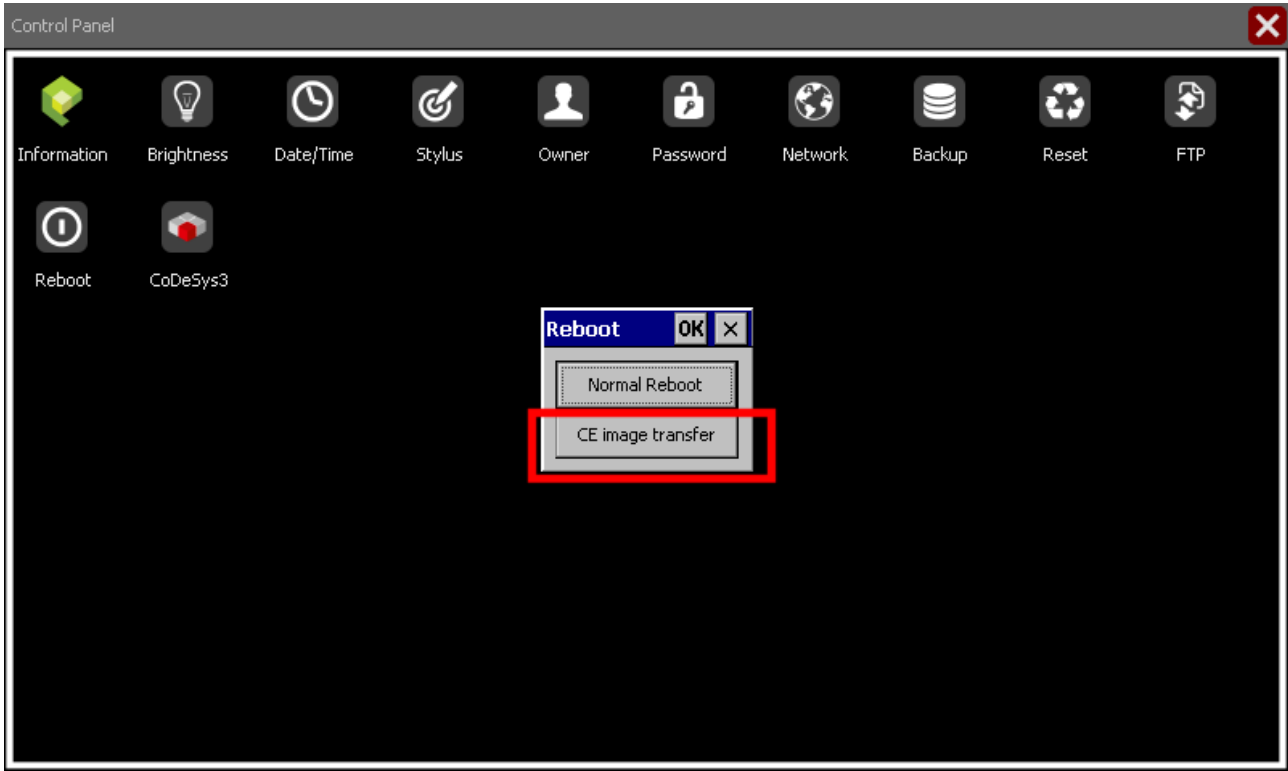
PROCEDURE:

From the Control Panel of the EW terminal, click the “Reboot” icon

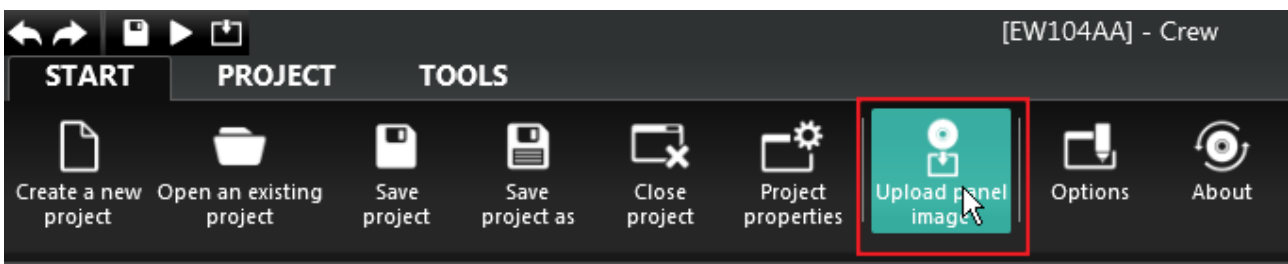


# CREW Manual

From the Control Panel of the EW terminal, click the “CE image transfer” icon

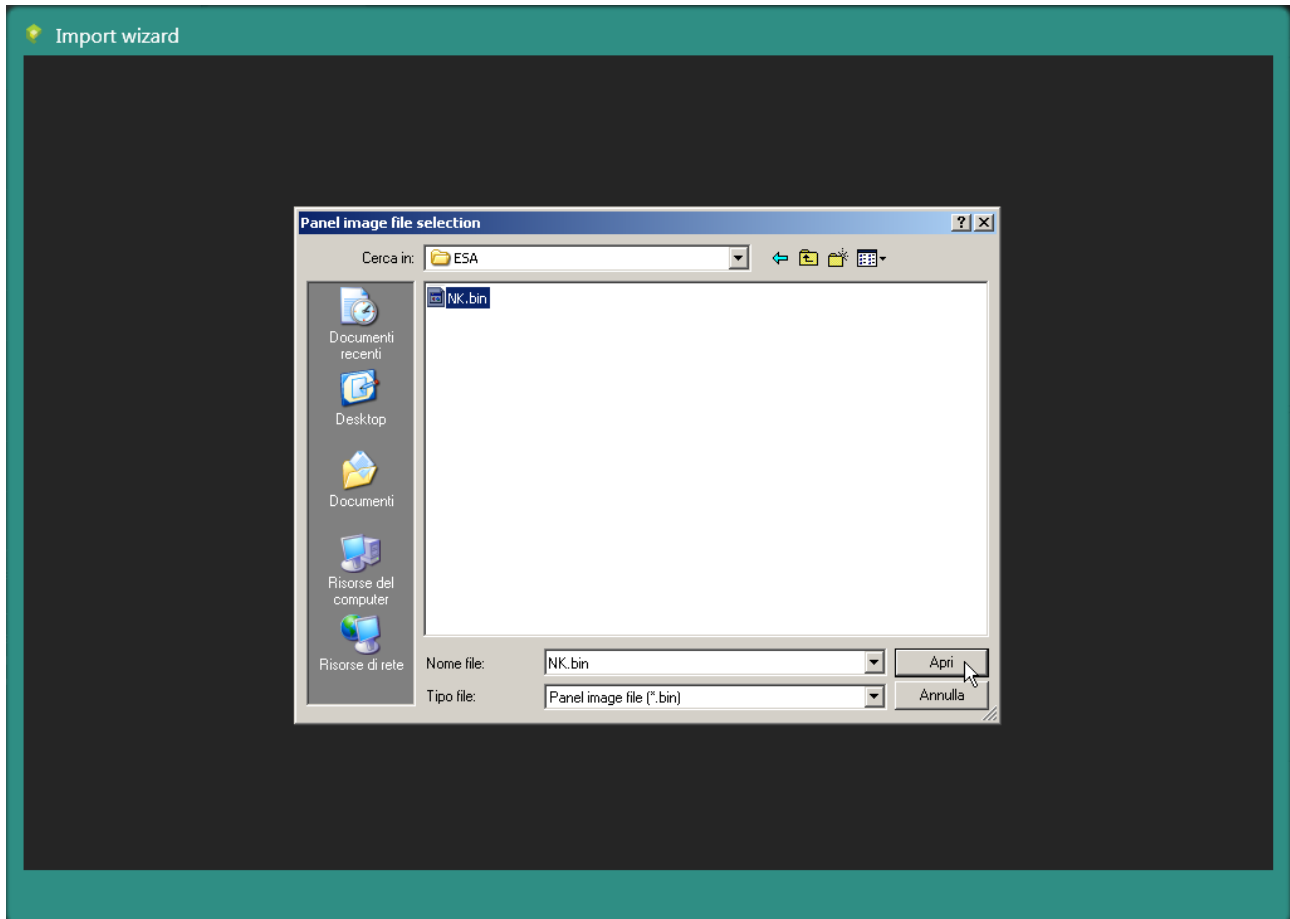


On Crew, select “Upload panel image”.



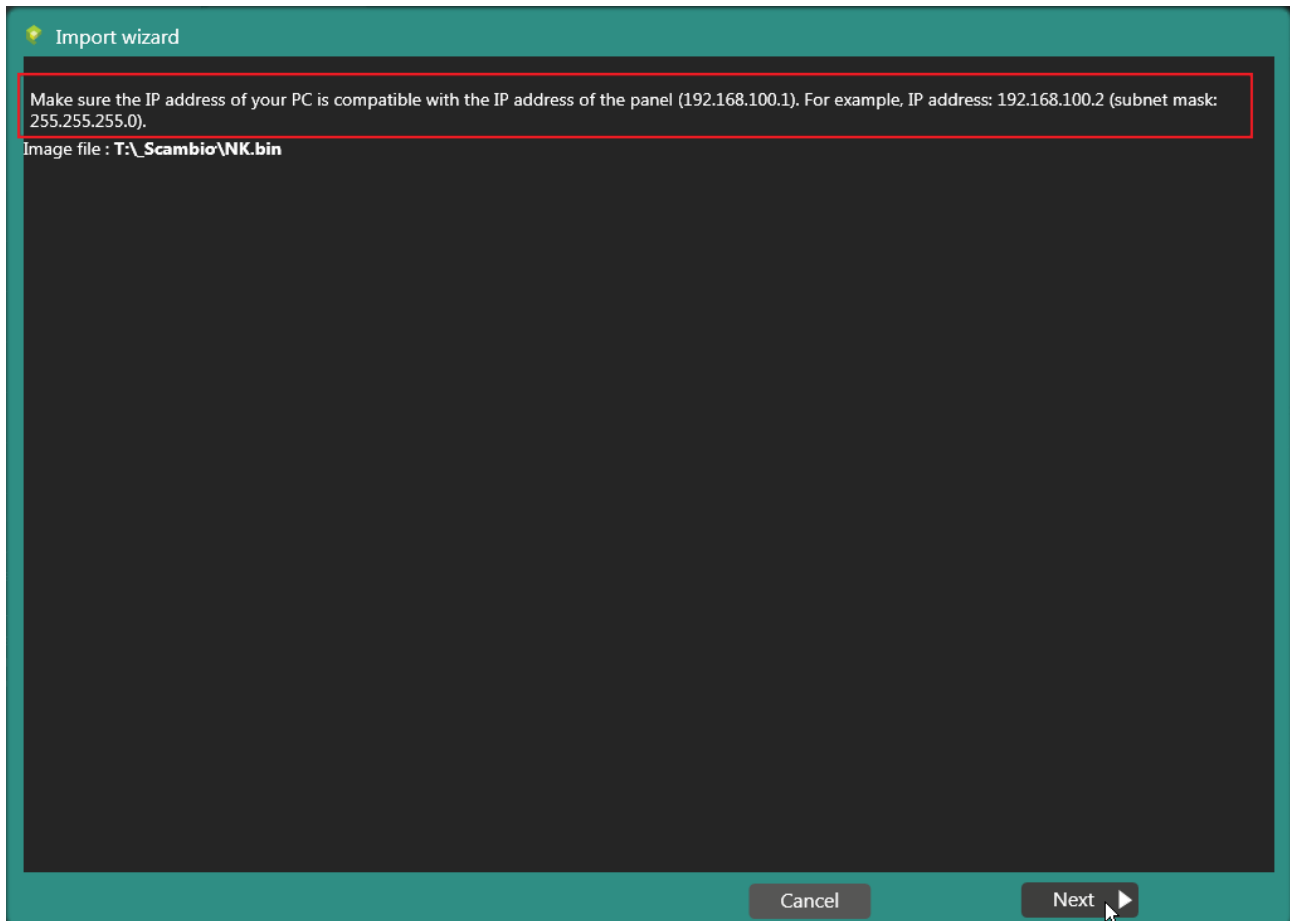
# CREW Manual

Select the "NK.bin" file.



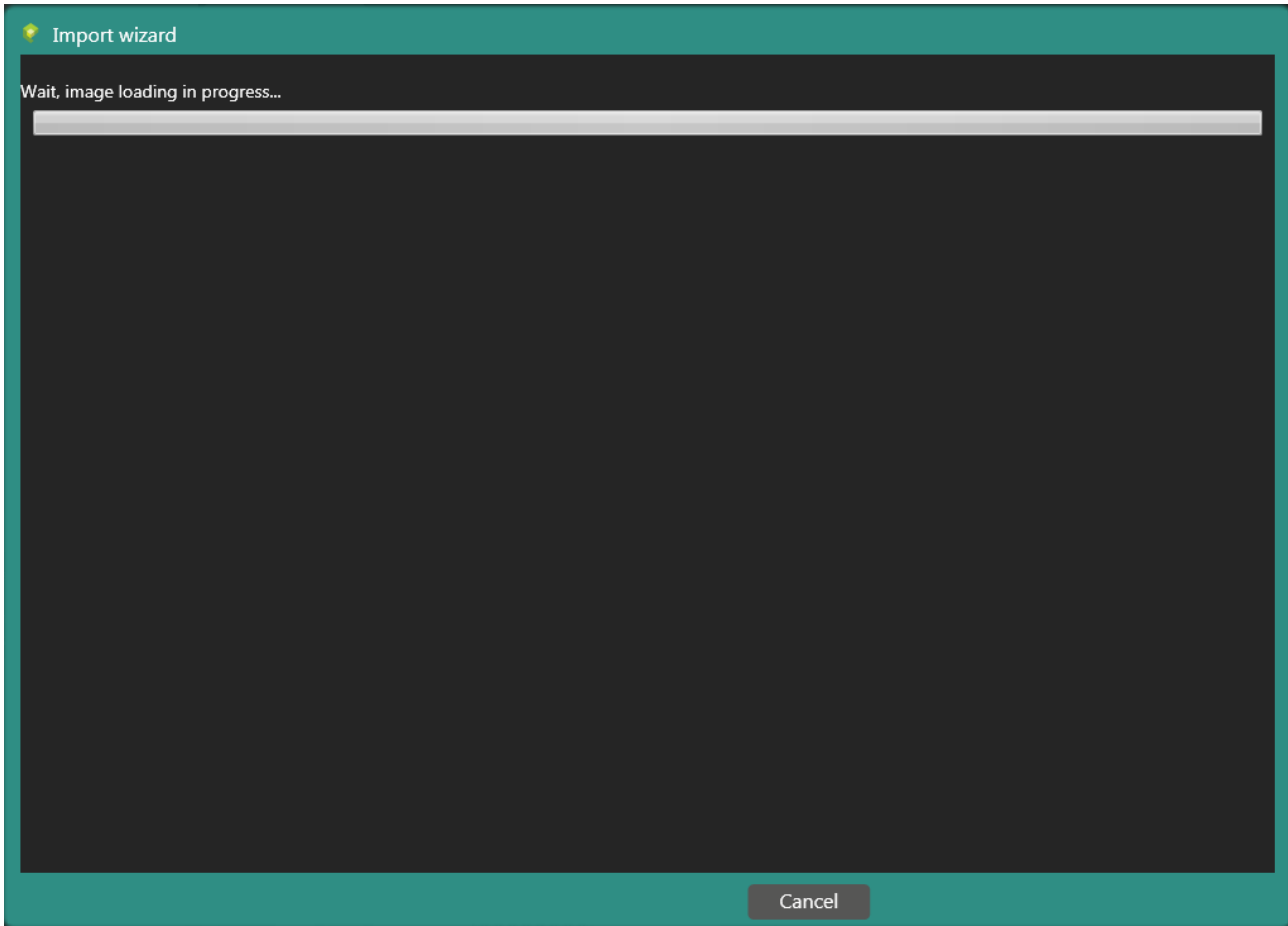
# CREW Manual

Configure the PC's IP address, then click "Next".



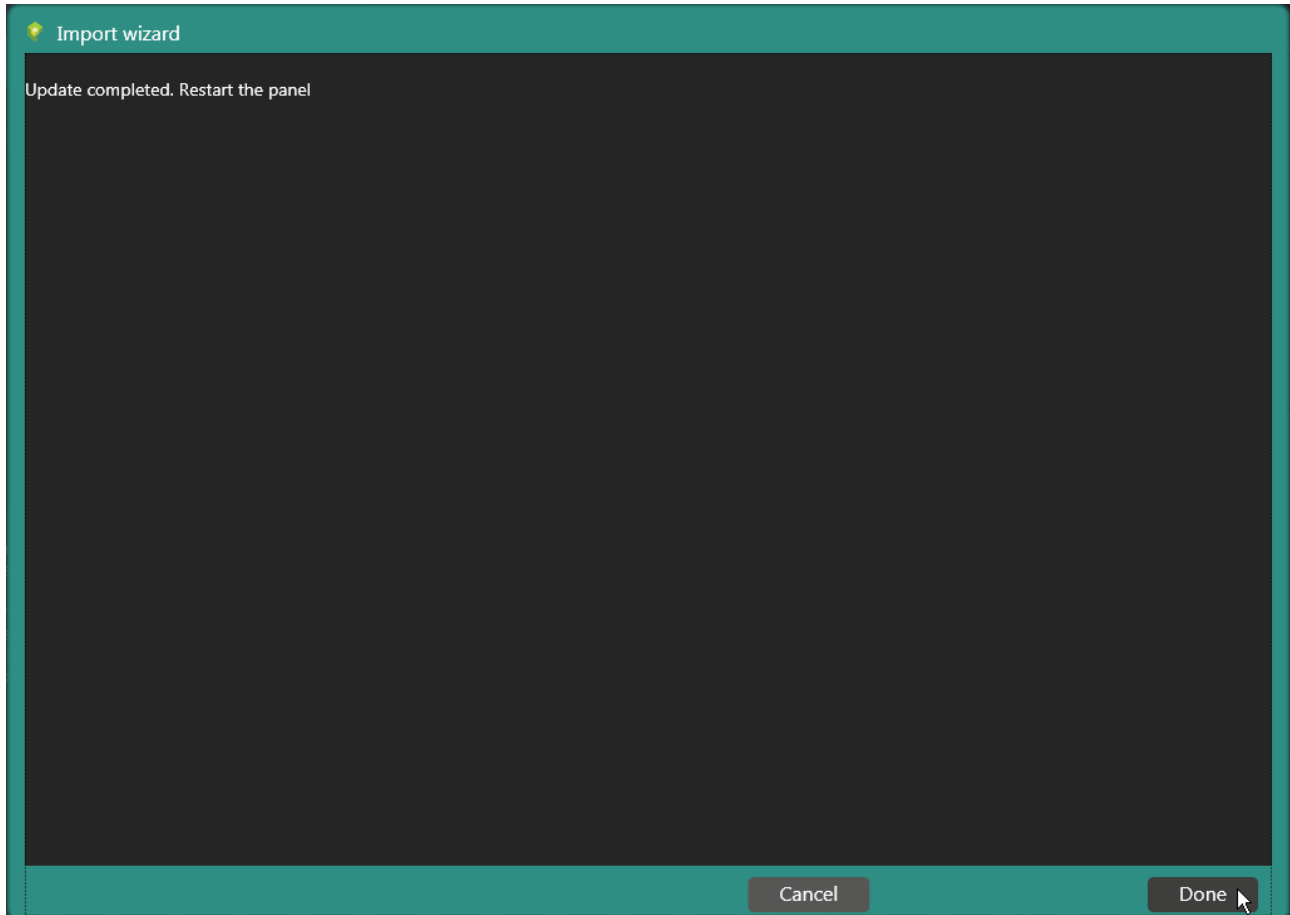
# CREW Manual

Wait for the image uploading procedure to finish.



# CREW Manual

At the end click “Done”.



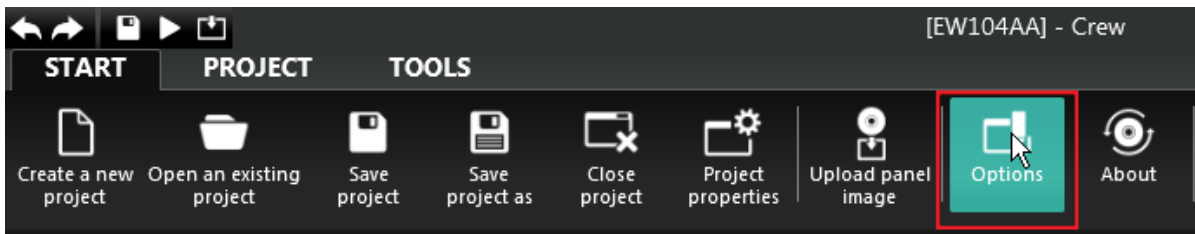
When the download is correctly completed, the panel restarts automatically. Wait for the terminal to ask to calibrate the touch screen (for this procedure, see section "[Control Panel](#)", paragraph "Stylus").



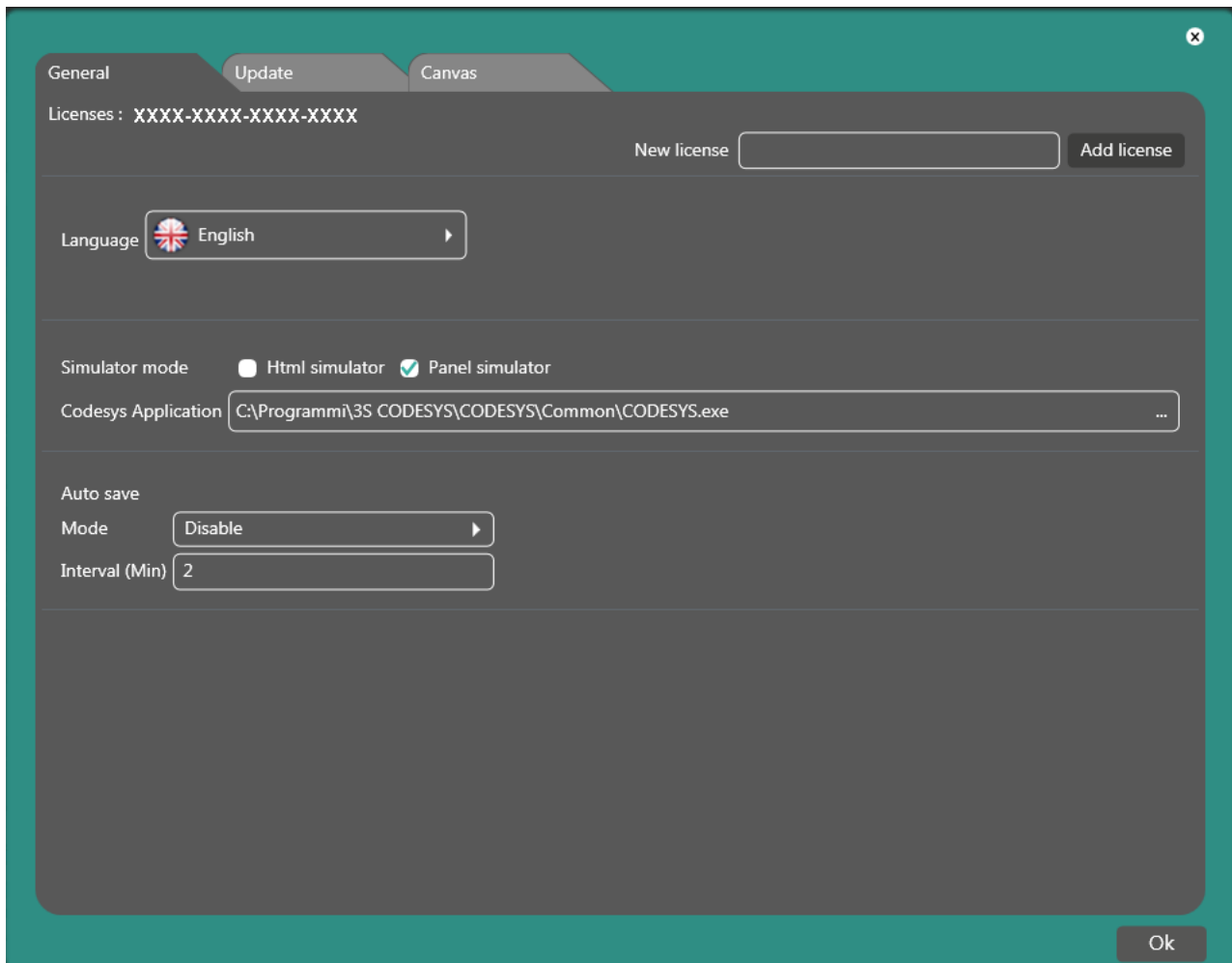
Note: If the version of the image is the same as the one currently on the terminal, touch screen calibration is not requested.

# CREW Manual

## Options



Click the “Options” menu to open the following window.



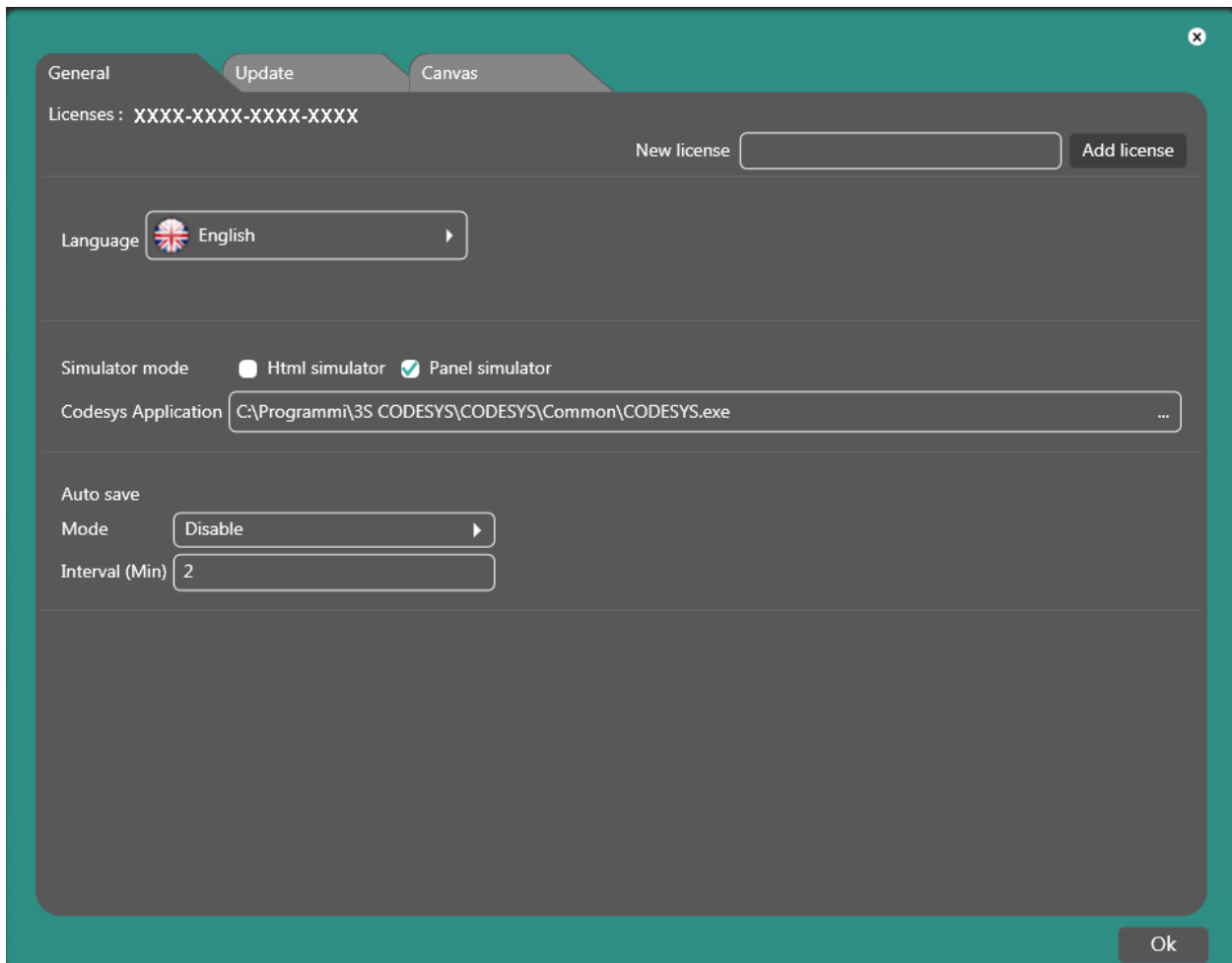


# CREW Manual

The Options window contains the following entries:

- General
- Update
- Canvas (choice of graphic design view settings inside Crew)

## General



# CREW Manual

From this mask it is possible to:

- View the serial number of the main Crew licence and enter any other licenses (for example the FDA licence).

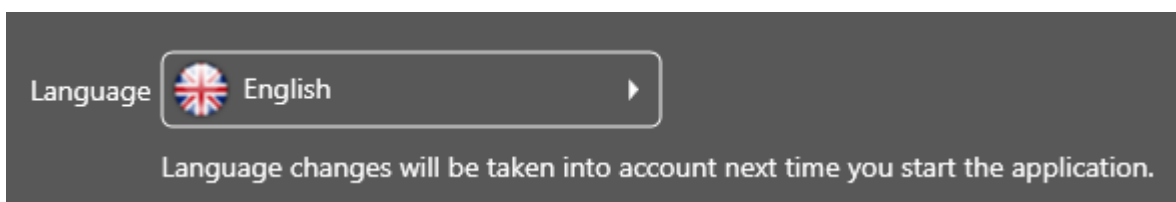


Note : For more information on the FDA regulation, refer to the "[FDA 21 CFR Part 11](#)" section

- Select the Crew default language.



The new language is applied the next time Crew is started up.

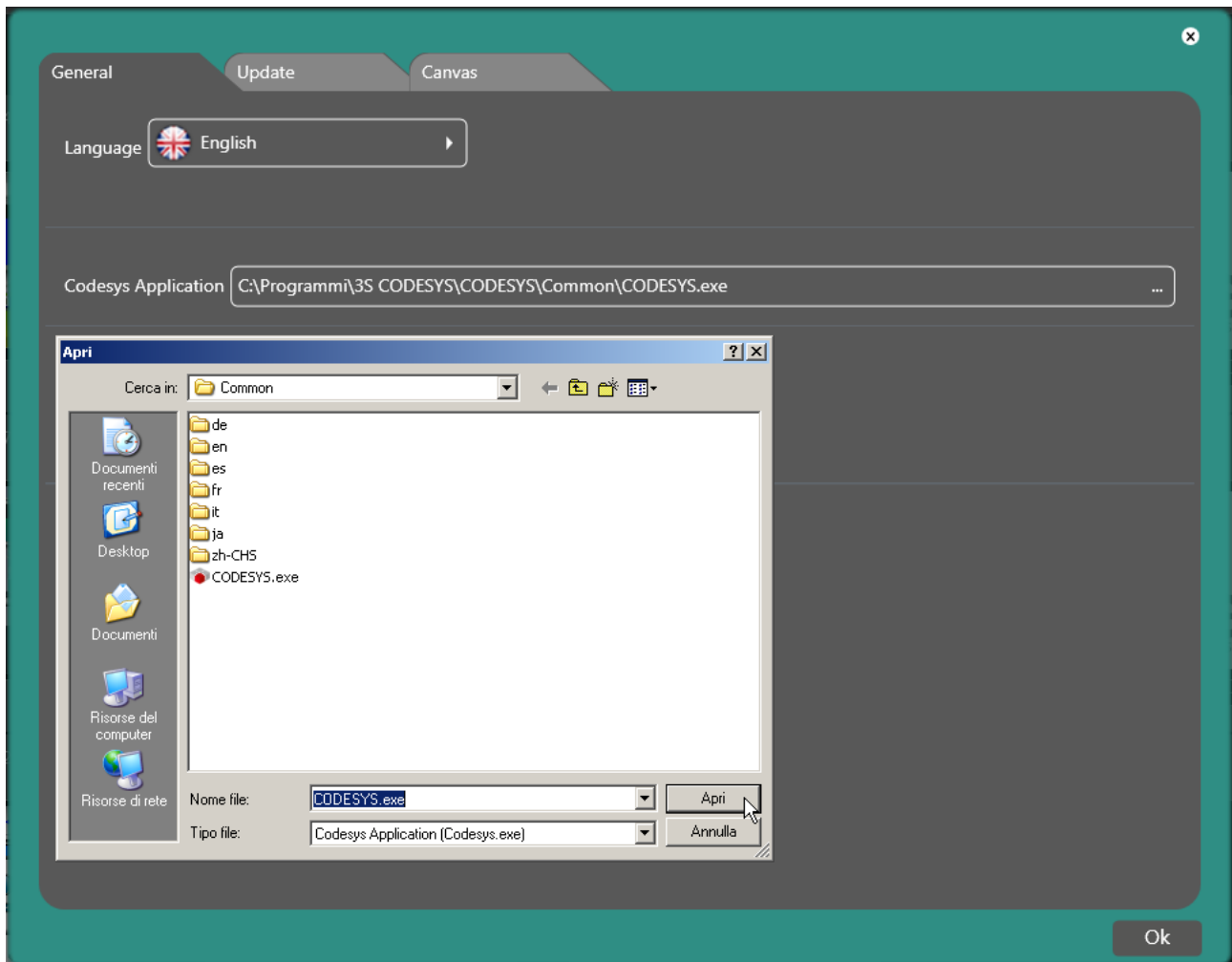


- Choose the type of Runtime simulator (HTML or PANEL).

# CREW Manual

- CODESYS Application:

To select the path where you can choose to open the "CODESYS" application from.



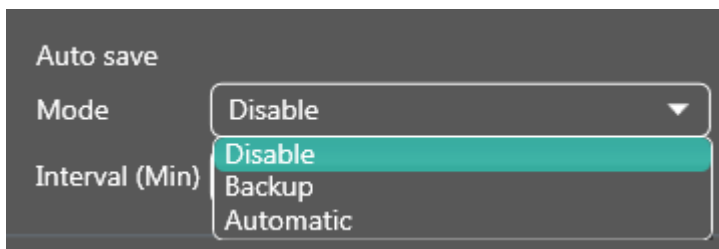
# CREW Manual

- Automatic storage:

To enable or disable automatic project storage.



It is possible to decide whether to enable the option or not by choosing from the following options.



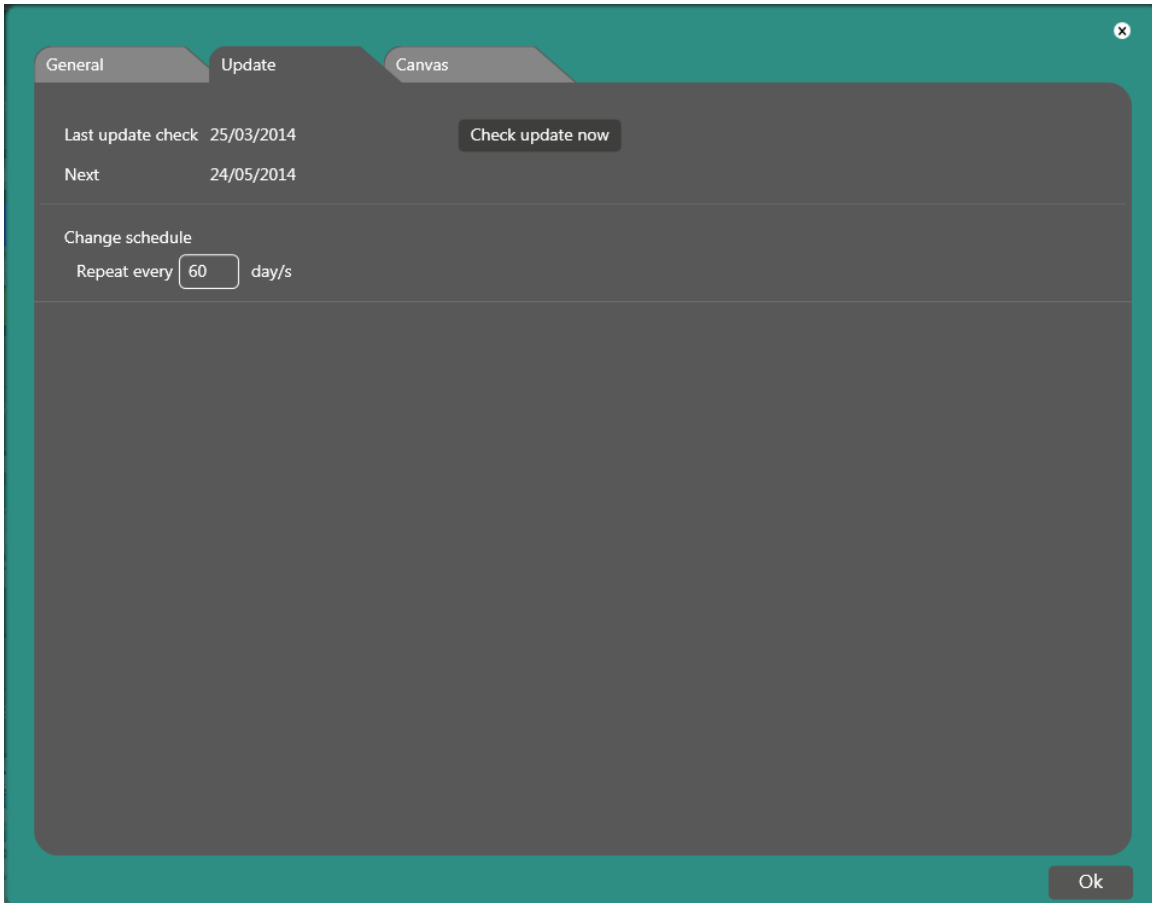
Disabled: automatic storage is never executed.

Backup: the project back up is executed, saving it in a different path from the current project.

Automatic: the project is stored automatically at the frequency established by the user.

# CREW Manual

## Update



From this mask it is possible to:

Check whether there are any recent Crew updates (Internet connection necessary).



# CREW Manual

View the data of the last completed update search and the date of the next scheduled update search.

Last update check 27/03/2014

Next 26/05/2014

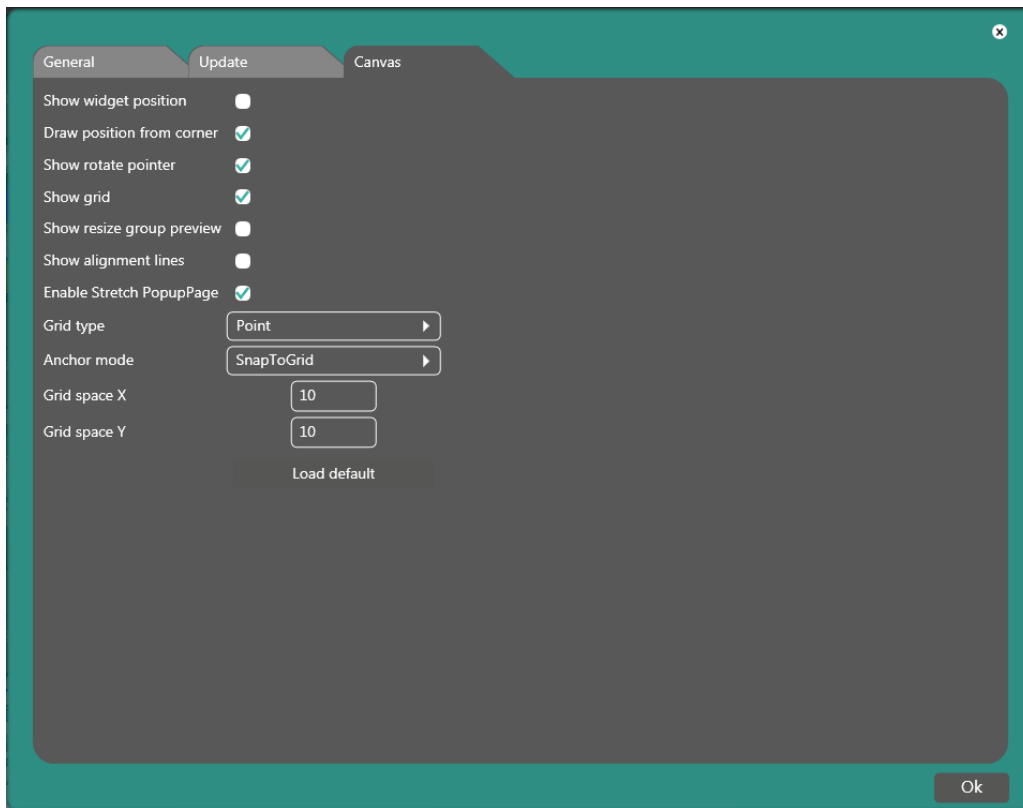
Set the next update search.

Change schedule

Repeat every  day/s

# CREW Manual

## Canvas



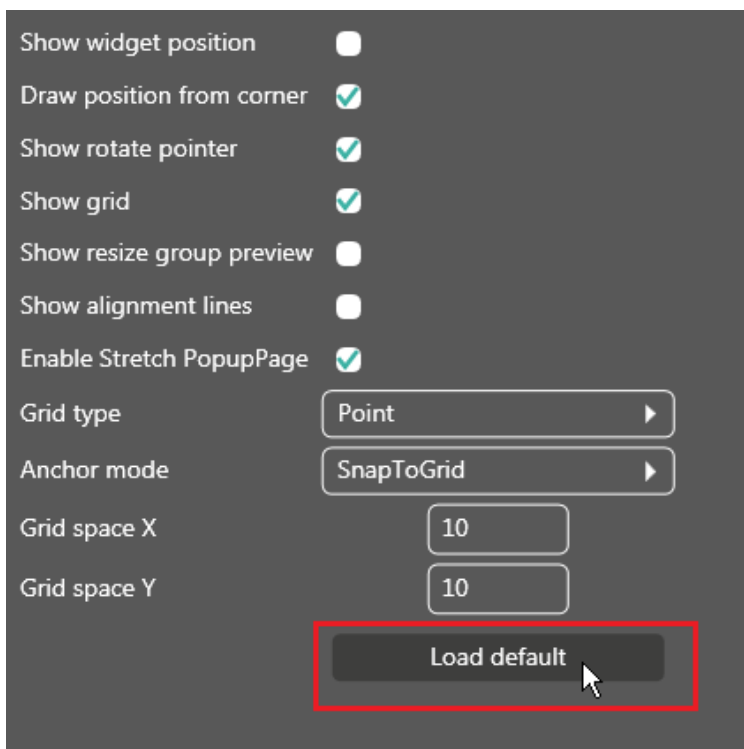
From this mask it is possible to select the following graphic settings to view the page in a Crew project and the elements contained in it:

- Show object position (disabled by default)
- Draw angle position (enabled by default)
- Show rotation pointer (enabled by default)
- View grid (enabled by default)
- Show preview of unit resizing (disabled by default)
- Show alignment lines (disabled by default)
- Enable Popup stretch (enabled by default)
- Grid type (dot / dash-dot / dash / line /dash-line)
- Anchoring mode (none / align grid / align object)

# CREW Manual

- X grid space
- Y grid space

Plus, it is also possible to load the default settings.

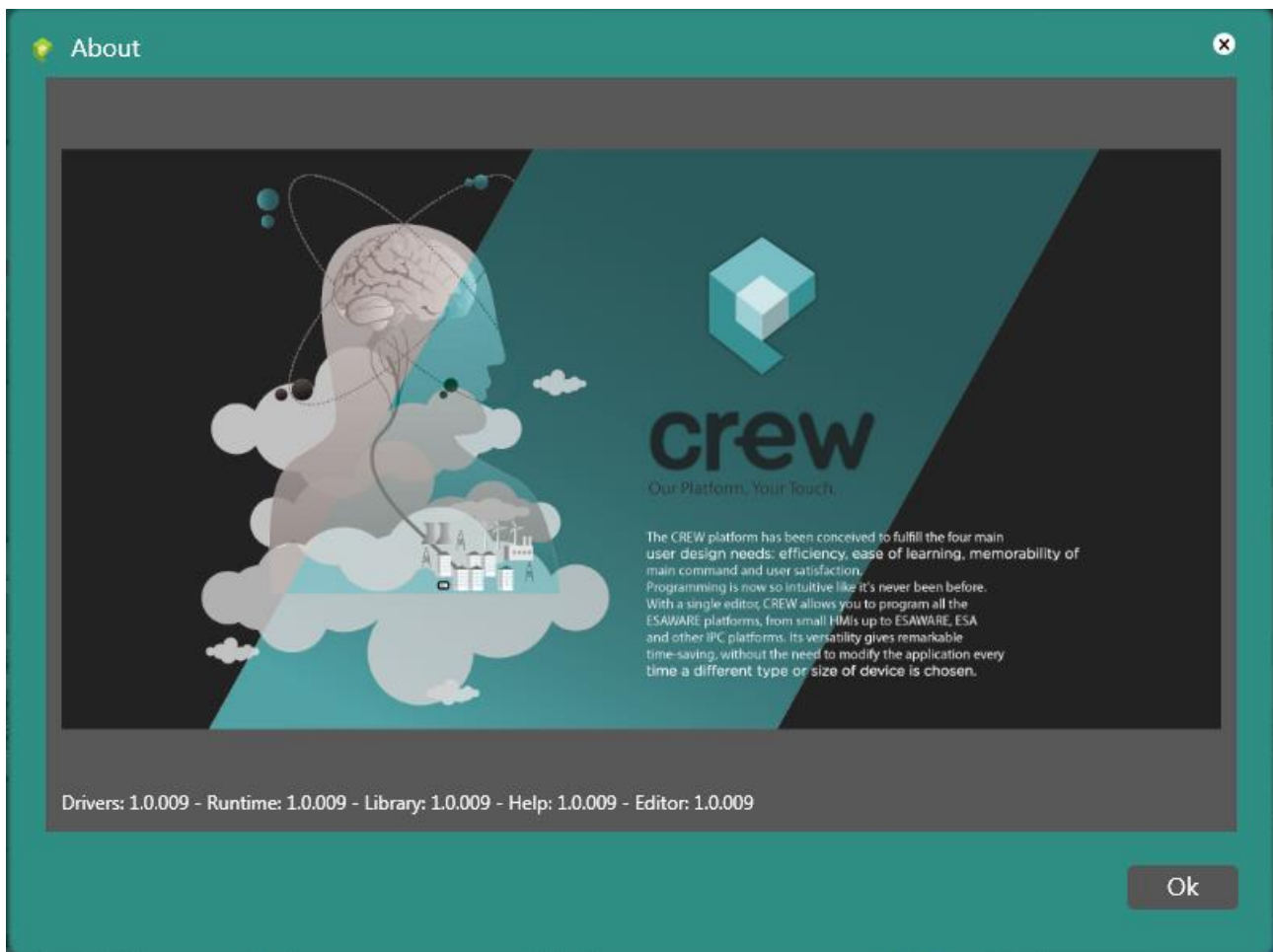




# CREW Manual

## About

To obtain general information about Crew. For example, the software, Runtime and driver version.



# CREW Manual

## PROJECT Menu



When a project is open, click “PROJECT” to open the submenu with the following properties:

- Panel
- Languages
- Dictionary
- Groups

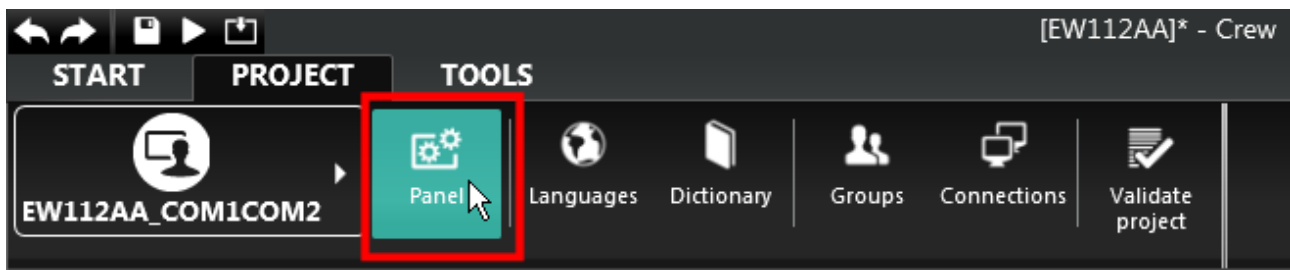
# CREW Manual

- Connections
- Project validation

Using “Explore Project” it is possible to view and edit the parts that comprise the project itself:

- Pages
- Popup
- Sequences
- Tags
- Alarms
- Datalog
- Recipes
- Script
- Timer

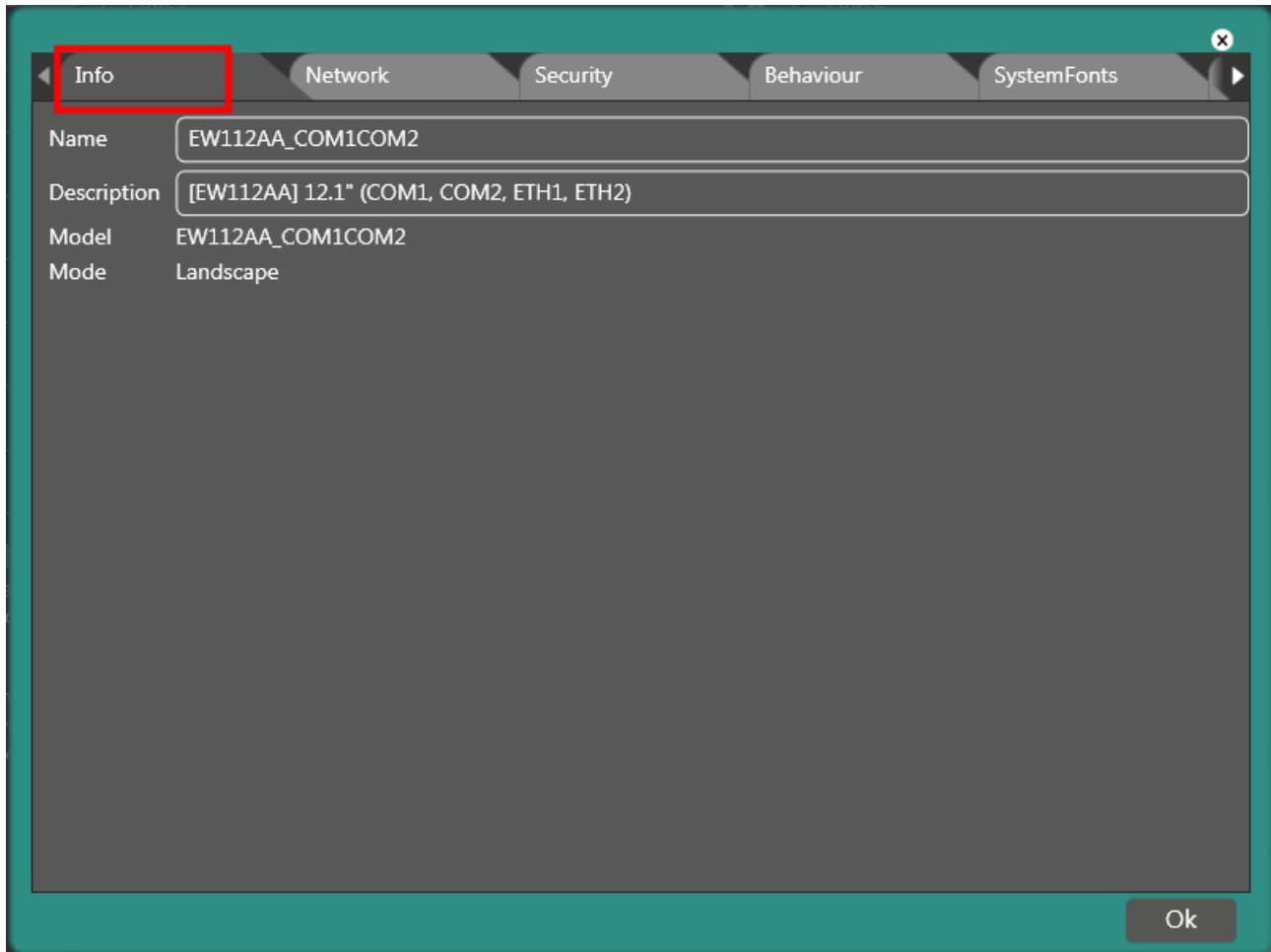
## Panel



The “Panel” option shows a window where the user can view and enter the panel properties, such as: information, network, safety, system font behaviour and events.

# CREW Manual

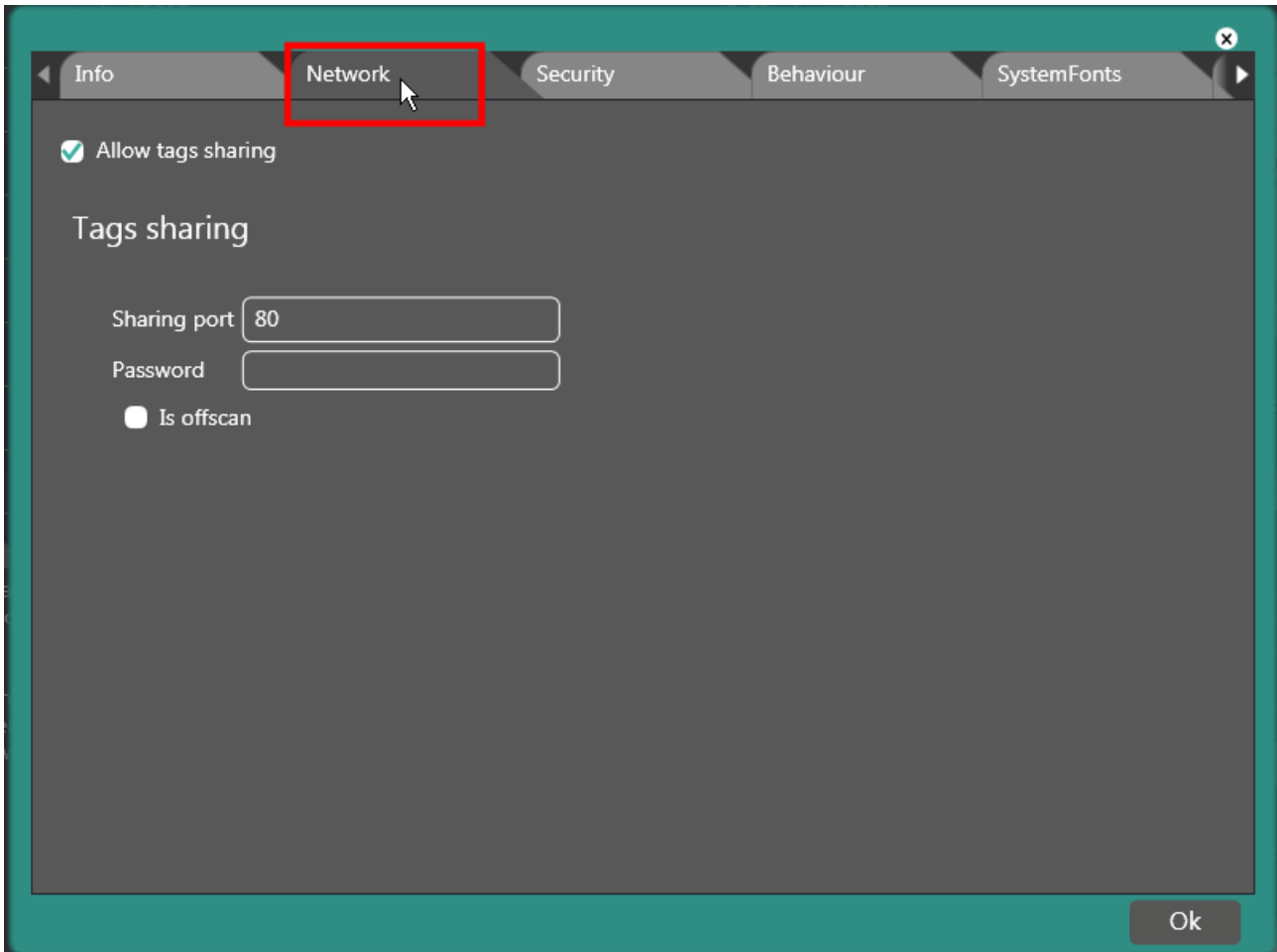
## Info



From the “Info” mask it is possible to view - and edit if necessary - the name and description attributed to the terminal, and view the model and set-up (vertical or horizontal).

# CREW Manual

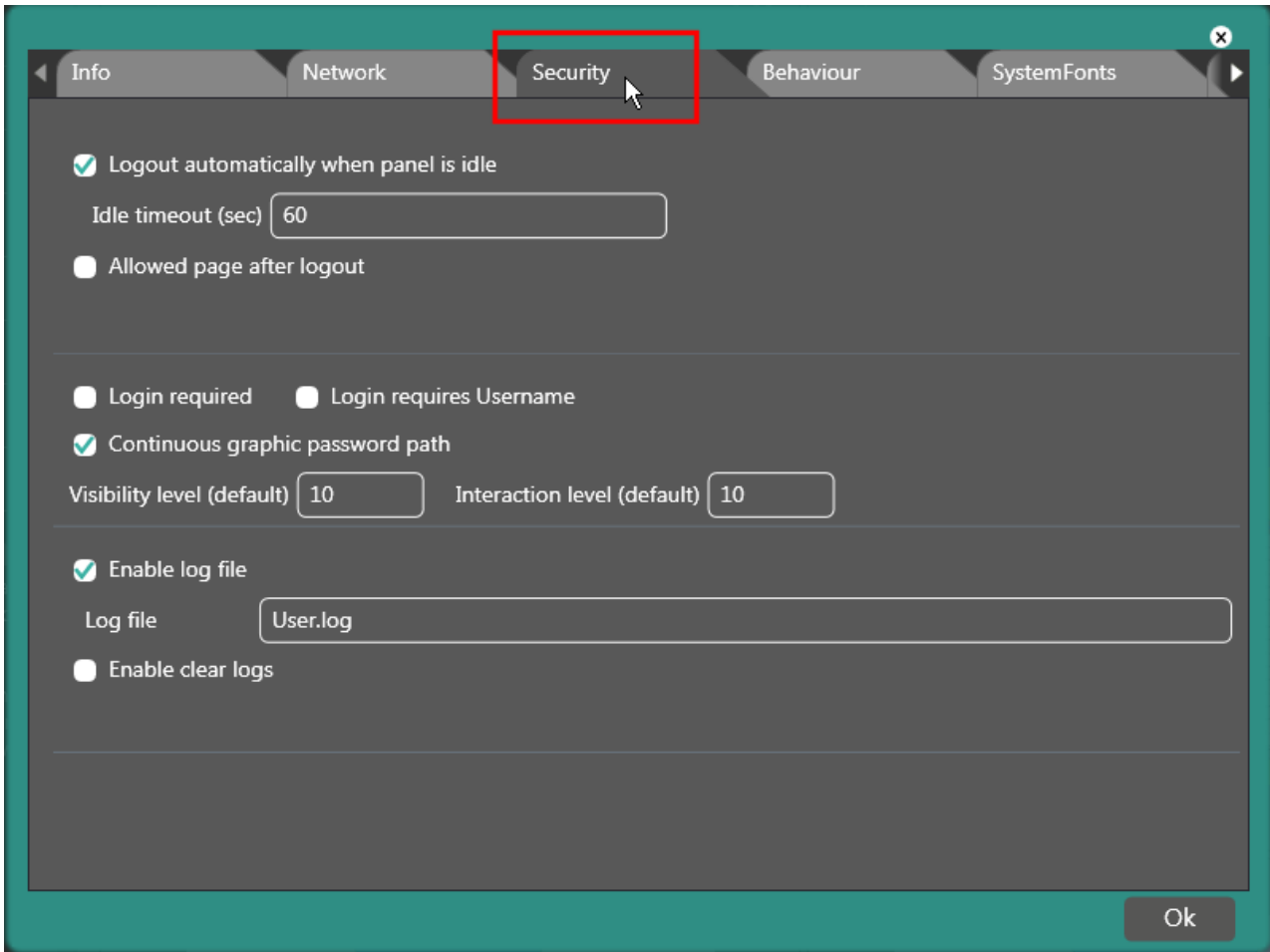
## Network



The “Network” option makes it possible to enable sharing the Tag and selecting the port number and, possibly, the possibility of accessing sharing with a password. The “Offscan” mode is an attribute of the Tag that determines whether it can be enabled or disabled to be interrogated by the device.

# CREW Manual

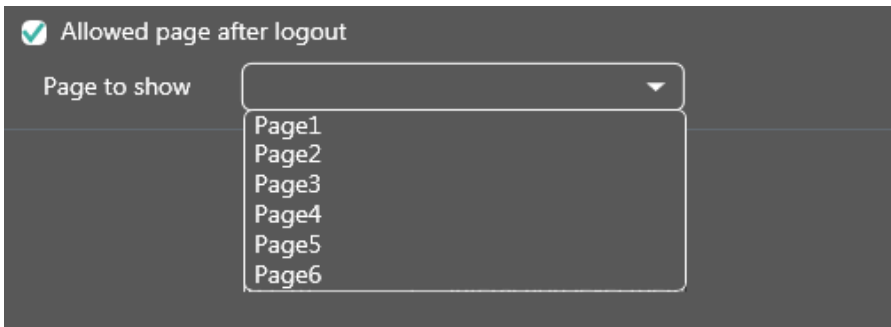
## Security



The “Security” option makes it possible to:

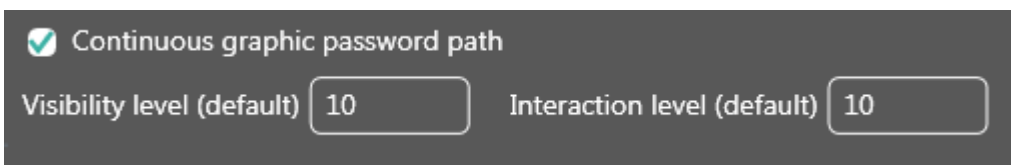
1 Enable or disable automatic “logout” from the project on the panel when said panel is inactive (user-settable inactivity time). Plus, it is possible to establish what page to view after “Logout”.

# CREW Manual



2 Establish whether the user needs to Login to access the project and whether the user name is requested during access to the project (Login).

3 Be able to select the authorisation level for viewing and interaction, in reference to the graphic objects at start up.



The default level (10) only allows the user to see and/or change the non-password protected objects. With level 1 you have the maximum freedom of use.

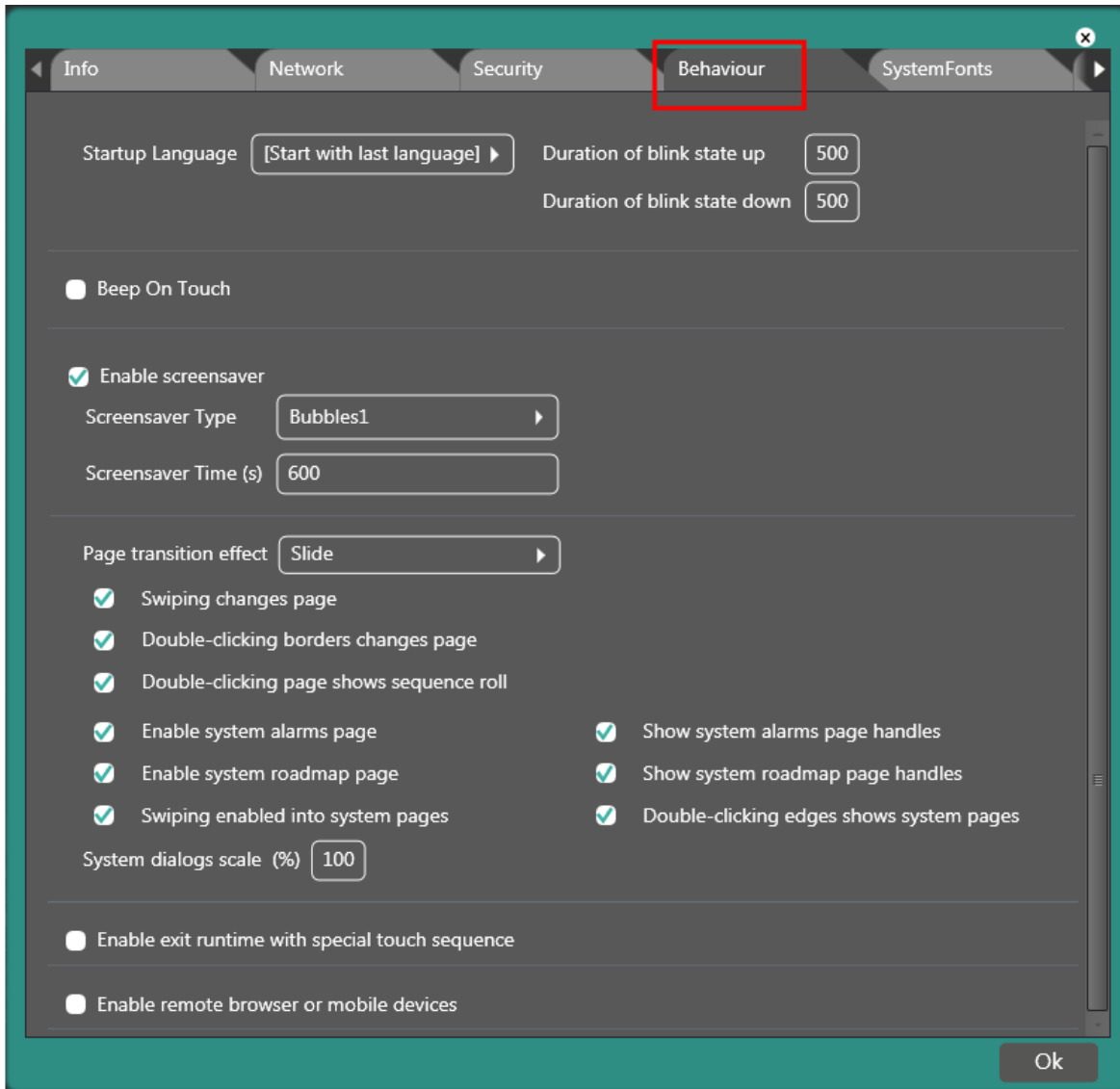
4 Enable, or not, the creation of Log files, namely of a file with user Log registrations.

5 Enable, or not, the possibility of deleting Log file creation (the user can decide after how many days the Log file needs to be deleted).



# CREW Manual

## Behaviour

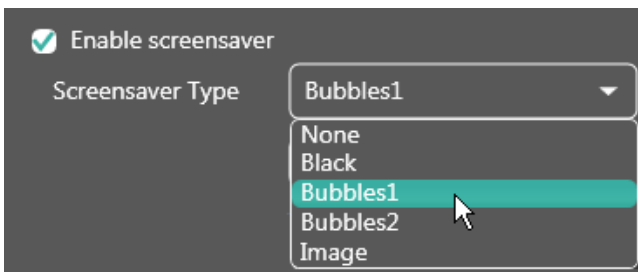




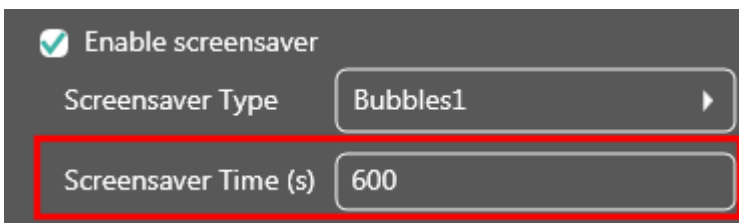
# CREW Manual

The “Behaviour” option makes it possible to:

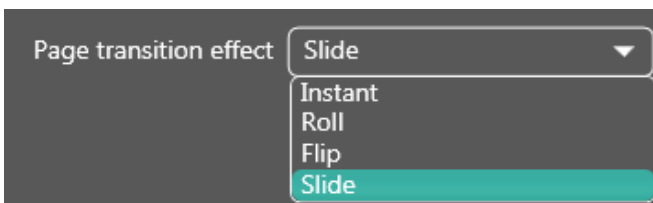
- 1 Select the project language at start-up.
- 2 Set the duration of flashing for the objects contained in the project (value expressed in milliseconds).
- 3 Enable the "Beep", or not, when the objects contained in the project pages are pressed.
- 4 Enable the Screen Saver, or not, choosing from the selection.



It is possible to set after how much inactivity time (expressed in seconds) the screen saver starts:

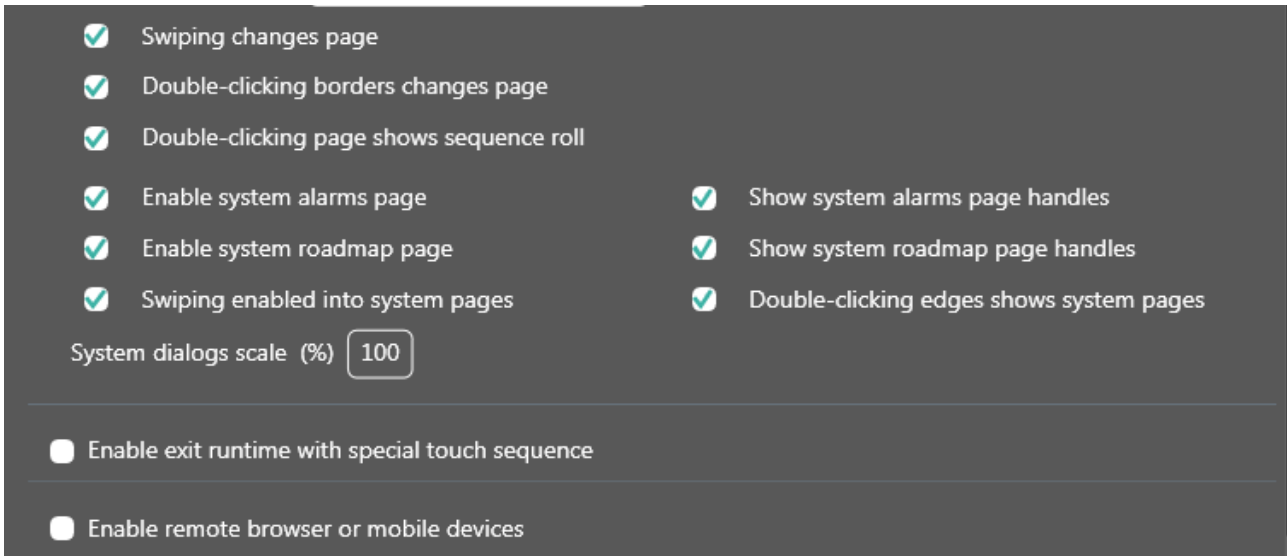


- 5 Choose the "Type of page transition effect" for Runtime by choosing from the selection.



# CREW Manual

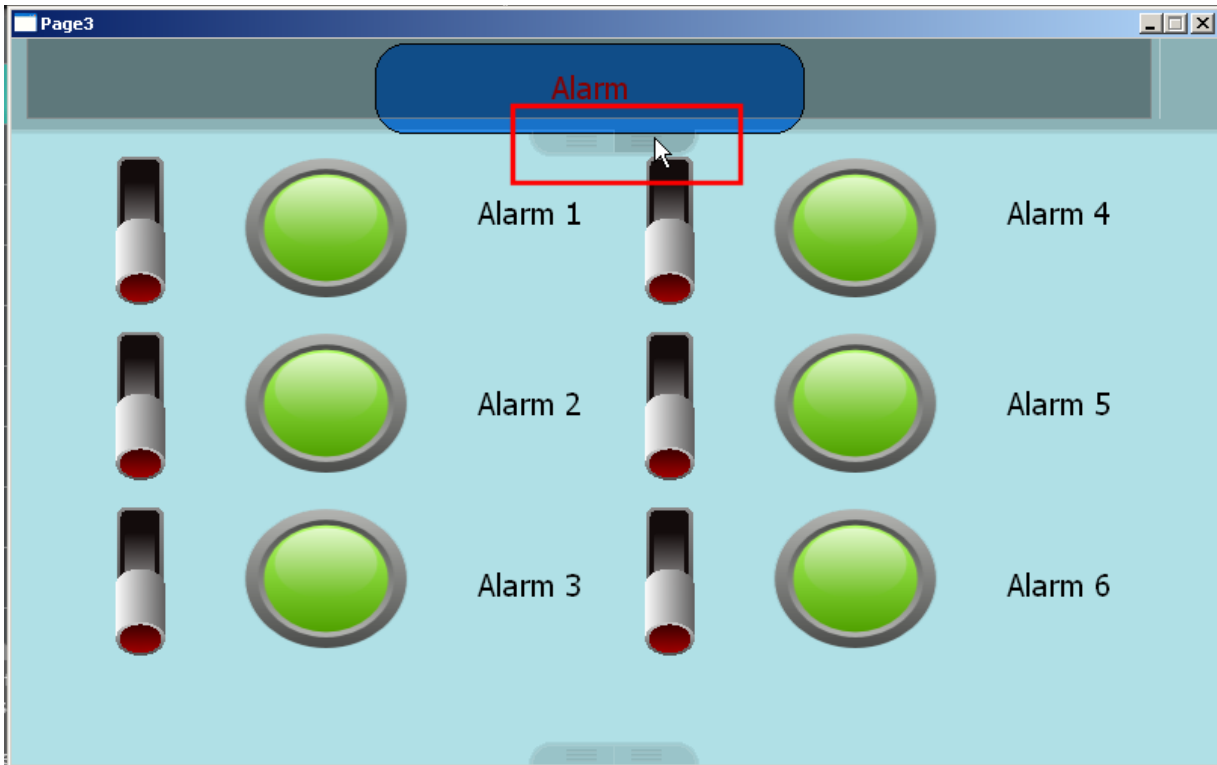
There is also a series of options that can be selected, allowing the user to customise some of the Runtime features.



- Enable Swipe, or not, to change pages.
- Enable changing pages, or not, by double clicking on the edge of the page.
- Enable, or not, viewing the sequence of pages contained in the project by double clicking the centre of the page.
- Enable, or not, viewing the system alarms page, the alarm banner and the “handles” to open them.

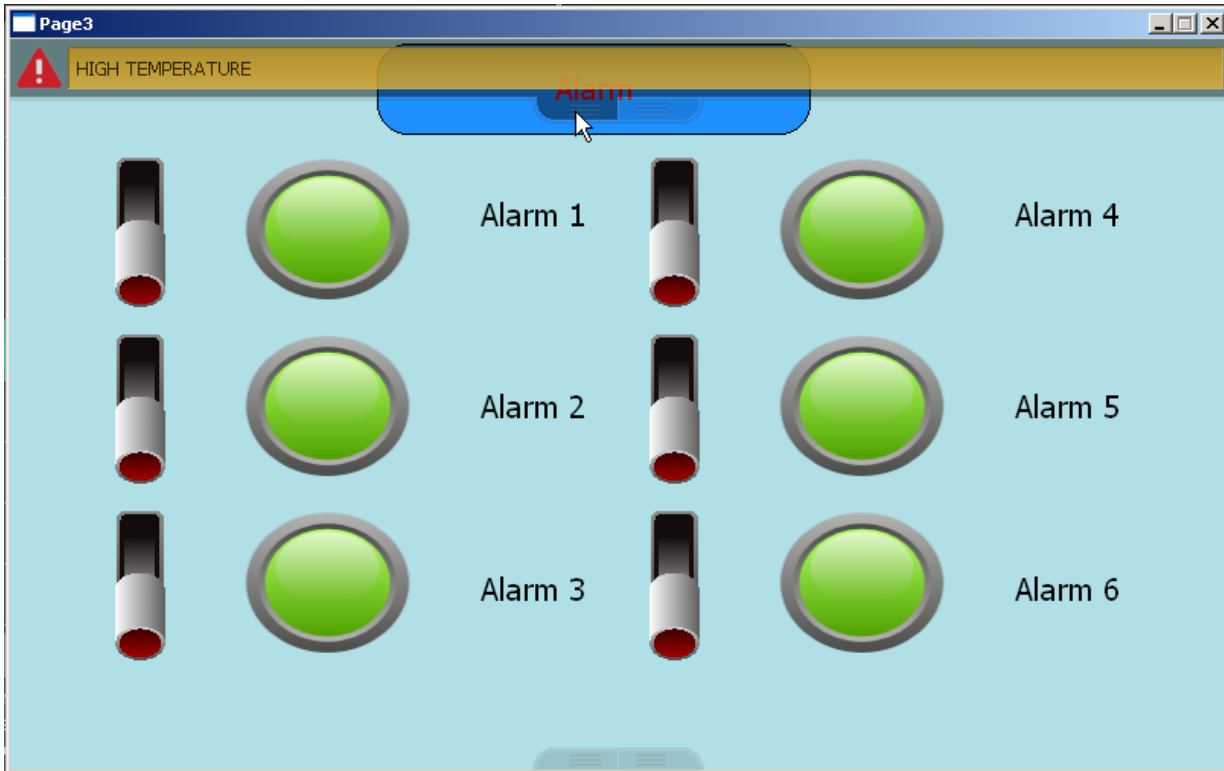
# CREW Manual

## System alarms



# CREW Manual

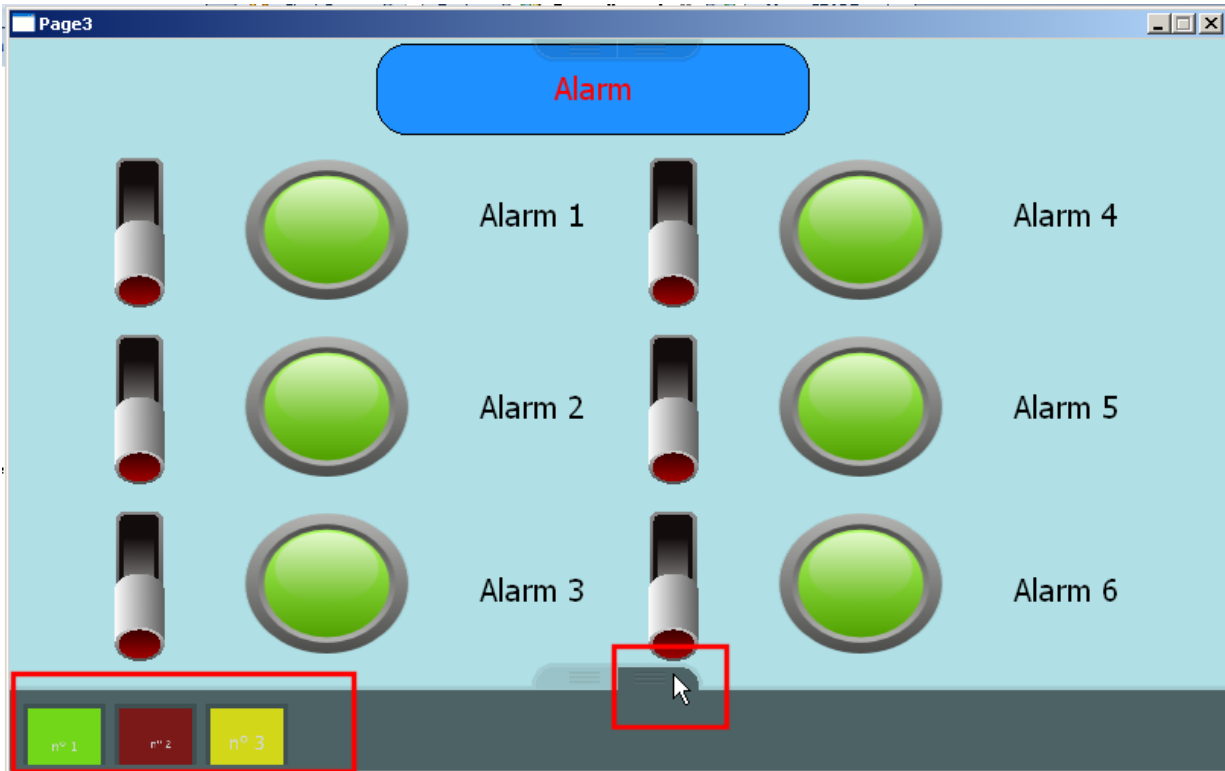
## Alarm banner



- Enable, or not, viewing the system pages (“full screen” pages and “popup” pages) and the “handles” to open them.

# CREW Manual

## Pop-up



# CREW Manual

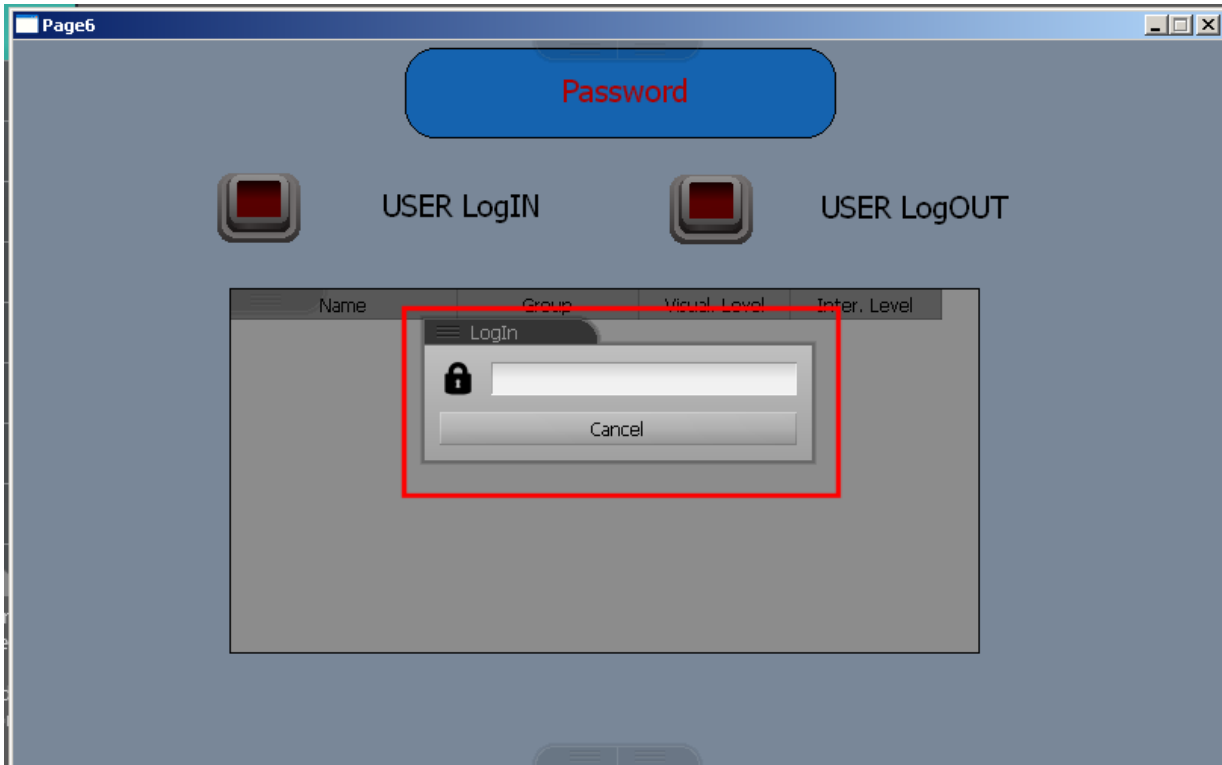
Full screen



- Enable, or not, the Swipe function on the system and active alarm pages.
- Enable, or not, viewing the system and active alarm pages by double clicking the sides of them.
- Enable, or not, the possibility of re-sizing (by setting the percentage values) the dialog boxes that appear in Runtime.

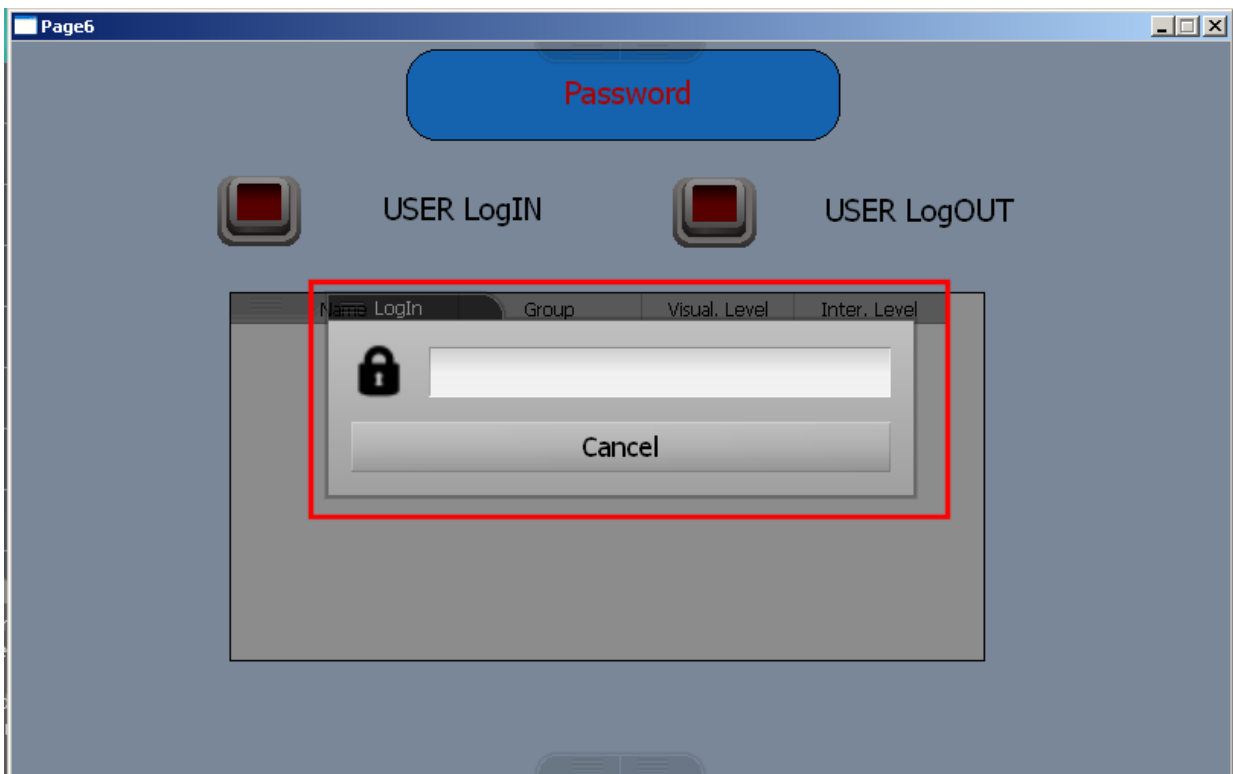
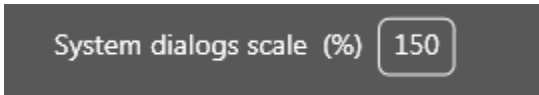
# CREW Manual

100%

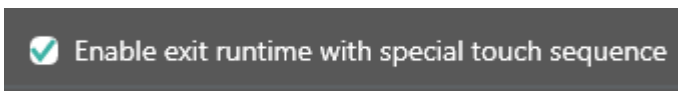


# CREW Manual

150%

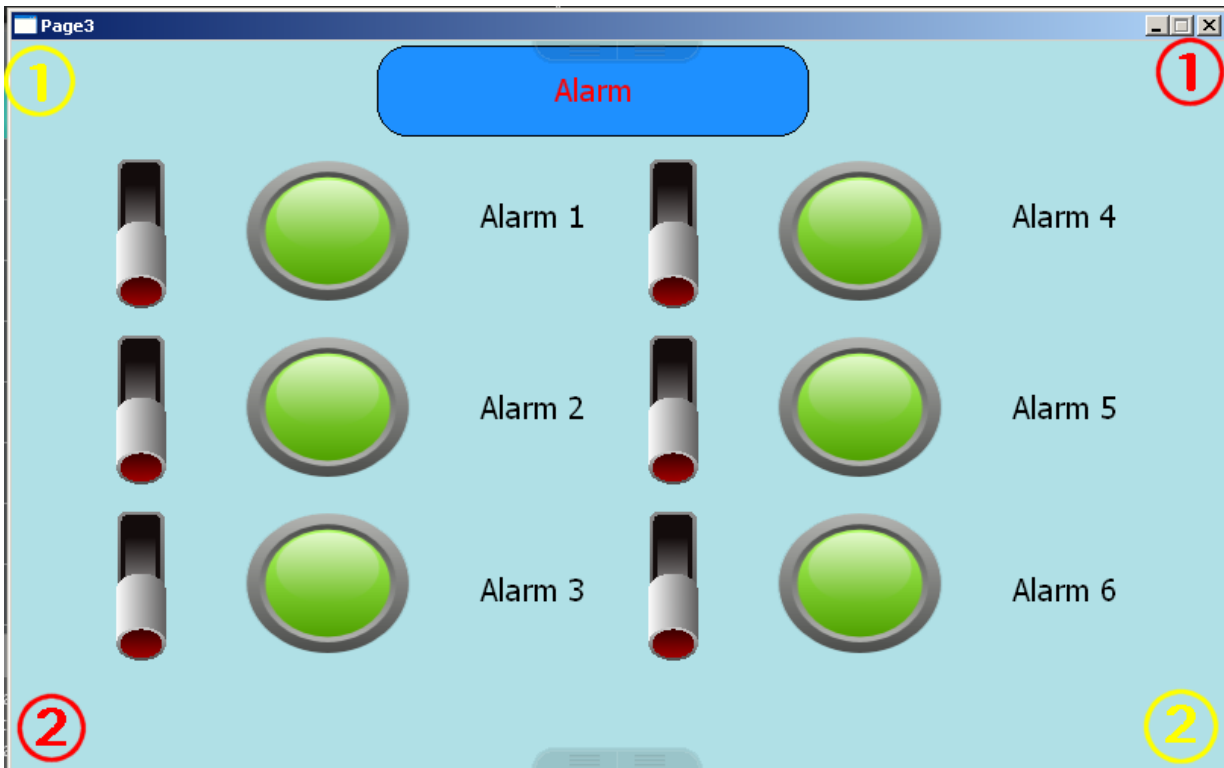


- Enable, or not, exiting Runtime by pressing the two corners on the top right and on the bottom left (or top left and bottom right).



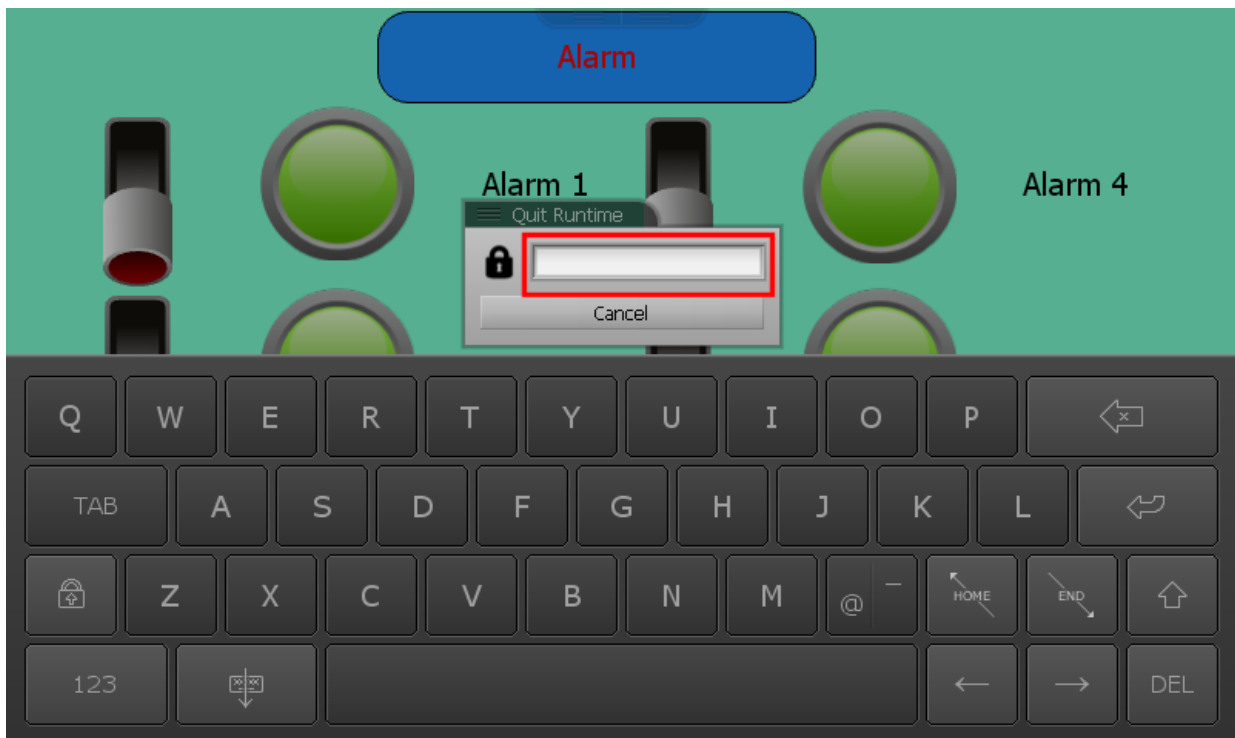


# CREW Manual

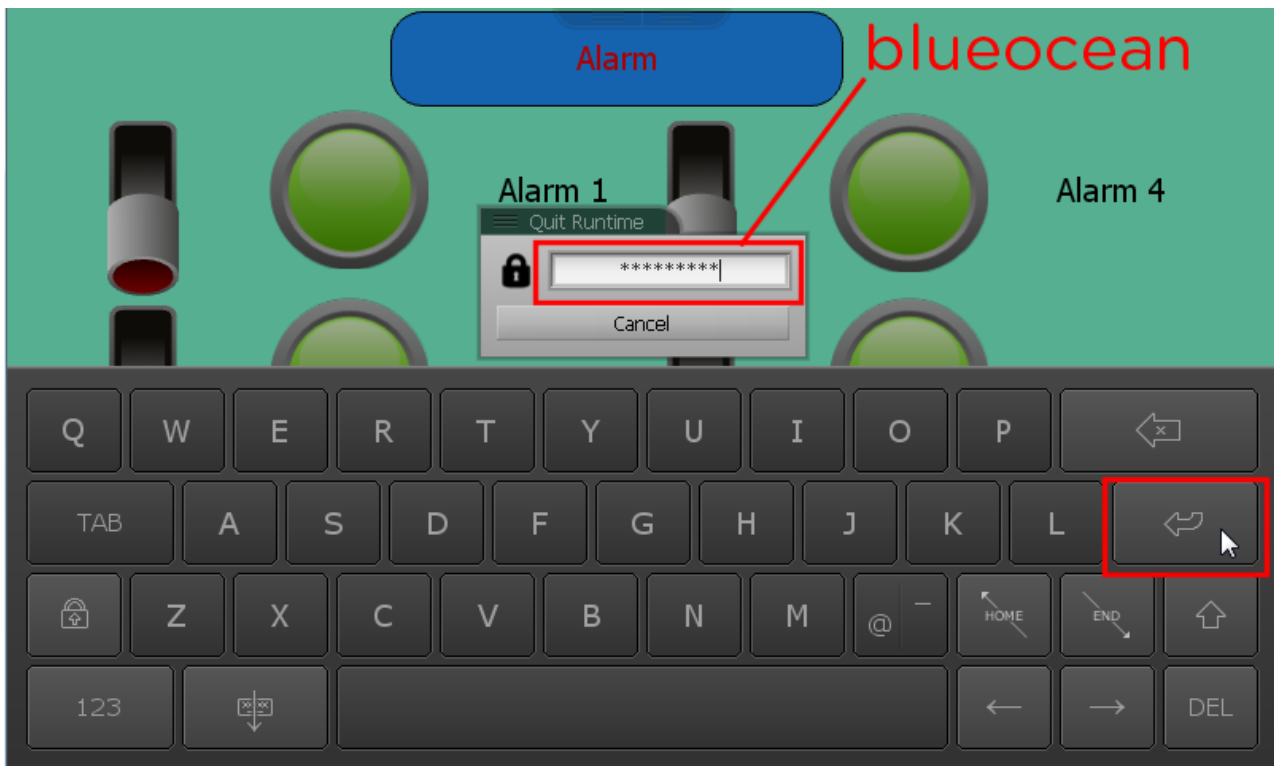


# CREW Manual

When you have pressed the opposite corners, the following image will appear, where you need to enter the default password (which is blueocean) to exit Runtime.



# CREW Manual



# CREW Manual

At the end, the following service page will appear.



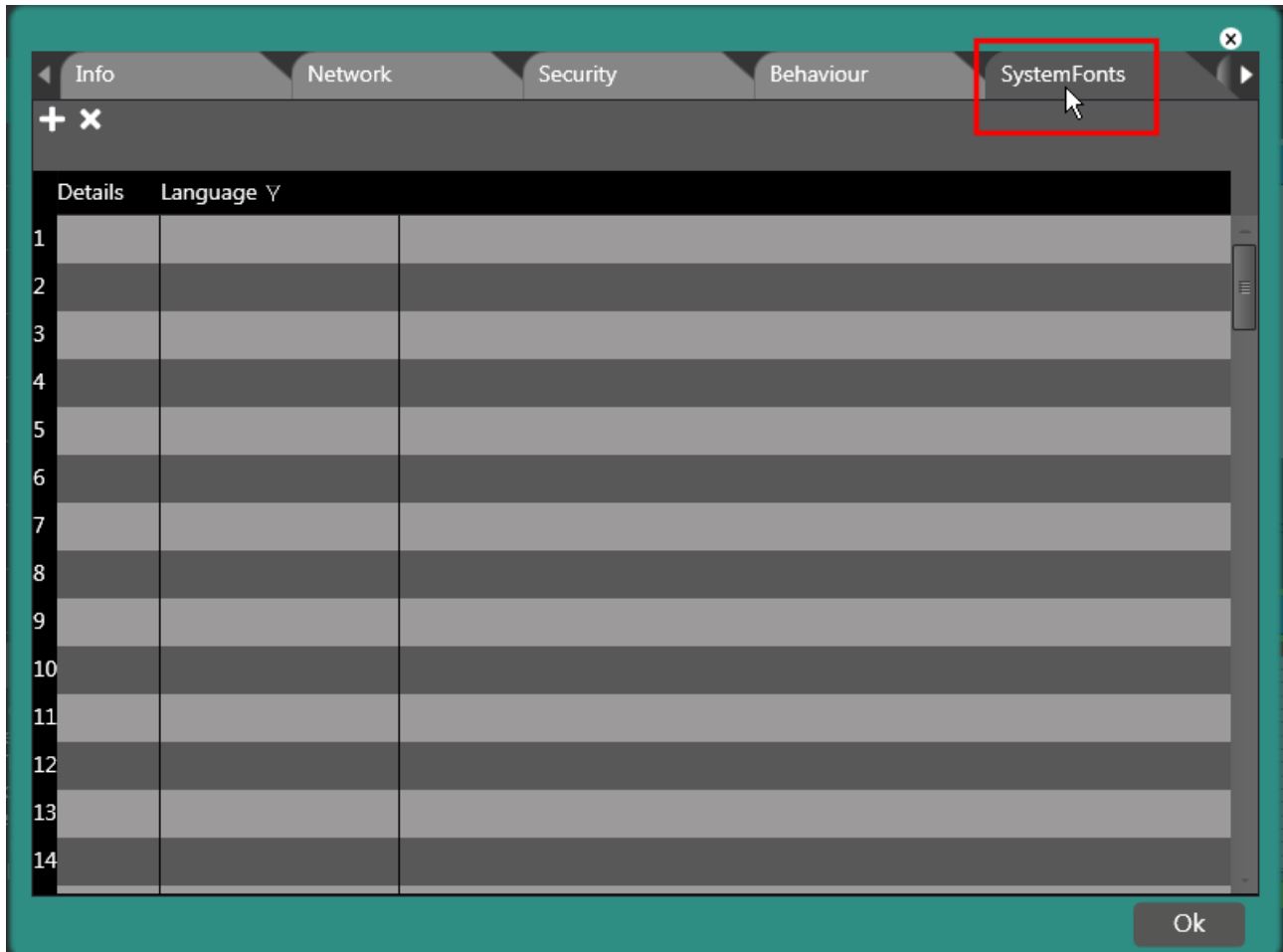
- Enable or not, the possibility of accessing the panel from the remote mobile device.

Enable remote browser or mobile devices

Port

# CREW Manual

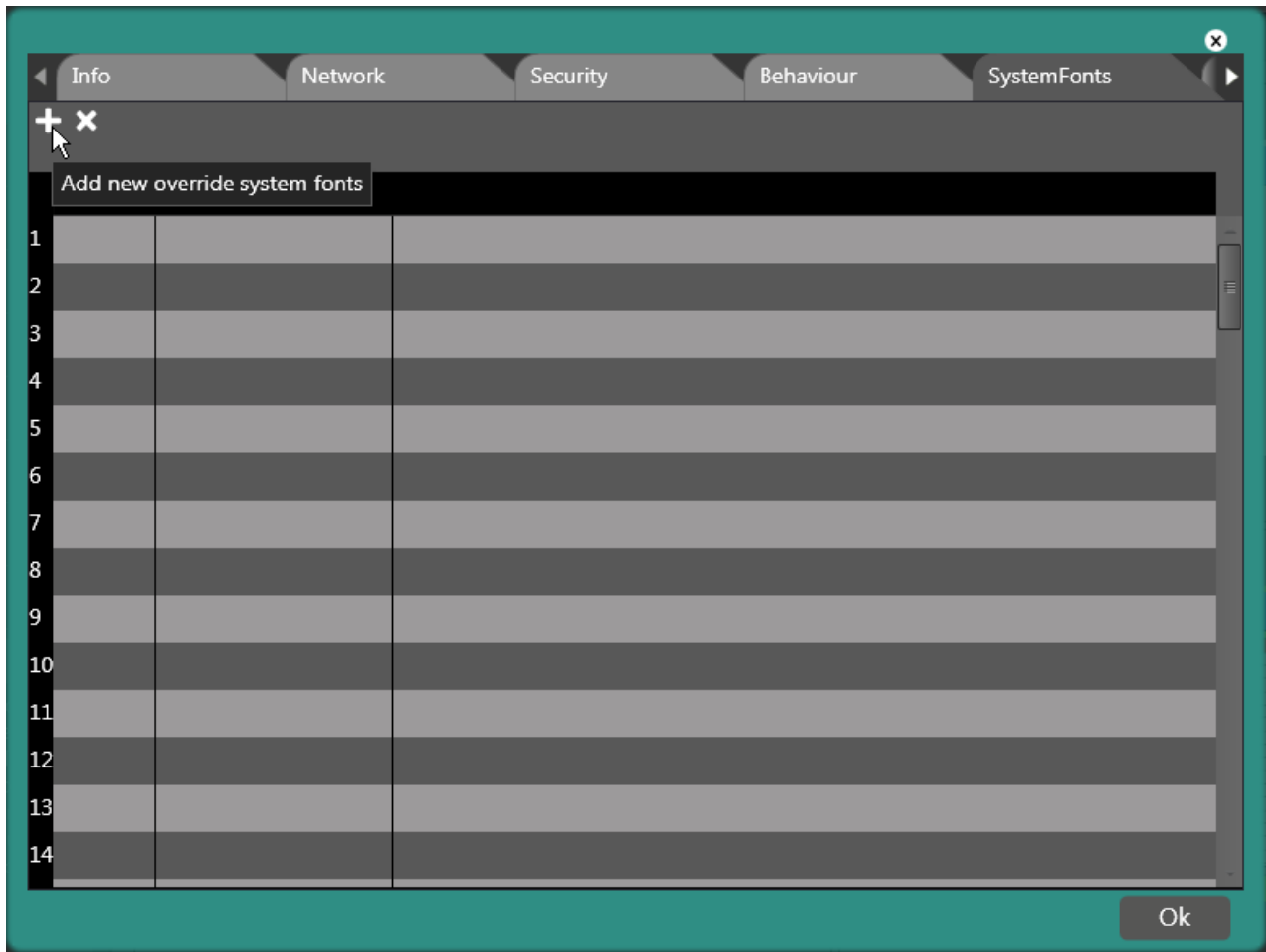
## System Font



The “System Font” option makes it possible to add new fonts in the project, for two types:

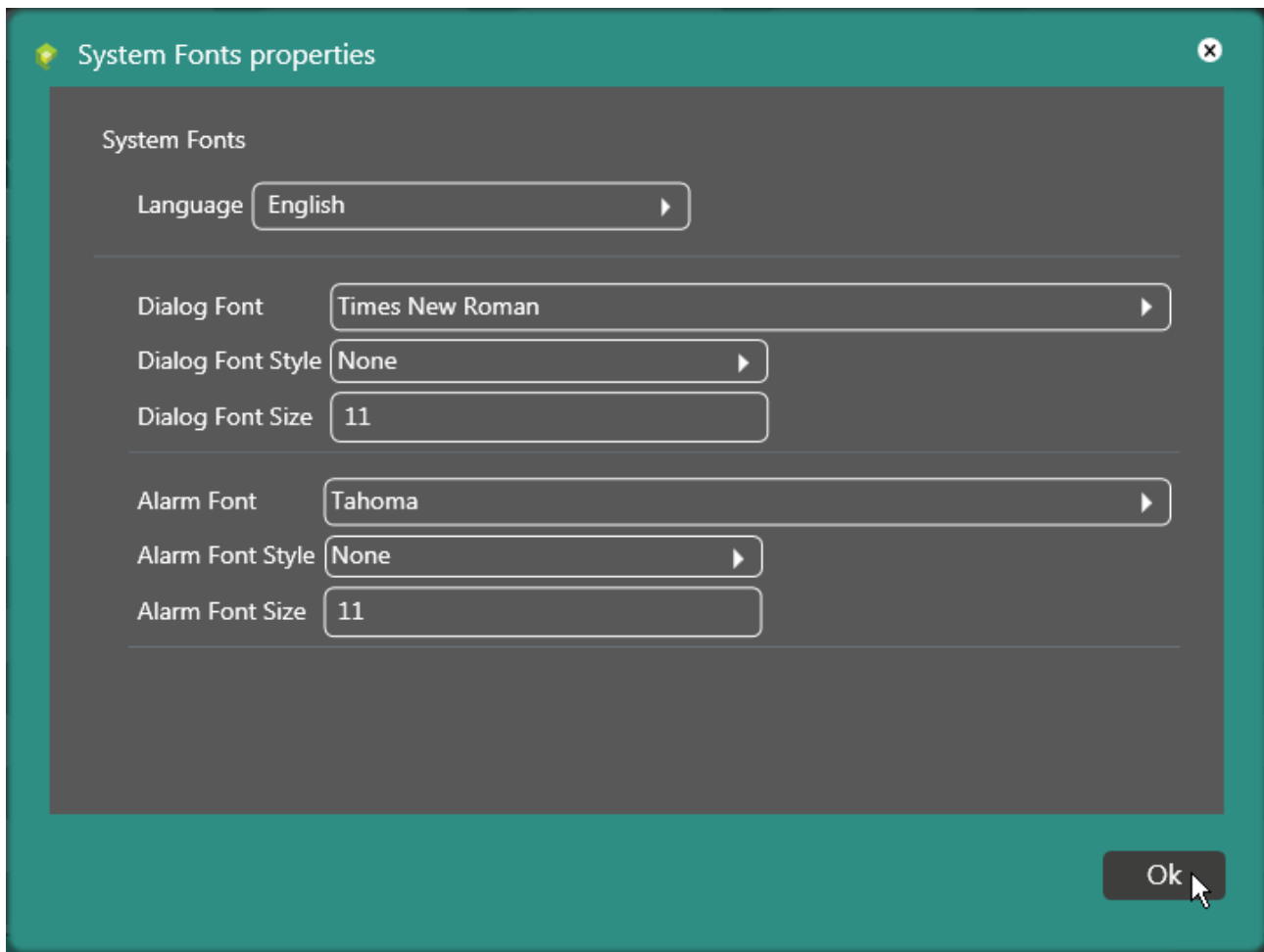
- 1 Dialog font
- 2 Alarm Font

# CREW Manual



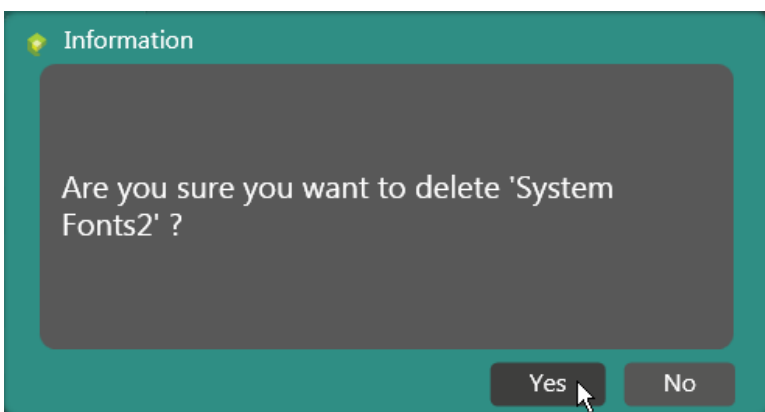
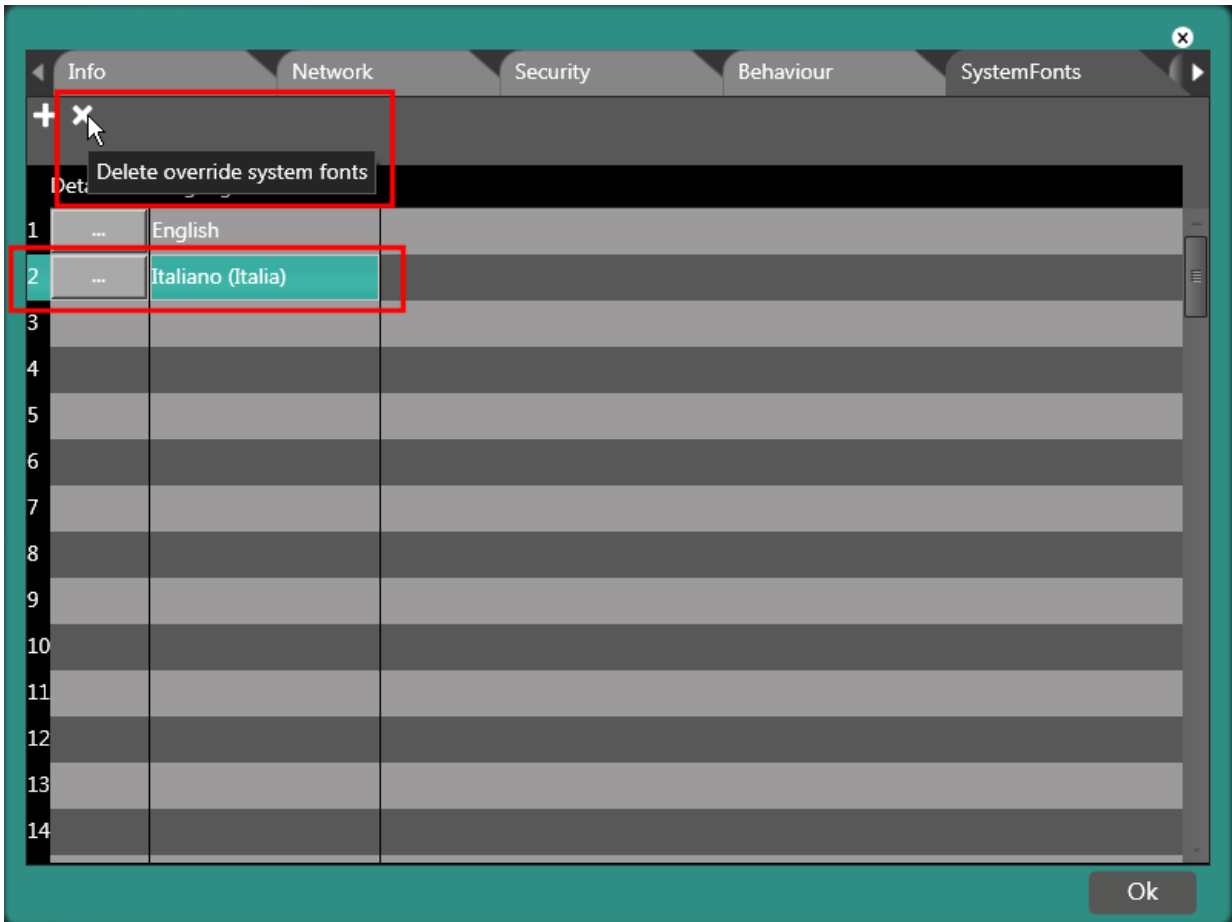
It is also possible to choose the language that the new font is used for, to select the type of font, to choose the style (none, bold, italic or underlined) and size.

# CREW Manual



# CREW Manual

It is possible to eliminate the selected fonts.

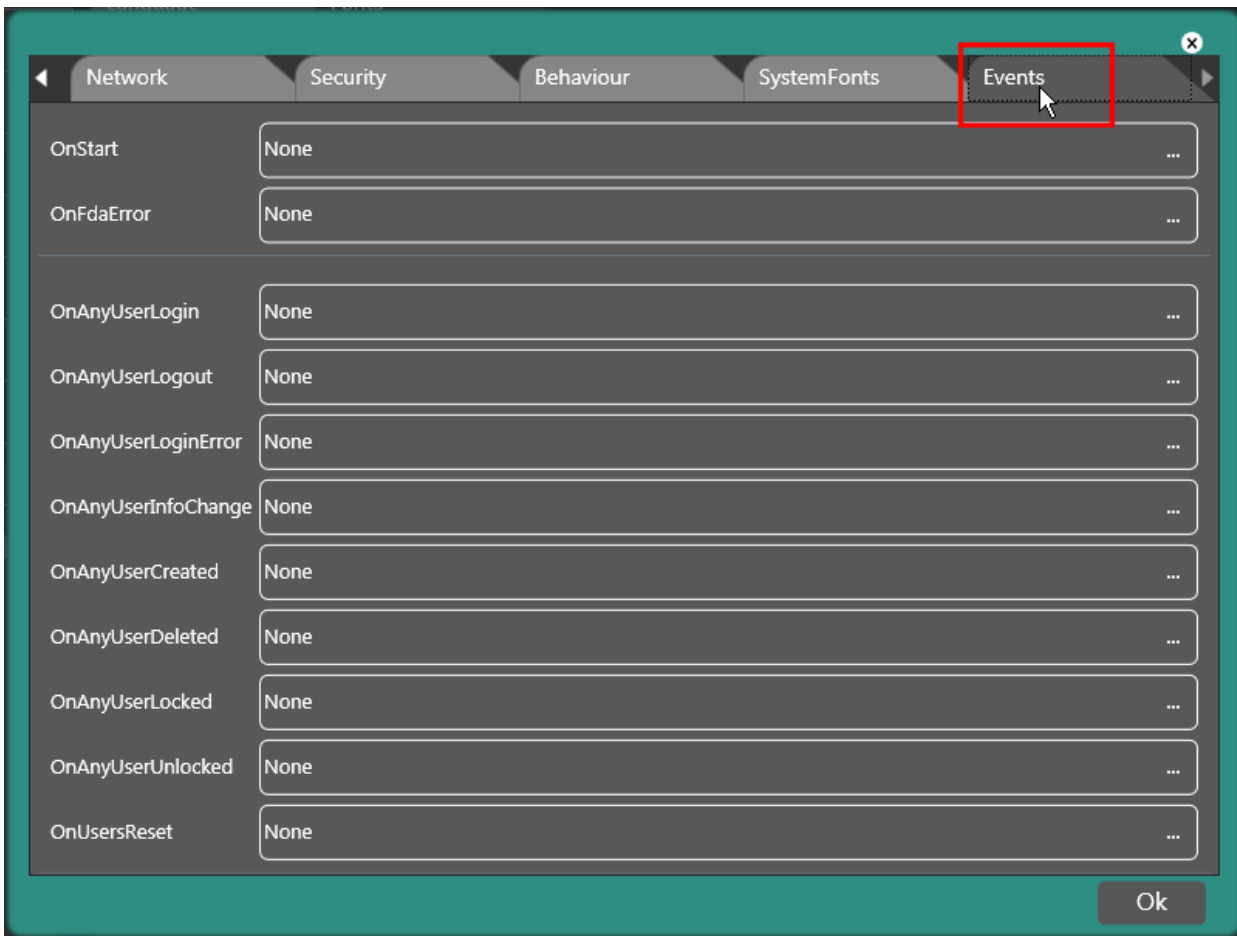




# CREW Manual

## Events

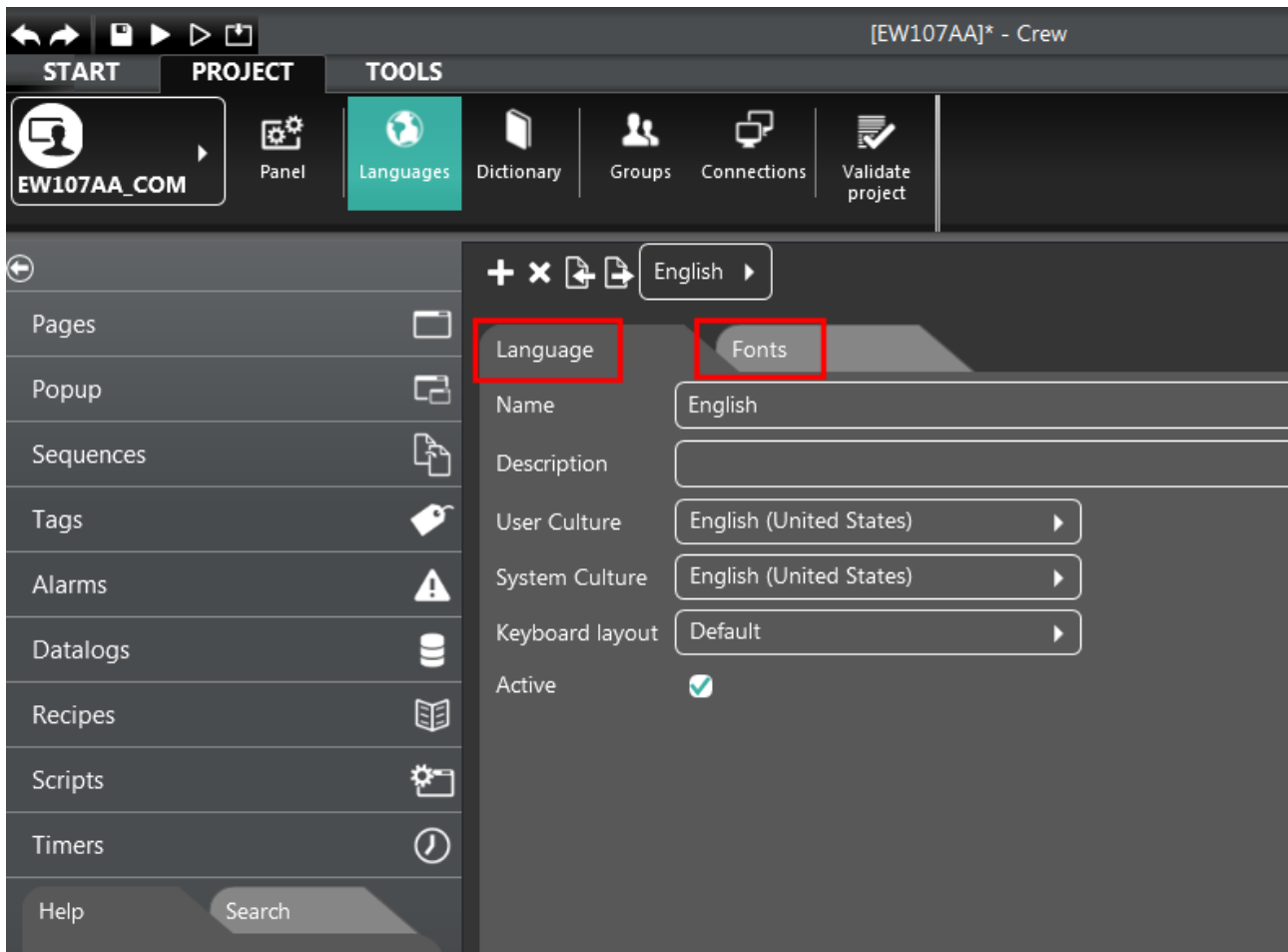
With the “Events” option the user can associate functions to the project conditions listed in the image.



To read how to associate the functions, refer to the "[Events](#)" section.

# CREW Manual

## Languages



The “Languages” menu includes these options:

- Language
- Fonts

# CREW Manual

## Language



In the “Language” menu it is possible to:

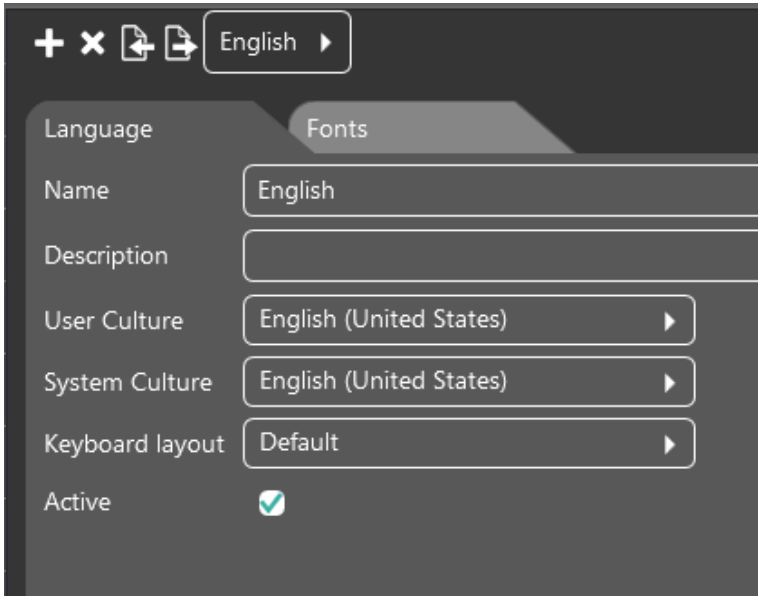
- Create new languages.
- Delete an existing language.
- Import languages.
- Export languages.

Plus, it is also possible to:

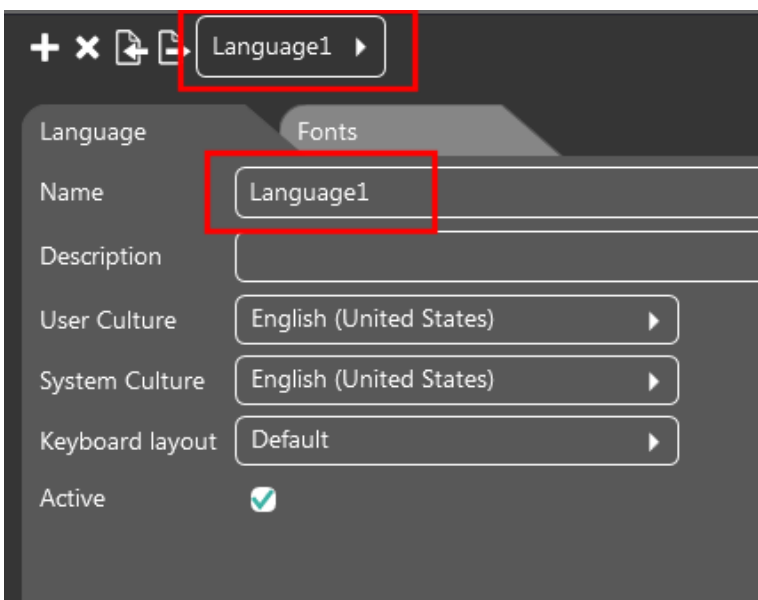
- Enter the name and description associated to the language.
- Select the “User Language”, namely the language associated to the project.
- Select the “System language”, namely the language associated to system messages (to the "dialog boxes" for example).
- Choose the type of keyboard (normal or Cyrillic) for data entry.
- Establish whether to make the selected language active or not.

# CREW Manual

## Creating new Languages



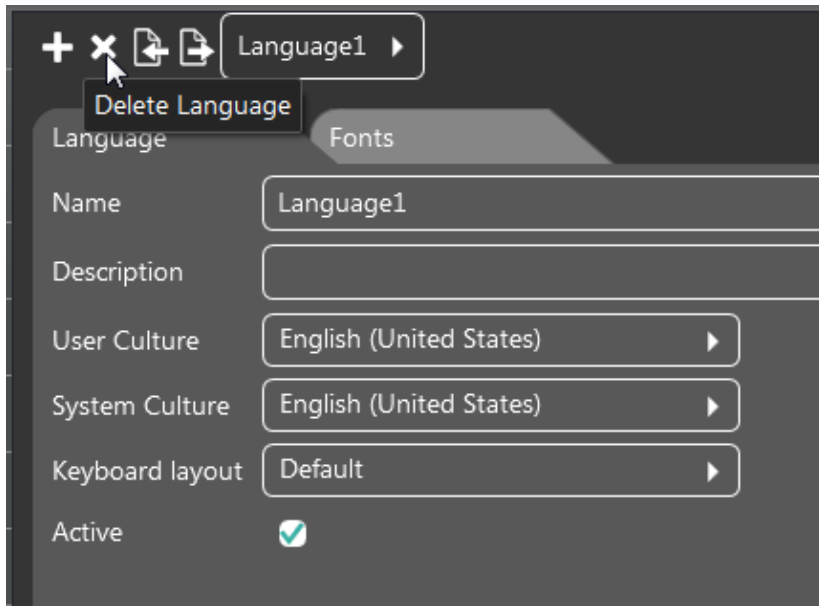
Click the “Create new language” key to open the creation page shown in the image below.



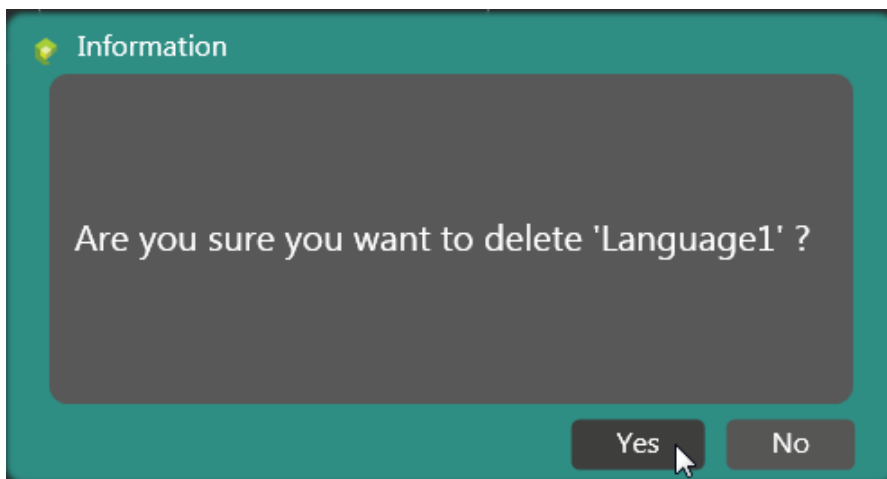
# CREW Manual

## Deleting an existing language

After selecting the language to be deleted, click the “Delete language” key.



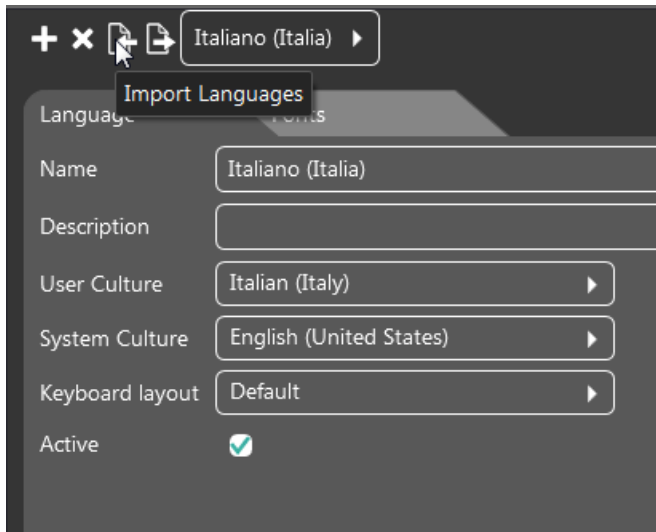
Then confirm the operation.



# CREW Manual

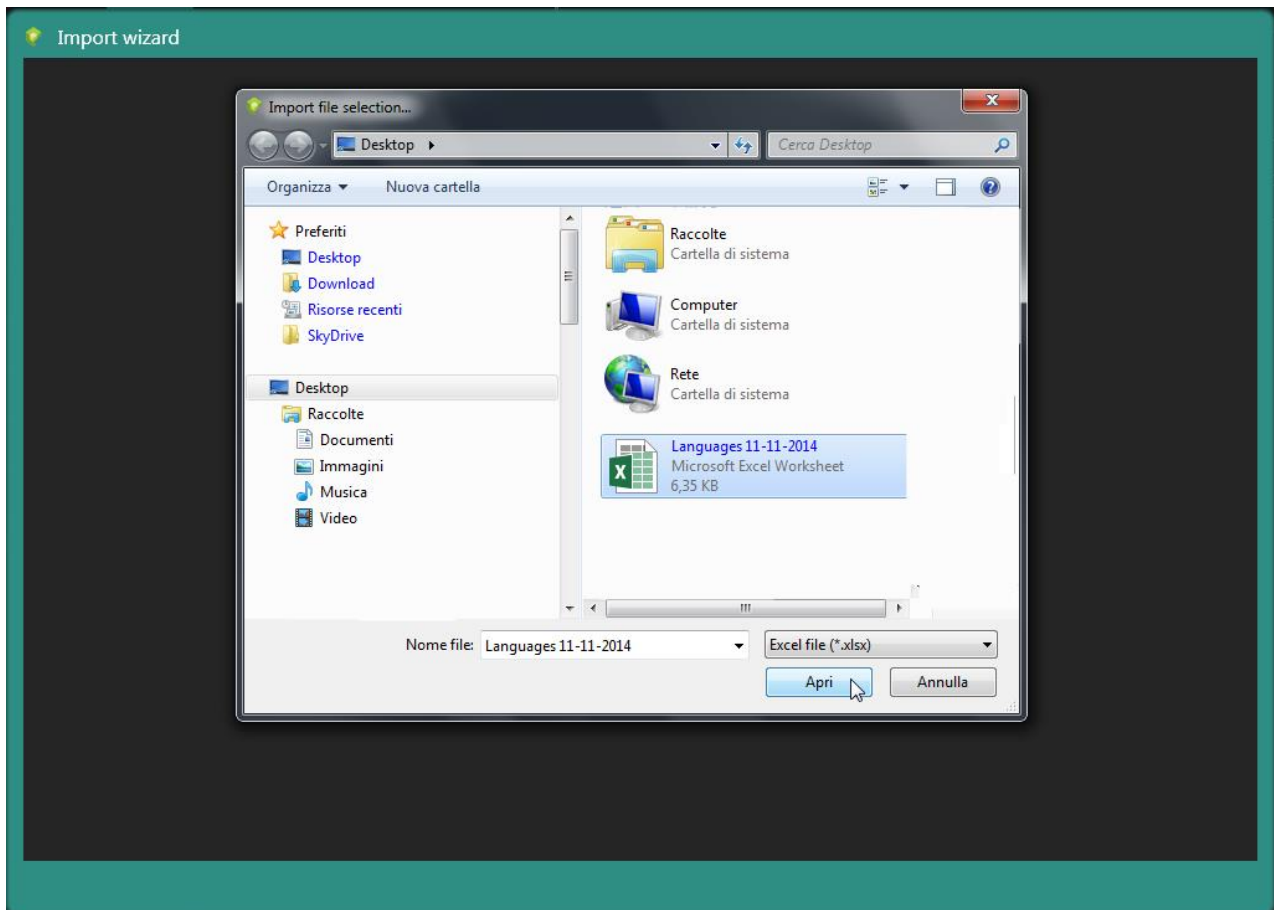
## Importing languages

Press the “Import languages” key to import a previously created language (the file needs to be in Excel format).



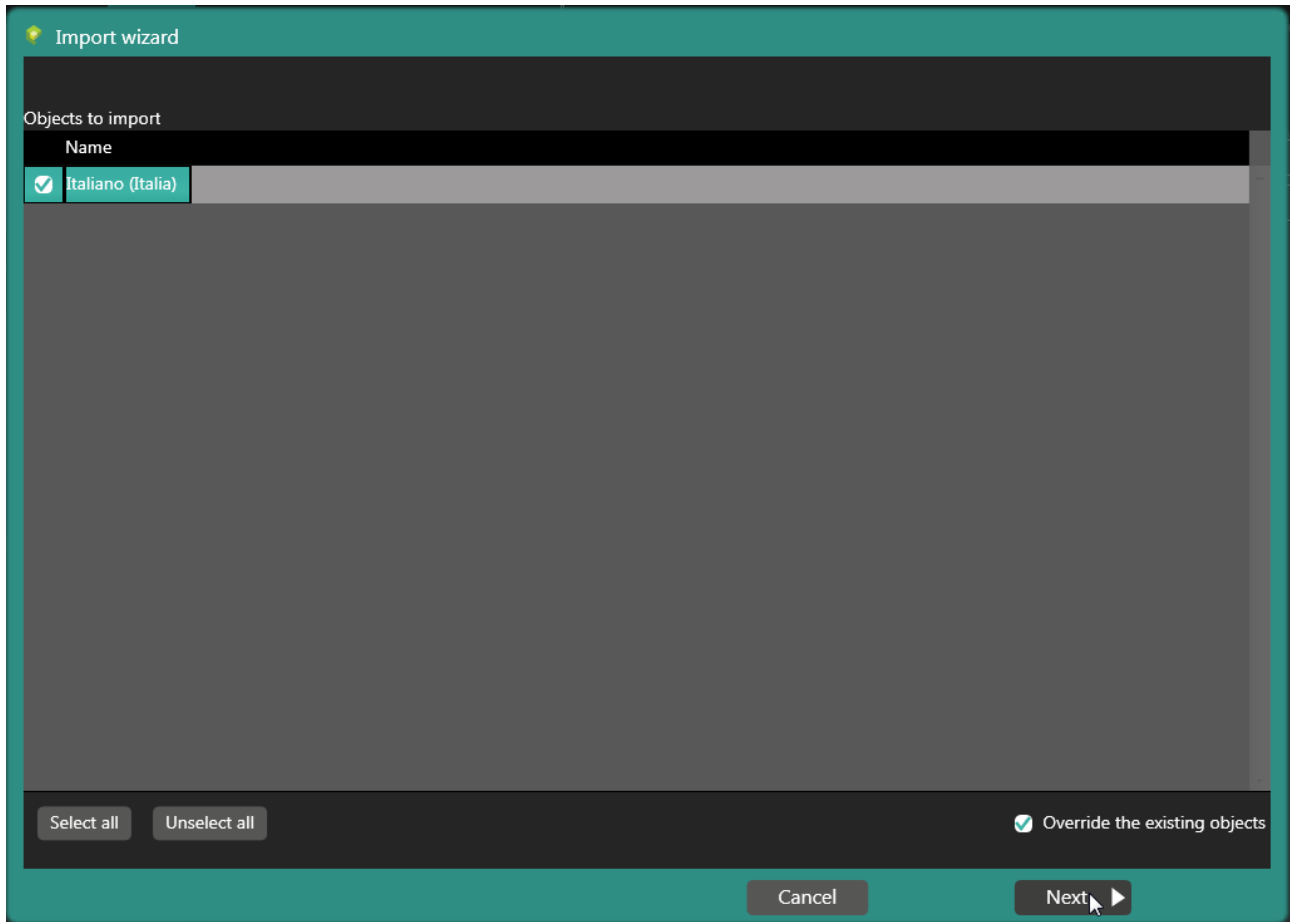
# CREW Manual

Select the file for import and click “Open”.



# CREW Manual

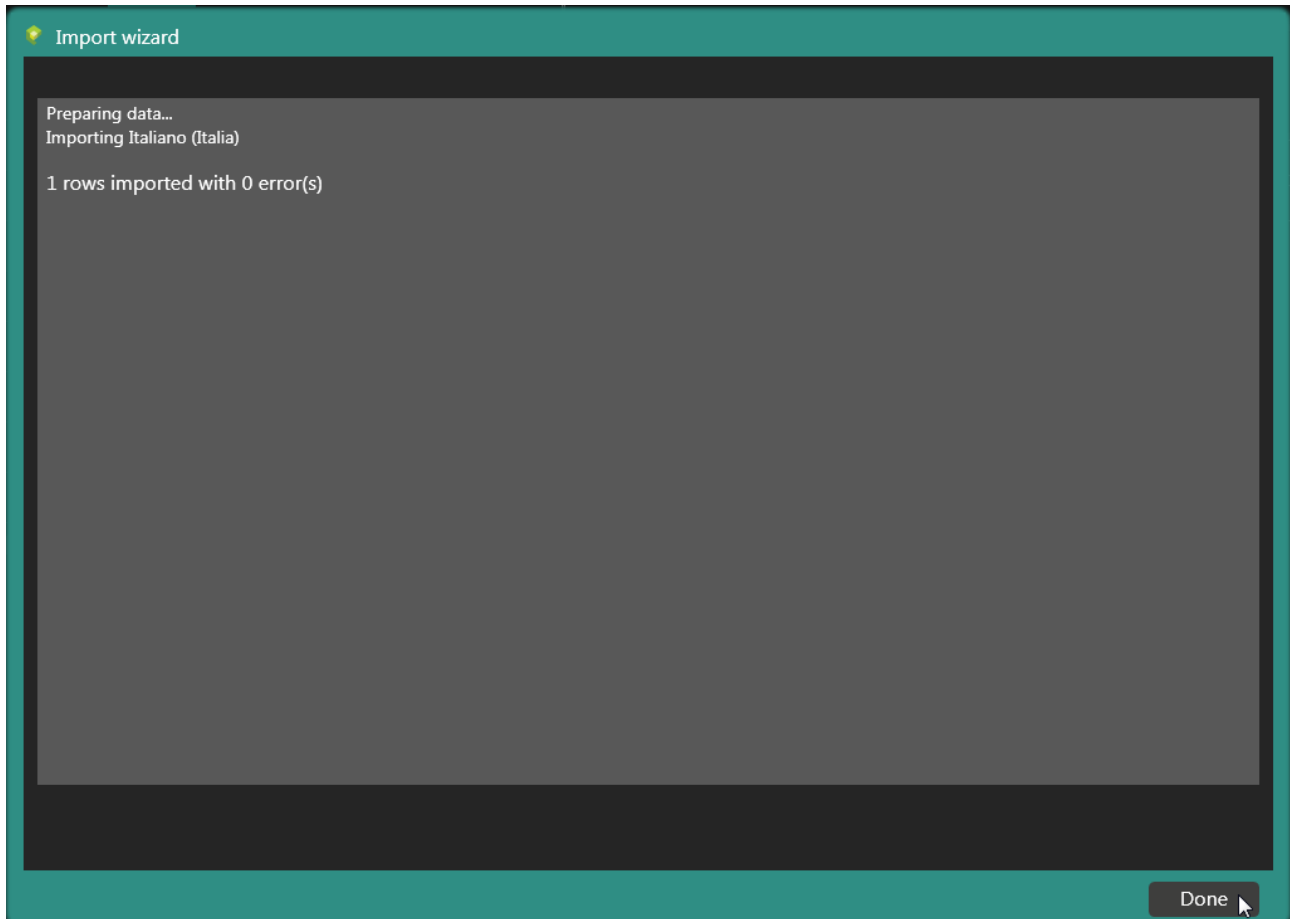
Select the objects for import and decide whether to overwrite the existing objects or not. Then click “Next”.





# CREW Manual

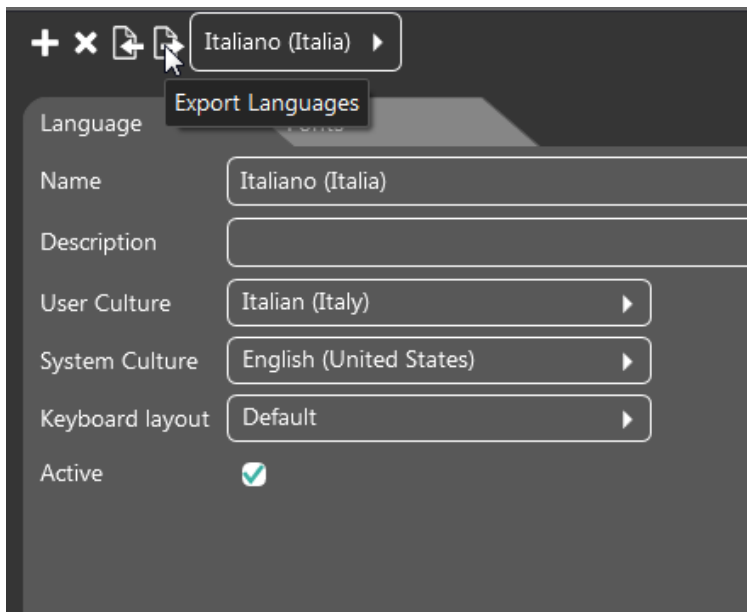
At the end of the operation click “End”.



# CREW Manual

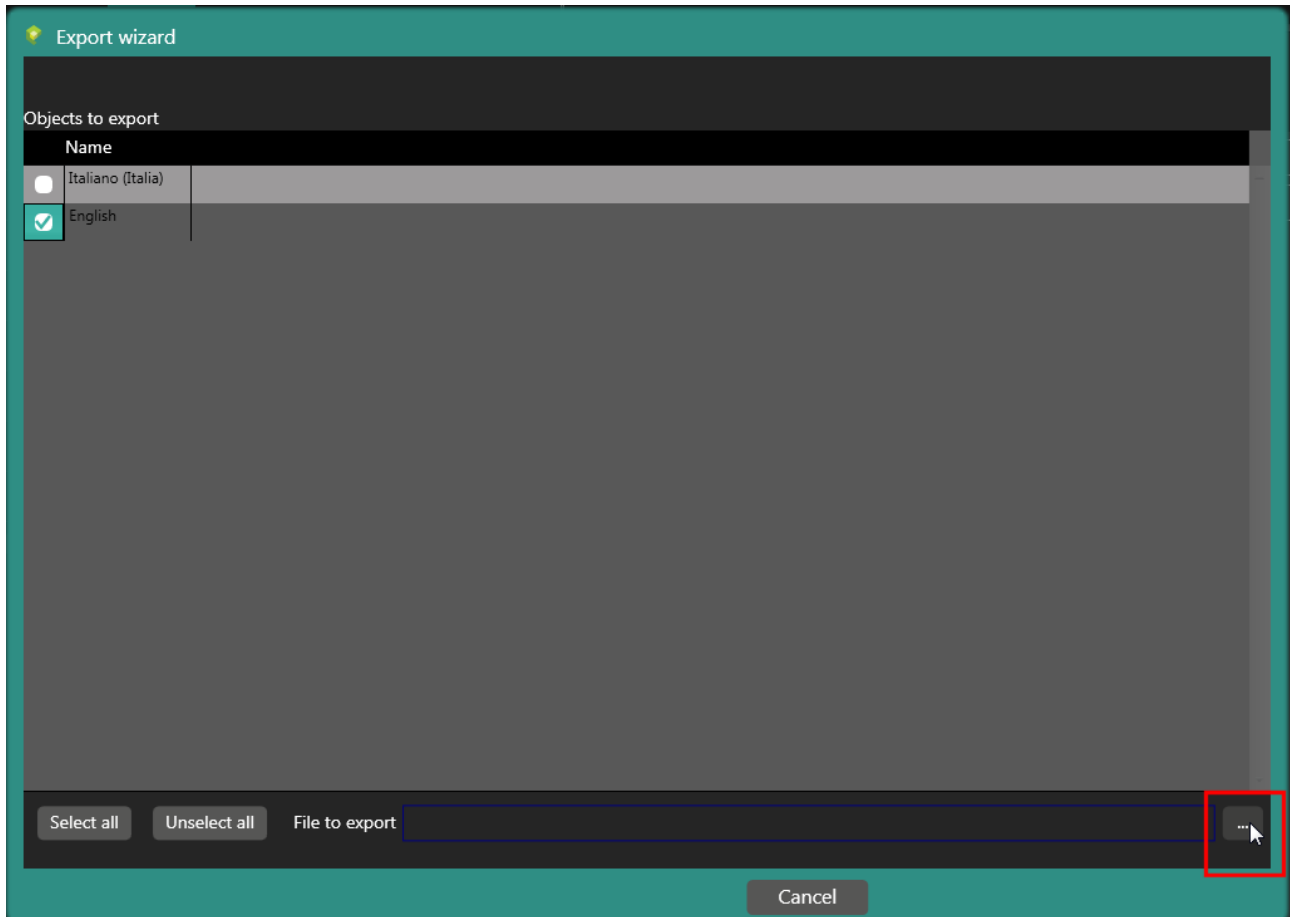
## Exporting languages

Press the “Export languages” key to export languages to an Excel file.



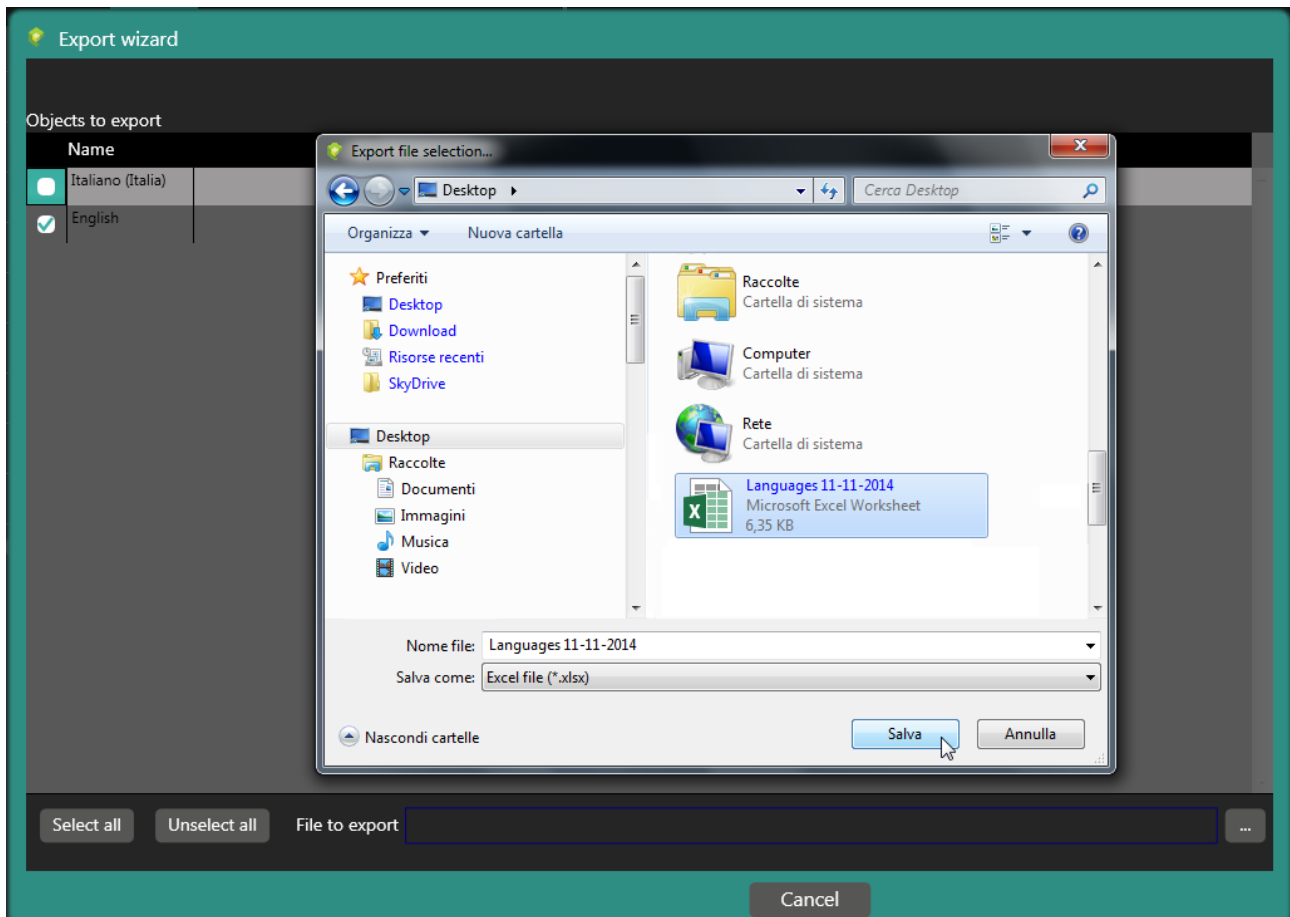
# CREW Manual

Select the file for export and click the key on the bottom right.



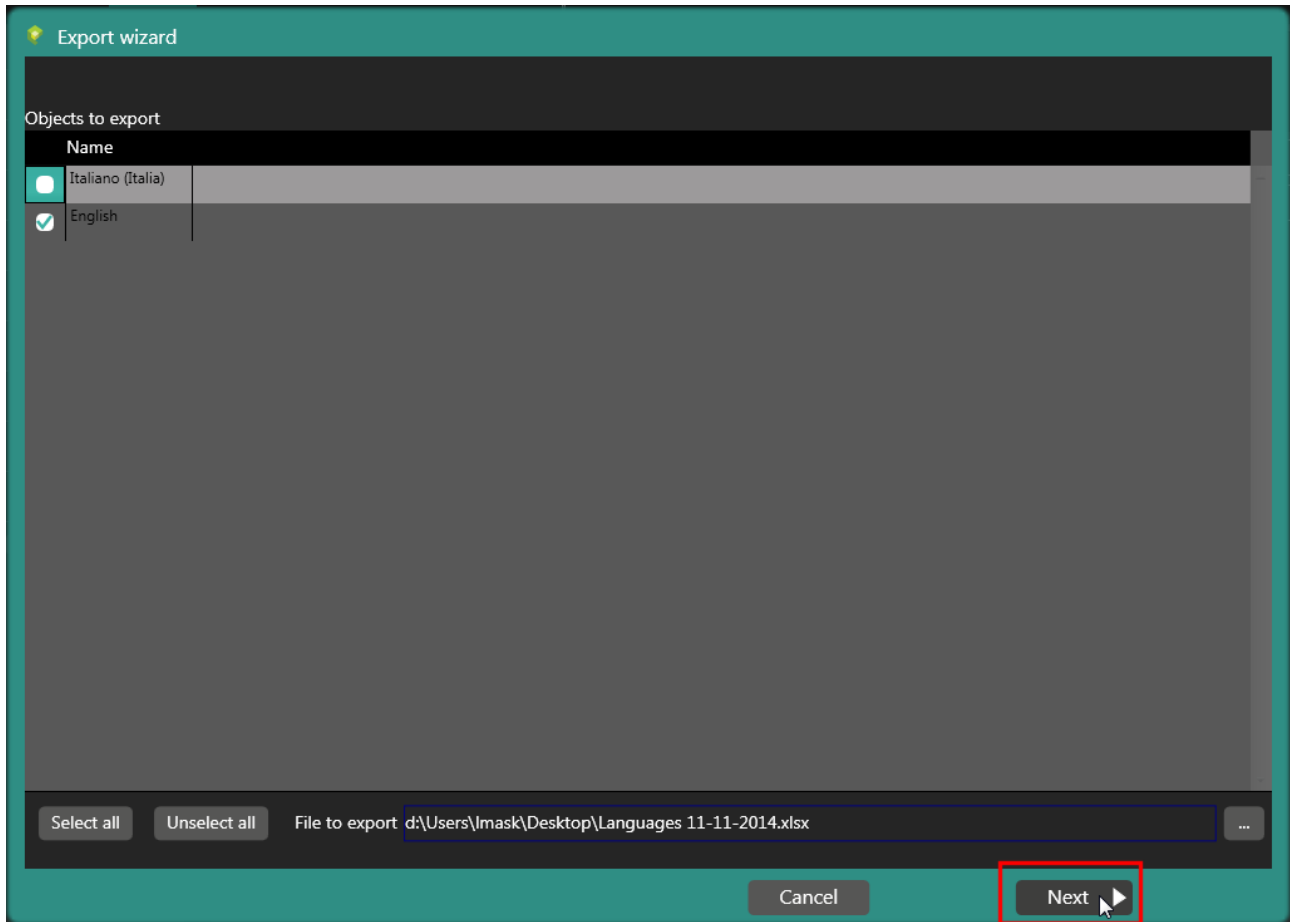
# CREW Manual

Select the required path and click “Save”.



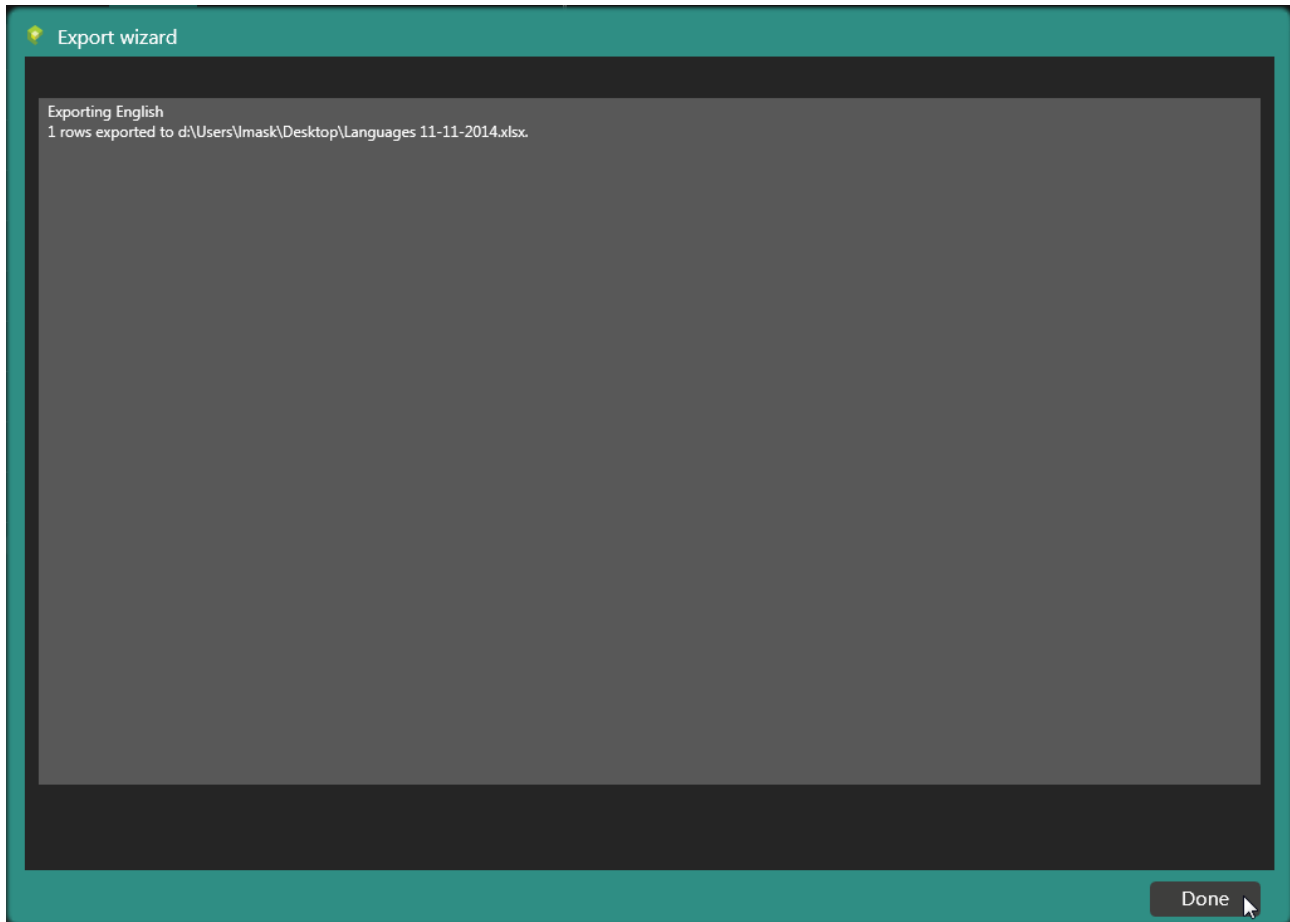
# CREW Manual

Select the objects for export using the relative keys and click “Next”.



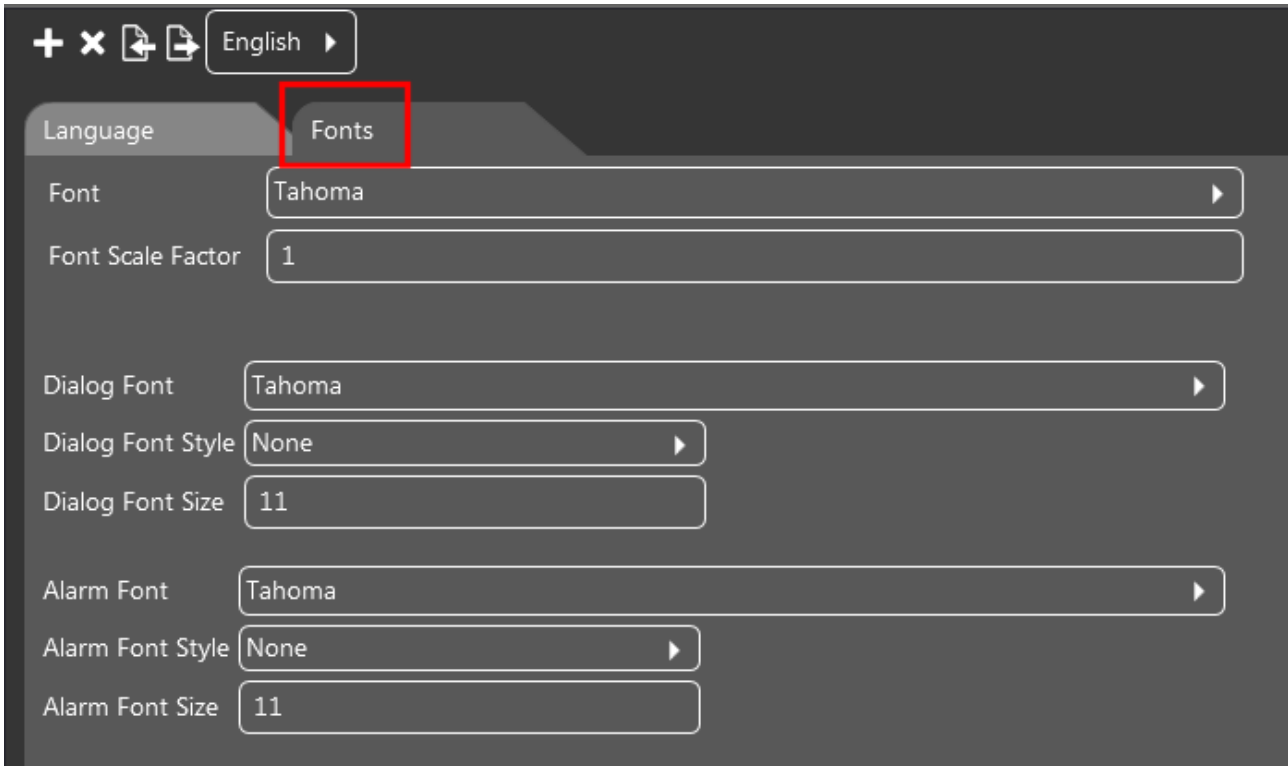
# CREW Manual

At the end of the operation click “End”.



# CREW Manual

## Fonts



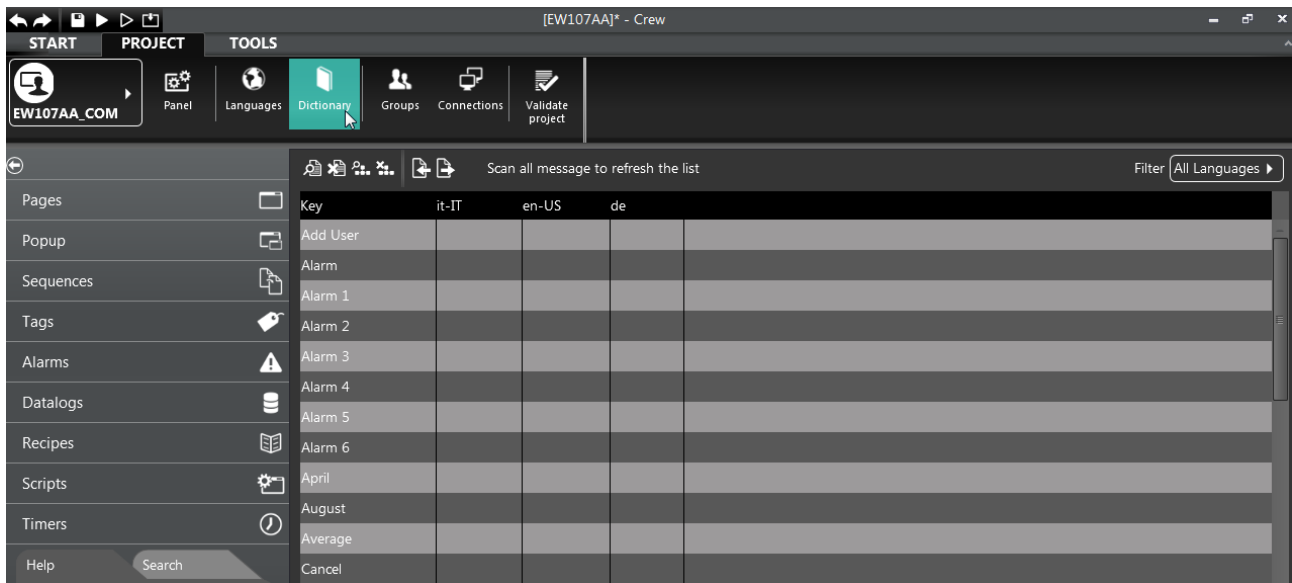
From the “Fonts” menu it is possible to:

- Enter new project fonts and define the scale factor. For example, setting a scale factor of 1 for Italian, whereas for German, which has longer words, chose a lower scale factor, such as 0.5 or 0.75.
- Define the type of dialog font (fonts associated to system messages, to "dialog boxes" for example).
- Define the type of fonts associated to the alarms.

To each font it is possible to assign a style and size.

# CREW Manual

## Dictionary

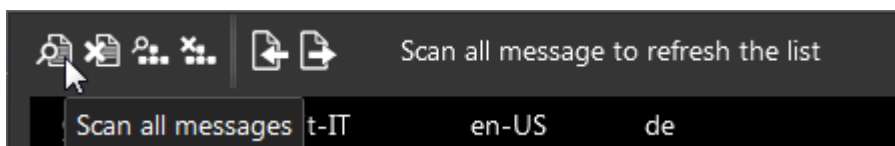


The “Dictionary” menu makes it possible to manage the translations of texts (keys) used in the project. It is possible to manually associate the translations inside Crew or import a list in Excel format.

From the “Dictionary” menu it is possible to:

### Check all messages

Click the relative key to run a check on the last tests (keys) entered in the project. For example, if you enter a new text on a page, this appears in the list after having pressed the “Check all messages” key.

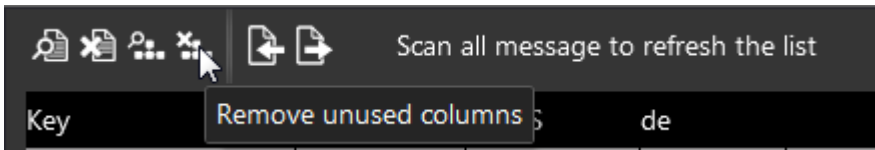




# CREW Manual

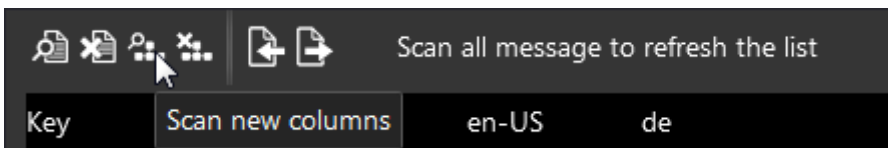
## Remove unused messages

Click the relative key to delete the texts that are no longer used in the project (for example, after deleting an object containing text in a project page).



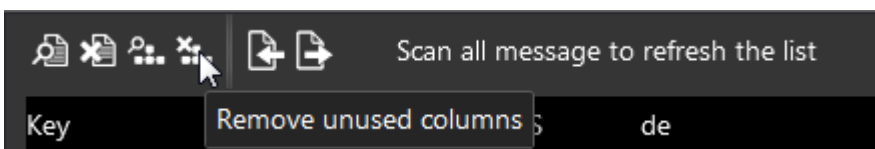
## Check new columns

Click on the relative key to check if new columns have been entered in the dictionary (for example, after adding a new user language).



## Remove unused columns

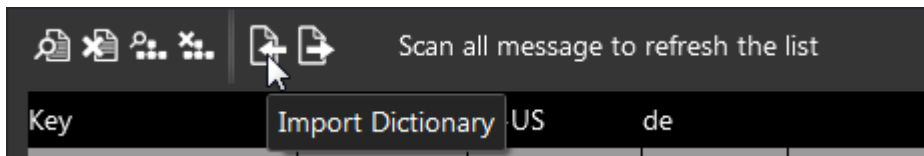
Click on the relative key to remove unused columns (for example, after eliminating a user language).



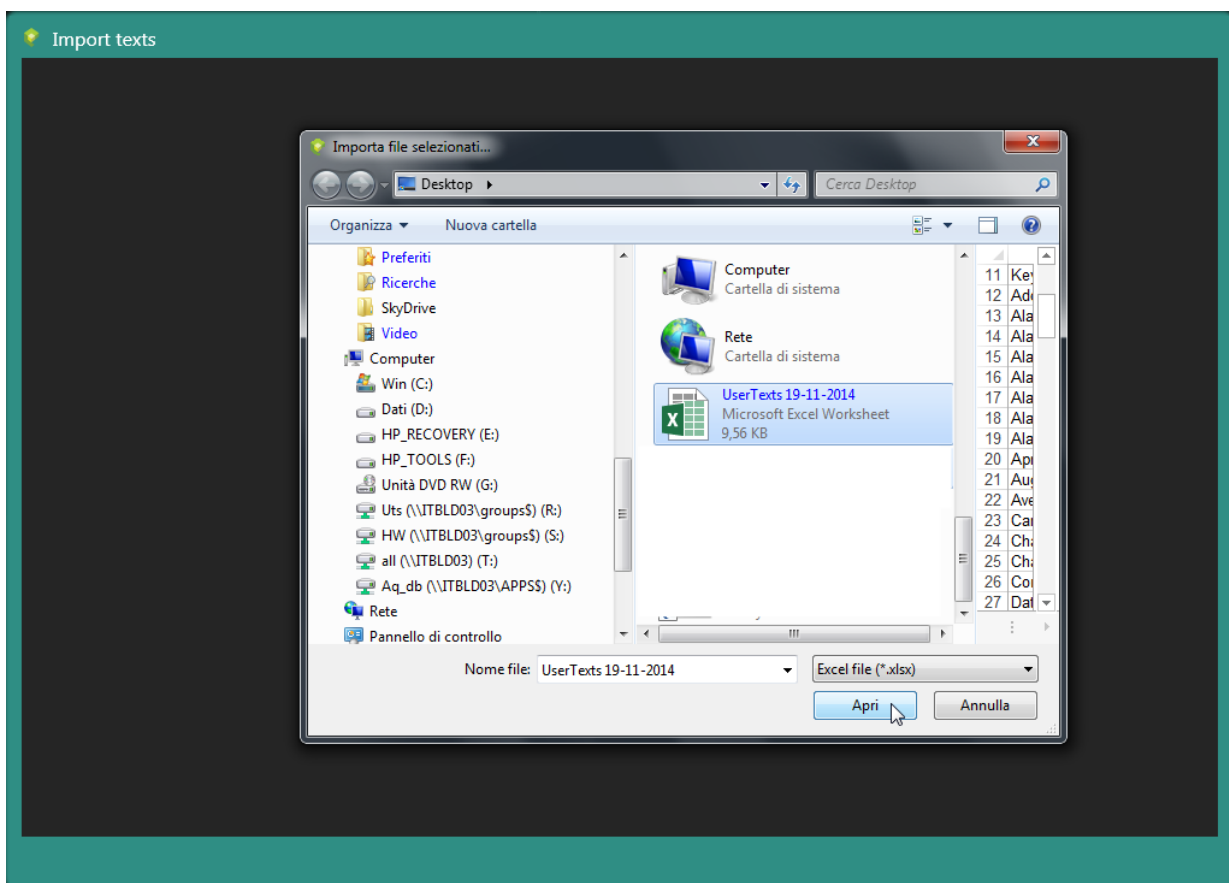
# CREW Manual

## Import a dictionary

Press the “Import dictionary” key to import an Excel file containing one or more dictionary languages.

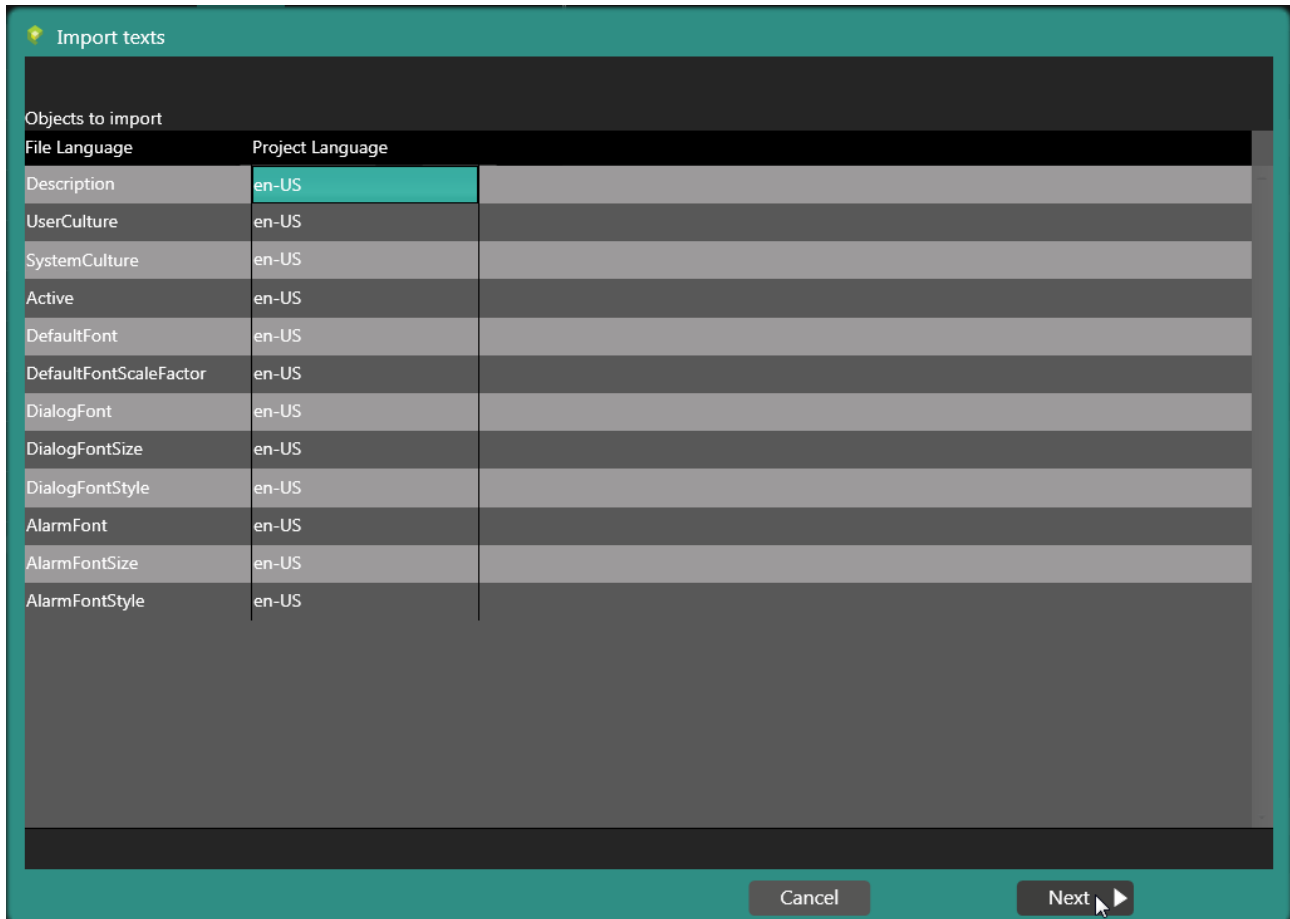


Select the file for import and click “Open”.



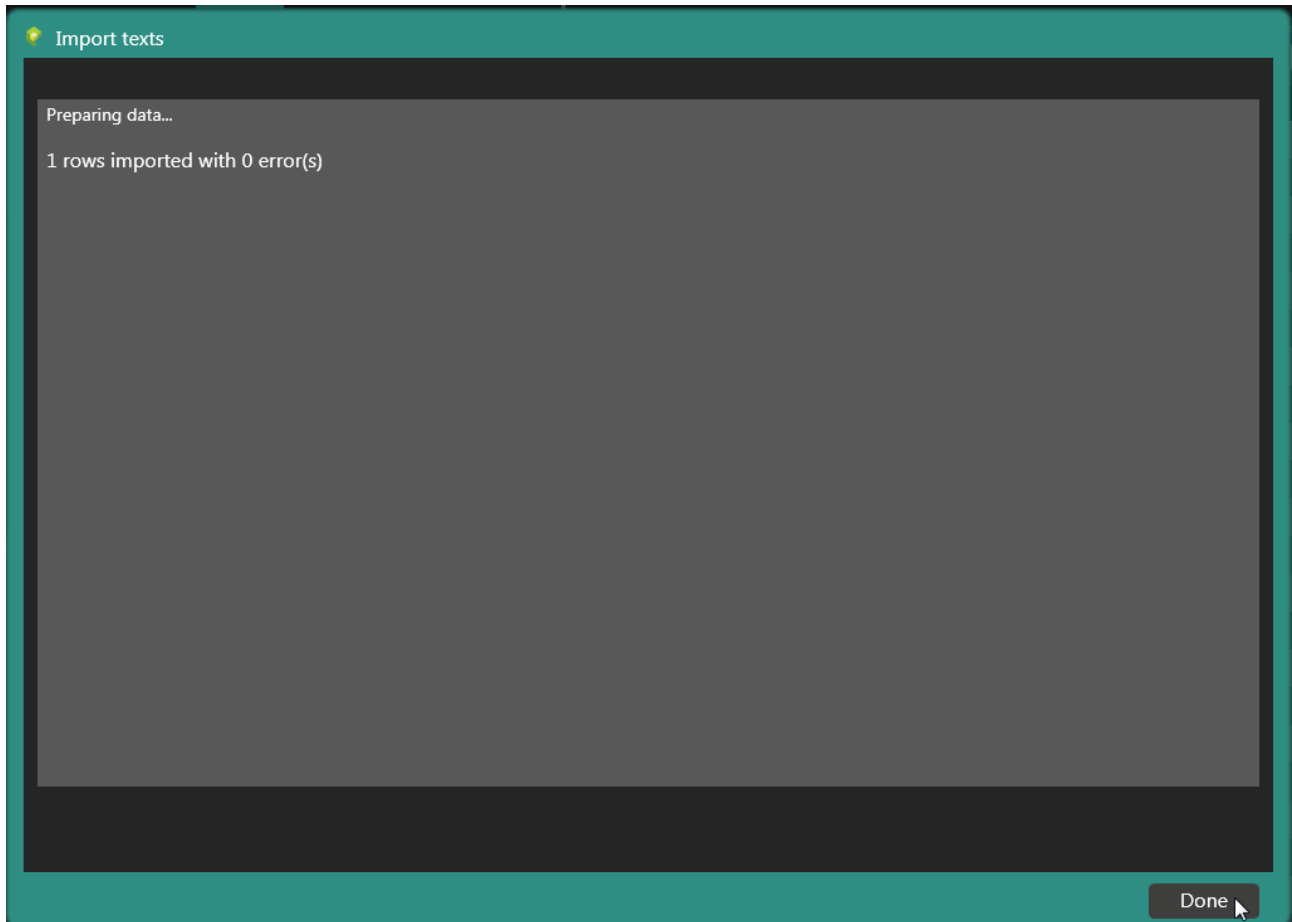
# CREW Manual

Select the objects for import and then click “Next”.



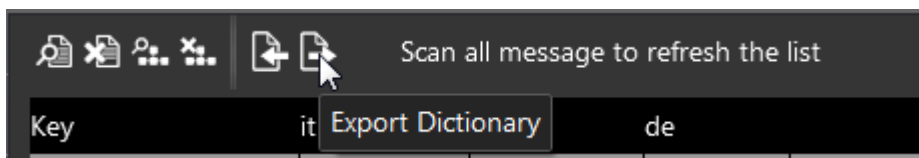
# CREW Manual

At the end of the operation click “End”.



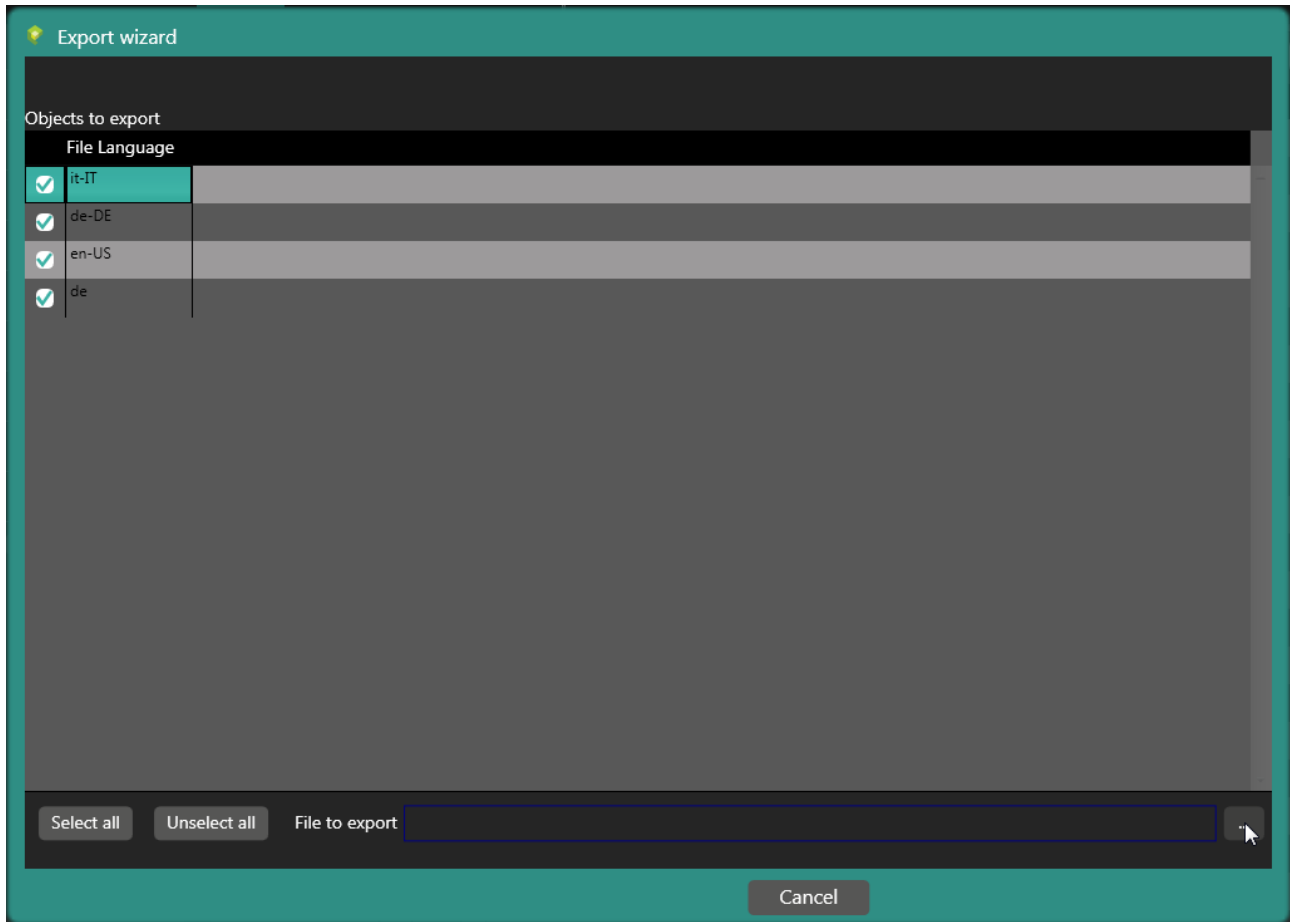
## Export a dictionary

It is possible to export the dictionary to an Excel format file by pressing the “Export dictionary” key.



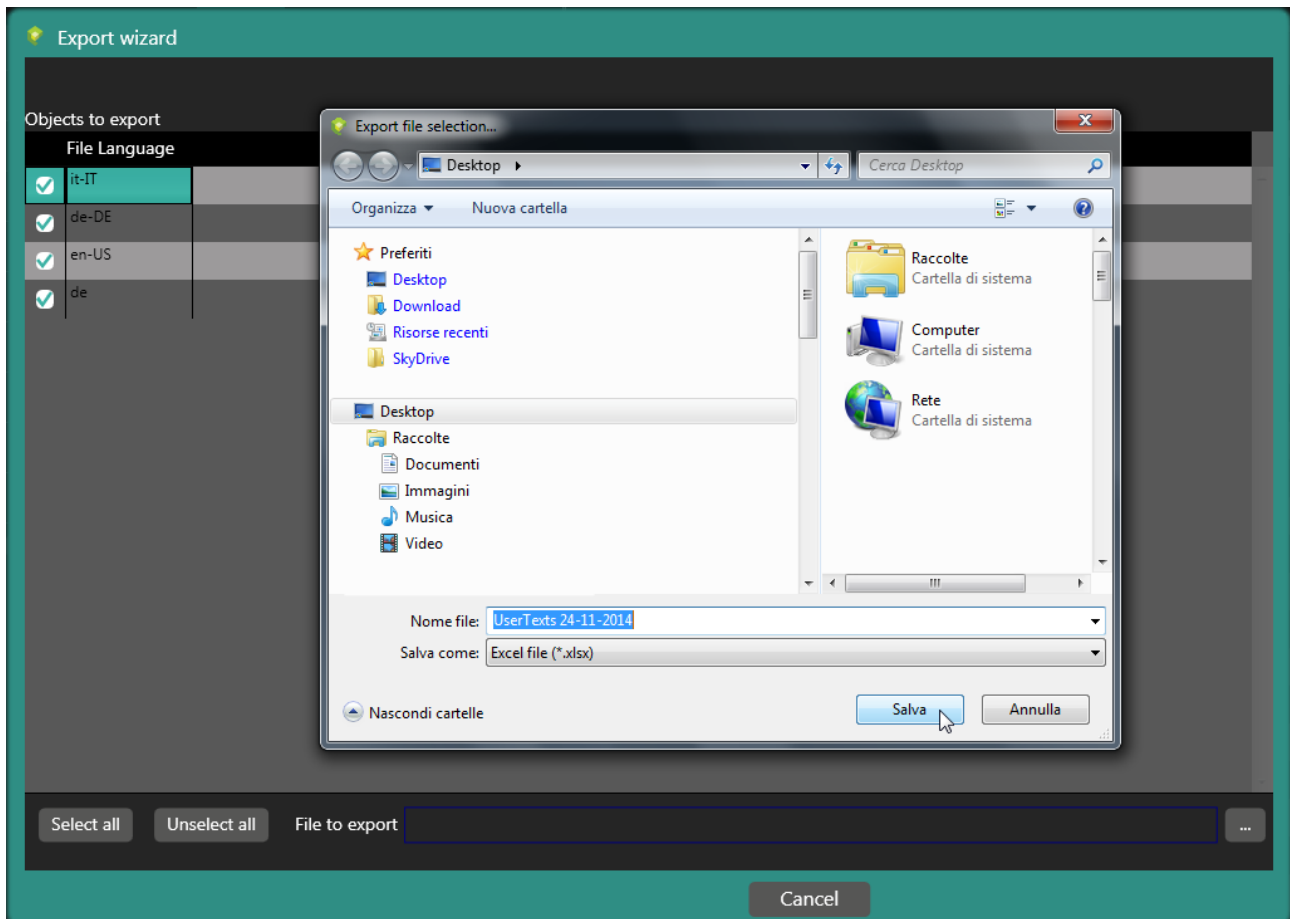
# CREW Manual

Select one or more files for export and click the key on the bottom right.



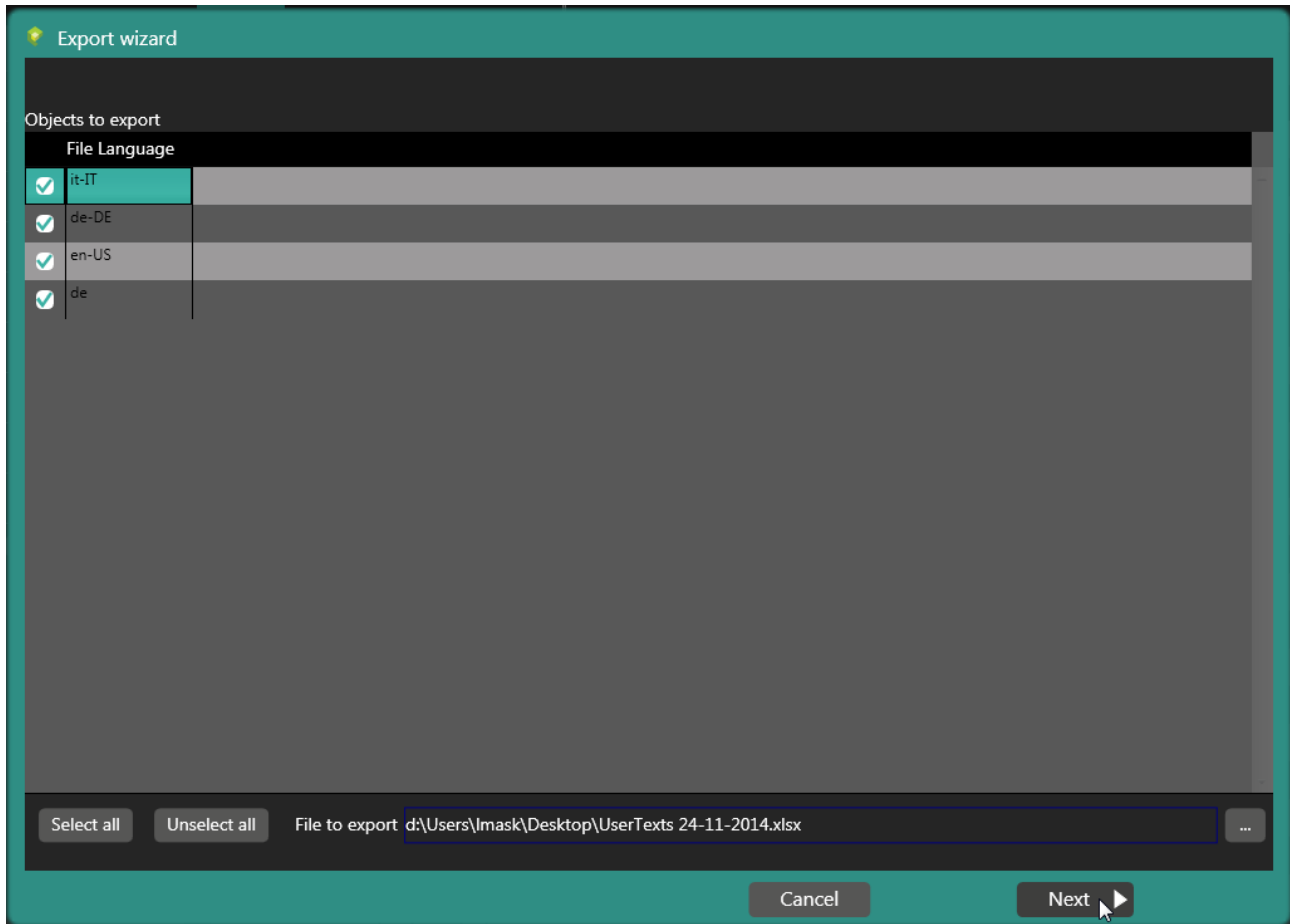
# CREW Manual

Select the required path and then click “Save”.



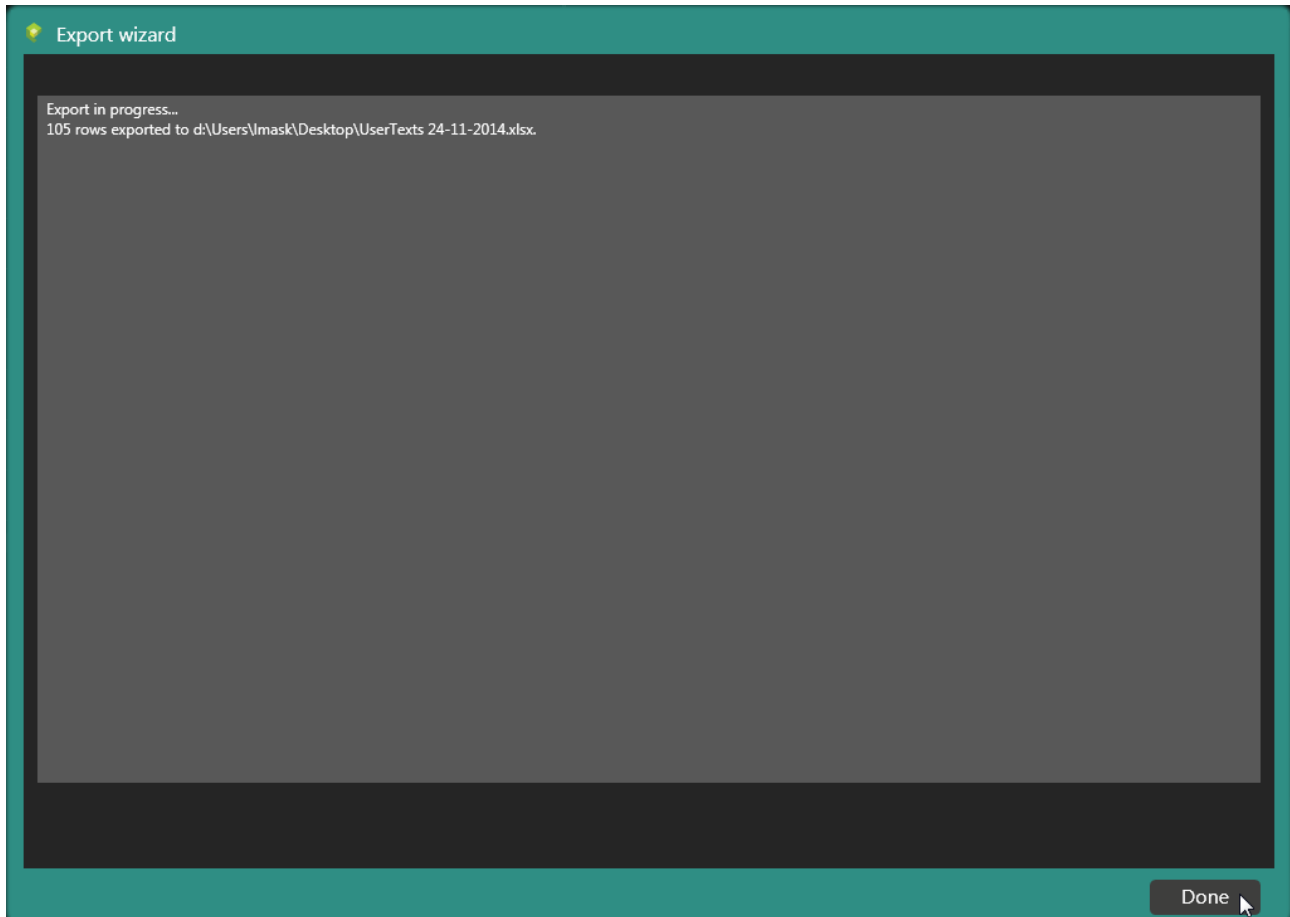
# CREW Manual

Using the relative keys select the objects for export and click “Next”.



# CREW Manual

At the end of the operation click “End”.





# CREW Manual

## Run a language filter

Select a language from the “Filter” drop down menu to view only the terms entered in that specific language.

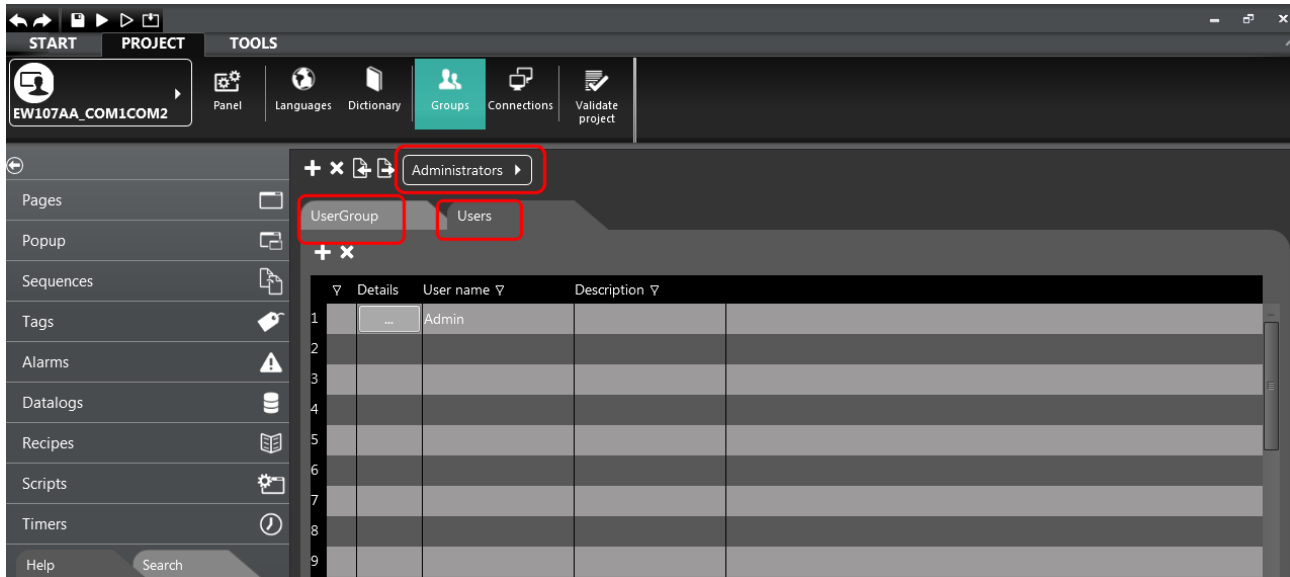


For example, if you select “English” only the column containing terms in English will be displayed.



# CREW Manual

## Groups



The users that interact with the project can be grouped together into “Groups”, who are assigned varying access levels depending on their credentials as administrator (level 1 by default) or as simple users (level 10 by default).

More specifically, within a project it is possible to define the levels of authentication to control access to specific areas. This feature has the purpose of distinguishing and controlling the degree of operational freedom that each user can possess during his/her work session.

With Crew, the programmer can establish access policies for certain features (access to buttons, pages, recipe management etc., for example) and thereby prevent operators without adequate credentials from accessing data or editing it improperly.

During panel use, each operator can identify him/herself through a user name and password (login operation) and thereby have his/her access level recognised by the system.

Only one logged on operator (the last user to have logged on) can use the panel at any given moment.

It is possible to define up to ten access levels, with the lowest level (usually 1) being the one with the greatest degree of operational freedom.

# CREW Manual

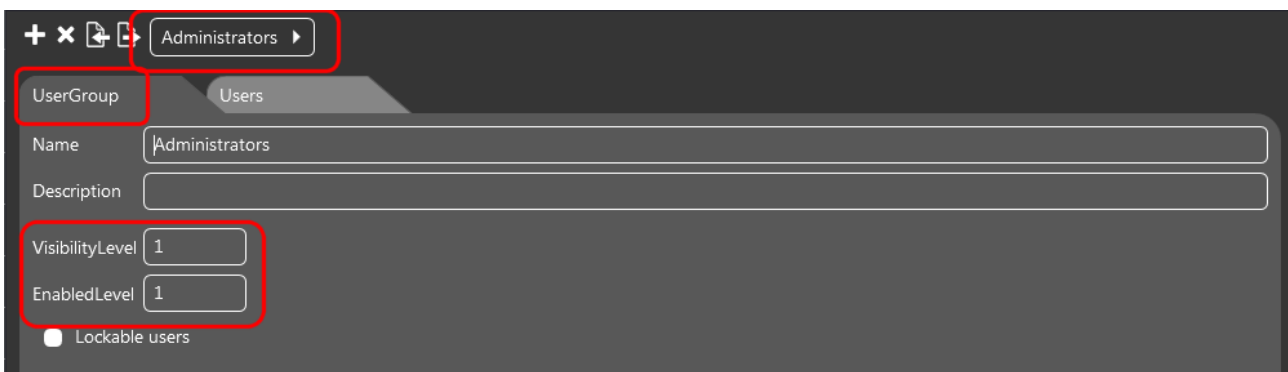
If the user does not log in, he/she is treated by the system as a user of level 10 (lowest level of freedom) and can only access the features allowed for that level. If the non-logged in user tries to perform a task on a level lower than the tenth, the user will be asked to login again through an appropriate pop-up page predefined by the system.

Crew defines the levels of initial users, or rather those who will be present at project startup.

It is also possible to add or change users. For this purpose it is possible to enter a predefined control, referred to as “User List”, in the pages.

For security reasons each operator with access to protected areas of the project can only view and use the object and/or features of the users of equal or higher level. For example, a level 5 user can see and use (by entering his/her password) protected areas of the project that are accessible to level 5, 6, 7, 8, 9 and 10 operators.

## User group - Administrators -



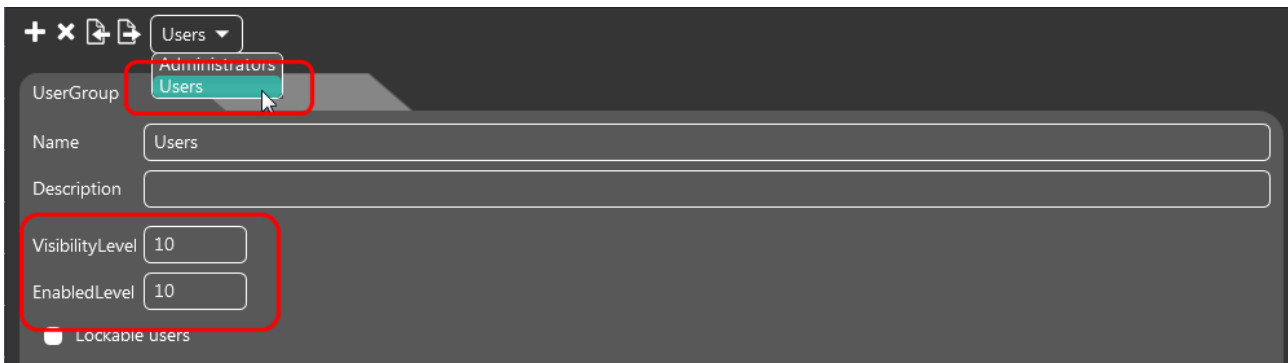
By selecting the “Administrators” and “User Group” option you will note that, as explained above, the levels (viewing level and use level) are set at 1 by default, namely maximum freedom of action within the project; obviously the levels are modifiable by the project executor, the levels range from 1 to 10 and, normally, the administrator is level 1

# CREW Manual

From where we are it is possible to perform the following operations:

- Create new groups
- Delete an existing group
- Import groups
- Export groups

## User group - Users -



By selecting the “Users” and “User Group” option we will note that the levels (viewing level and use level) are set at 10 by default, namely the user belonging to this group can view and/or use the unprotected parts of the project. Also in this case, the levels are modifiable by the executor of the project.

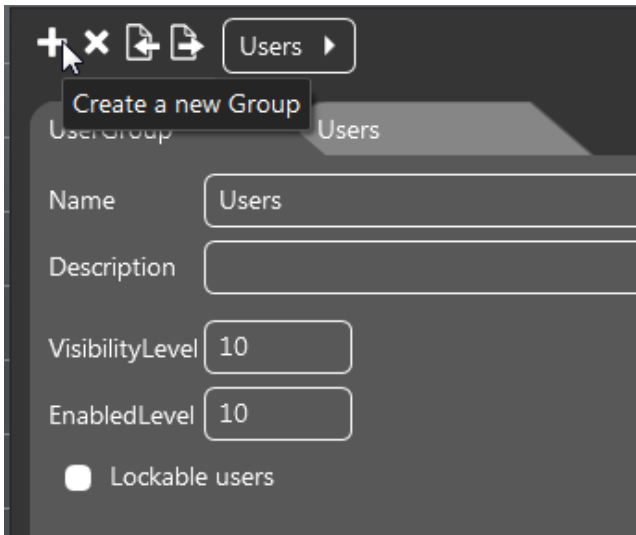
From here it is possible to perform the following operations:

- Create new groups.
- Delete an existing group.
- Import groups.
- Export groups.

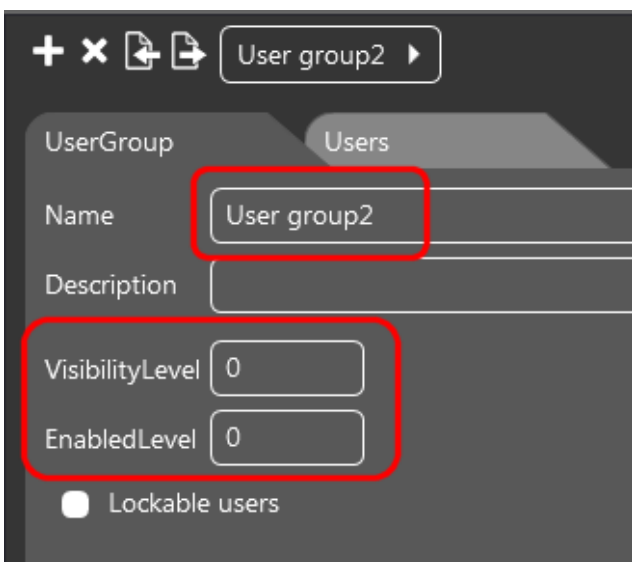
# CREW Manual

## Creating new groups

To create a new group, click on the relative key.



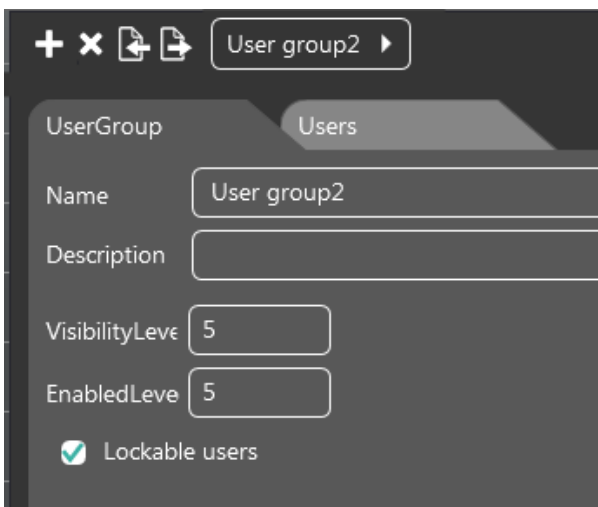
When you have clicked, a new group is created, referred to as “Gruppo utenti2/User group 2” which is assigned default level “0”.



# CREW Manual

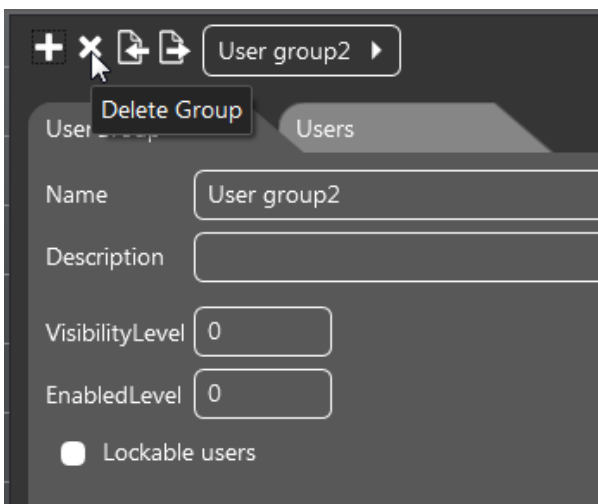
Level “0” must be changed by the user, who can establish whether to assign a level equal to the “Administrators” group (level 1) or whether to choose a level equal to or higher than “2” and thereby determine the degree of importance of the newly-created group.

Enable the “Blockable users” check box if you wish to be able to block users belonging to the group (in this case “Gruppo utenti2/User group 2”) at a later time. Refer to the "[User Properties](#)" section for more details.



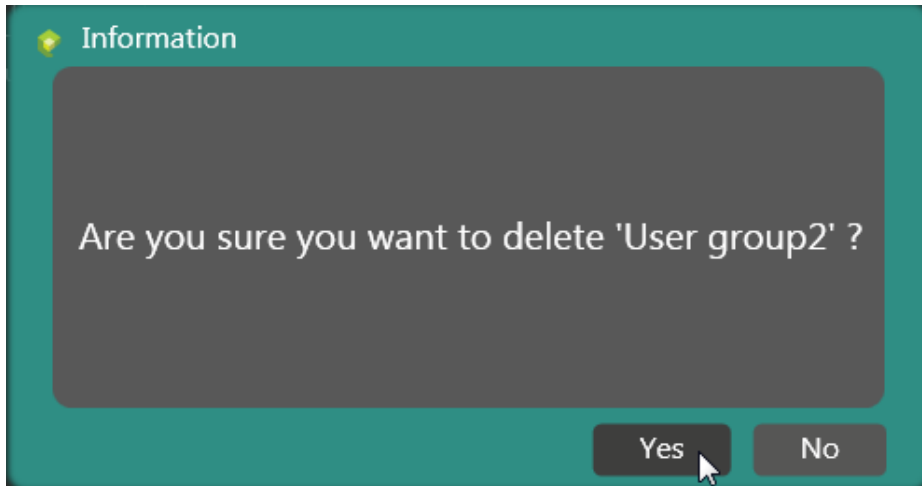
## Deleting an existing group

Click the “Delete Group” key to delete a group.



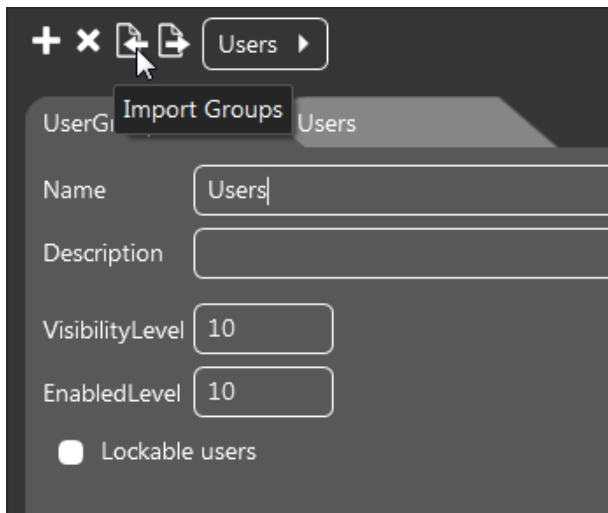
# CREW Manual

Click "Yes" to confirm your choice.



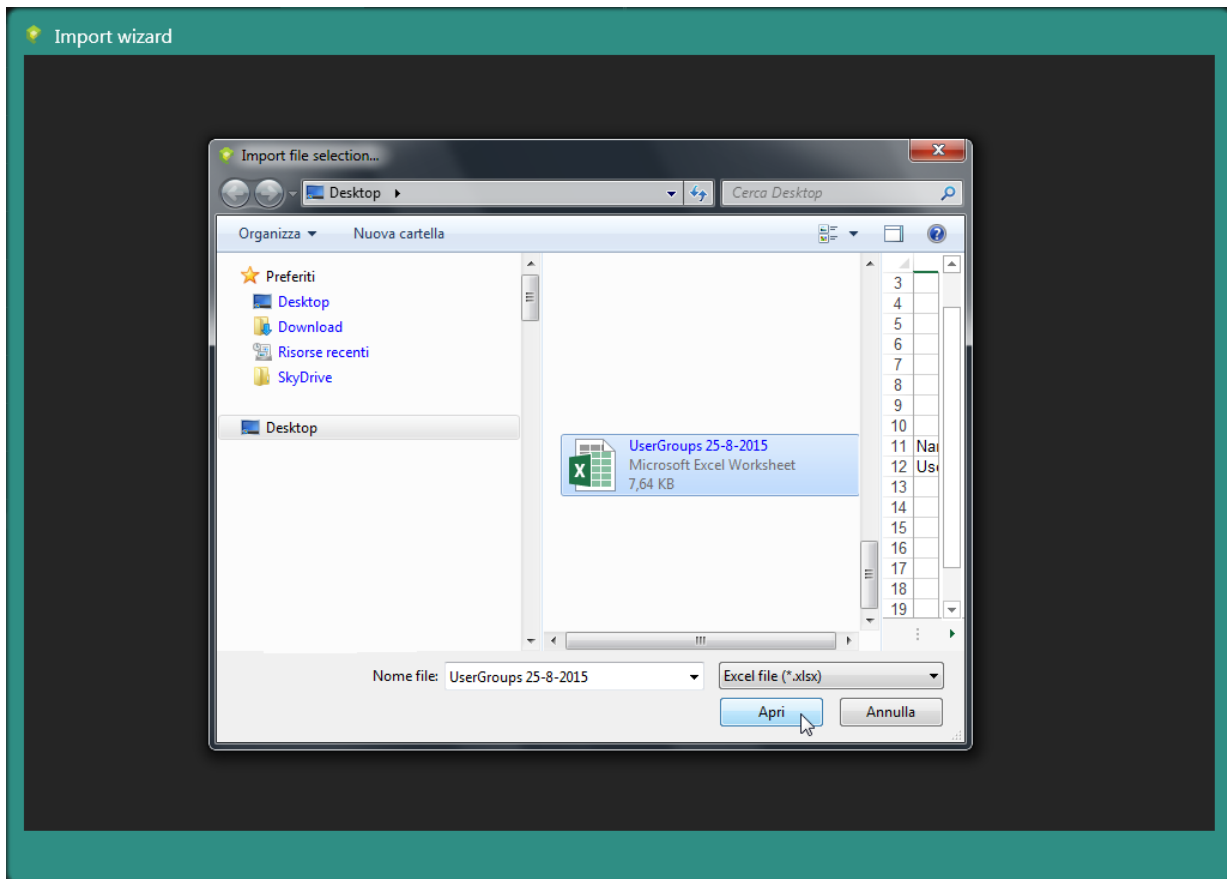
## Importing Groups

Press the "Import Groups" key to import a previously created user groups (in Excel file format).



# CREW Manual

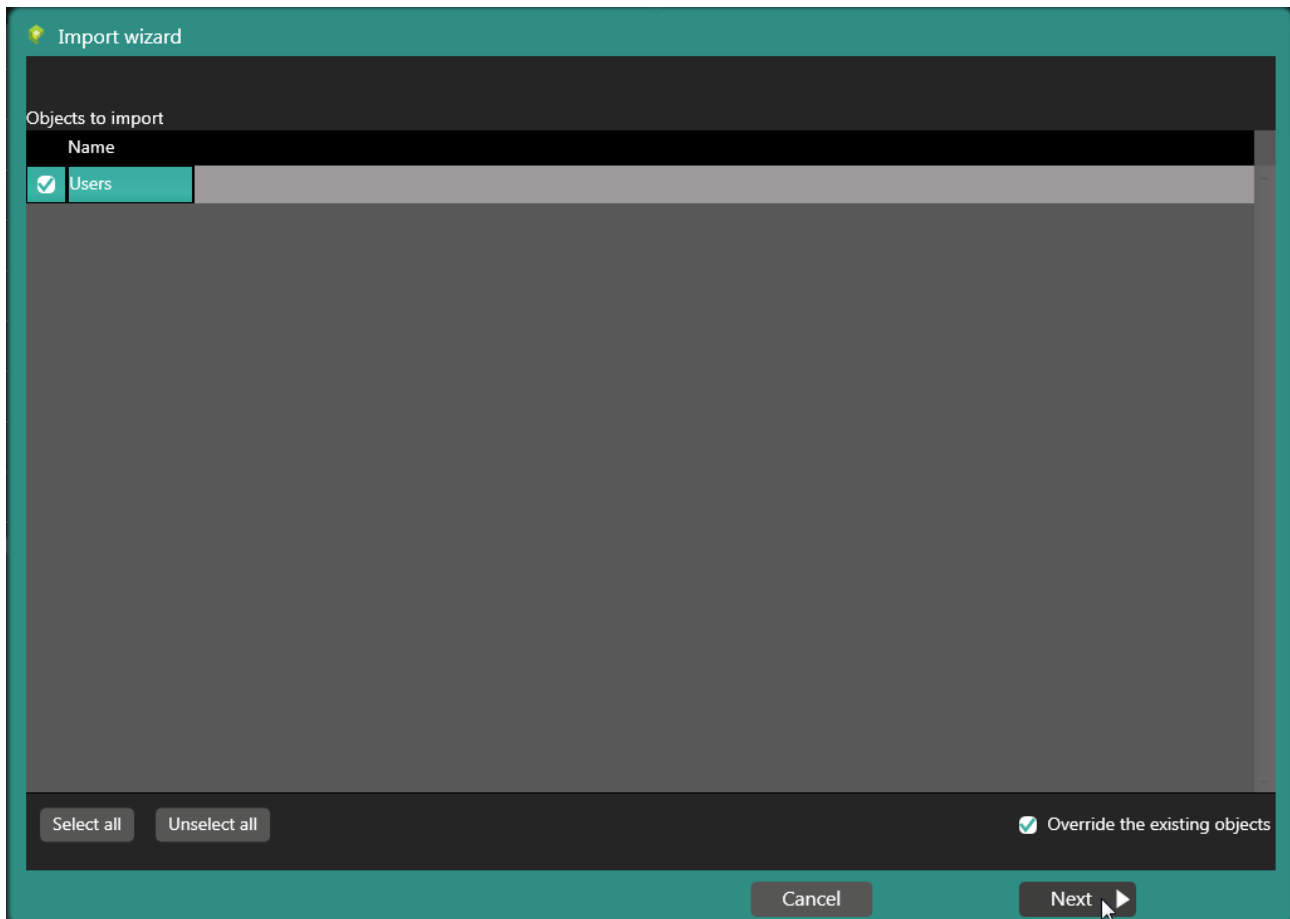
Select the file for import and click “Open”.





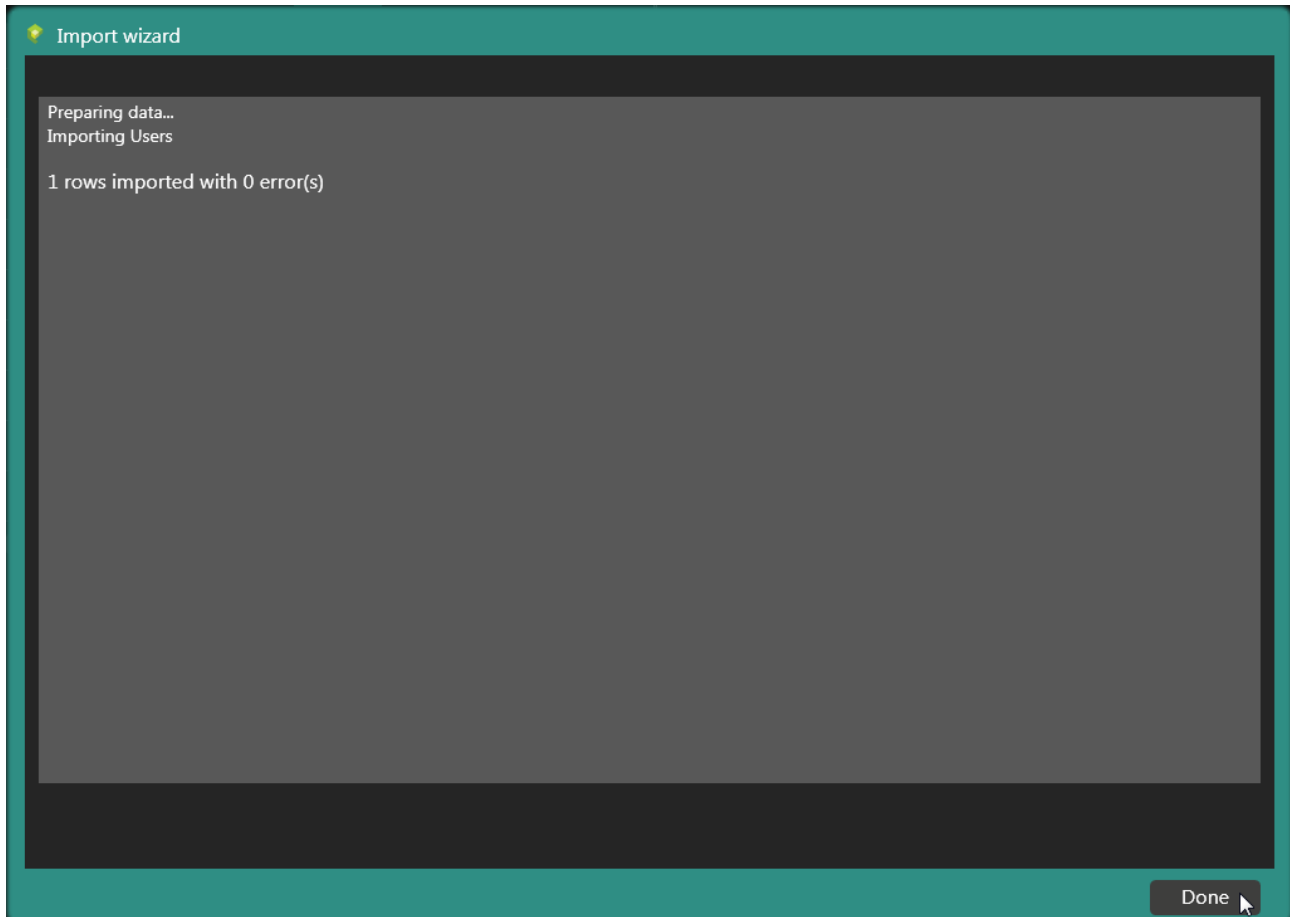
# CREW Manual

Select the objects for import and decide whether to overwrite the existing objects or not, using the relative keys. Then click “Next”.



# CREW Manual

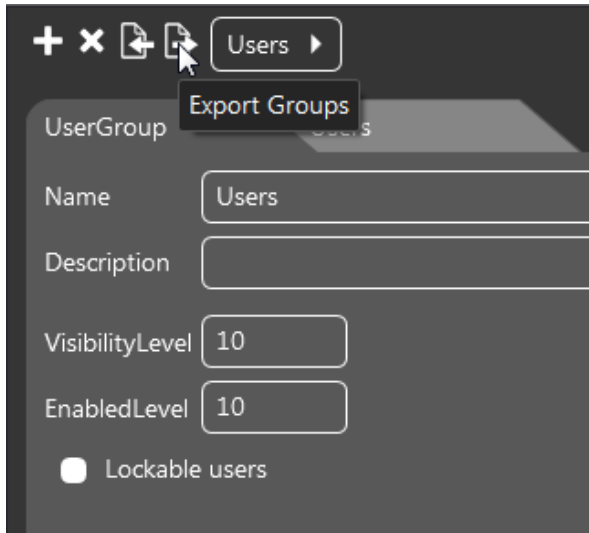
At the end of the operation click “End”.



# CREW Manual

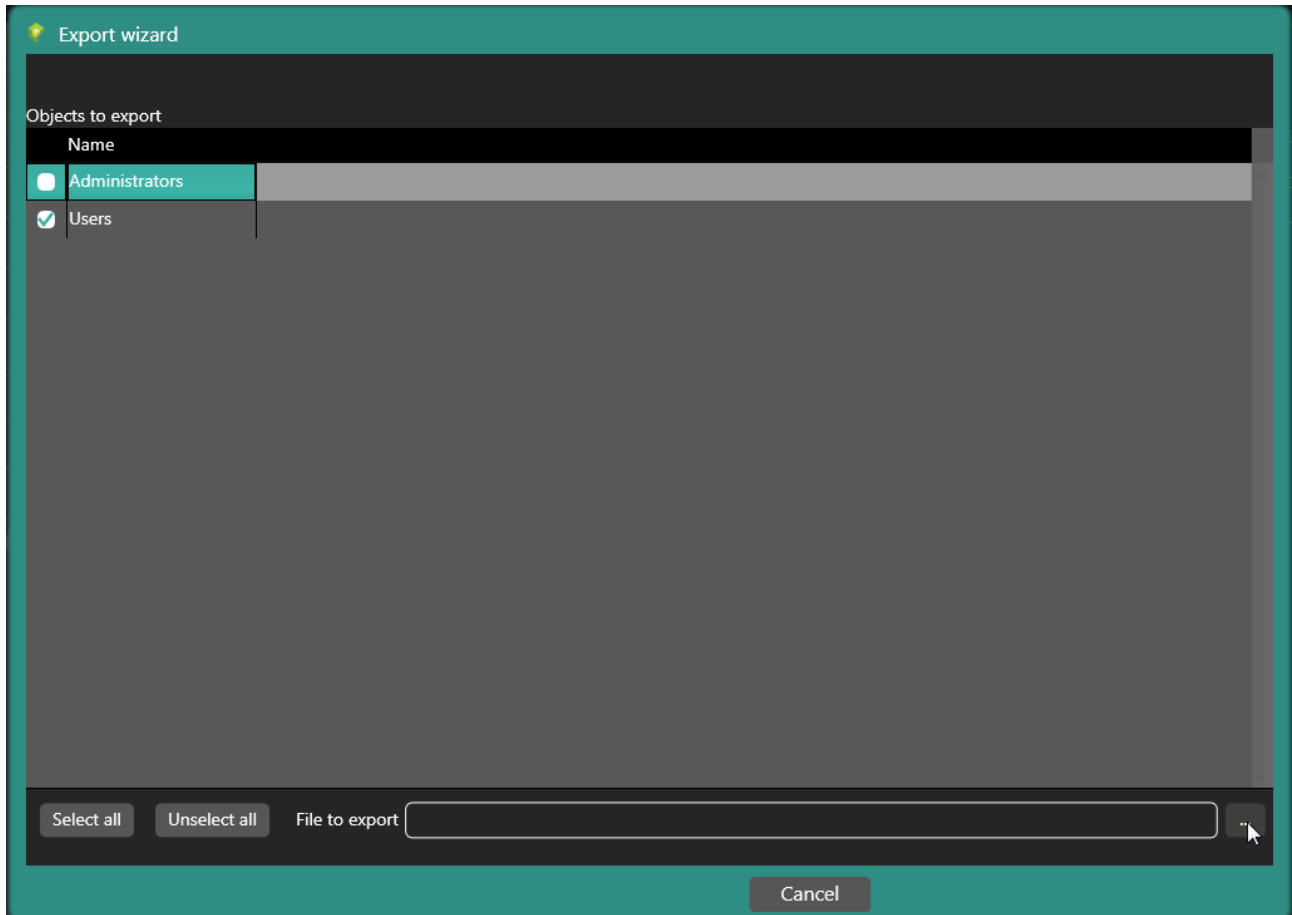
## Exporting Groups

Press the “Export Groups” key to export user groups in Excel format.



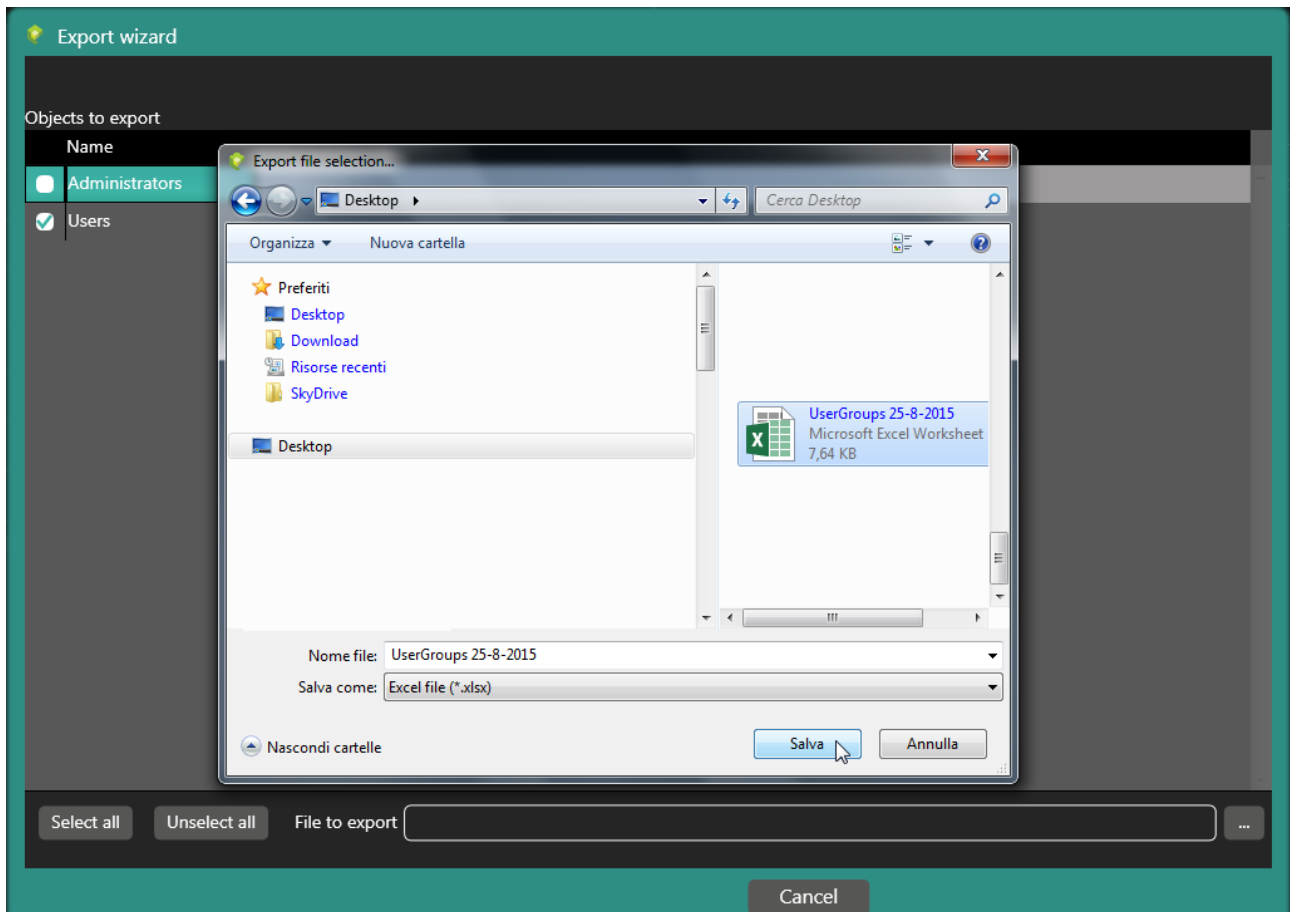
# CREW Manual

Select the file for export and click the key on the bottom right.



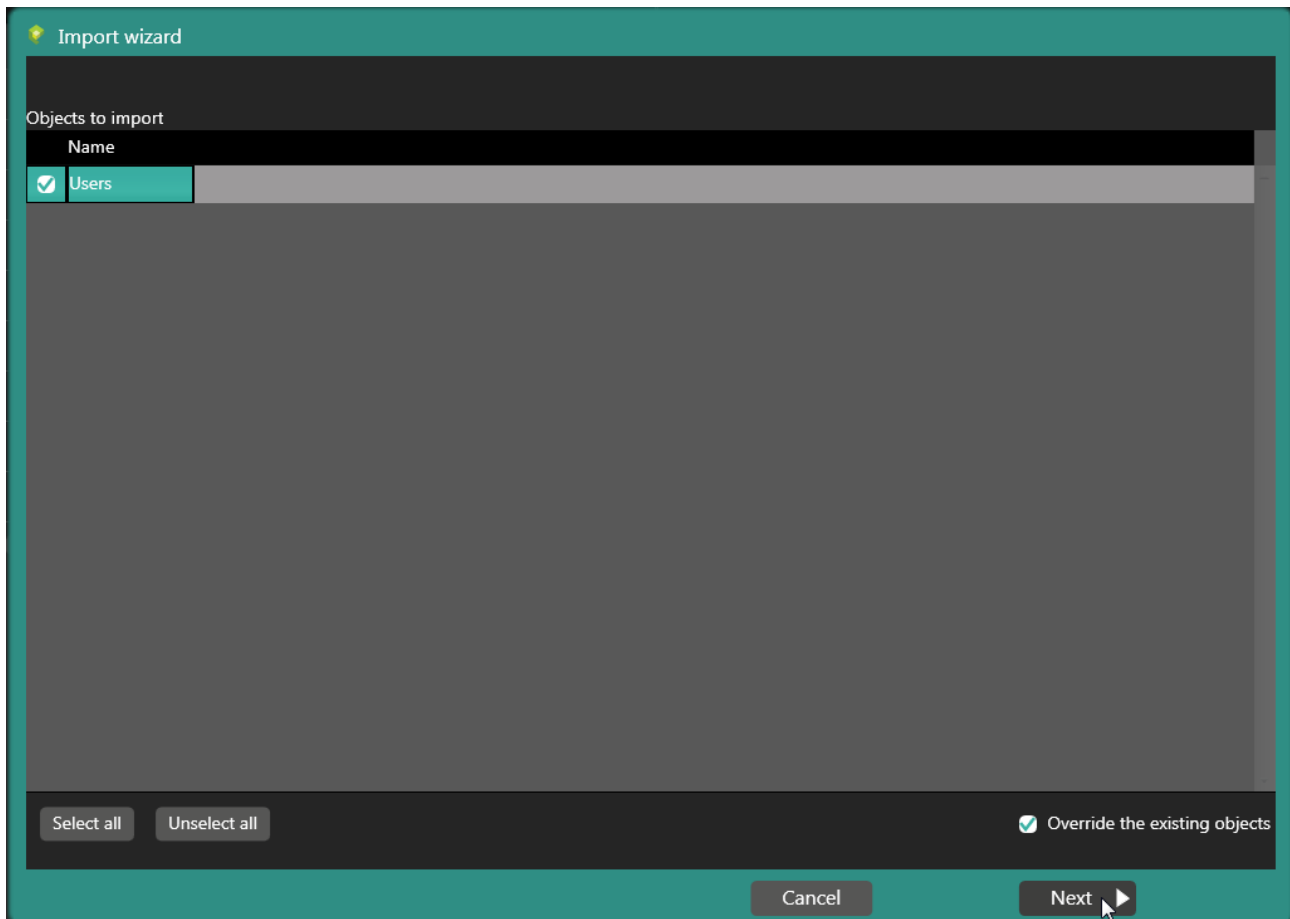
# CREW Manual

Select the required path and click “Save”.



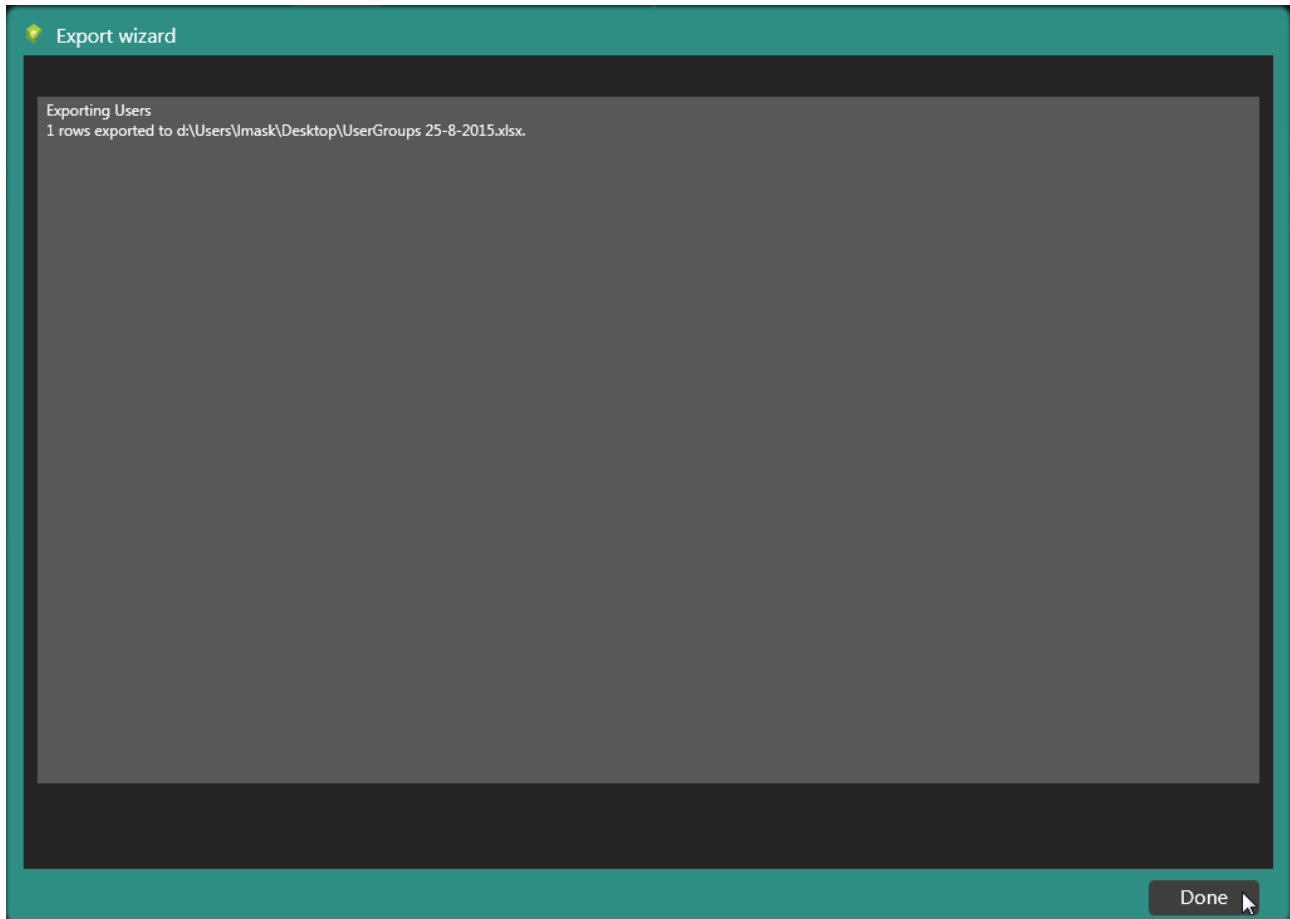
# CREW Manual

Select the objects for export with the relative keys and click “Next”.



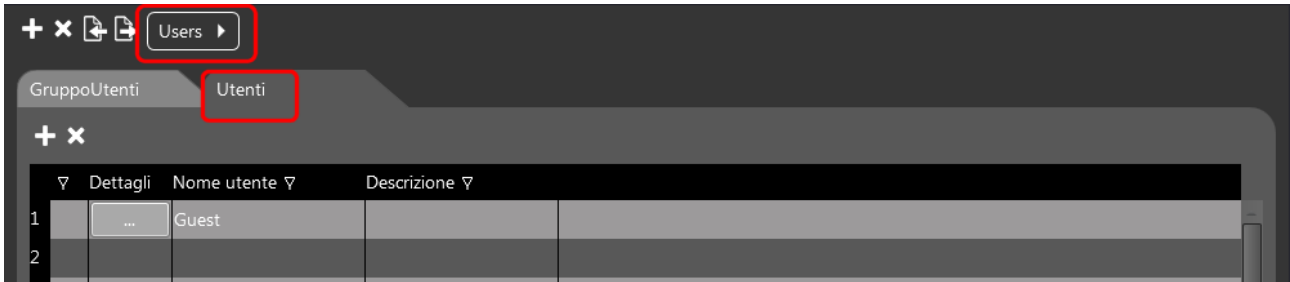
# CREW Manual

At the end of the operation click “End”.



# CREW Manual

## Users

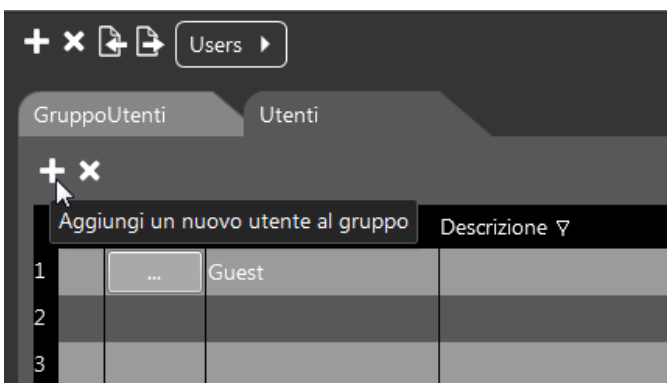


It is possible to enter users into the Users user group, as needed. Obviously users who are part of this group have normal user credentials (level 10 by default, therefore they cannot access the protected areas of the project). Also in this case, the executor of the project can change the levels.

From here it is possible to perform the following operations:

- Add new users.
- Delete an existing user.
- Set username, password and entry mode (normal password or graphics password).

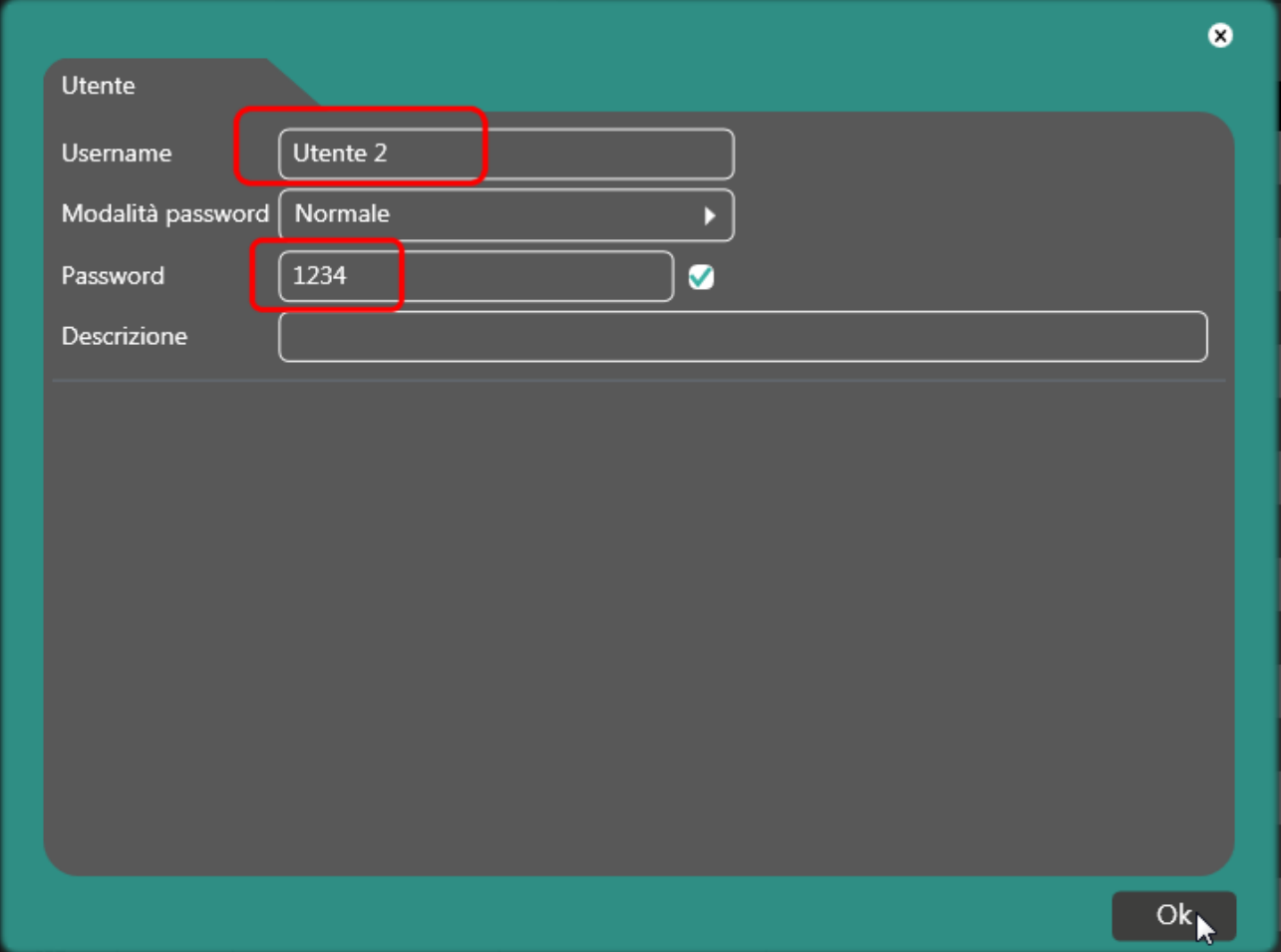
### Adding Users - Users -





# CREW Manual

Enter username and password and confirm with “Ok”.



Utente

Username: Utente 2

Modalità password: Normale

Password: 1234

Descrizione:

Ok

The newly created user is visible in the list of users in the Users group. Also note that on the right of the mask the option “Move to” appears, which can be used to move the selected user to a different group.



Users

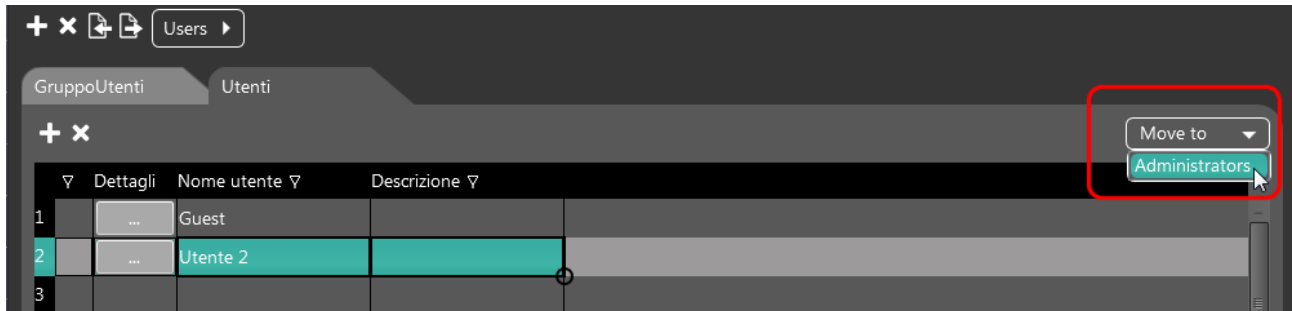
GruppoUtenti: Utenti

	Dettagli	Nome utente	Descrizione
1	...	Guest	
2	...	Utente 2	
3			

Move to

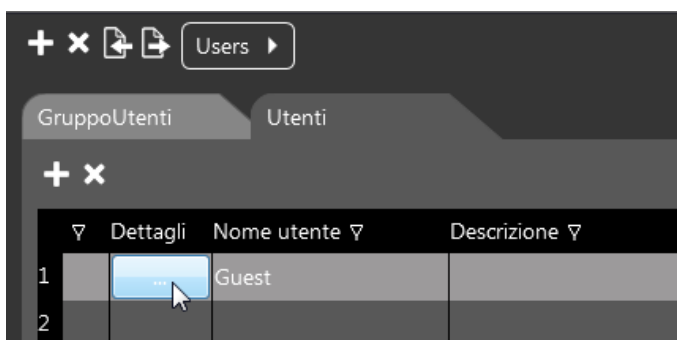
# CREW Manual

For example, it is possible to move the user to the “Administrators” group.



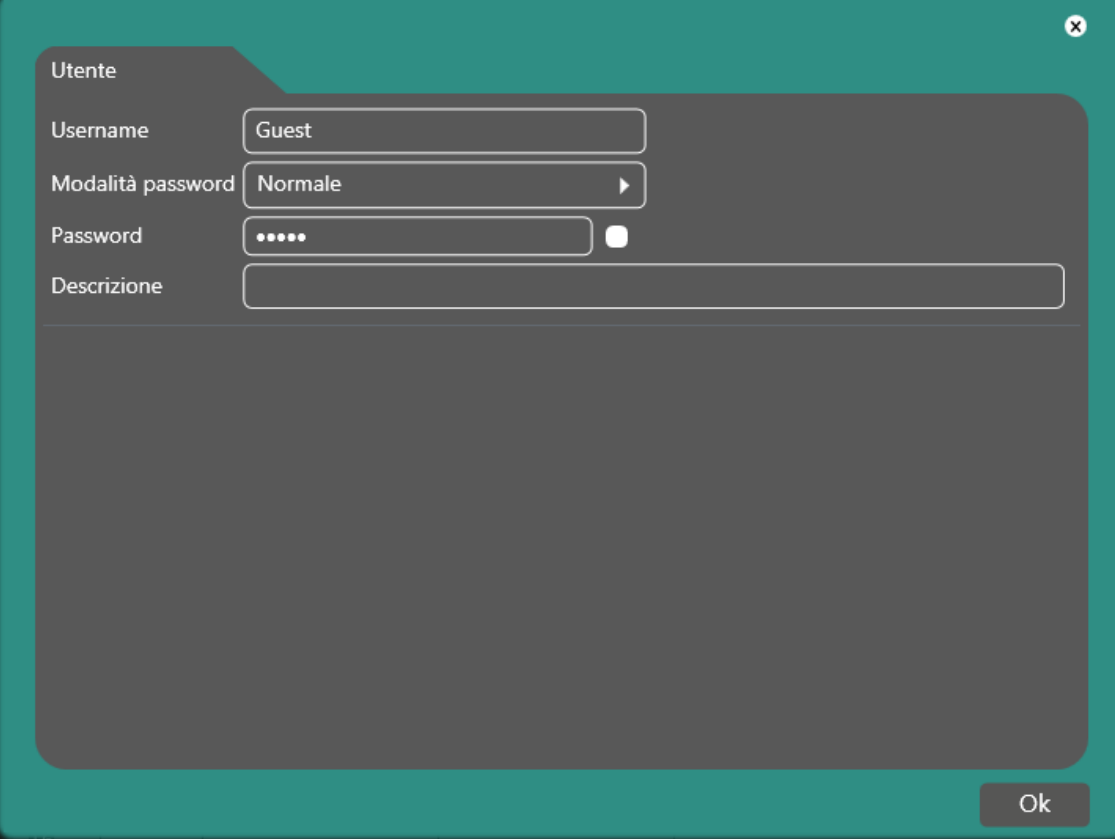
Username and Password - Users -

Click the “Details” key to set username and password.



# CREW Manual

The username and password are set by default in the window that appears.



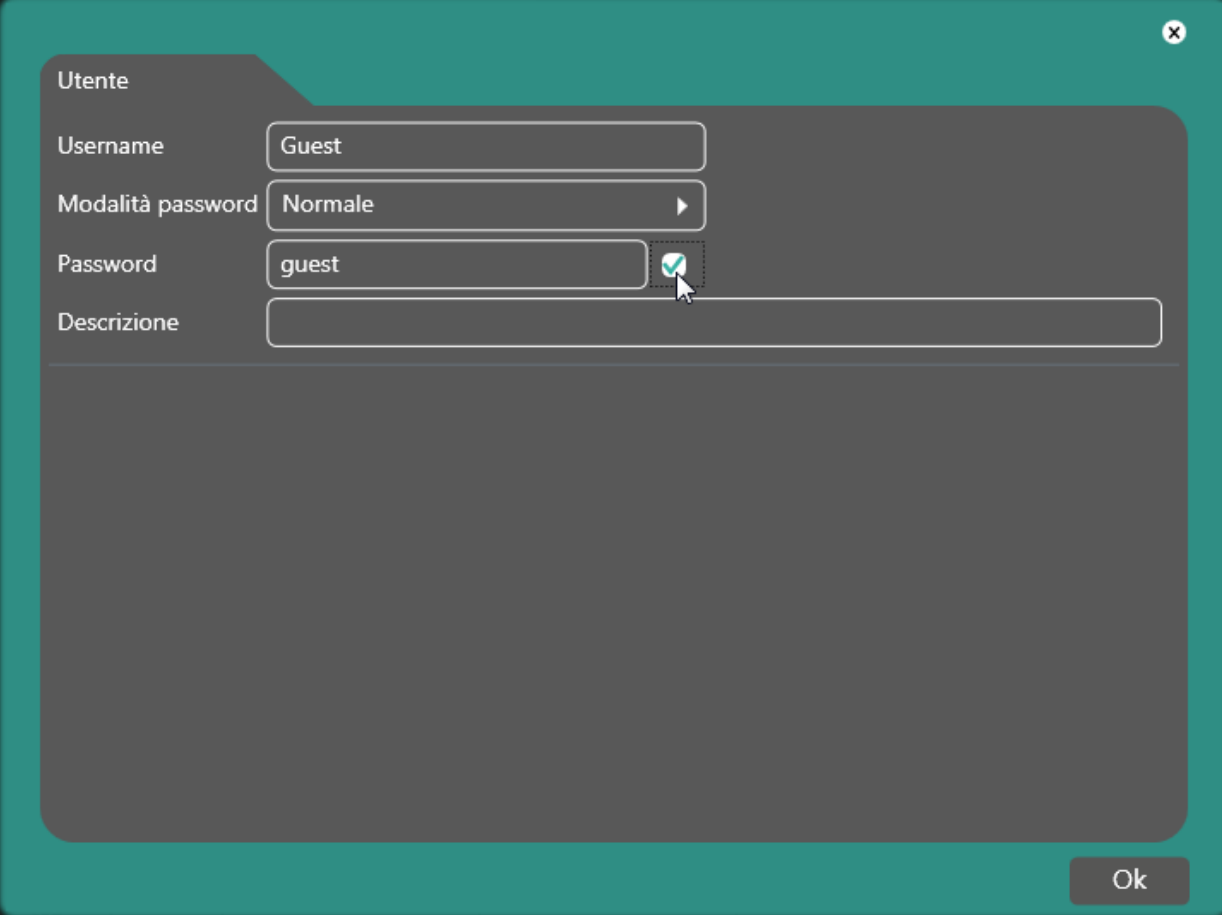
The image shows a software window titled "Utente" (User) with a close button (X) in the top right corner. The window contains the following fields:

- Username:** A text input field containing the value "Guest".
- Modalità password:** A dropdown menu currently set to "Normale" (Normal).
- Password:** A text input field containing five dots, indicating a masked password. To the right of the field is a small square icon, likely for toggling password visibility.
- Descrizione:** A large, empty text area for entering a description.

An "Ok" button is located at the bottom right of the window.

# CREW Manual

Select the corresponding checkbox to make the password visible.

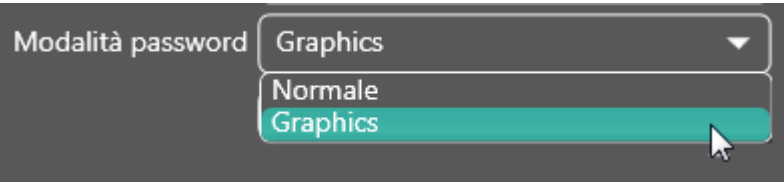


The screenshot shows a dialog box titled "Utente" with a close button (X) in the top right corner. It contains the following fields:

- Username:** A text input field containing "Guest".
- Modalità password:** A dropdown menu currently set to "Normale".
- Password:** A text input field containing "guest". To its right is a checked checkbox, which is being clicked by a mouse cursor.
- Descrizione:** An empty text input field.

An "Ok" button is located at the bottom right of the dialog box.

Select the "Graphics" password mode



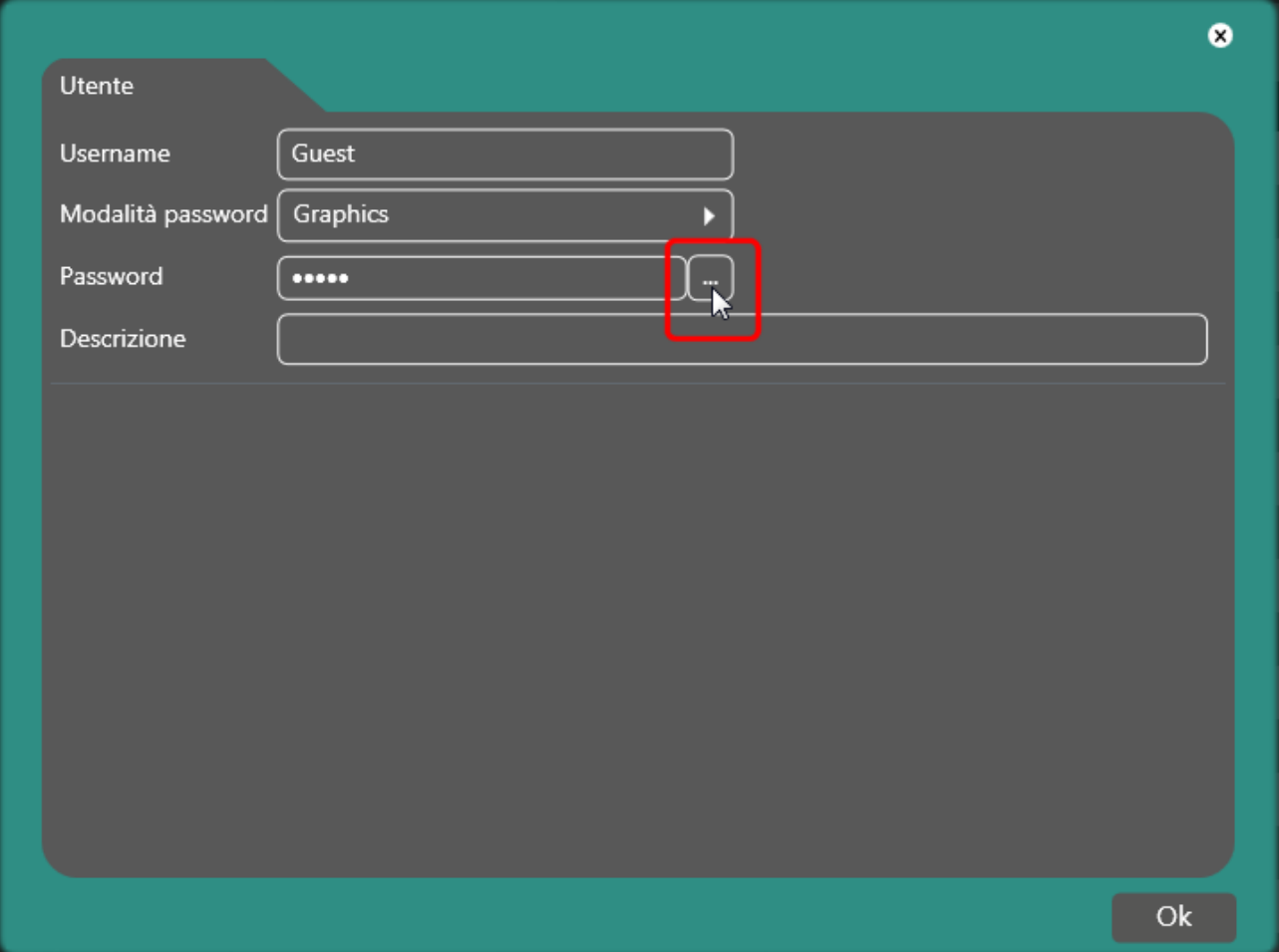
This image is a close-up of the "Modalità password" dropdown menu. The menu is open, showing three options:

- Graphics
- Normale
- Graphics

The second "Graphics" option is highlighted in teal, and a mouse cursor is pointing at it.

# CREW Manual

and click the “Browse” key



Utente

Username

Modalità password

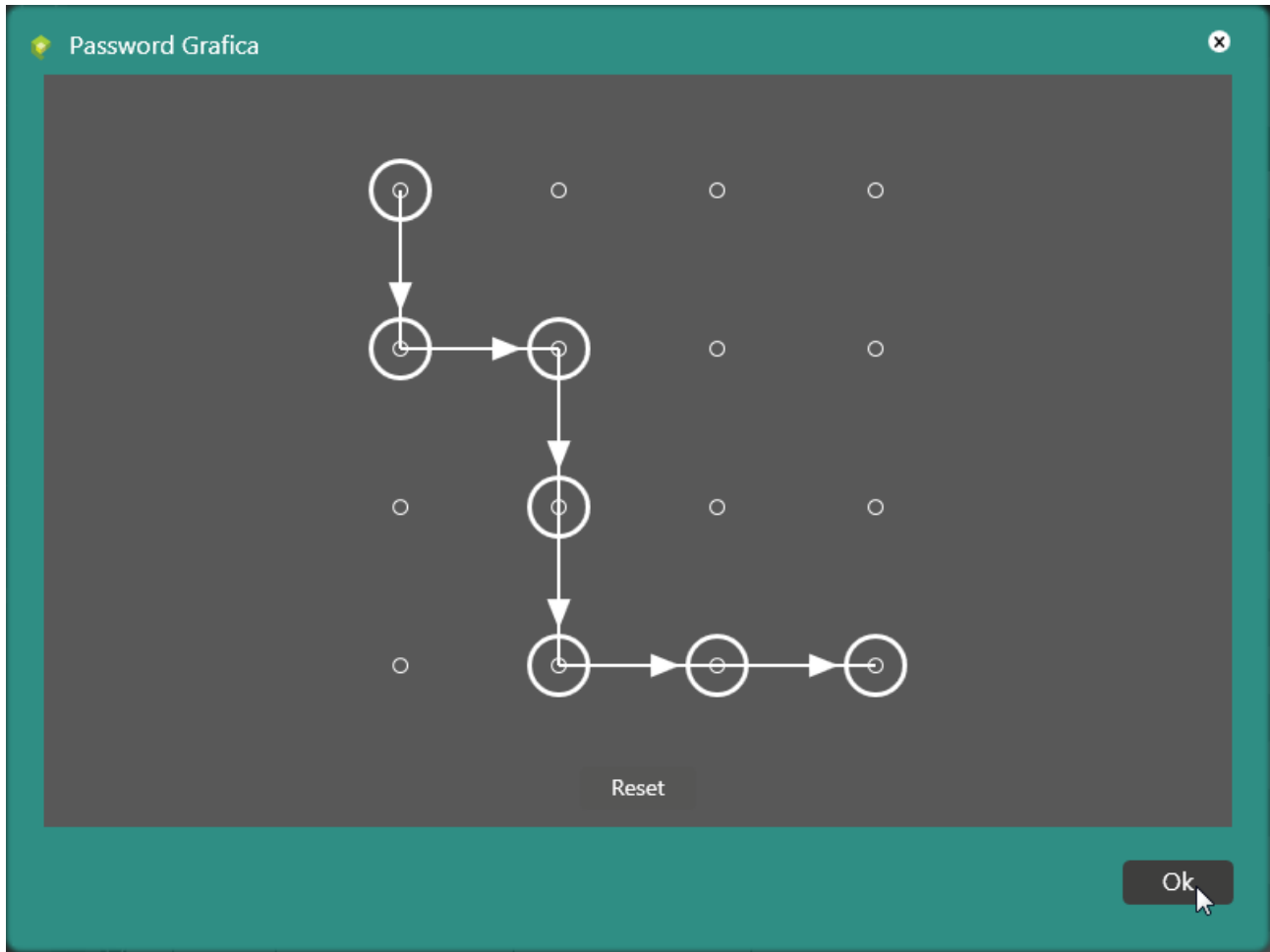
Password

Descrizione

Ok

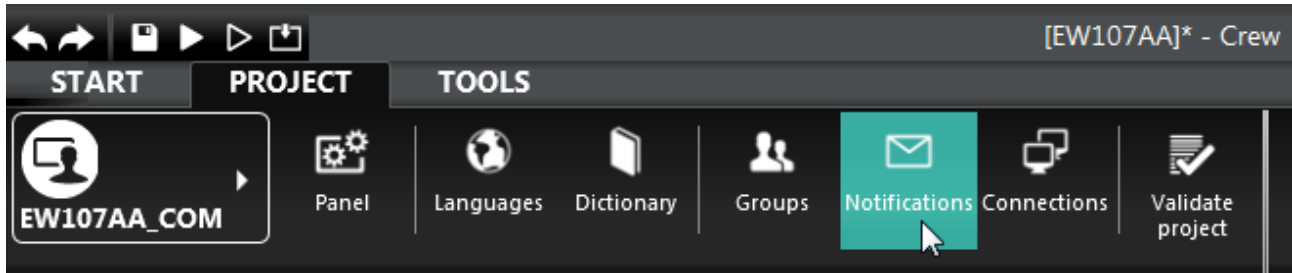
# CREW Manual

to set the graphics password.



# CREW Manual

## Notifications



Crew offers SMS- and email-management of events or alarms notifications (from Crew version 1.06).

It is possible to match a personalised SMS and/or E-mail, for every single event or verification of a specific alarm condition.



**Note:** The “Notifications” function can be used by having preventively installed the Everyware software on the terminal where the project is kept. Plus, there needs to be an internet connection with the Everyware server.

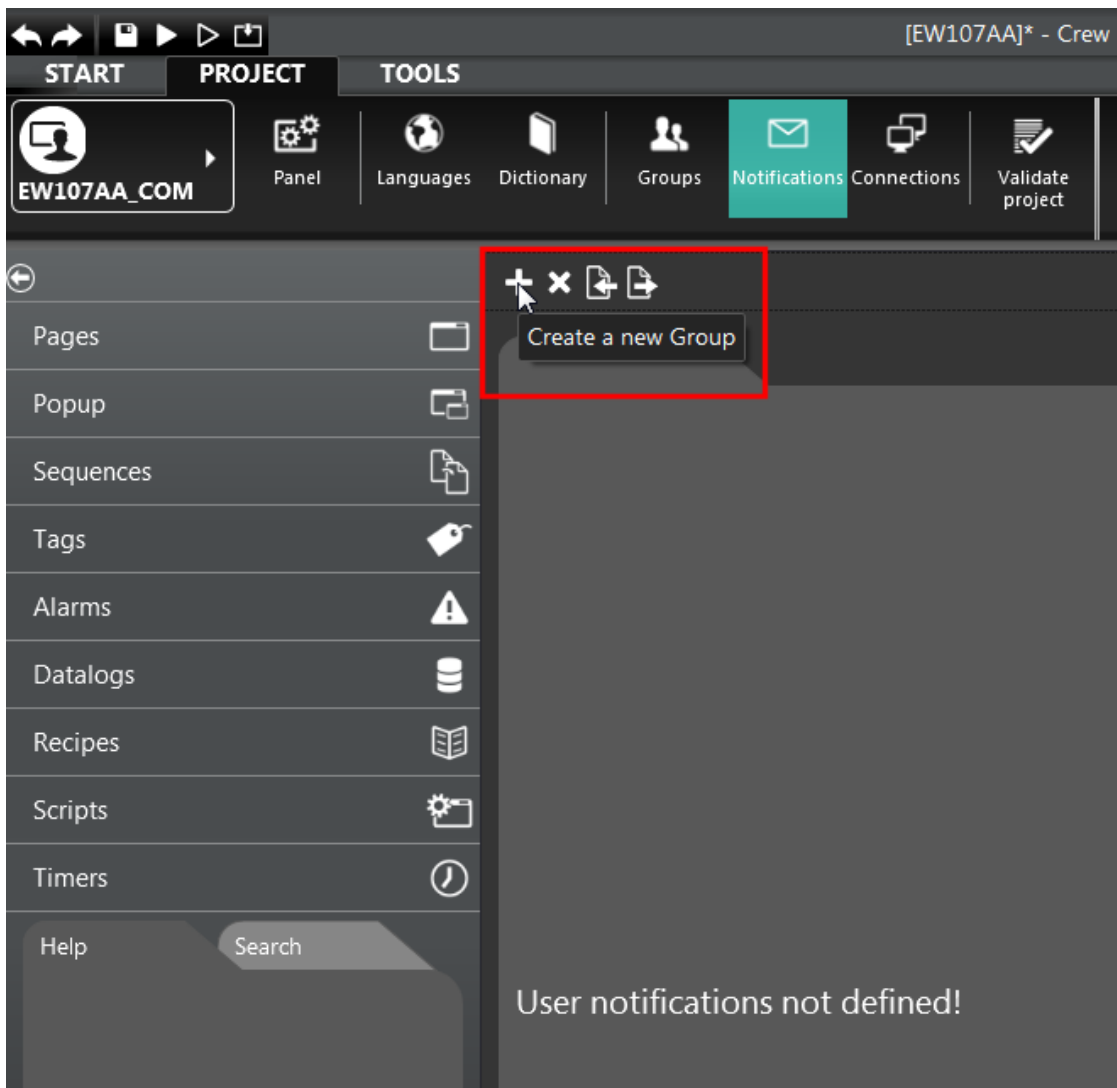
Through Crew it is possible to limit sending these communications to predefined user groups.

For example, when the notification message is associated to a given alarm in the project, a message can be sent to one or more users.

The message can be associated to one or more of the three alarm states: “Raised”, “Acknowledged”, “Acquired”.

# CREW Manual

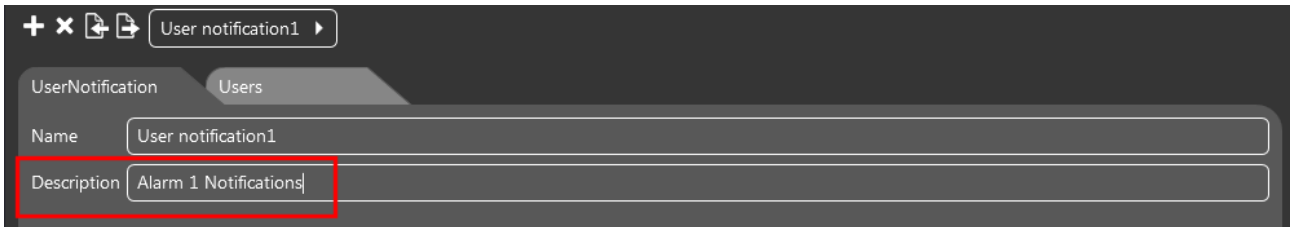
From the “Project” menu, click “Notifications” to access the notification area. Here, it is firstly necessary to create a new “Group for notifications” by clicking the relative icon.





# CREW Manual

It is possible to enter a description of the created group.



Click “Users” to add new users to this group.

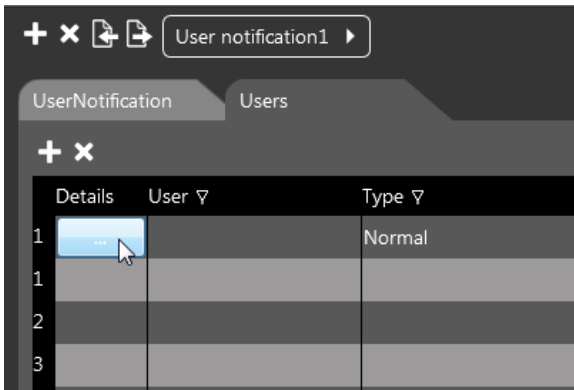


Click the “Add a new user” key.

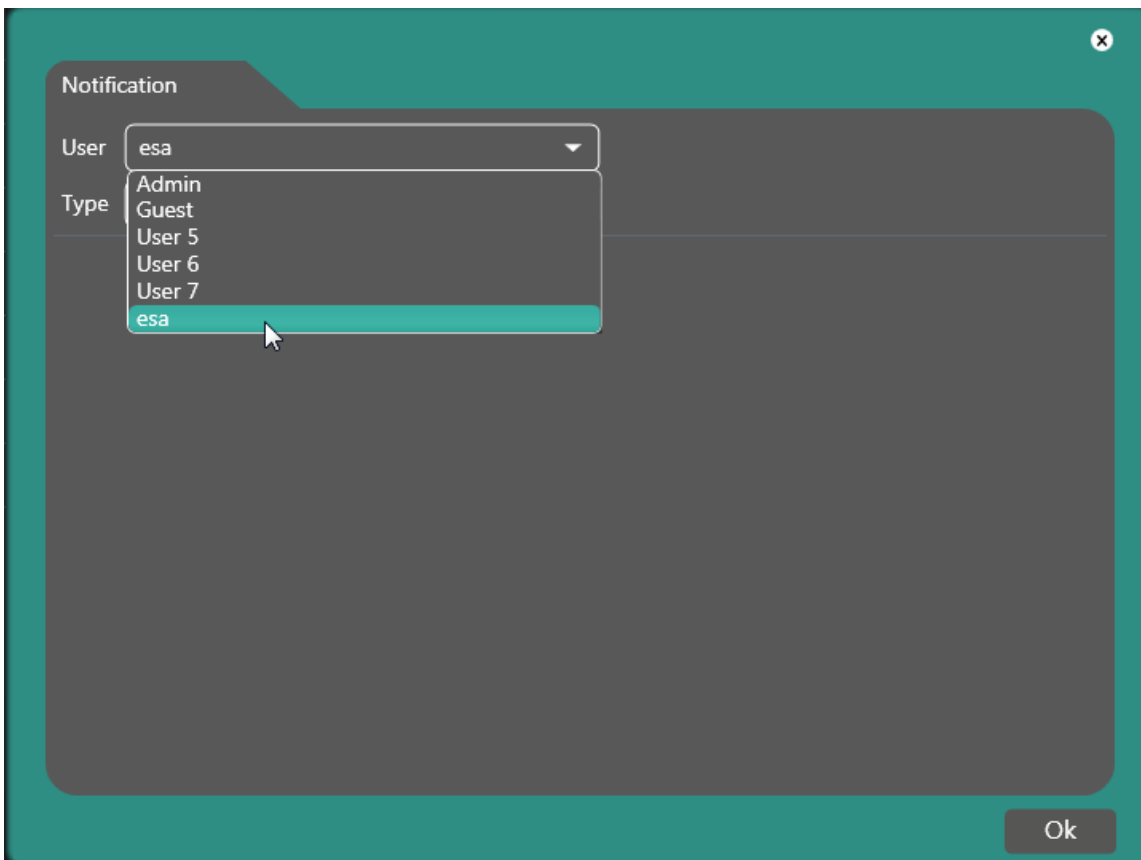


# CREW Manual

Click the “Details” key.

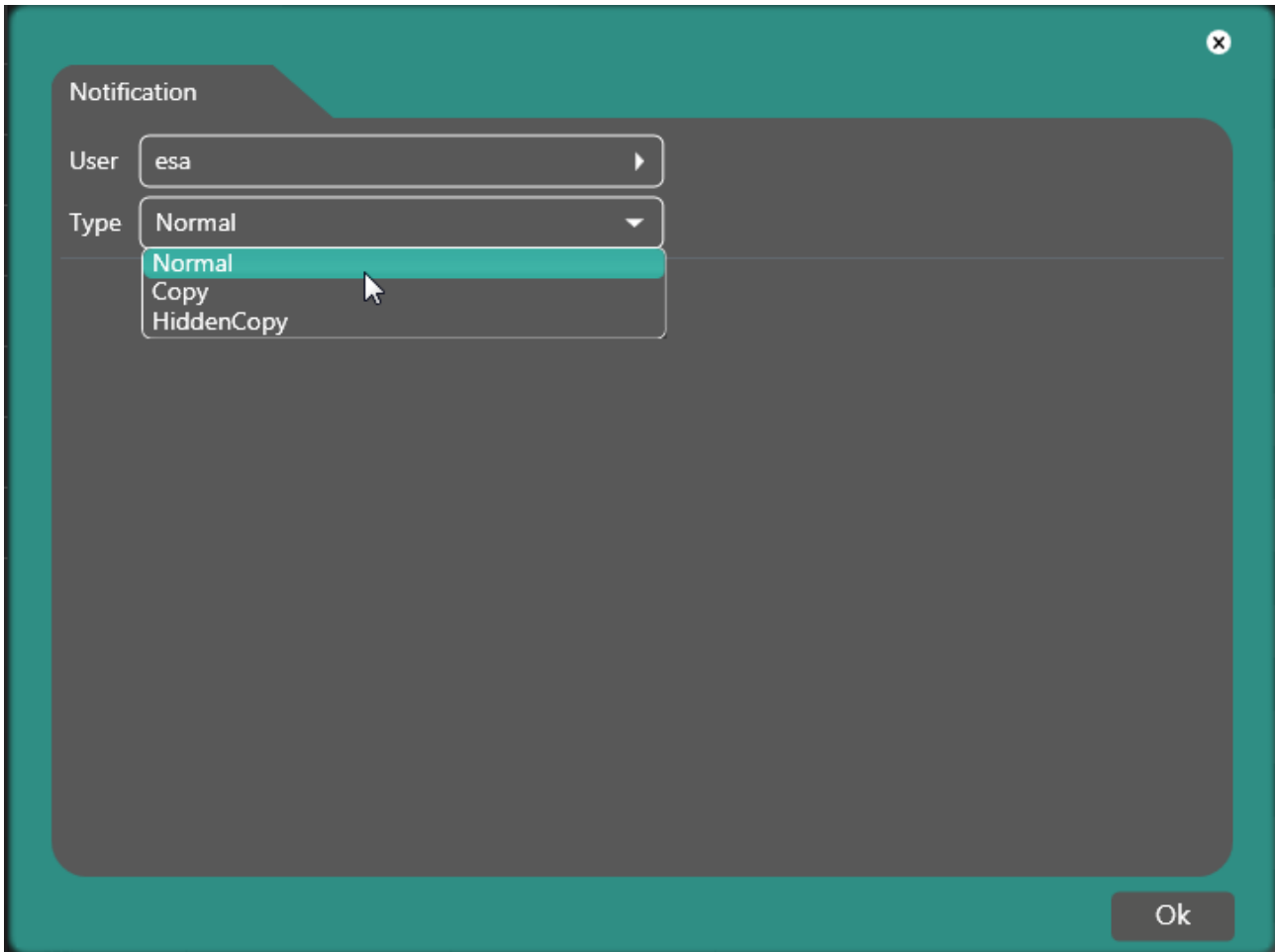


For example, you can select the “esa” user already in the project.



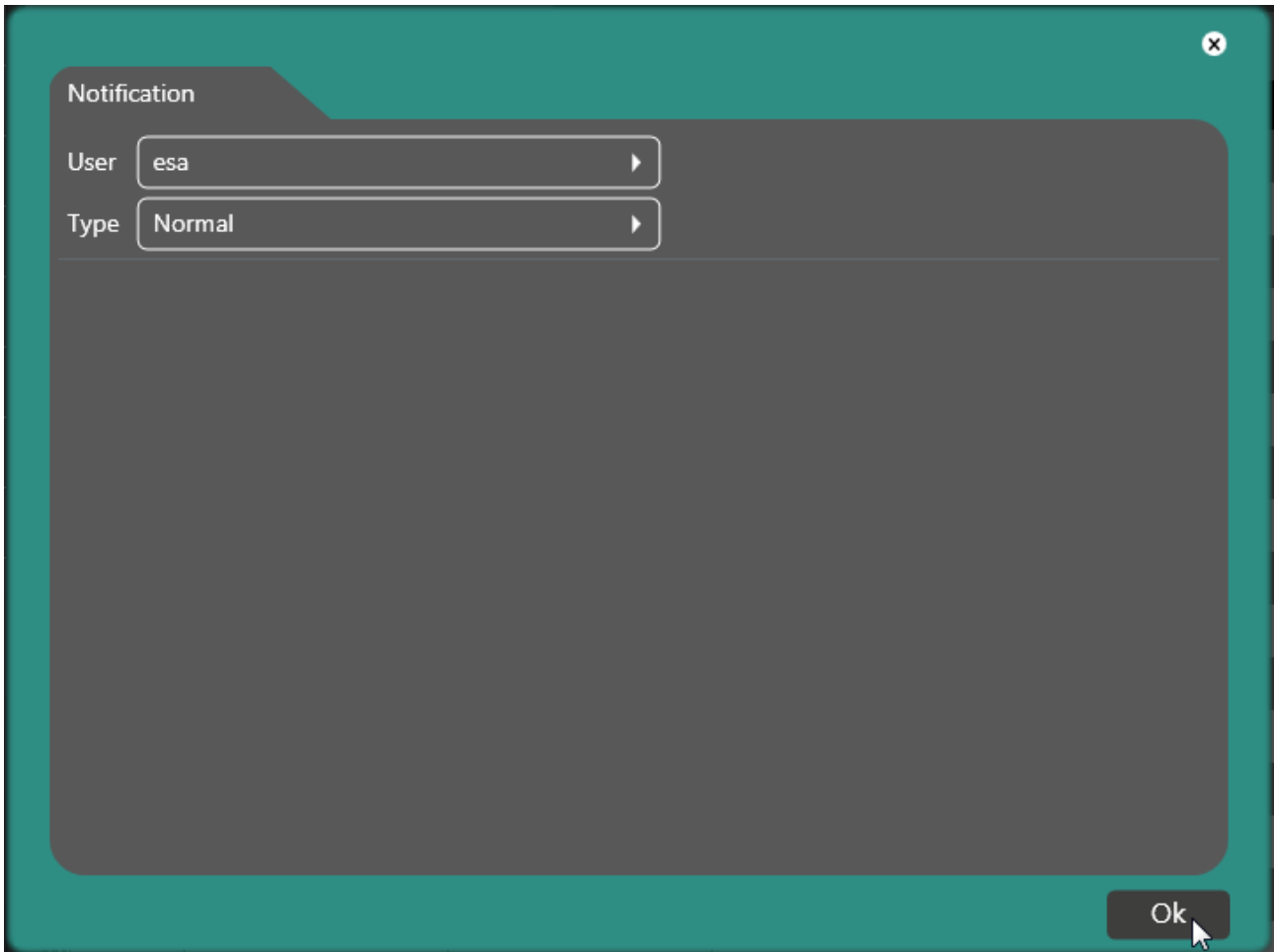
# CREW Manual

Select what type of notification to send to this user (normal, in carbon copy or blind carbon copy).



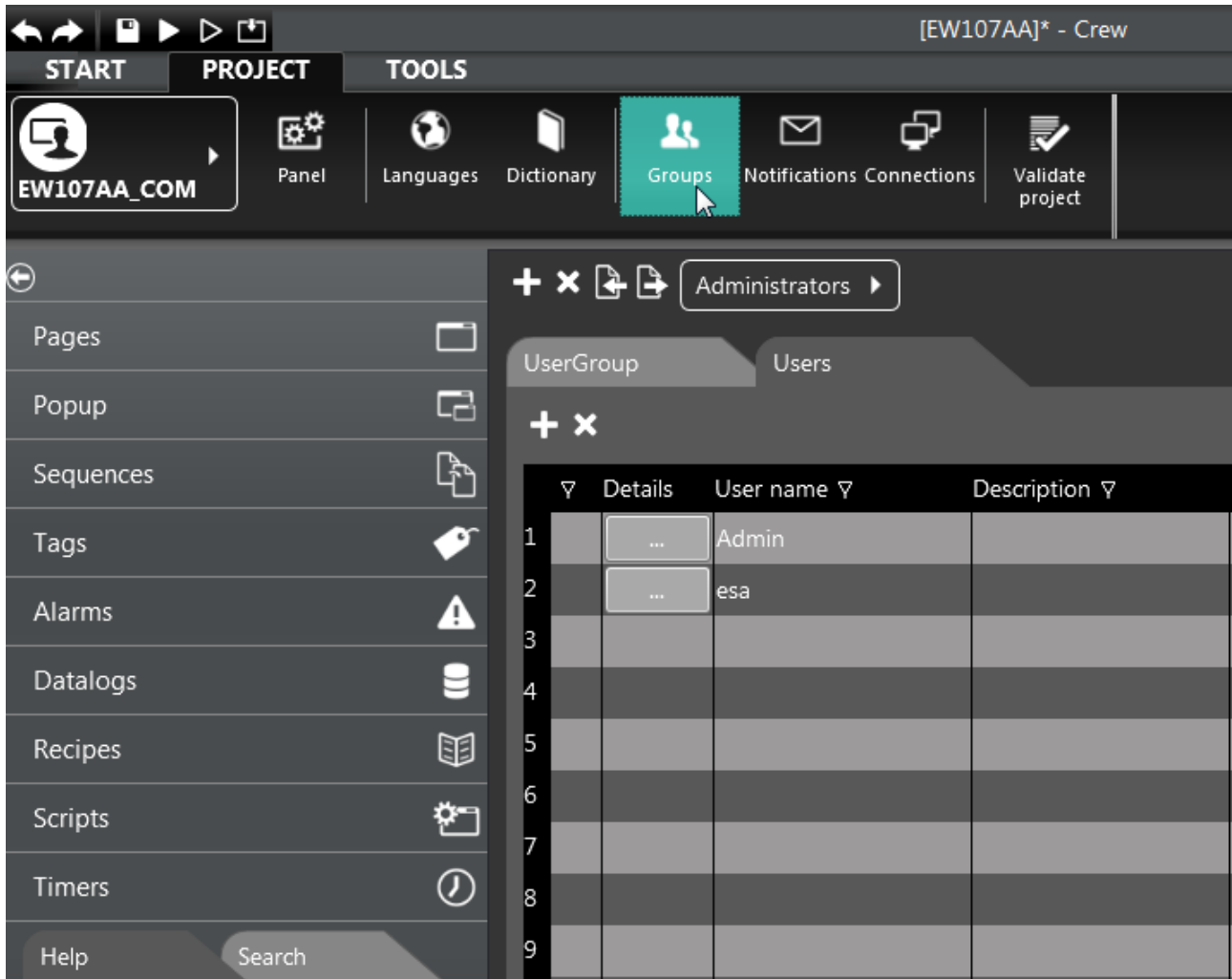
# CREW Manual

Click “Ok” to confirm.



# CREW Manual

You now need to enter the "esa" user's email address. Click "Groups".

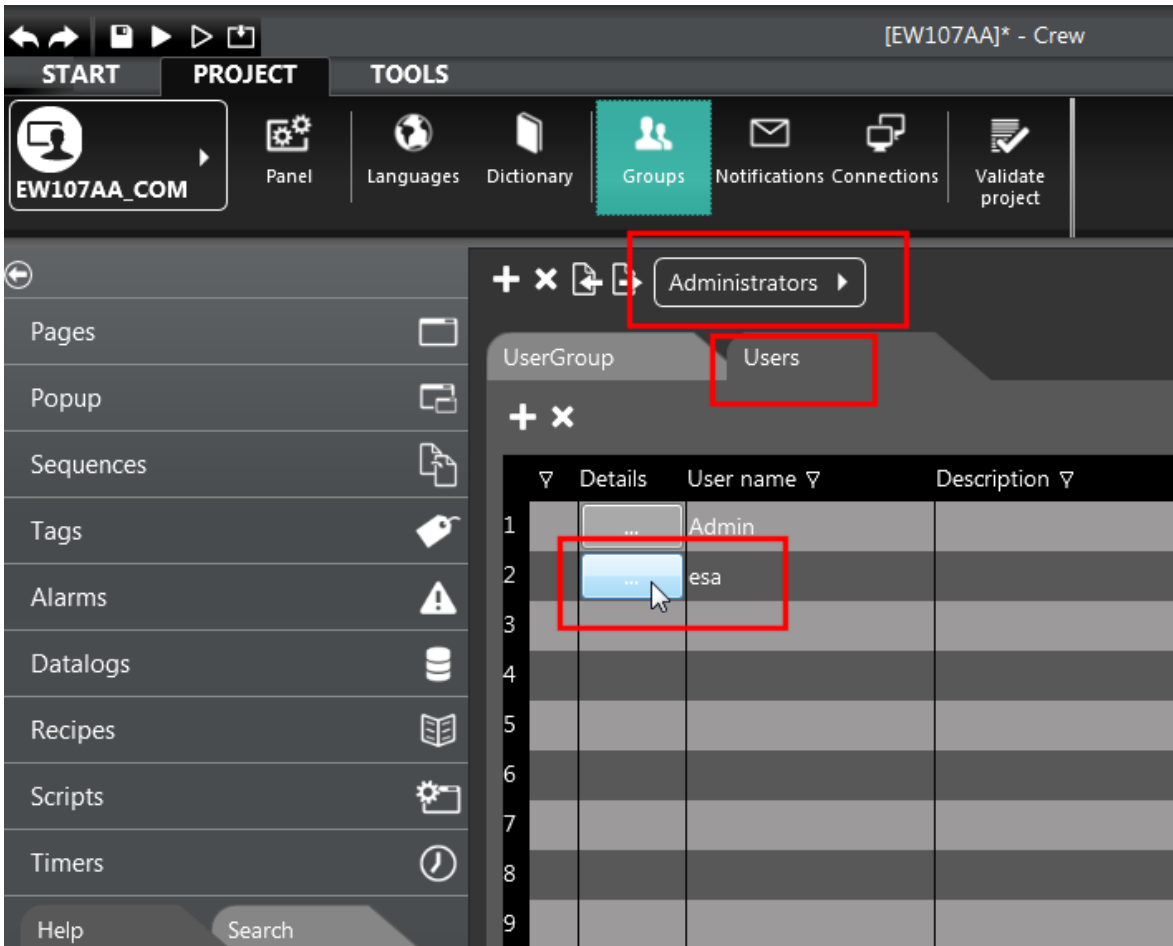


The screenshot shows the CREW software interface. The top navigation bar includes 'START', 'PROJECT', and 'TOOLS'. The 'TOOLS' menu is open, and the 'Groups' icon is highlighted. Below the menu, a sidebar lists various project components like Pages, Popup, Sequences, Tags, Alarms, Datalogs, Recipes, Scripts, and Timers. The main workspace displays the 'Administrators' window, which is divided into 'UserGroup' and 'Users' tabs. The 'Users' tab is active, showing a table with the following data:

	Details	User name	Description
1	...	Admin	
2	...	esa	
3			
4			
5			
6			
7			
8			
9			

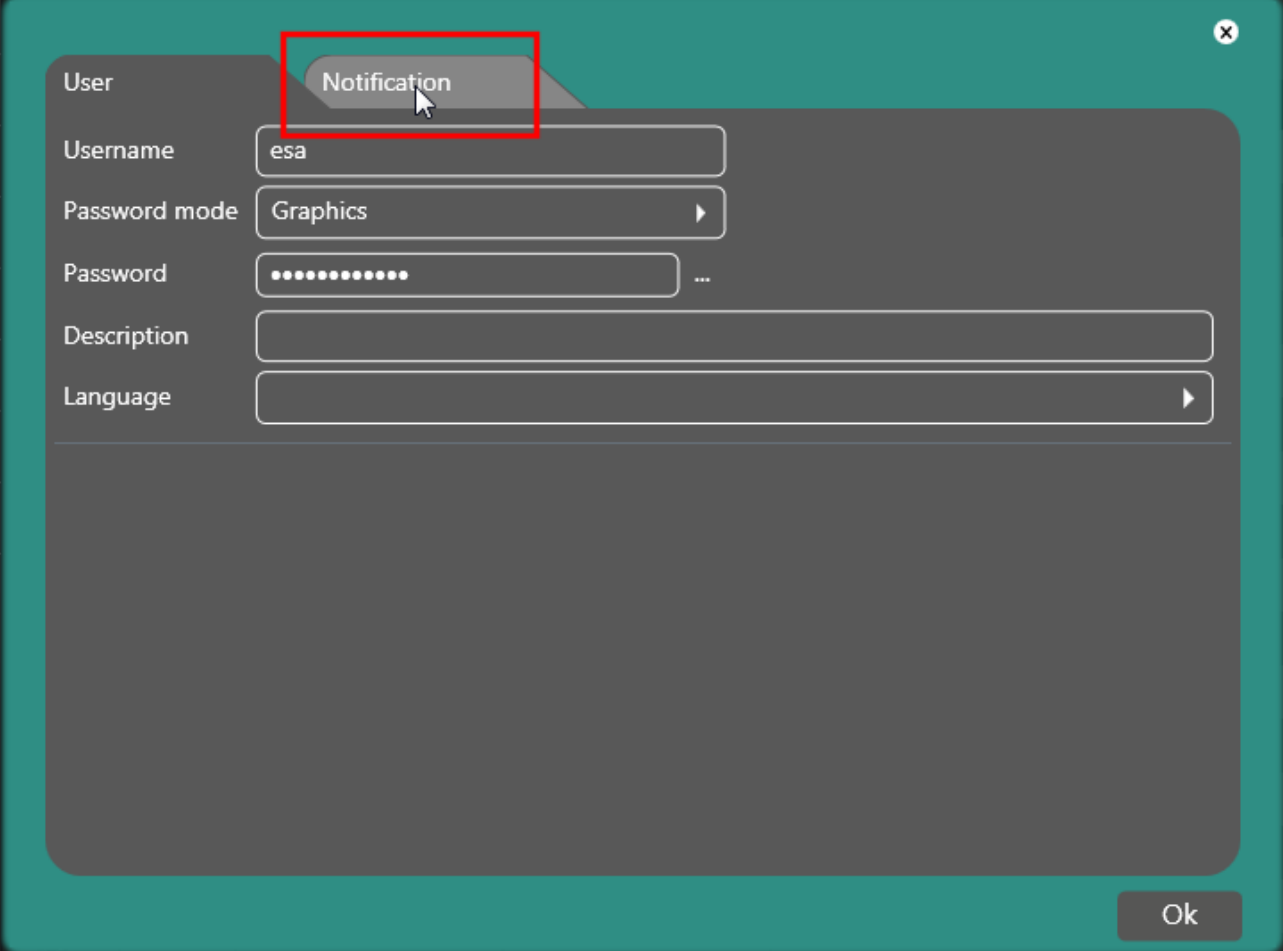
# CREW Manual

Then click on the “esa” user’s “Details” and select the “Administrator” group and the “Users” option.



# CREW Manual

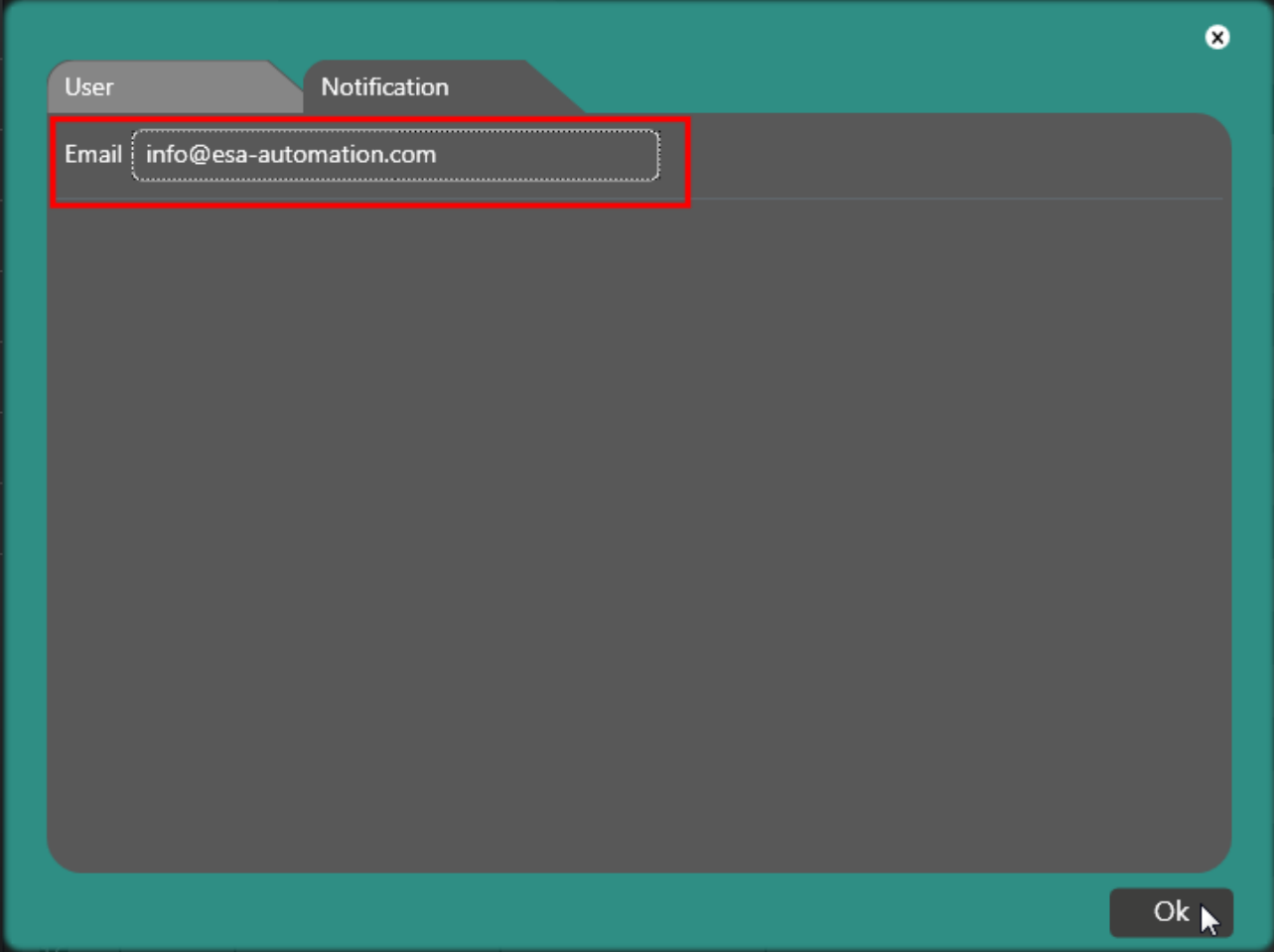
Select the “Notifications” option.



The image shows a software dialog box titled "User" with a close button (X) in the top right corner. The dialog contains several input fields: "Username" with the value "esa", "Password mode" set to "Graphics", "Password" (masked with dots), "Description", and "Language". A red rectangular box highlights the "Notification" option, which is currently selected. An "Ok" button is located at the bottom right of the dialog.

# CREW Manual

Enter the “esa” user’s email address.

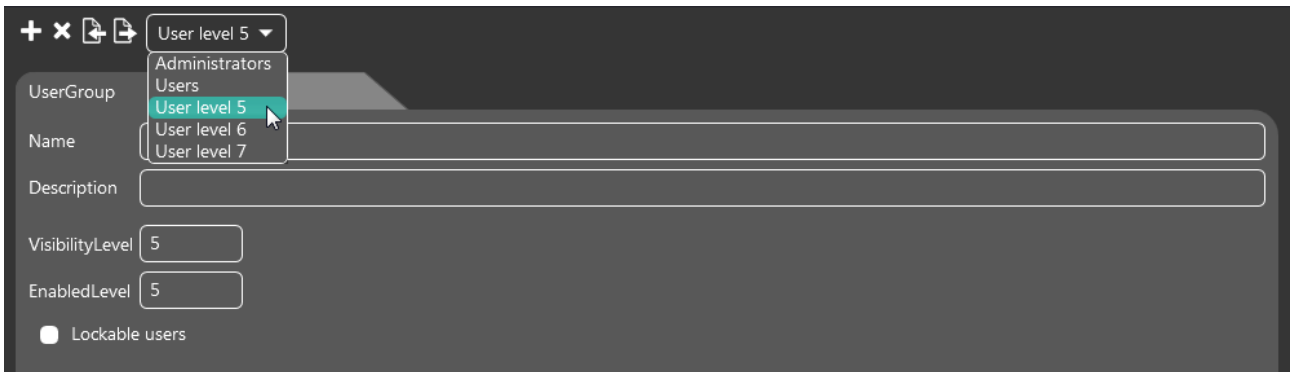


The image shows a screenshot of a software dialog box with a teal border. At the top, there are two tabs: "User" and "Notification". The "User" tab is selected. Below the tabs, there is a text input field labeled "Email" containing the text "info@esa-automation.com". This input field is highlighted with a red rectangular border. At the bottom right of the dialog, there is an "Ok" button with a mouse cursor icon. A close button (an 'x' in a circle) is located in the top right corner of the dialog.

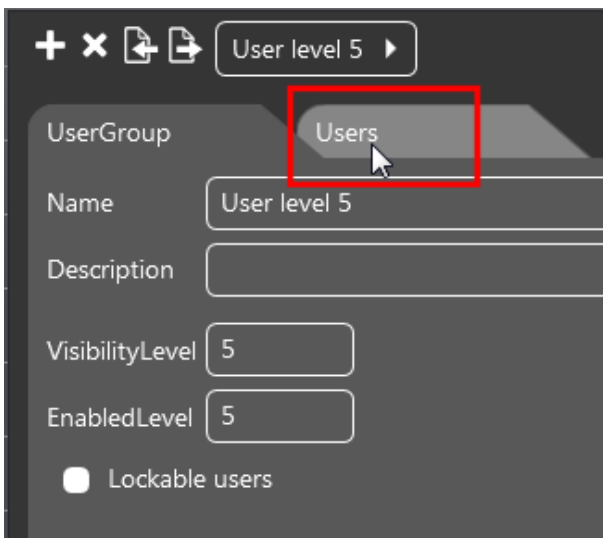


# CREW Manual

The “esa” user is now configured and enabled to receive emails. If you wish to enable another user (for example, a user belonging to the “User level 5” group, as shown in the image), simply select the group it belongs to.

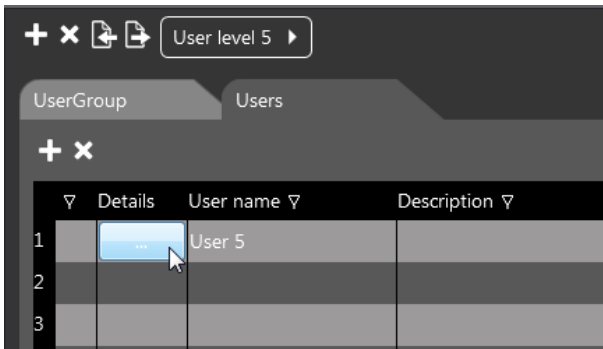


Click “Users”.

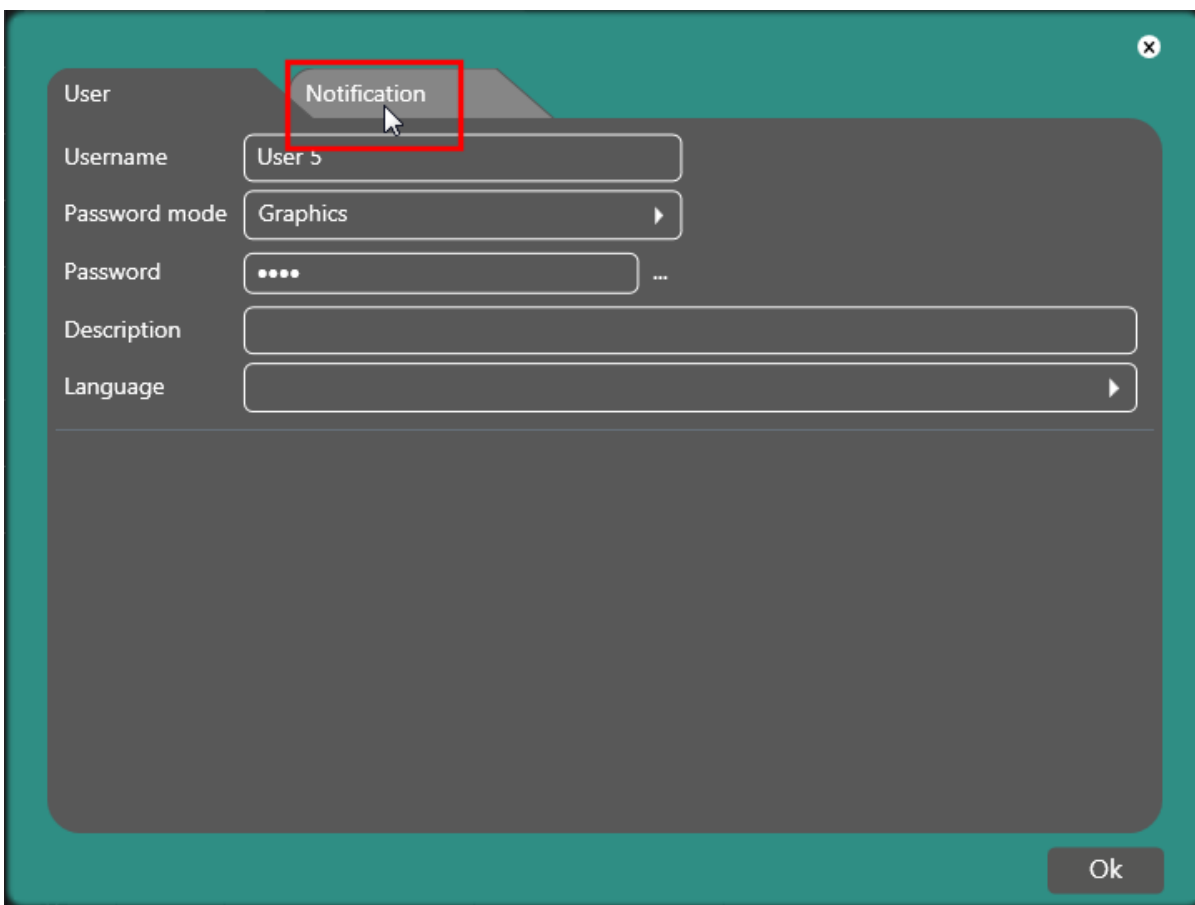


# CREW Manual

Select “User 5” and click the “Details” key.

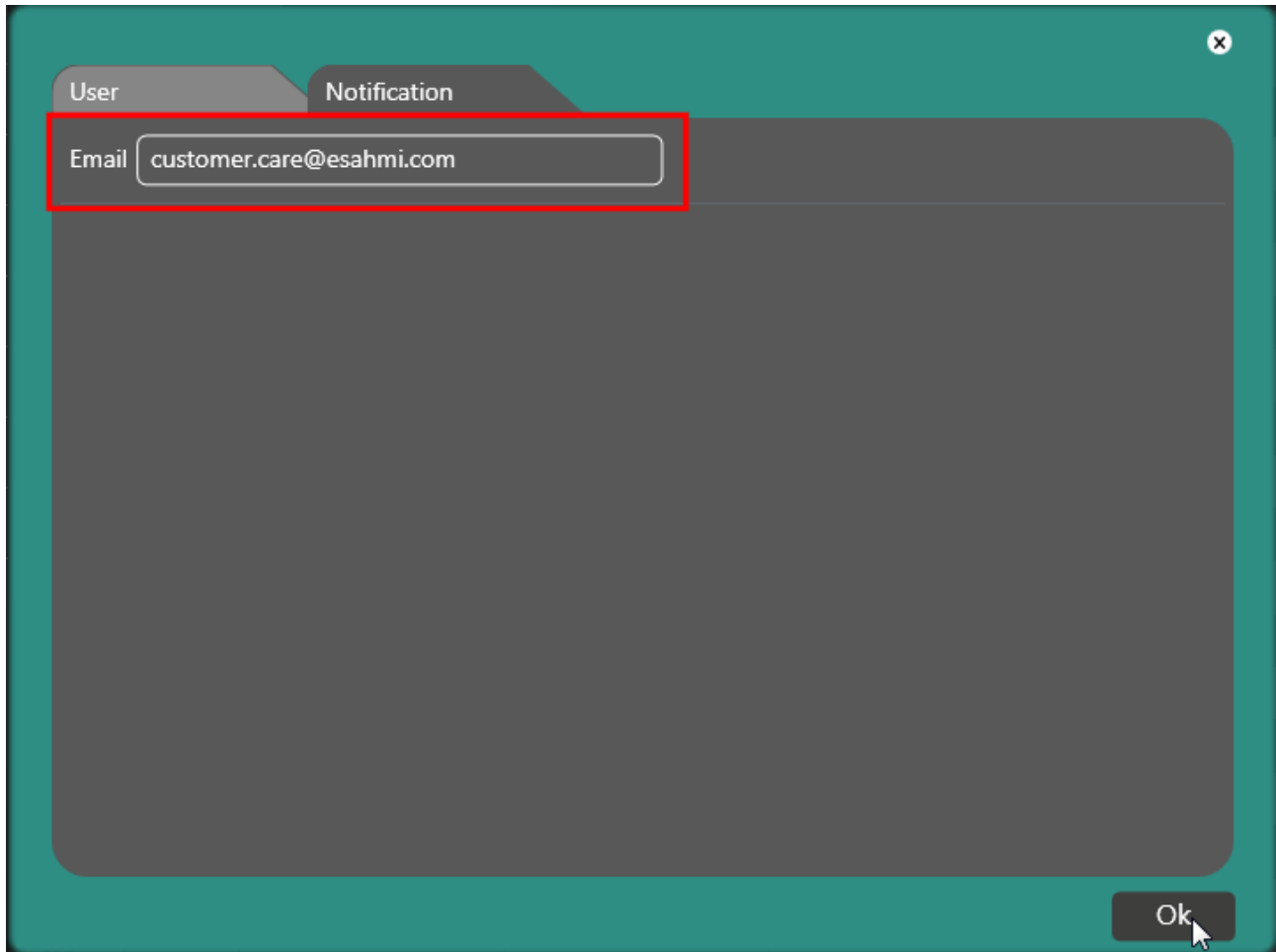


Click “Notifications”:



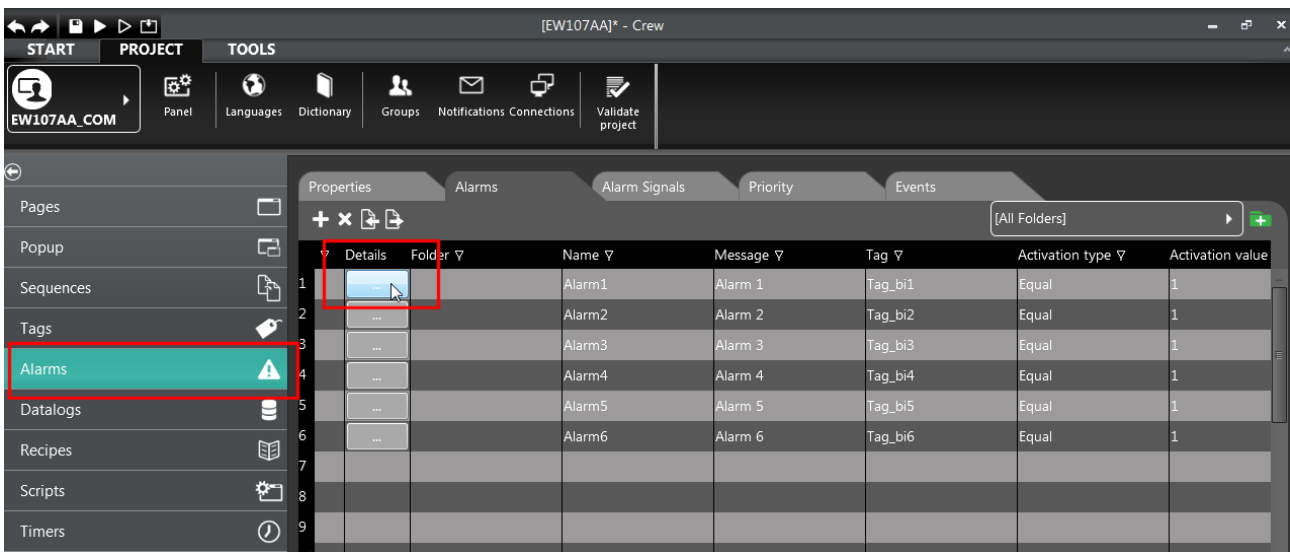
# CREW Manual

and enter the “User 5” user’s email address:



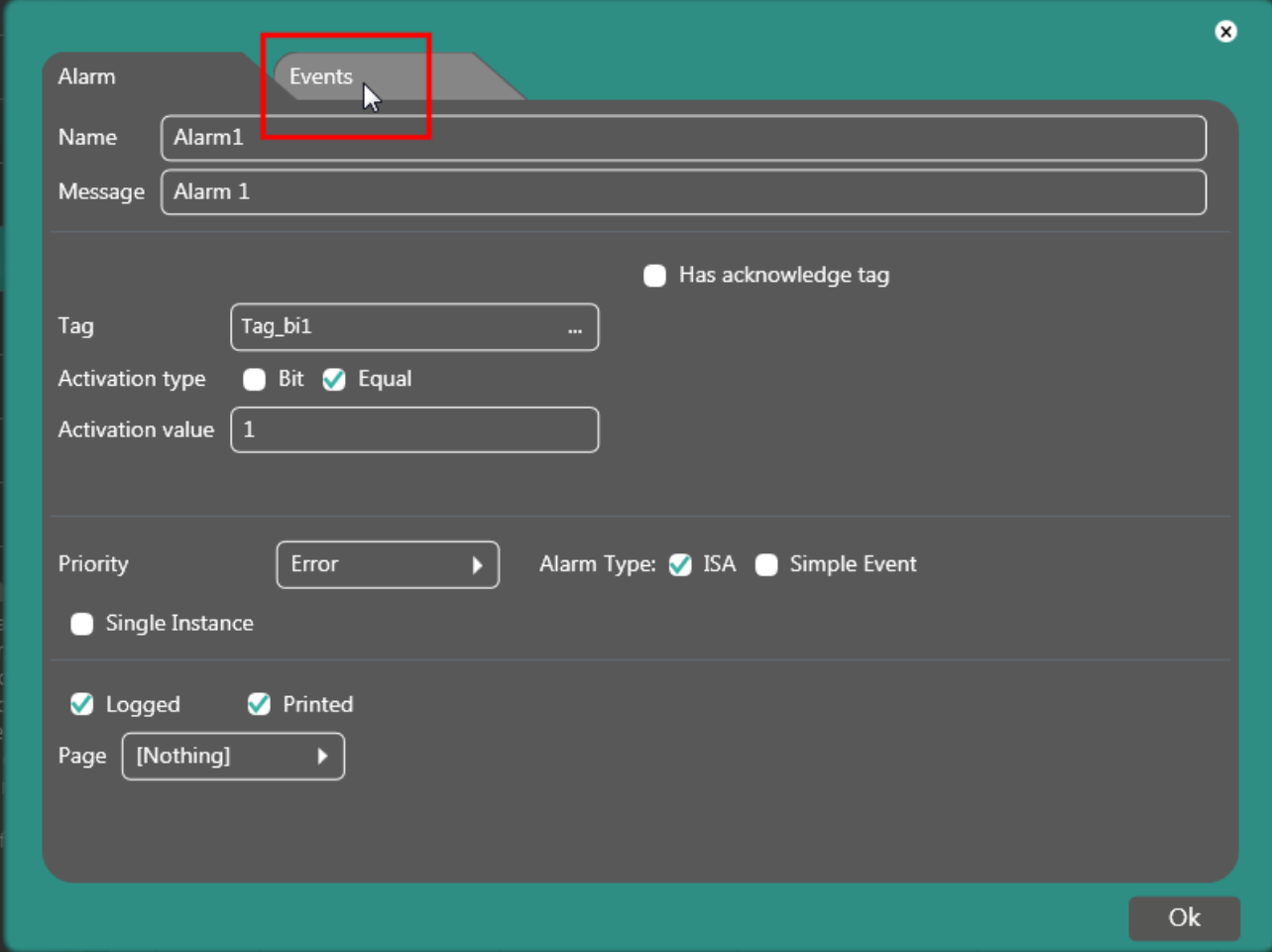
# CREW Manual

We have now associated 2 users (one belonging to the "Administrators" group and another belonging to the user level 5 group) enabling them to receive notification messages (in this case an email), we now have to establish when this message needs to be sent; from "Explore Project" we select the "Alarms" option and click "Details" for "Alarm1":



# CREW Manual

Click "Events":



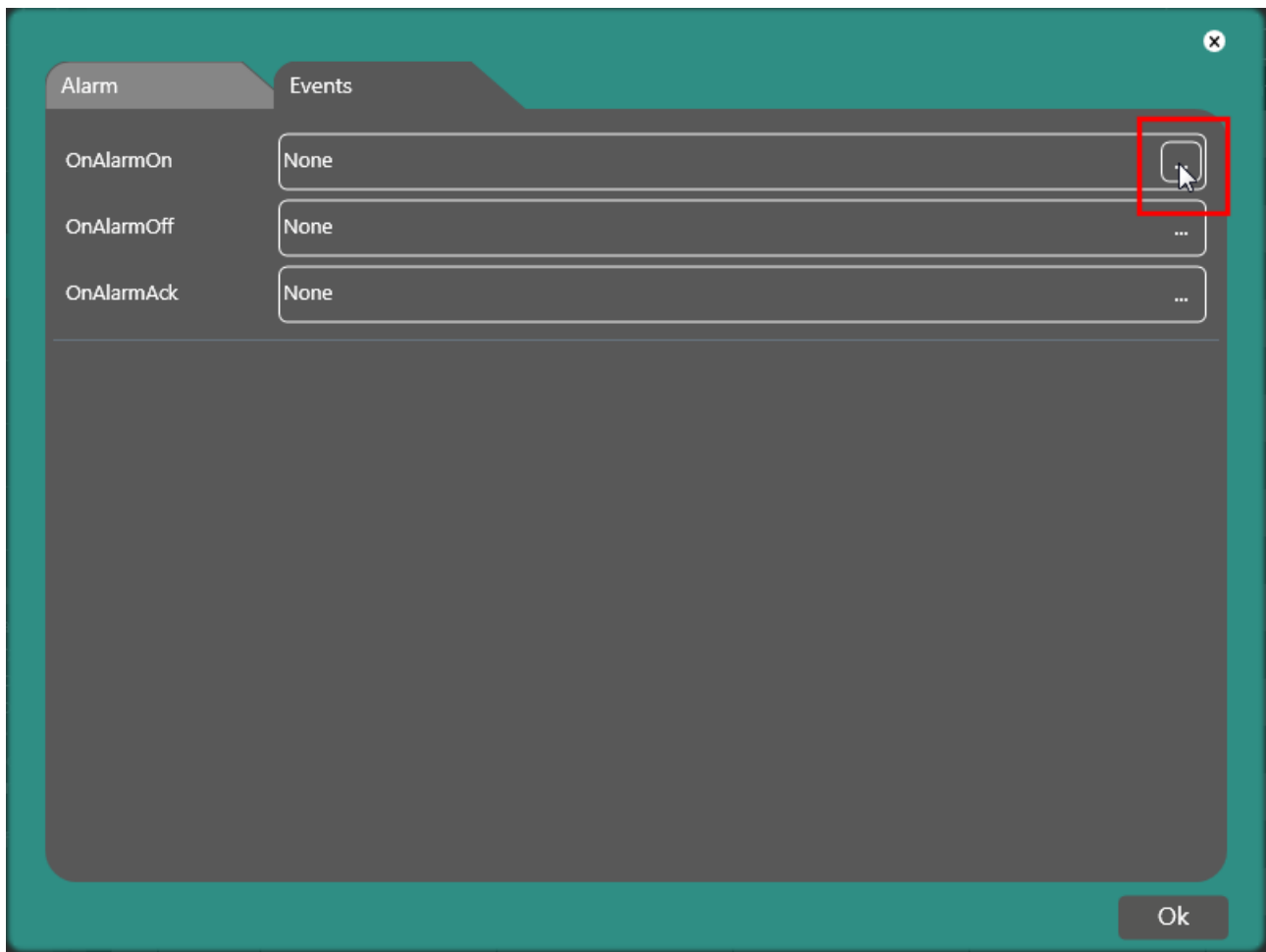
The screenshot shows a configuration window for an alarm. The window has a teal header with a close button (X) in the top right corner. Below the header, there are two tabs: "Alarm" and "Events". The "Events" tab is selected and highlighted with a red rectangular box. A mouse cursor is pointing at the "Events" tab. The main content area is dark grey and contains the following fields and options:

- Name:** Alarm1
- Message:** Alarm 1
- Has acknowledge tag:**
- Tag:** Tag\_bi1
- Activation type:**  Bit  Equal
- Activation value:** 1
- Priority:** Error
- Alarm Type:**  ISA  Simple Event
- Single Instance:**
- Logged:**  **Printed:**
- Page:** [Nothing]

An "Ok" button is located at the bottom right of the dialog box.

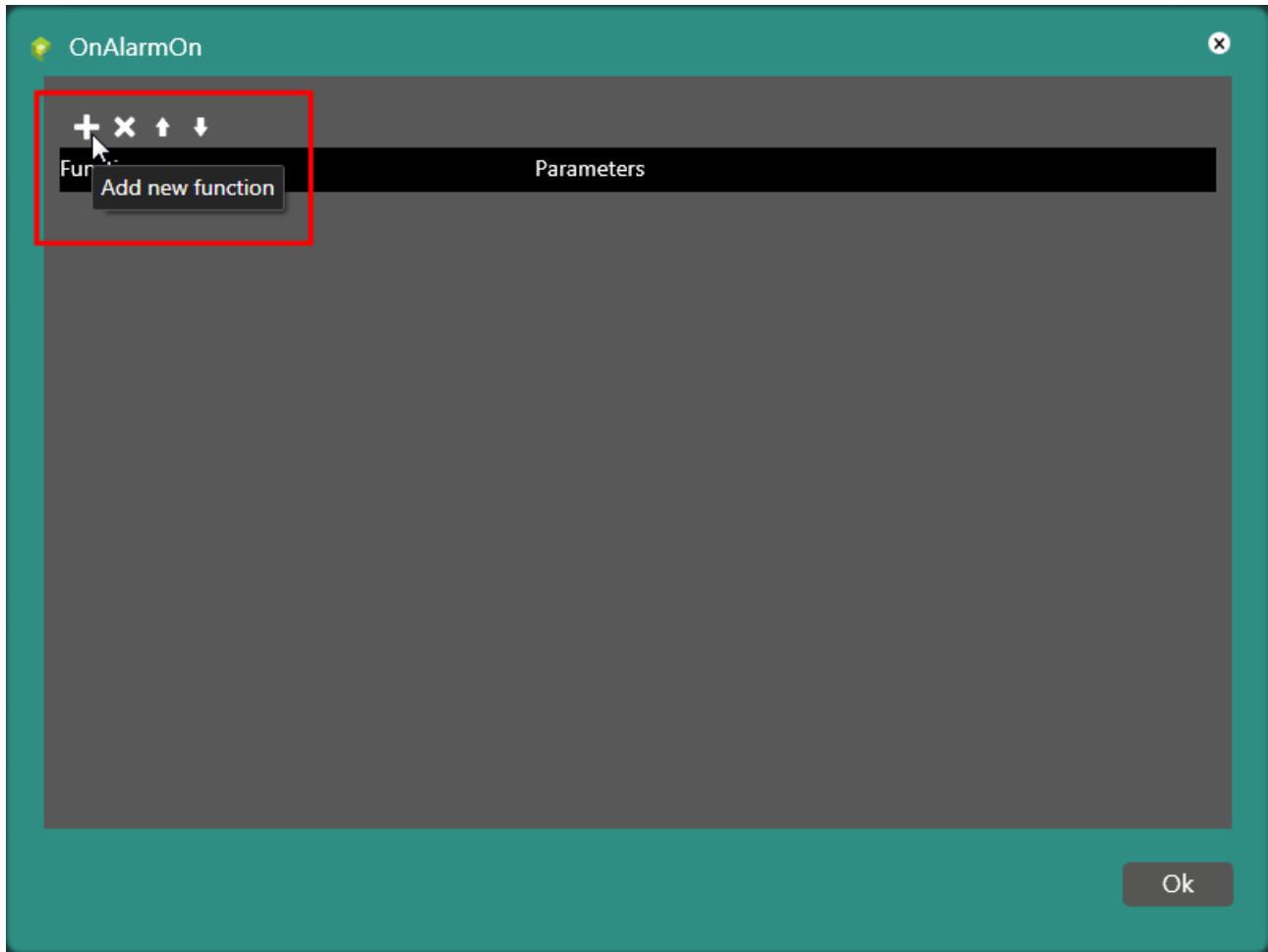
# CREW Manual

We decide to associate the predefined function that makes it possible to send emails in the "OnAllarmeOn/OnAlarmOn" alarm condition, namely when the alarm occurs; Click "Browse".



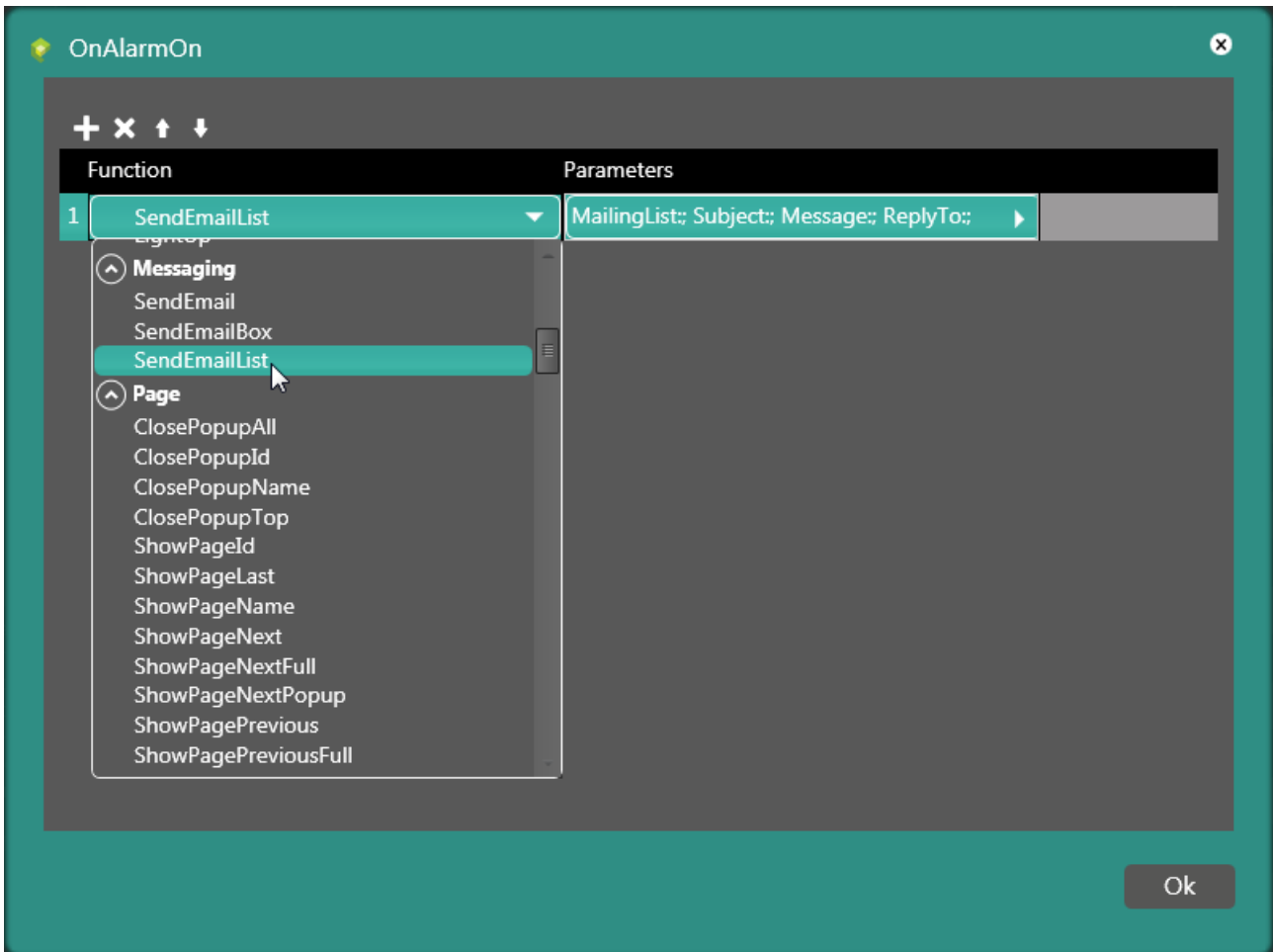
# CREW Manual

Click the “Add new function” key:



# CREW Manual

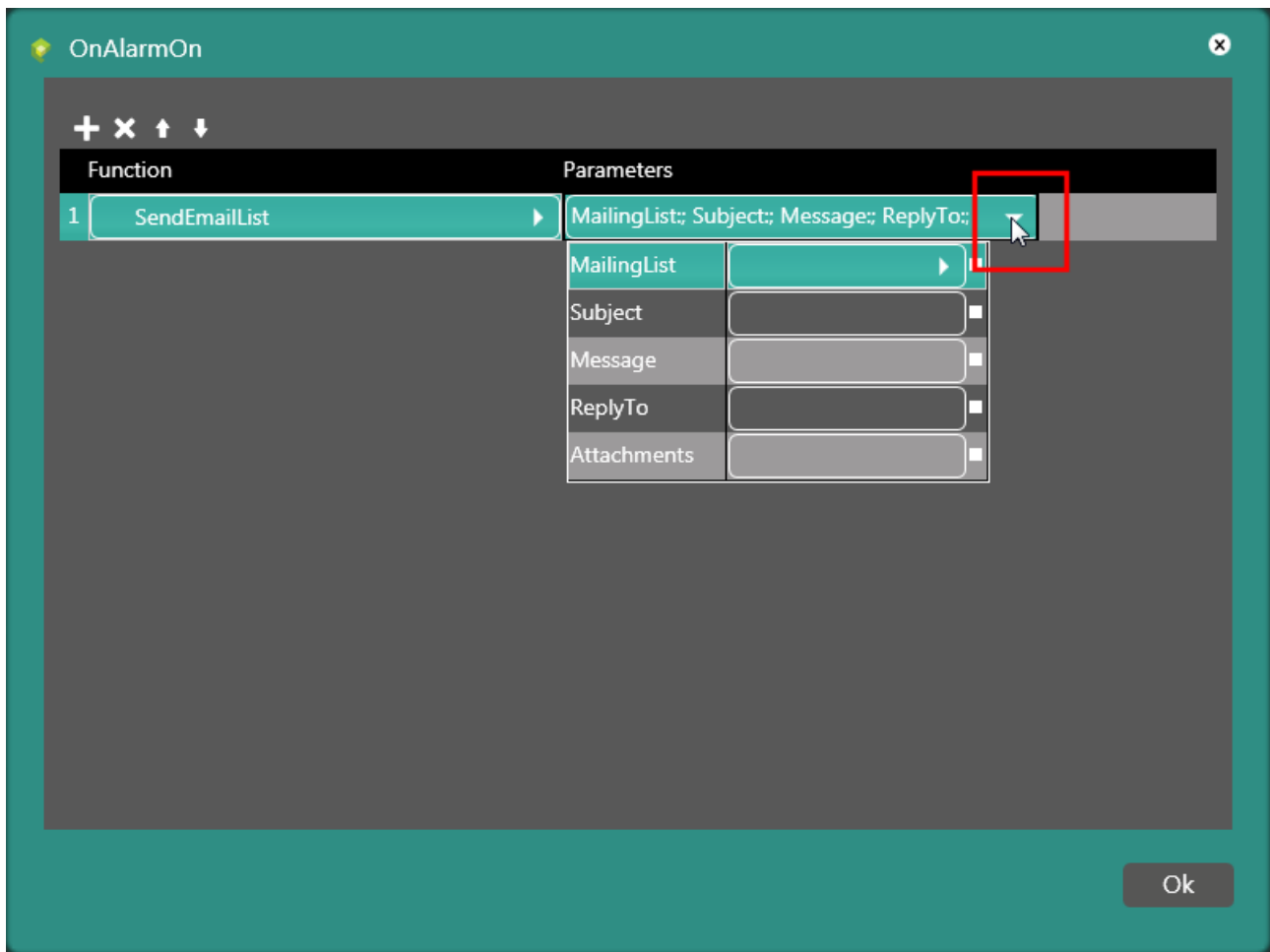
Choose the "SendEmailList" function by selecting it from the "Messaging" group of the predefined functions group (see "[Functions relative to Messages](#)" in the "[Predefined functions](#)" section):





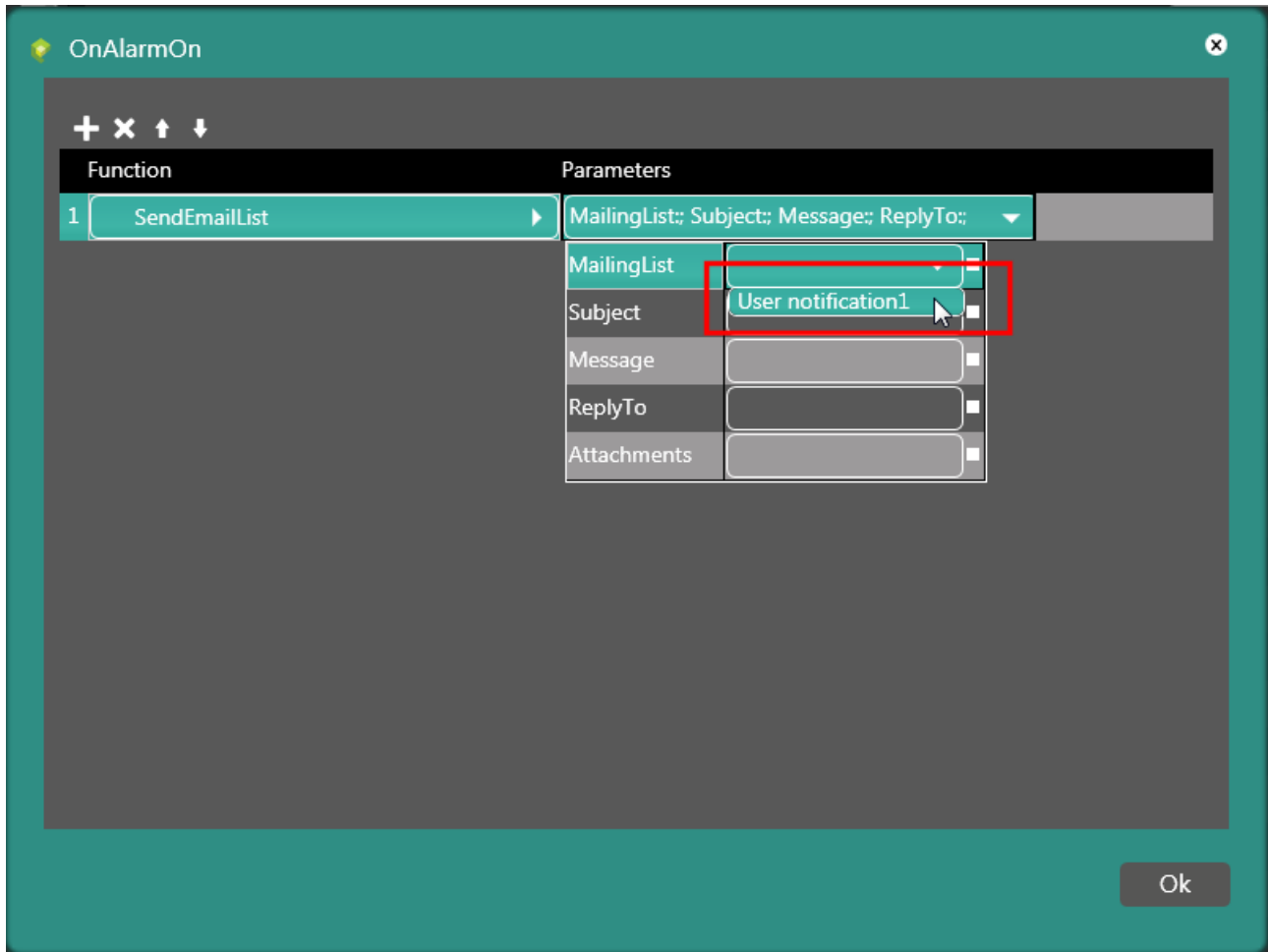
# CREW Manual

The "SendEmailList" function makes it possible to send every user in the group (in our case the group is called "User notification1") an email when the alarm happens (in our case "Alarm1"; click on the drop down menu:



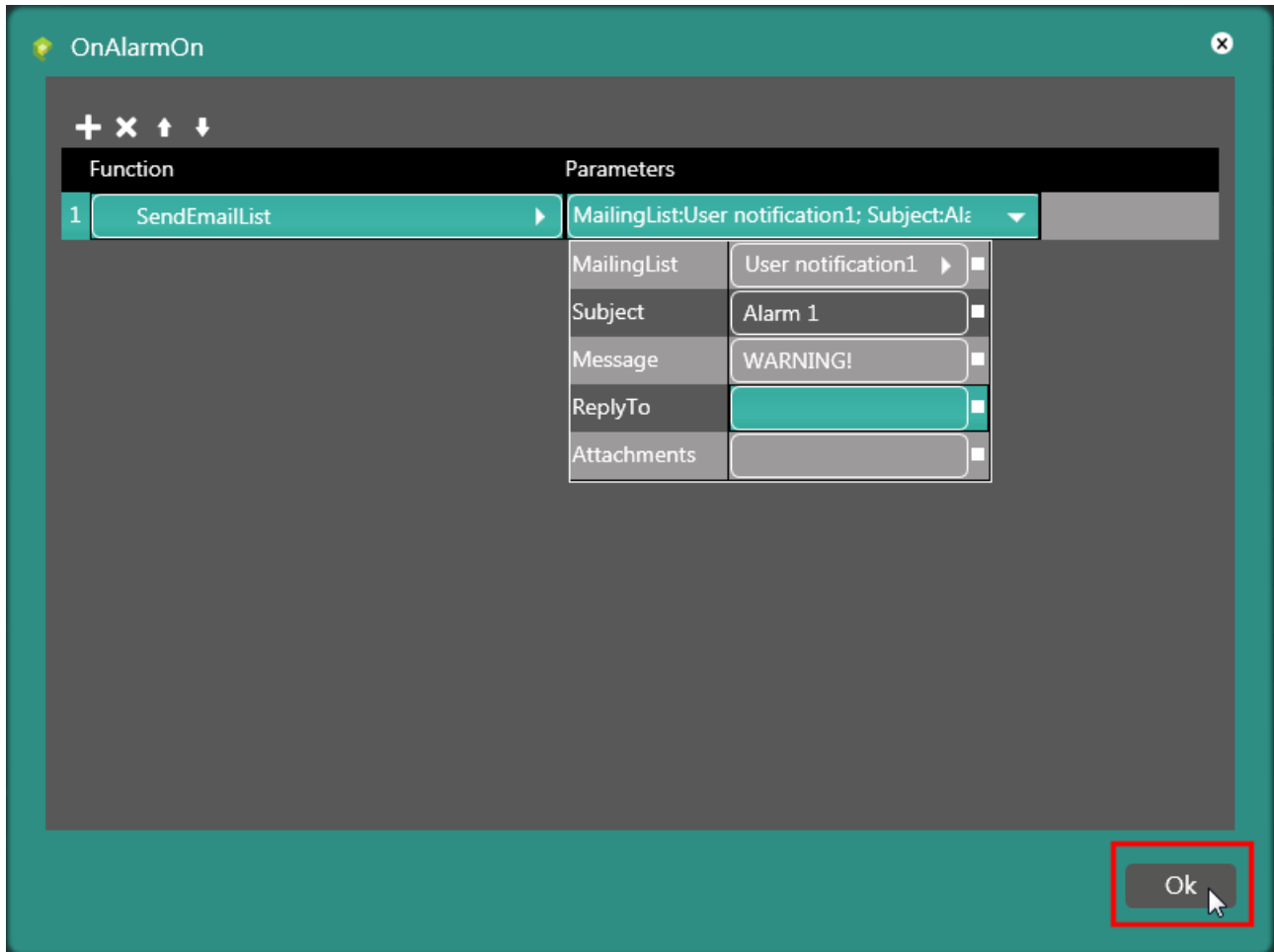
# CREW Manual

Select the notification group:



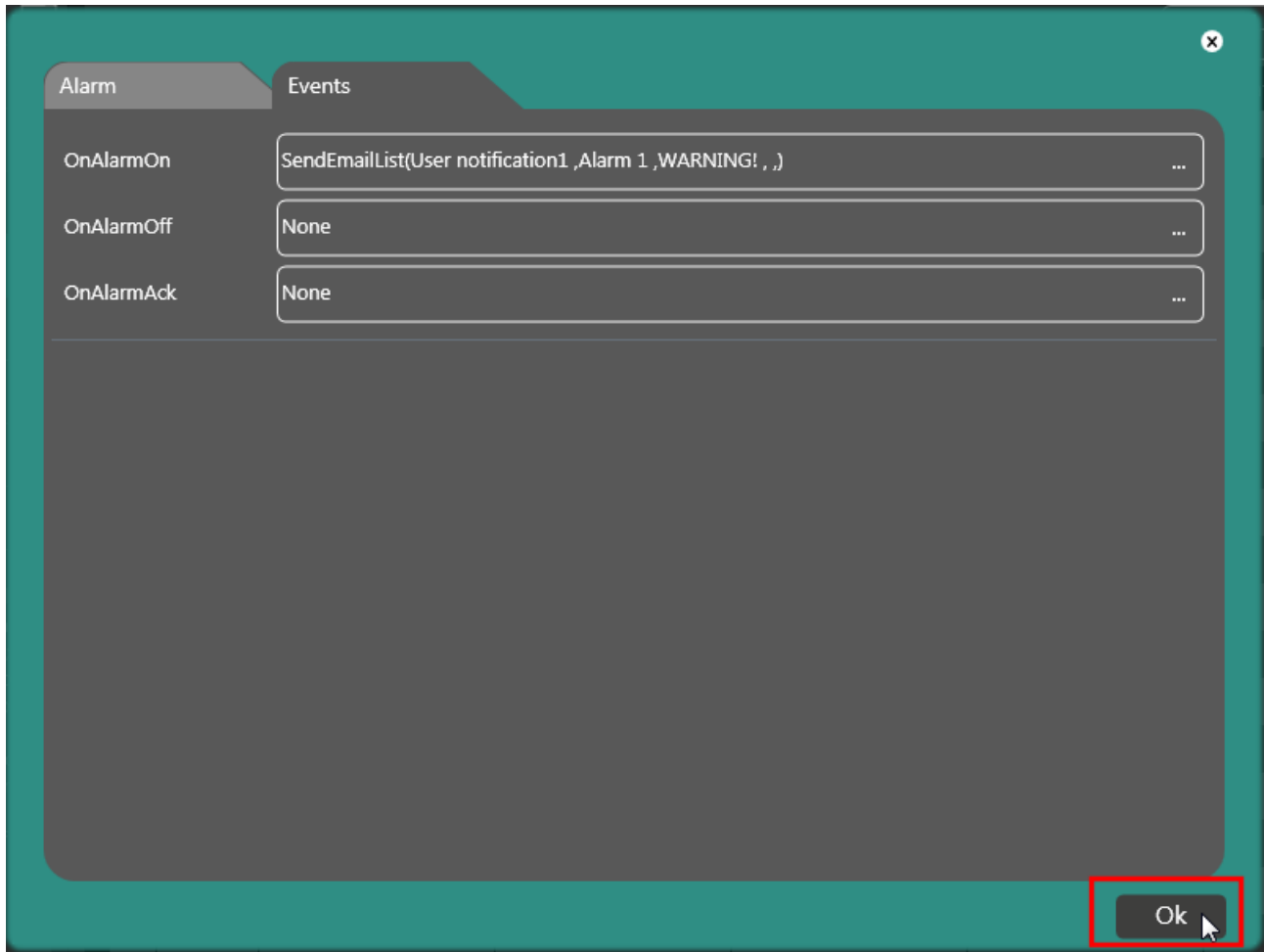
# CREW Manual

Enter the subject of the email and possibly a message:



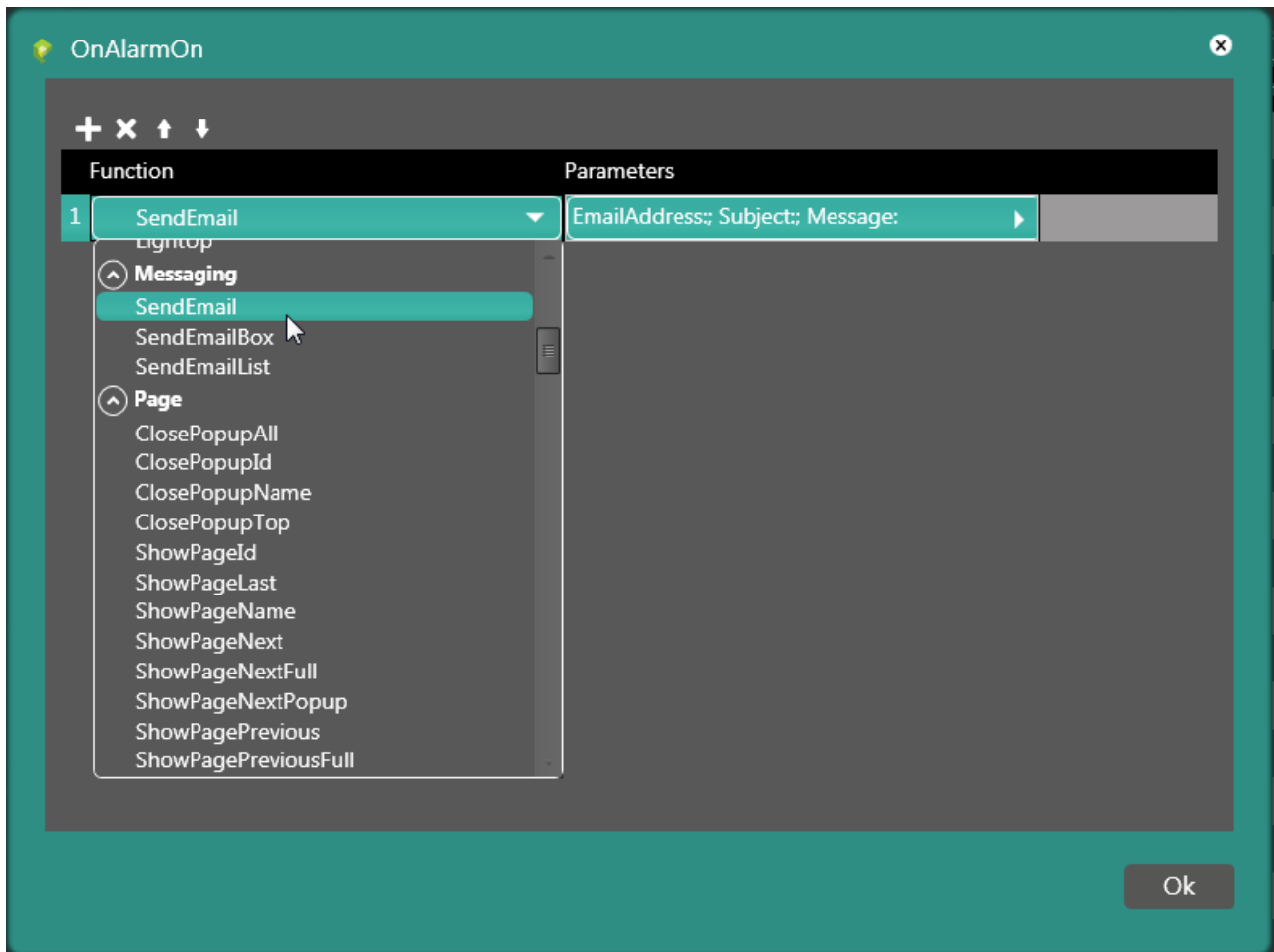
# CREW Manual

Confirm with “OK”:



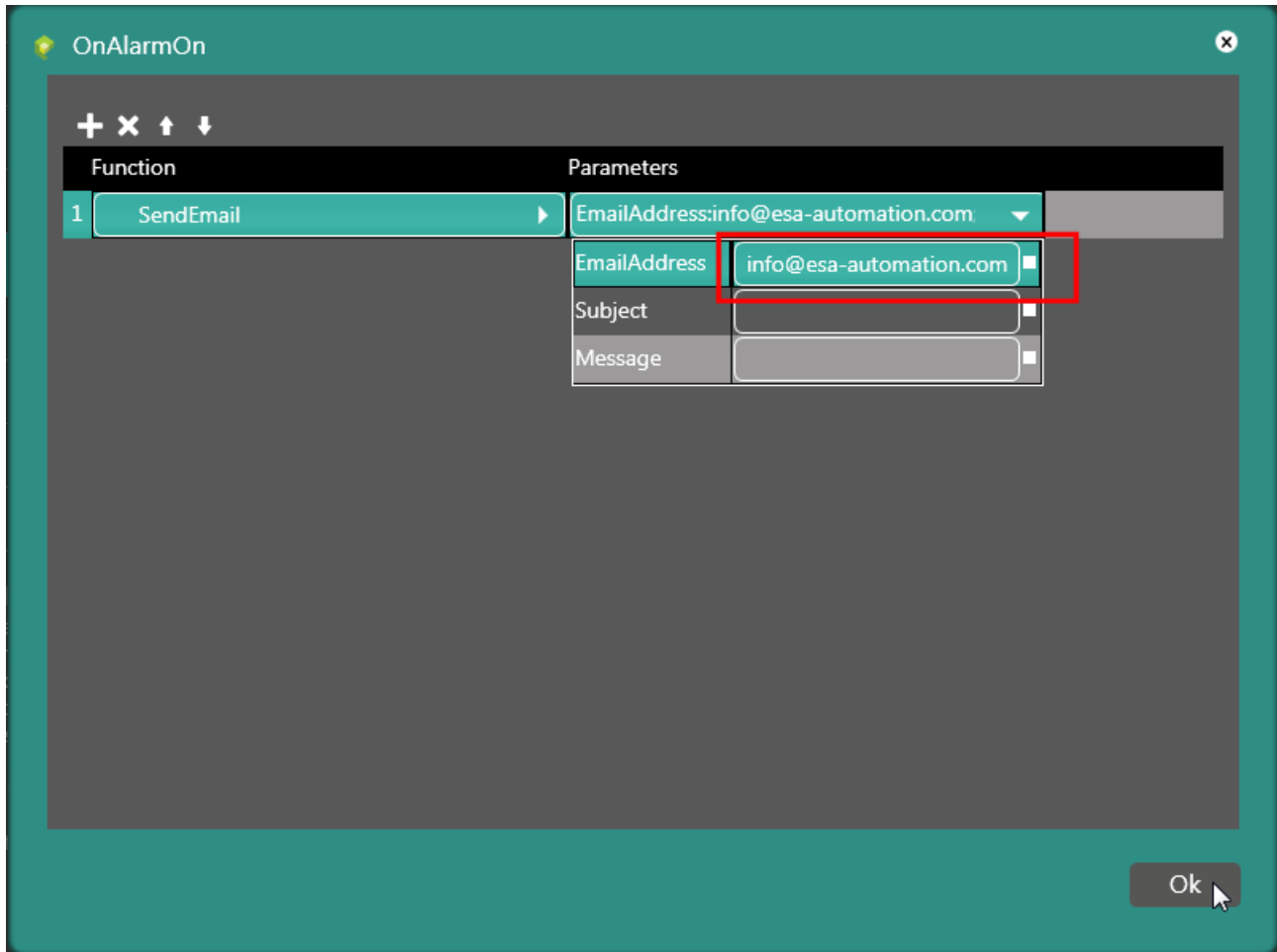
# CREW Manual

It is also possible to send an email to only one address by changing the type of function associated to the alarm; the function in this case is called “SendEmail”:



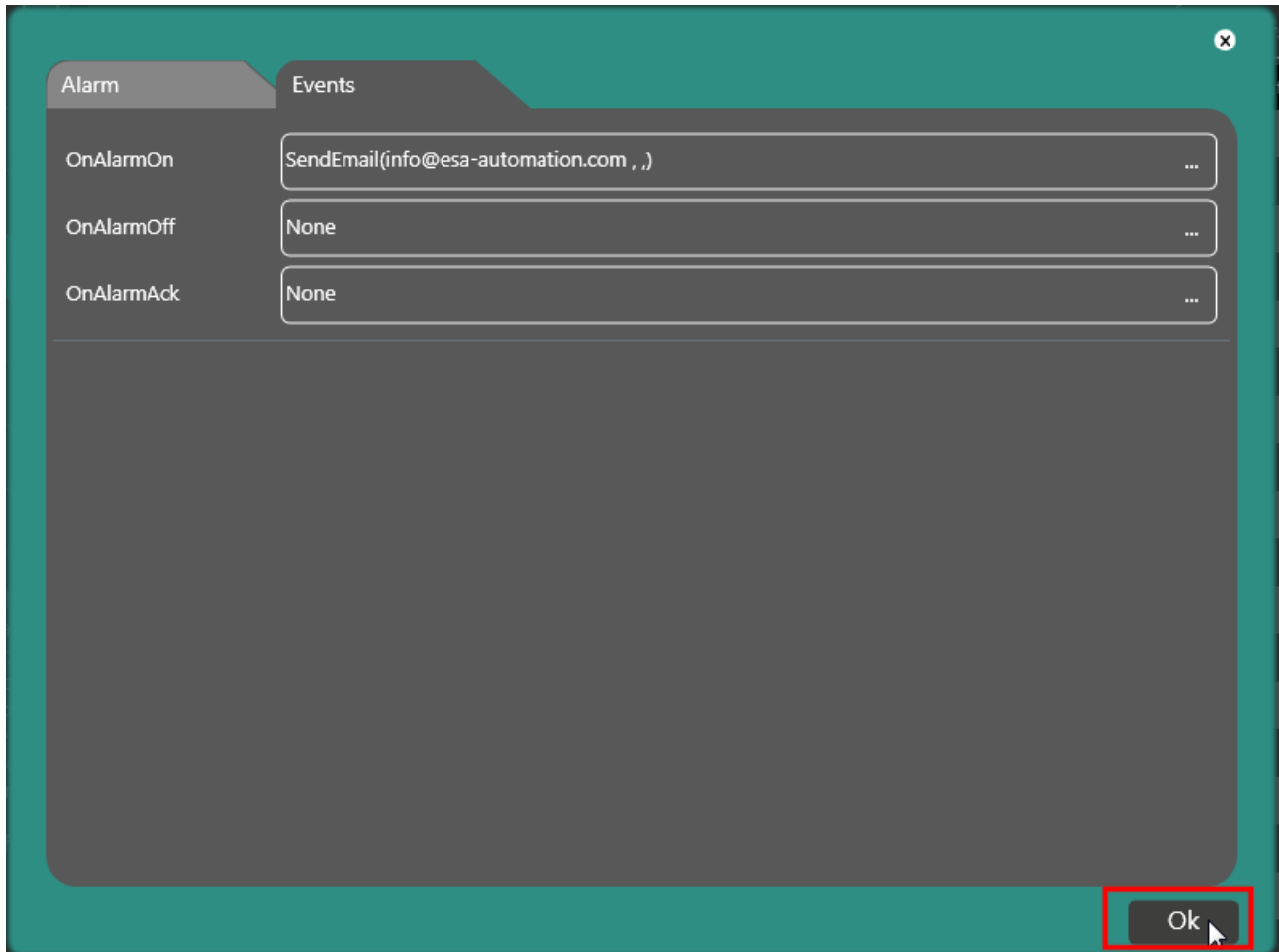
# CREW Manual

Enter the required email address:



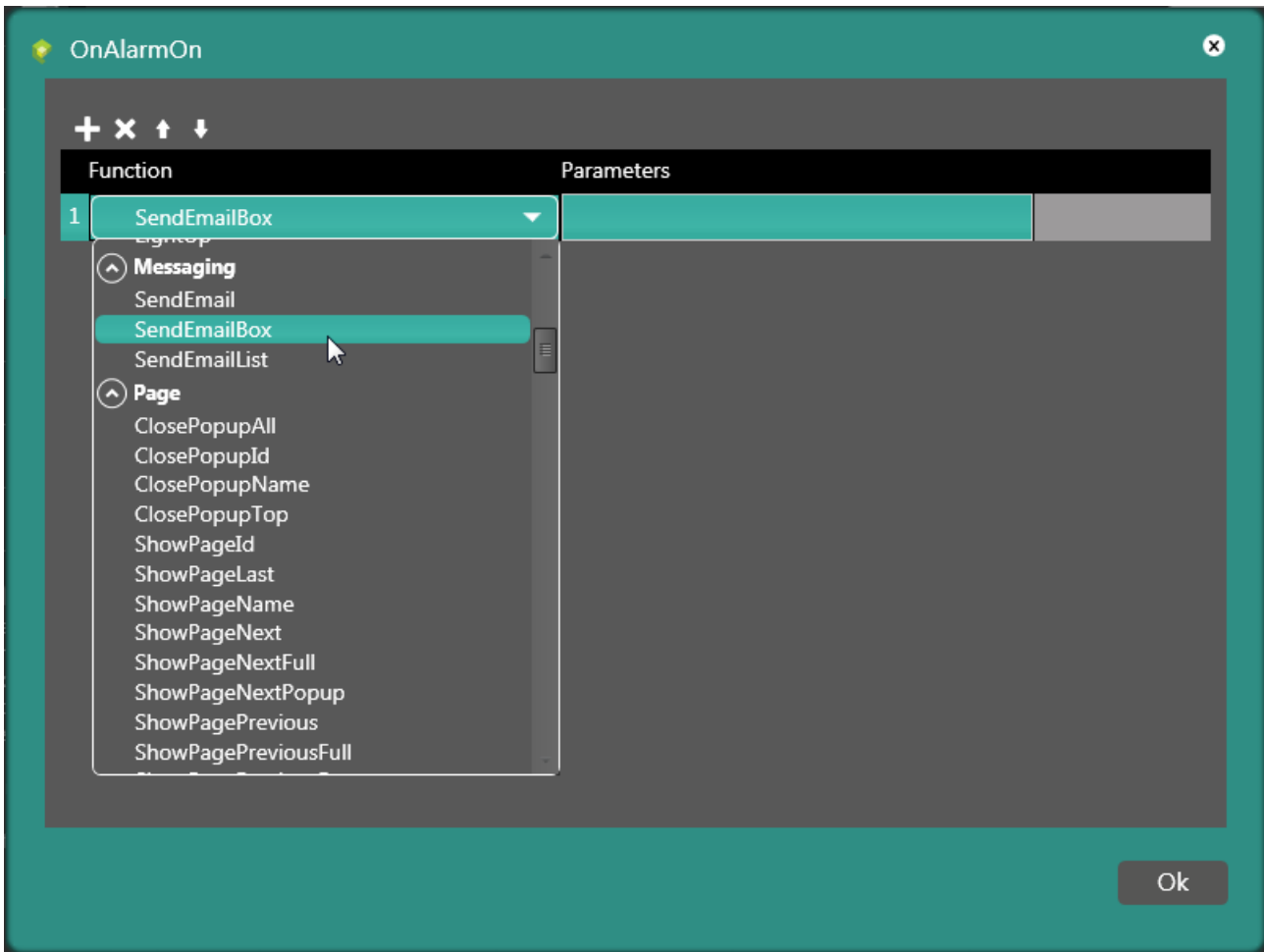
# CREW Manual

Confirm with “OK”:



# CREW Manual

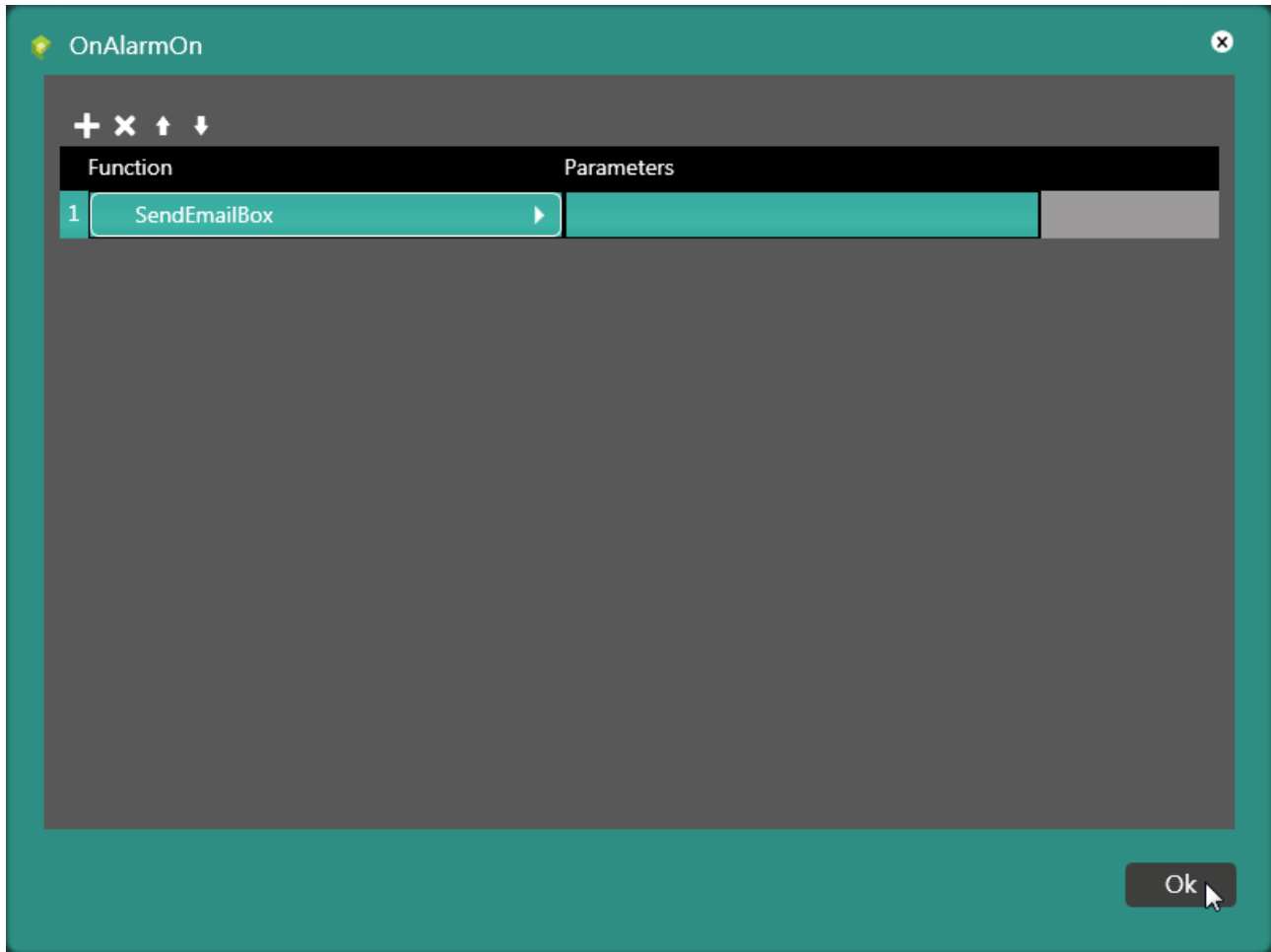
With the "SendEmailBox" function it is also possible to send an email manually from the dialog box that appears on Runtime when an alarm happens. Simply enter the data in the box that appears.





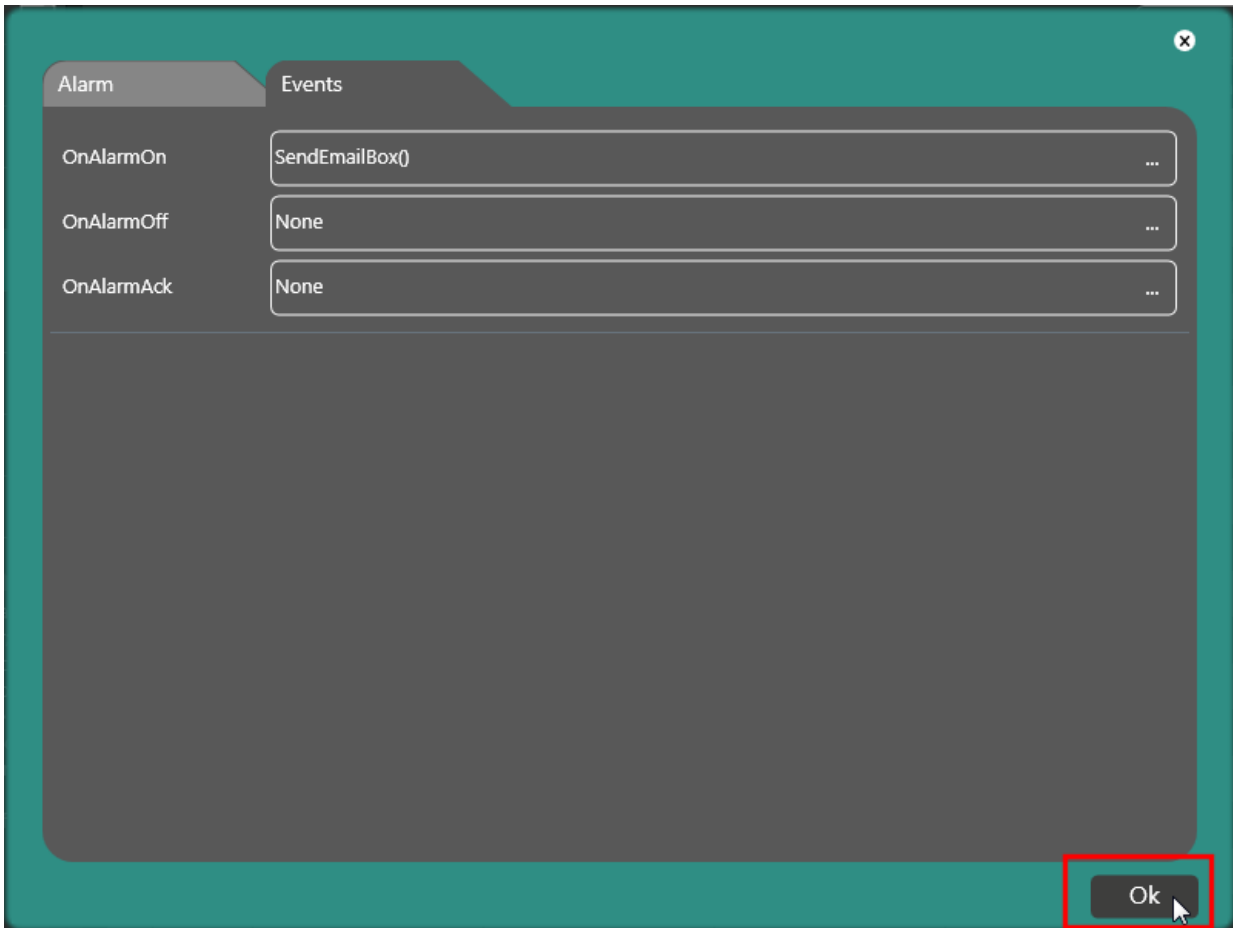
# CREW Manual

When the function has been associated, confirm with "Ok".

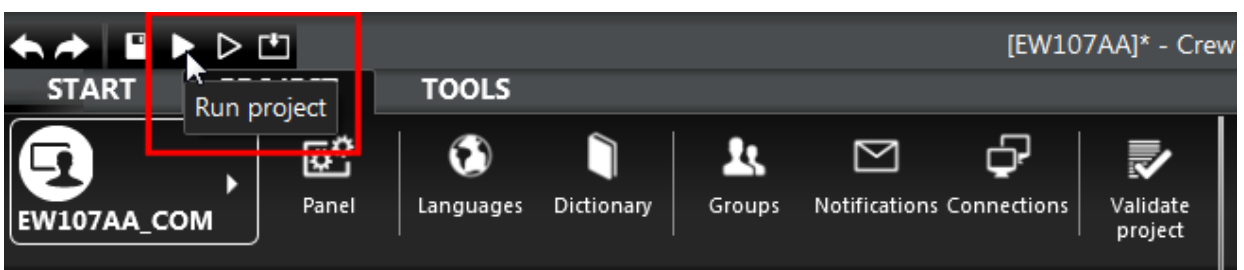


# CREW Manual

Confirm again with "Ok".

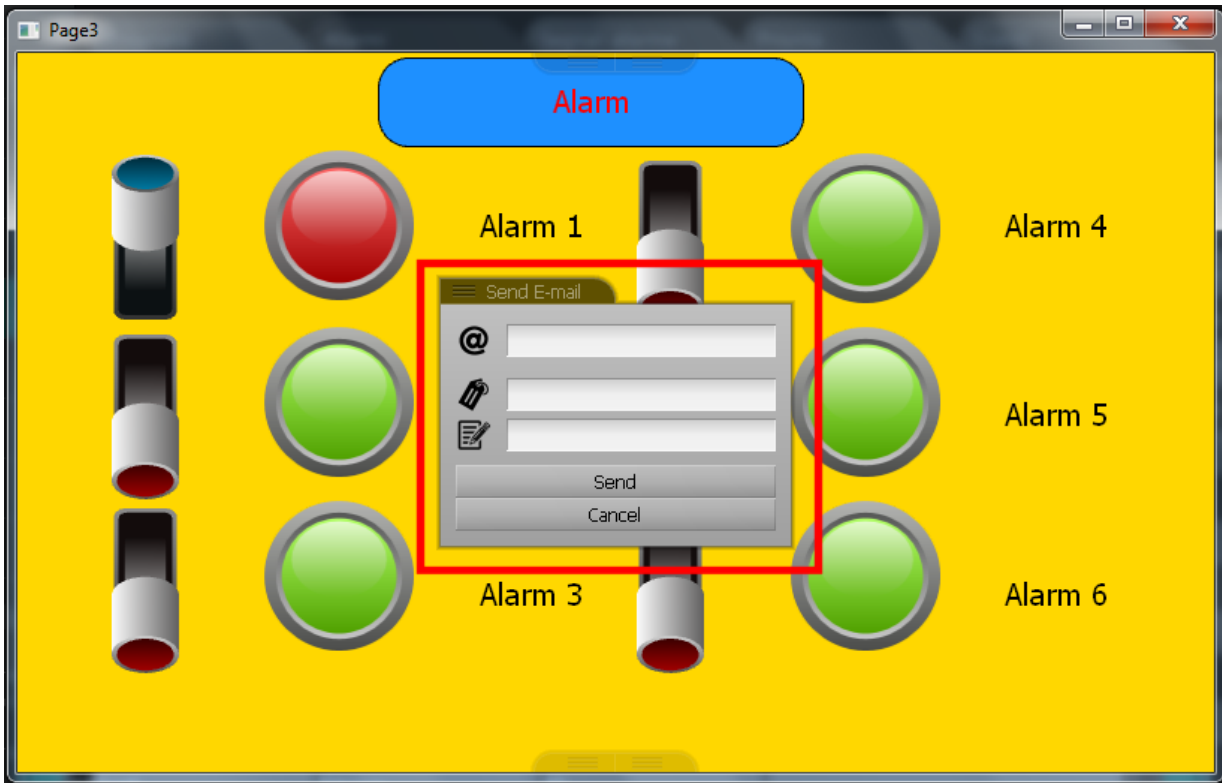


Now click the "Simulate" key.



# CREW Manual

When “Alarm1” is triggered a dialog box appears where all of the data required to send the email is entered.



# CREW Manual


Enter the email address, the subject of the email and a description of the reason why the email was sent.



# CREW Manual

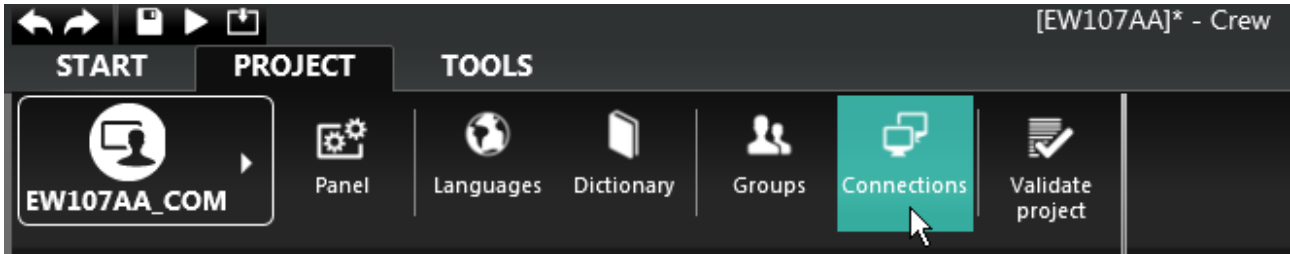
Click “Send” to send the alarm to the entered address.



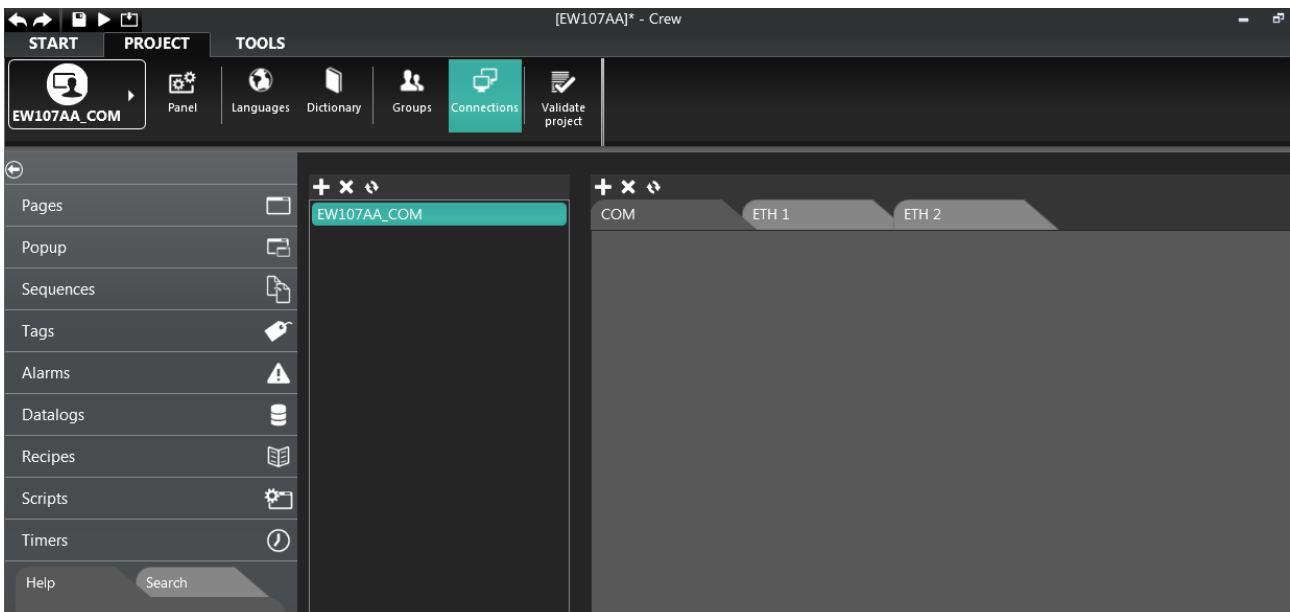
 Note : The email is **ONLY** delivered if the Everyware feature is enabled on the receiving terminal (refer to “[Everyware](#)” section).

# CREW Manual

## Connections



The “Connections” option is used to connect a protocol to the ports of the panel used in the project. To do so, select the protocol from the list based on the type of port available on the panel. In the example below we have used an EW107AA panel (see "[EW107xxxxx](#)" in the "Esaware Products" section), which has one serial port and two ethernet ports.



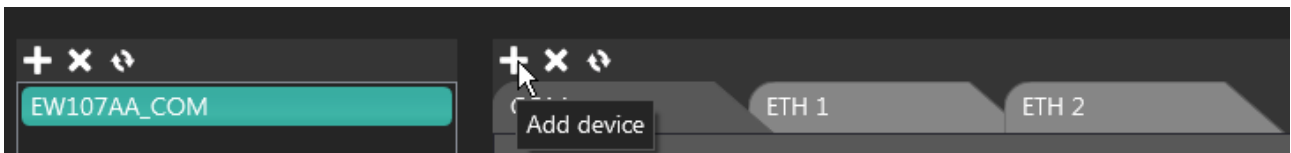
# CREW Manual

## Adding a serial protocol

Select the “COM” serial port on the panel.

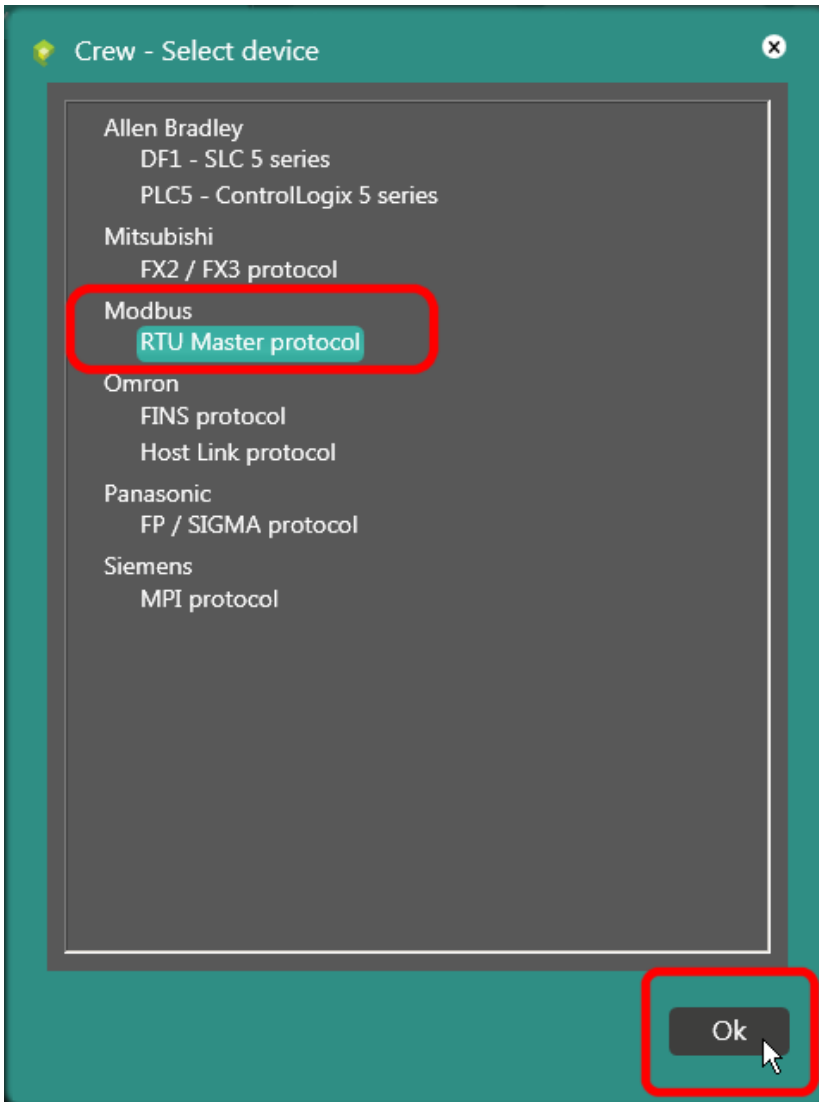


Click the “Add device” key.




# CREW Manual

Now select a serial protocol from those provided by Crew and click on “Ok” to confirm.



 Note: It is possible to connect a protocol for each serial port on the panel.

 Note : For information on protocol parameters, refer to the "[Drivers](#)" section.



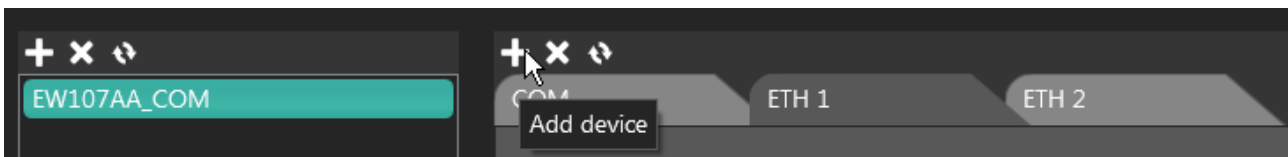
# CREW Manual

## Adding an ethernet protocol

To add a protocol on the ETH1 ethernet port,

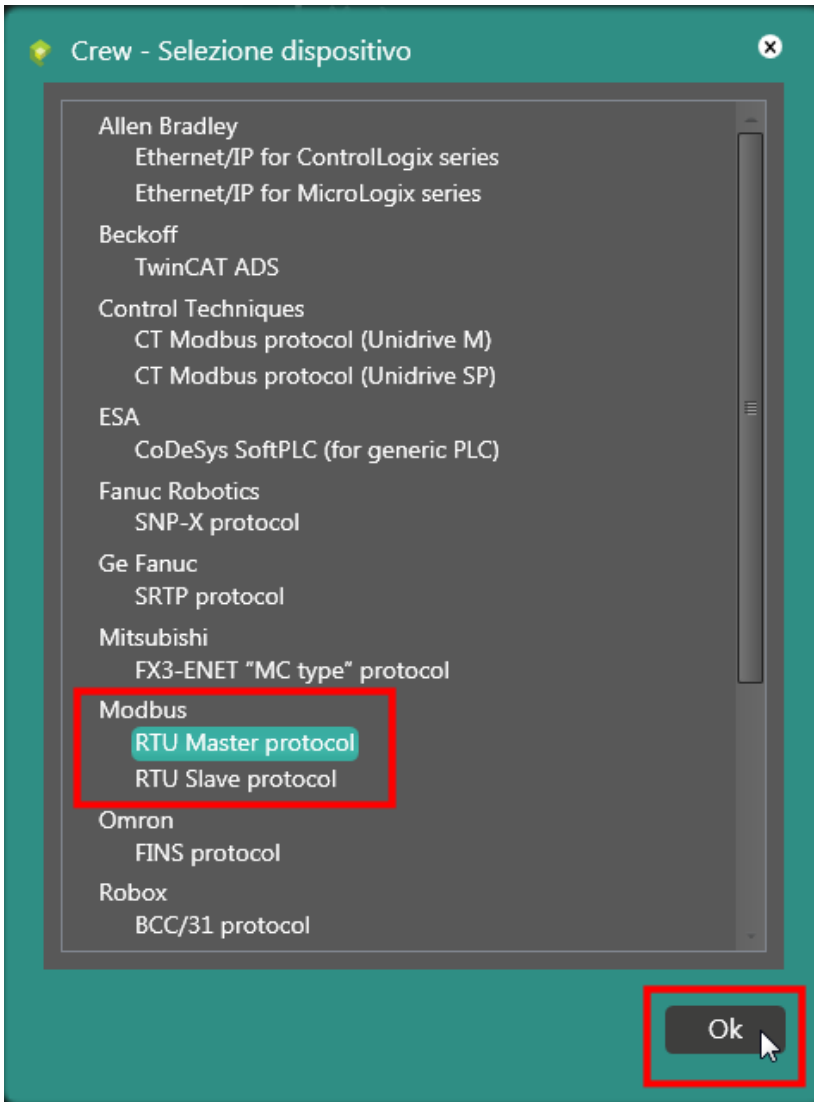



click the “Add device” key.




# CREW Manual

Then select an ethernet protocol from those provided by Crew and click on “Ok” to confirm.



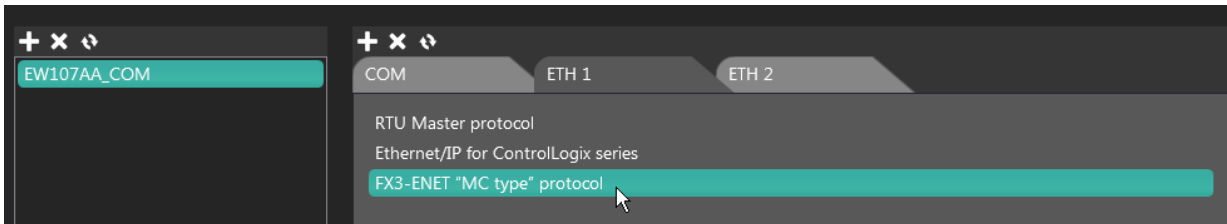
 Note: It is possible to connect up to two protocols for each ethernet port on the panel.

 Note : For information on protocol parameters, refer to the "[Drivers](#)" section.

# CREW Manual

## Removing a protocol

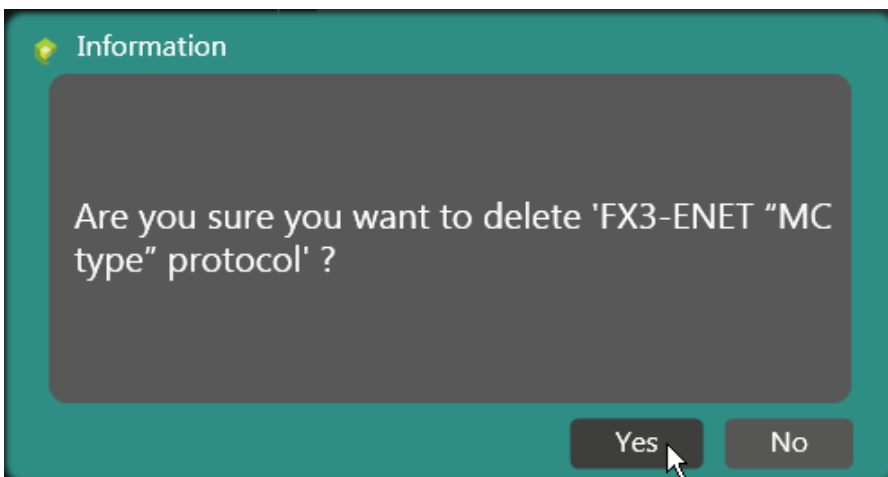
To eliminate a protocol, select the element you wish to remove.



Click the "Eliminate device" key.



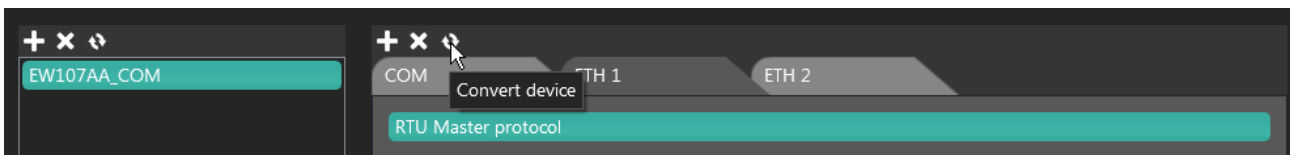
Click "Ok" to confirm.



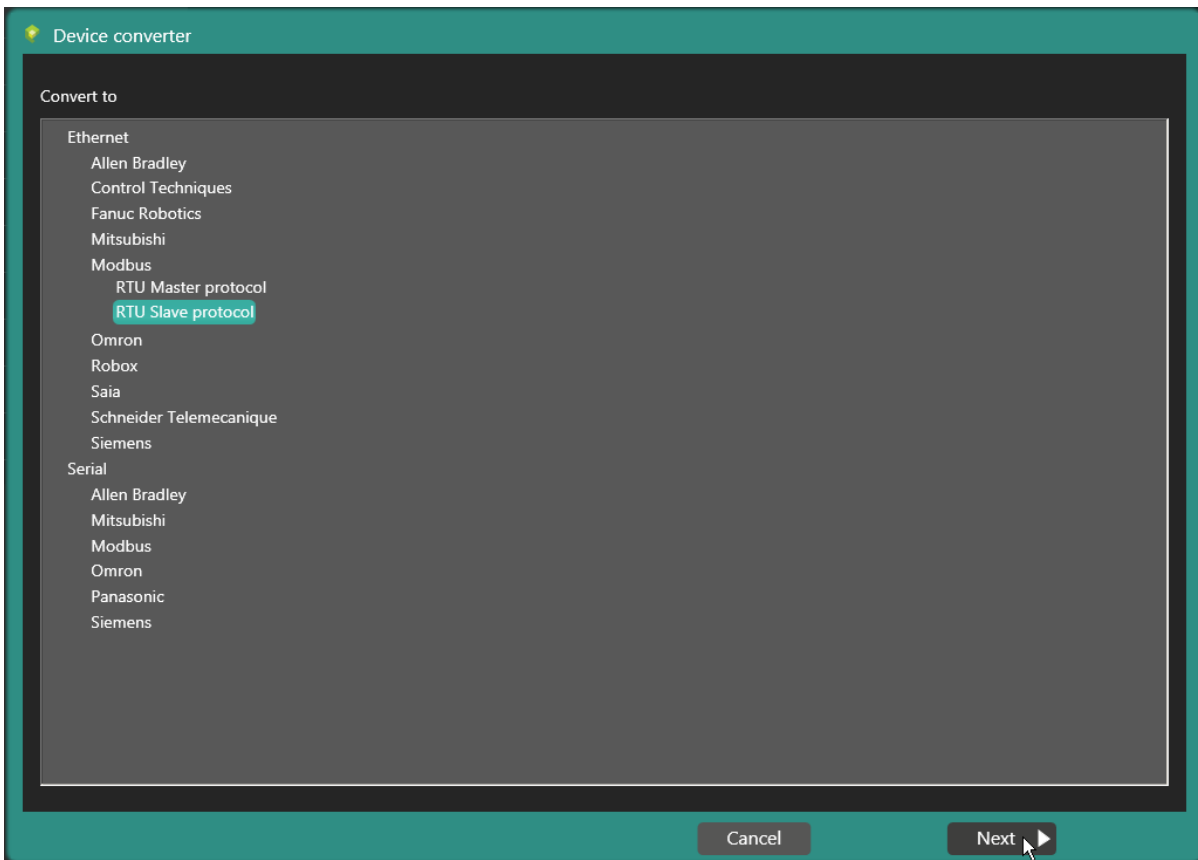
# CREW Manual

## Convert device

The “Convert device” option is used to convert one protocol into another. For example, if you wish to convert an RTU Master protocol into an RTU Slave on the ETH1 port, you need to select the protocol of origin and click on the “Convert device” key.

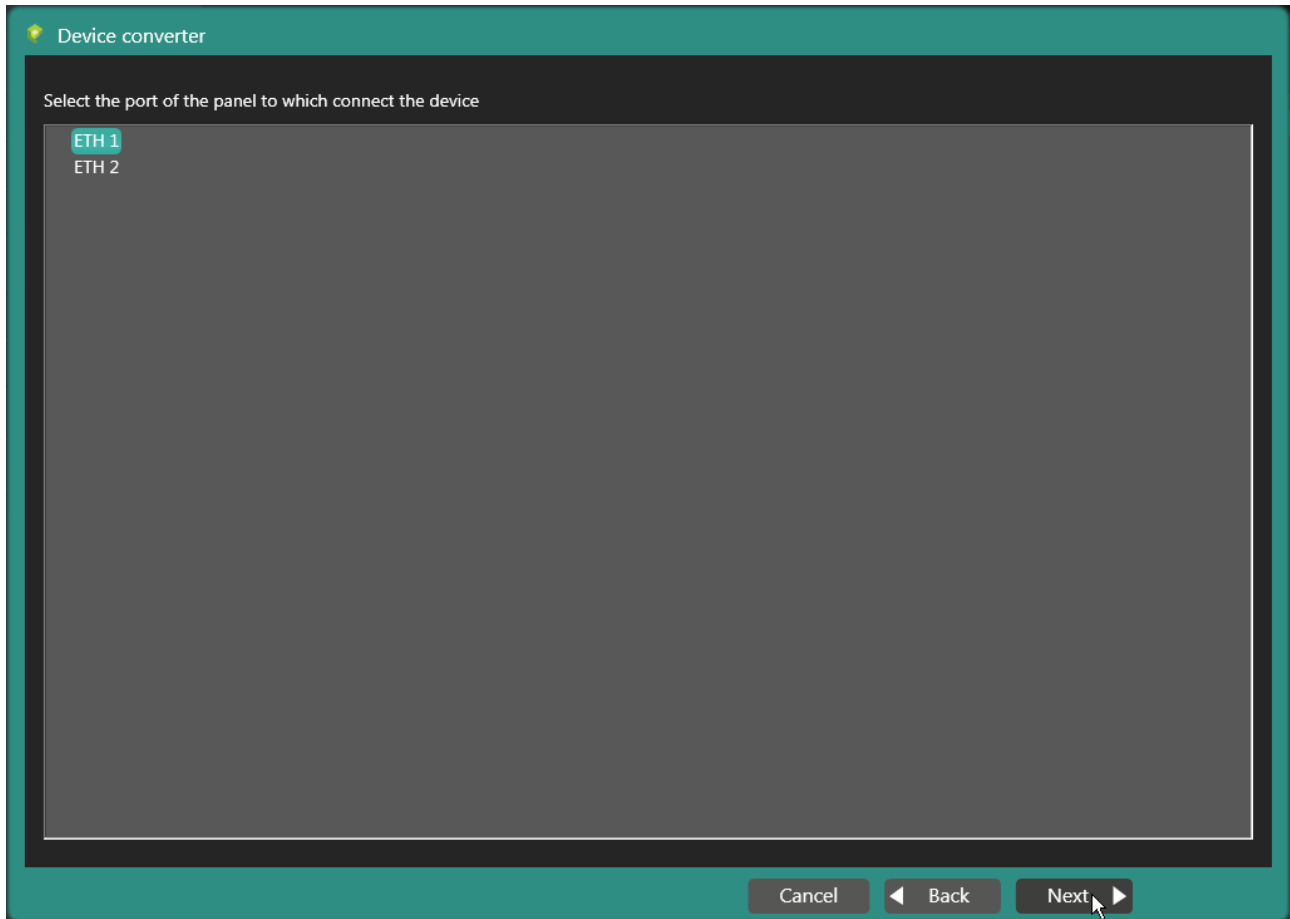


When the box appears with the selection options for the target protocol, select the required protocol and click on “Next”.



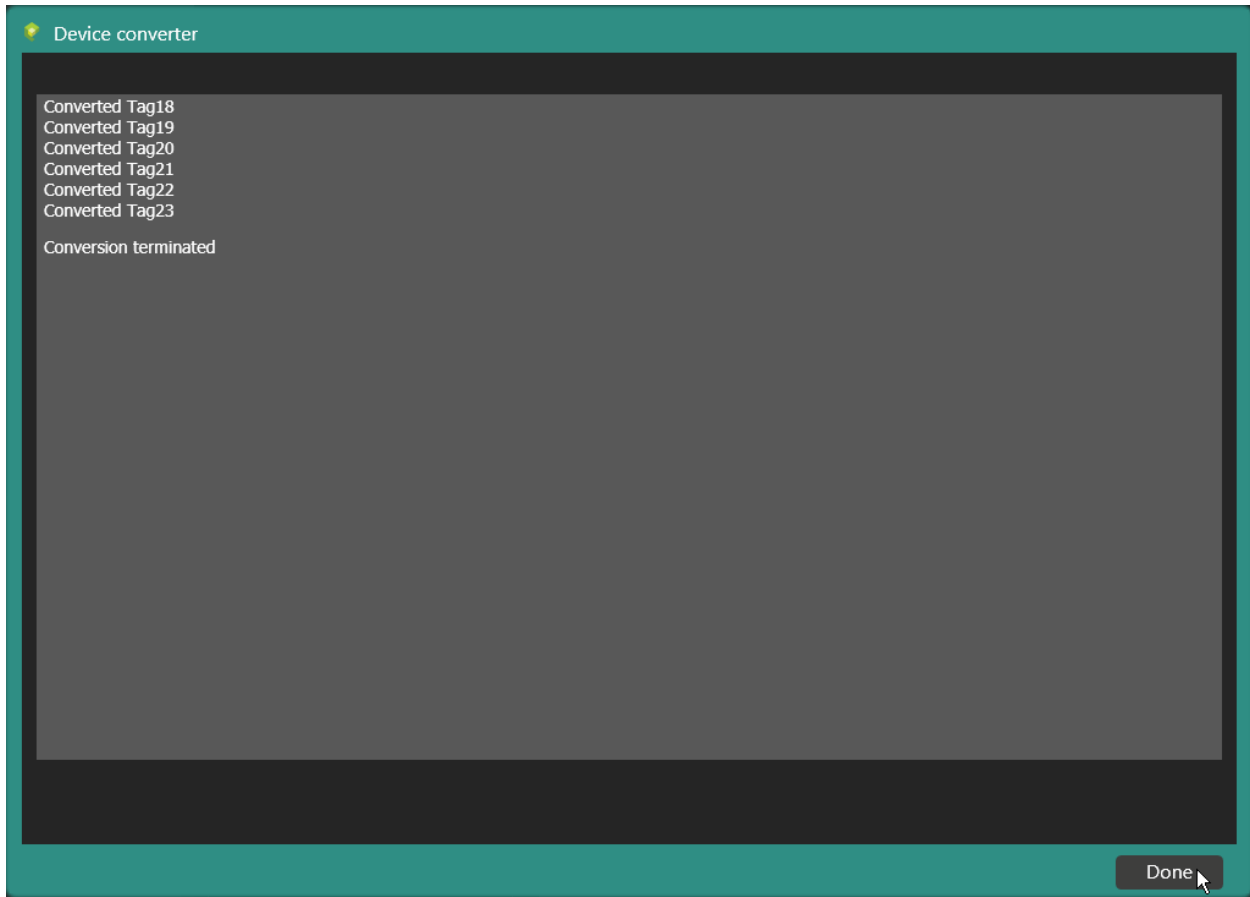
# CREW Manual

Select the port that the end protocol needs to be connected to and click “Next”.



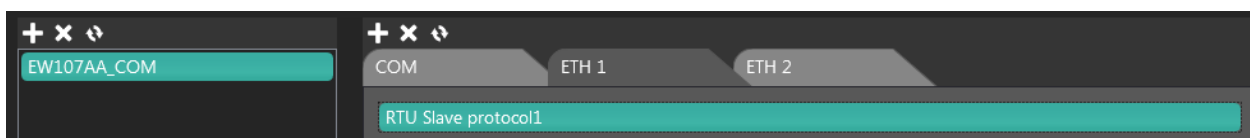
# CREW Manual

Check the mask to ensure there are no errors with the converted variables and click “Done”.



Note: It is also possible to convert an ethernet protocol into serial or vice versa, under the condition that there is addressing compatibility of the memory areas of the relative devices.

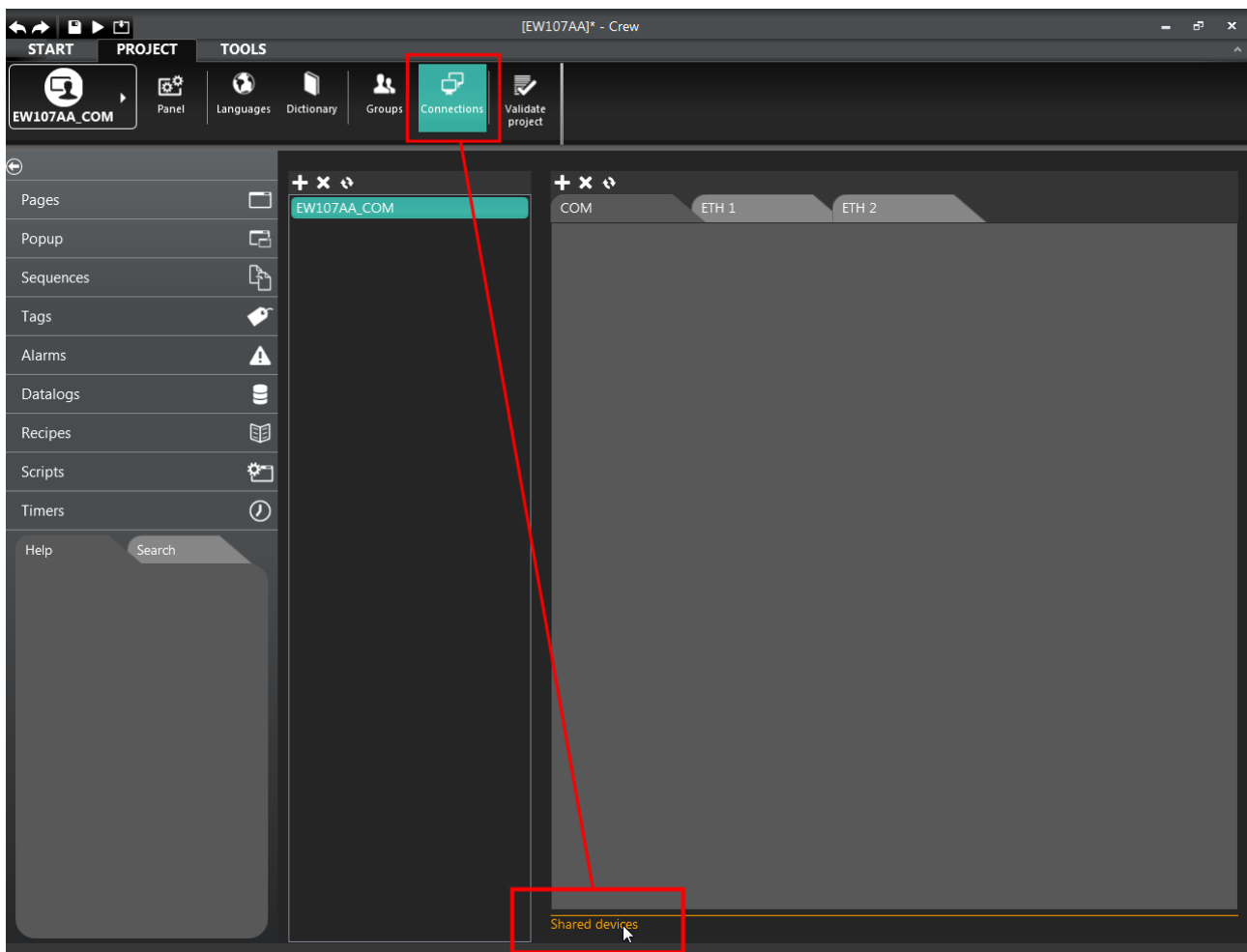
The modified protocol appears in place of the original one.



# CREW Manual

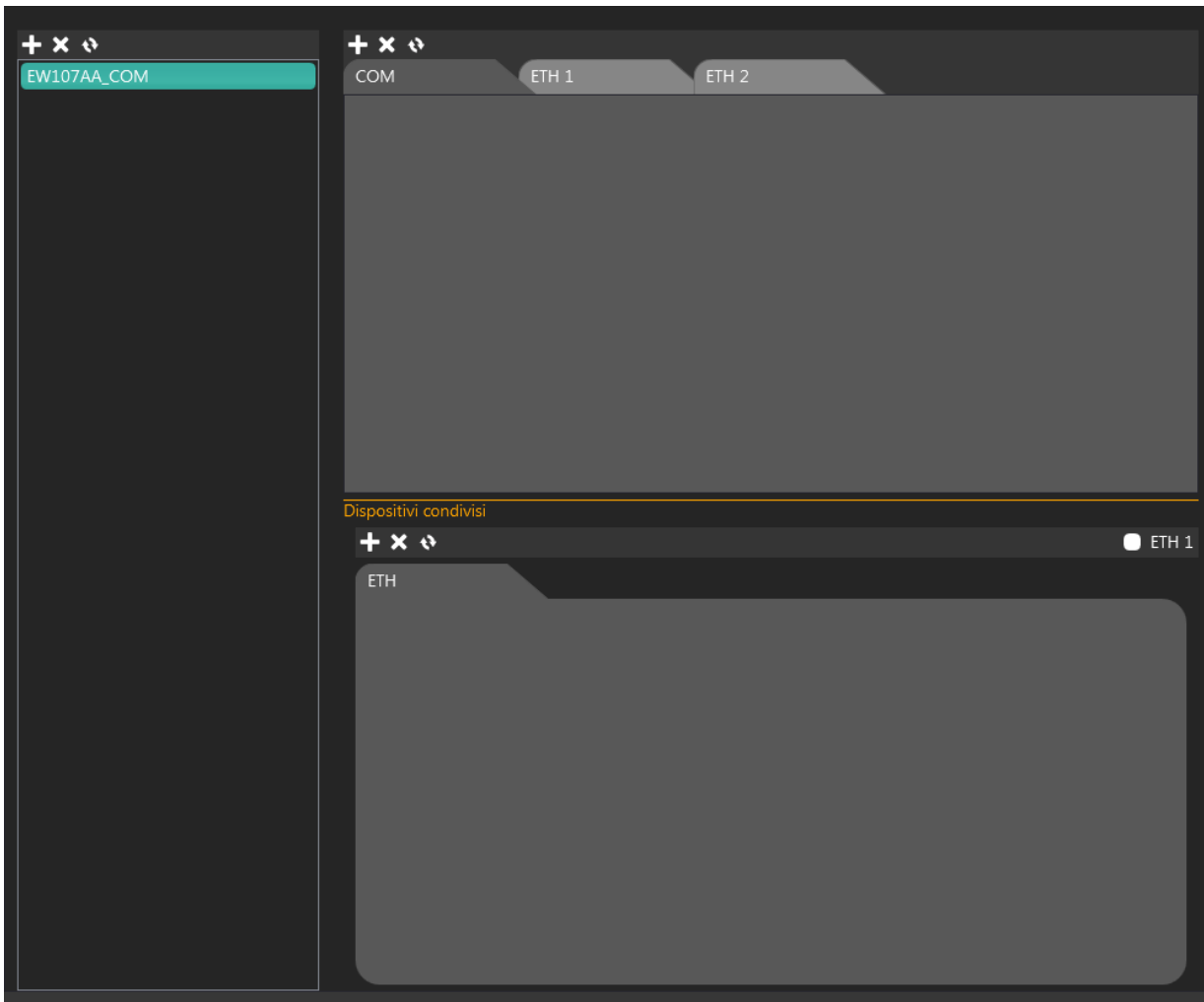
## Shared devices

At the bottom of the “Connections” option mask there is the “Shared devices” function.



# CREW Manual

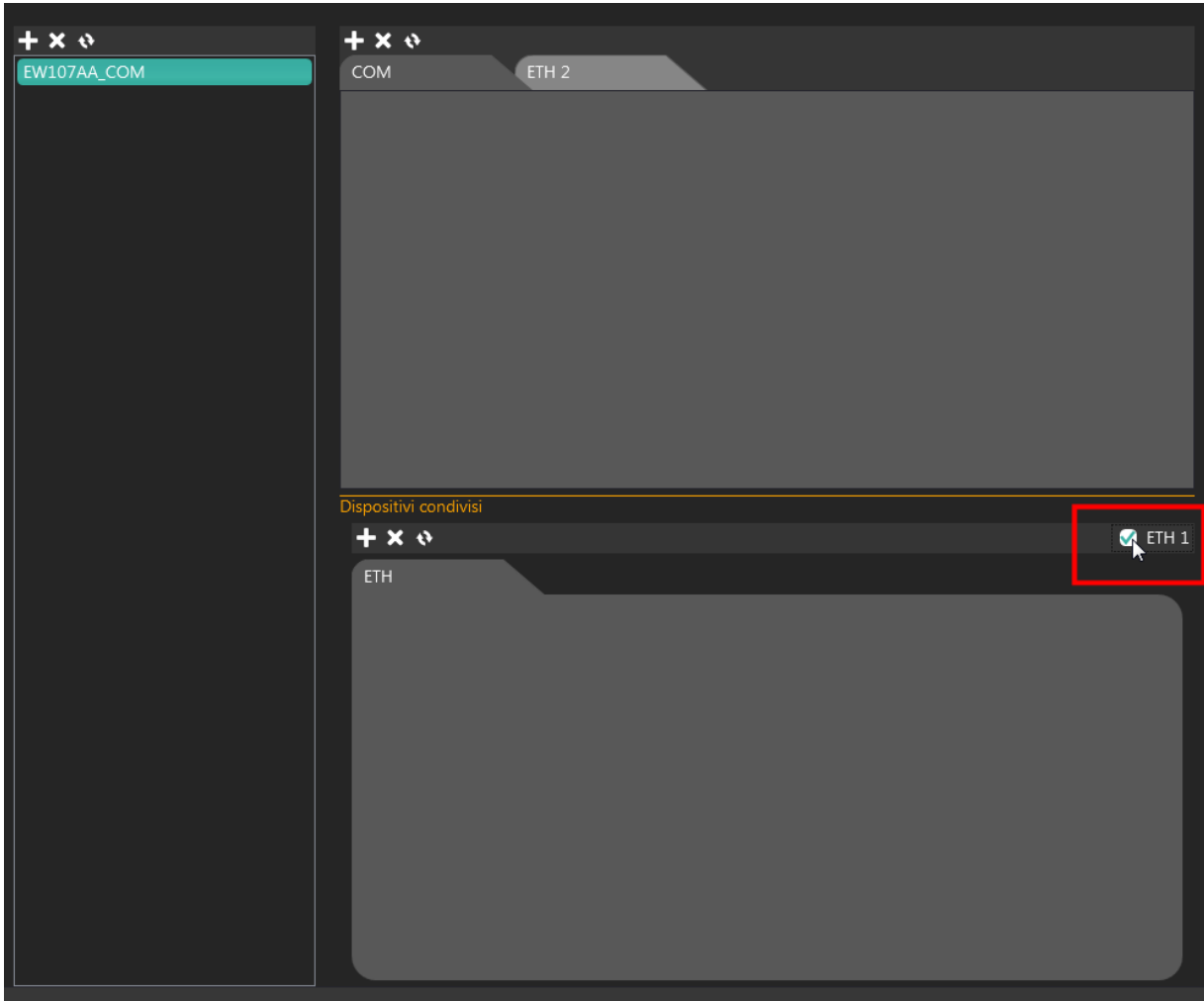
The “Shared devices” option can be used when there is a system with multiple panels that read shared data areas in a single device (PLC). To do so it is necessary to firstly include a device in the project, to share on the ETH1 port. After clicking “Shared devices” the following image appears.



When the ETH 1 “check box” is enabled, the ETH 1 port disappears from those available on the EW panel (at top of previous image), as it is now occupied by the shared device.

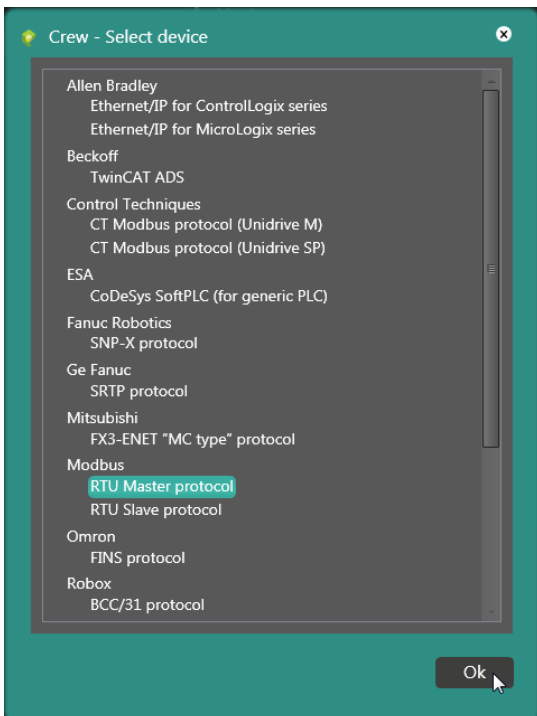
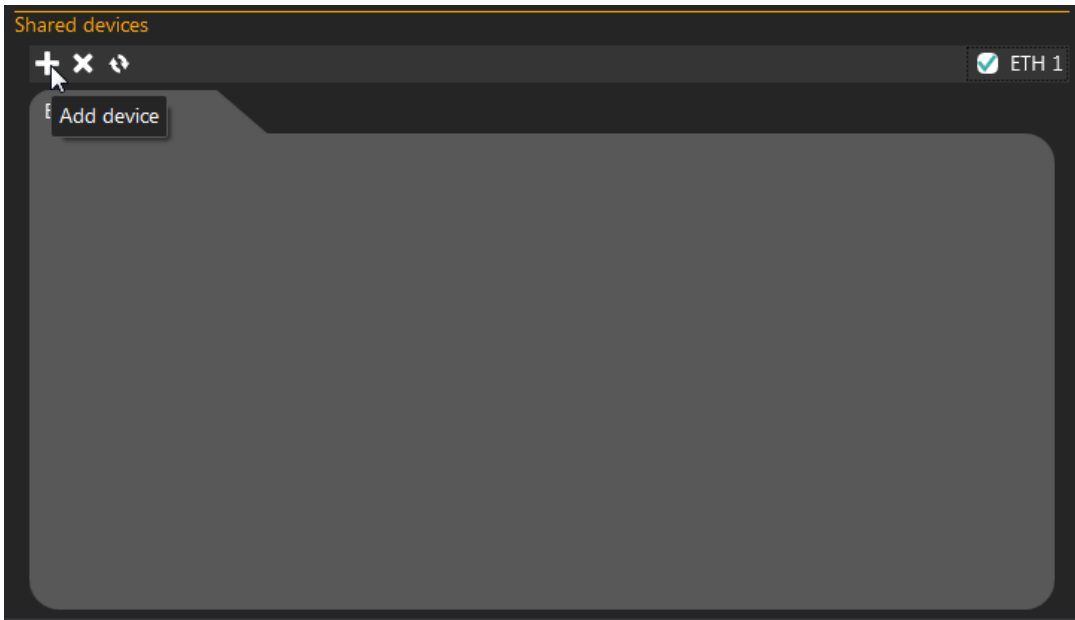


# CREW Manual



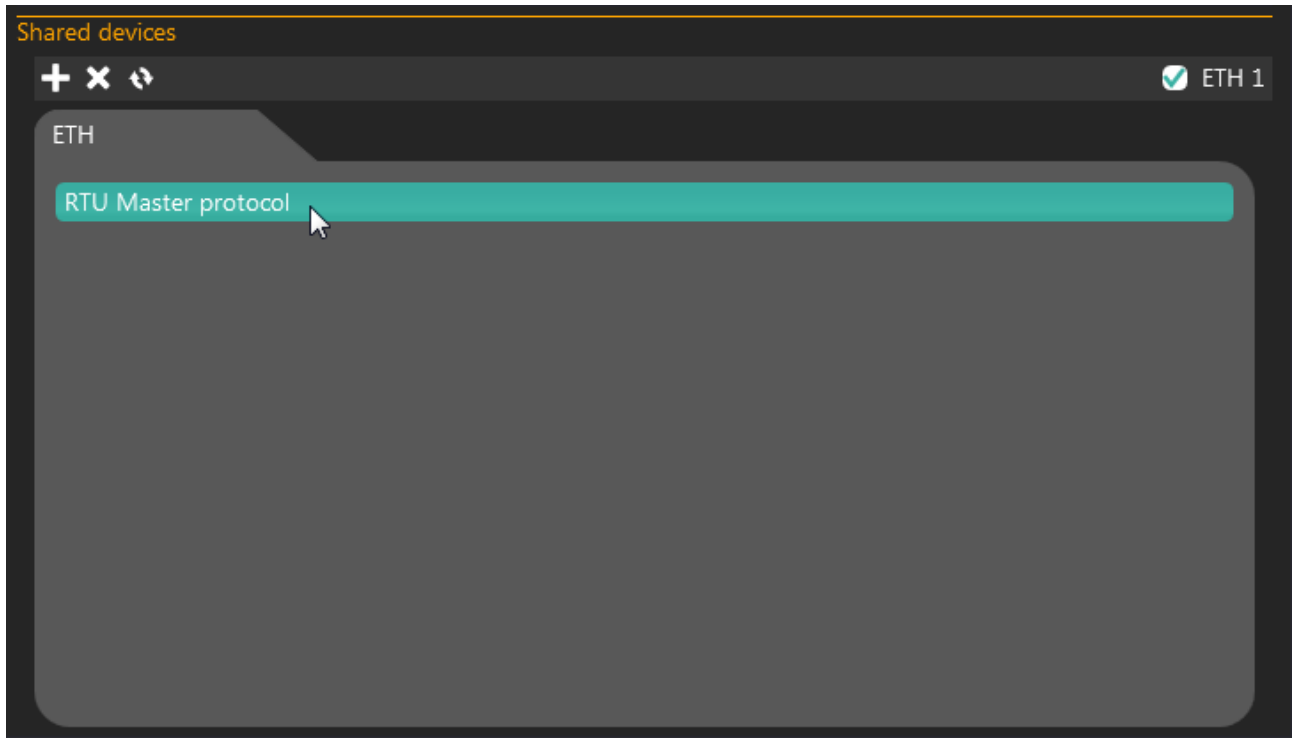
# CREW Manual

Now, choose the type of device to share and confirm with “Ok”.



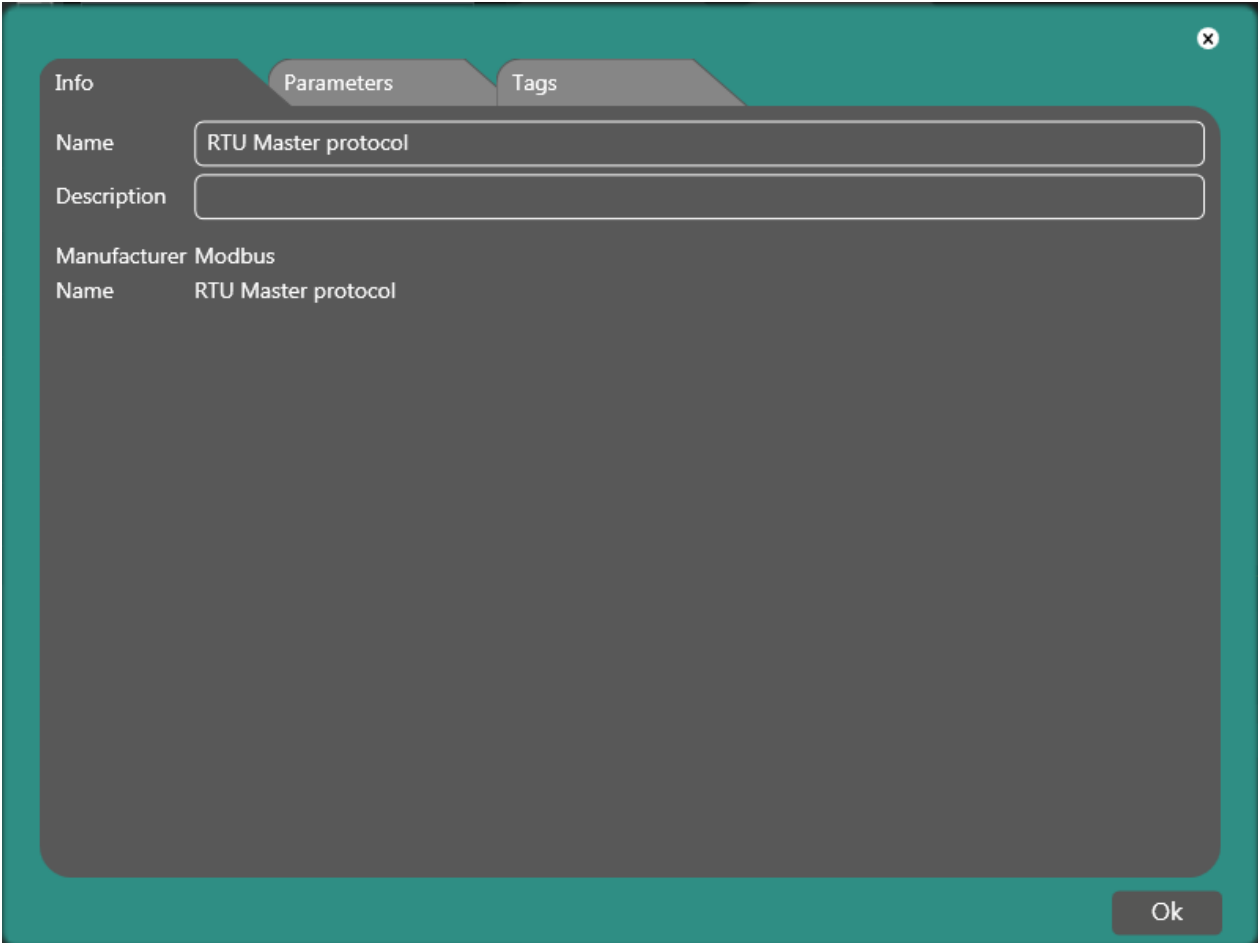
# CREW Manual

To check and/or change the device parameters, double click the following entry.



# CREW Manual

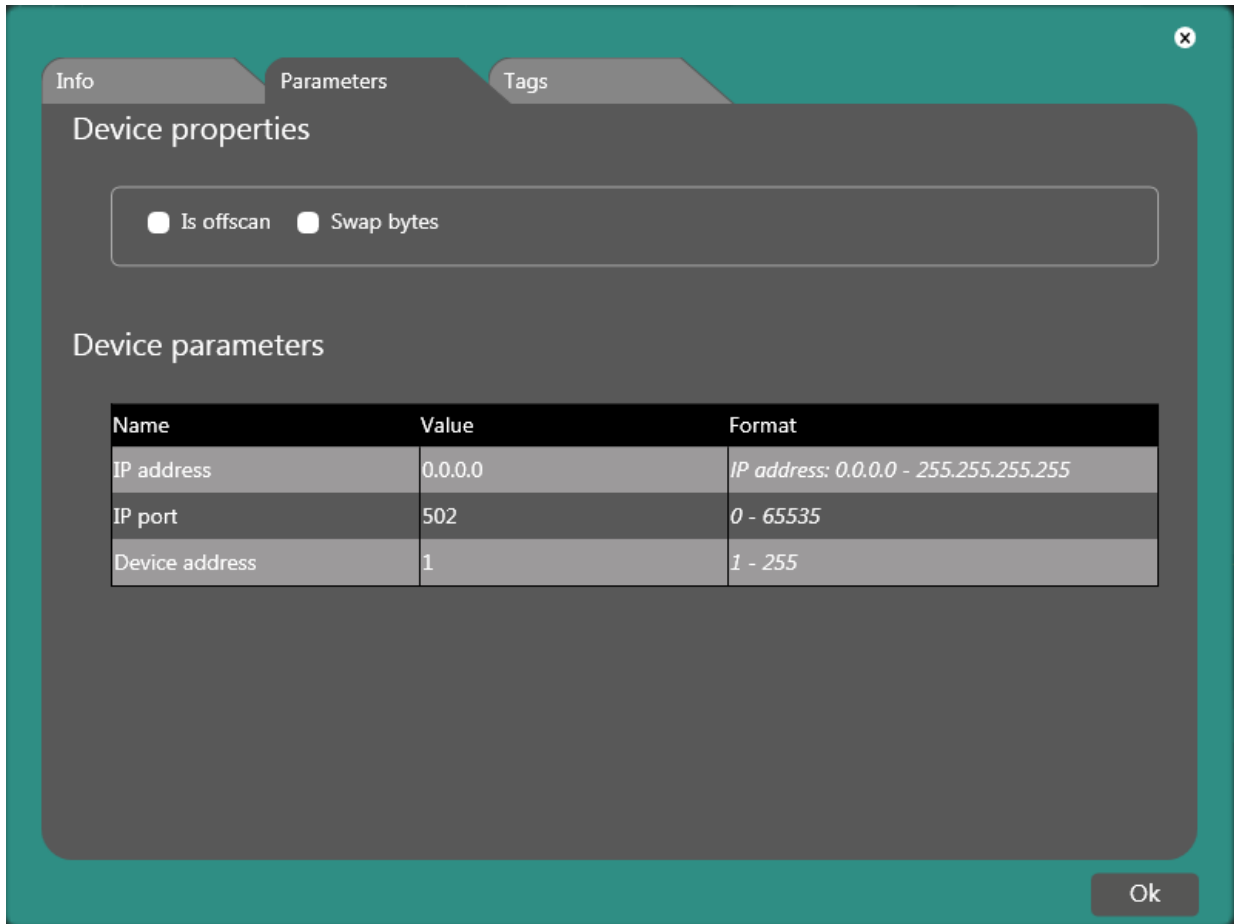
From the “Info” mask it is possible to change the name and description of the device.



The screenshot shows a software interface with a dark teal background. At the top, there are three tabs: 'Info', 'Parameters', and 'Tags'. The 'Info' tab is selected. Below the tabs, there are three input fields: 'Name' (containing 'RTU Master protocol'), 'Description' (empty), and 'Manufacturer Modbus' (containing 'RTU Master protocol'). An 'Ok' button is located at the bottom right of the mask.

# CREW Manual

From the “Parameters” mask it is possible to change some of the properties and parameters of the device.



## Device properties

Offscan Properties: The "Offscan" property is an attribute of the device that determines whether it can be enabled or disabled to be interrogated by the terminal. If the check box is enabled, the device - and all of the tags contained in it - is never interrogated by the terminal.



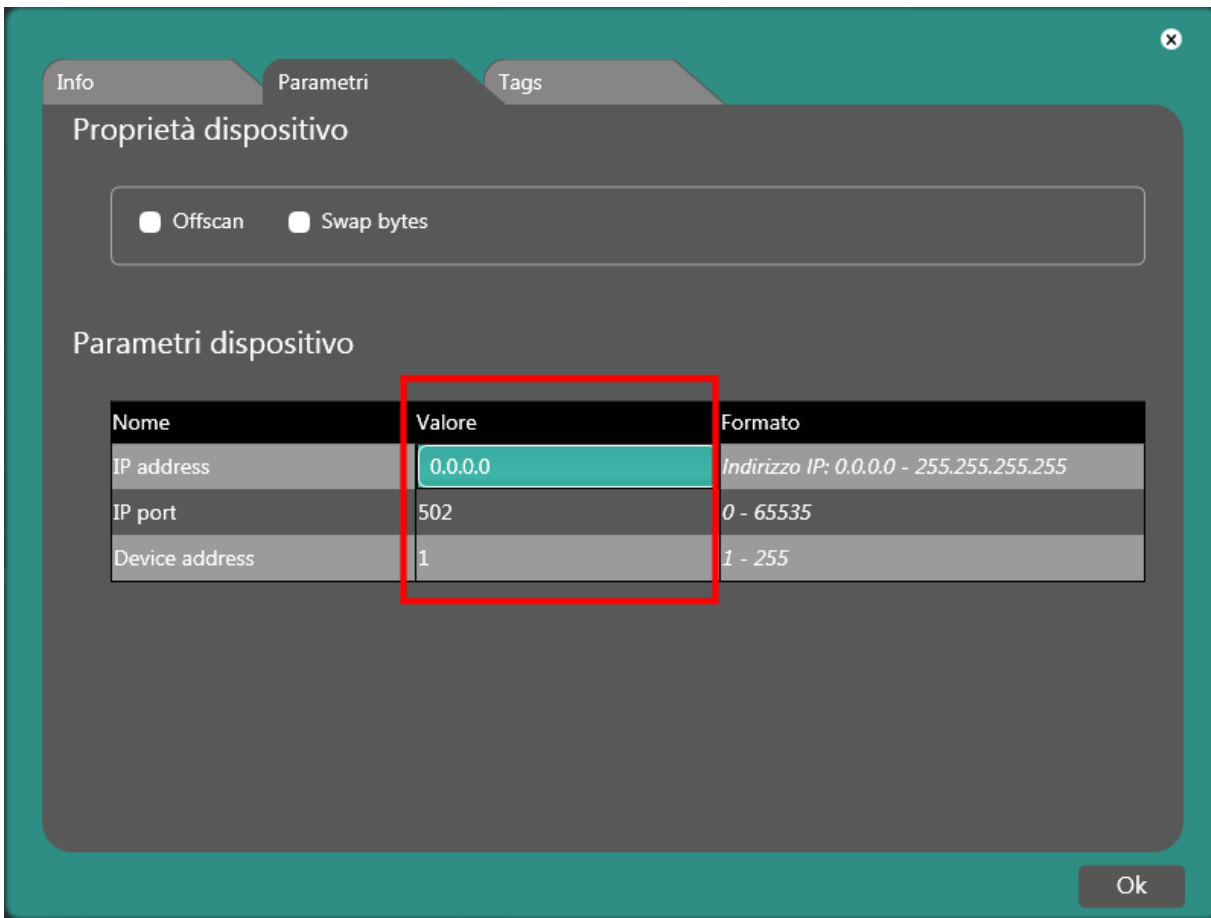
Note: It is possible to interrogate one or more tags on the device after having changed the "[OffScan Mode](#)" option.

"Swap Bytes" property: The "Swap Bytes" property defines the order of the bytes that the terminal reads from the device.

# CREW Manual

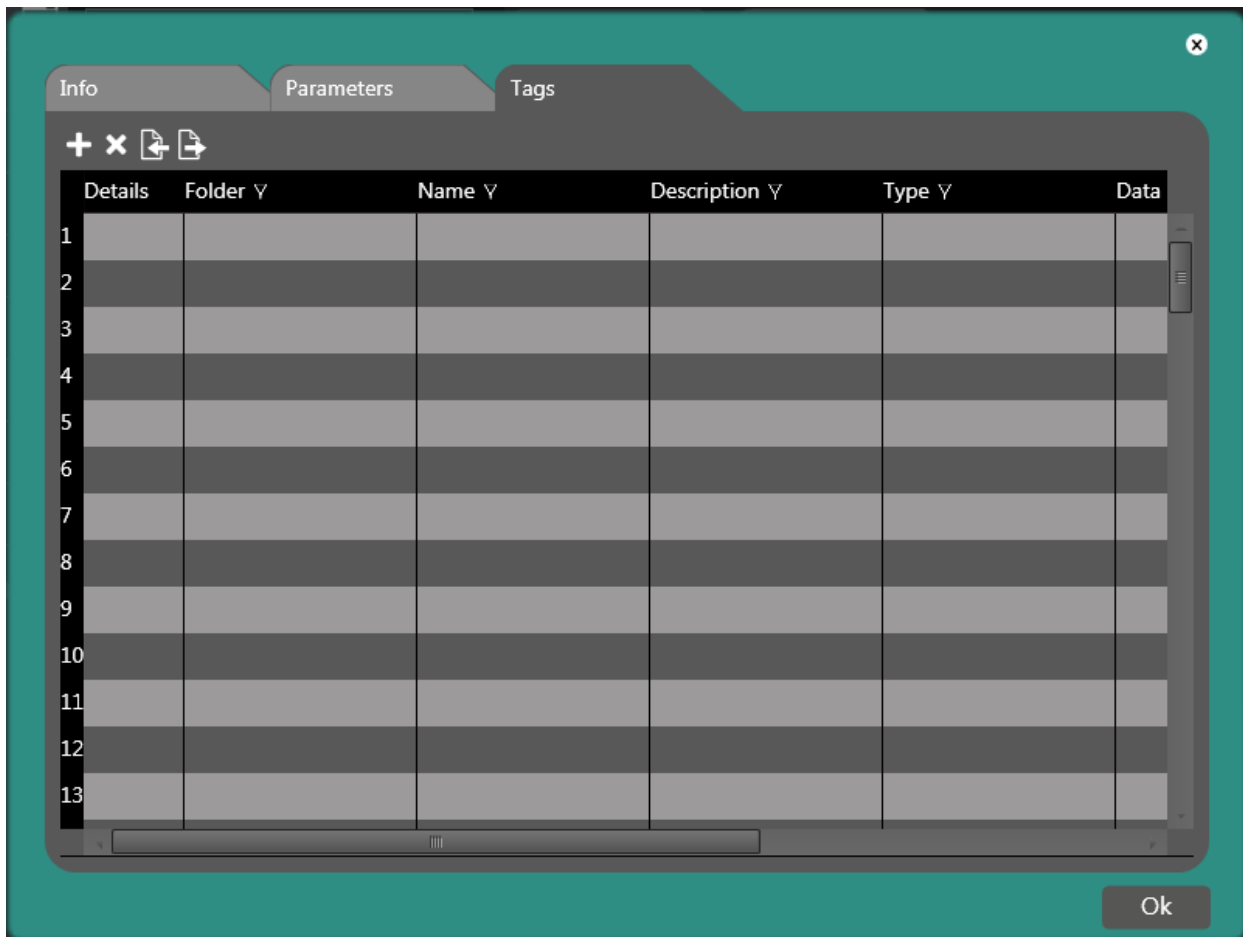
## Device parameters

By double clicking it is possible to change the items in the “Value” column of the device parameters (a “Modbus RTU Master” in our example). Simply enter the values contained within the ranges provided in the “Format” column.



# CREW Manual

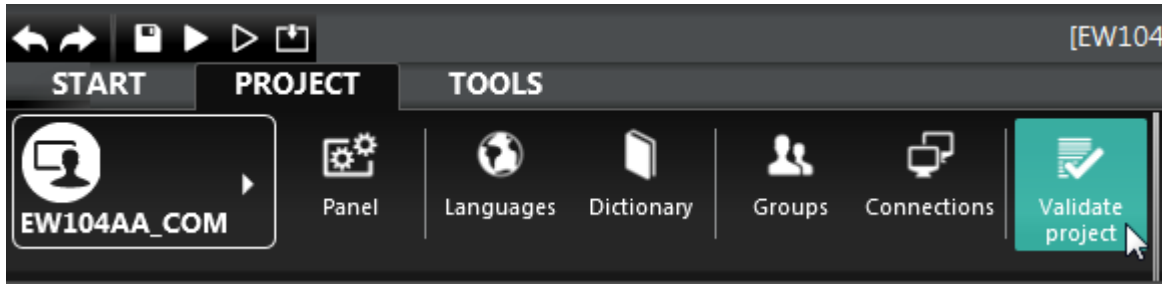
## Tags



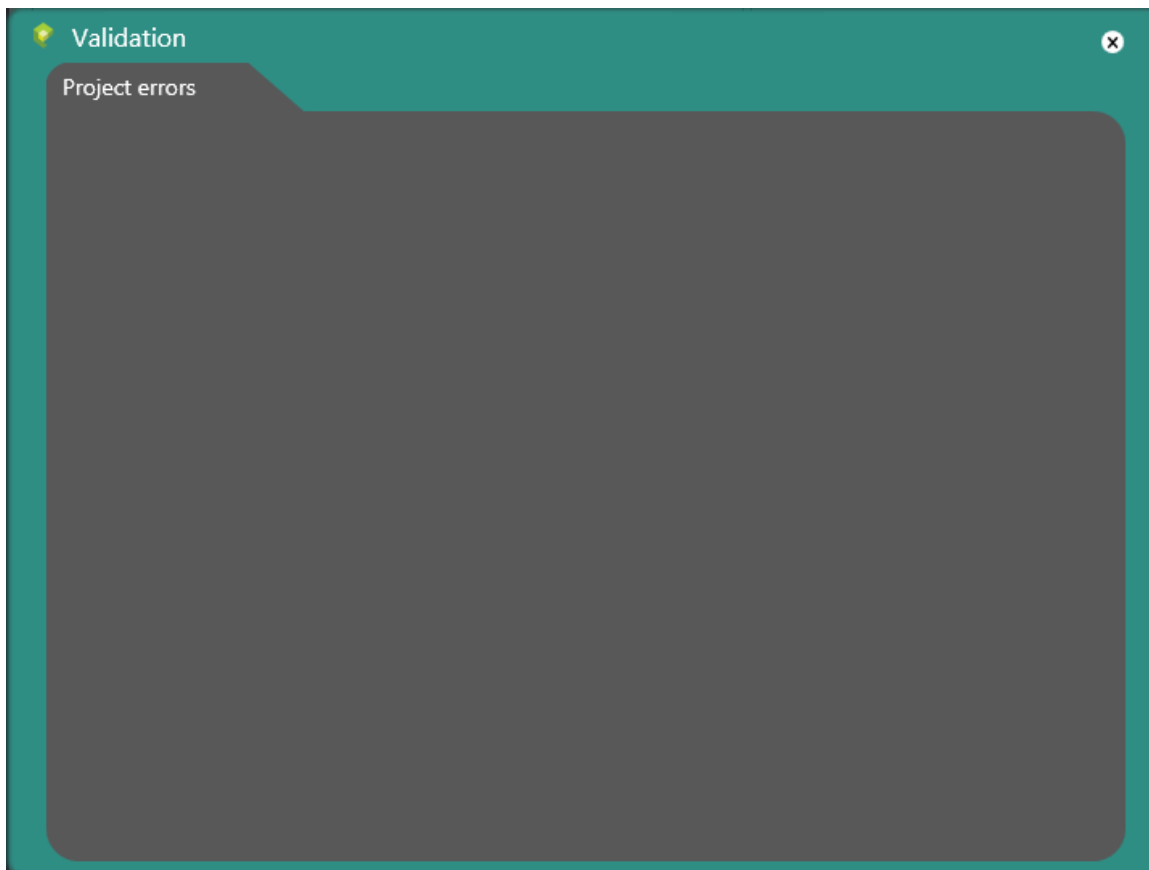
With the "Tags" option it is possible to enter new variables into the device (refer to the ["Shared Device Tag"](#) section).

# CREW Manual

## Project validation



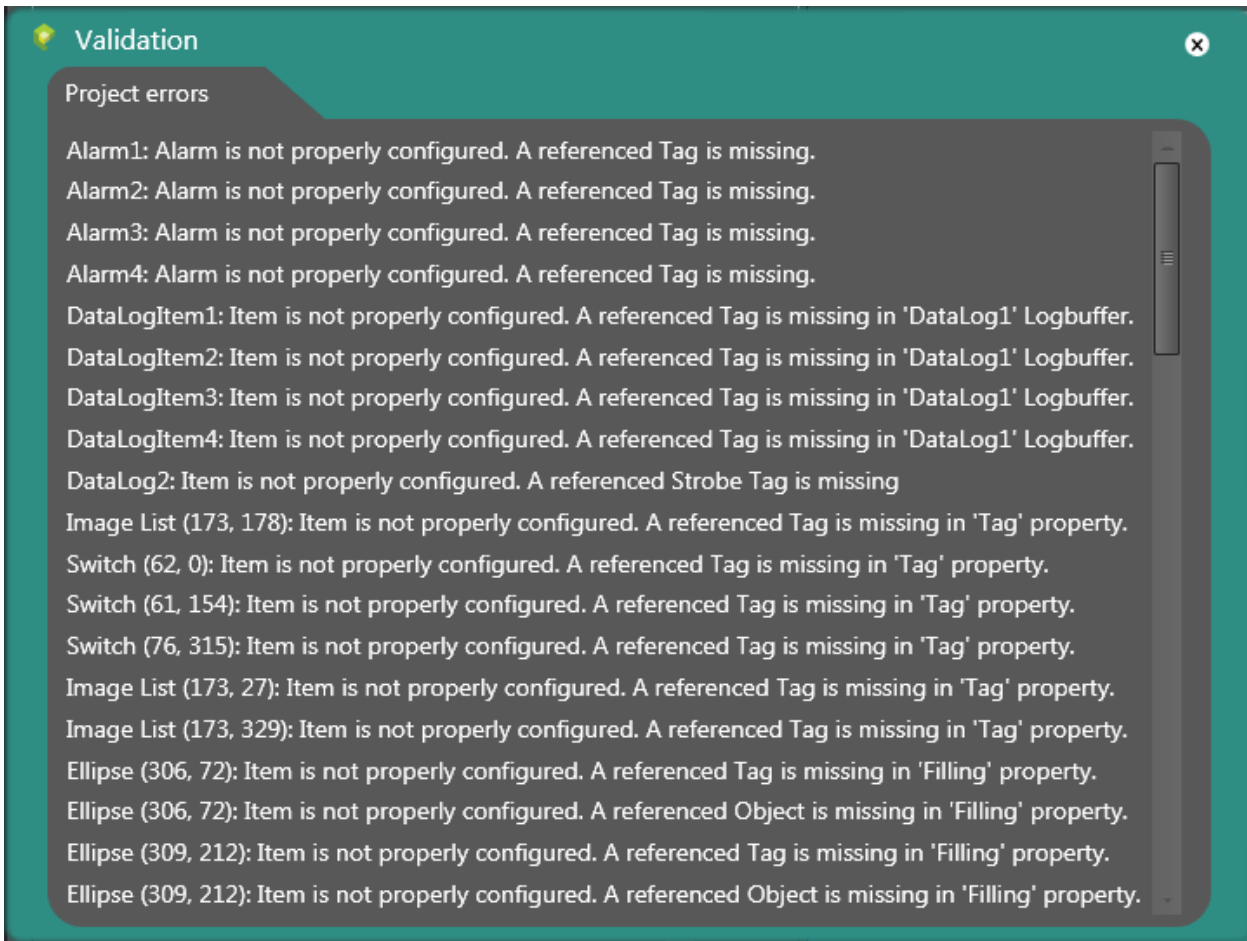
Validation is the operation that verifies the coherence of the objects entered in the project. The presence of any errors is reported in the following box.



Examples of errors during validation.



# CREW Manual



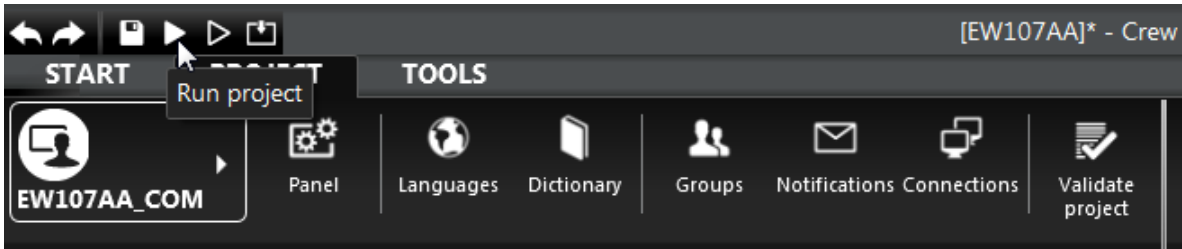
The screenshot shows a 'Validation' dialog box with a close button (X) in the top right corner. The title bar contains a yellow diamond icon and the text 'Validation'. Below the title bar, the text 'Project errors' is displayed. The main area of the dialog lists 17 error messages, each describing a configuration issue for a specific object in the project.

Project errors

- Alarm1: Alarm is not properly configured. A referenced Tag is missing.
- Alarm2: Alarm is not properly configured. A referenced Tag is missing.
- Alarm3: Alarm is not properly configured. A referenced Tag is missing.
- Alarm4: Alarm is not properly configured. A referenced Tag is missing.
- DataLogItem1: Item is not properly configured. A referenced Tag is missing in 'DataLog1' Logbuffer.
- DataLogItem2: Item is not properly configured. A referenced Tag is missing in 'DataLog1' Logbuffer.
- DataLogItem3: Item is not properly configured. A referenced Tag is missing in 'DataLog1' Logbuffer.
- DataLogItem4: Item is not properly configured. A referenced Tag is missing in 'DataLog1' Logbuffer.
- DataLog2: Item is not properly configured. A referenced Strobe Tag is missing
- Image List (173, 178): Item is not properly configured. A referenced Tag is missing in 'Tag' property.
- Switch (62, 0): Item is not properly configured. A referenced Tag is missing in 'Tag' property.
- Switch (61, 154): Item is not properly configured. A referenced Tag is missing in 'Tag' property.
- Switch (76, 315): Item is not properly configured. A referenced Tag is missing in 'Tag' property.
- Image List (173, 27): Item is not properly configured. A referenced Tag is missing in 'Tag' property.
- Image List (173, 329): Item is not properly configured. A referenced Tag is missing in 'Tag' property.
- Ellipse (306, 72): Item is not properly configured. A referenced Tag is missing in 'Filling' property.
- Ellipse (306, 72): Item is not properly configured. A referenced Object is missing in 'Filling' property.
- Ellipse (309, 212): Item is not properly configured. A referenced Tag is missing in 'Filling' property.
- Ellipse (309, 212): Item is not properly configured. A referenced Object is missing in 'Filling' property.

# CREW Manual

## Project simulation

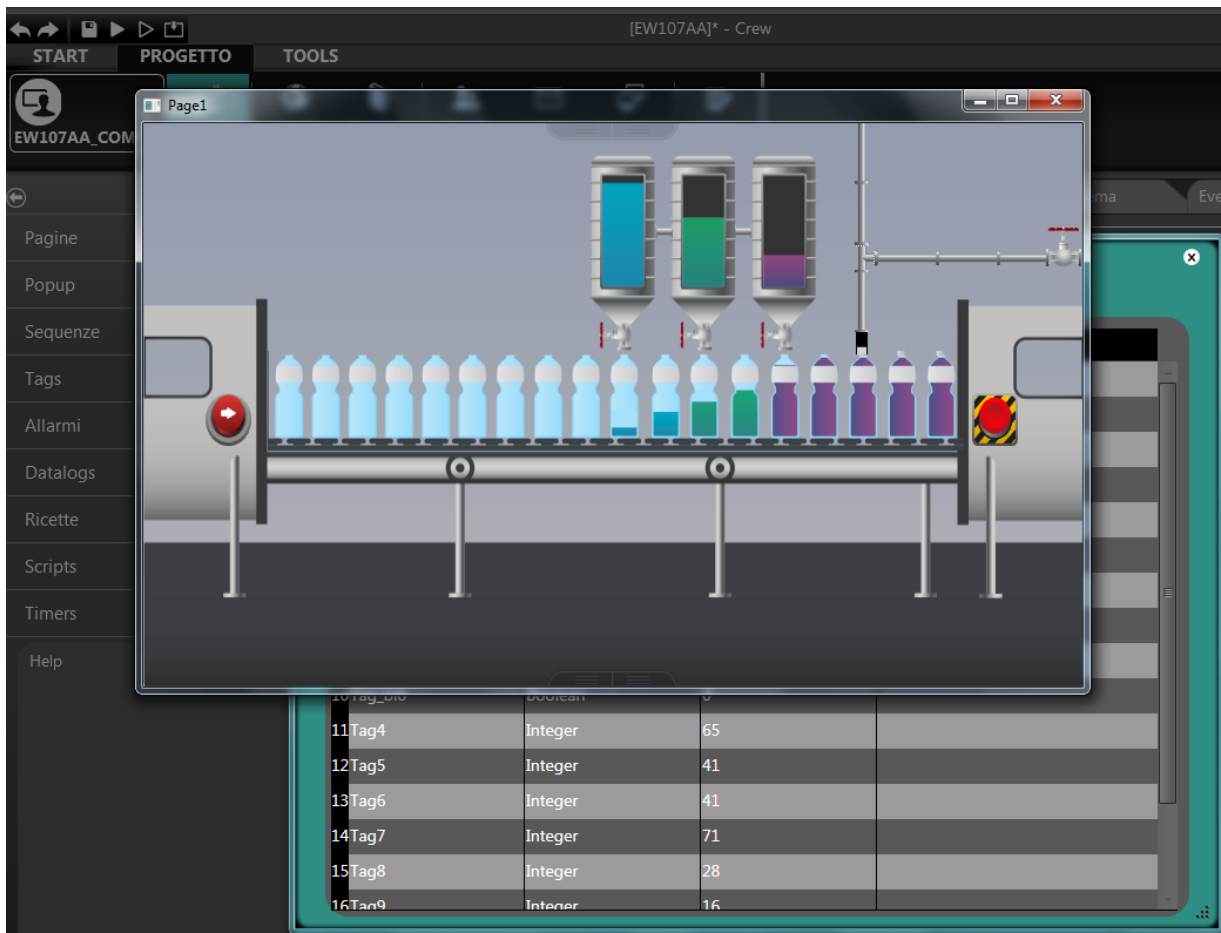


Project simulation makes it possible to run a test on the project without having to transfer it to the terminal. During simulation it is possible to run all of the operations within the project, as though you were at the terminal.

Normally simulation is conducted after Validation, when it has been verified that there are no errors in the project.

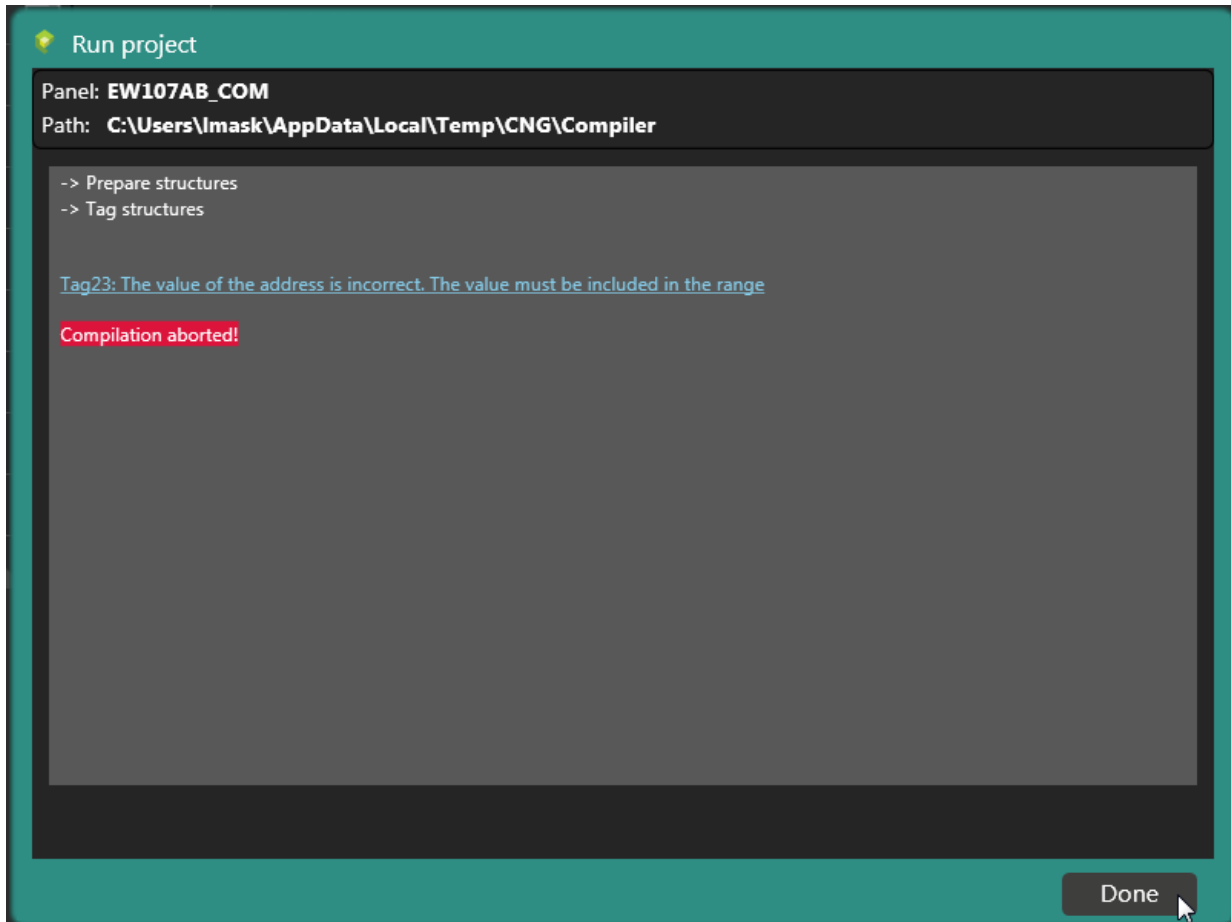
# CREW Manual

After clicking on the relative key (if there are no errors) the project is opened with Crew Runtime.



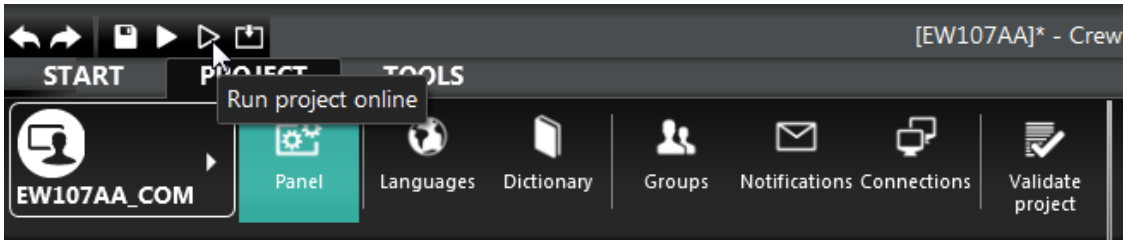
# CREW Manual

If there are errors in the project, on the other hand, compilation is interrupted and Runtime does not open.

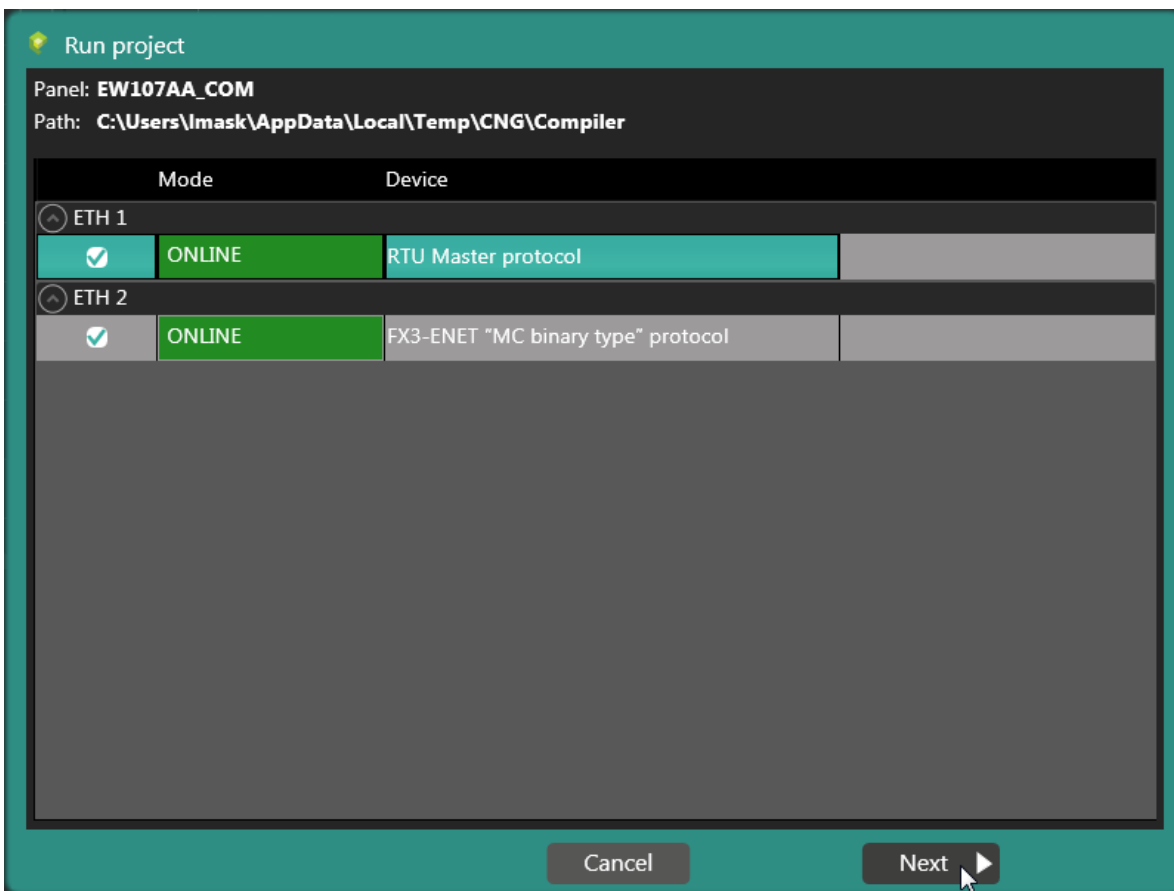


# CREW Manual

## Online simulation

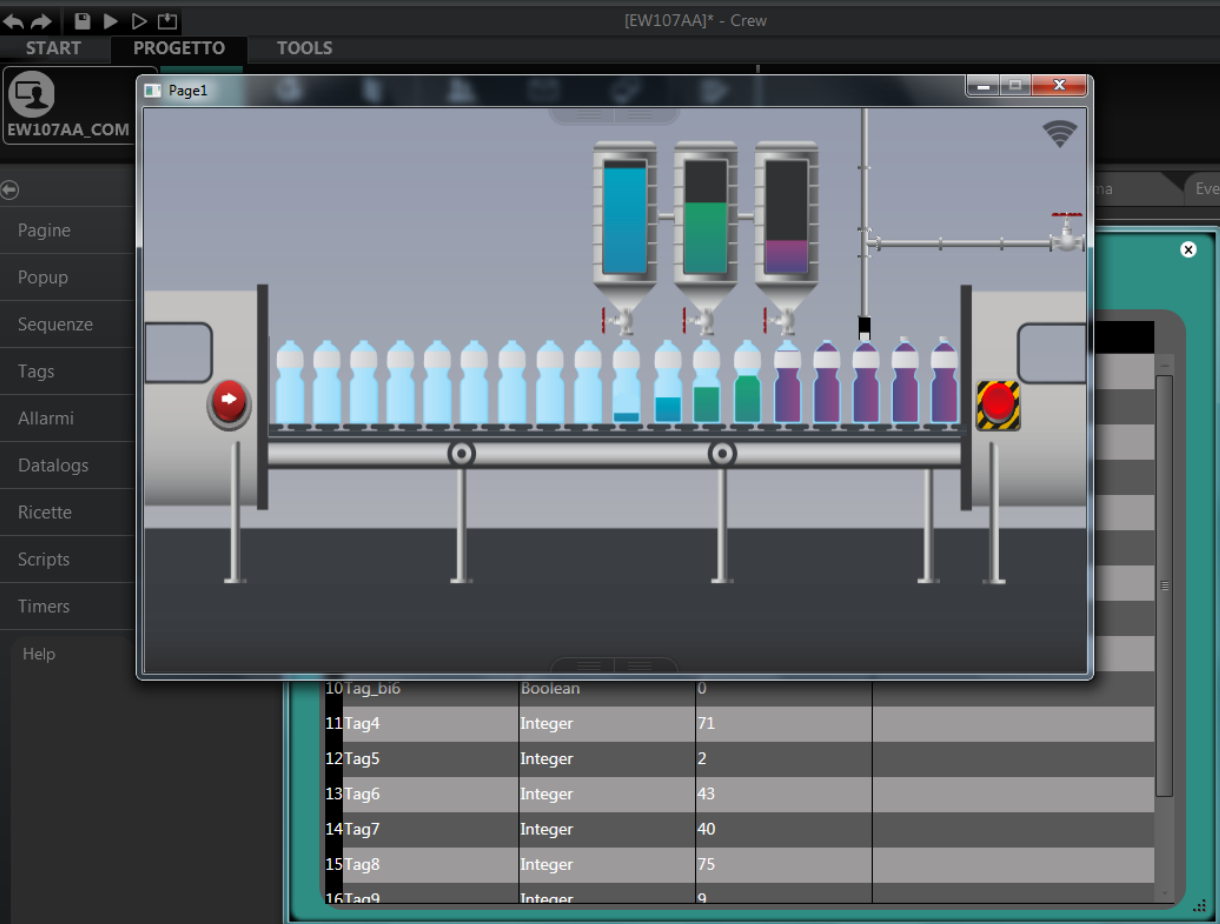


Online simulation of the project makes it possible to test project operation with the associated PLC/s. Unlike standard simulation, all of the device's variables and correct project operation can be verified with the connected device. Communication with the device/s is tested by enabling the port that each device is connected to.



# CREW Manual

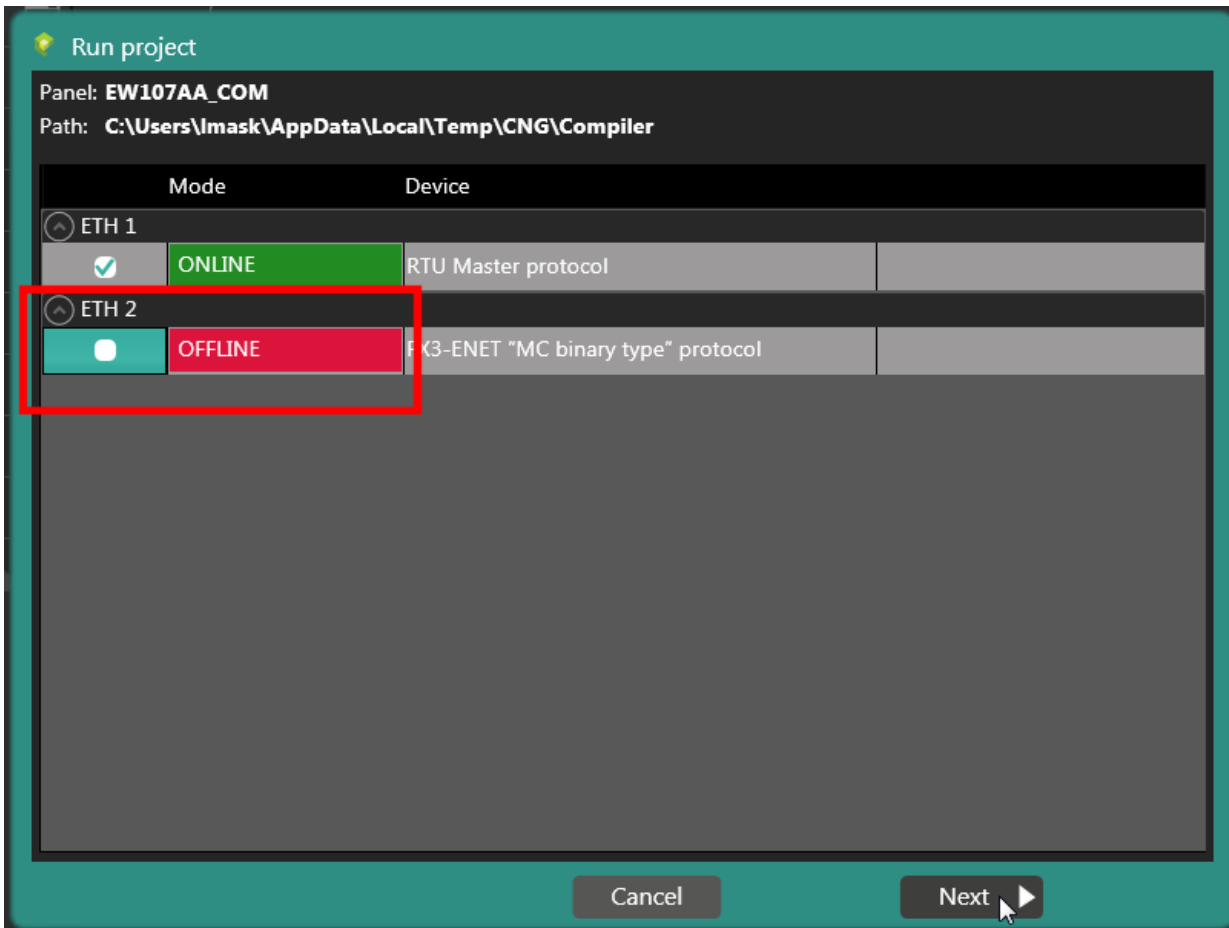
In the example it is possible to observe that, with both ETH ports enabled, the two associated devices are connected correctly and are therefore online. Simply press the “Next” key to start project simulation in Runtime.



10 Tag_bi6	Boolean	0
11 Tag4	Integer	71
12 Tag5	Integer	2
13 Tag6	Integer	43
14 Tag7	Integer	40
15 Tag8	Integer	75
16 Tag9	Integer	9

# CREW Manual

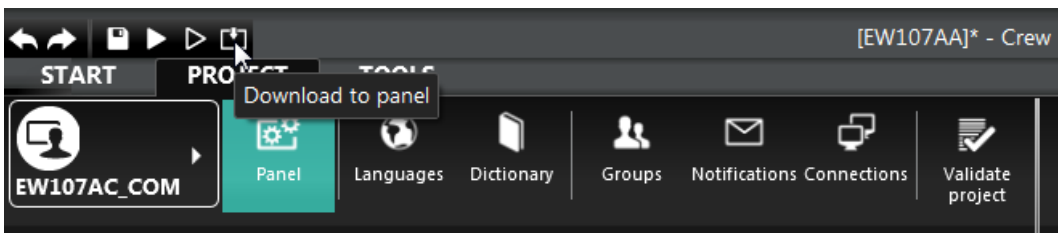
If, on the other hand, you wish to disable the connection with the device connected to the panel's second ETH port, disable the checkbox to set the device connected to that port as offline.



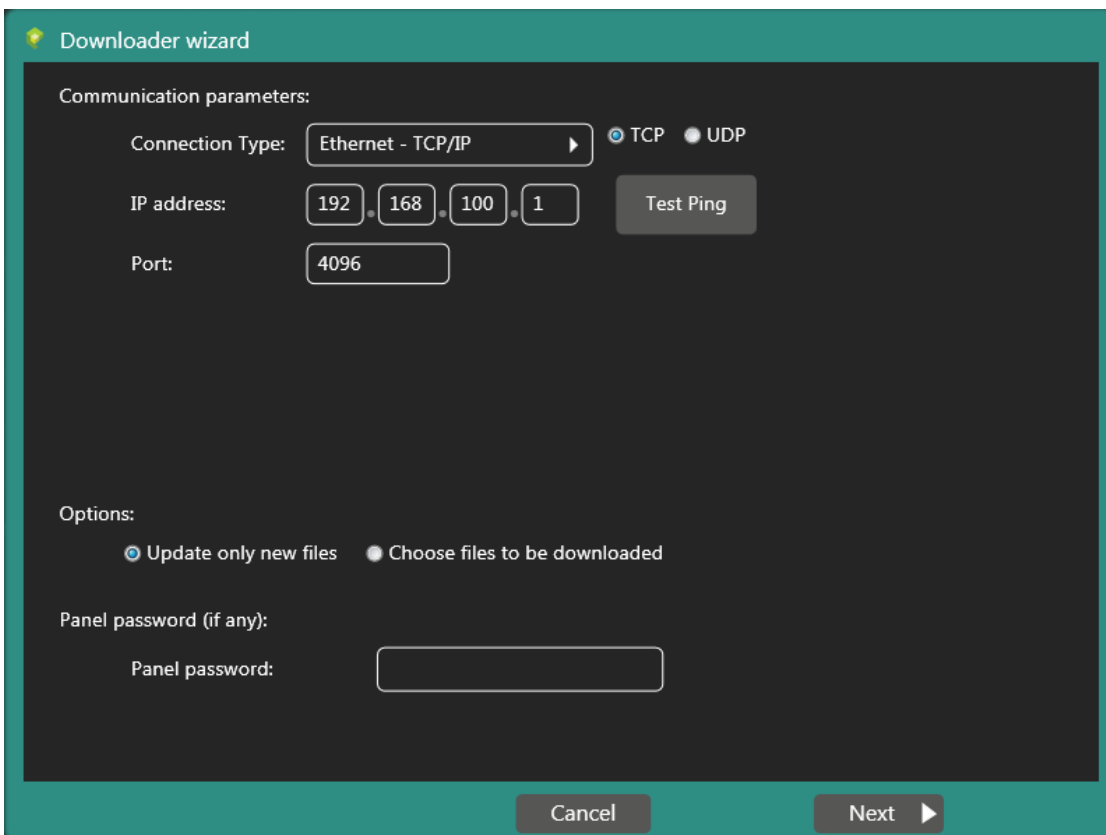
# CREW Manual

## Project download

When a project is filled in, it is ready for transfer to the terminal through the “Download” function. To start the transfer it is sufficient to click the icon on the tool bar.



In the download window, it is possible to select the parameters relative to the type of connection to be used.





# CREW Manual

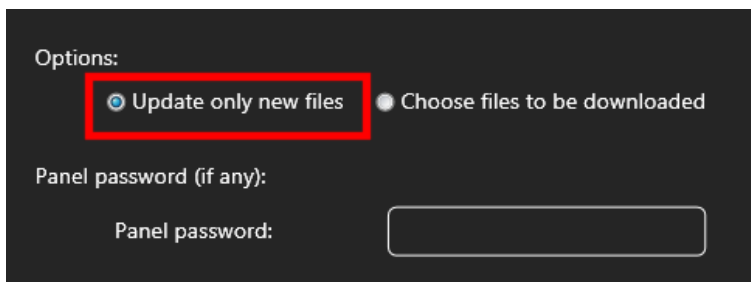
The available connection types are:

- “Ethernet - TCP/IP” (see “[Project download - Ethernet TCP IP](#)” section)
- “Local”
- “USB” (see “[Project download - USB](#)” section)



Note: To transfer in “Local” connection mode, it is necessary to have Crew “Runtime” installed on the PC. Crew Runtime is included in the purchased licence (CRRTxxxxxxx).

Regardless of the type of connection it is possible to decide whether to perform a full update of the elements on the panel (firmware, project, and all other components) or whether to download only the elements that need updating.

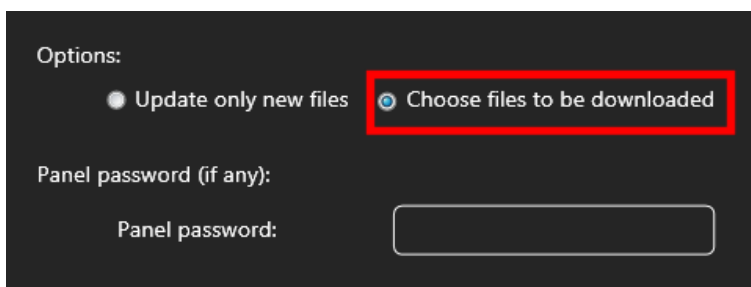


Options:

Update only new files  Choose files to be downloaded

Panel password (if any):

Panel password:



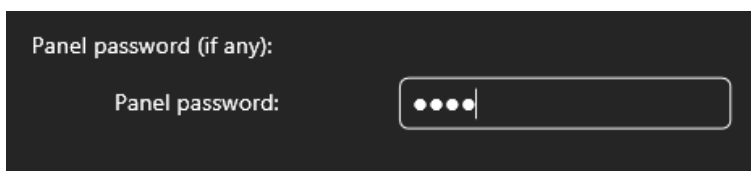
Options:

Update only new files  Choose files to be downloaded

Panel password (if any):

Panel password:

Enter the panel password if required.

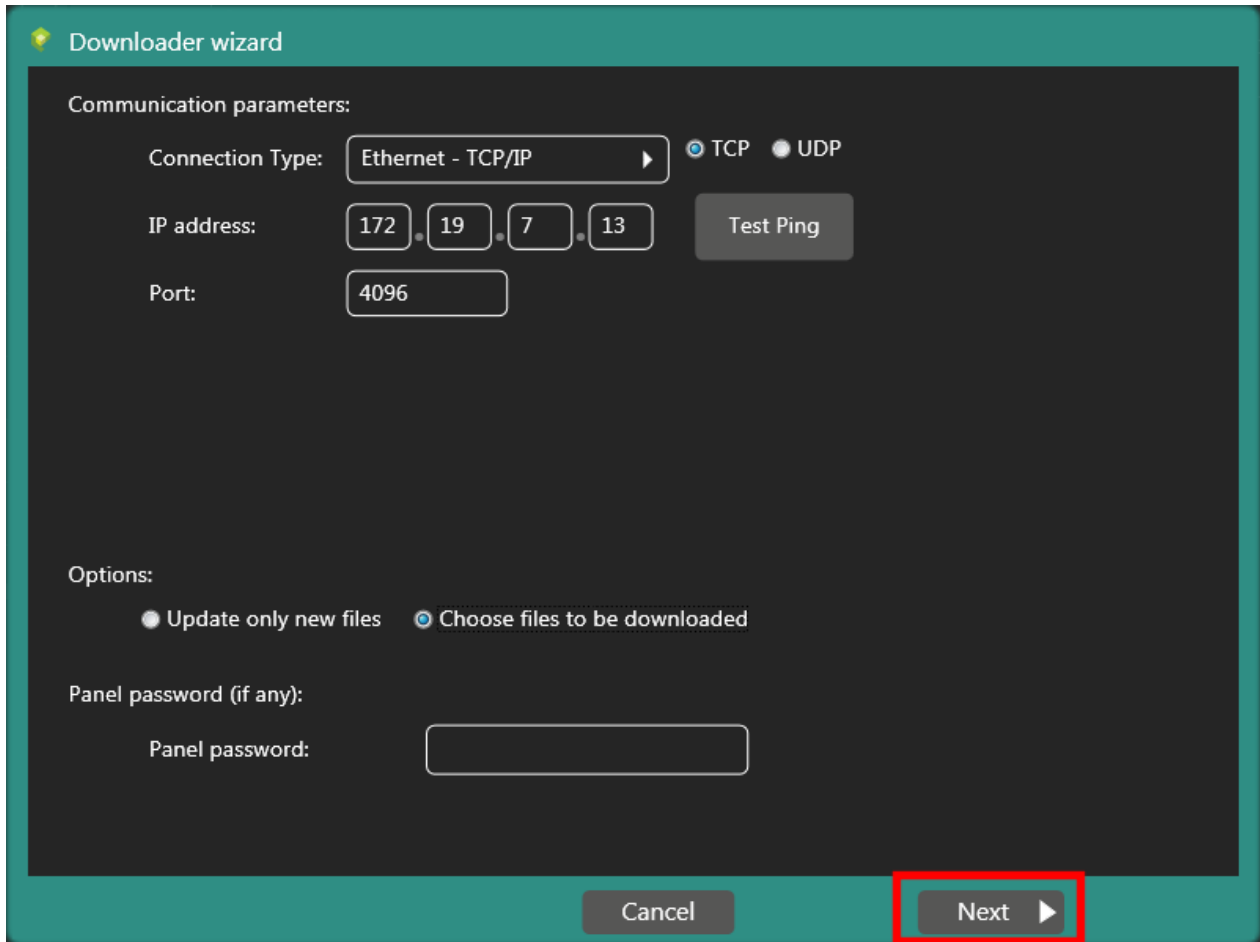


Panel password (if any):

Panel password:

# CREW Manual

Click "Next".



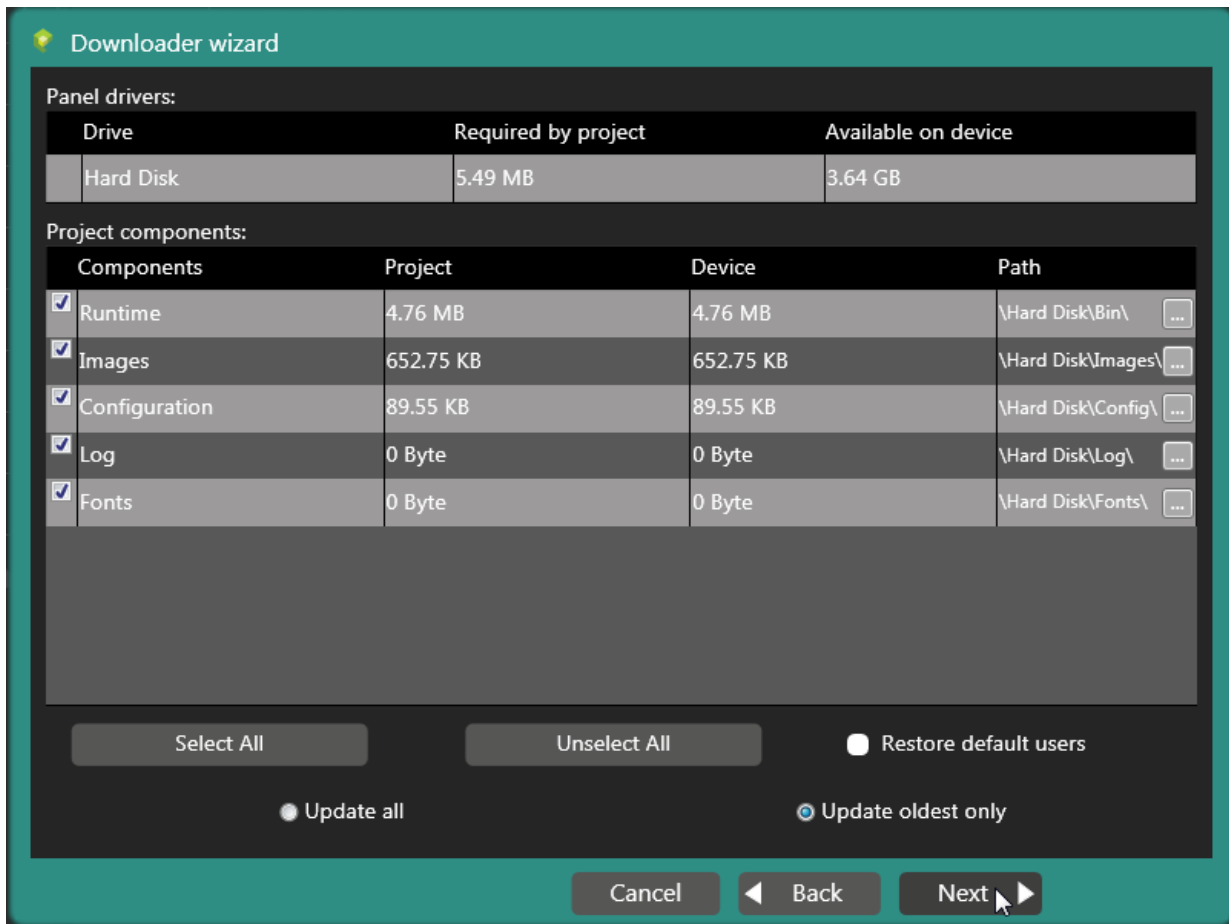
The image shows a 'Downloader wizard' dialog box with a dark background and teal borders. It contains the following fields and controls:

- Communication parameters:**
  - Connection Type: A dropdown menu showing 'Ethernet - TCP/IP'.
  - Protocol: Radio buttons for 'TCP' (selected) and 'UDP'.
  - IP address: Four input boxes containing '172', '19', '7', and '13', with a 'Test Ping' button to the right.
  - Port: An input box containing '4096'.
- Options:**
  - Radio buttons for 'Update only new files' and 'Choose files to be downloaded' (selected).
- Panel password (if any):**
  - An input box for the password.

At the bottom, there are two buttons: 'Cancel' and 'Next'. The 'Next' button is highlighted with a red rectangular box.

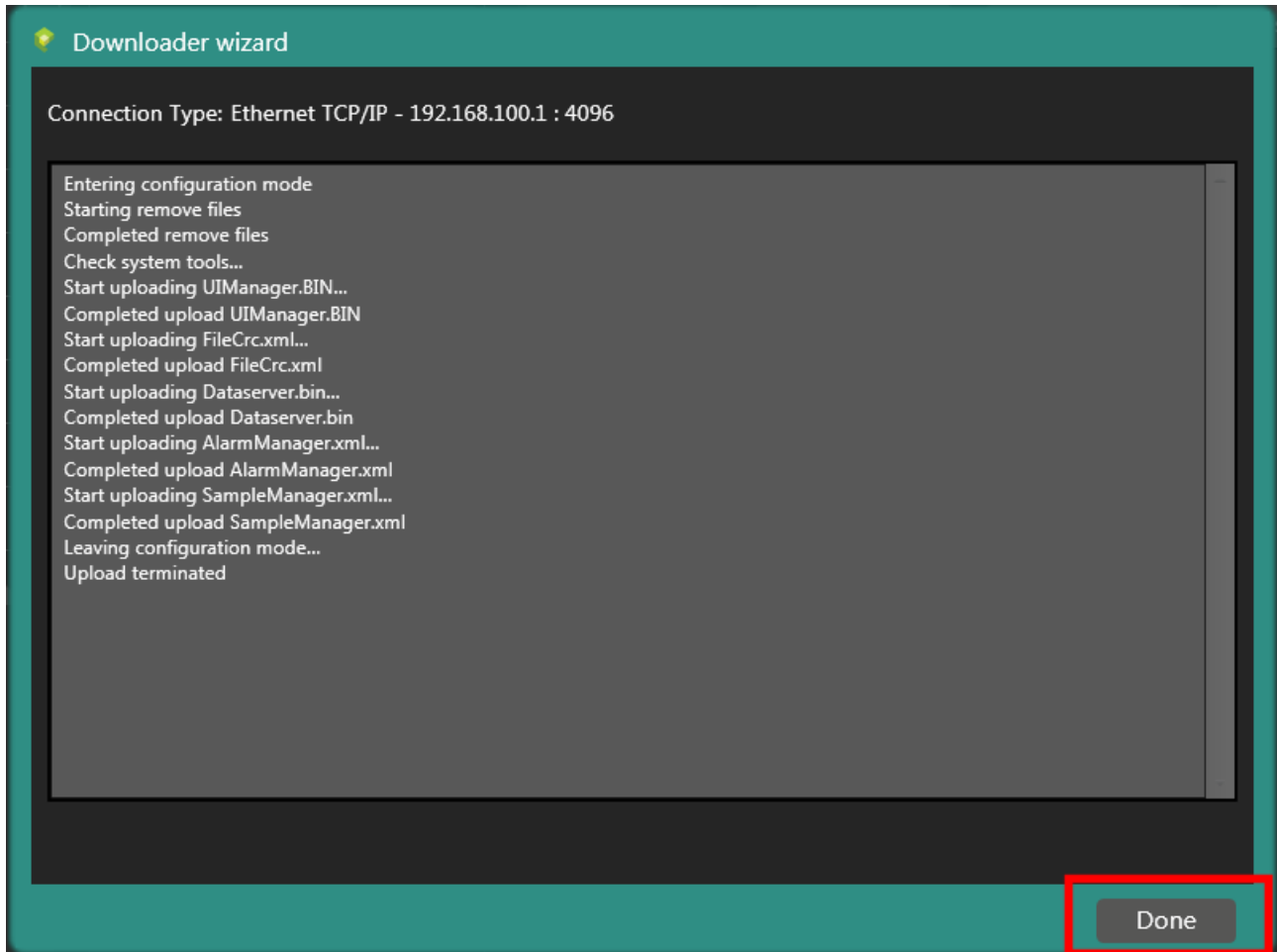
# CREW Manual

To choose which files to download, select the necessary items from the window that appears.



# CREW Manual

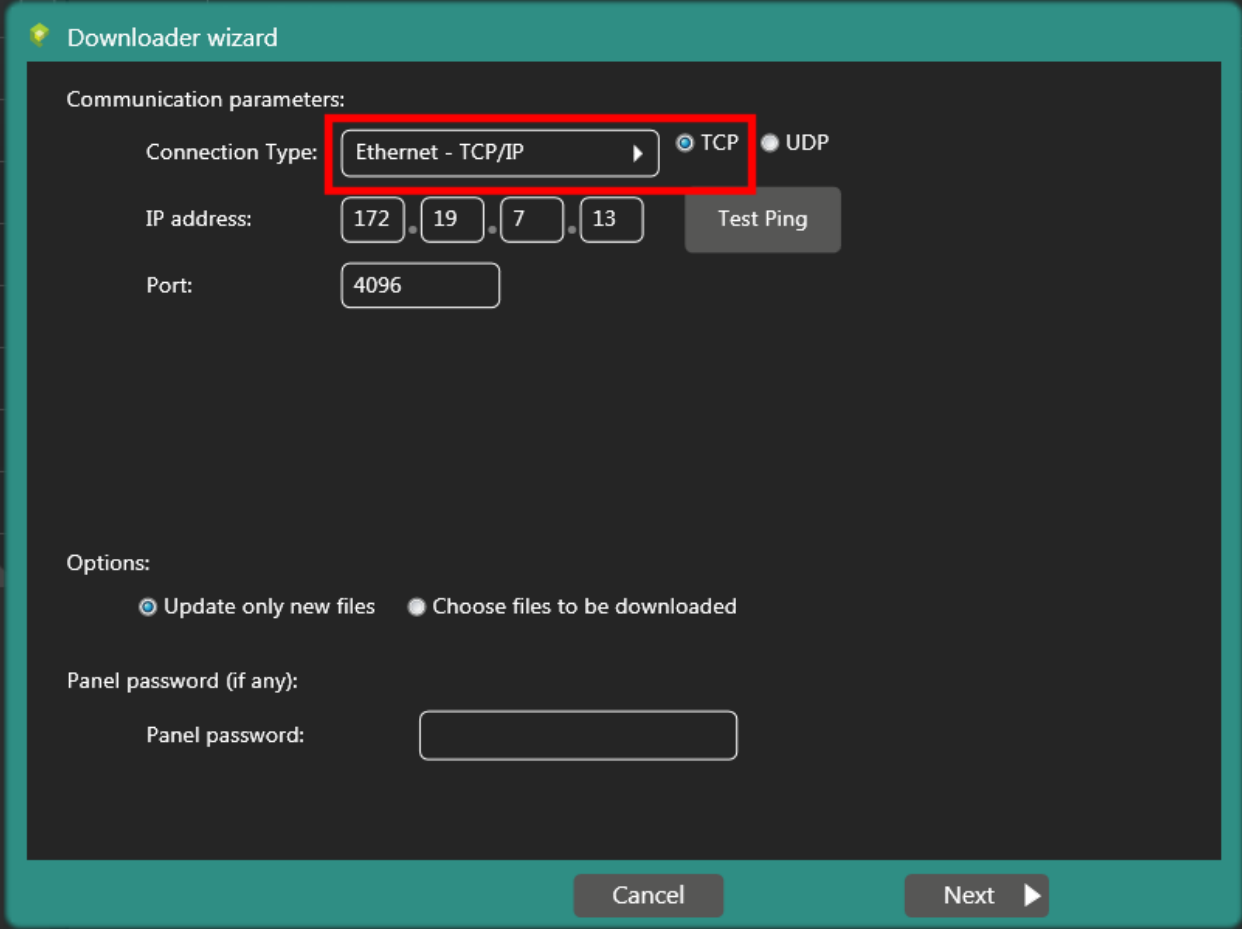
At the end click “Done”.



# CREW Manual

Project download - Ethernet TCP IP -

TCP / IP - TCP -



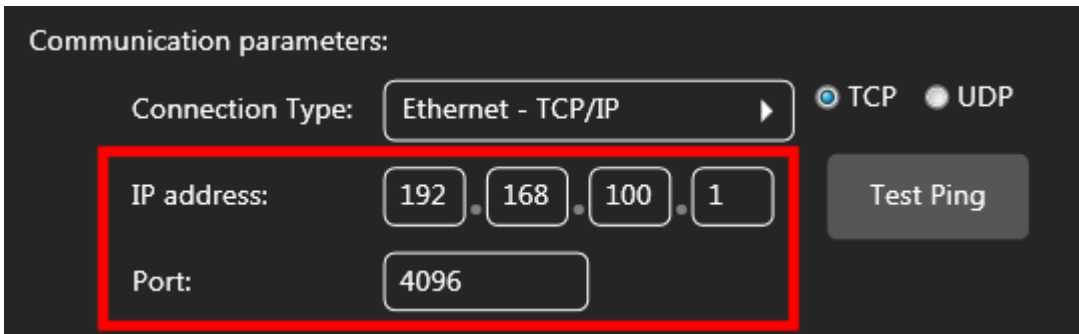
The screenshot shows a 'Downloader wizard' window with a dark background and teal borders. The title bar contains a yellow star icon and the text 'Downloader wizard'. The main content area is divided into sections:

- Communication parameters:**
  - Connection Type:** A dropdown menu is set to 'Ethernet - TCP/IP', which is highlighted with a red rectangular box. To its right are radio buttons for 'TCP' (selected) and 'UDP'.
  - IP address:** Four input fields contain the numbers '172', '19', '7', and '13' respectively, separated by dots. A 'Test Ping' button is located to the right of these fields.
  - Port:** An input field contains the number '4096'.
- Options:**
  - Two radio buttons are present: 'Update only new files' (selected) and 'Choose files to be downloaded'.
- Panel password (if any):**
  - An input field labeled 'Panel password:' is currently empty.

At the bottom of the window, there are two buttons: 'Cancel' on the left and 'Next' with a right-pointing arrow on the right.

# CREW Manual

To choose the “Ethernet - TCP/IP” connection it is necessary to specify the parameters to set up the connection, such as IP address and communication port.



Communication parameters:

Connection Type:   TCP  UDP

IP address:  .  .  .

Port:



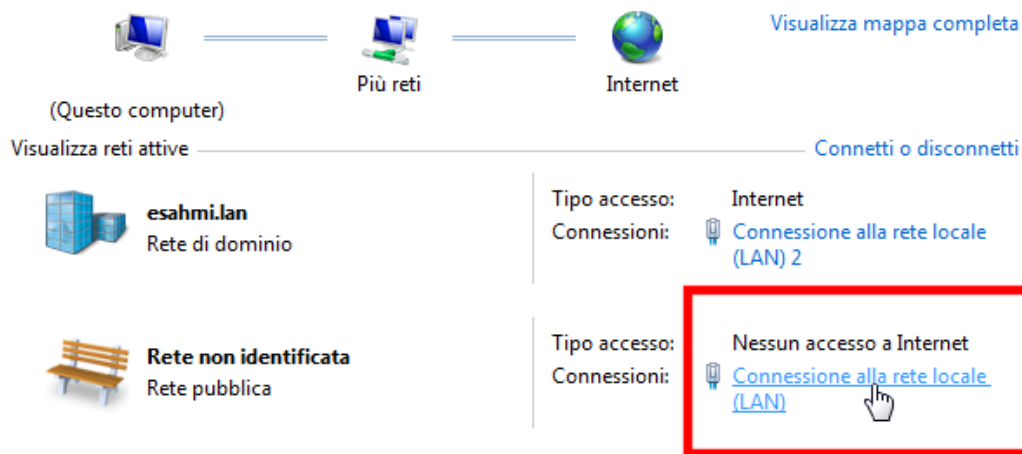
Note: The IP address set on the terminal must match the one on the PC that is used to transfer the project through Crew.

# CREW Manual

## IP address settings on PC side

Below is a brief description of the operations to be carried out to set a correct IP address on the PC.





### Visualizzare le informazioni di base sulla rete e configurare le connessioni



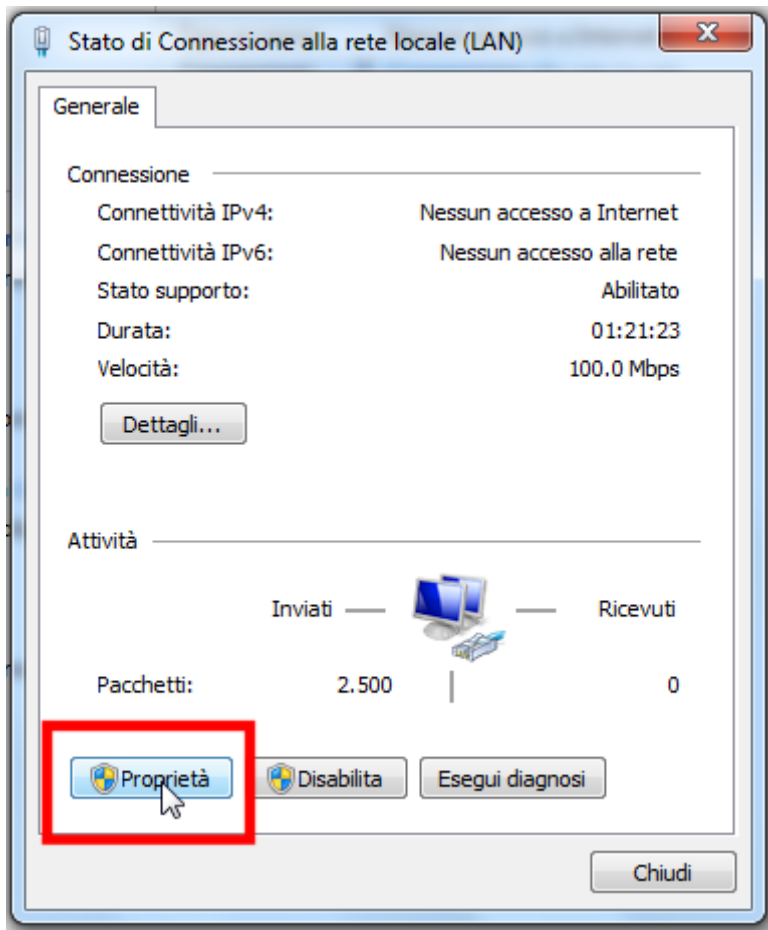
The screenshot shows the Windows Network and Sharing Center. At the top, there is a diagram with three icons: a computer labeled '(Questo computer)', a group of computers labeled 'Più reti', and a globe labeled 'Internet'. To the right of the diagram is a link 'Visualizza mappa completa'. Below the diagram, there are two tabs: 'Visualizza reti attive' (selected) and 'Connetti o disconnetti'. Under 'Visualizza reti attive', there are two network profiles:

- esahmi.lan** (Rete di dominio):
  - Tipo accesso: Internet
  - Connessioni: [Connessione alla rete locale \(LAN\) 2](#)
- Rete non identificata** (Rete pubblica):
  - Tipo accesso: Nessun accesso a Internet
  - Connessioni: [Connessione alla rete locale \(LAN\)](#) (highlighted with a red box)

### Modifica impostazioni di rete

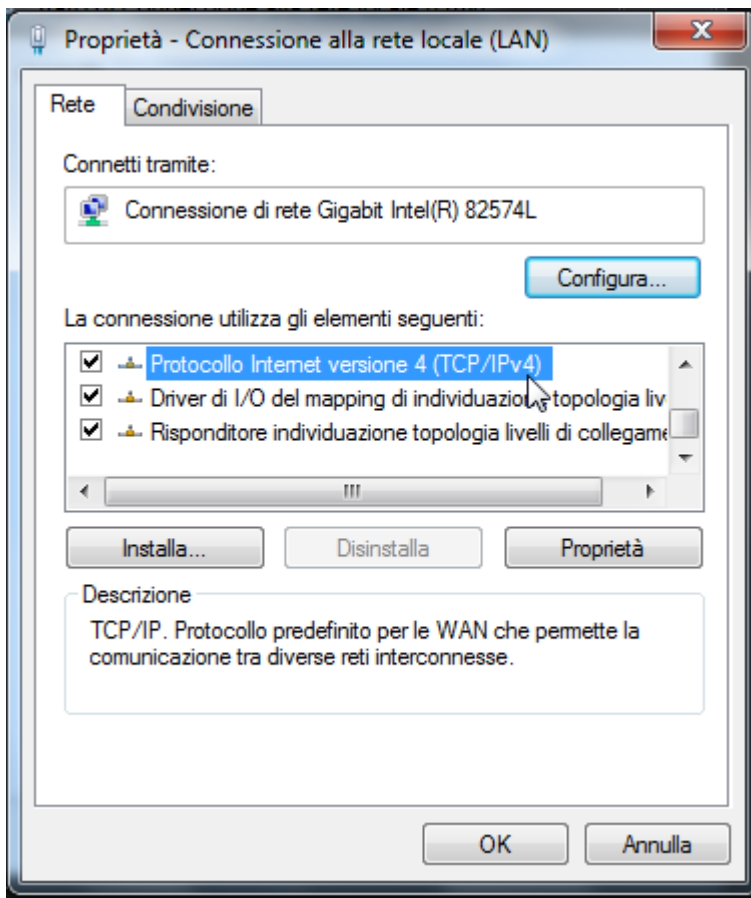
-  [Configura nuova connessione o rete](#)  
Configurare una connessione wireless, a banda larga, remota, ad hoc o VPN oppure configurare un router o un punto di accesso.
-  [Connessione a una rete](#)  
Connettere o riconnettere una connessione di rete wireless, cablata, remota o VPN.
-  [Selezione gruppo home e opzioni di condivisione](#)  
Accedere ai file e alle stampanti disponibili in altri computer della rete oppure modificare le impostazioni di condivisione.
-  [Risoluzione problemi](#)  
Eeguire la diagnosi e la correzione di problemi di rete oppure ottenere informazioni per la risoluzione dei problemi.

# CREW Manual

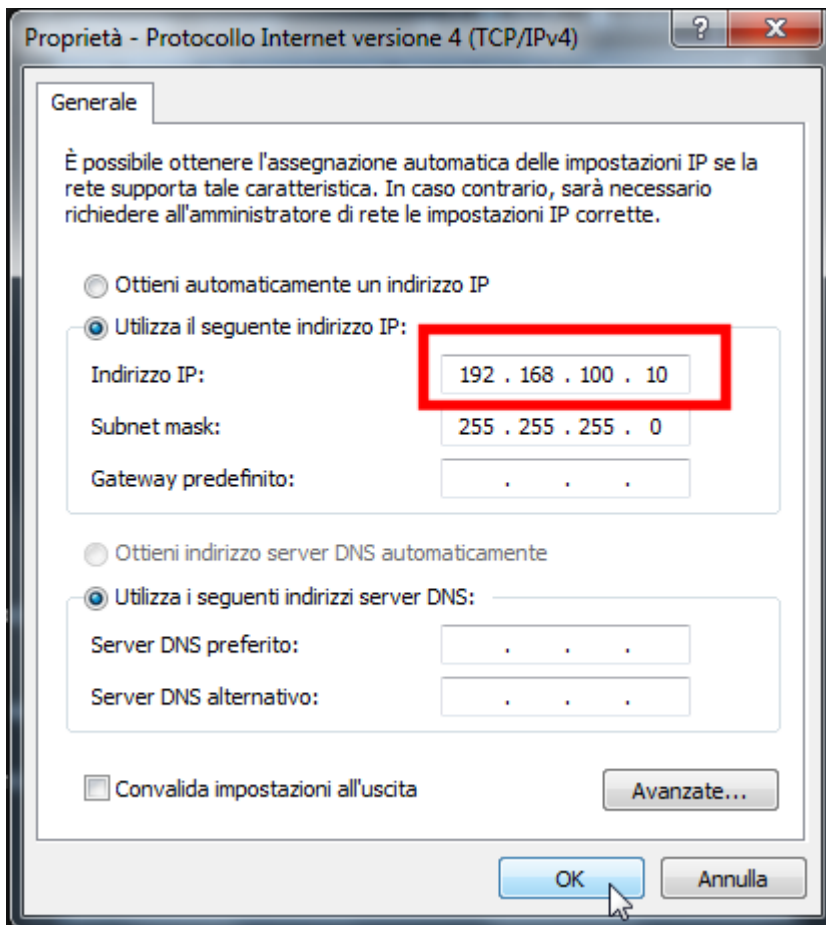




# CREW Manual



# CREW Manual

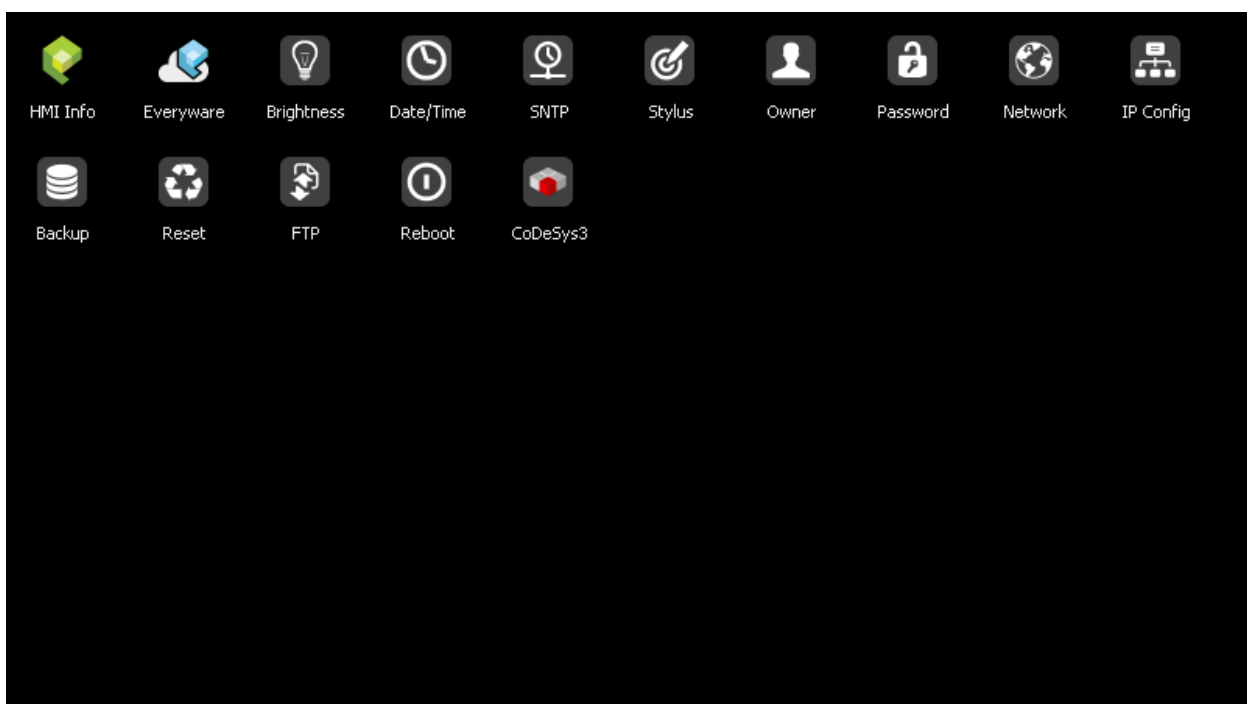


# CREW Manual

## IP address settings on EW side

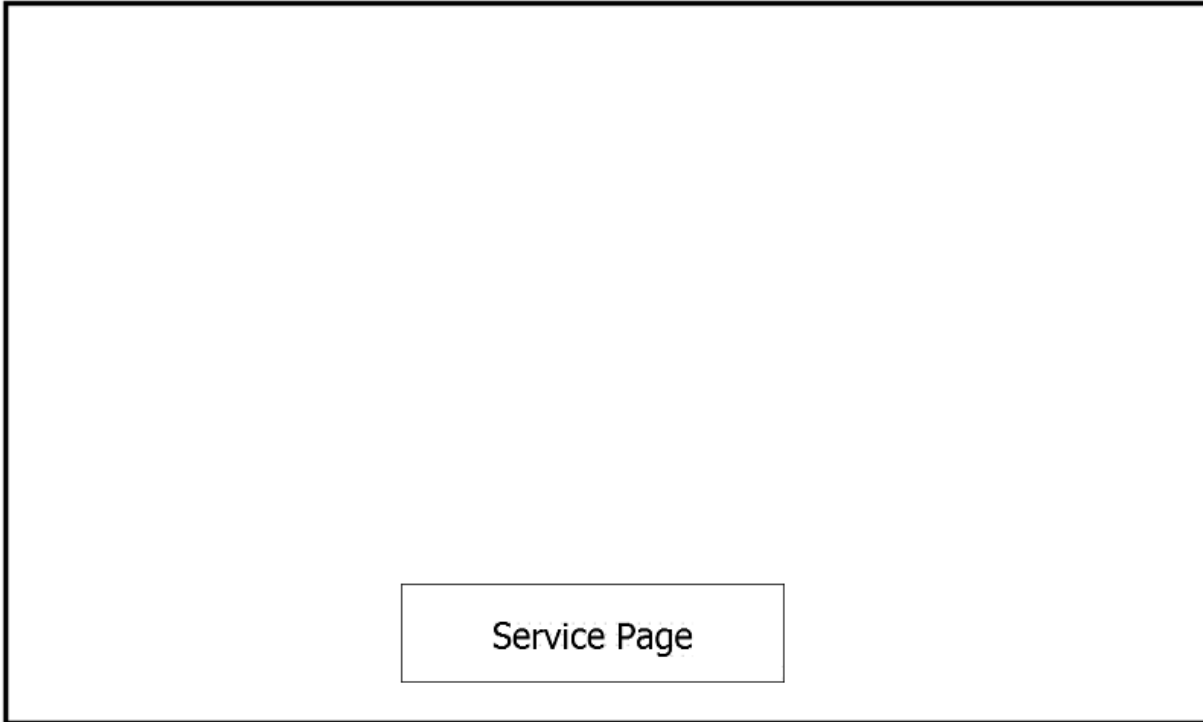
Below is a brief description of the operations to be carried out to set a correct IP address on the terminal.

Enter the terminal's control panel.



# CREW Manual

If no project has been transferred to the terminal yet, to enter the control panel simply click the “Service page” key.



# CREW Manual

Then click "Control panel".



## CREW Manual

To set up a key to enter the control panel (if this has not been done yet), press the touch screen for approximately 10 seconds, while the terminal is being powered.



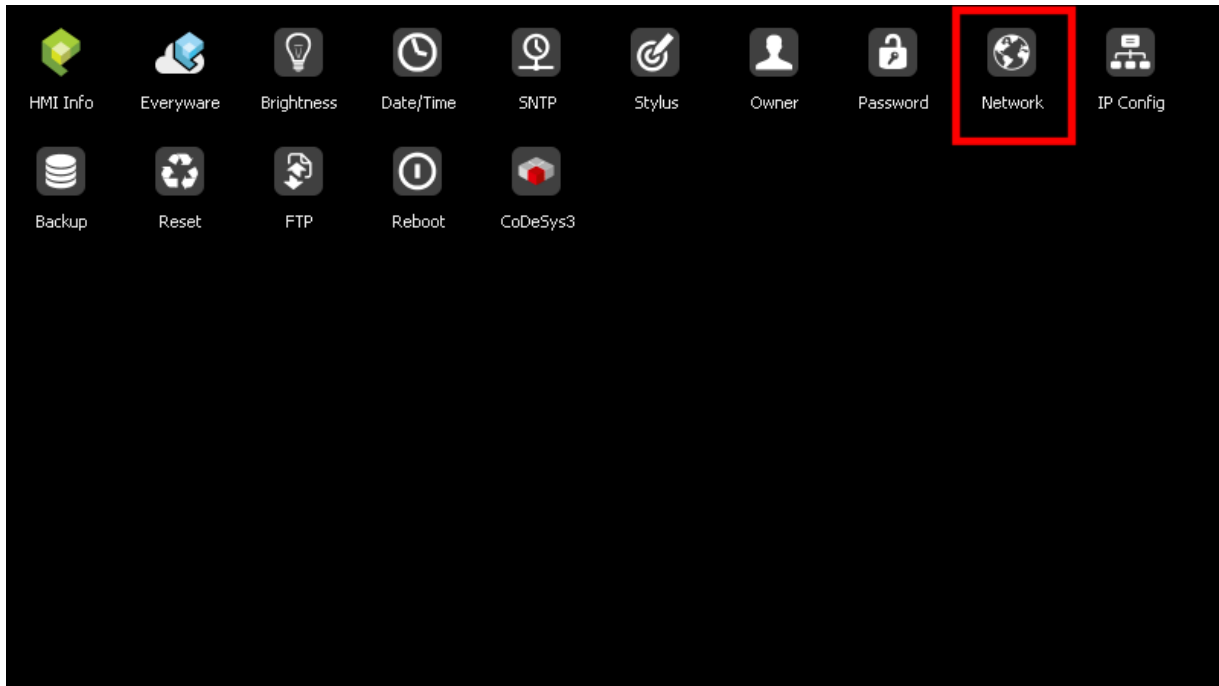
# CREW Manual

Accordingly you stop the project from starting up and the service page appears, where it is possible to access the control panel.

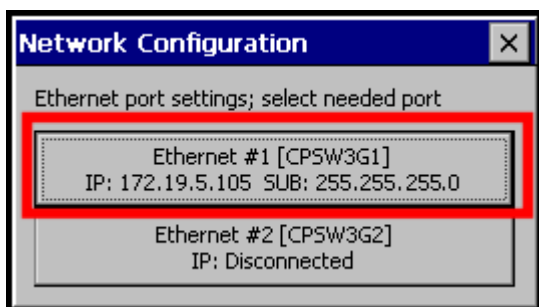


# CREW Manual

From the control panel, click on the “Network” icon.



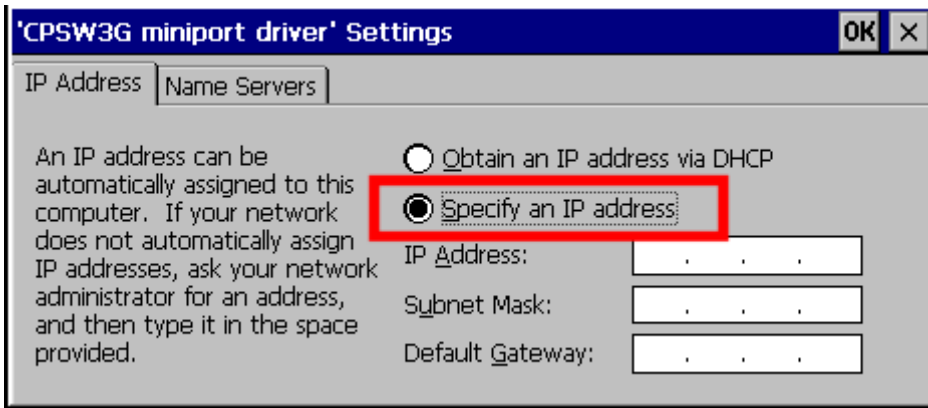
Then click on the connected Ethernet port.



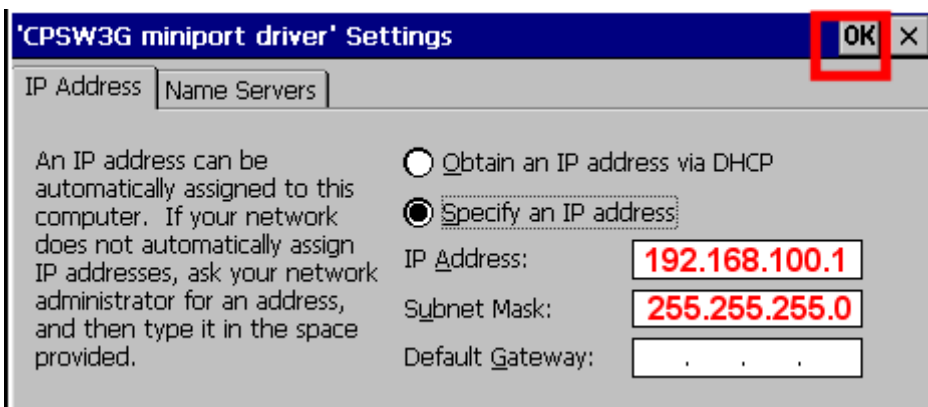


# CREW Manual

Select the “Specify an IP address” option.

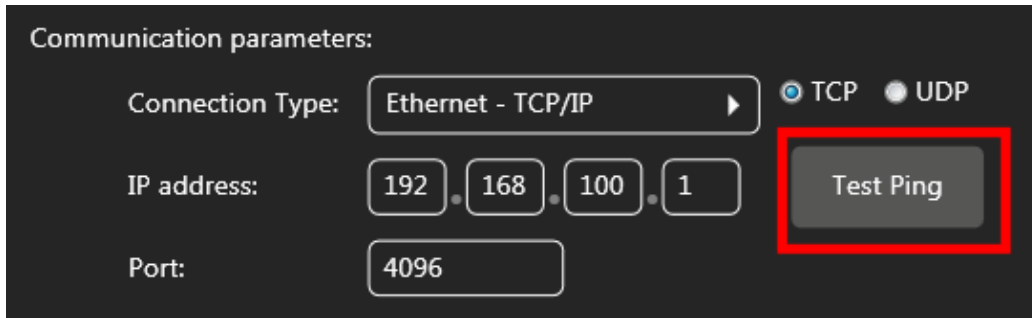


Assign an IP address that is compatible with the previously set one, and click “Ok” to confirm.

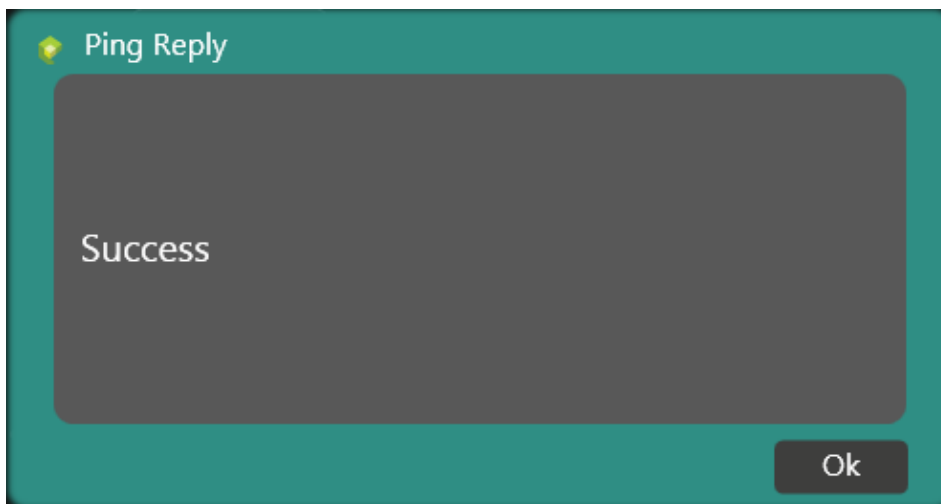


# CREW Manual

It is possible to run a “Test Ping” to check the coherency of the entered values.



If the entered values are coherent, the “correct response to Ping” message appears.



# CREW Manual

After setting the IP address it is necessary to select the transfer method on the panel as well (which in turn needs to be selected in Crew). Go to the service page.

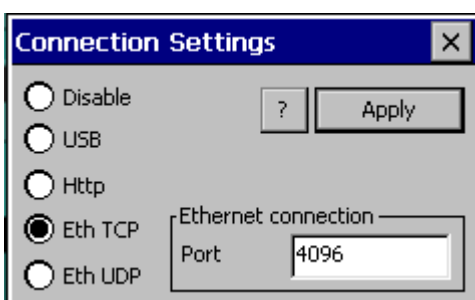


# CREW Manual

Select "Downloader Configuration".

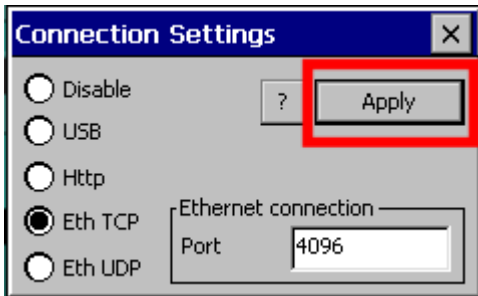


Select "Eth TCP".

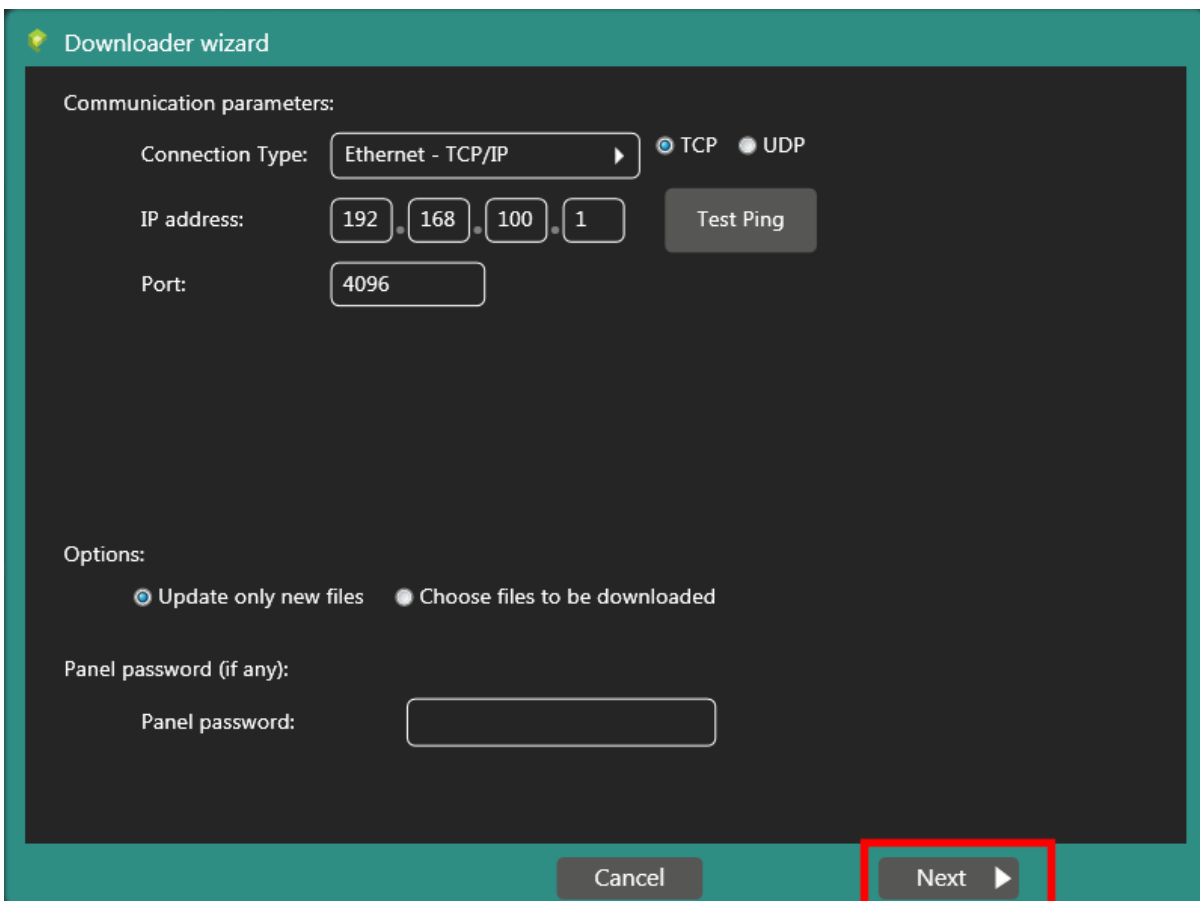


# CREW Manual

Click “Apply” to confirm the setting.

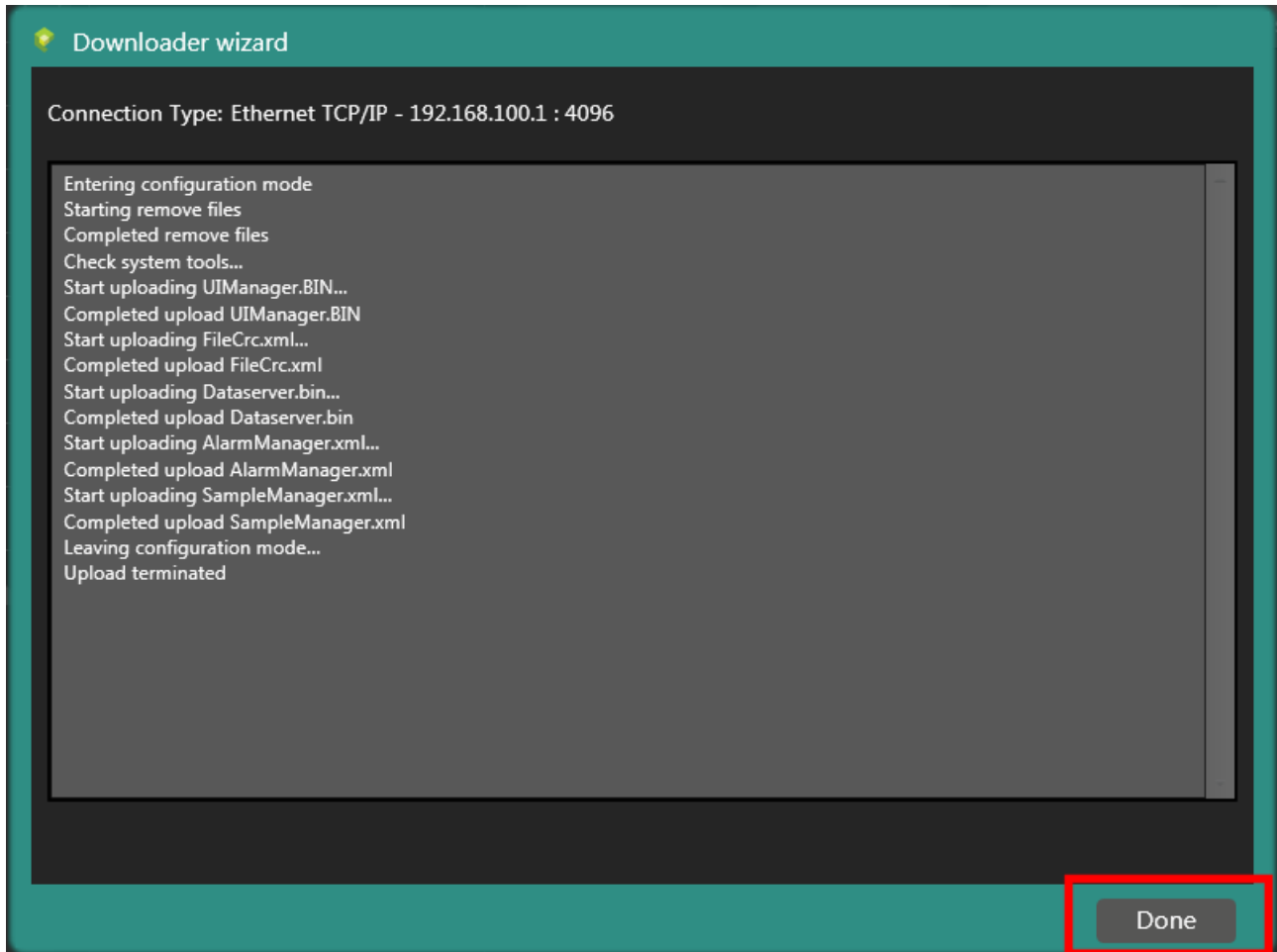


When you have correctly set all of the parameters, click “Next” to continue downloading the project.



# CREW Manual

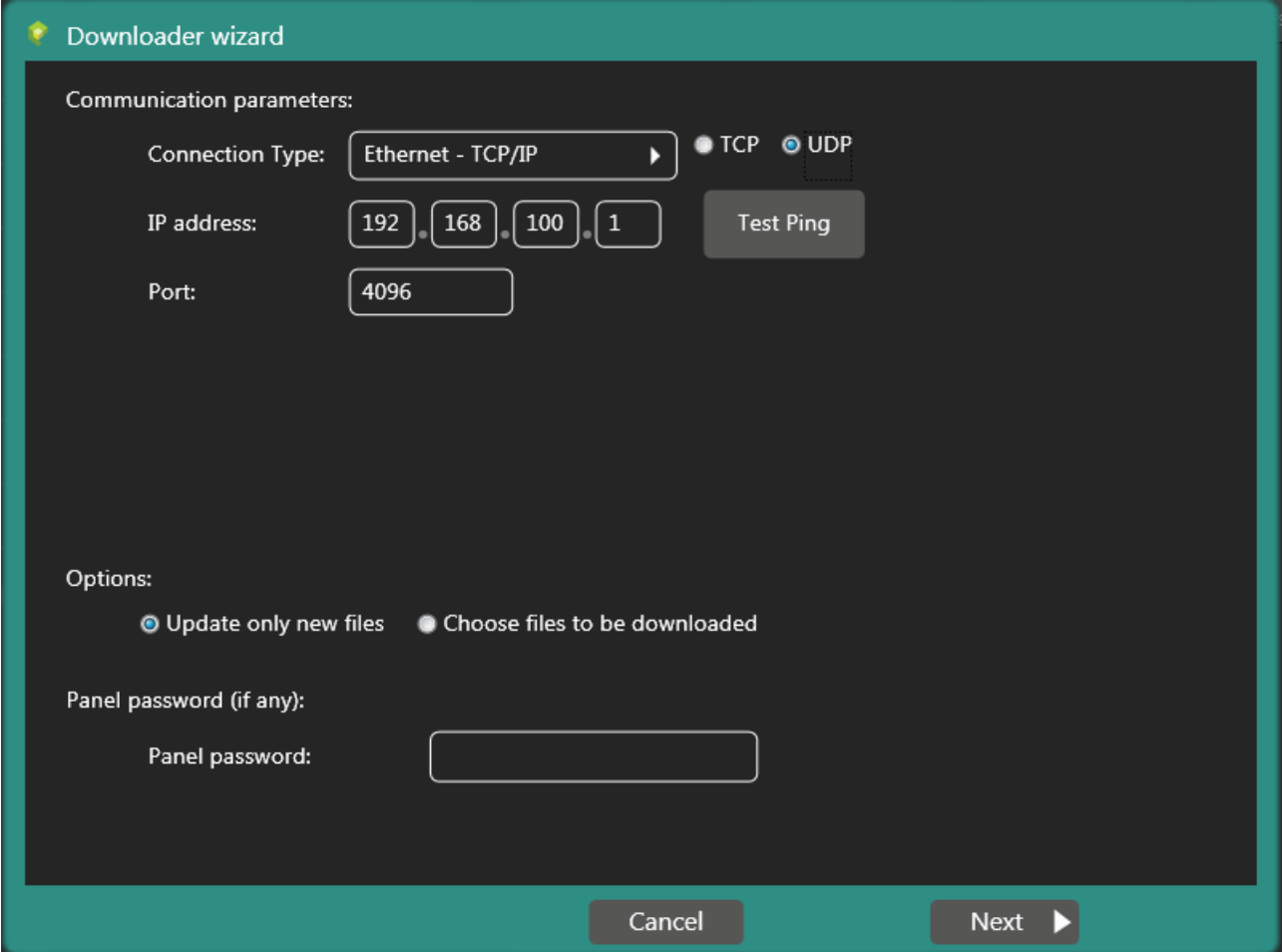
At the end click “Done”.



# CREW Manual

## Download project TCP/IP - UDP -

There is also a second Download protocol referred to as “UDP” (User Data Protocol).



The screenshot shows a 'Downloader wizard' dialog box with a dark background and teal accents. It is divided into two main sections: 'Communication parameters' and 'Options'. In the 'Communication parameters' section, the 'Connection Type' is set to 'Ethernet - TCP/IP'. To the right, there are radio buttons for 'TCP' and 'UDP', with 'UDP' being selected. Below this, the 'IP address' is entered as '192.168.100.1' in four separate input boxes, followed by a 'Test Ping' button. The 'Port' is set to '4096'. In the 'Options' section, there are two radio buttons: 'Update only new files' (selected) and 'Choose files to be downloaded'. At the bottom, there is a 'Panel password (if any):' label and an empty password input field. At the very bottom of the dialog, there are 'Cancel' and 'Next' buttons.



Note: The UDP protocol is faster but not as secure as TCP. This is because with UDP transmission all of the data is sent simultaneously without confirmation of correct reception, whereas with TCP transmission, each data package is sent only following confirmation of reception of the previous package.

# CREW Manual

Proceed as seen previously for the "TCP/IP - TCP -" connection.

After setting the IP address it is necessary to select the transfer method on the panel as well (which in turn needs to be selected in Crew). Go to the service page.



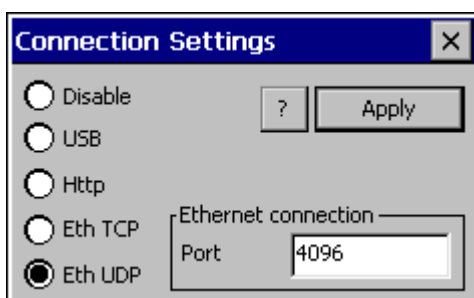


# CREW Manual

Select "Downloader Configuration".

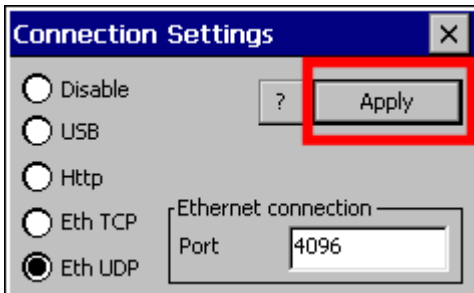


Select "Eth UDP".

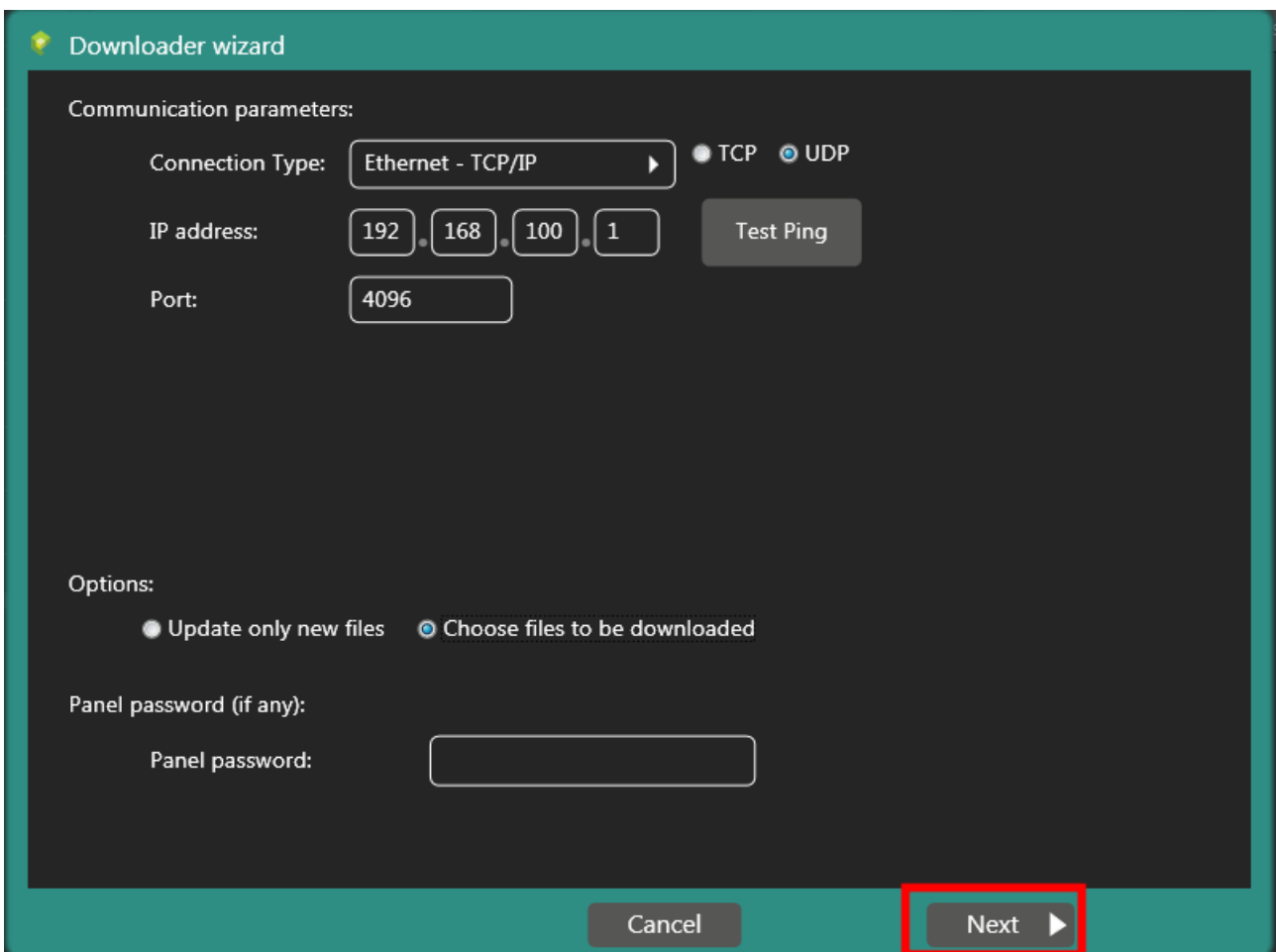


# CREW Manual

Click “Apply” to confirm the setting.

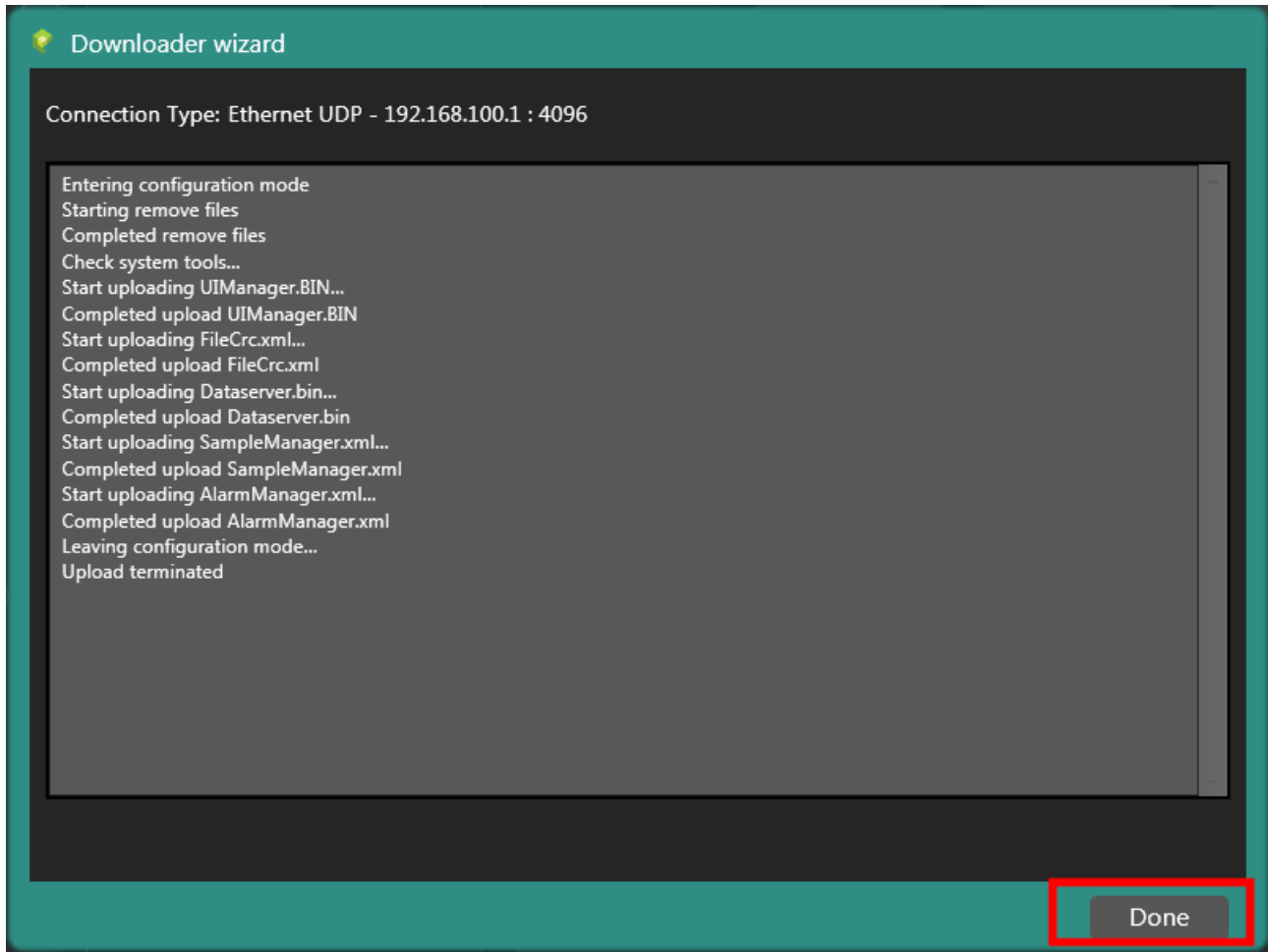


Click “Next” to continue downloading the project.



# CREW Manual

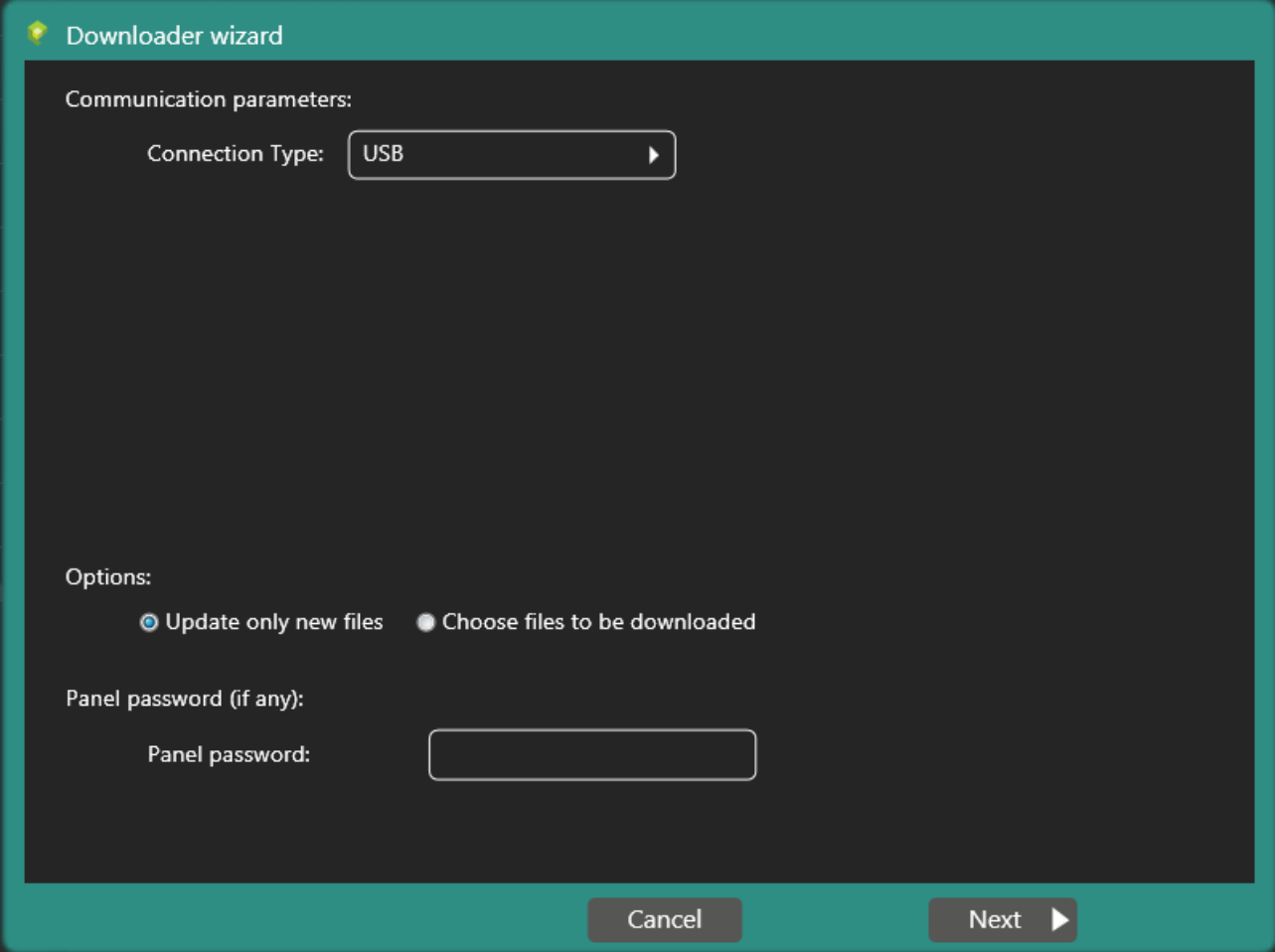
At the end click “Done”.



# CREW Manual

## Project download - USB

Select “USB” type of connection.



The screenshot shows a dialog box titled "Downloader wizard" with a teal header. The main content area is dark grey. It contains the following elements:

- Communication parameters:**
  - Connection Type: A dropdown menu with "USB" selected and a right-pointing arrow.
- Options:**
  - Update only new files: A radio button that is selected.
  - Choose files to be downloaded: A radio button that is not selected.
- Panel password (if any):**
  - Panel password: An empty text input field.

At the bottom of the dialog, there are two buttons: "Cancel" on the left and "Next" with a right-pointing arrow on the right.

# CREW Manual

Select the transfer method on the panel as well (which in turn needs to be selected in Crew). Go to the service page.

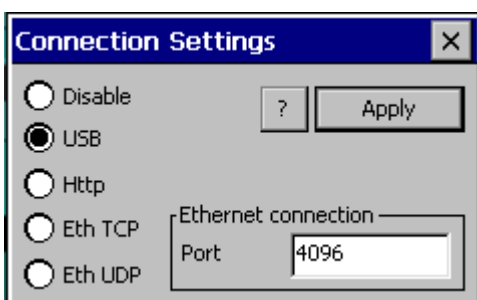


# CREW Manual

Select "Downloader Configuration".

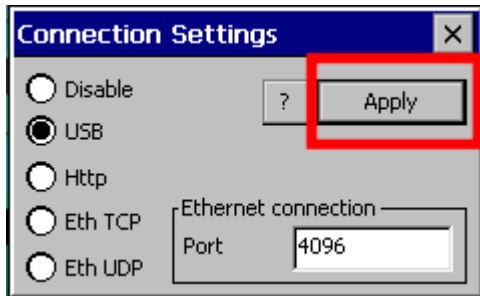


Select "USB".

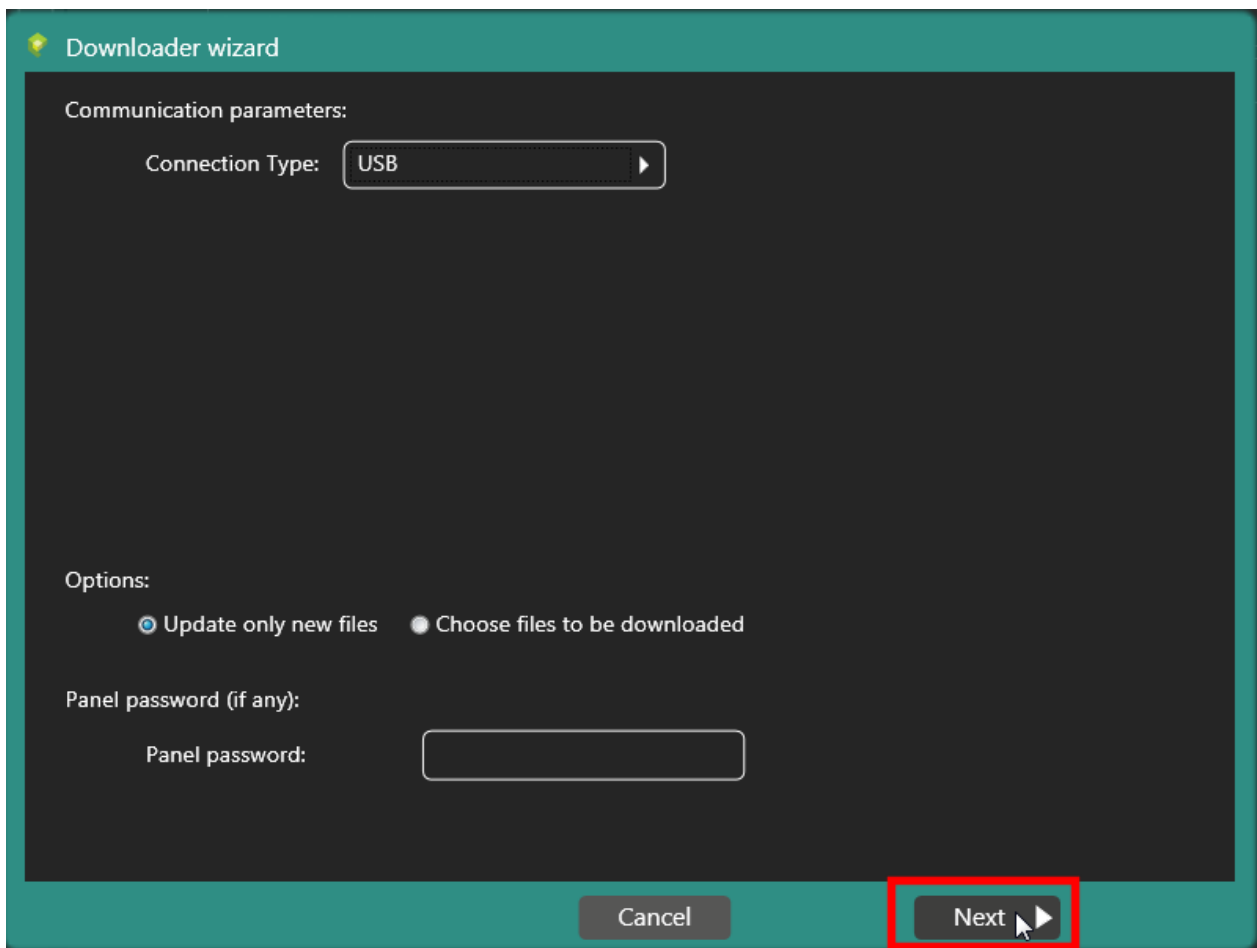


# CREW Manual

Click “Apply” to confirm the setting.

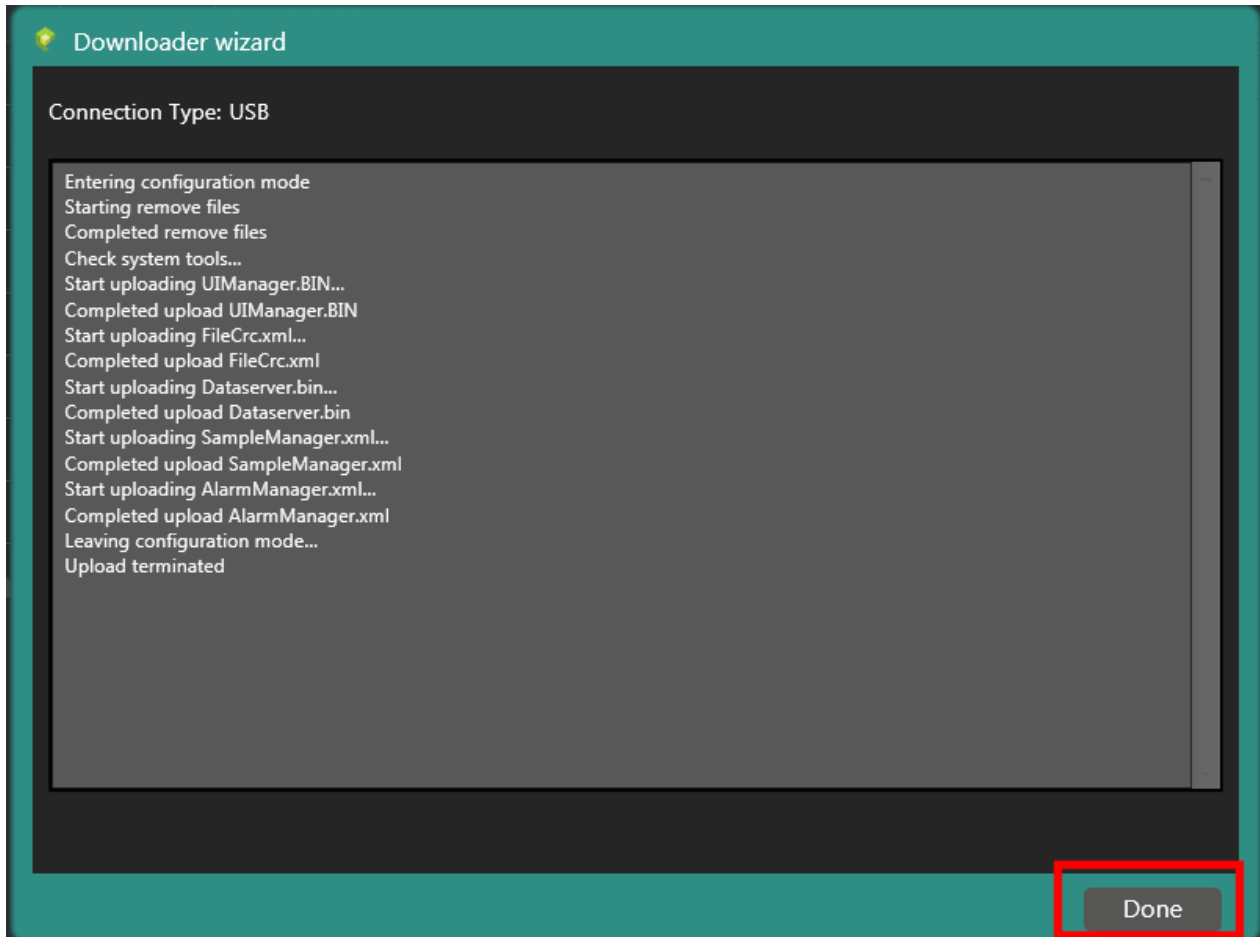


Click “Next” to continue downloading the project.



# CREW Manual

At the end click “Done”.

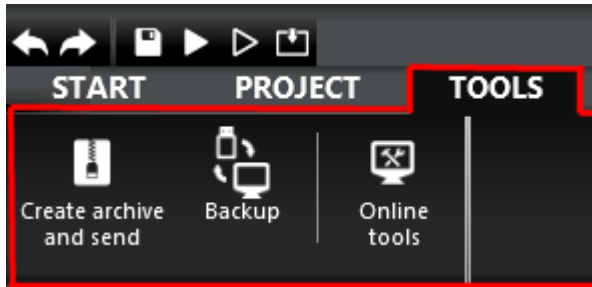




# CREW Manual

---

## TOOLS Menu



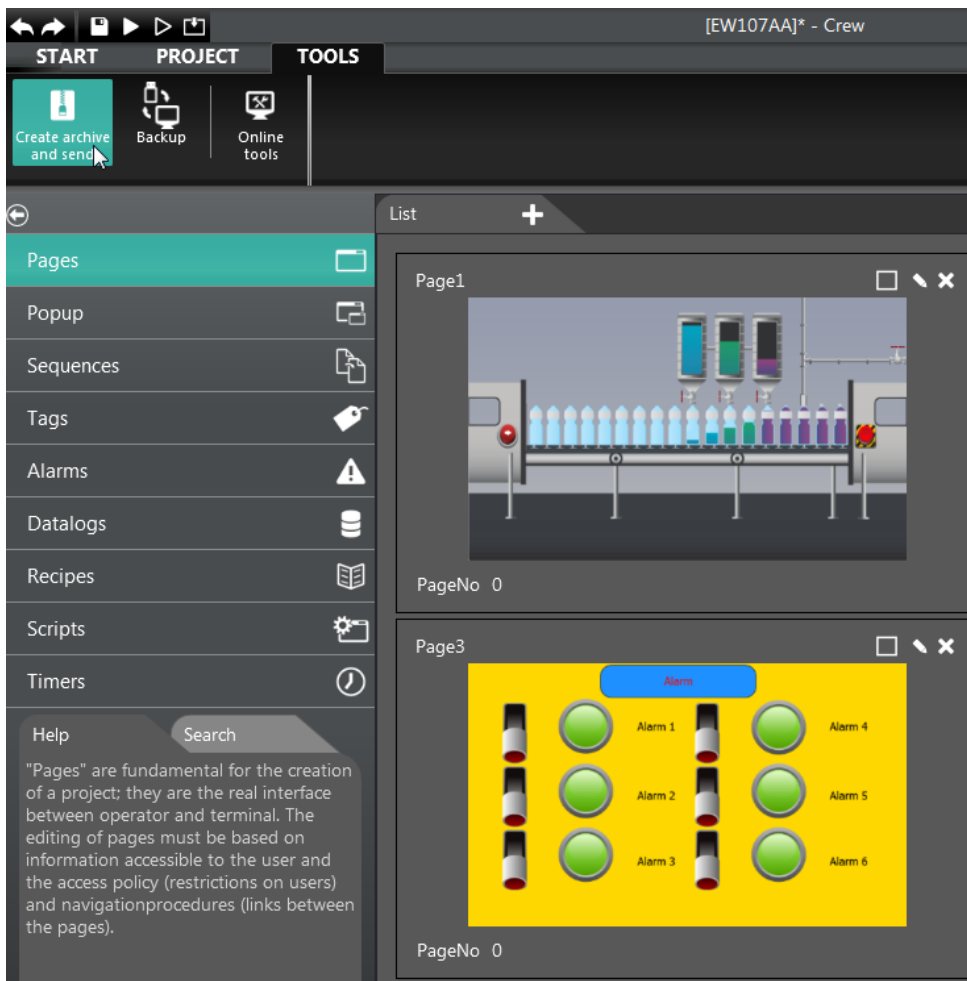
The Tools Menu is composed of the following options:

- Create archive and send
- Backup
- Online Tools

# CREW Manual

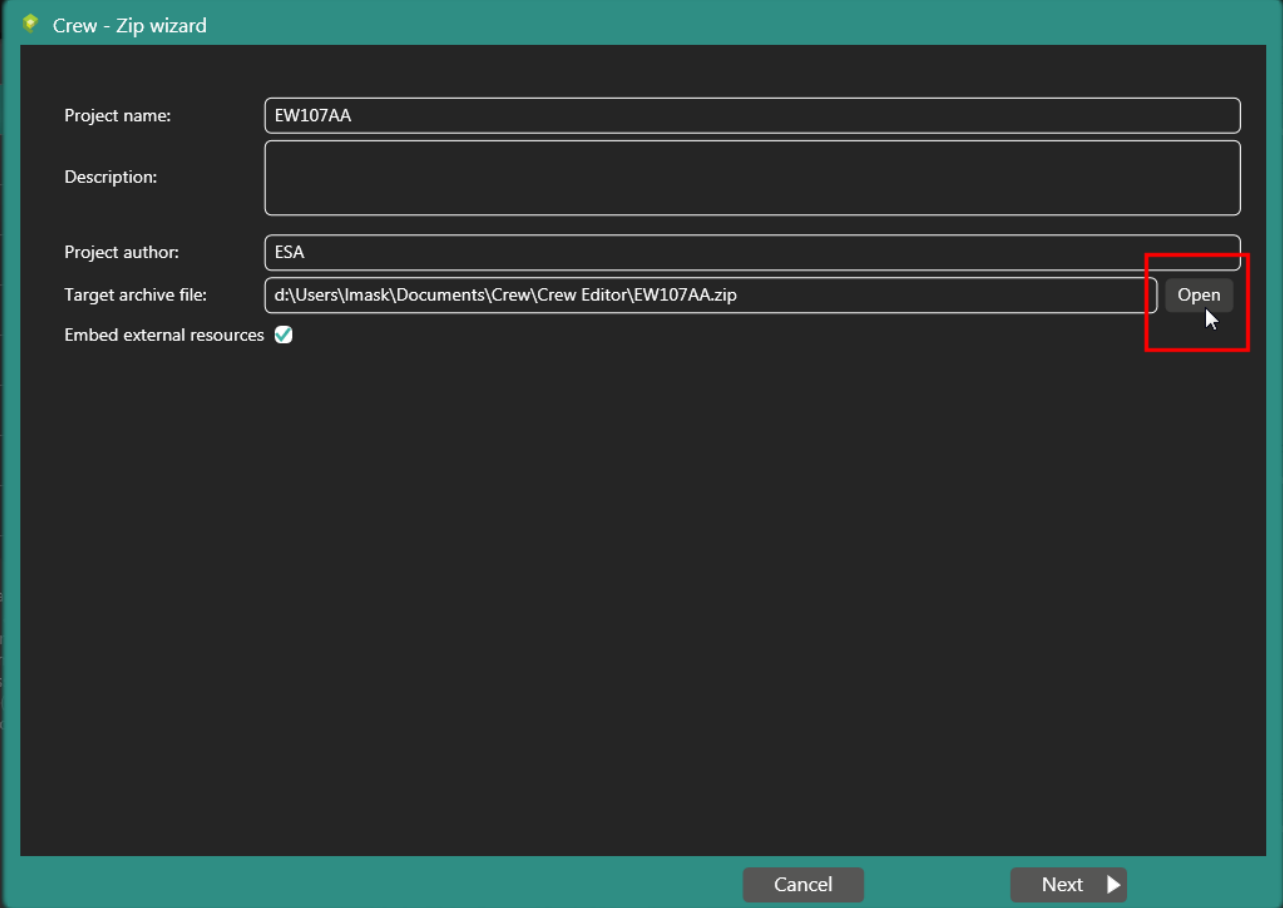
## Create archive and send

With the “Create archive and send” function it is possible to save a zipped copy of the project, to then handle it conveniently (for example, send it attached to an email).



# CREW Manual

Click “Open” and select the required path to save the file.



Crew - Zip wizard

Project name: EW107AA

Description:

Project author: ESA

Target archive file: d:\Users\Imask\Documents\Crew\Crew Editor\EW107AA.zip

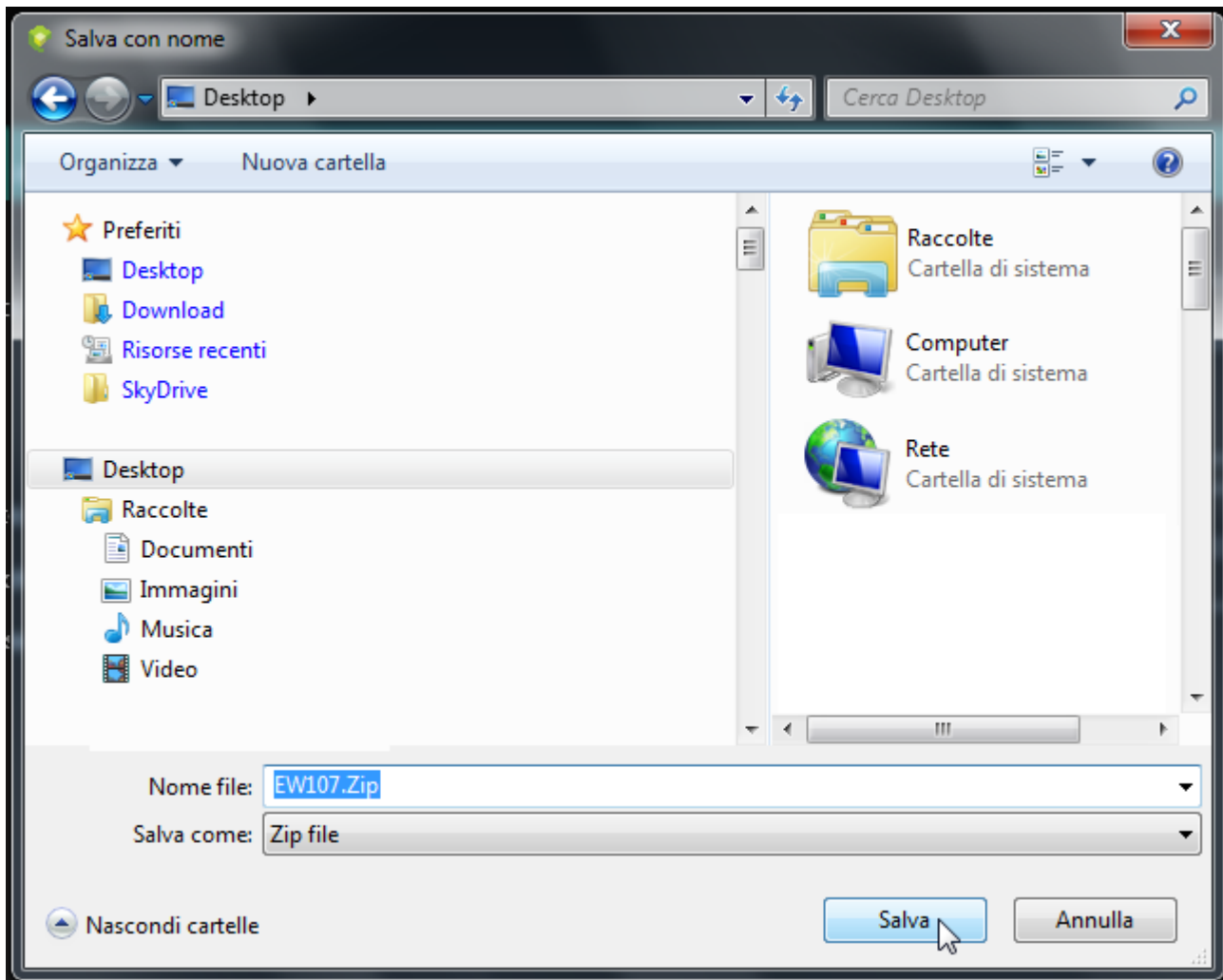
Embed external resources

Open

Cancel Next ▶

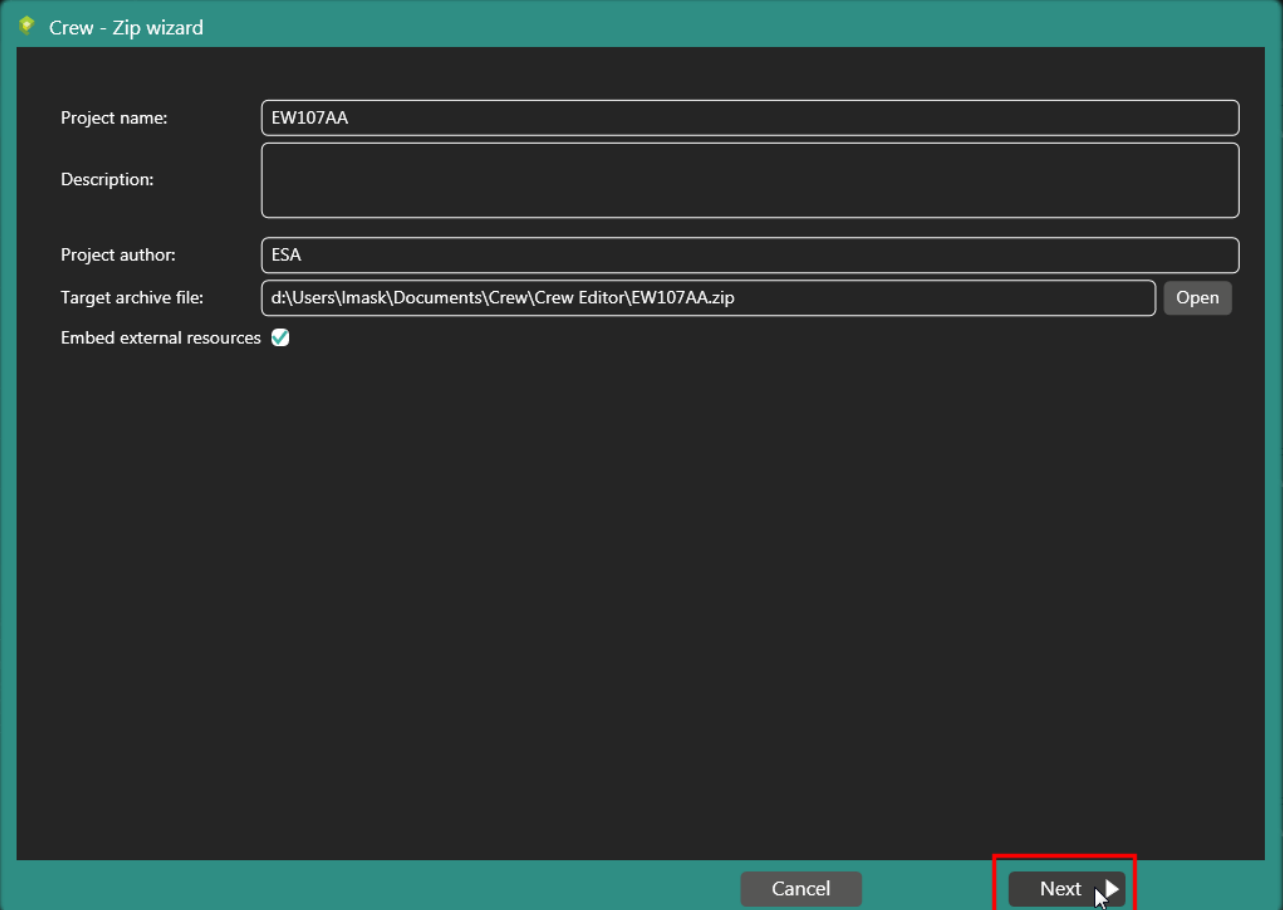
# CREW Manual

Assign a name to the file and save and click “Save”.



# CREW Manual

Click “Next” to compile the .zip file and to save it in the previously indicated path.



Crew - Zip wizard

Project name: EW107AA

Description:

Project author: ESA

Target archive file: d:\Users\Imask\Documents\Crew\Crew Editor\EW107AA.zip Open

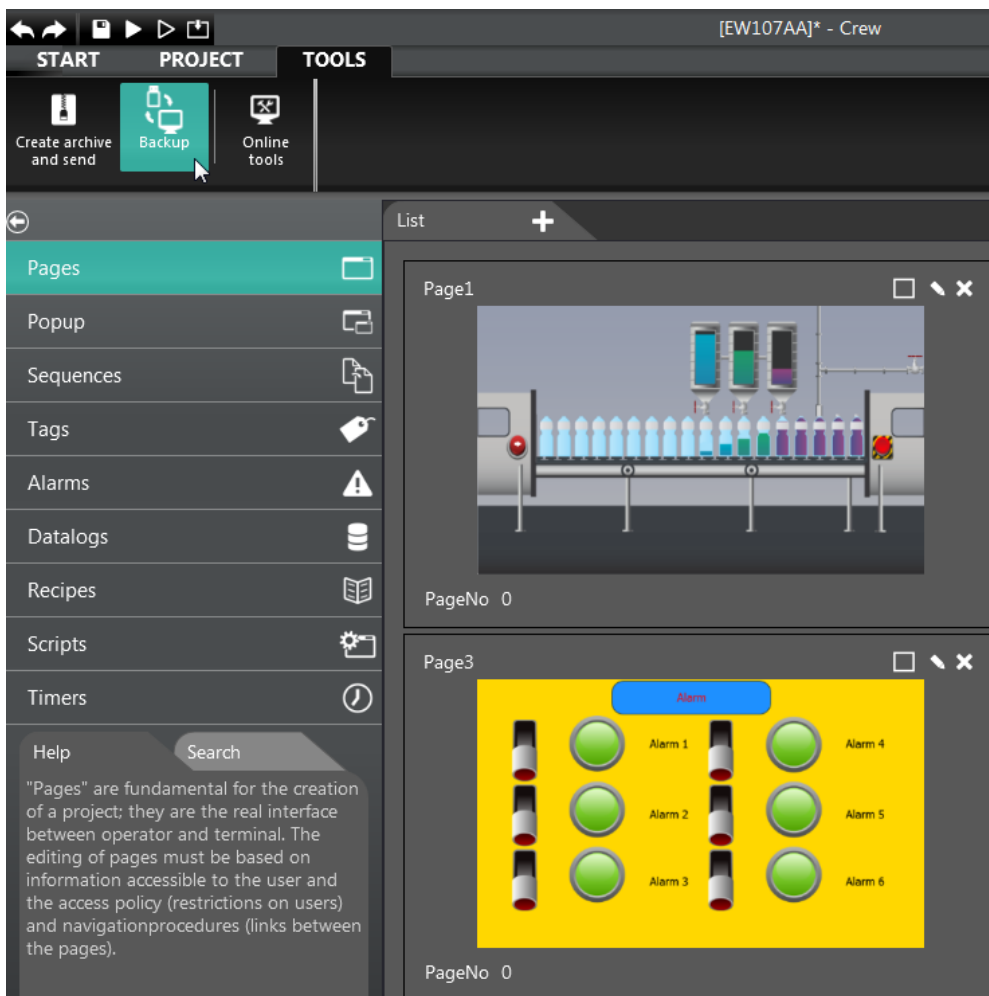
Embed external resources

Cancel Next

# CREW Manual

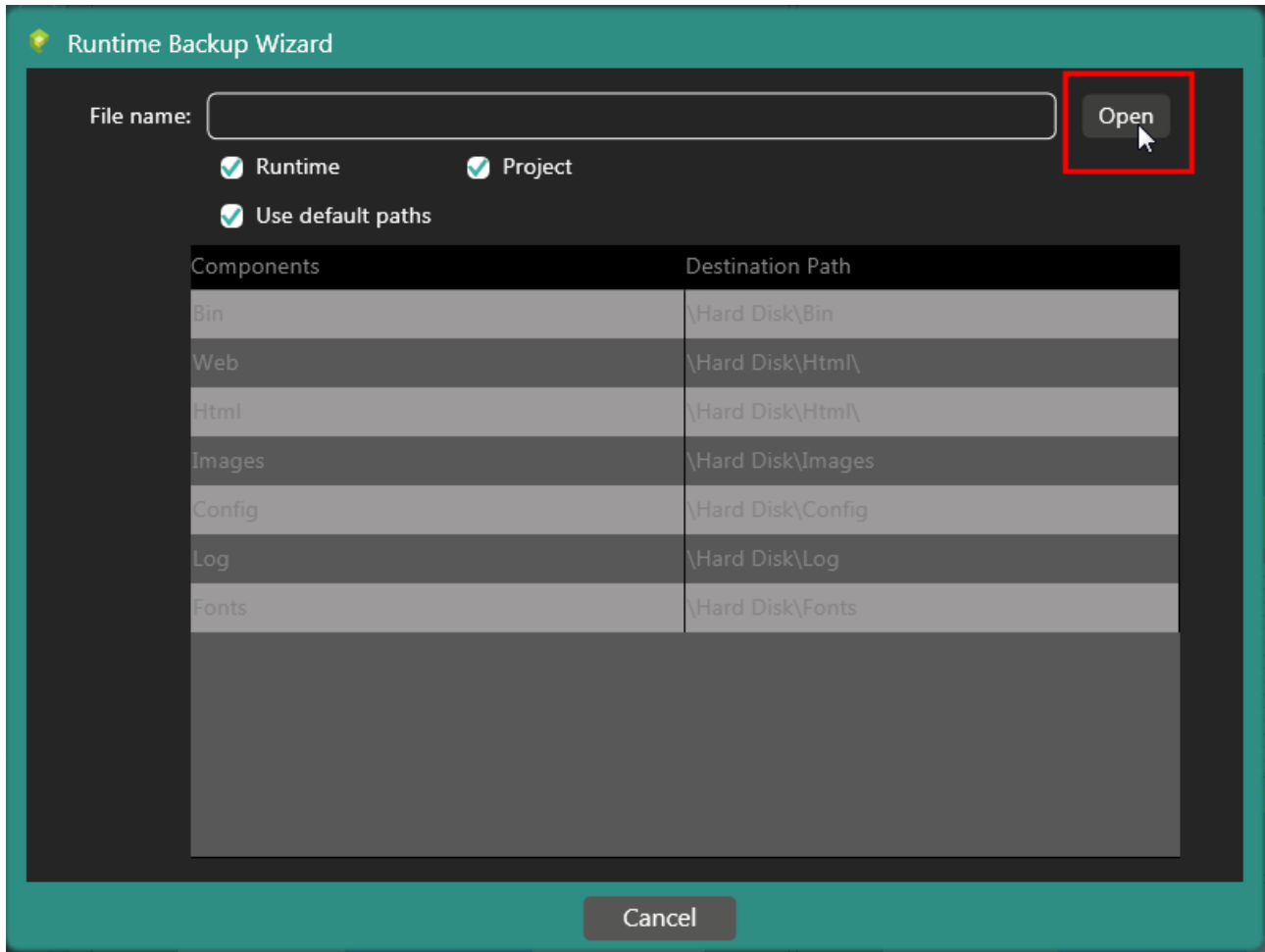
## Backup (offline)

The “Backup” function is used to create a file compiled in the project to be used at a later time (by running a “Restore”) on the terminal.



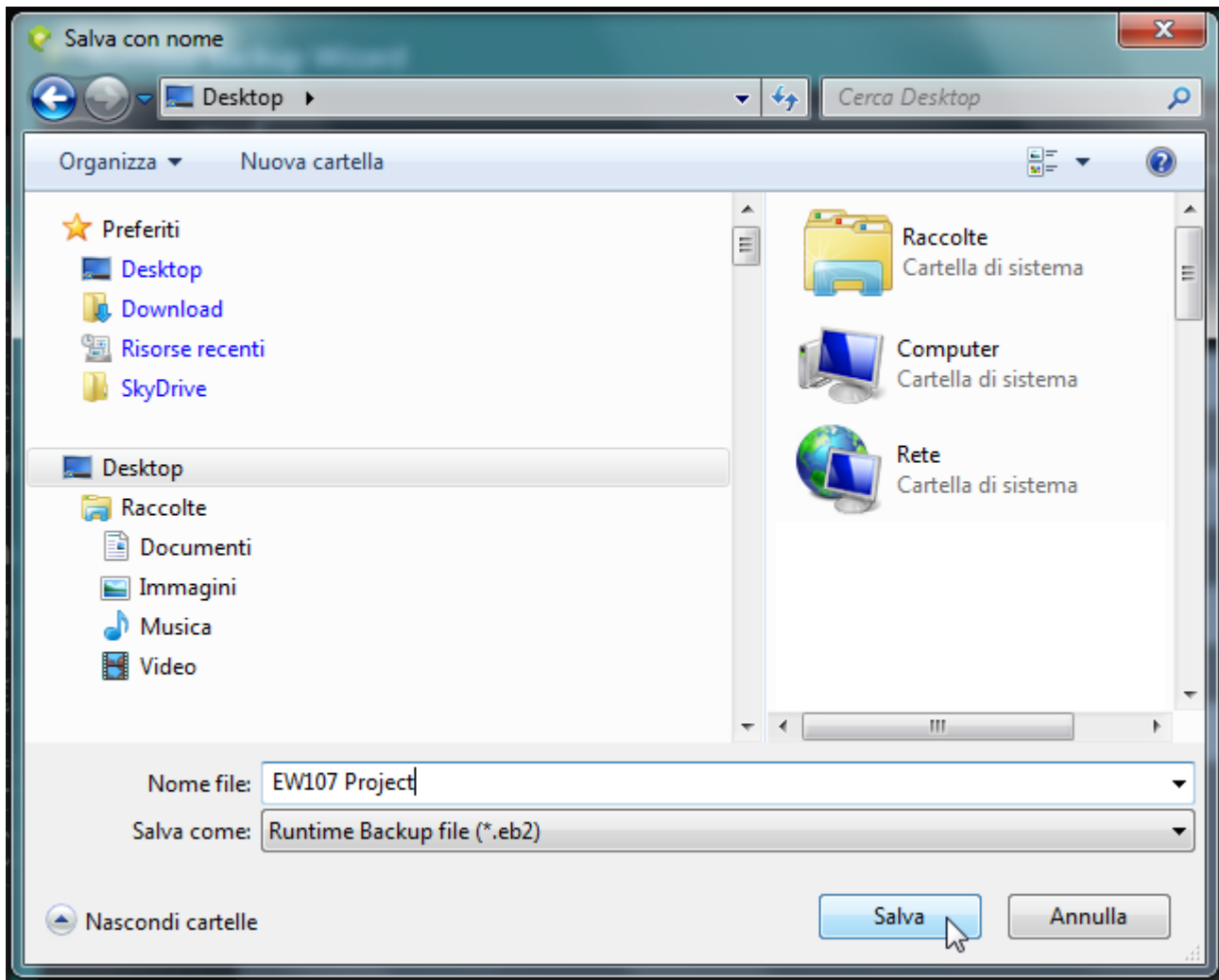
# CREW Manual

Click the “Open” key.



# CREW Manual

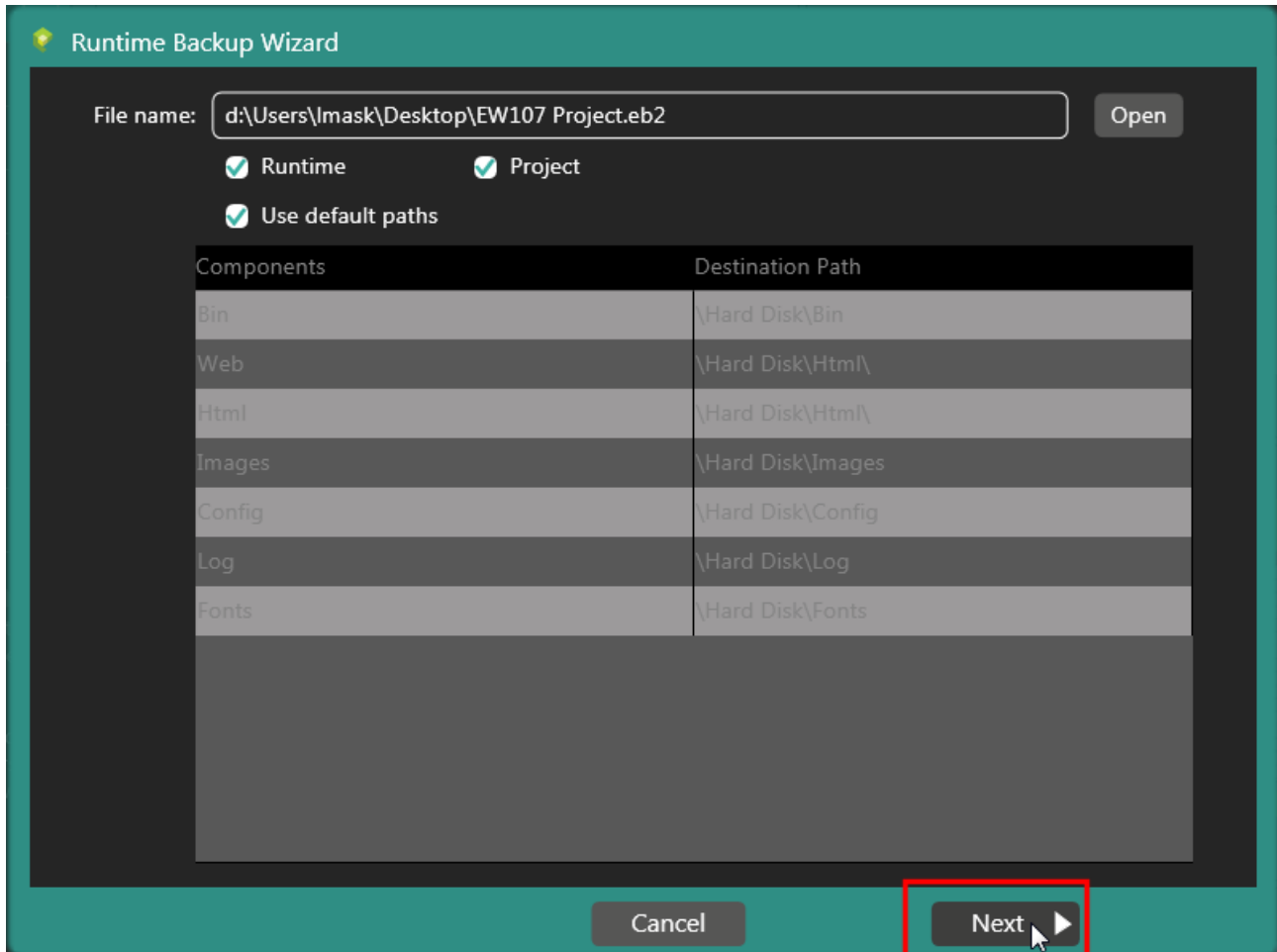
Choose the target path and assign a name to the file to be saved.





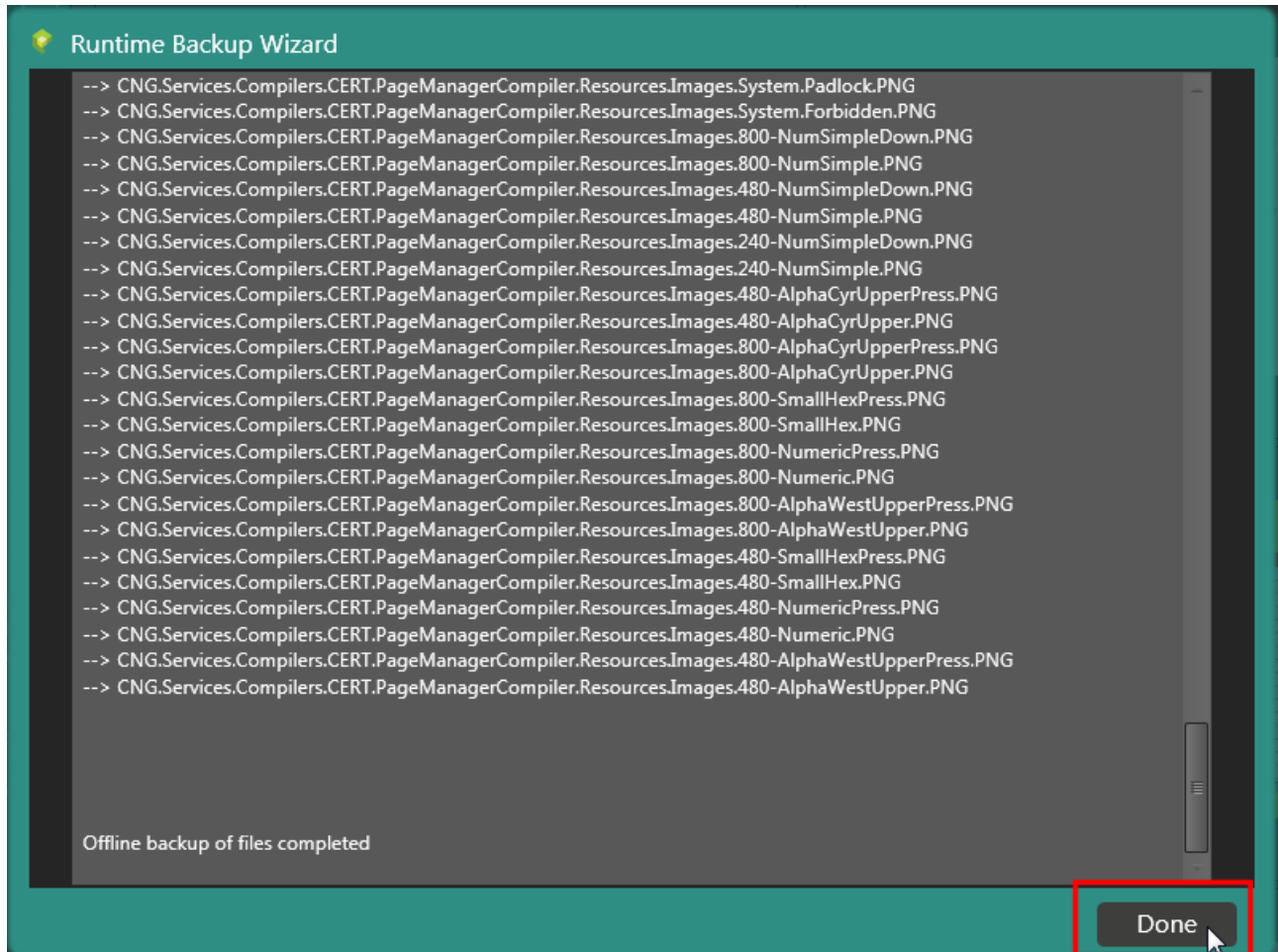
# CREW Manual

Click "Next".



# CREW Manual

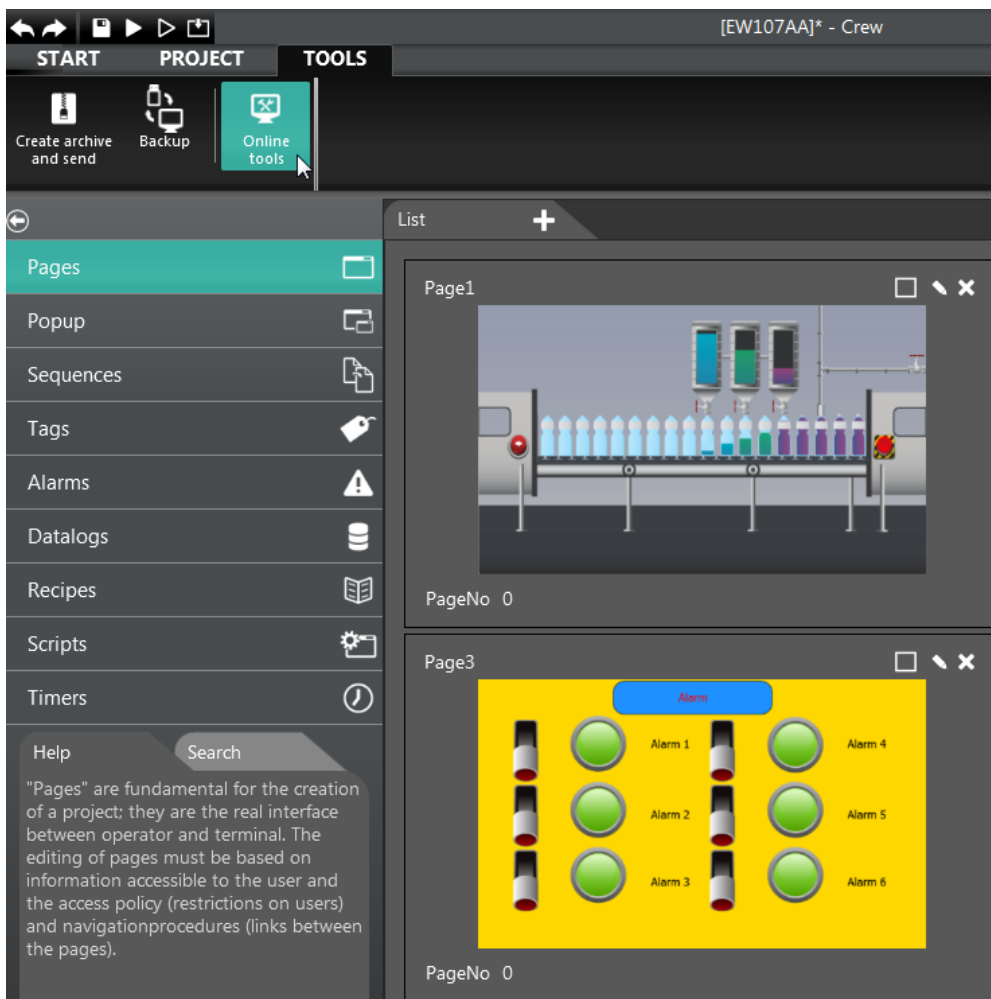
At the end of the backup, click “End”.



# CREW Manual

## Online Tools

The “Online Tools” function is used to access the terminal, explore its contents and possibly make changes to it. Click the relative icon.

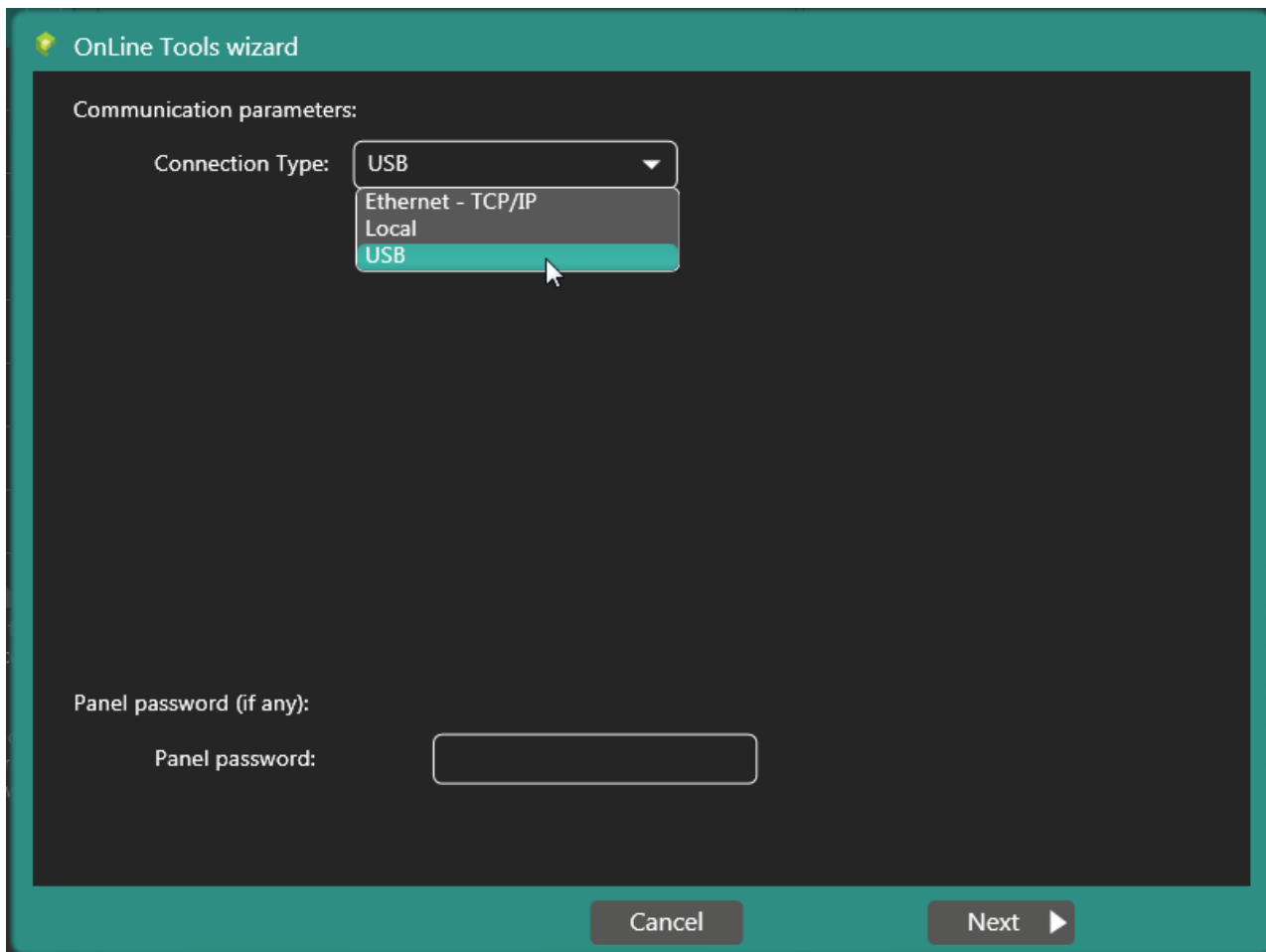


# CREW Manual

The connection modes to the terminal are:

- 1 Ethernet
- 2 Local
- 3 USB

In our example, the USB type of connection has been chosen.



# CREW Manual

Select the transfer method on the panel as well (which in turn needs to be selected in Crew). Go to the service page.

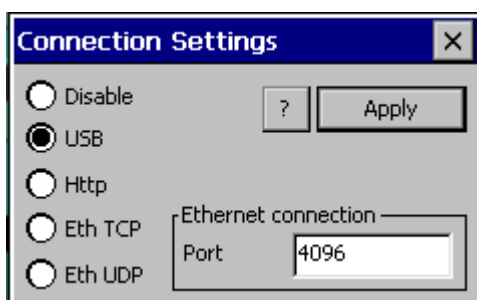


# CREW Manual

Select "Downloader Configuration".

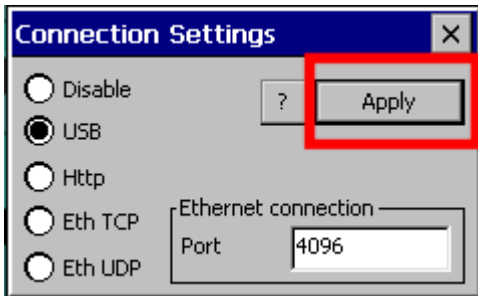


Select "USB".

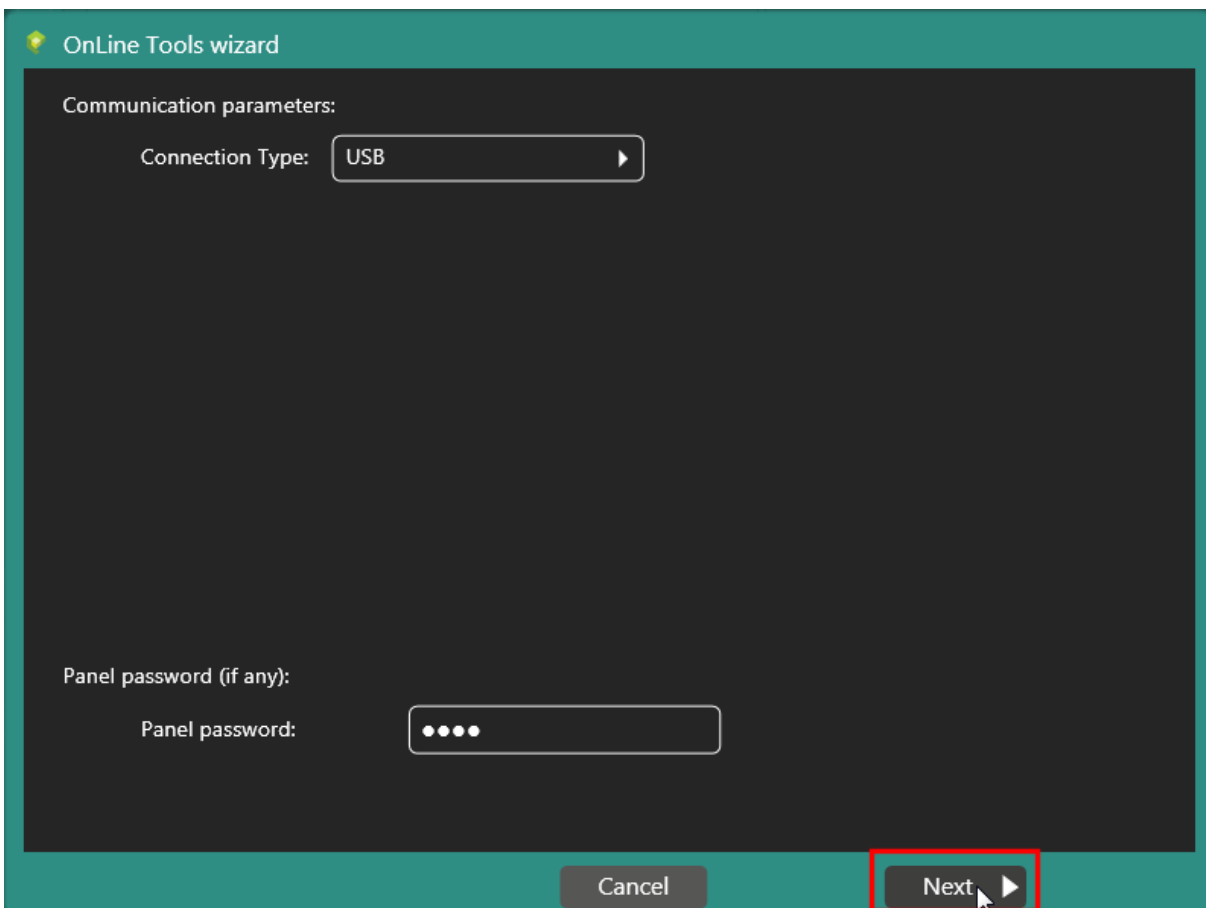


# CREW Manual

Click “Apply” to confirm the setting.

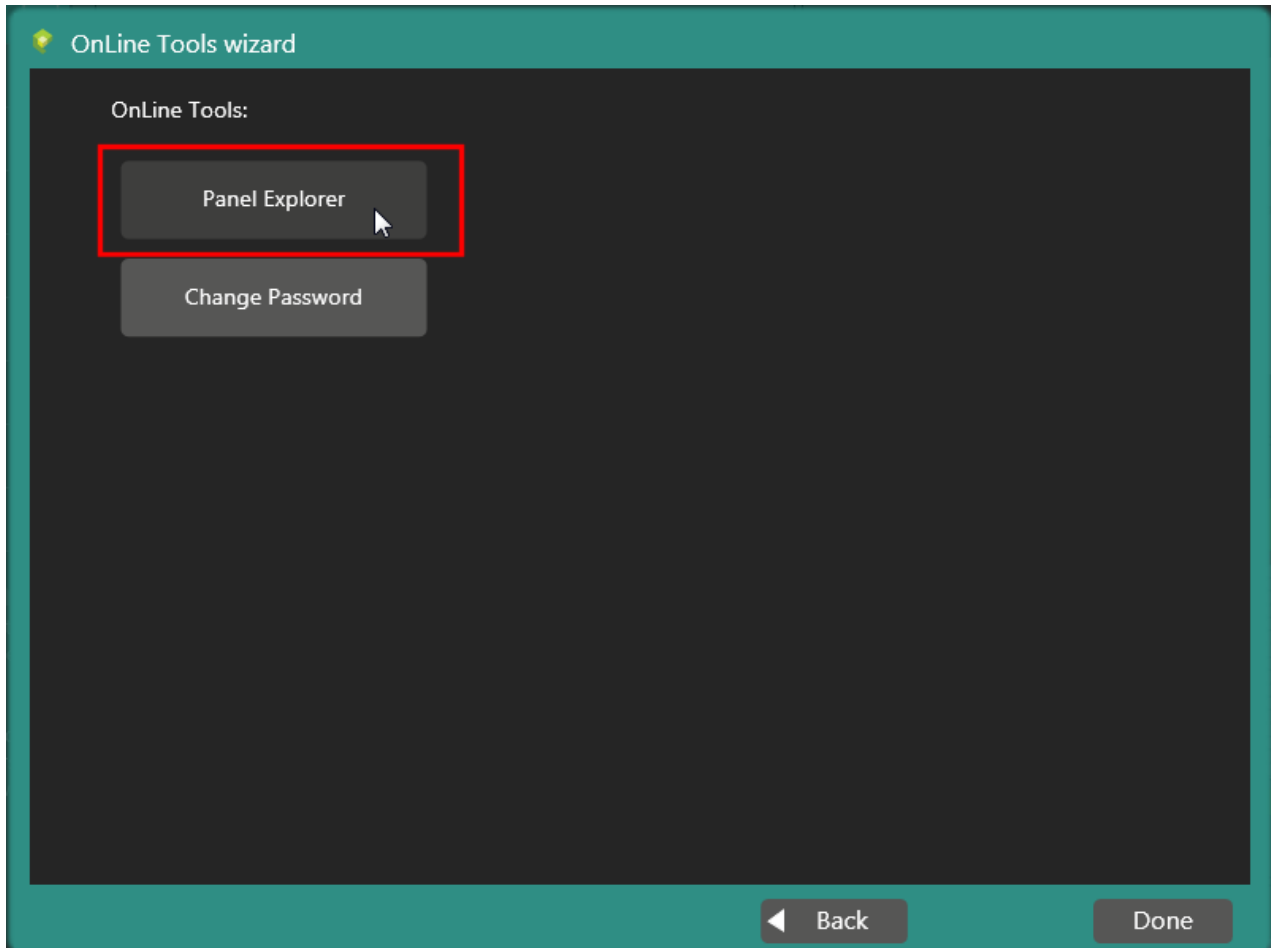


Enter any necessary password and click “Next” to continue.



# CREW Manual

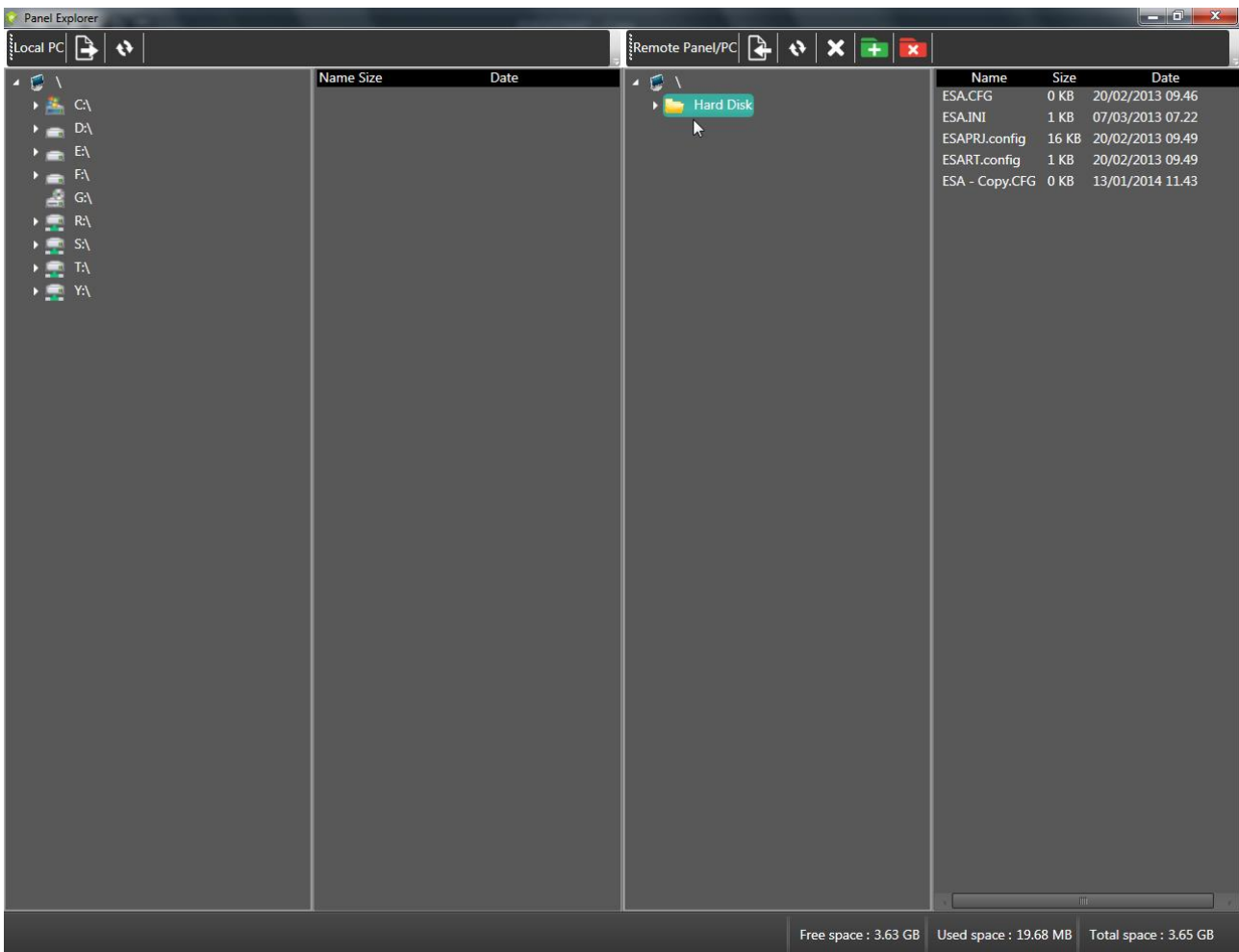
If you wish to explore the panel, click on the relative key.





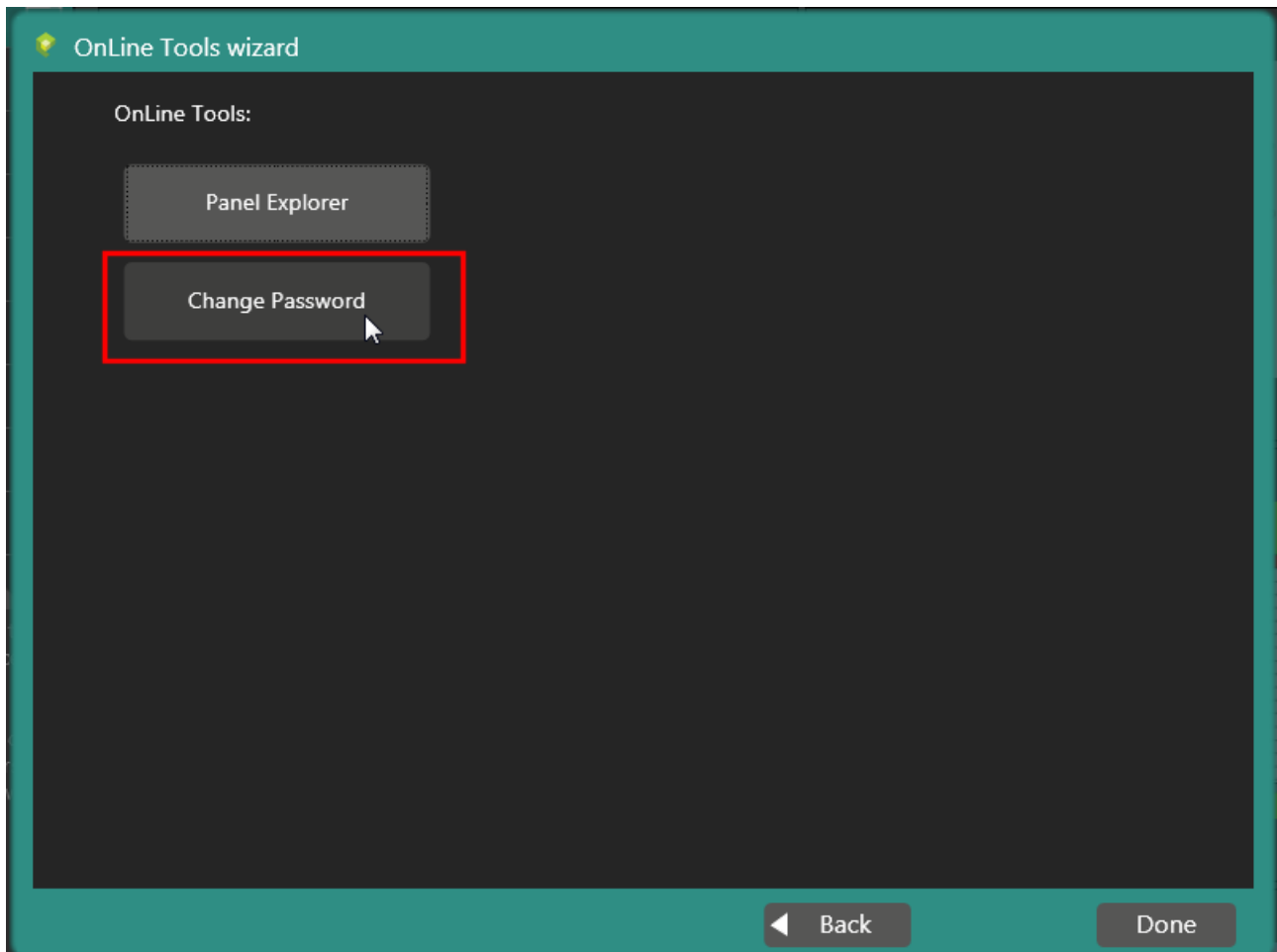
# CREW Manual

From the window that appears it is possible to check both the panel and all of the folders/files contained in it, as well as the PC that Crew is installed on. Clearly, all of the normal copy / delete/ move / add / rename operations can be carried out on all of the files and/or folders on both PC (left side) and HMI panel (right side).



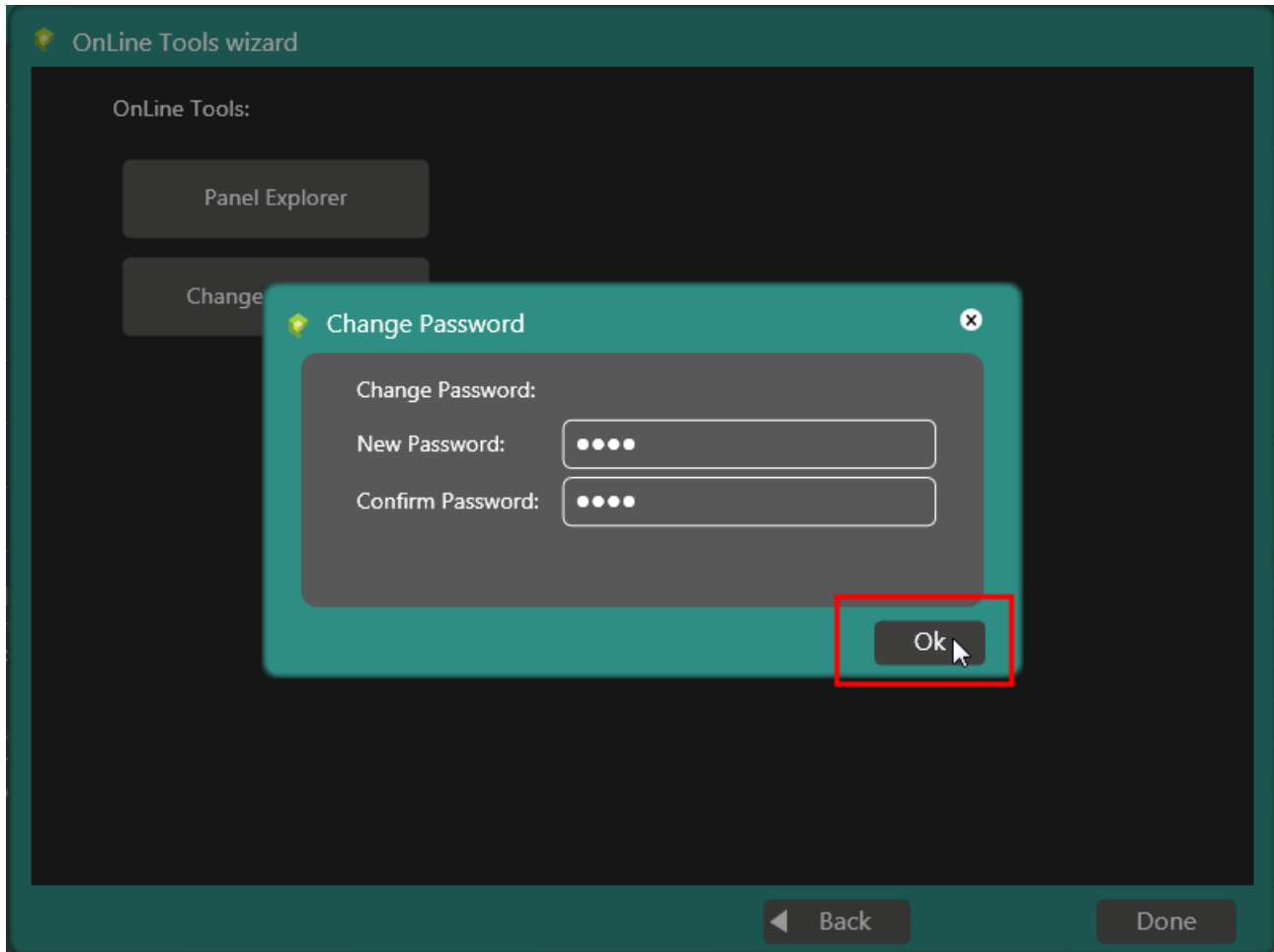
# CREW Manual

From the “Online Tools” function it is also possible to choose the “Change Password” option.



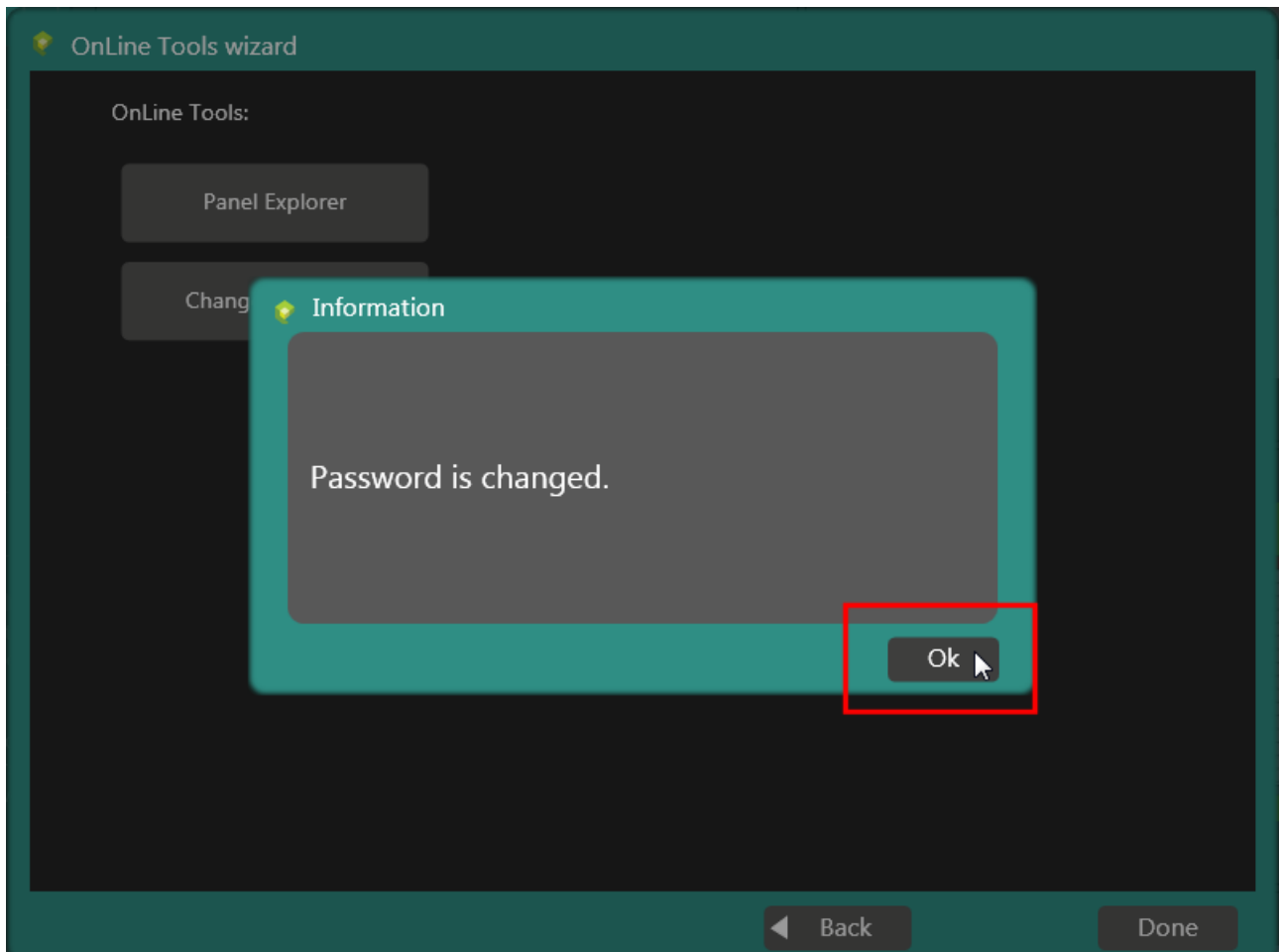
# CREW Manual

If you wish to change the password to access the panel, enter the new password, confirming it and clicking “Ok”.



# CREW Manual

When the window appears to confirm the password change, click “Ok” to apply the change.

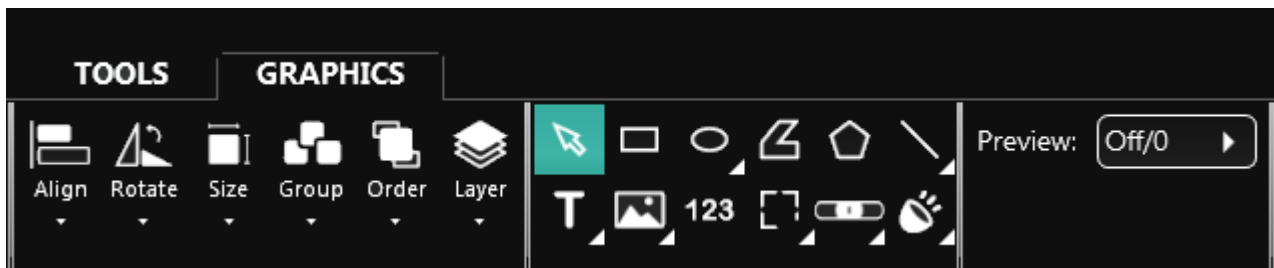


# CREW Manual

---

## GRAPHICS Menu

Crew contains a series of predefined graphic elements that can be entered on a page. These elements can have simple graphic functions, navigation functions, data viewing and editing functions. The icons for these objects are found in the “Graphics” menu bar.



All of the graphic elements have been grouped together into three different sections of the menu. The first contains the following:

- Align
- Rotate
- Size
- Group
- Order
- Layer

The second section of the menu contains the following elements:

- Selection tool
- Rectangle
- Ellipse
- Circular sector
- Arch
- Polygon

# CREW Manual

- Regular polygon
- Line
- Polygonal chain
- Pipeline
- Text - Label -
- Text list
- Image
- Image list
- Edit Box - Numerical field -
- Sensitive area
- Sensitive area - Ellipse
- Switch
- Button
- Selector
- Bar
- Indicator
- Active Alarms
- Alarm Log
- Data Log
- Recipe
- Recipe List
- Trend
- XY Trend
- Trend Touch
- Users

# CREW Manual

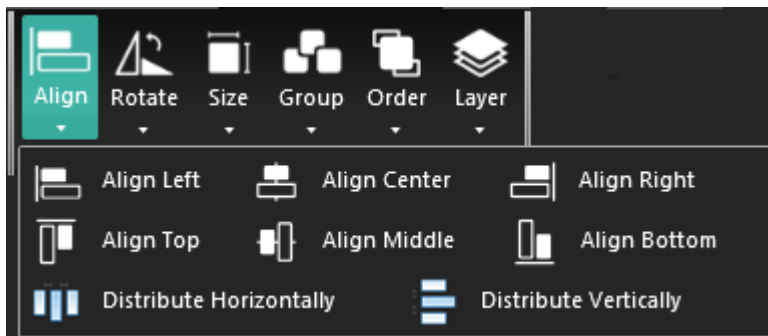
The third section of the menu contains the "Preview" function (see [Preview Submenu](#)).



The next sections have a list of all of the predefined graphic elements that can be entered on a page. For each element, the relative modifiable properties and events that can be associated to them, are listed schematically.

# CREW Manual

## Alignment Submenu



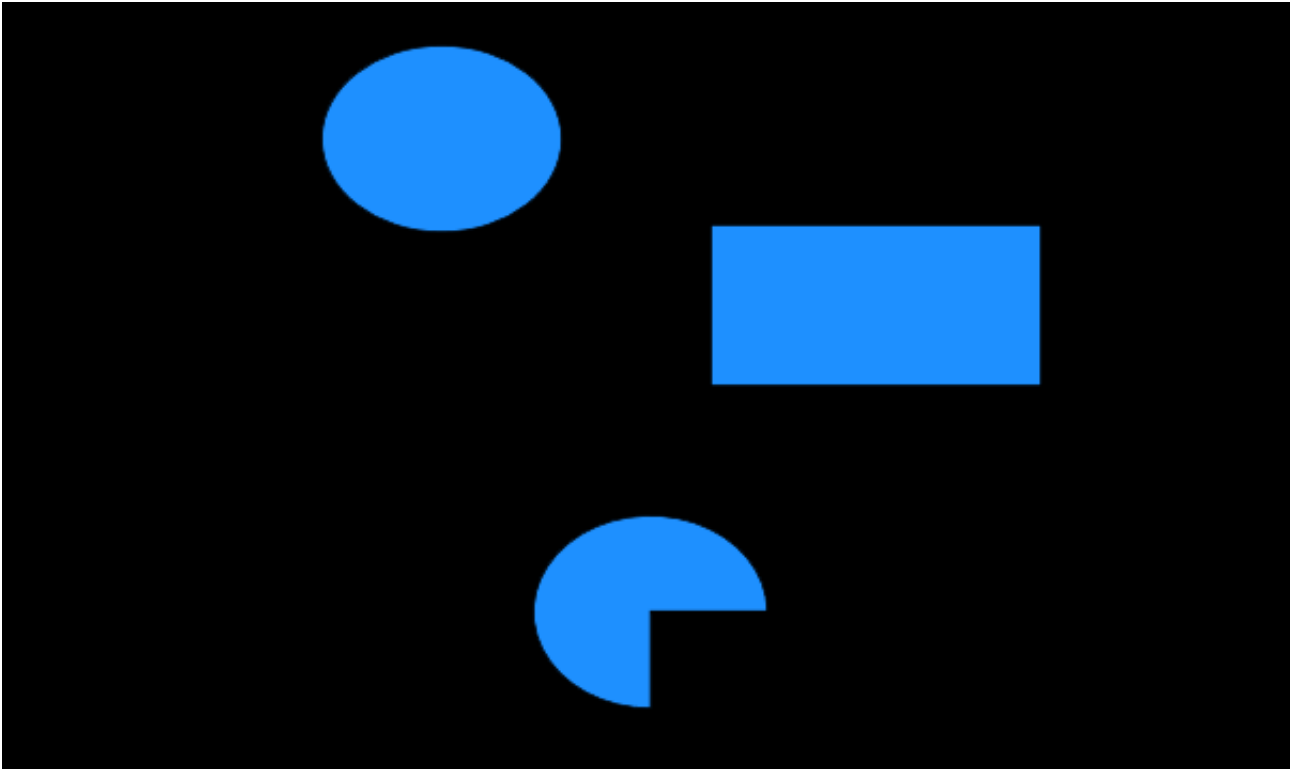
When there are two or more elements on a page, Crew provides the tools to align them. To access these tools simply enter the “Alignment” submenu from the “Graphics Menu” and select the respective icons.

Below is an example describing the behaviour of the various alignment functions, through three elements (an ellipse, a rectangle and a circular sector) positioned on the page.



# CREW Manual

If there are several objects needing alignment on the page, they need to be selected in order to be aligned.



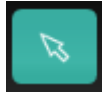
The following icons refer to alignment:

- Align to the left: align with the left of the selected object.
- Align in the centre: align with the centre of the selected object.
- Align to the right: align with the right of the selected object.
- Align with the top: align with the top of the selected object.
- Align in the middle: align with the middle of the selected object.
- Align with the bottom: align with the bottom of the selected object.
- Distribute horizontally: horizontally distribute the selected object.
- Distribute vertically: vertically distribute the selected object.

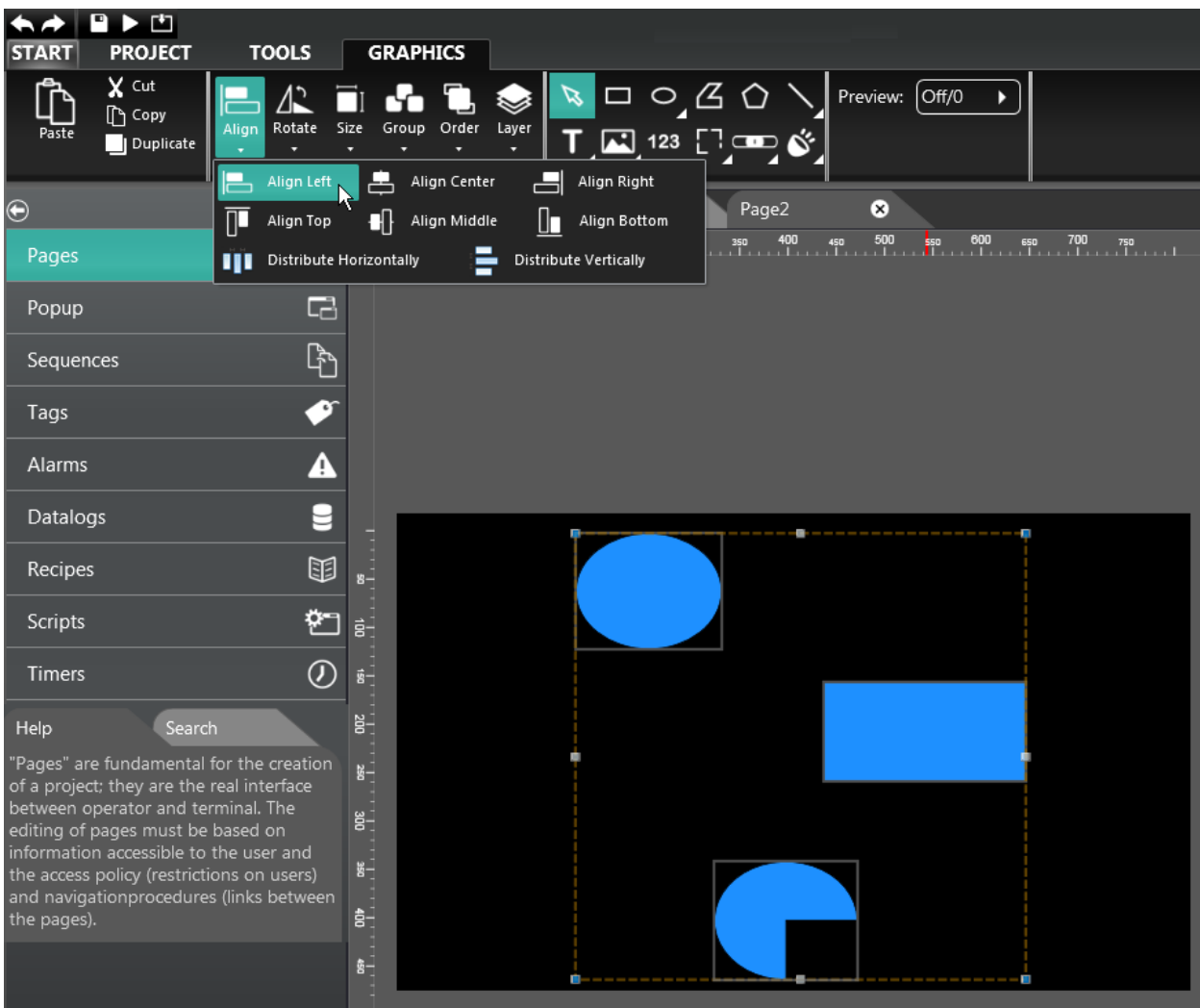
# CREW Manual

## Align to the left

Use the “Selection” tool.

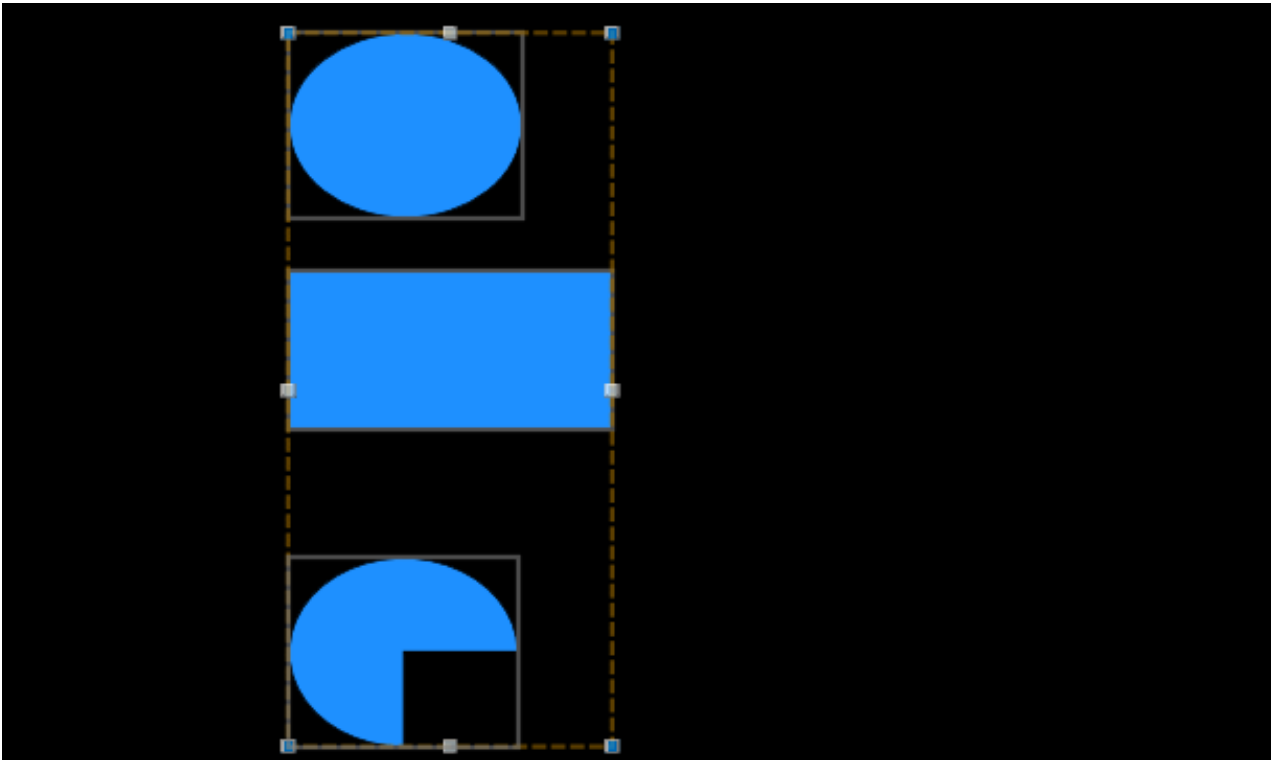


With the mouse, select the objects contained on the page one at a time while holding down the “Ctrl” key. Then choose the “Align to the left” option.



# CREW Manual

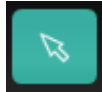
The left edges of all selected objects will now align with each other at the latitude of the left edge of the object selected first (in the example it is the ellipse).



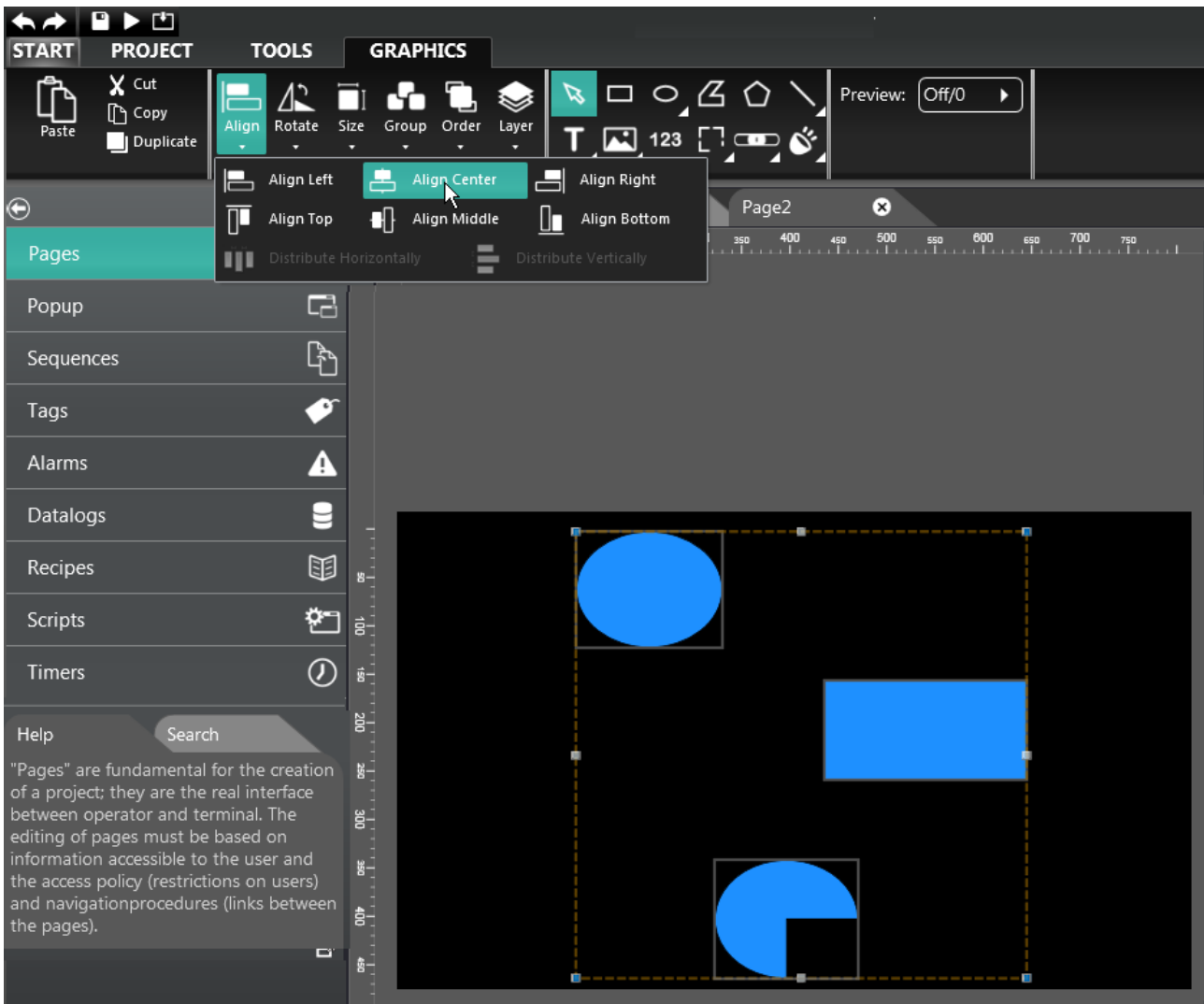
# CREW Manual

## Align in the centre

Use the “Selection” tool.

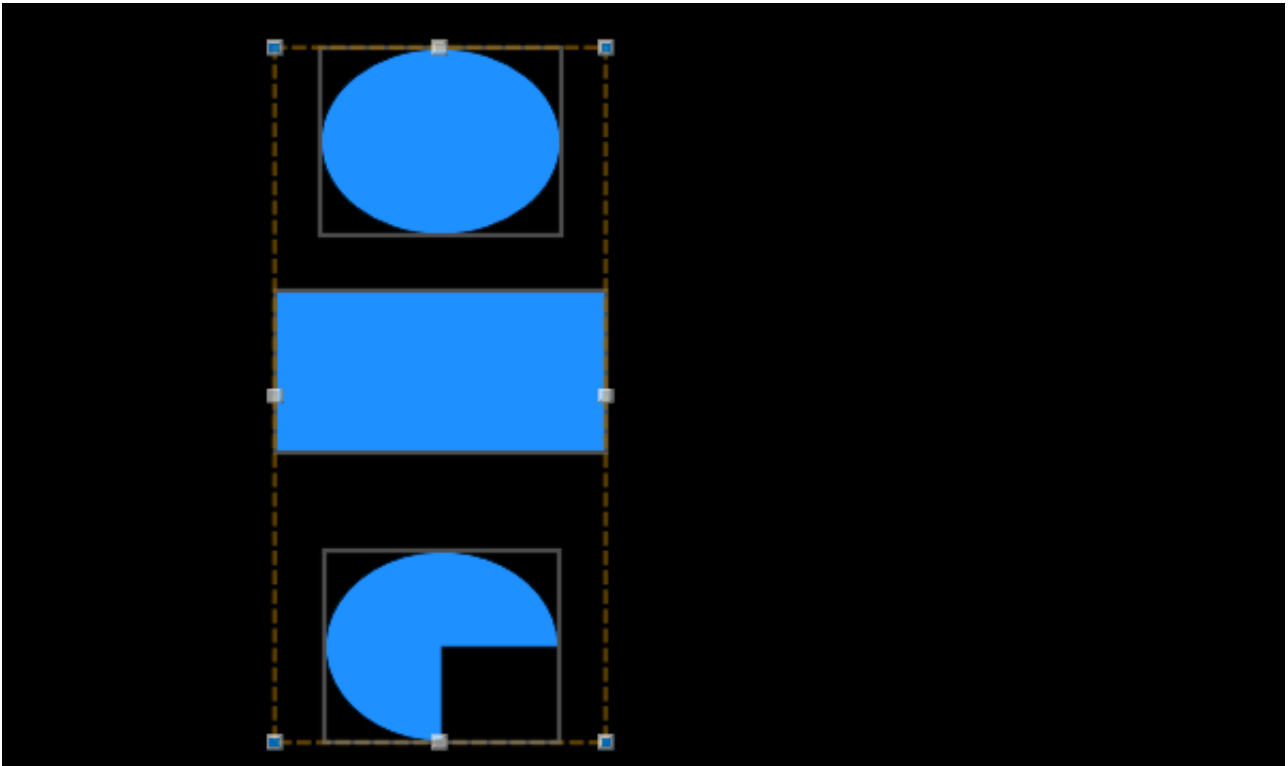


With the mouse, select the objects contained on the page one at a time while holding down the “Ctrl” key. Then choose the “Align in the centre” option.



# CREW Manual

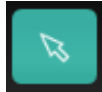
The mid points (horizontal) of all selected objects will align with each other at the latitude of the mid point (horizontal) of the object selected first (in the example it is the ellipse).



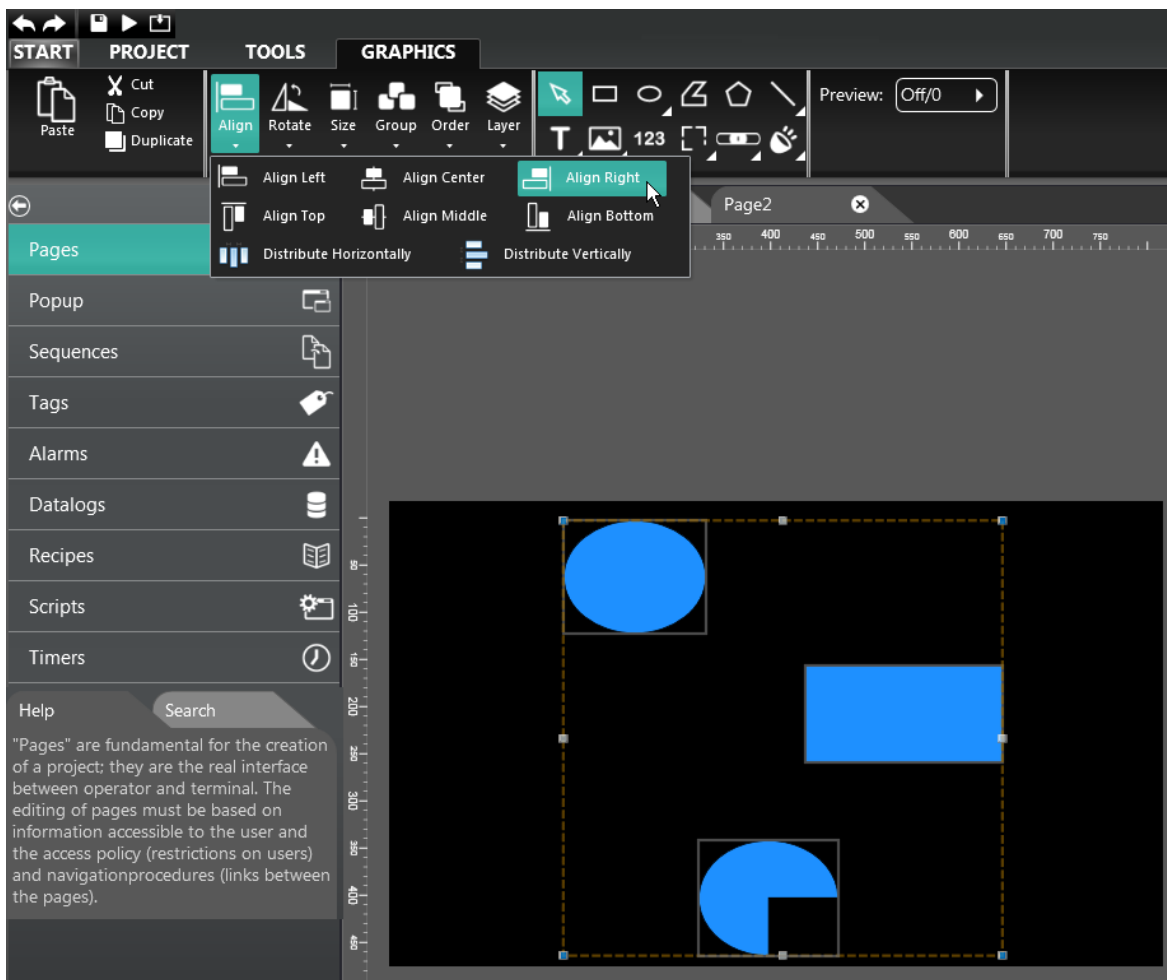
# CREW Manual

## Align to the right

Use the “Selection” tool.

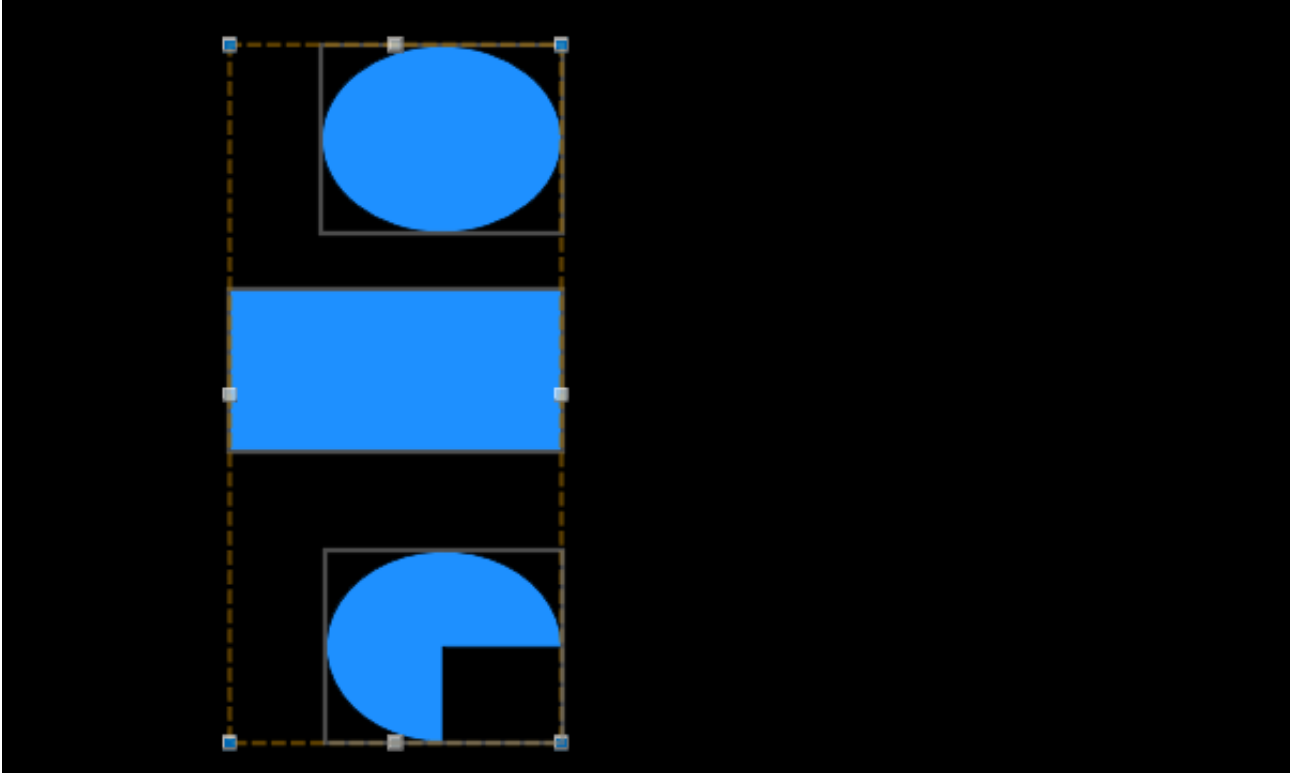


With the mouse, select the objects contained on the page one at a time while holding down the “Ctrl” key. Then choose the “Align to the right” option.



# CREW Manual

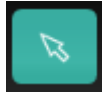
The right edges of all selected objects will align with each other at the latitude of the right edge of the object selected first (in the example it is the ellipse).



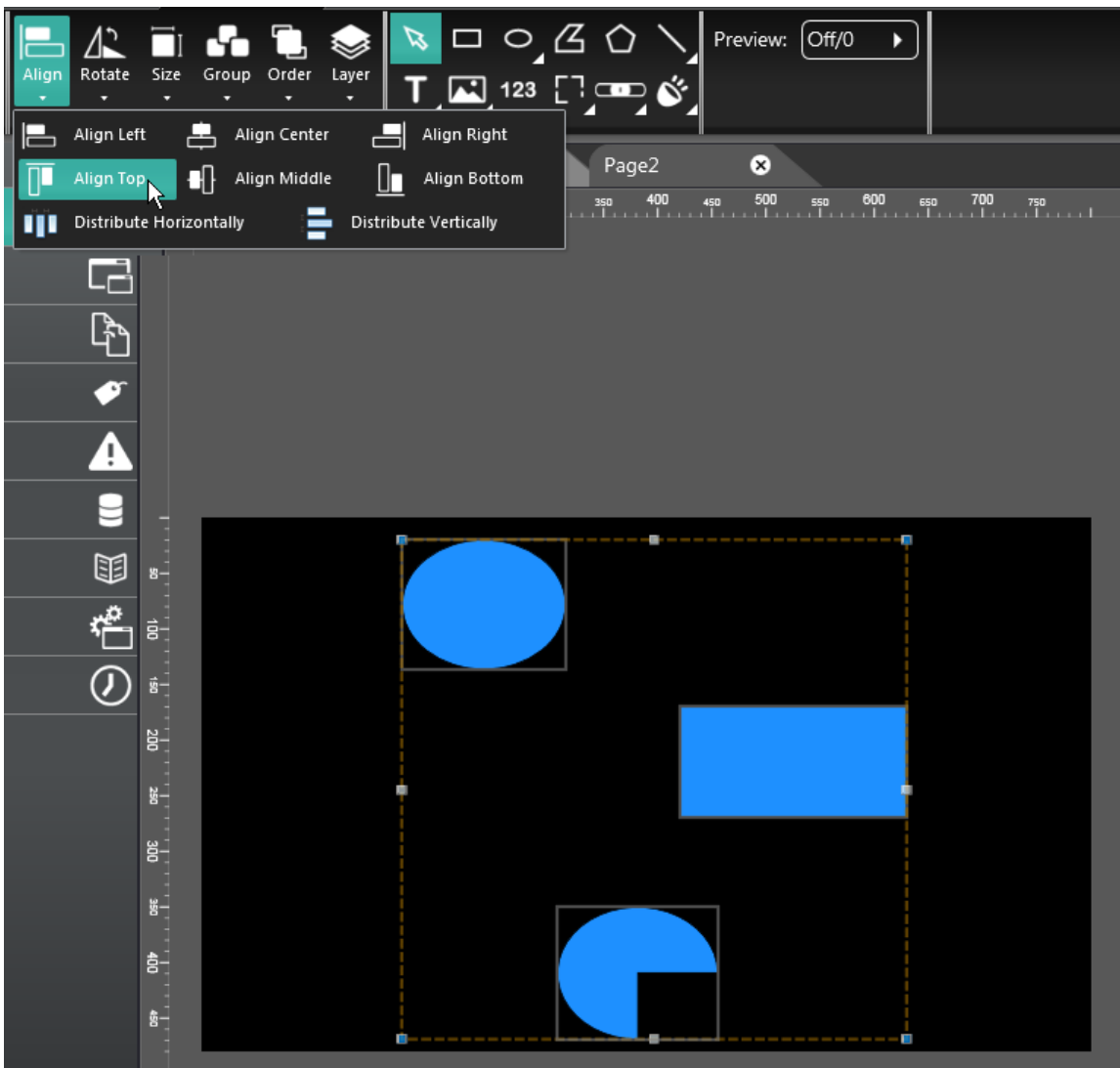
# CREW Manual

## Align with the top

Use the “Selection” tool.



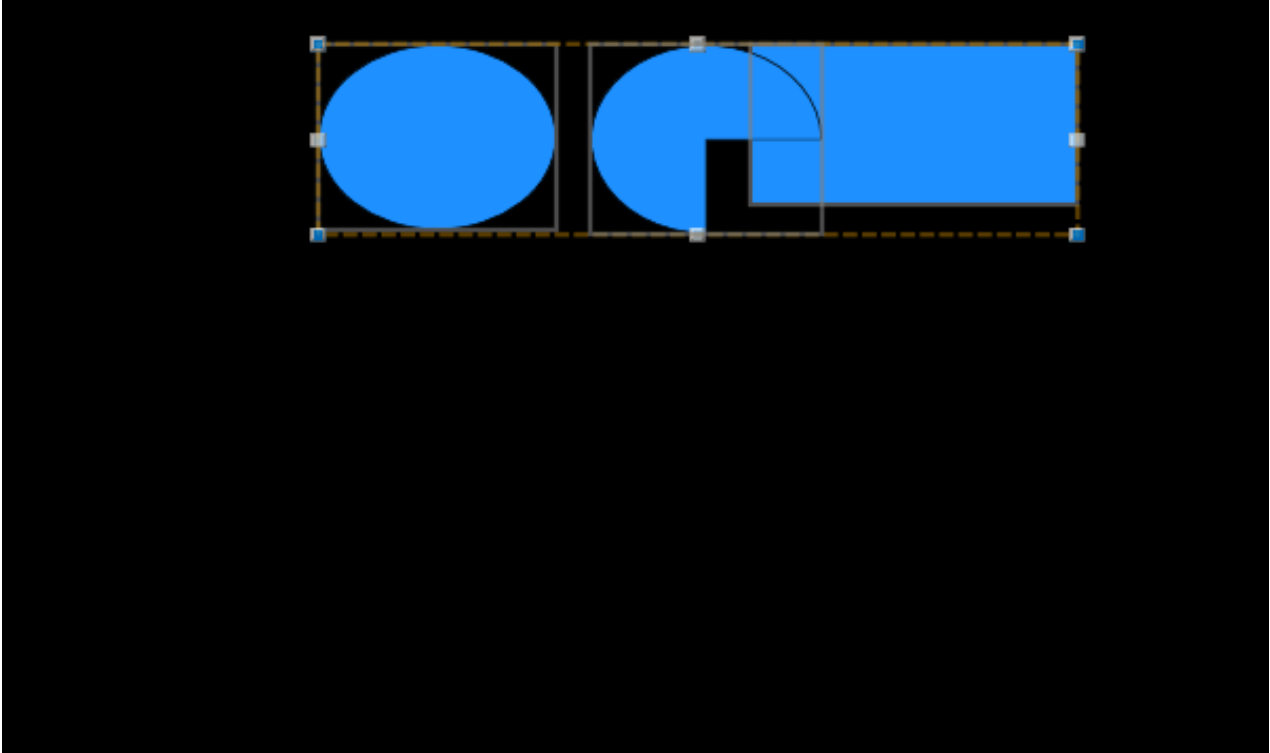
With the mouse, select the objects contained on the page one at a time while holding down the “Ctrl” key. Then choose the “Align with the top” option.





# CREW Manual

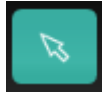
The top edges of all selected objects will align with each other at the height of the top edge of the object selected first (in the example it is the ellipse).



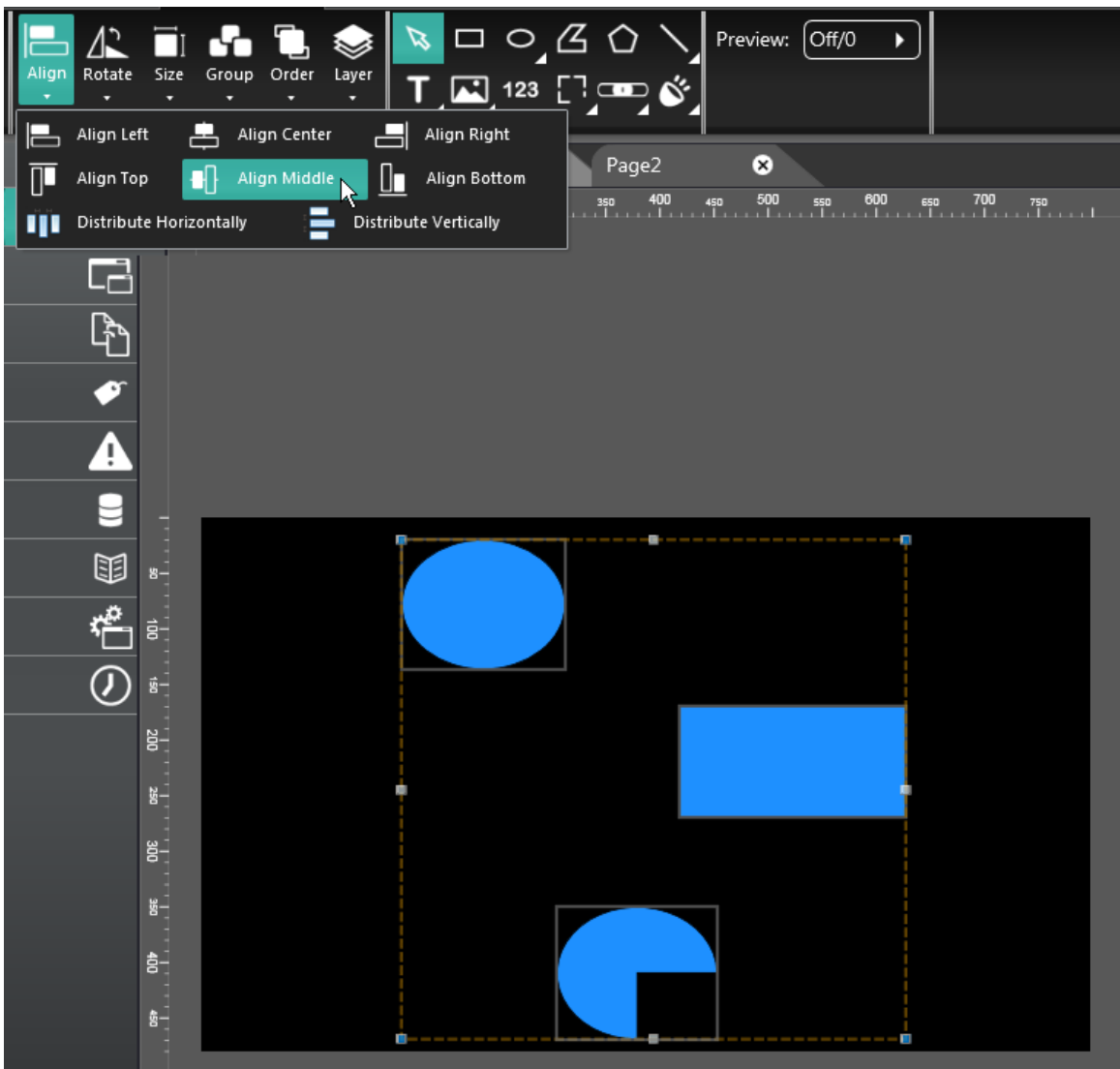
# CREW Manual

## Align in the middle

Use the “Selection” tool.

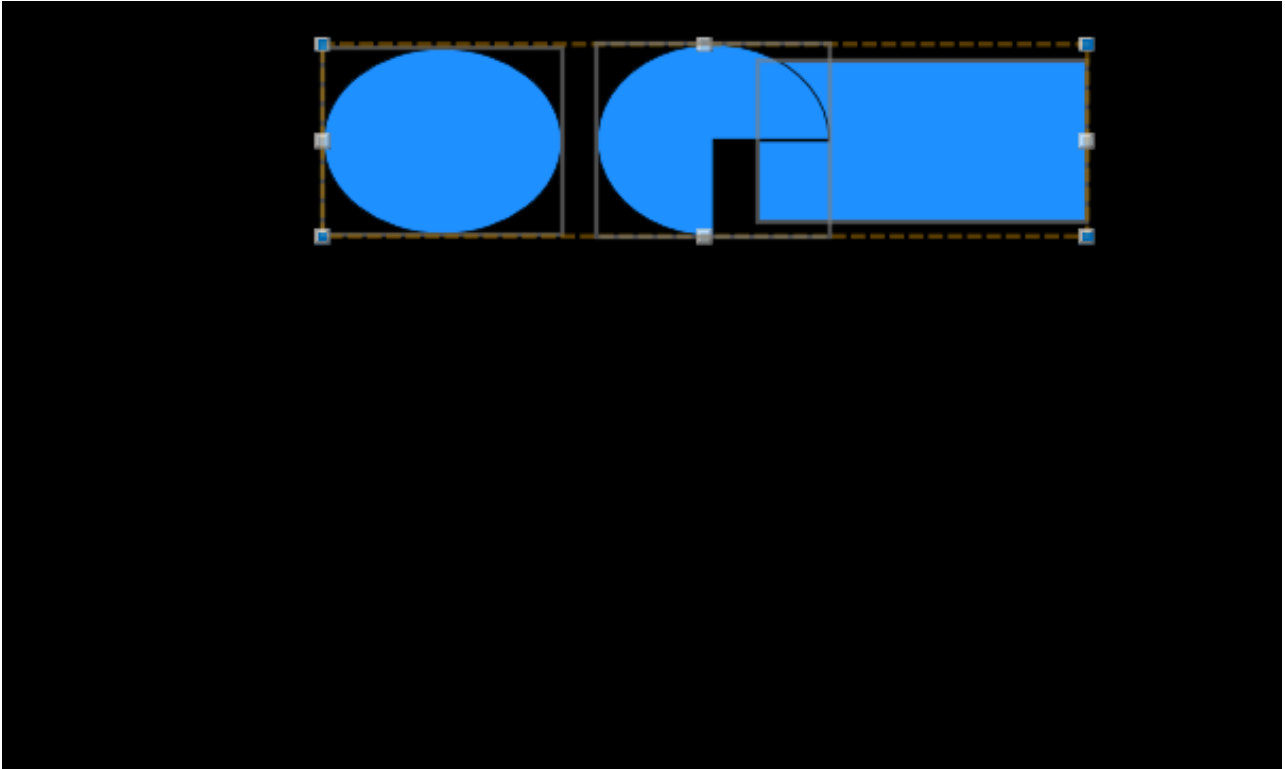


With the mouse, select the objects contained on the page one at a time while holding down the “Ctrl” key. Then choose the “Align in the middle” option.



# CREW Manual

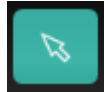
The mid points (vertical) of all selected objects will align with each other at the height of the mid point (vertical) of the object selected first (in the example it is the ellipse).



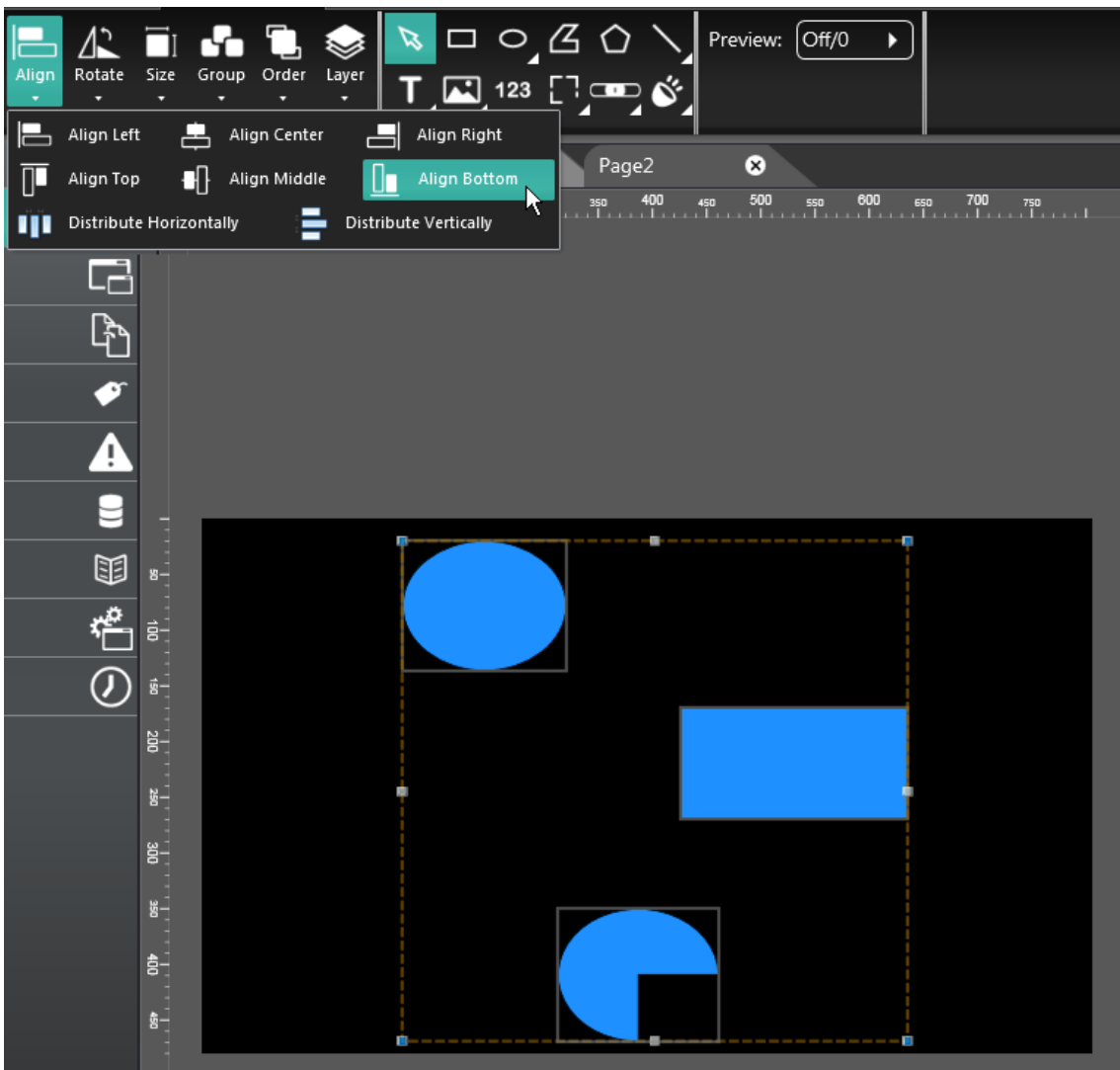
# CREW Manual

## Align with the bottom

Use the “Selection” tool.

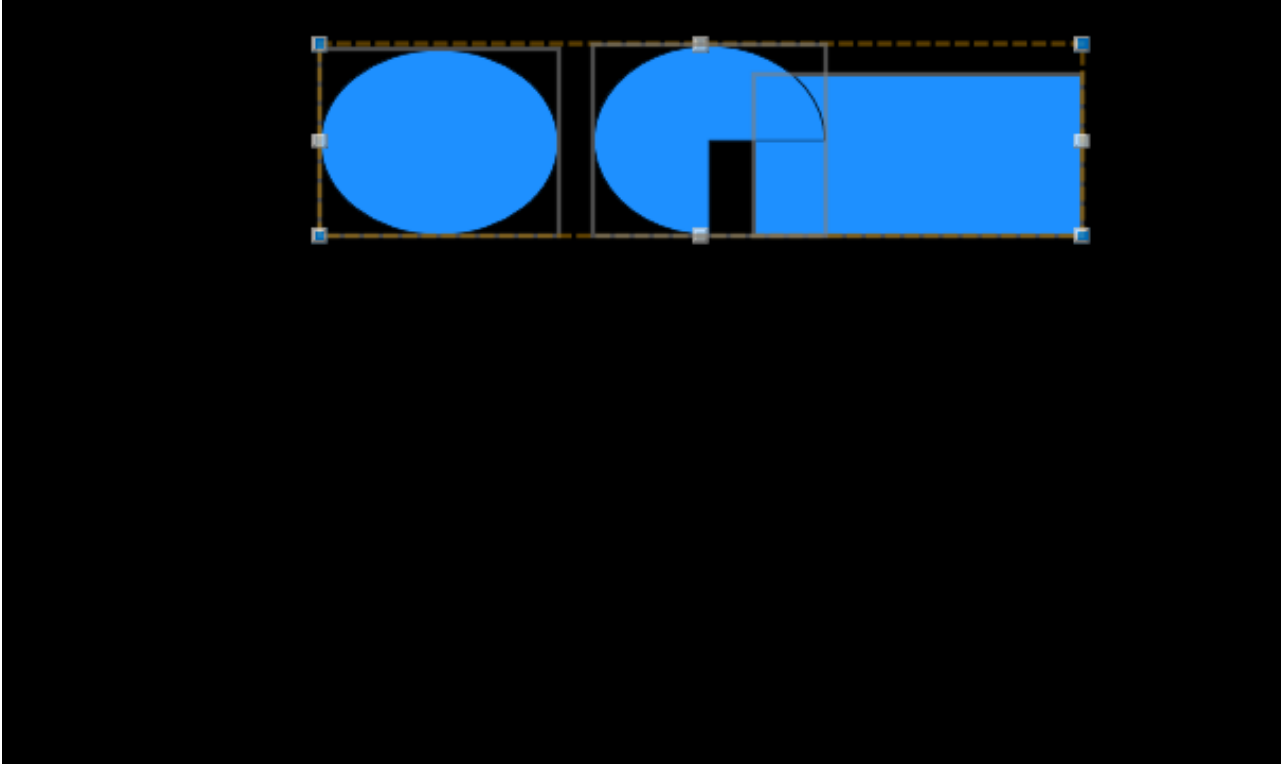


With the mouse, select the objects contained on the page one at a time while holding down the “Ctrl” key. Then choose the “Align with the bottom” option.



# CREW Manual

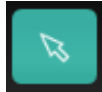
The bottom edges of all selected objects will align with each other at the height of the bottom edge of the object selected first (in the example it is the ellipse).



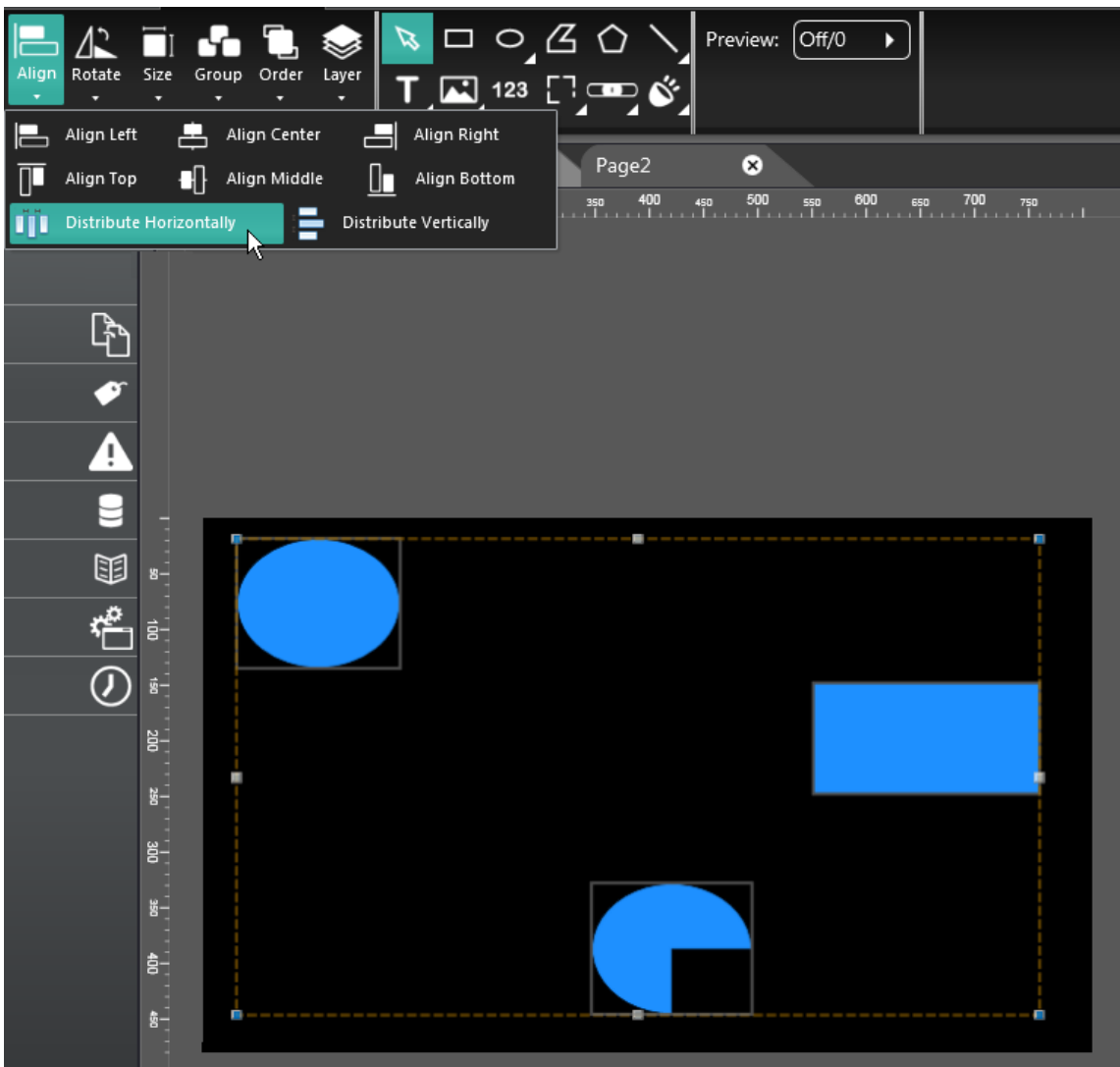
# CREW Manual

## Distribute horizontally

Use the “Selection” tool.

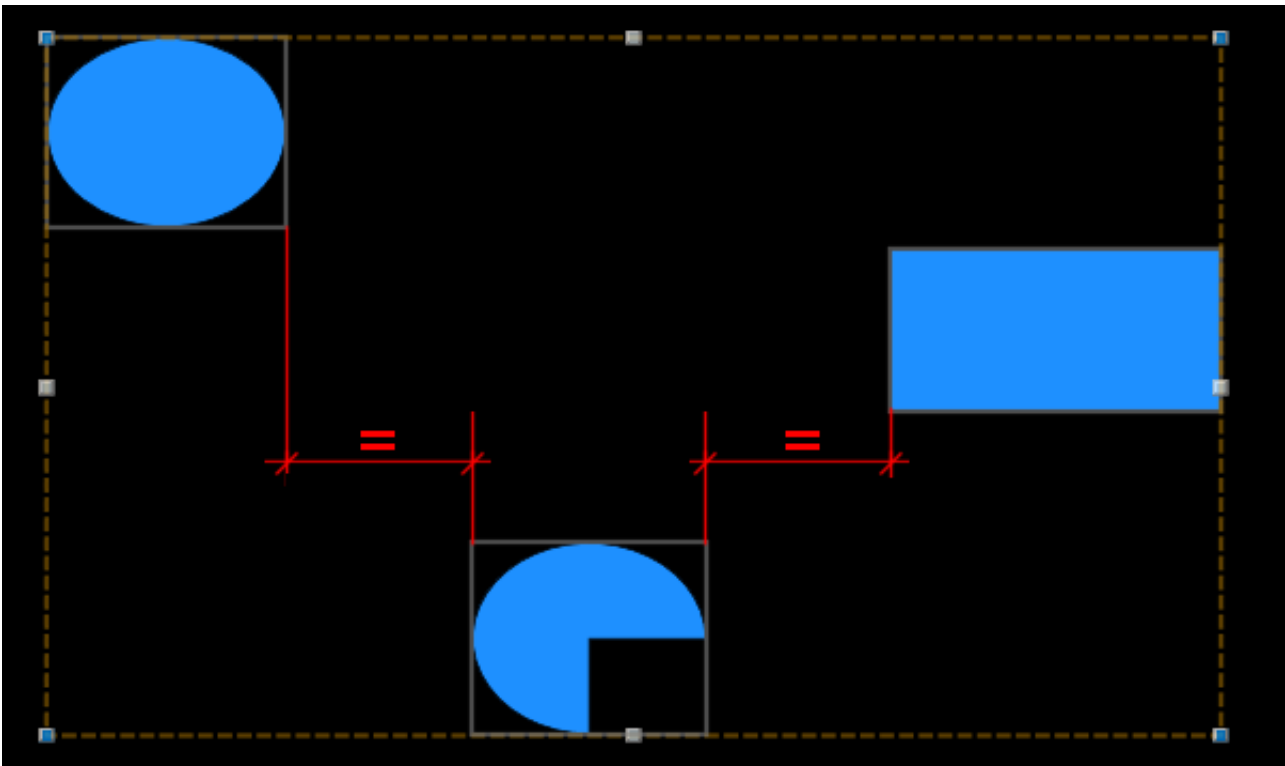


With the mouse, select the objects contained on the page one at a time while holding down the “Ctrl” key. Then choose the “Distribute horizontally” option.



# CREW Manual

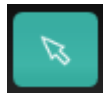
The selected objects are distributed horizontally so that the distance between the right side of the first object and the left side of the second object is always equal to the distance between the right side of the second object and the left side of the third object (from left to right), and so forth depending on the number of objects there are on the page.



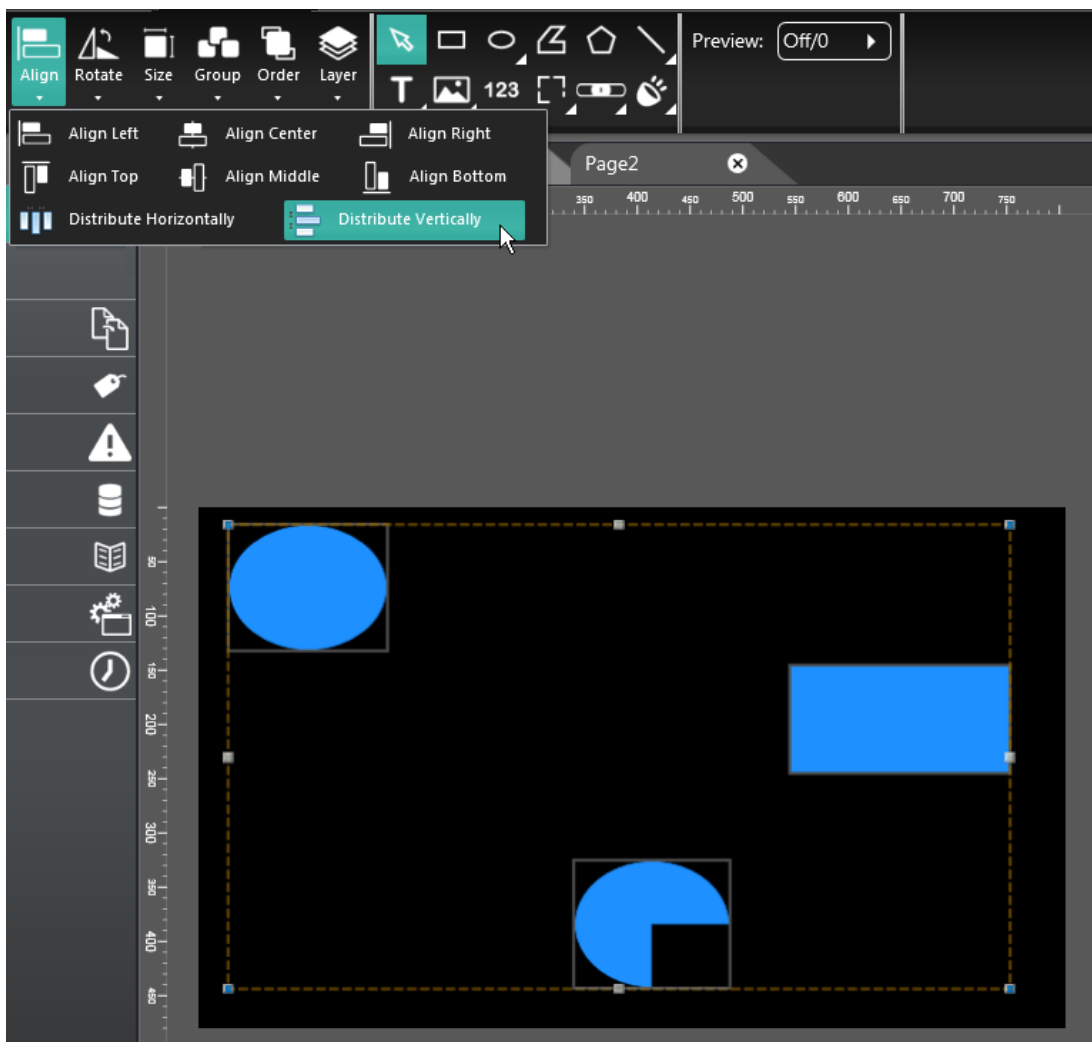
# CREW Manual

## Distribute vertically

Use the “Selection” tool.



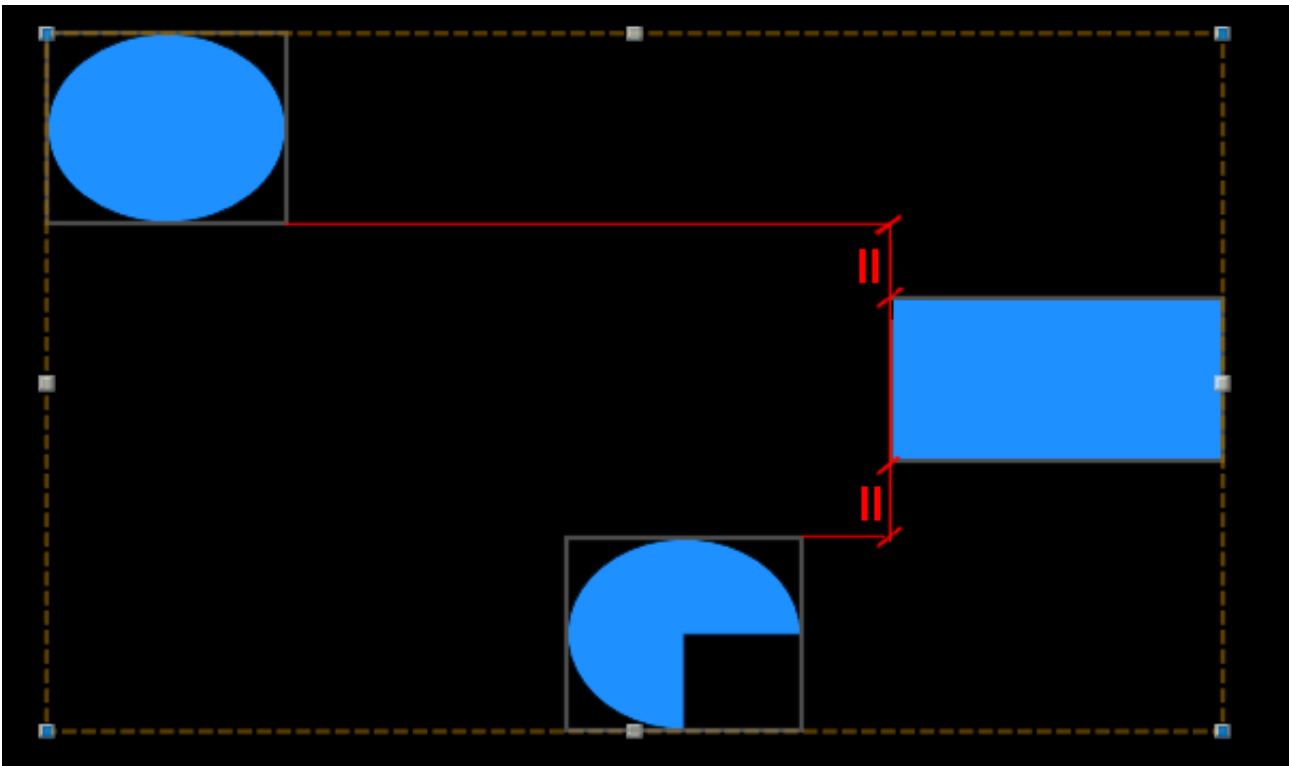
With the mouse, select the objects contained on the page one at a time while holding down the “Ctrl” key. Then choose the “Distribute vertically” option.





# CREW Manual

The selected objects are distributed vertically so that the distance between the bottom side of the first object and the top side of the second object is always equal to the distance between the bottom side of the second object and the top side of the third object (from top to bottom), and so forth depending on the number of objects there are on the page.



# CREW Manual

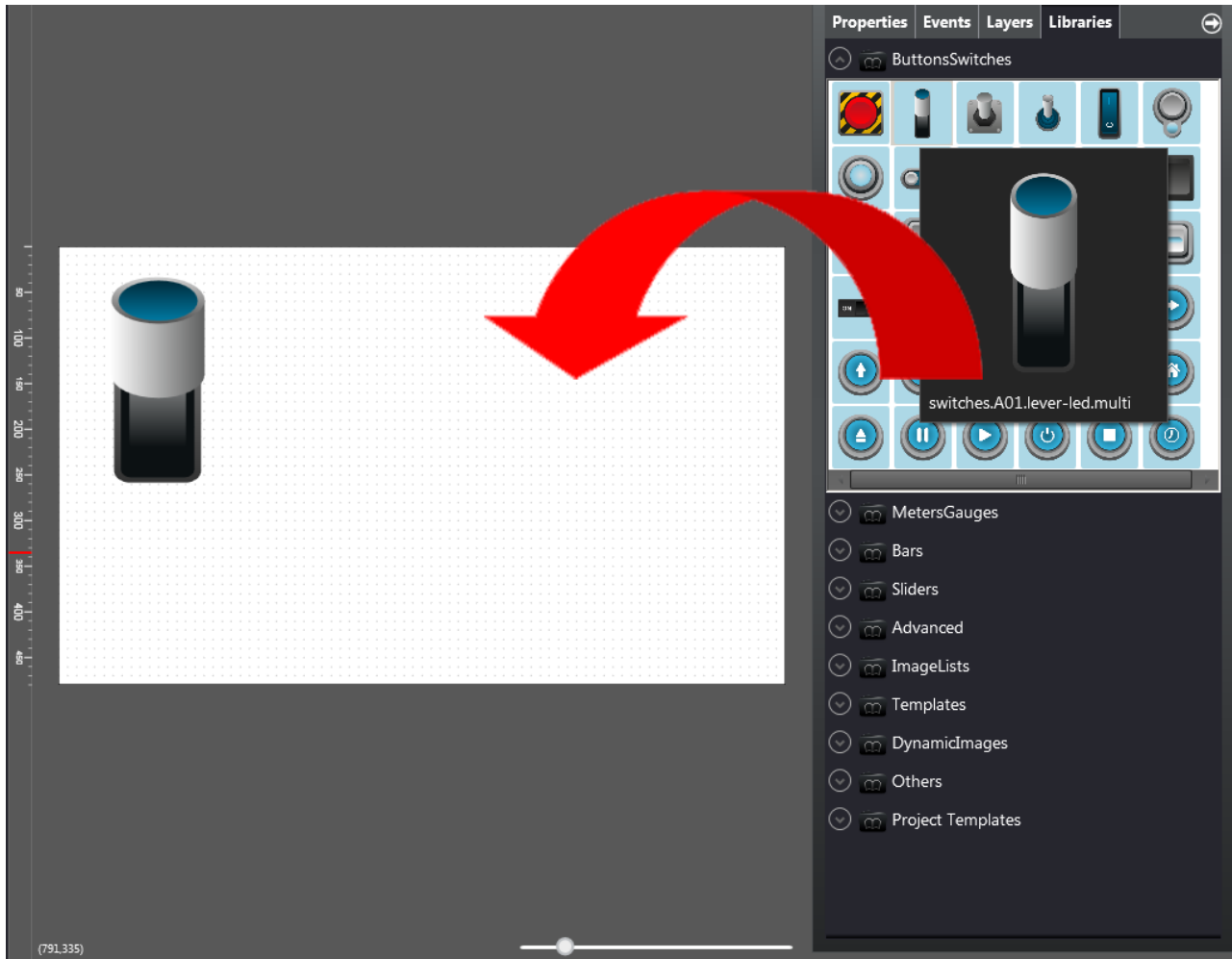
## Submenu Preview

The "Preview" function allows a real-time view of the status of any of the objects contained on the project page, up to a maximum of twenty different statuses.



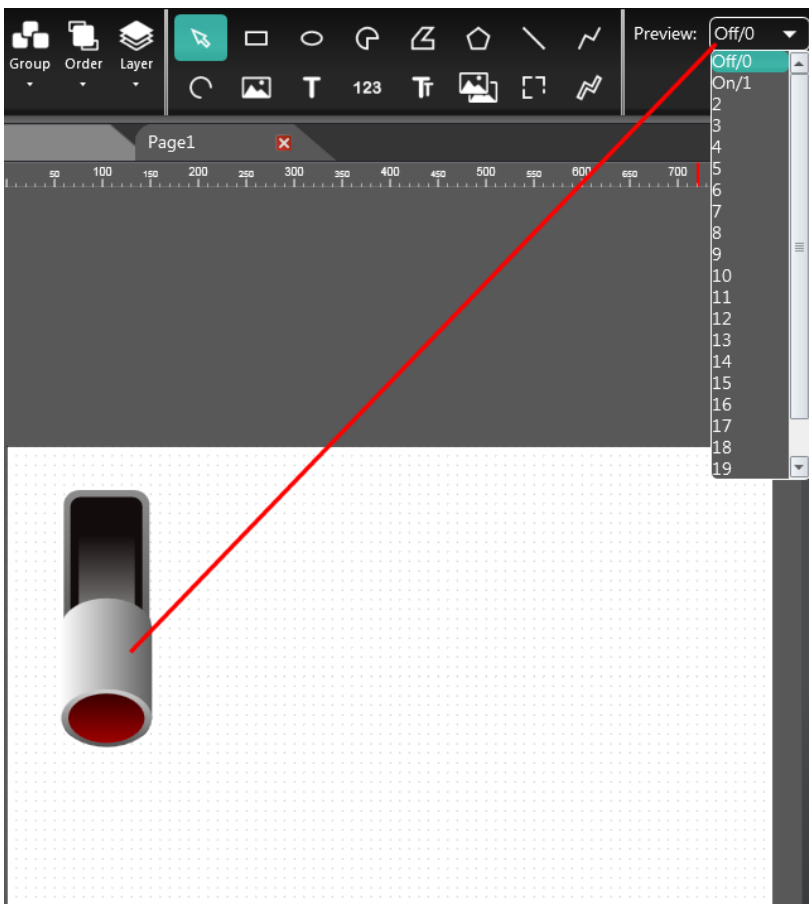
# CREW Manual

Choose an object from library (a Switch Button for example) and use the mouse to drag it around the page.



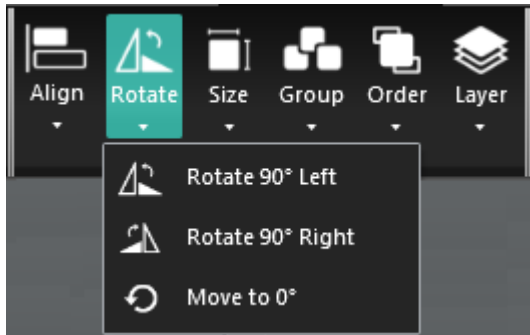
# CREW Manual

Use the “Preview” function and change the status of the object (for example the Switch Button from “1” to “0”) to view the image in real time or the text associated to the various statuses (up to 20).



# CREW Manual

## Rotation Submenu



When there is an object on a page, it is possible to use the “Rotation” option to automatically rotate it.

To enable the icons for rotation, you need to select the object that you wish to work on.

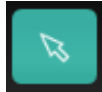
The following rotation options are available:

- Rotate 90° left: to turn the selected object 90° anticlockwise.
- Rotate 90° right: to turn the selected object 90° clockwise.
- Move to 0°: move the selected object to an inclination of 0°.

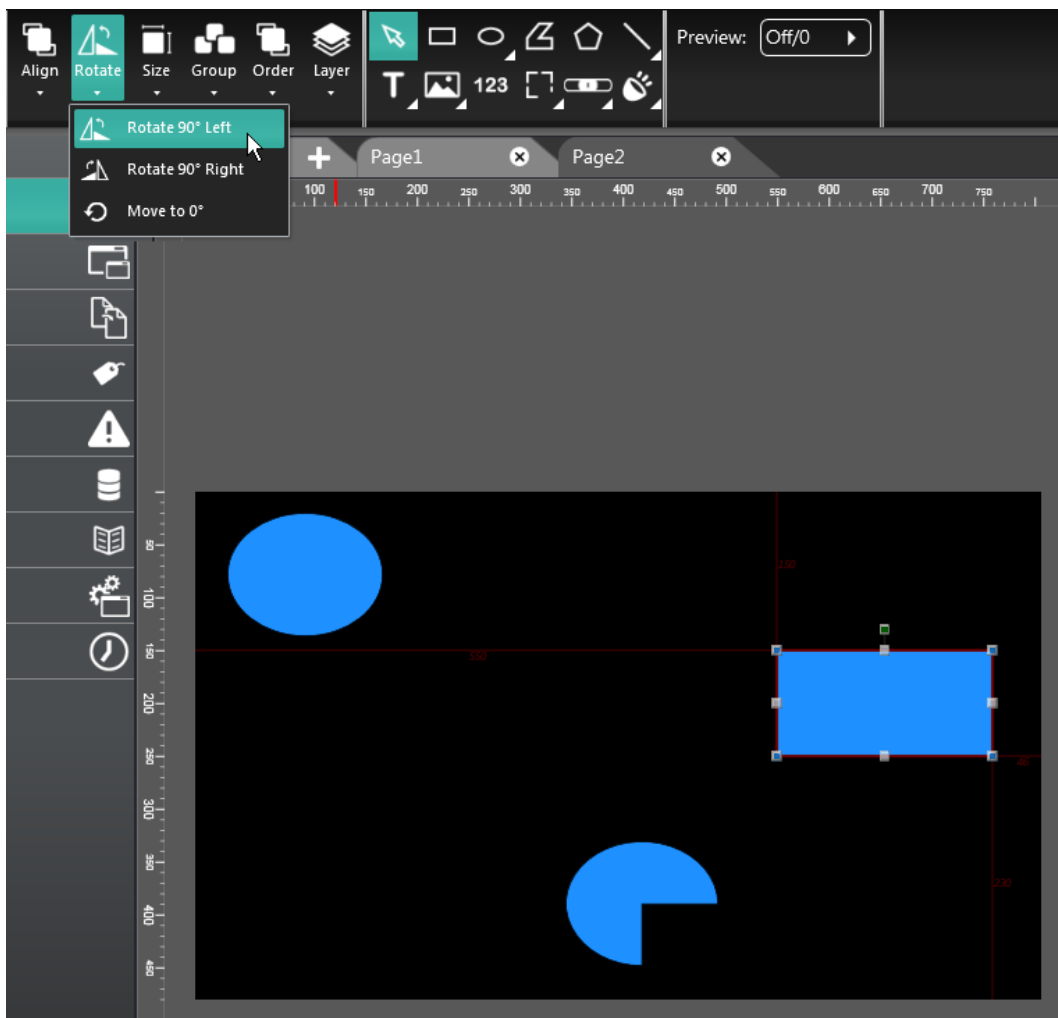
# CREW Manual

## Rotate 90° left

Use the “Selection” tool.



Use the mouse to select one of the objects contained on the page (the example uses the rectangle) and choose the “Rotate 90° left” option.



# CREW Manual

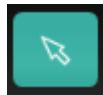
The object will turn 90° anticlockwise (as you can see in the following image with the rectangle).



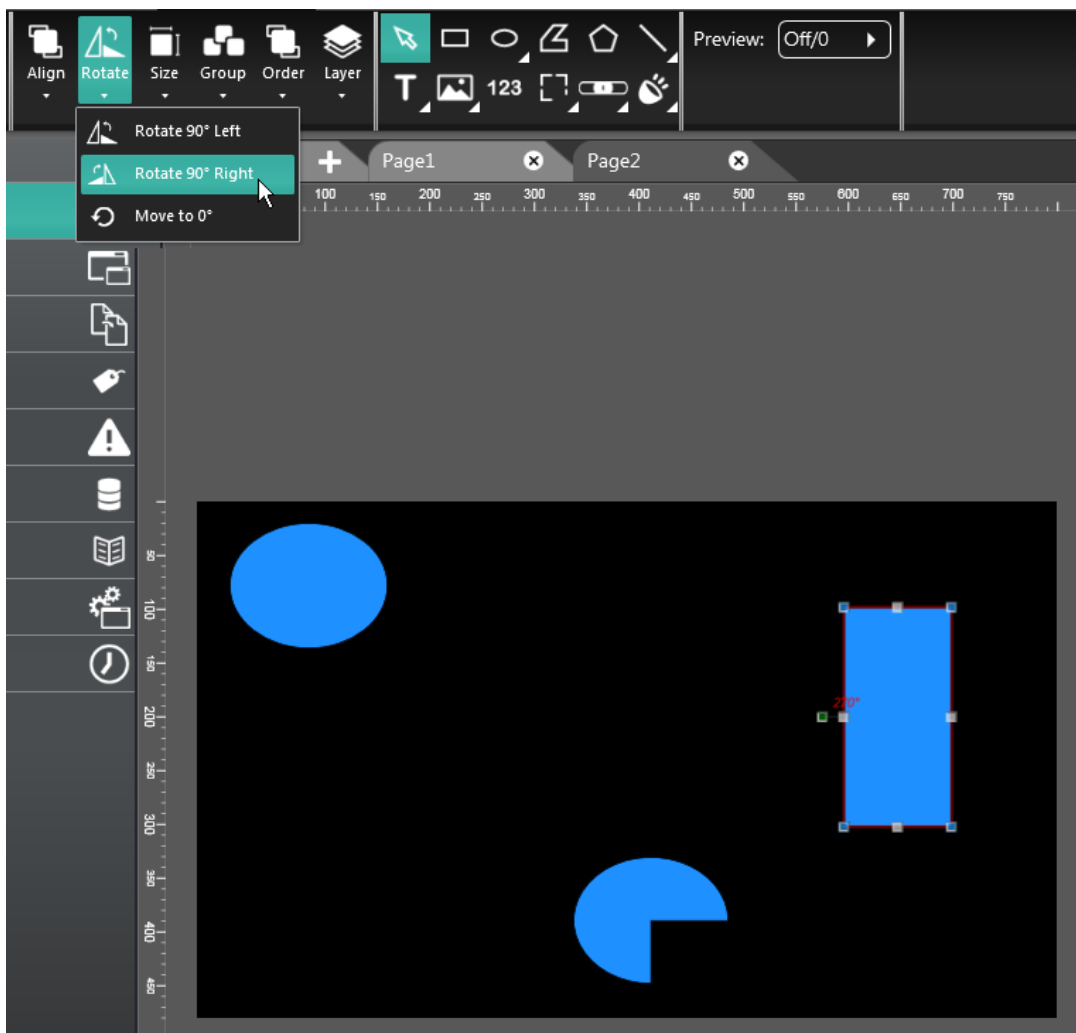
# CREW Manual

## Rotate 90° right

Use the “Selection” tool.



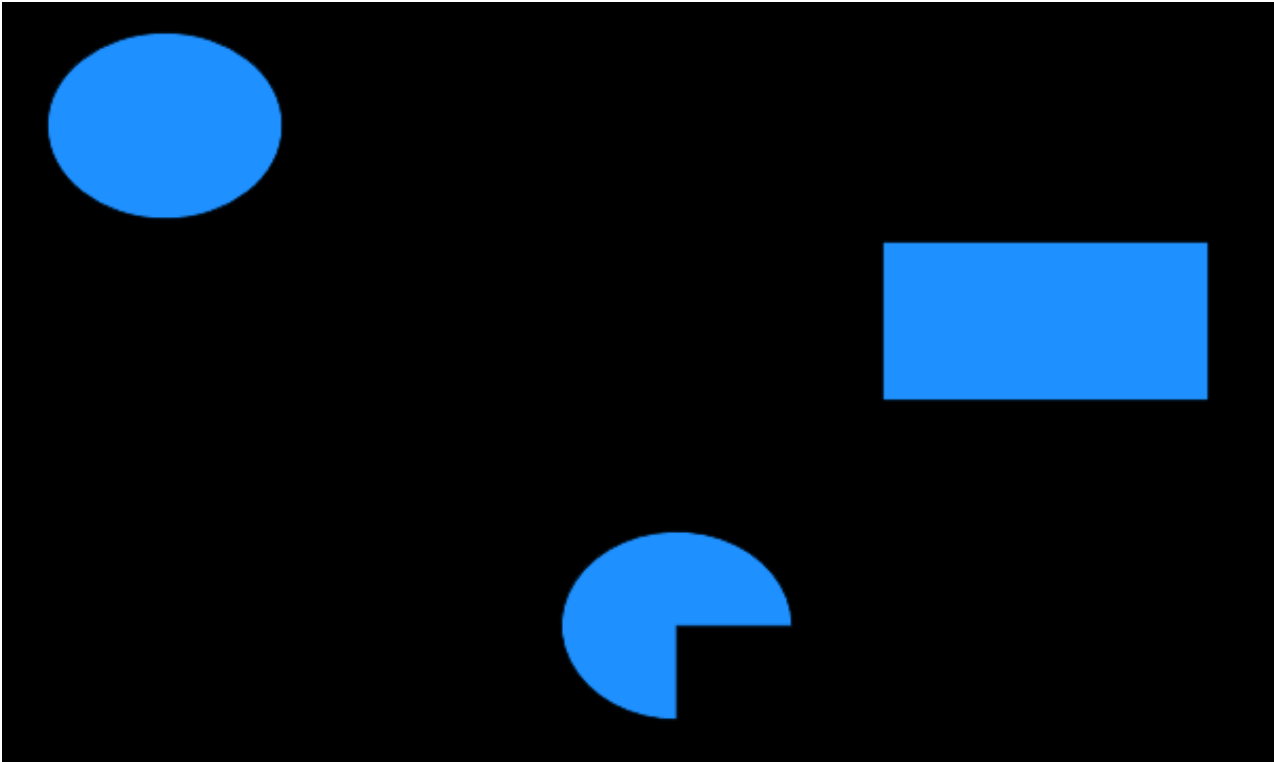
Use the mouse to select one of the objects contained on the page (the example uses the rectangle) and choose “Rotate 90° right”.





# CREW Manual

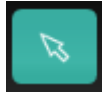
The object will turn 90° clockwise (as you can see in the following image with the rectangle).



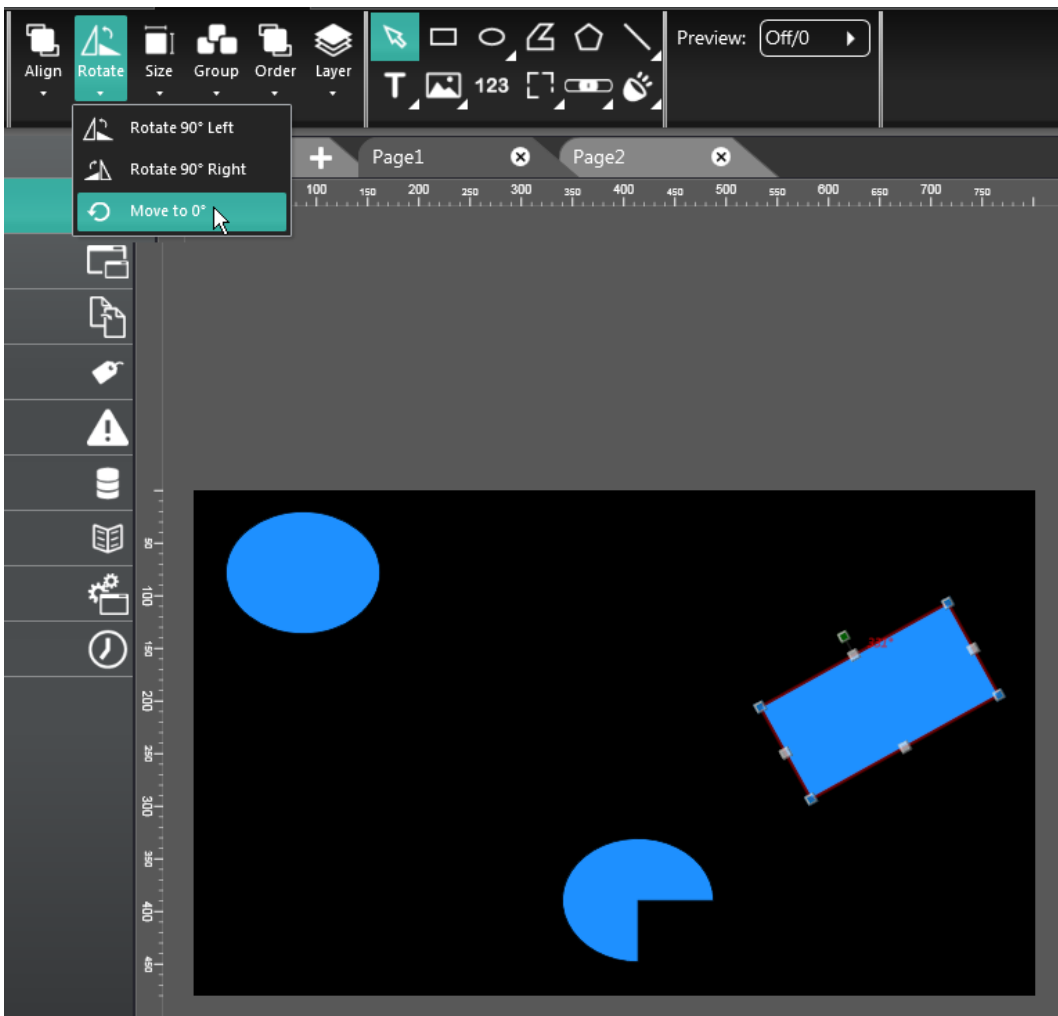
# CREW Manual

## Move to 0°

Use the “Selection” tool.

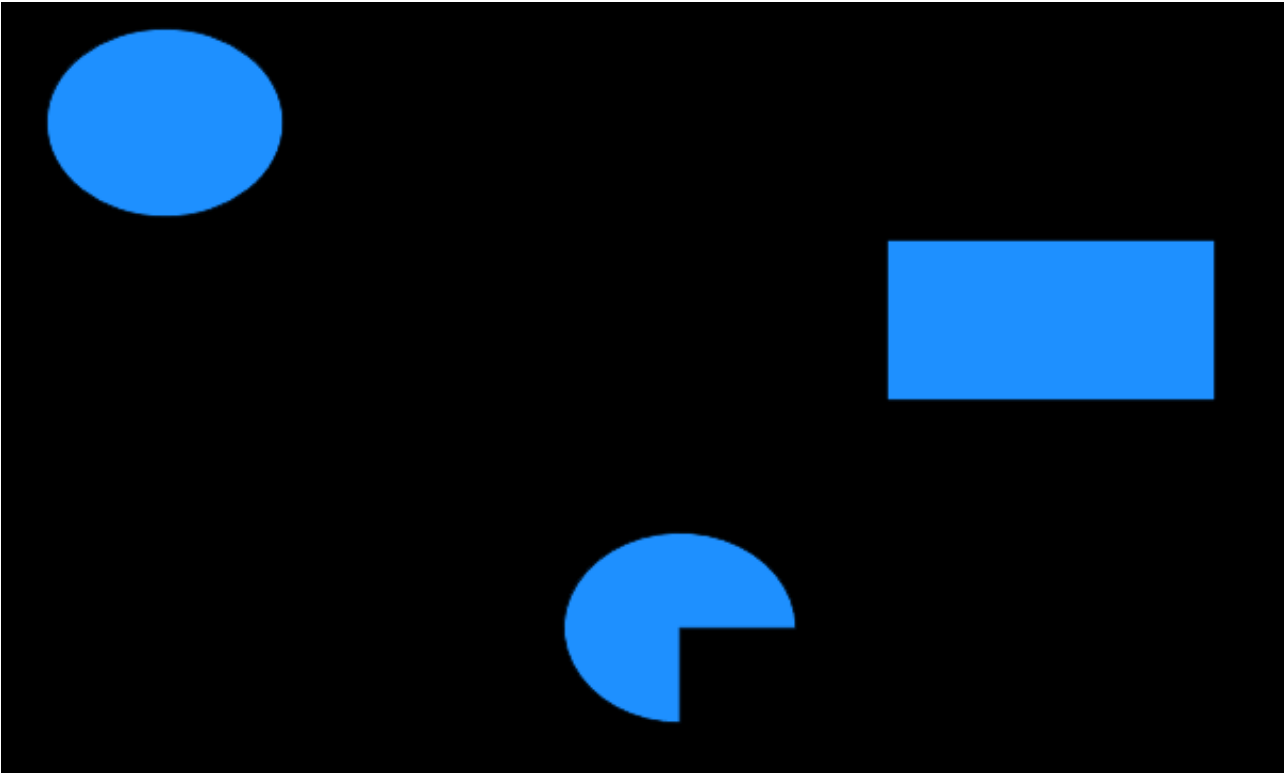


Use the mouse to select one of the objects contained on the page (the example uses the rectangle) and choose “Move to 0°”.



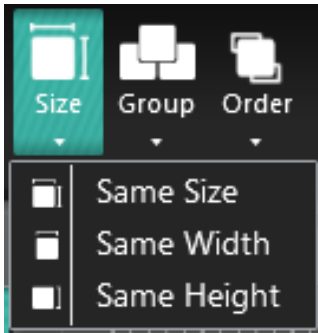
# CREW Manual

The object will be moved to its horizontal position, namely 0° (as you can see in the following image with the rectangle).



# CREW Manual

## Size Submenu



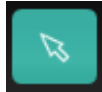
When there are two or more objects on a page, it is possible to use the “Size” option to automatically perform the following re-sizing options.

- Same size: resize all of the objects to the same size as the object selected first.
- Same length: resize the length of all the selected objects to the same length as the object selected first.
- Same height: resize the height of all the selected objects to the same height as the object selected first.

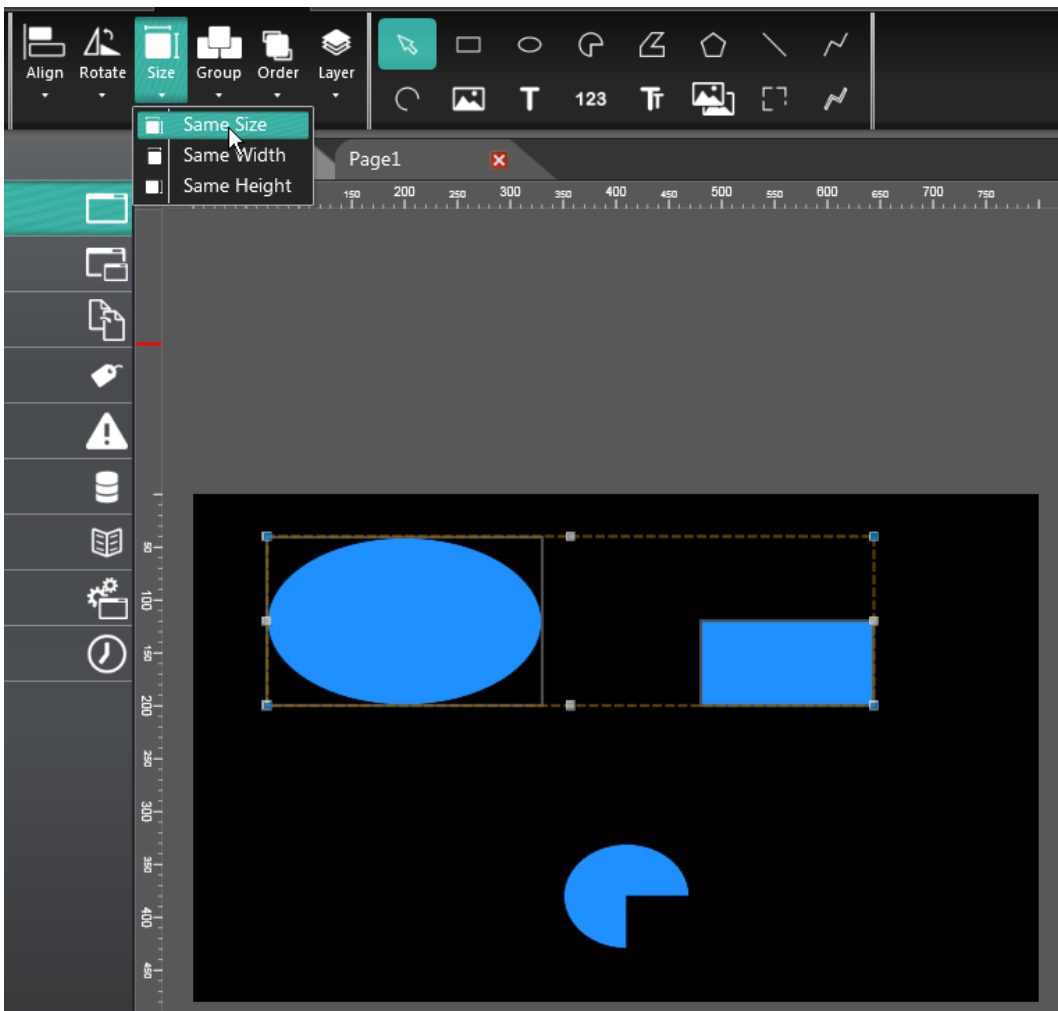
# CREW Manual

## Same size

Use the “Selection” tool.

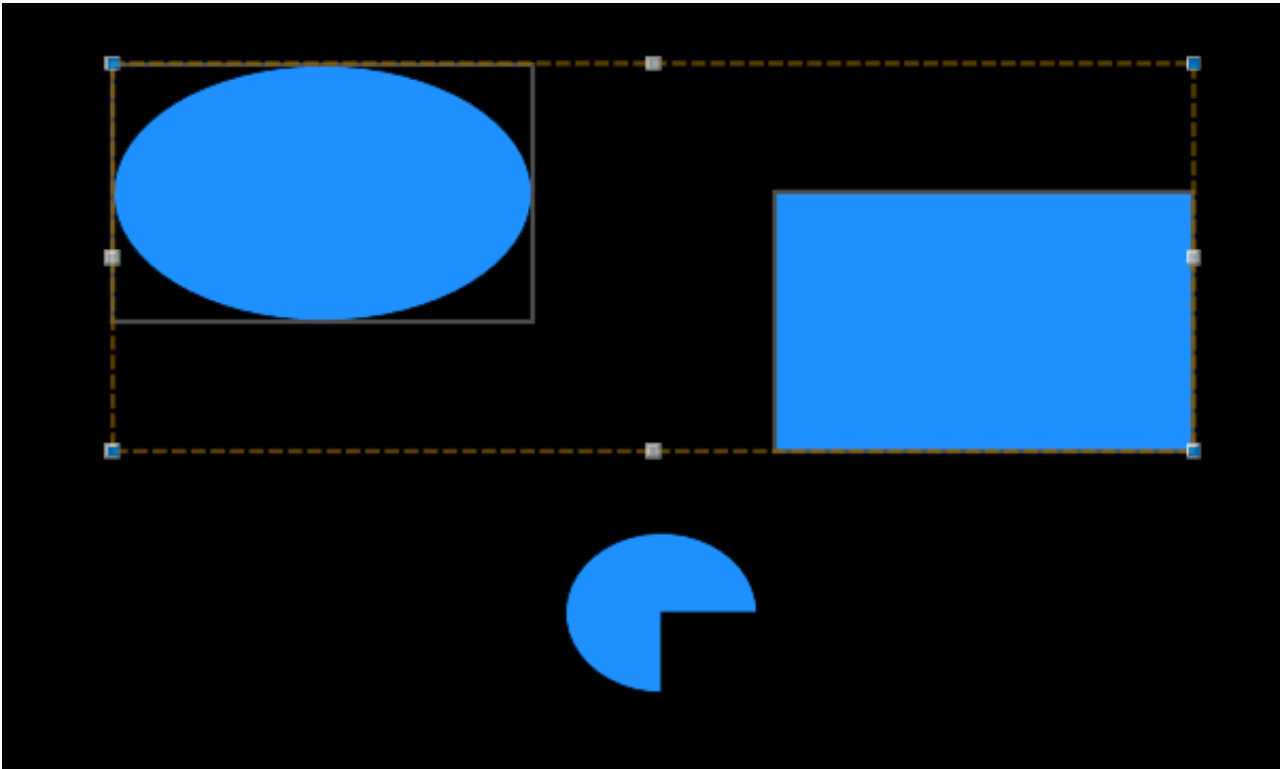


Use the mouse to select two or more objects on the page (the example uses the ellipse and the rectangle, with the ellipse being the first object to be selected) while holding down the “Ctrl” key, then choose the “Same size” option.



## CREW Manual

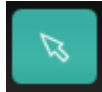
The objects are resized to the same size as the first object to be selected (in the example the rectangle turns the same size as the ellipse, which was selected first).



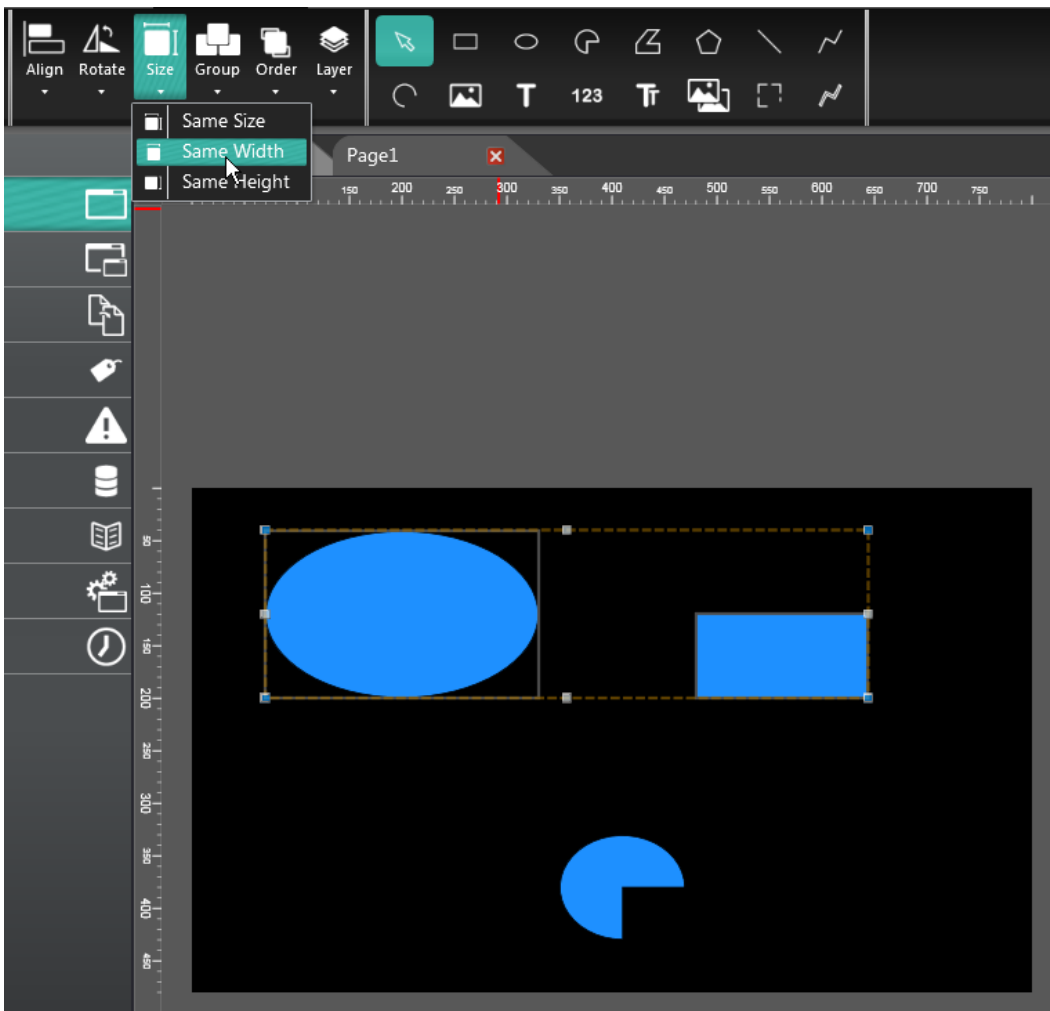
# CREW Manual

## Same length

Use the “Selection” tool.

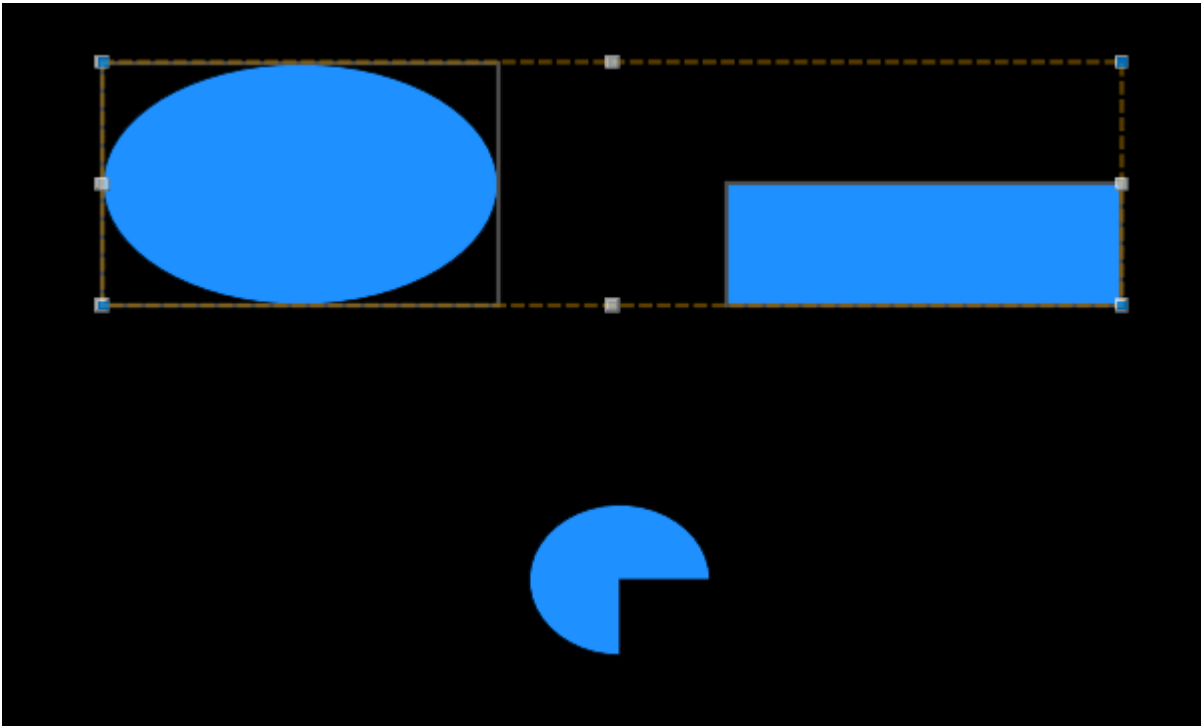


Use the mouse to select two or more objects on the page (the example uses the ellipse and the rectangle, with the ellipse being the first object to be selected) while holding down the “Ctrl” key, then choose the “Same length” option.



# CREW Manual

The length of the objects is resized to the same size as the length of the first object to be selected (in the example the length of the rectangle turns the same length as the ellipse, which was selected first).

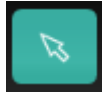




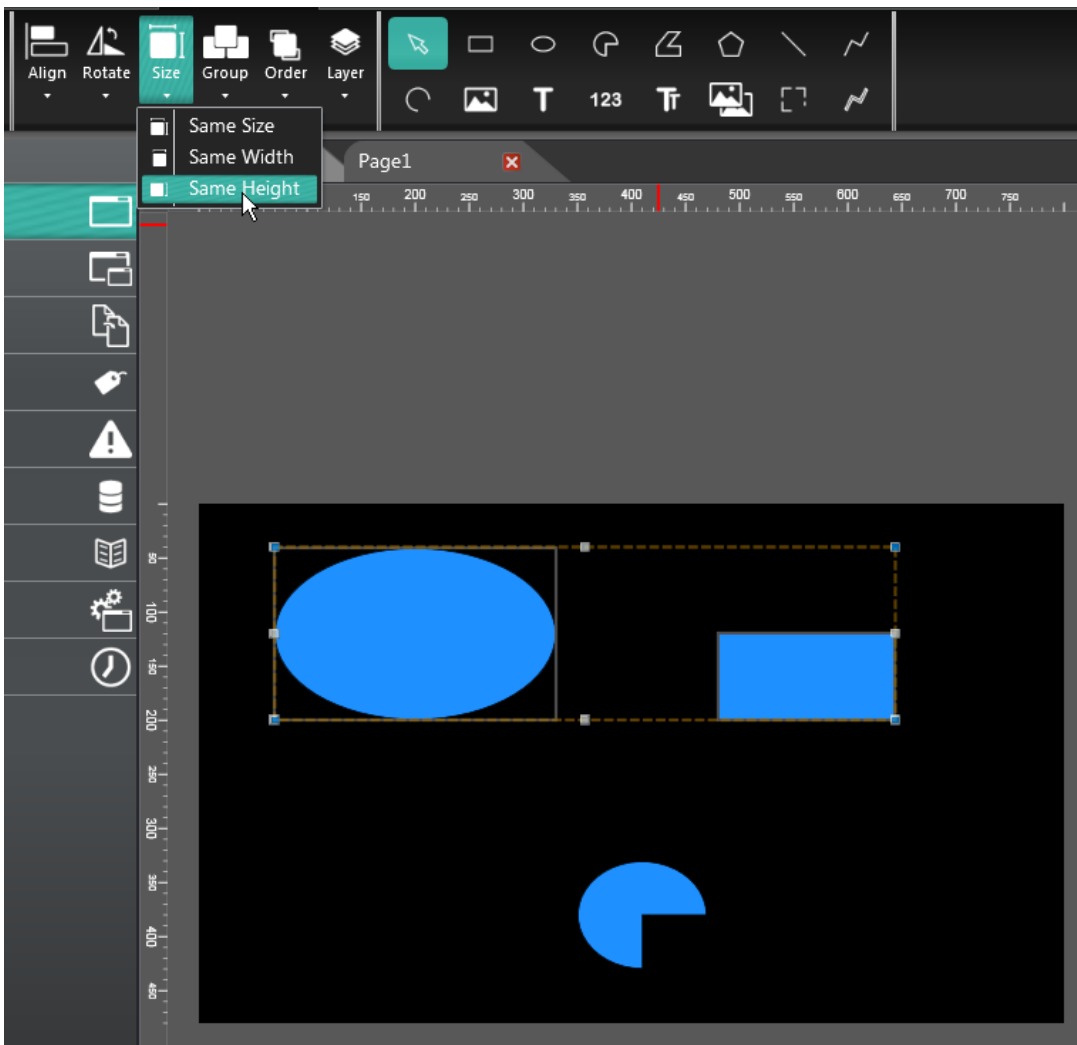
# CREW Manual

## Same height

Use the “Selection” tool.

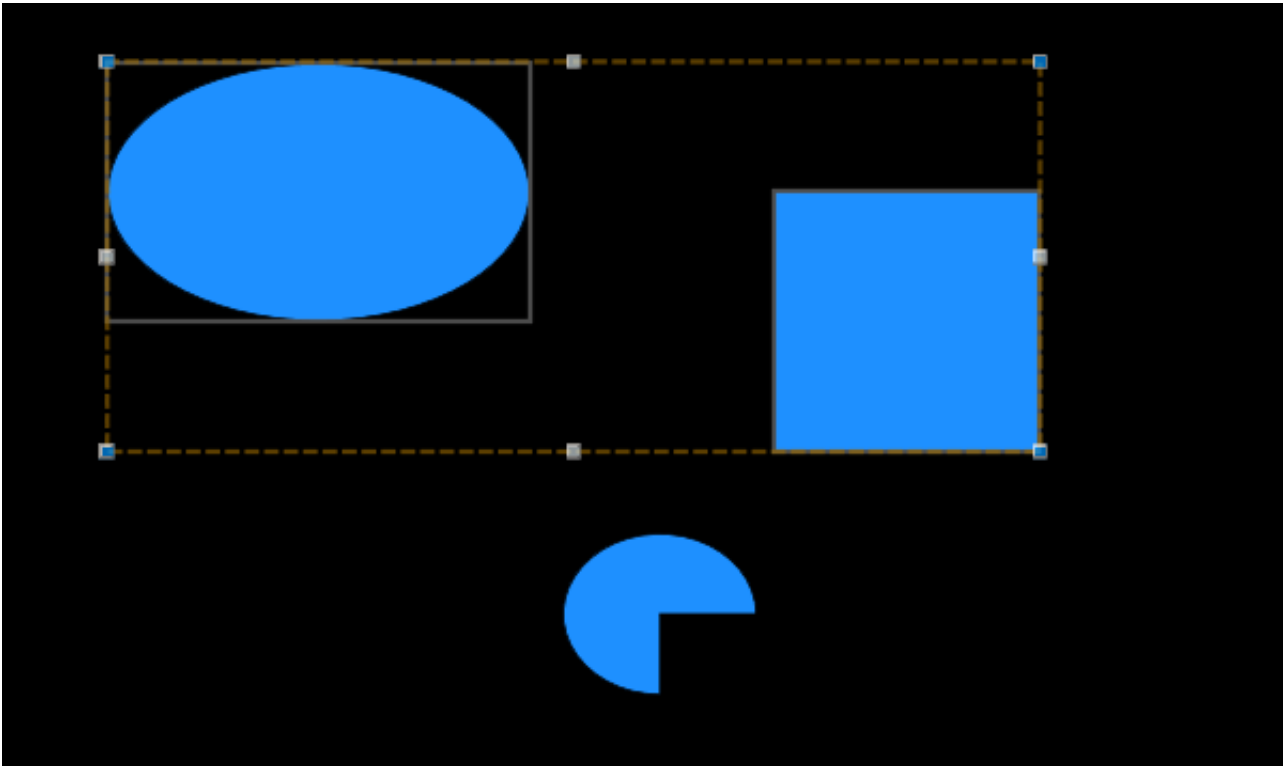


Use the mouse to select two or more objects on the page (the example uses the ellipse and the rectangle, with the ellipse being the first object to be selected) while holding down the “Ctrl” key, then choose the “Same height” option.



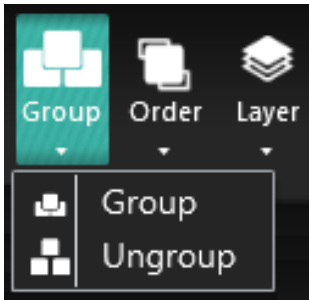
# CREW Manual

The height of the objects is resized to the same size as the height of the first object to be selected (in the example the height of the rectangle turns the same height as the ellipse, which was selected first).



# CREW Manual

## Grouping Submenu



The “Grouping” function is used to group together two or more graphic elements so as to run cumulative operations on all of the objects at the same time.

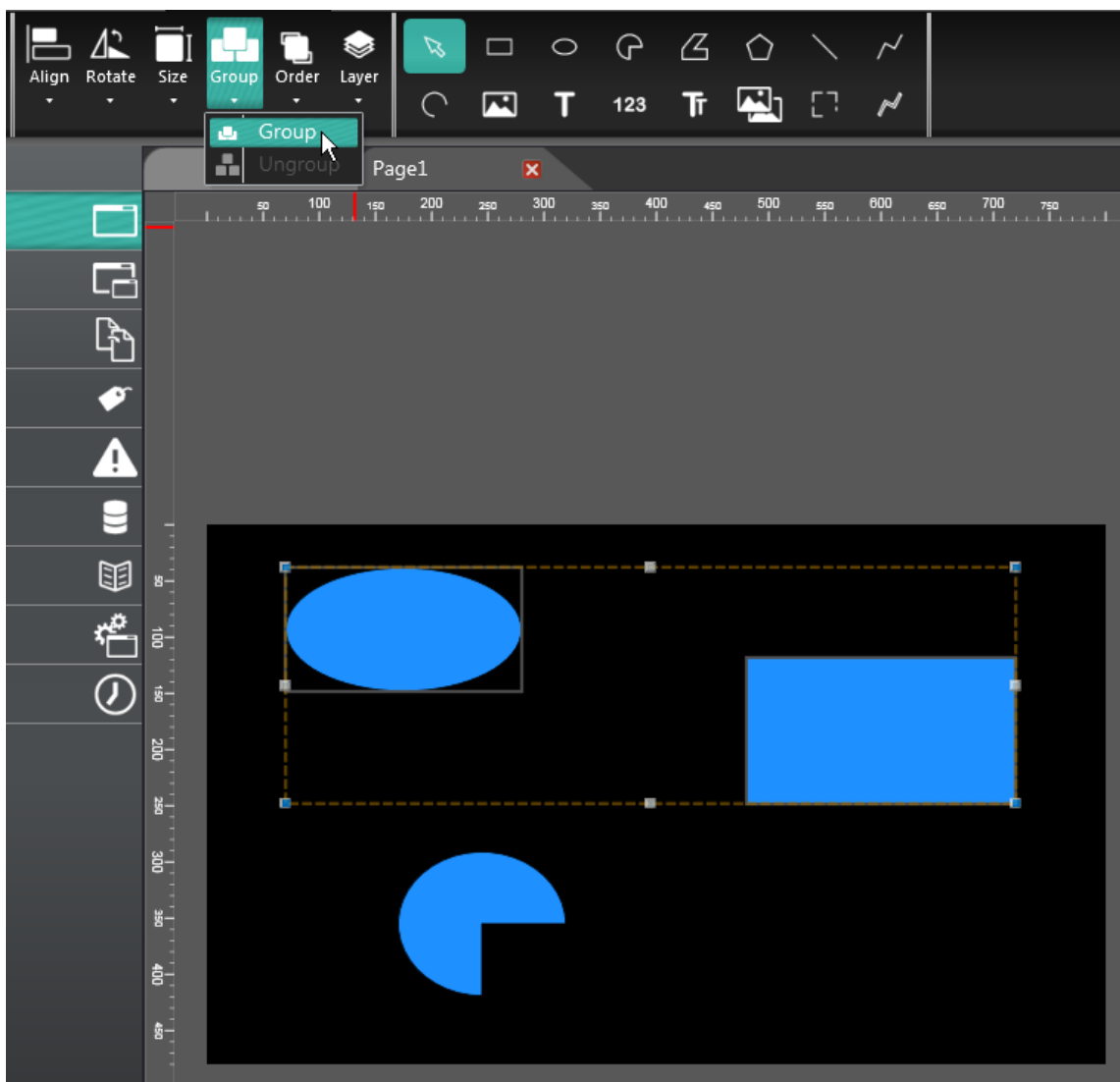
The available options are as follows:

- Grouping: to group together two or more selected elements.
- Separate: to separate two or more (previously grouped together) selected elements.

# CREW Manual

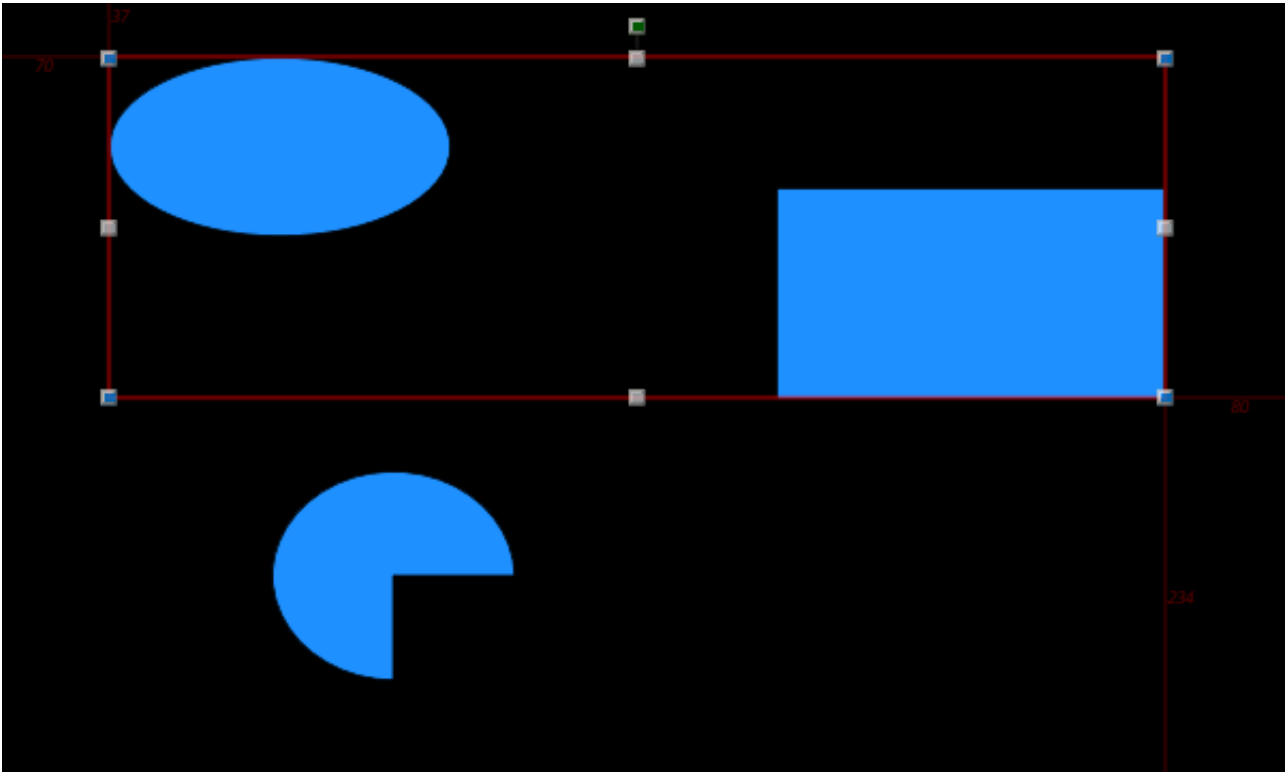
## Grouping

Use the mouse to draw an area that encloses two or more objects, to select them at the same time, or use the mouse to click the objects one at a time, holding down the “Ctrl” key, and choosing the “Grouping” option.



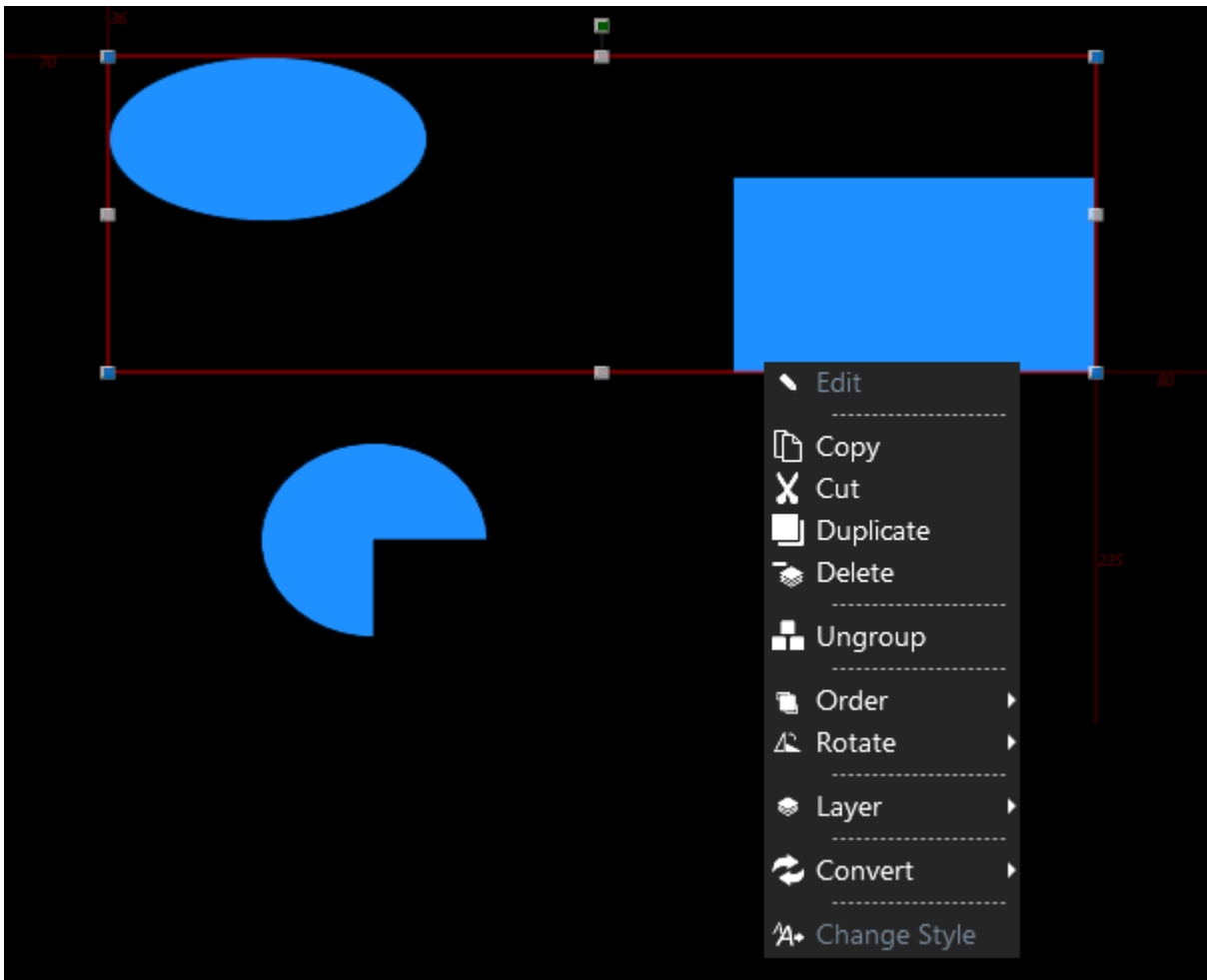
# CREW Manual

The selected objects then become a single structure.



## CREW Manual

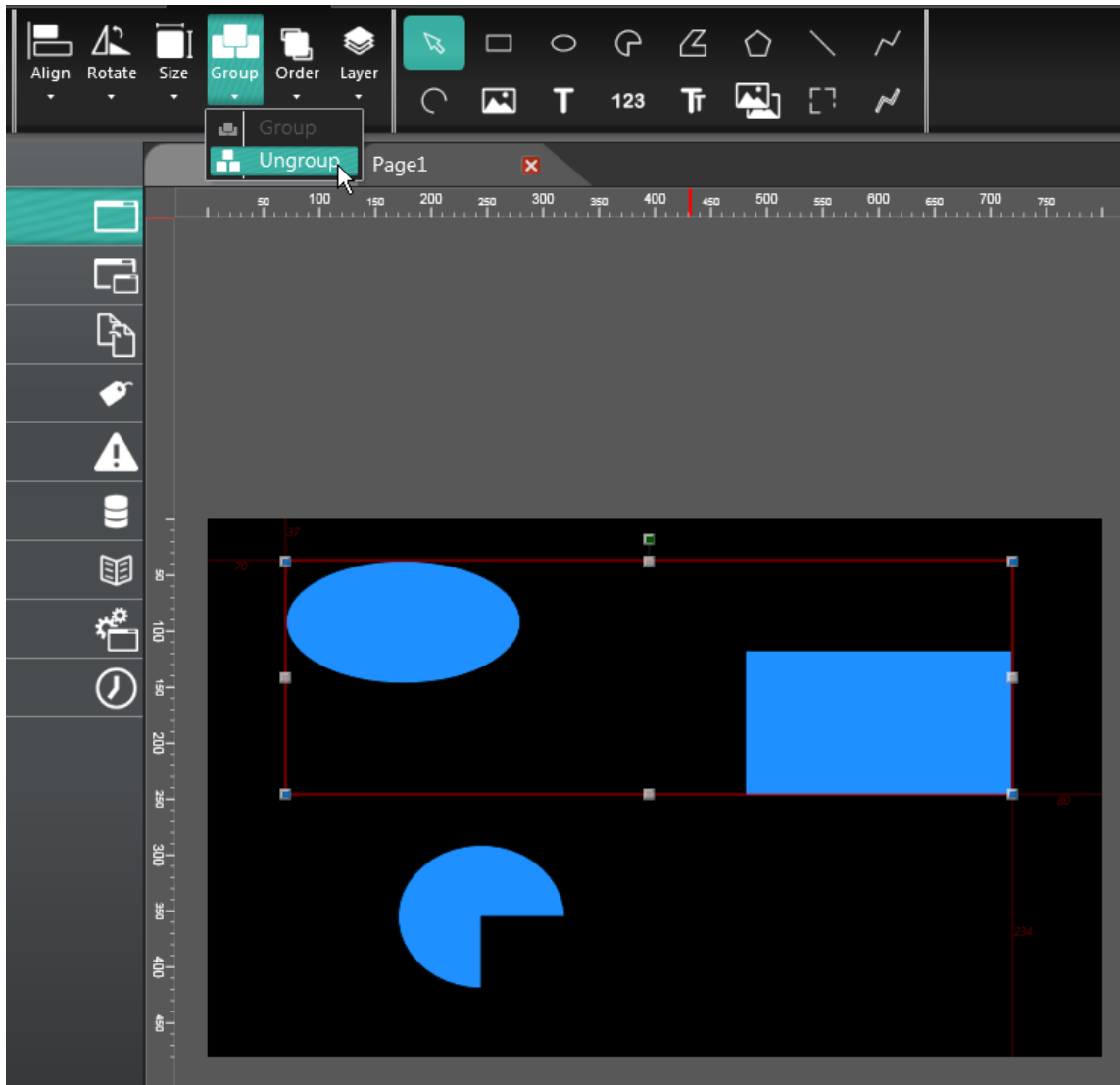
It is now possible to perform cumulative movements, resizing or duplicating on the group. To do so, select the group, right click with the mouse and choose the necessary operation.



# CREW Manual

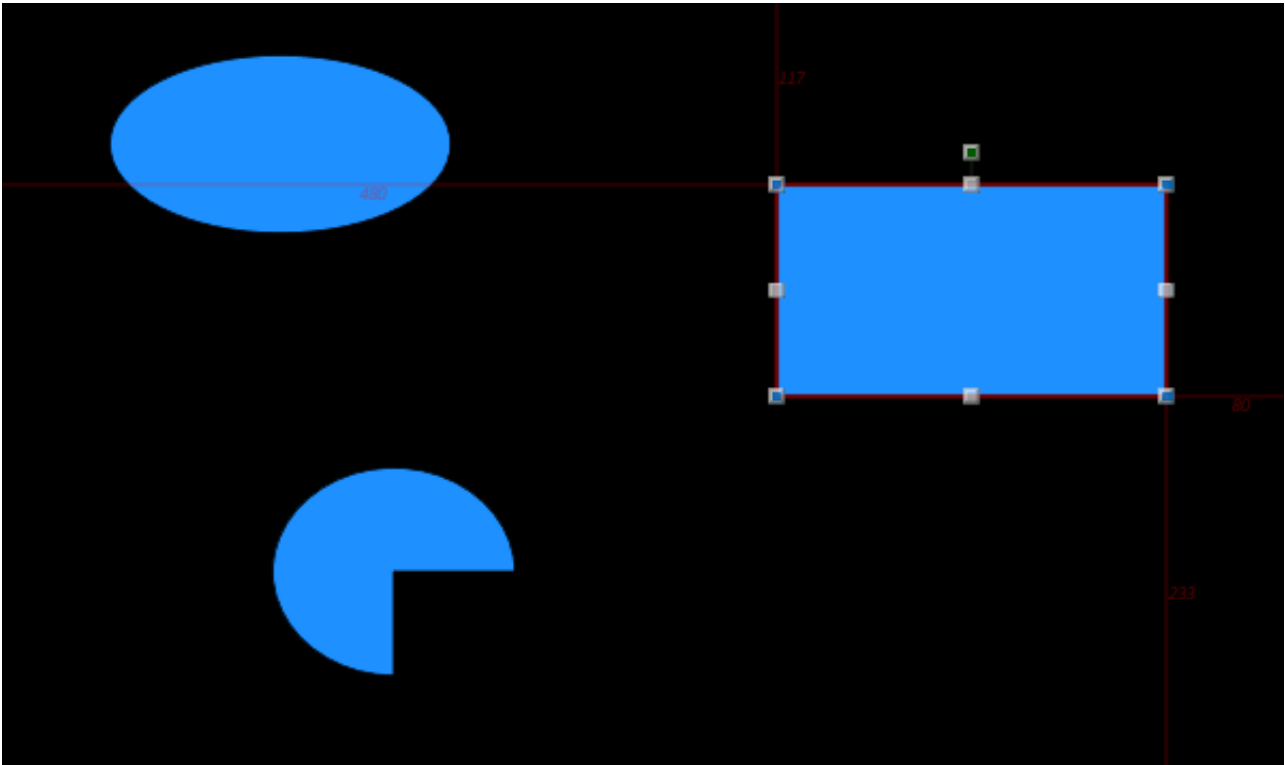
## Separate

Select a group of objects and click the “Separate” key.



# CREW Manual

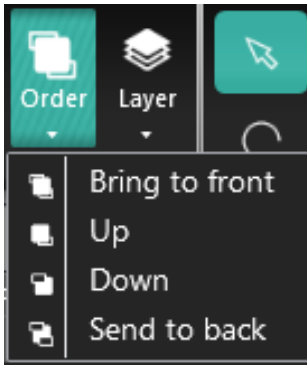
The elements go back to being individually modifiable.





# CREW Manual

## Order Submenu



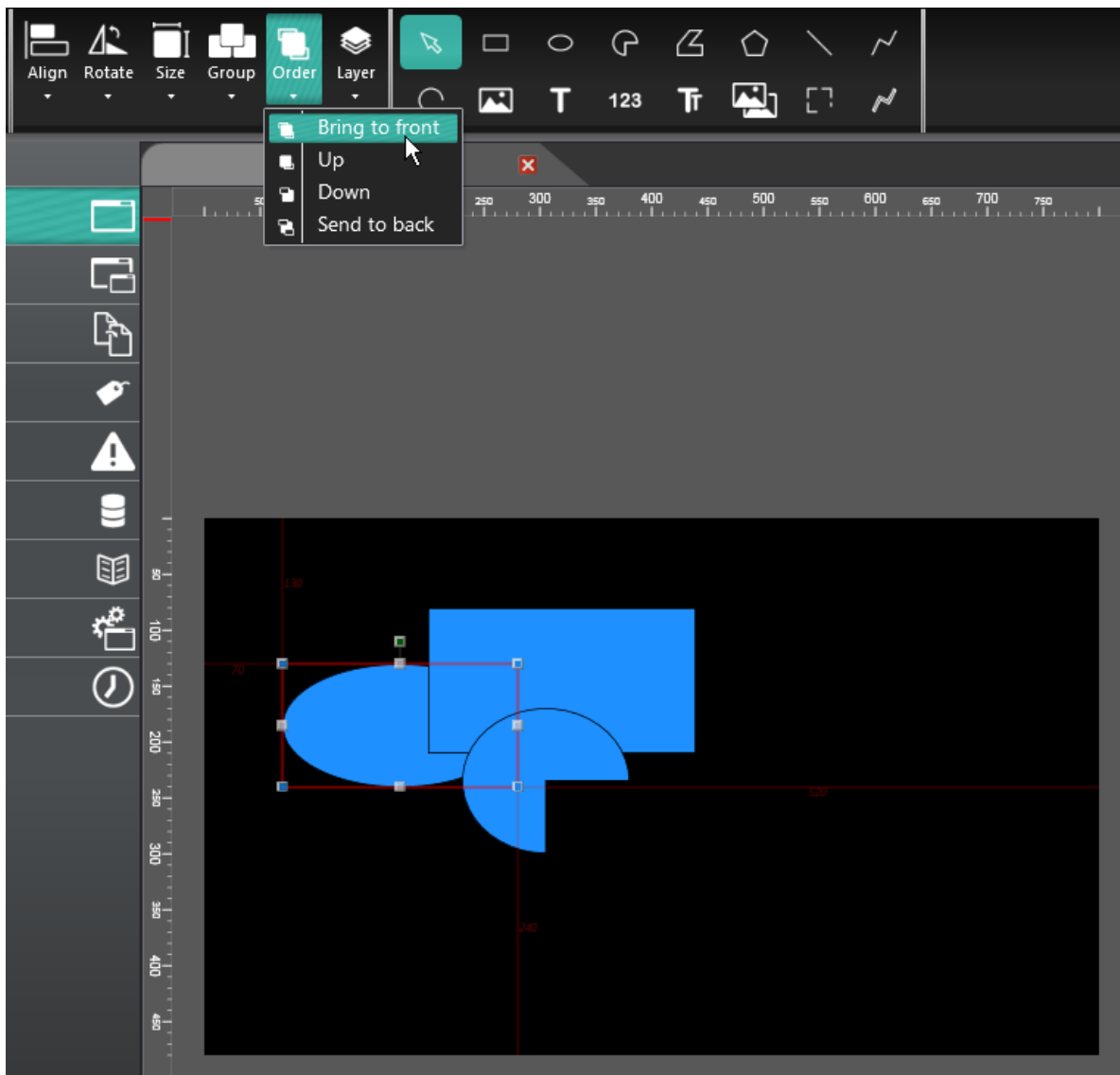
When multiple objects overlap on a page, it is possible to establish a viewing hierarchy for the overlapping objects. Simply select one of the objects and specify what depth level to position it, by pressing one of the following options:

- Place in foreground: bring the selected object to the surface.
- Up: increase the level of the selected object.
- Down: decrease the level of the selected object.
- Move down: move the selected object to the lower level.

# CREW Manual

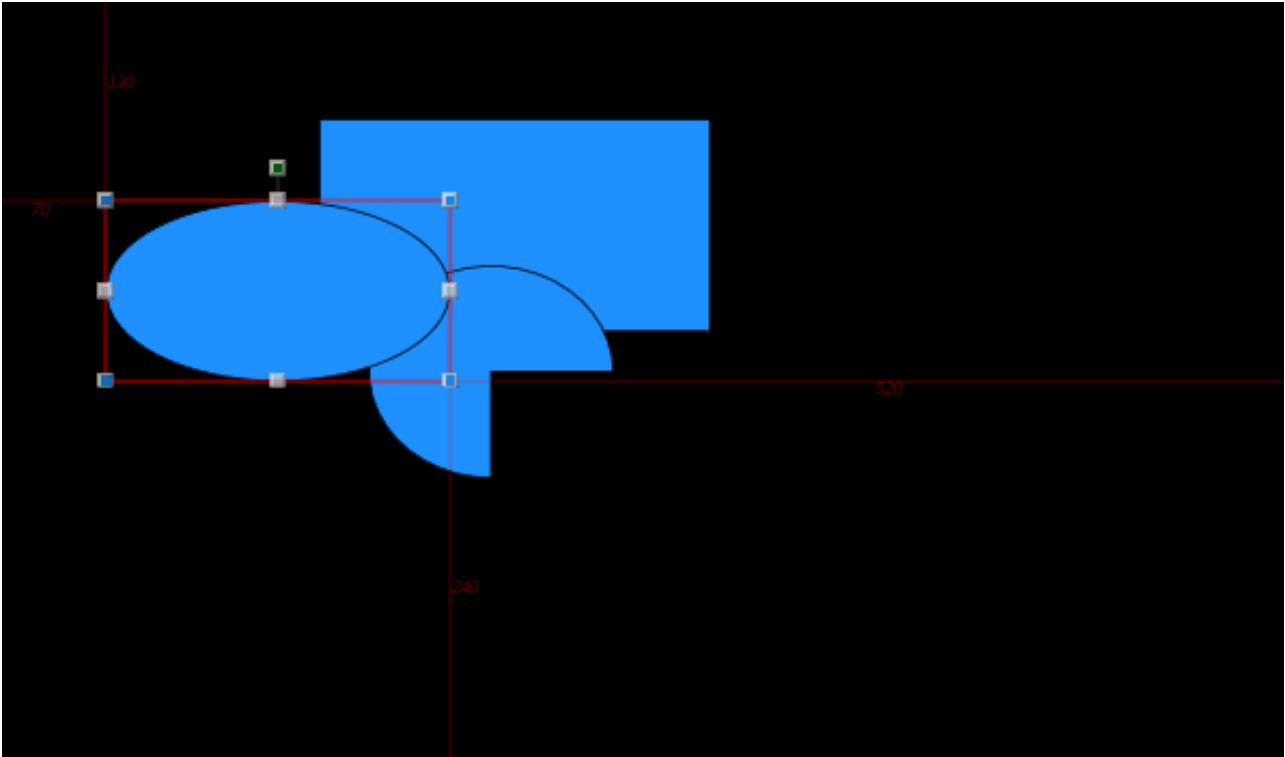
## Place in foreground

Select one of the overlapping objects (in the example, we choose the ellipse from the ellipse, rectangle and circular sector) and choose the “Place in foreground” option.



# CREW Manual

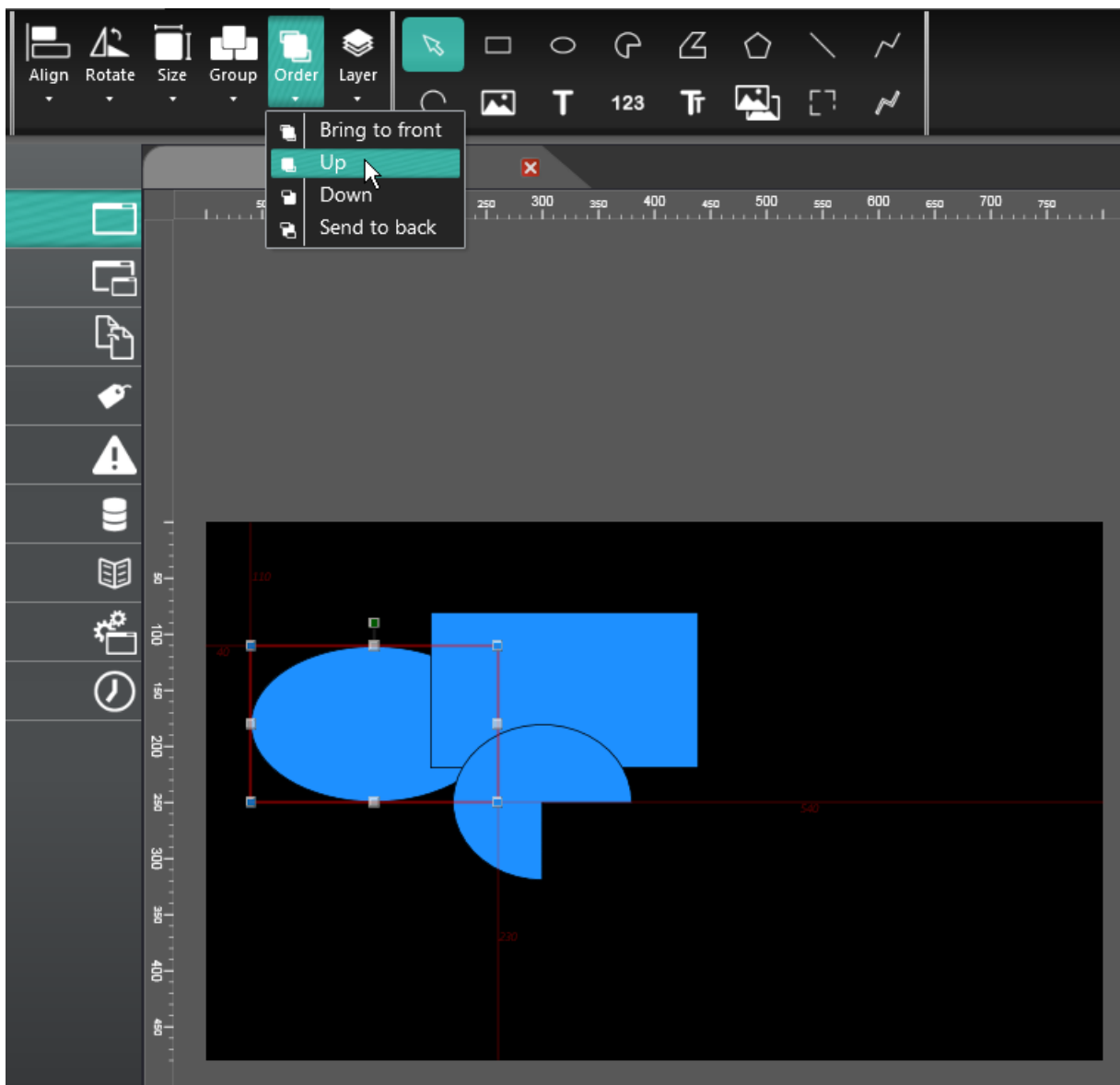
The selected object is placed in the foreground and covers all of the other objects (in the example the ellipse is the first object at the surface).



# CREW Manual

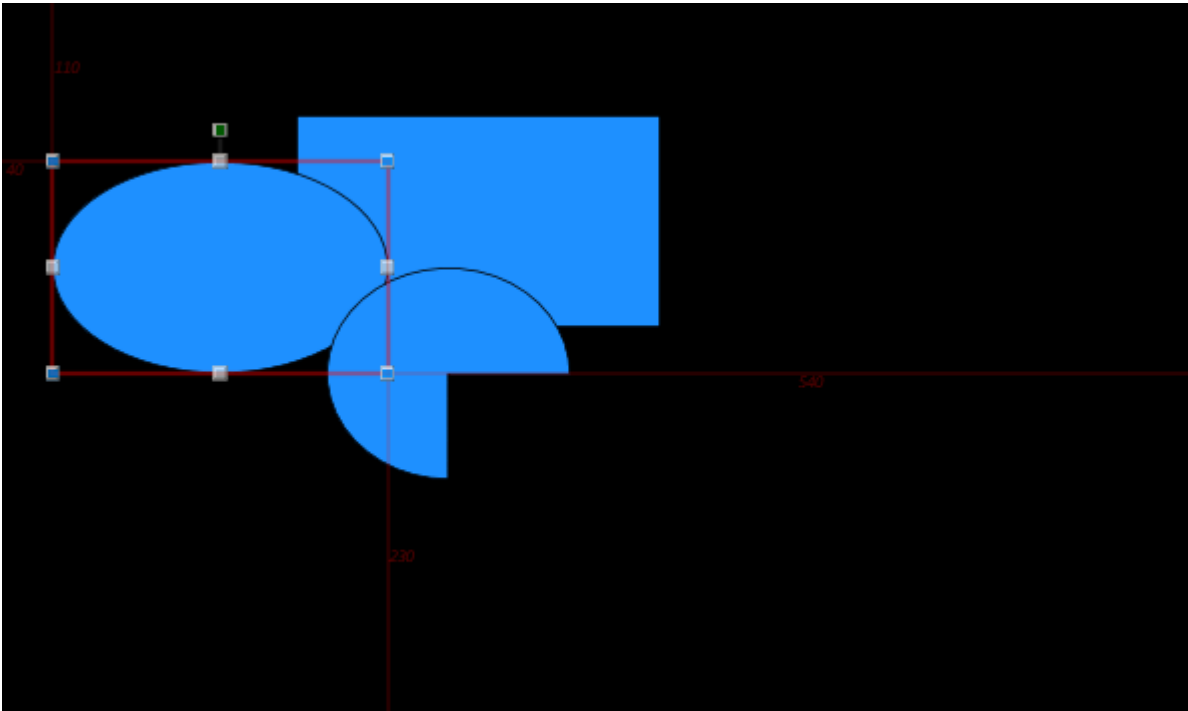
## Up

Select one of the overlapping objects (in the example, we choose the ellipse from the ellipse, rectangle and circular sector) and choose the “Up” option.



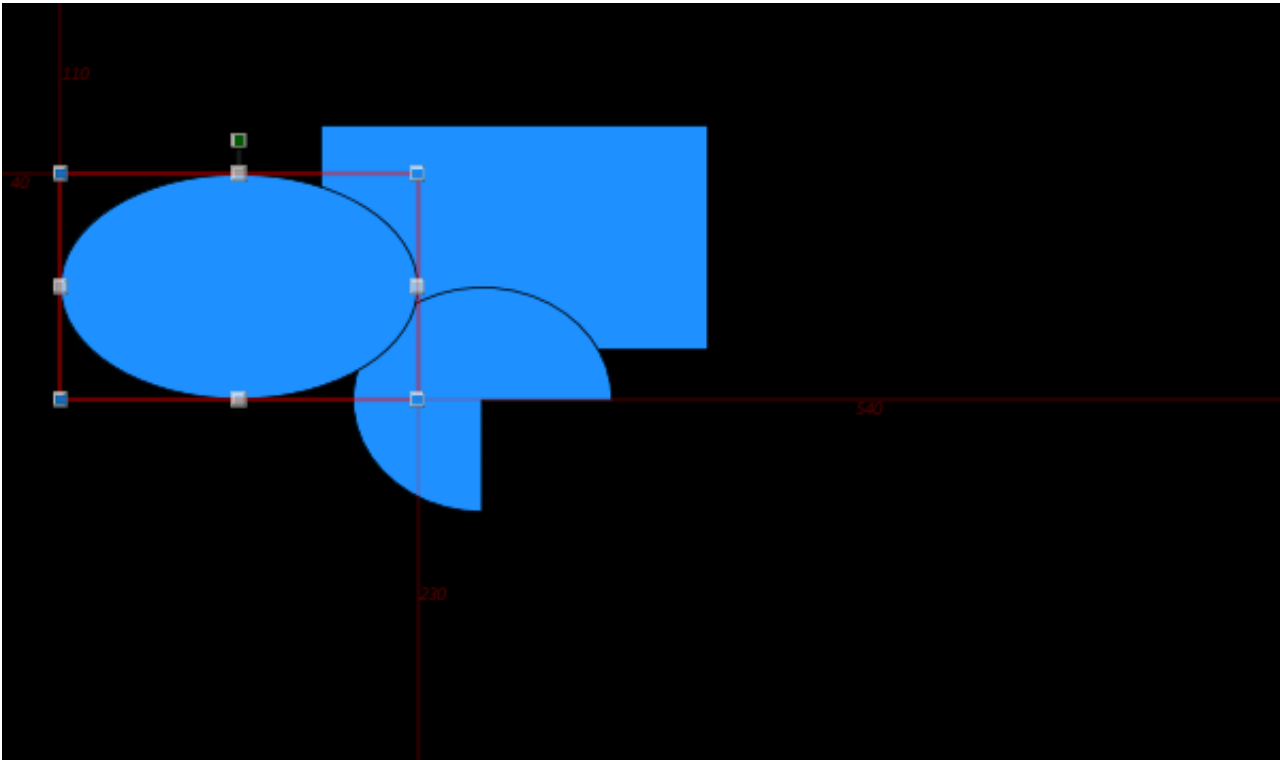
# CREW Manual

The selected object is moved to the surface by one level and only covers the object that was above it before the command was clicked (in the example the ellipse is now covering the rectangle).



# CREW Manual

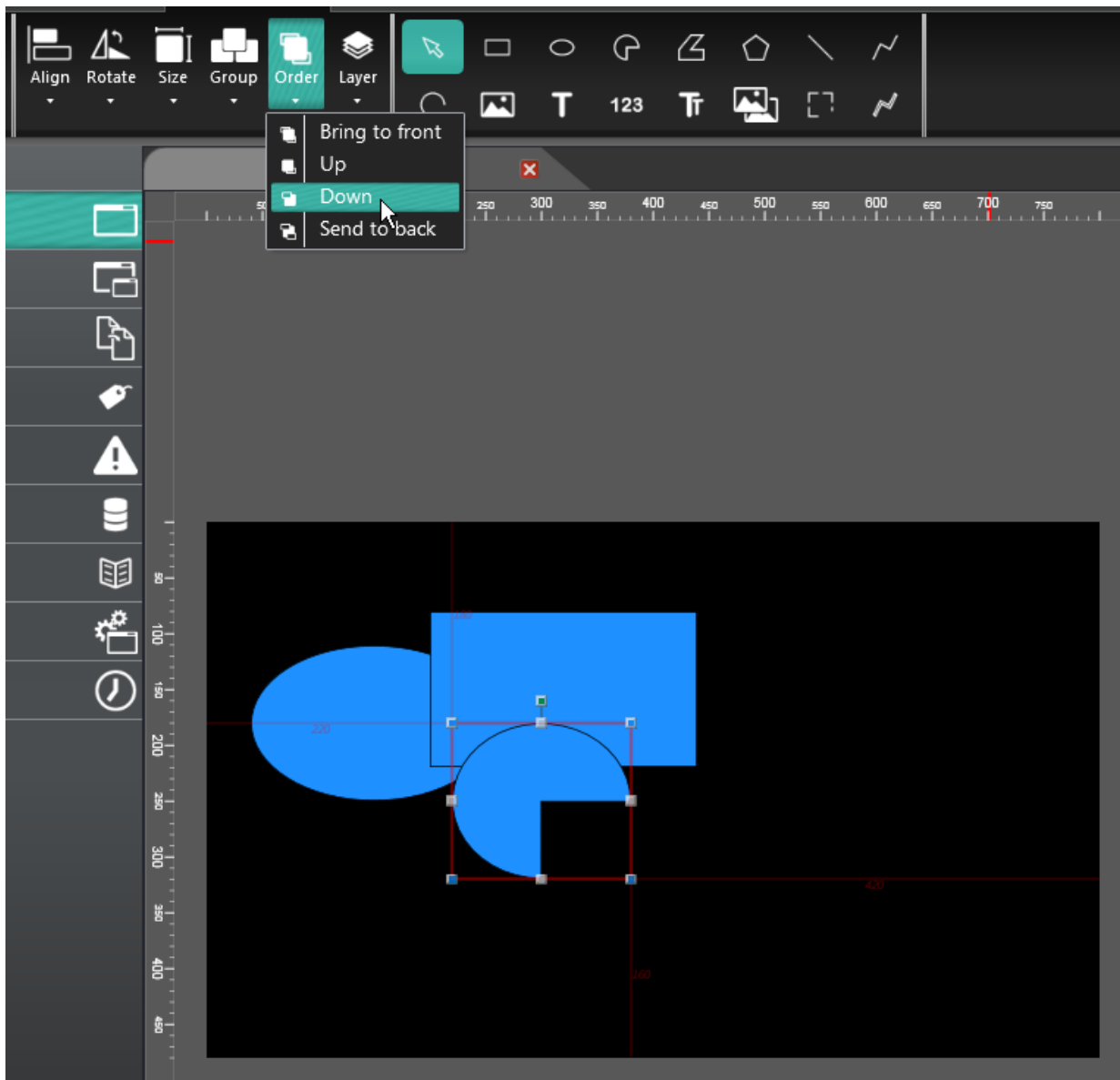
Every time this action is repeated (clicking “Up”), the selected object goes up a position in its viewing level (for example, the ellipse also covers the circular sector).



# CREW Manual

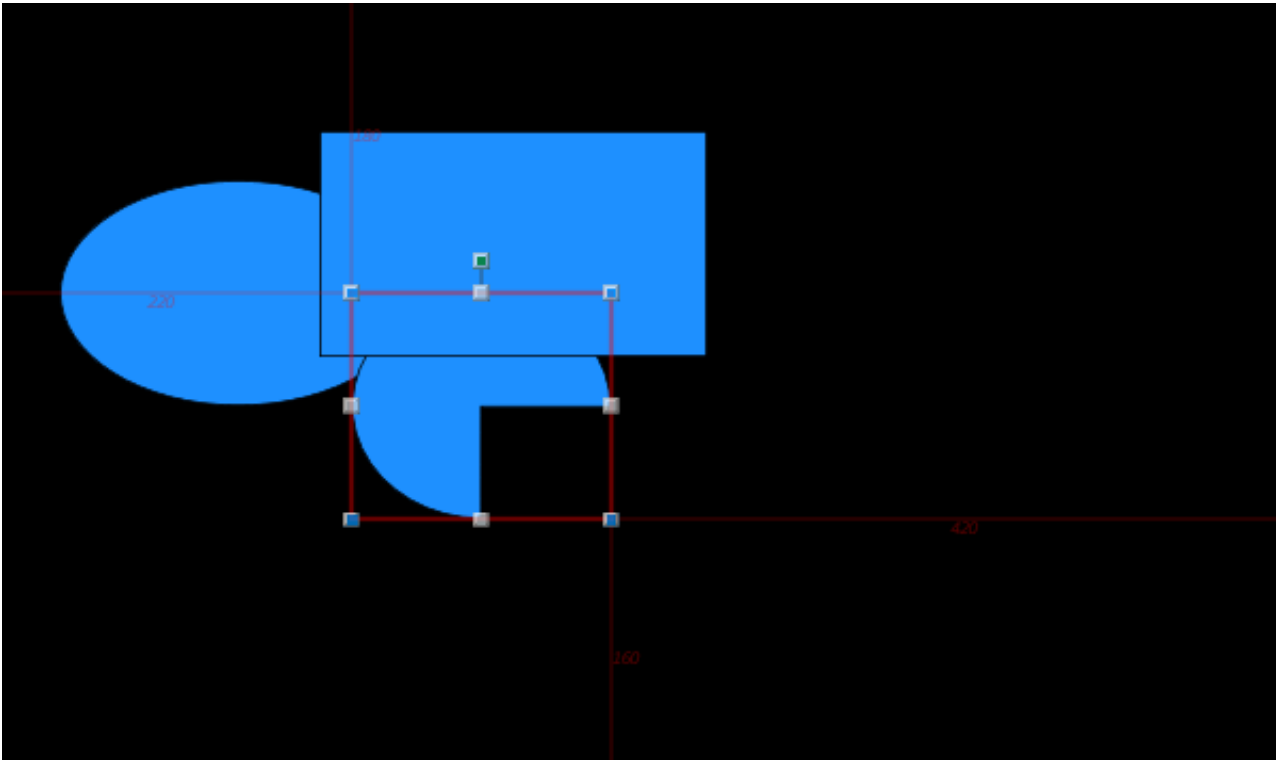
## Down

Select one of the overlapping objects (in the example, we choose the ellipse from the ellipse, rectangle and circular sector) and choose the “Down” option.



# CREW Manual

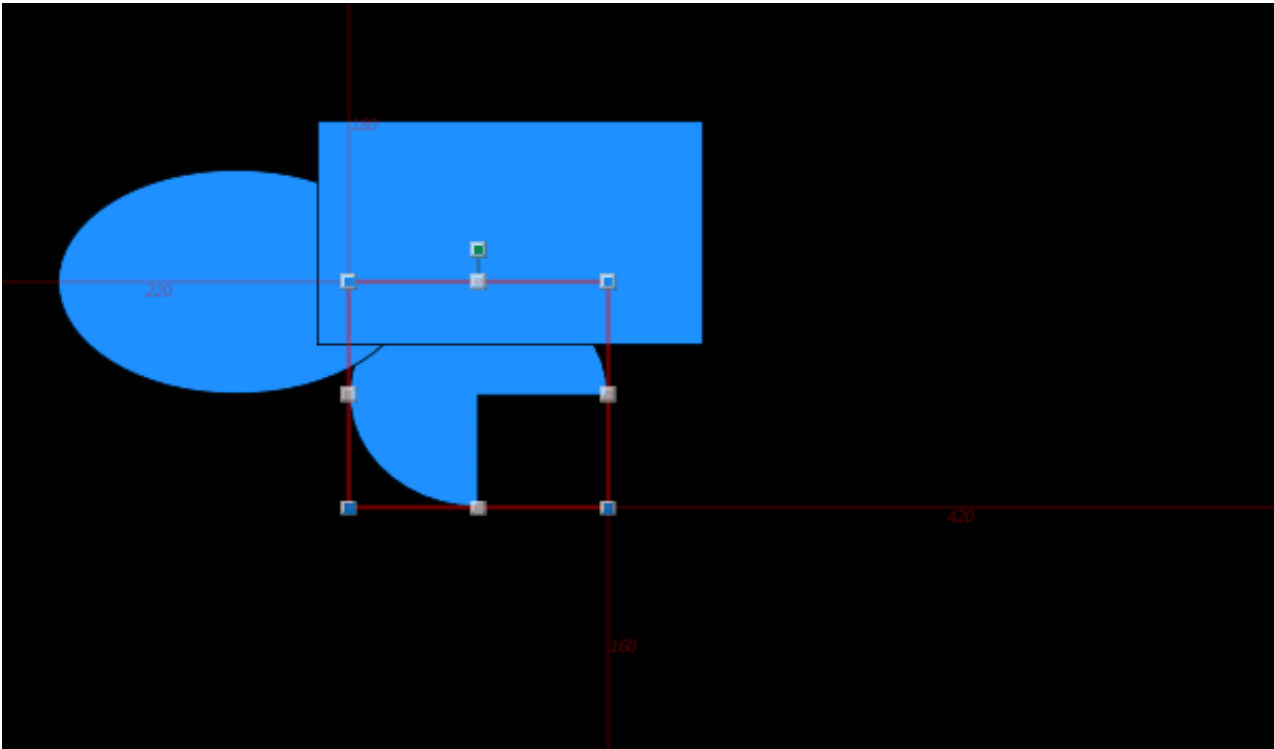
The selected object is moved back one level and is therefore covered by the object that was behind it before the command was clicked (in the example the circular sector is now under the rectangle).





# CREW Manual

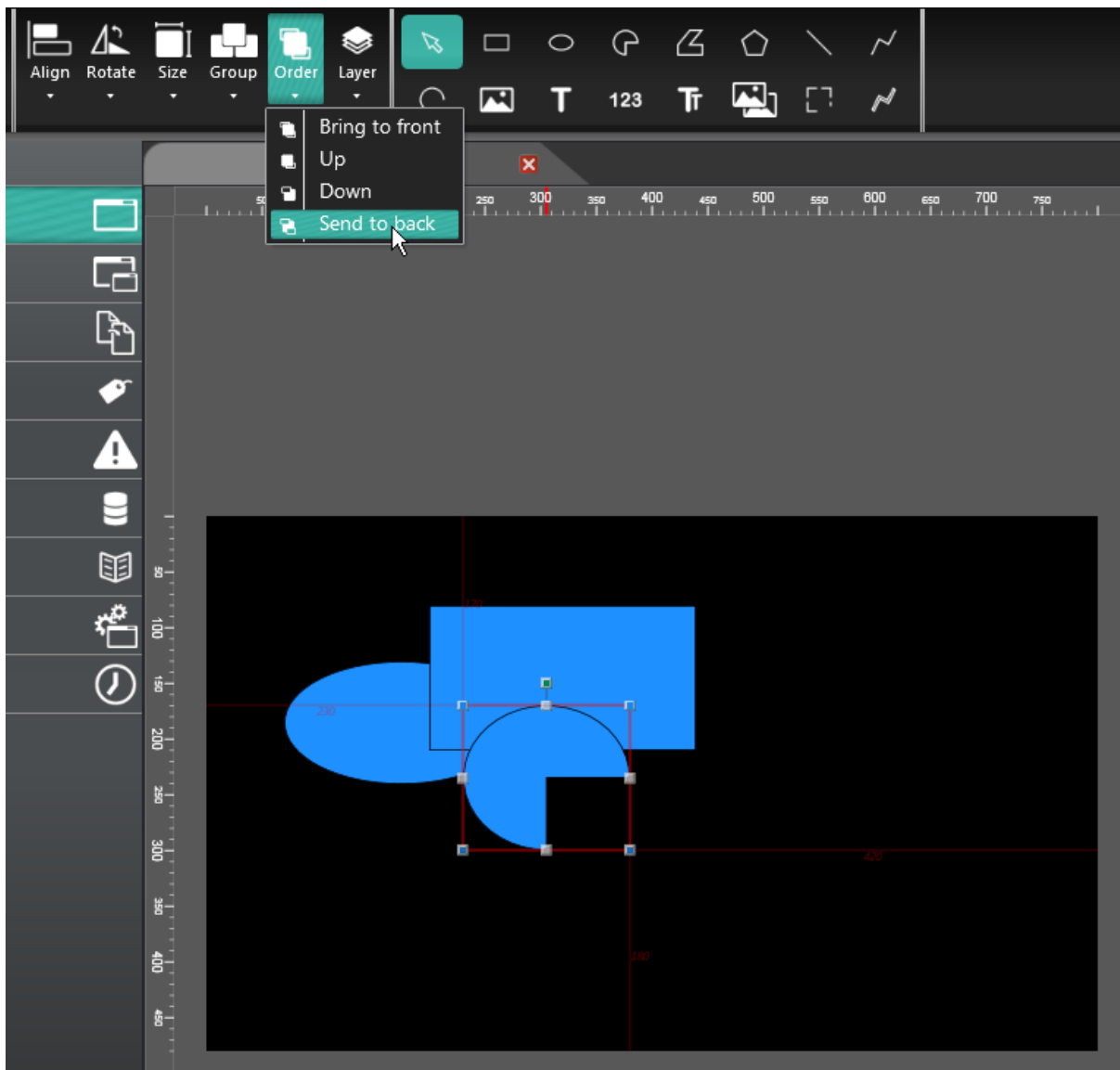
Every time this action is repeated (clicking “Down”), the selected object goes down a position in its viewing level (for example, the circular sector is now under the ellipse as well).



# CREW Manual

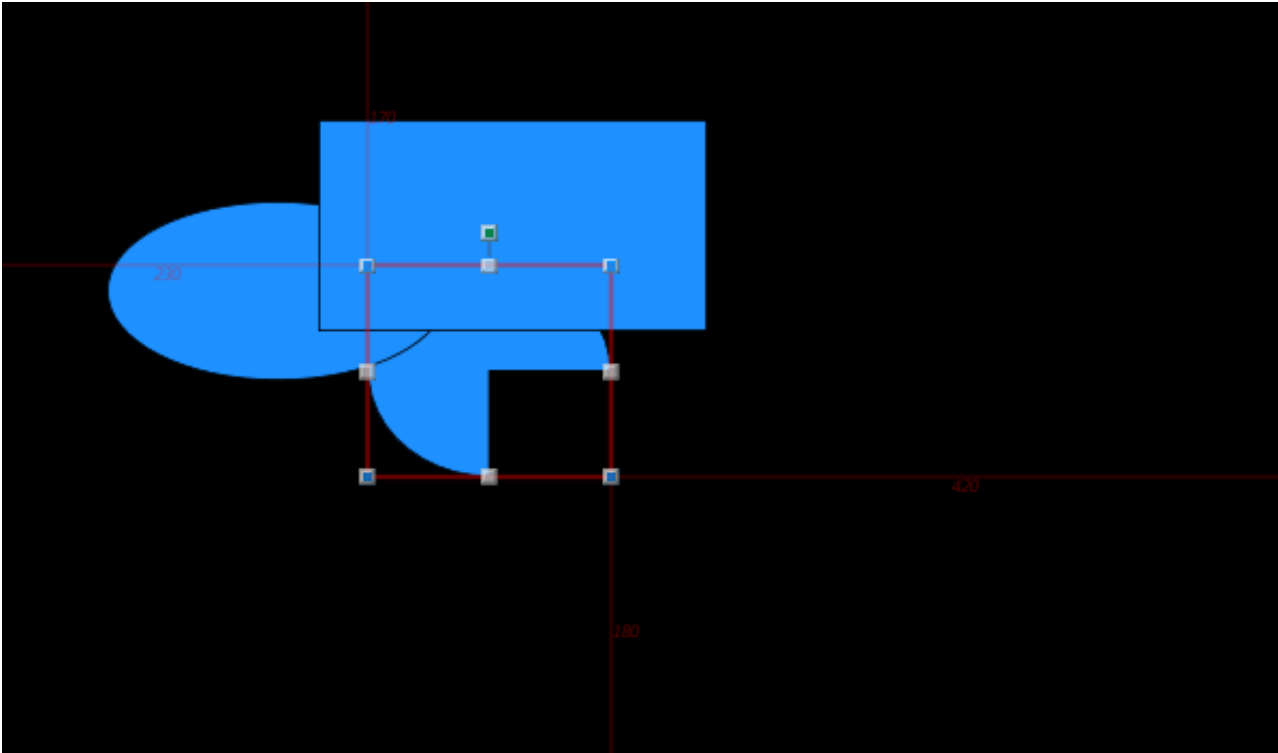
## Move back

Select one of the overlapping objects (in the example, we choose the ellipse from the ellipse, rectangle and circular sector) and choose the “Move back” option.



# CREW Manual

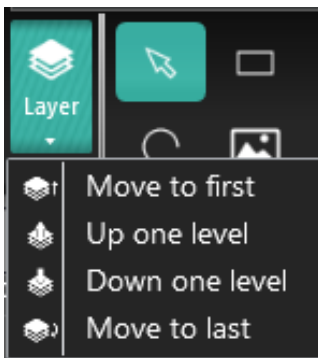
The selected object is moved to the back, i.e. behind all the other objects (in the example the circular sector is under all the objects).



# CREW Manual

## Layer Submenu

To know what operations can be carried out on the Layers, refer to the "[Layers](#)" section belonging to "Property Editor".



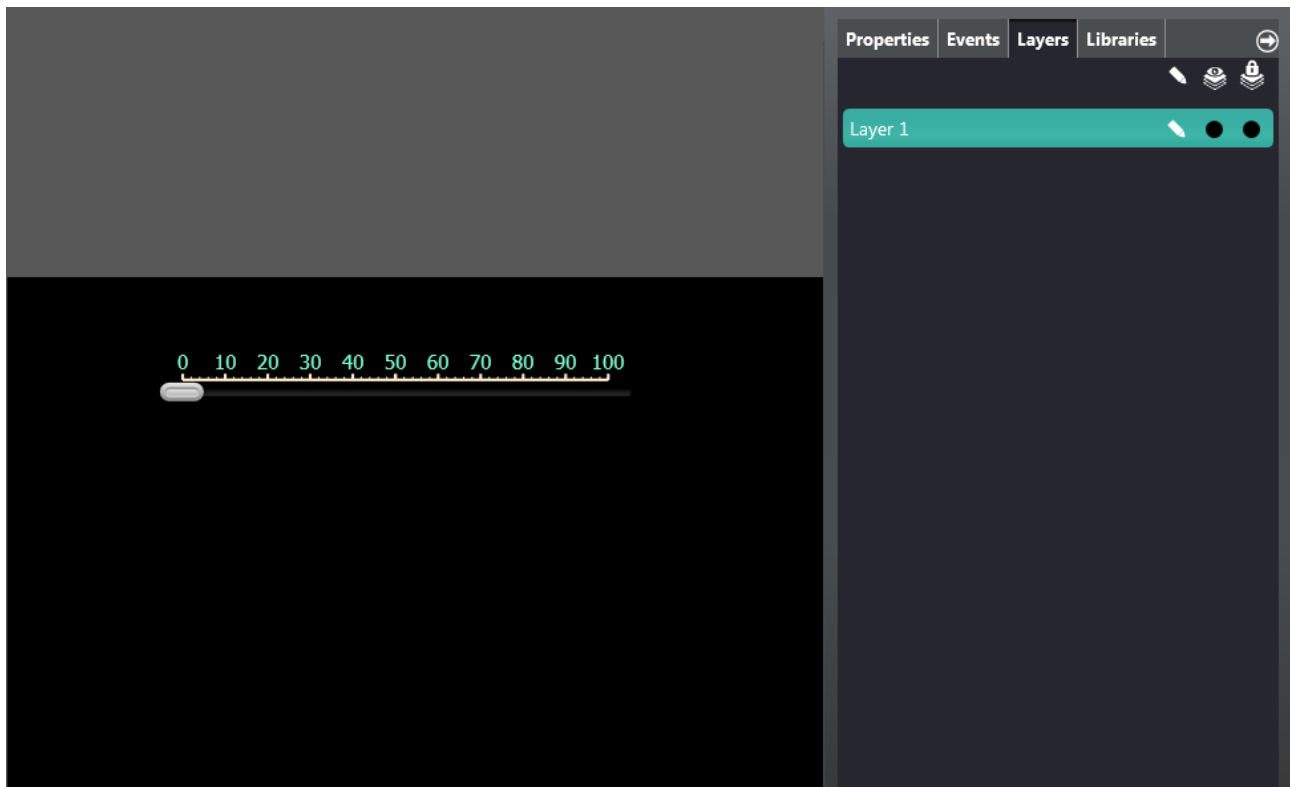
From the "Layer" option it is possible to perform the following actions:

- Move to the beginning: move the selected object/s to the first layer.
- Up a level: move the selected object/s up a layer.
- Down a level: move the selected object/s down a layer.
- Move to the end: move the selected object/s to the last layer.

# CREW Manual

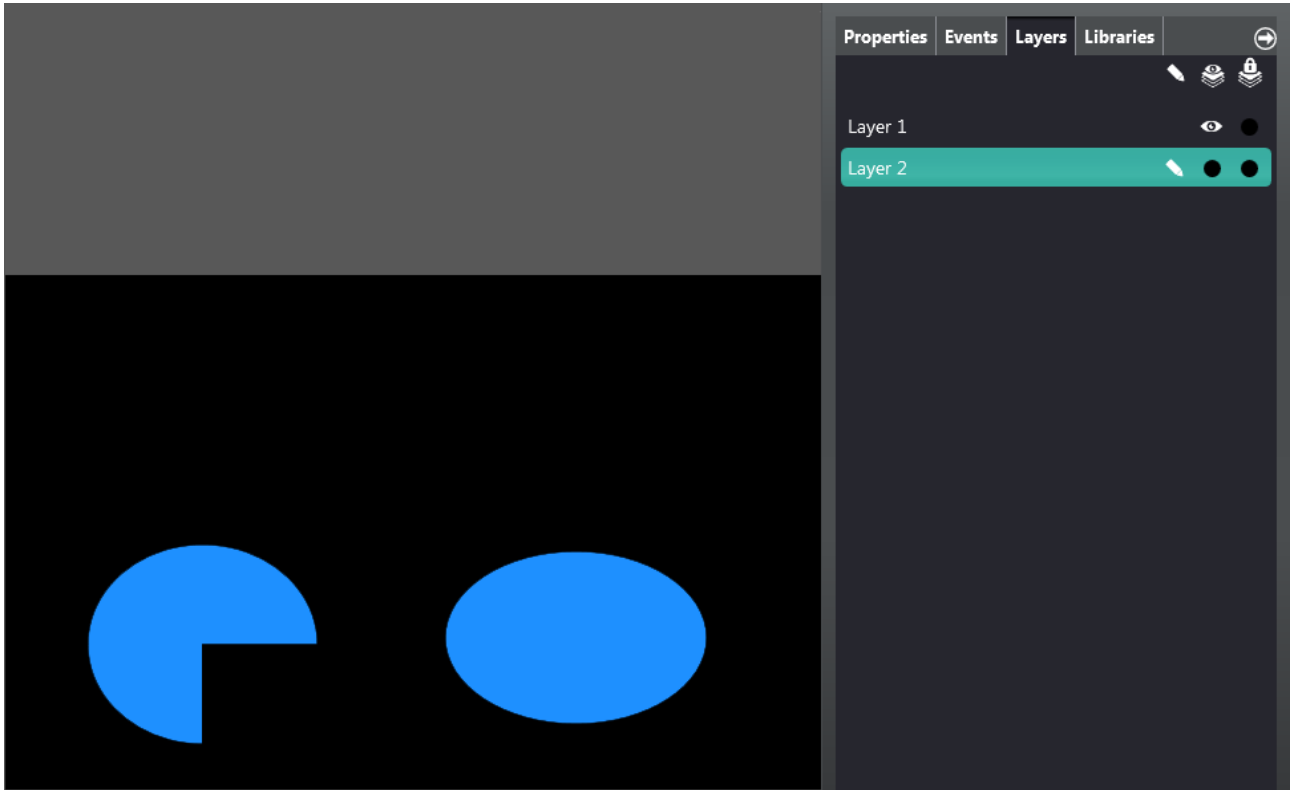
Move to first

Enter a selector in layer 1.



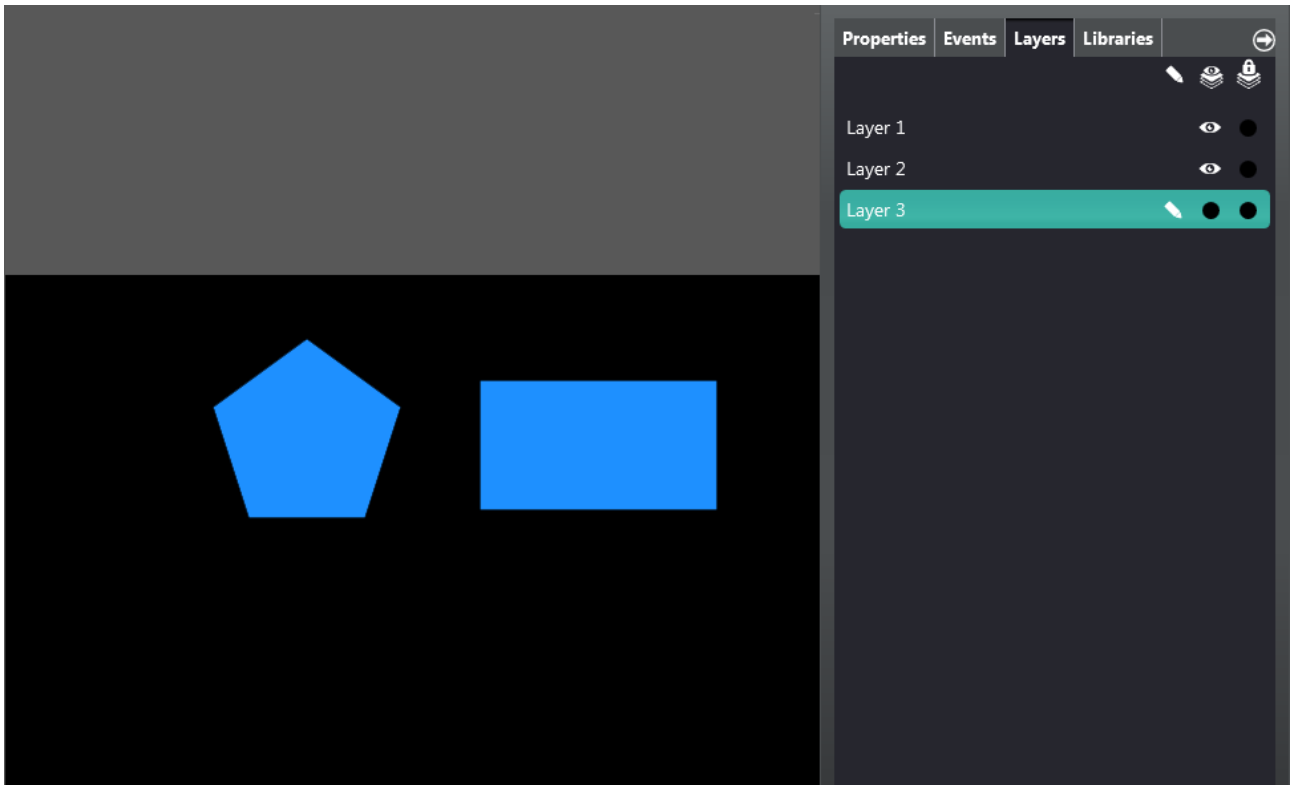
# CREW Manual

Enter a circular sector and an ellipse in layer 2.



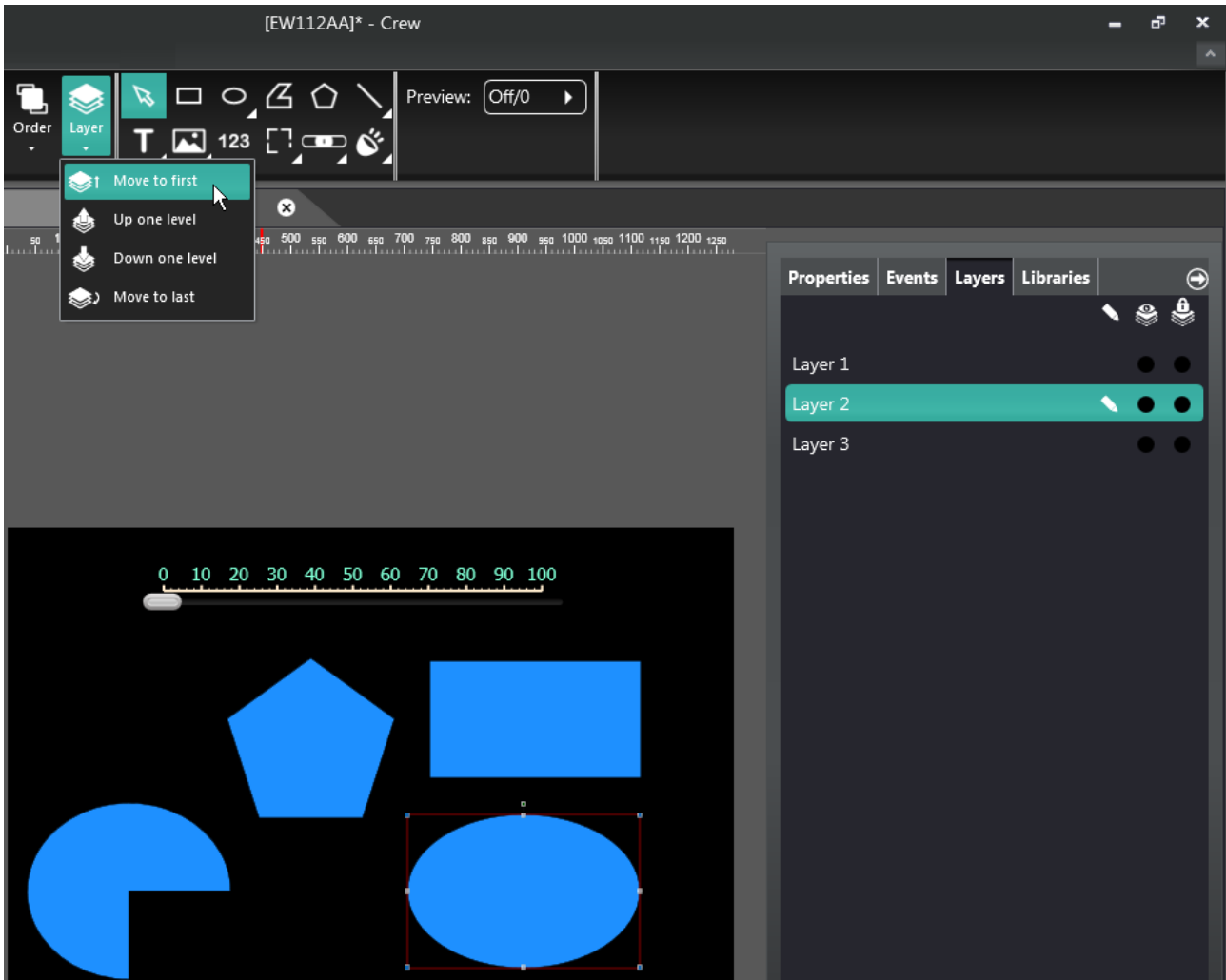
# CREW Manual

Enter a regular polygon and a rectangle in layer 3.



# CREW Manual

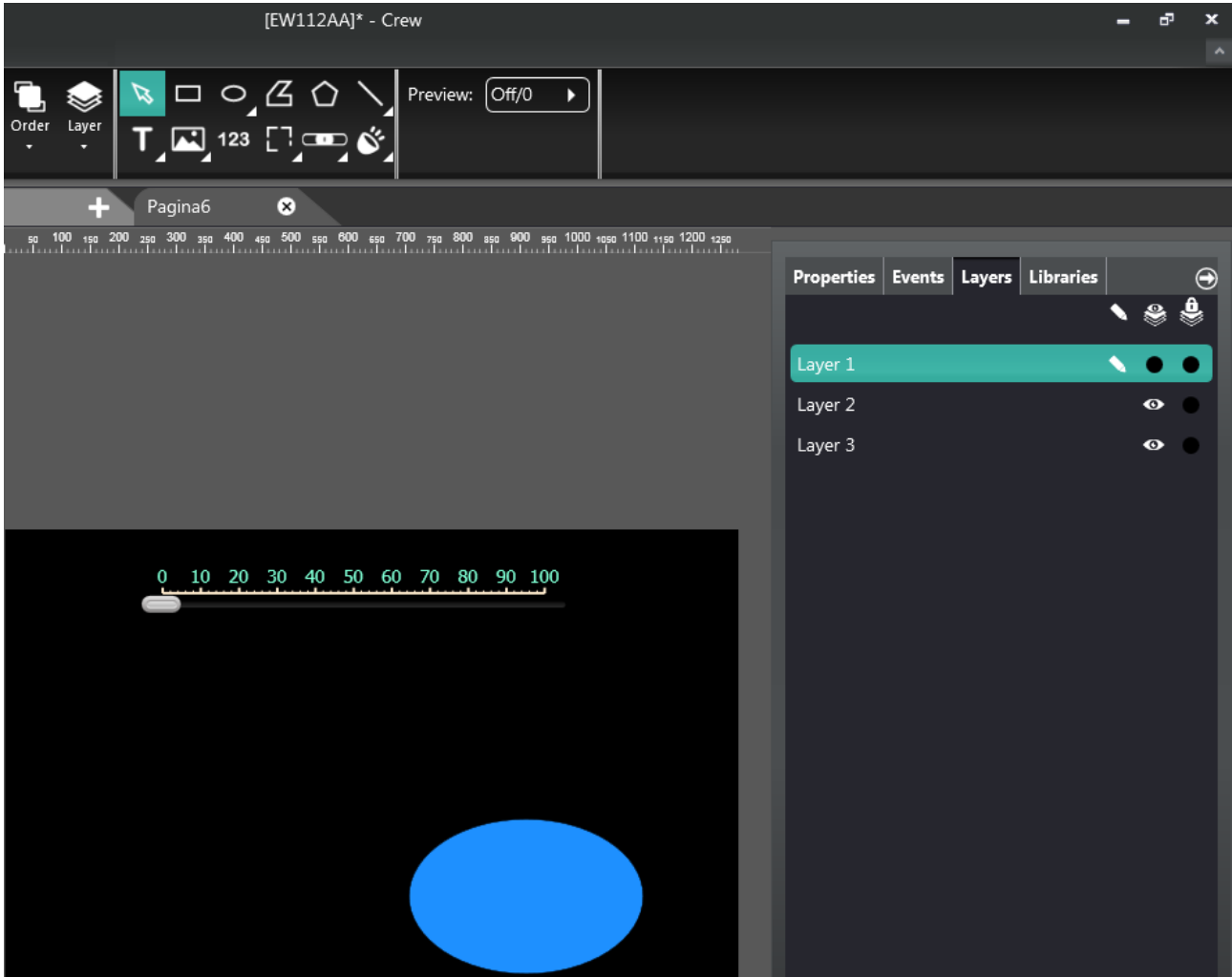
To move the ellipse from layer 2 to layer 1, select the object and click “Move to first” in the “Layer” submenu.





# CREW Manual

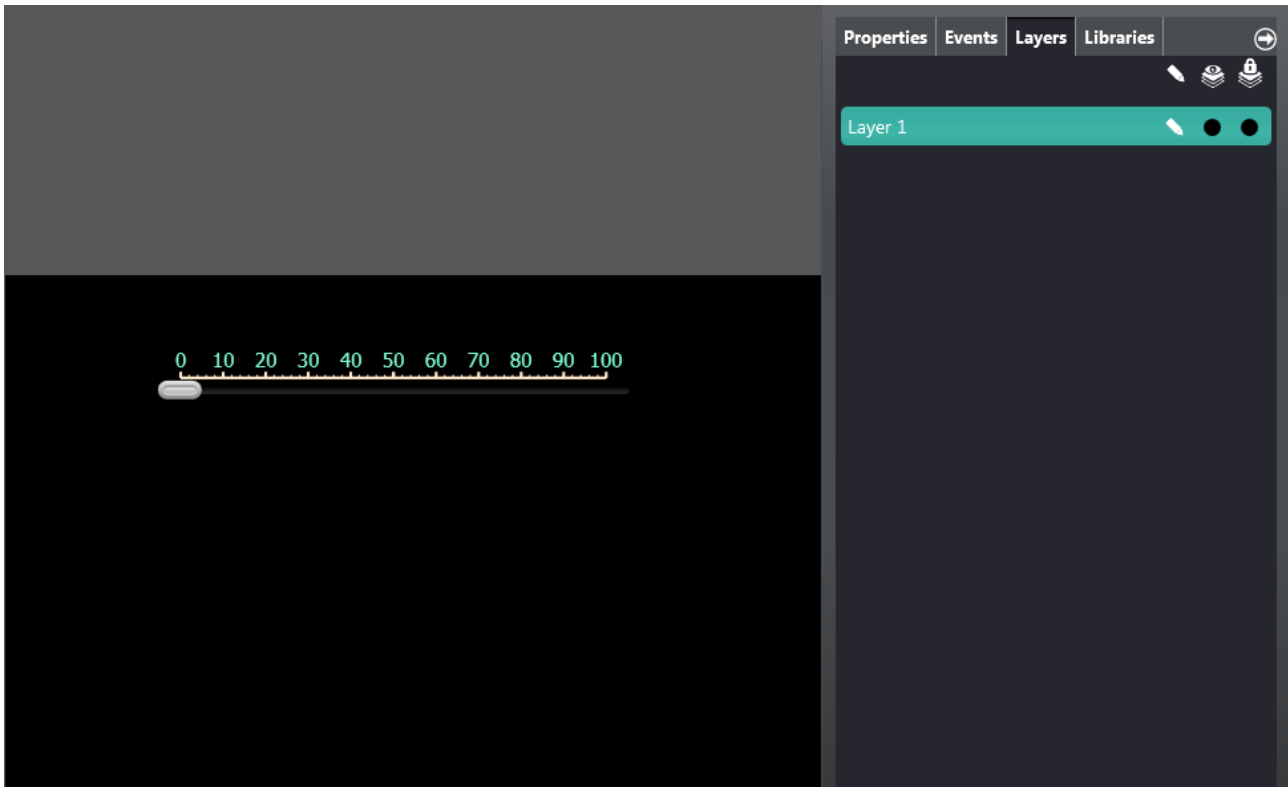
As you can see in the example, the ellipse is also in layer 1 now.



# CREW Manual

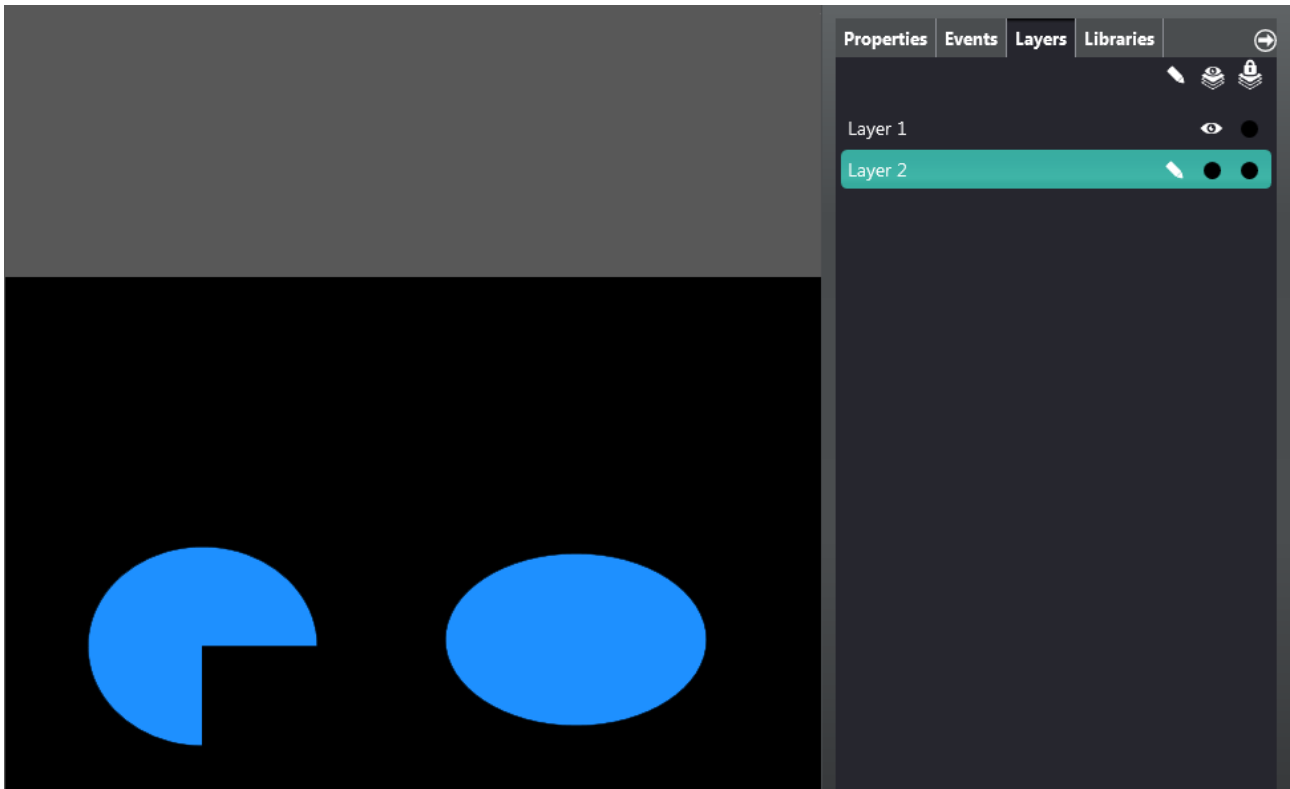
Up one level

Enter a selector in Layer 1.



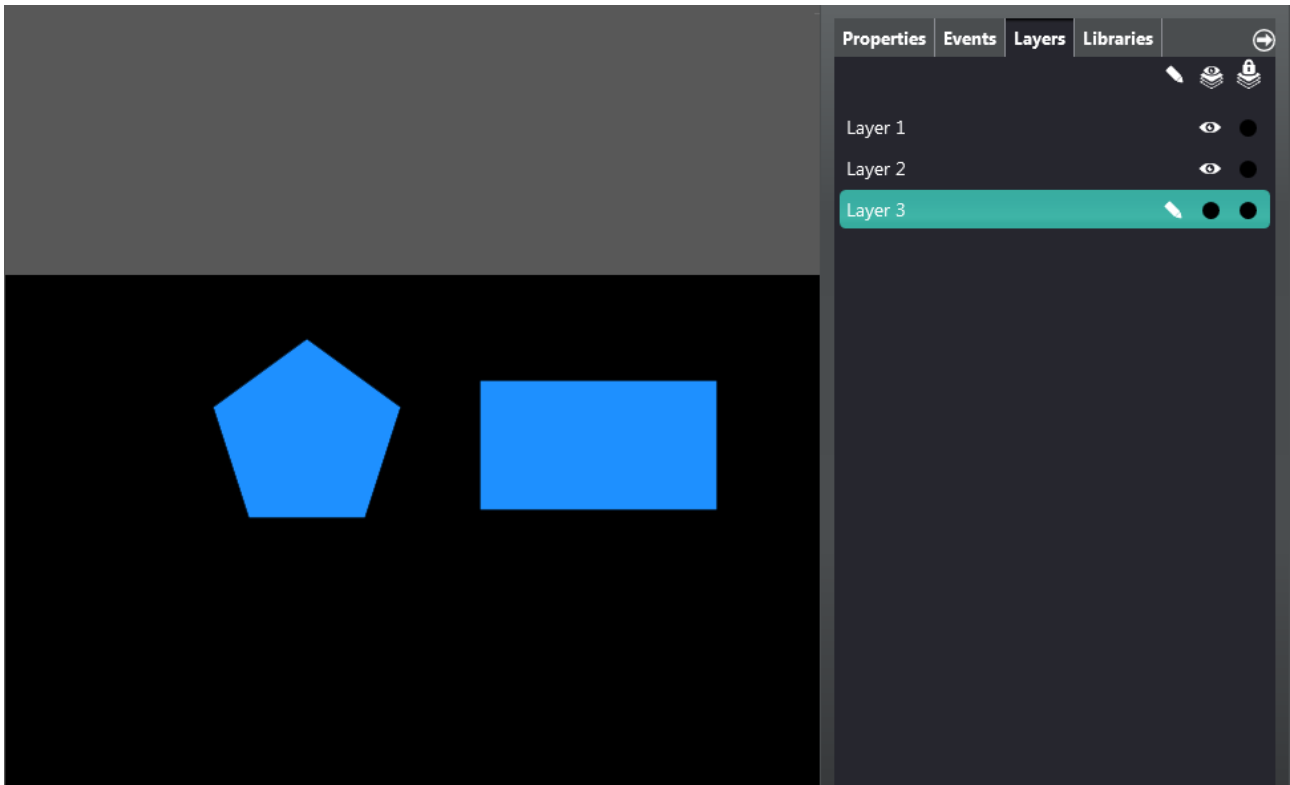
# CREW Manual

Enter a circular selector and an ellipse in Layer 2.



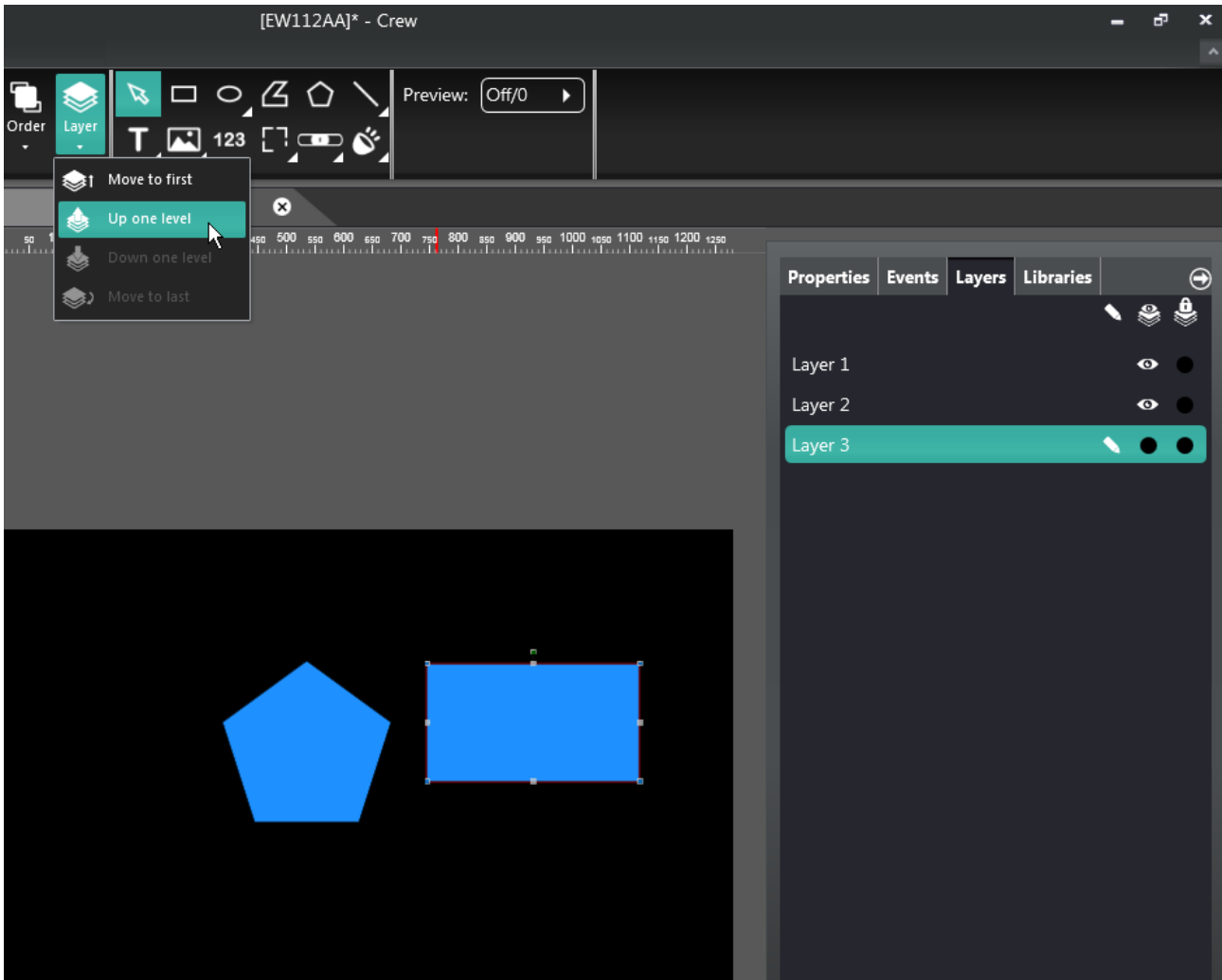
# CREW Manual

Enter a regular polygon and a rectangle in Layer 3.



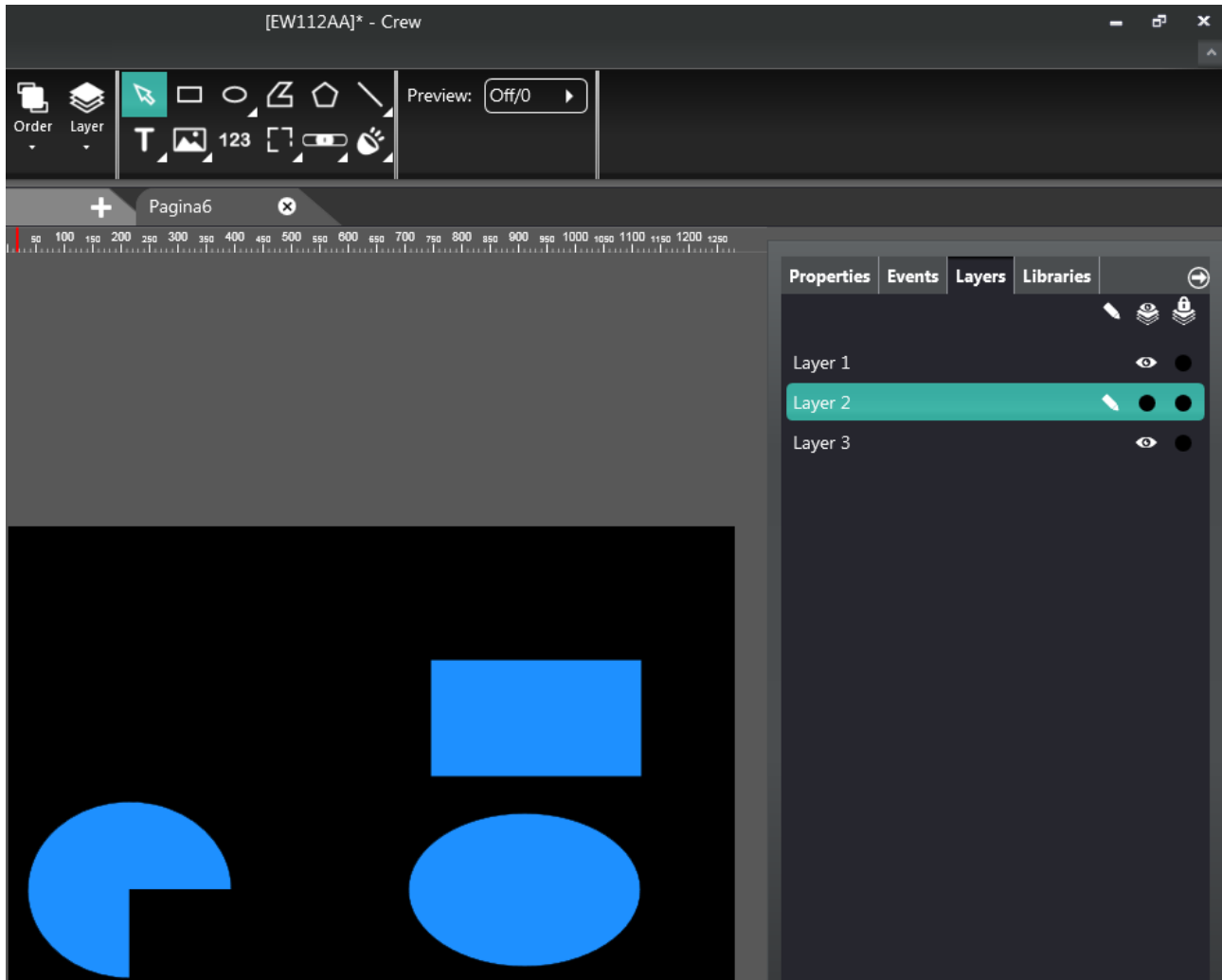
# CREW Manual

To move the rectangle up one level (in this case from layer 3 to layer 2), select the object and click “Up one level” in the “Layer” submenu.



# CREW Manual

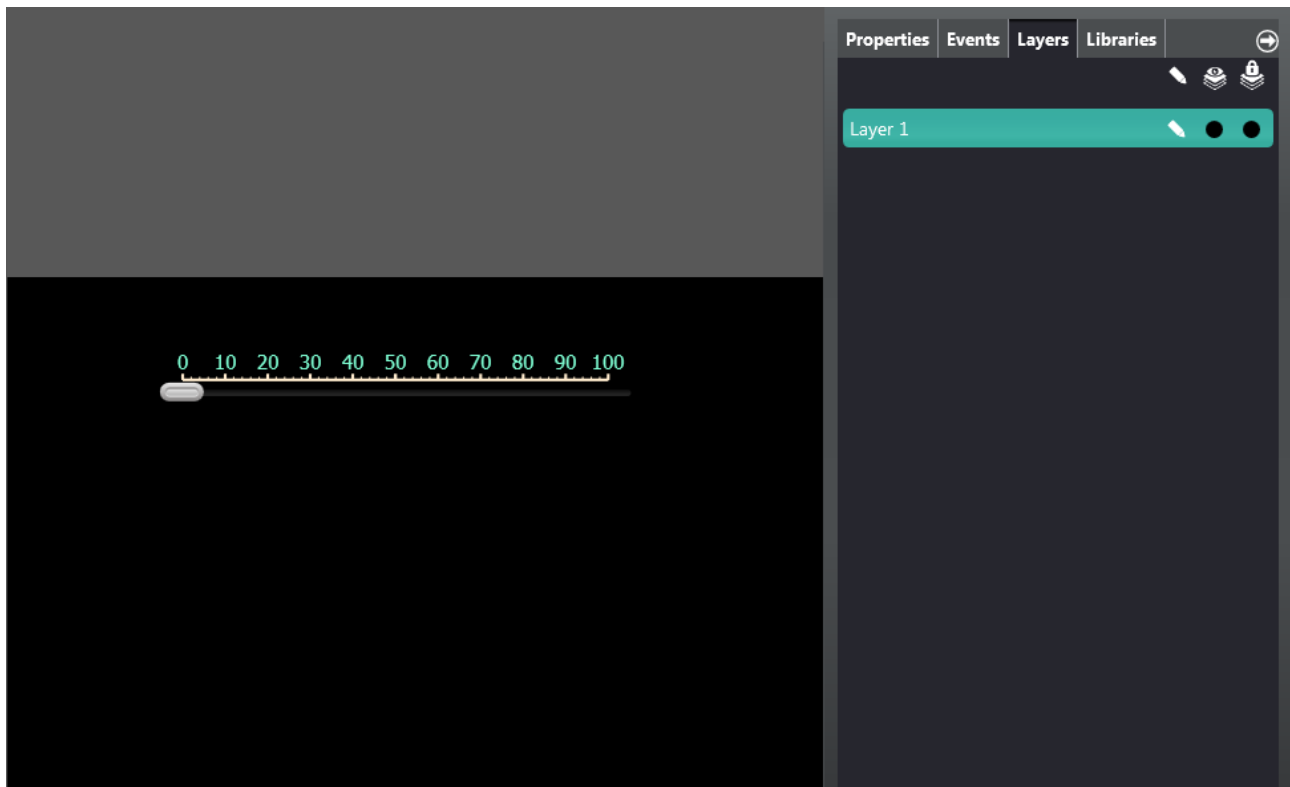
Now the rectangle is also in layer 2.



# CREW Manual

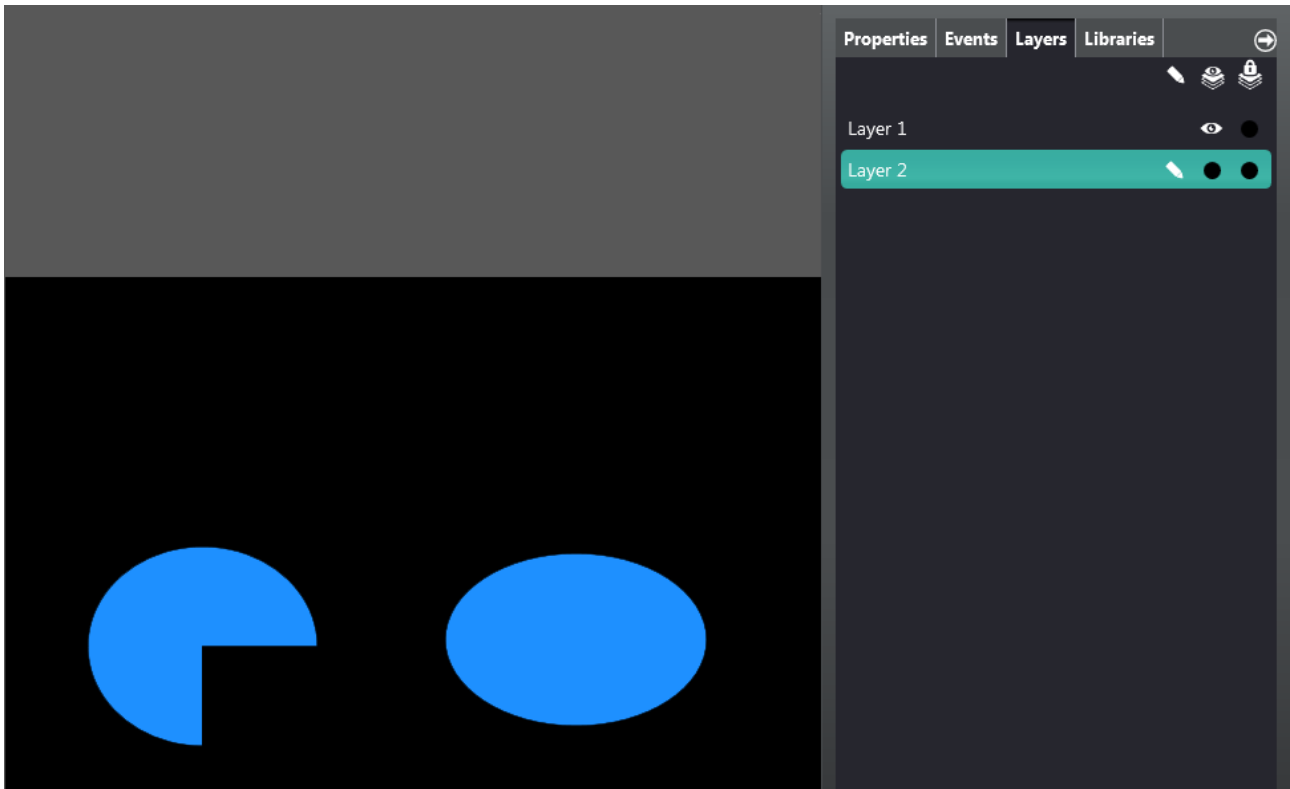
Down one level

Enter a selector in Layer 1.



# CREW Manual

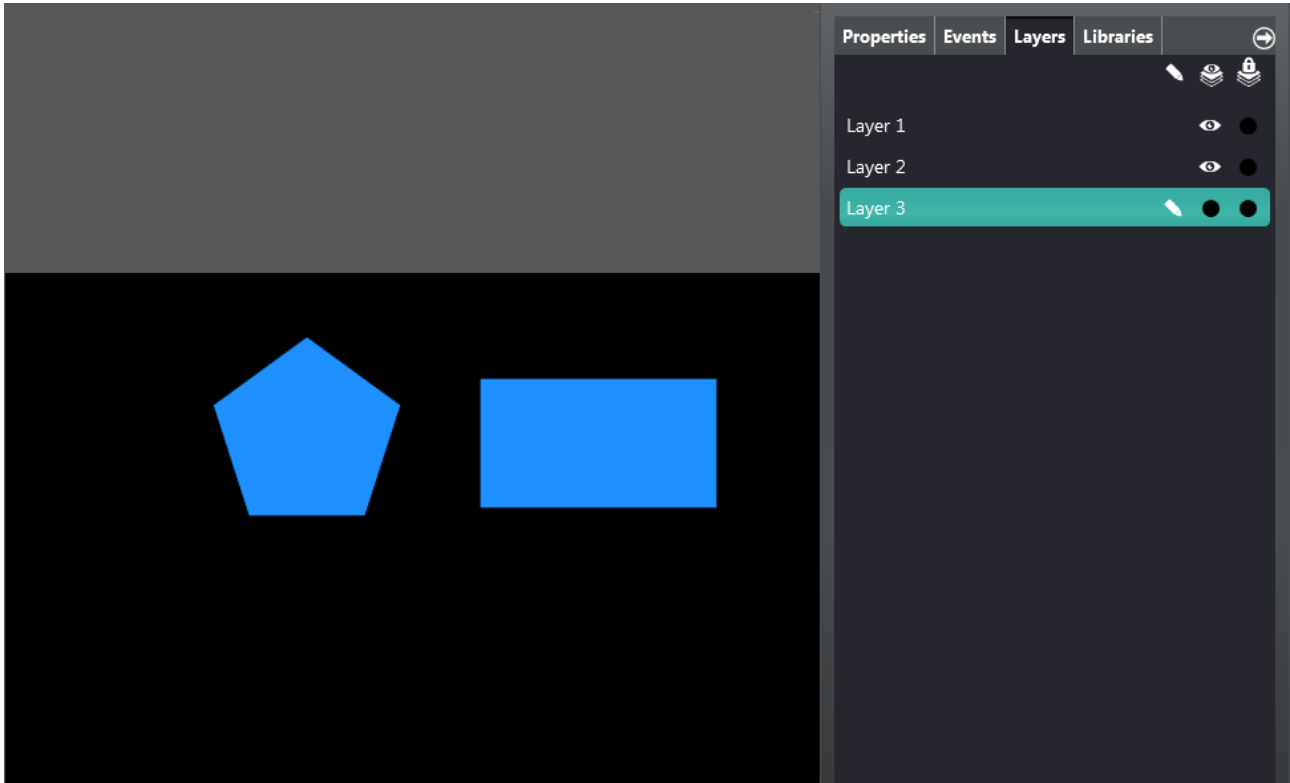
Enter a circular selector and an ellipse in Layer 2.





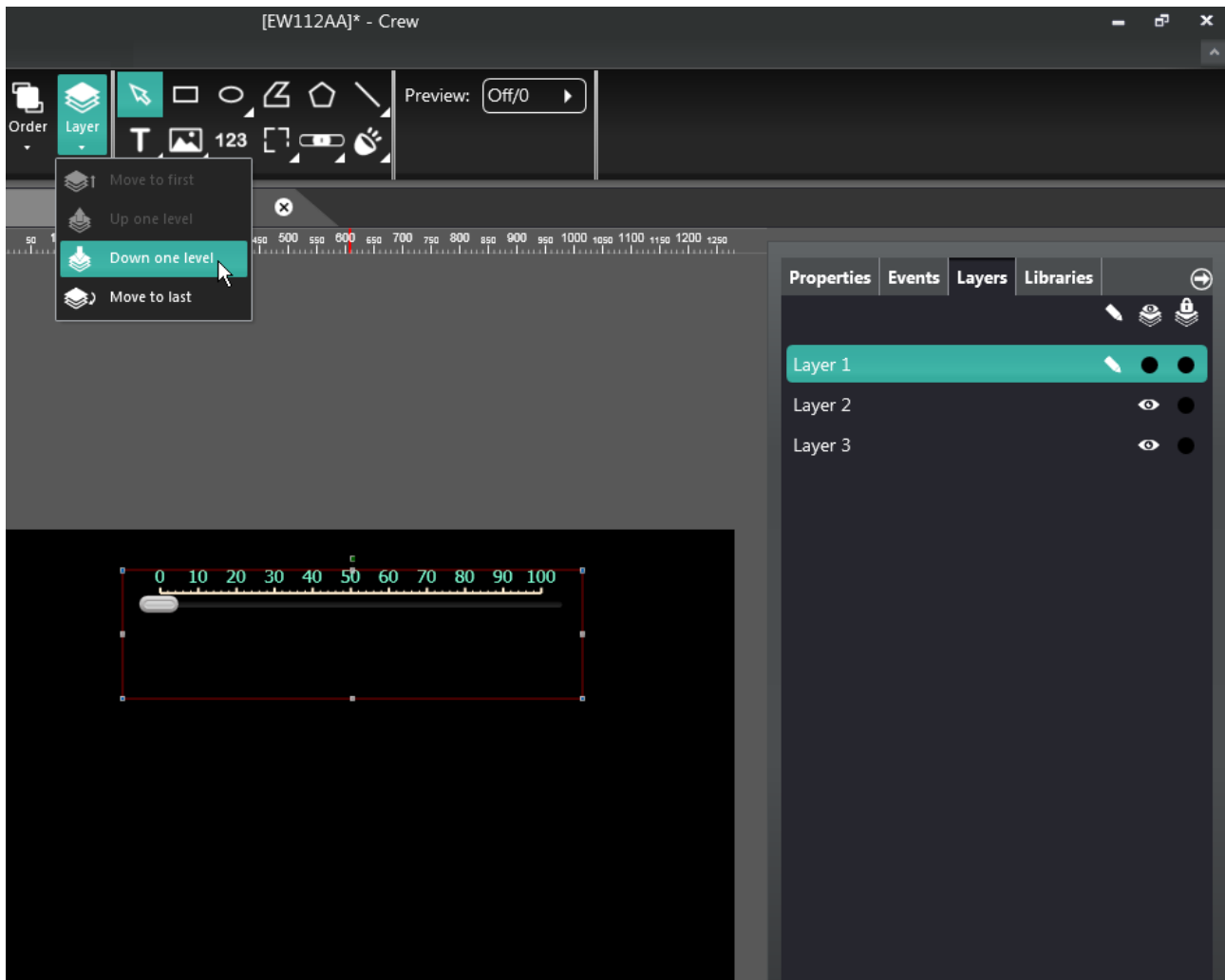
# CREW Manual

Enter a regular polygon and a rectangle in Layer 3.



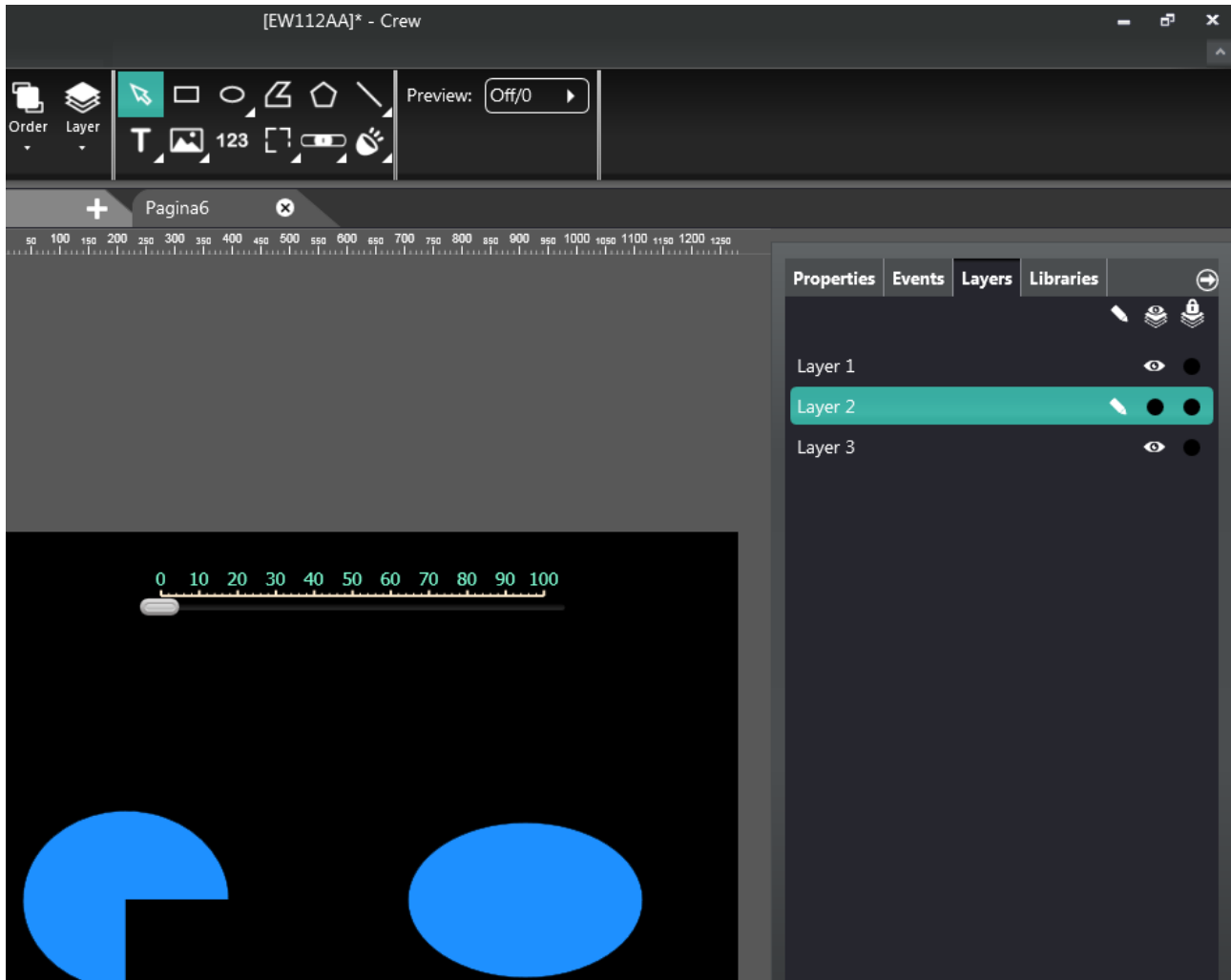
# CREW Manual

To move the selector down one level (in this case from layer 1 to layer 2), select the object and click “Down one level” in the “Layer” submenu.



# CREW Manual

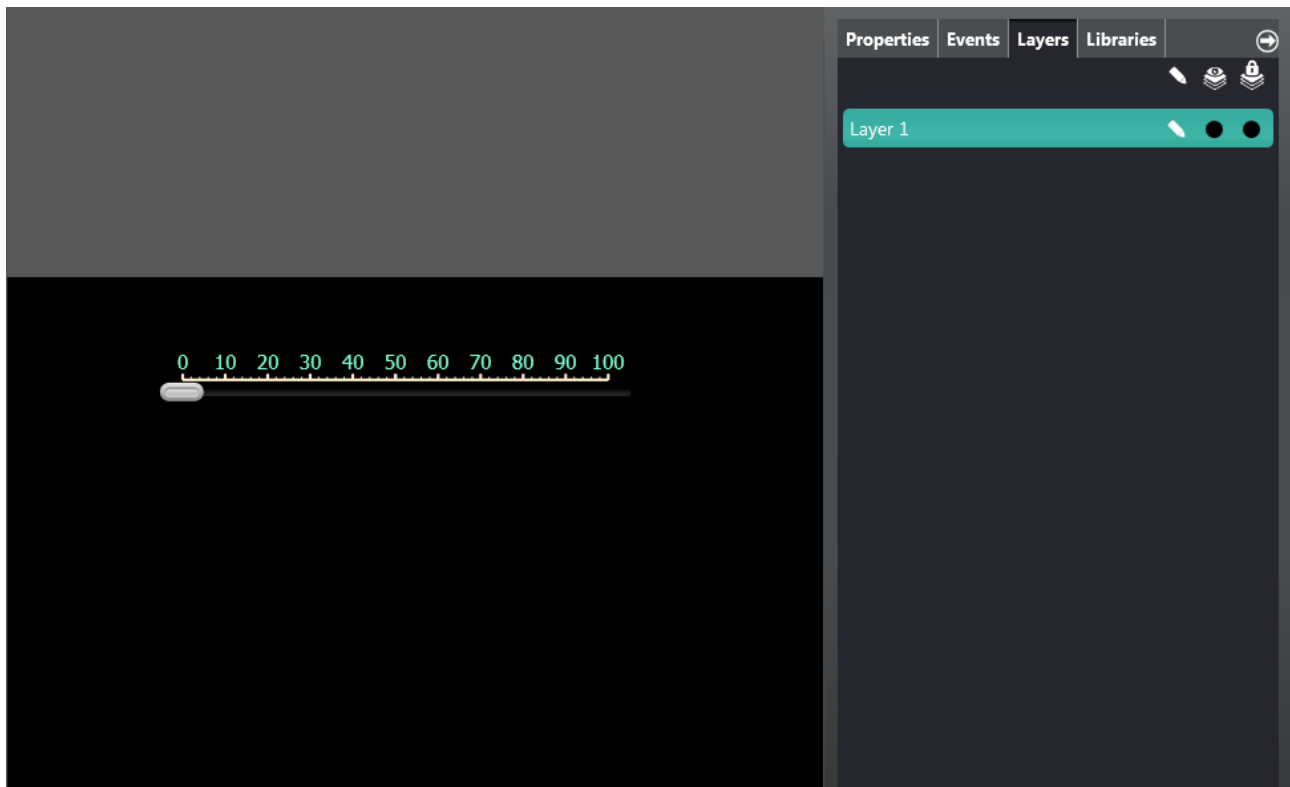
Now the selector is also in layer 2.



# CREW Manual

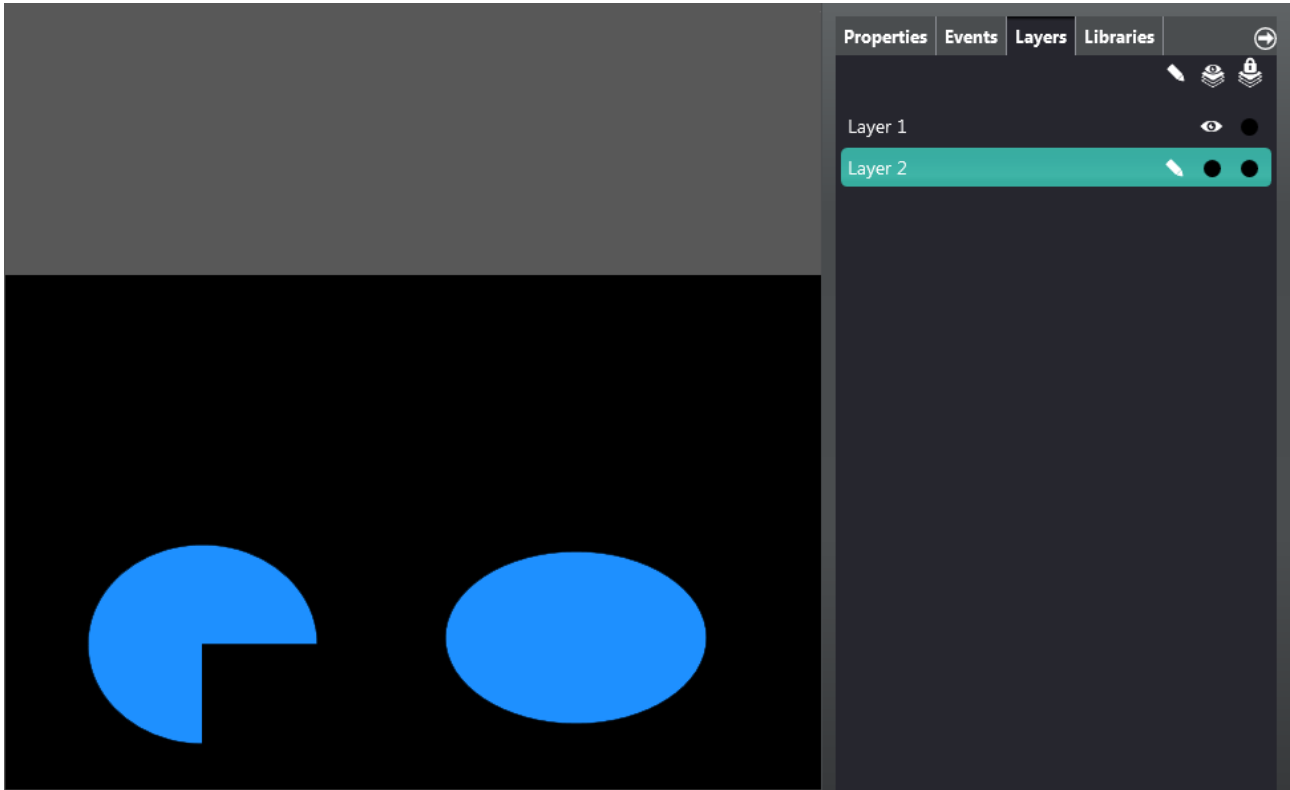
Move to last

Enter a selector in Layer 1.



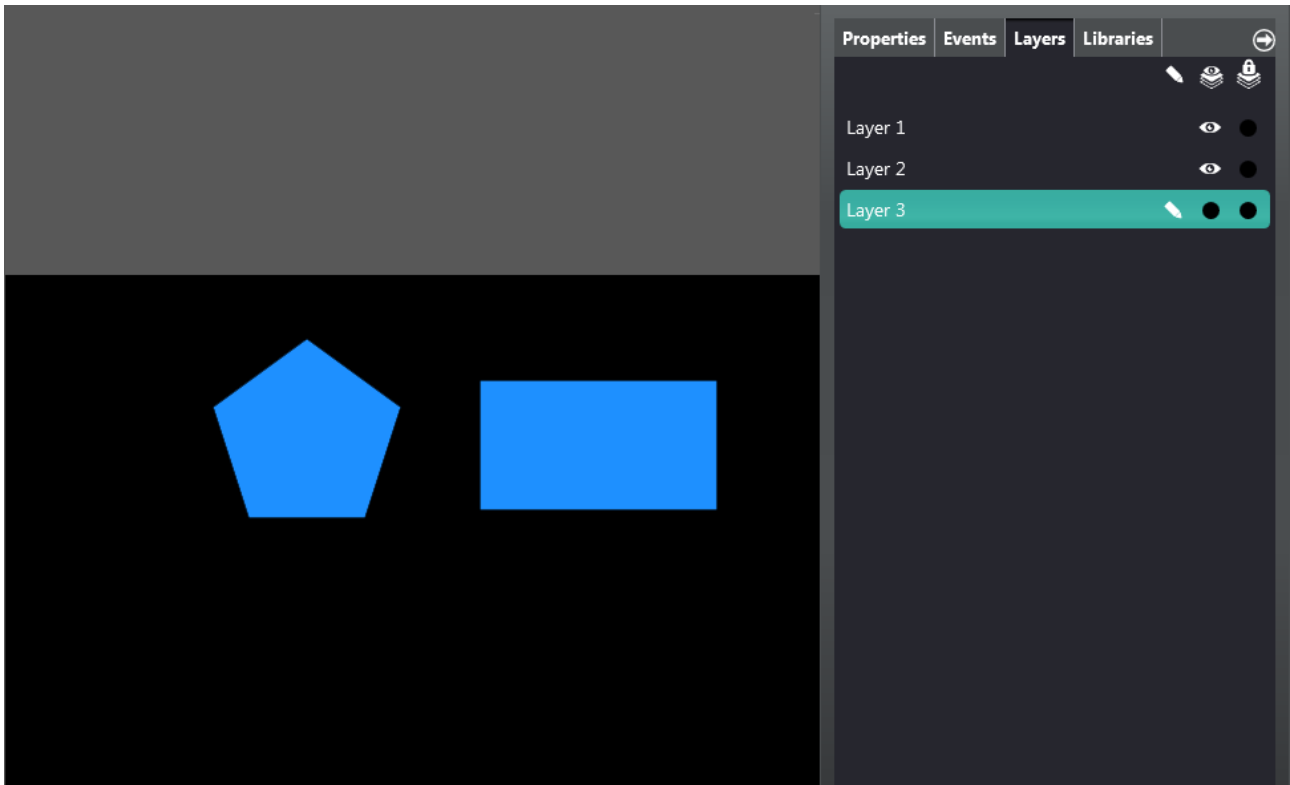
# CREW Manual

Enter a circular selector and an ellipse in Layer 2.



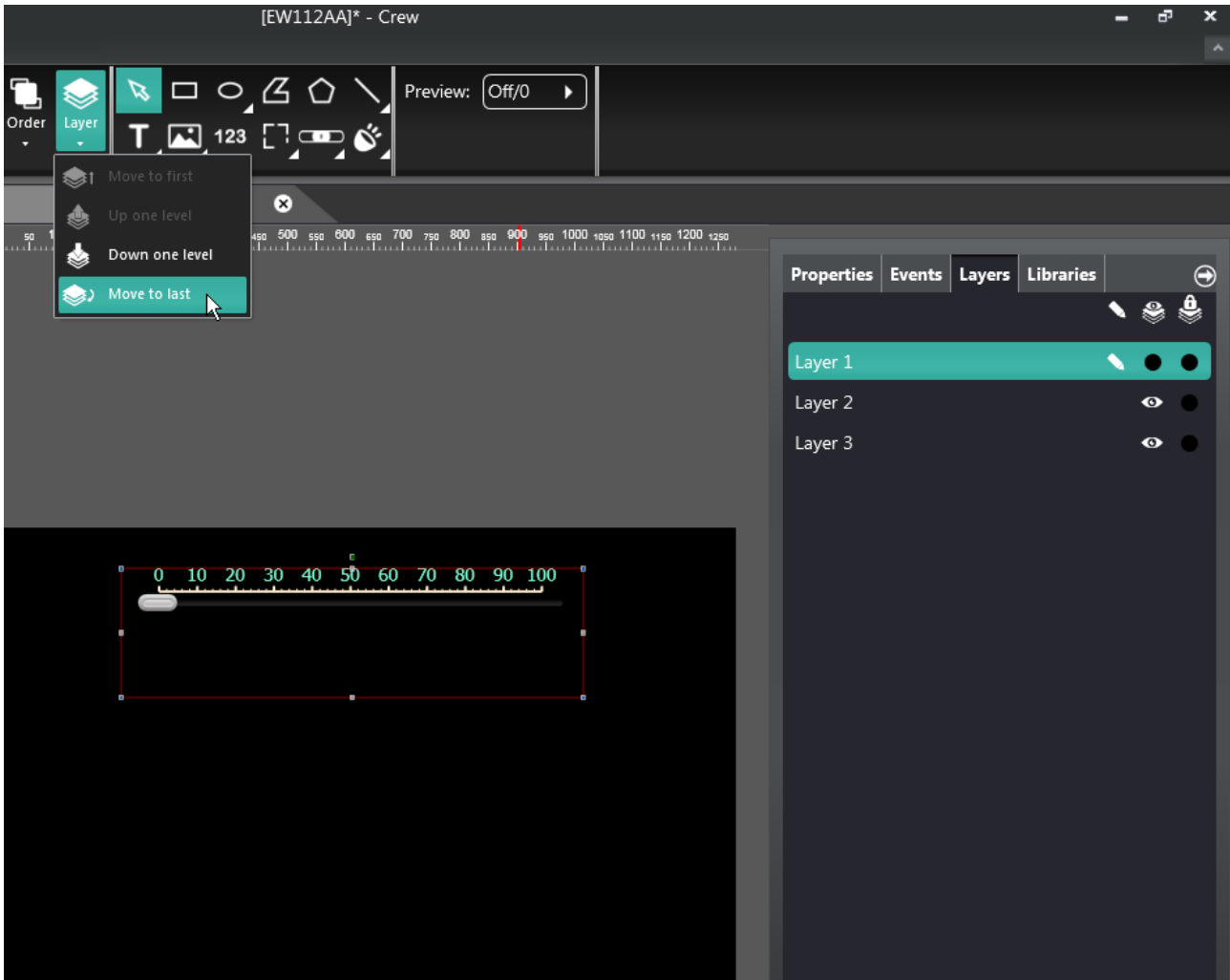
# CREW Manual

Enter a regular polygon and a rectangle in Layer 3.



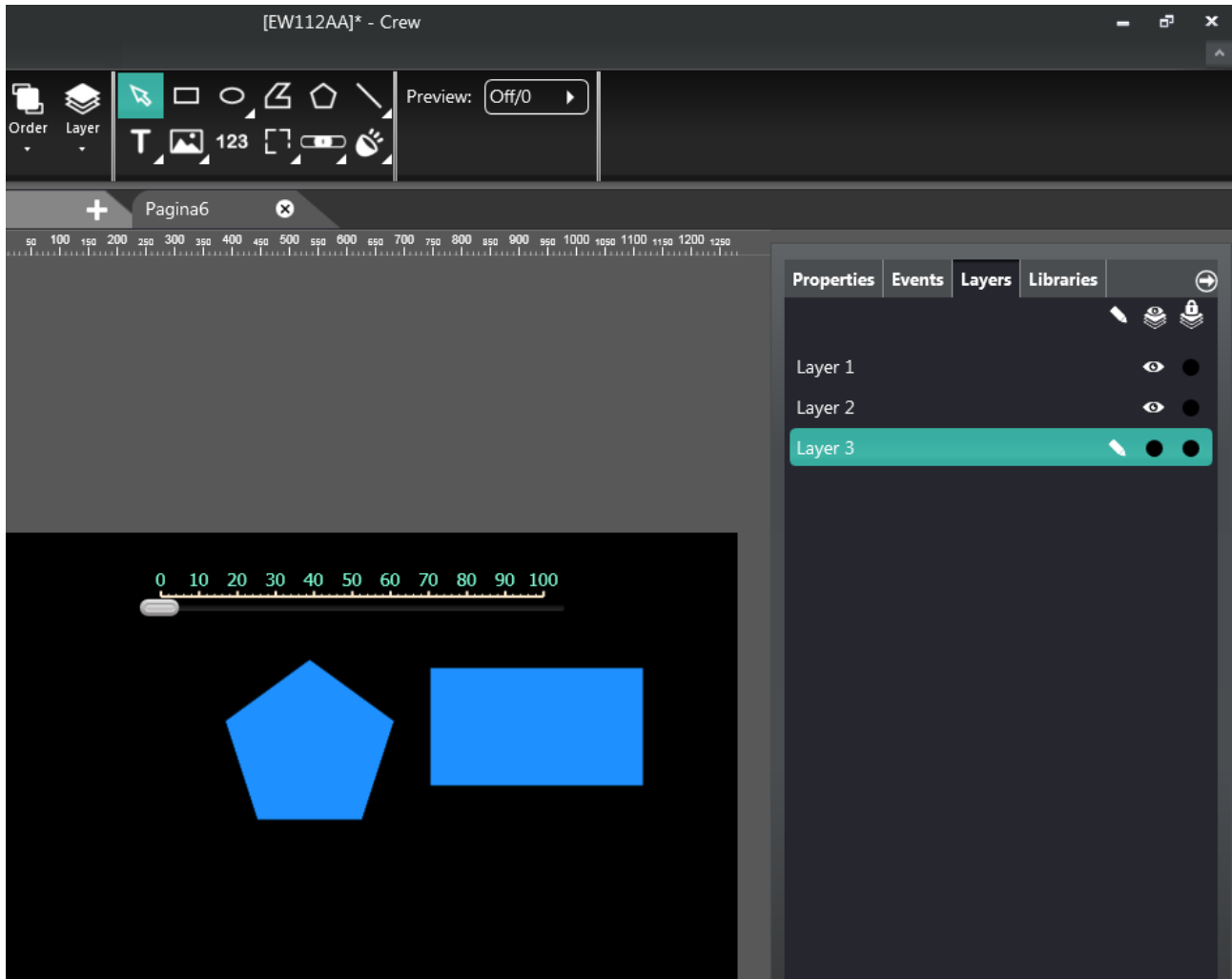
# CREW Manual

To move the selector to the last level (in this case to layer 3), select the object and click “Move to last” in the “Layer” submenu.



# CREW Manual

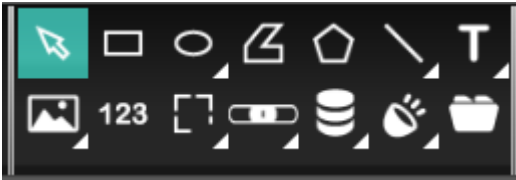
Now the selector is also in layer 3.



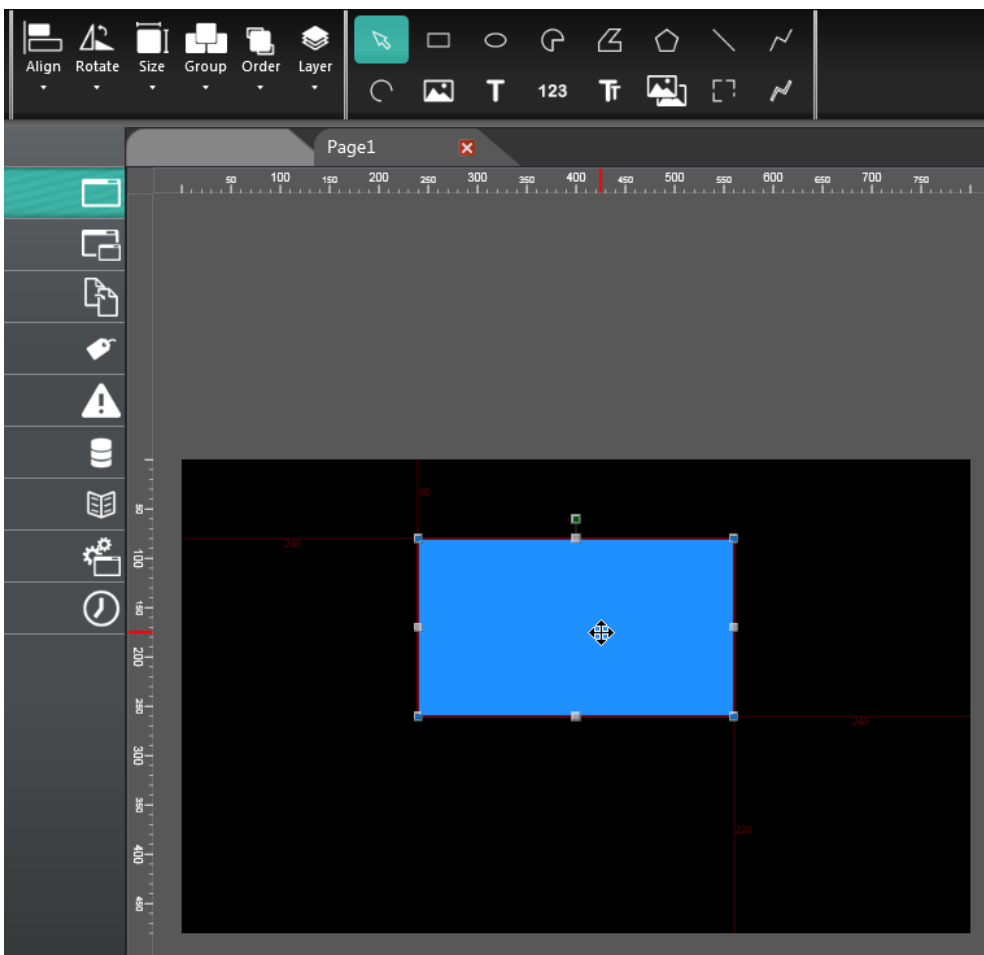


# CREW Manual

## Selection tool

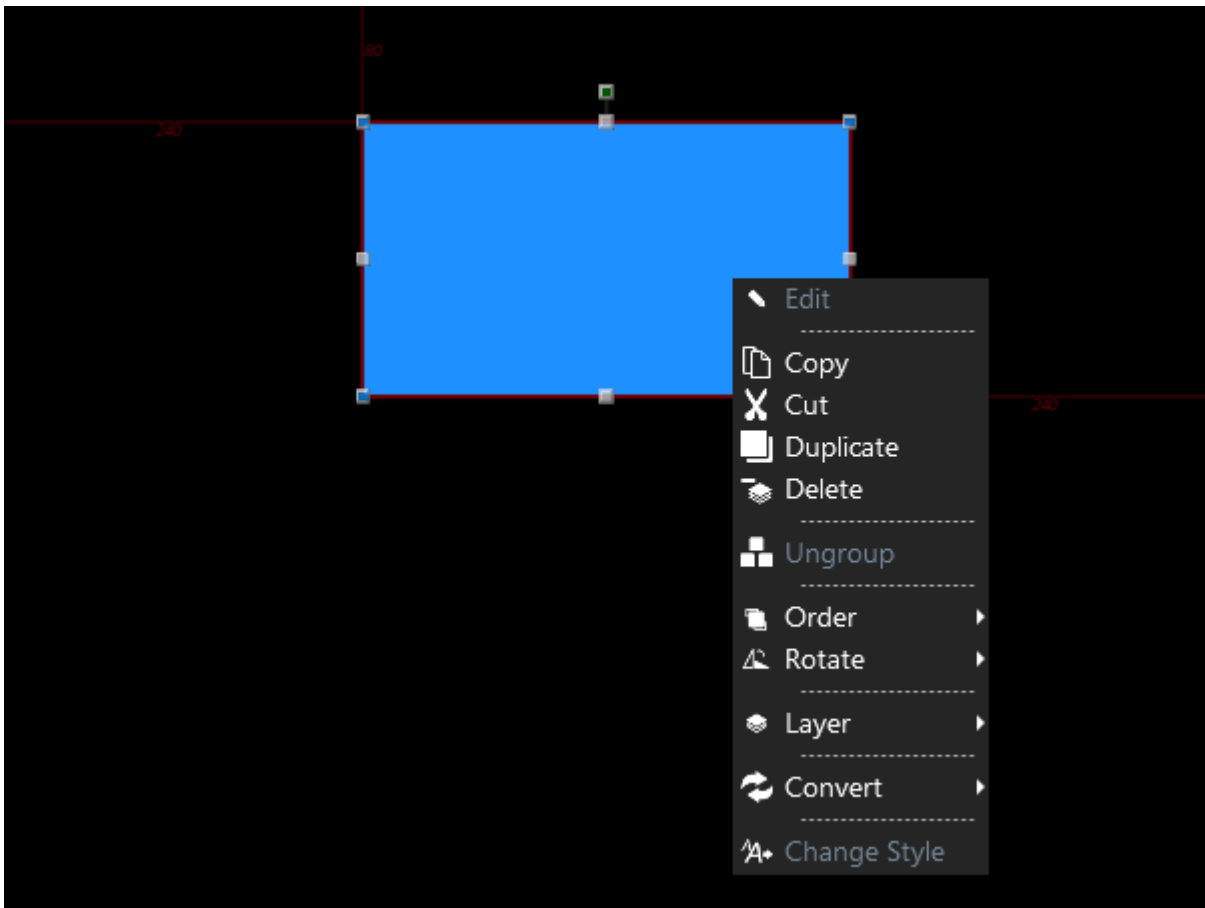


The “Selection” tool is used to select each object on a page, so that it can be modified. Place an object on the page (a rectangle for example), click on the “selection” tool and use the mouse to select the object.



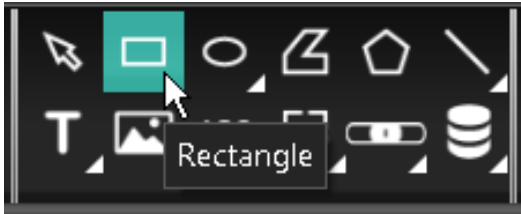
# CREW Manual

It is now possible to edit the properties of the object and right click on the mouse to access a menu with the following functions: “Copy”, “Cut”, “Duplicate”, “Delete”, “Order” (see “Order Submenu”), “Rotate”(see “Rotate Submenu”), “Layer” (see “Layer Submenu”), and “Conversion

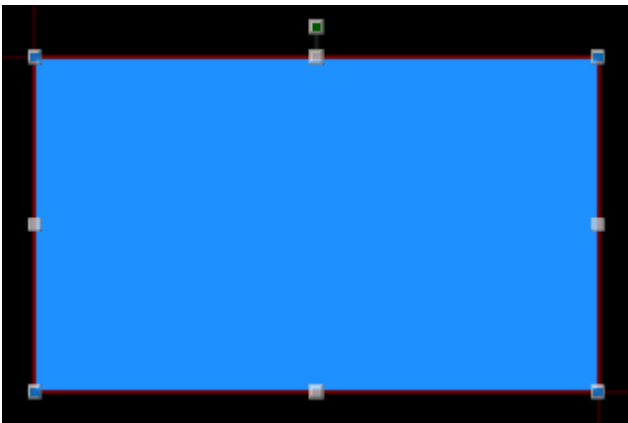


# CREW Manual

## Rectangle

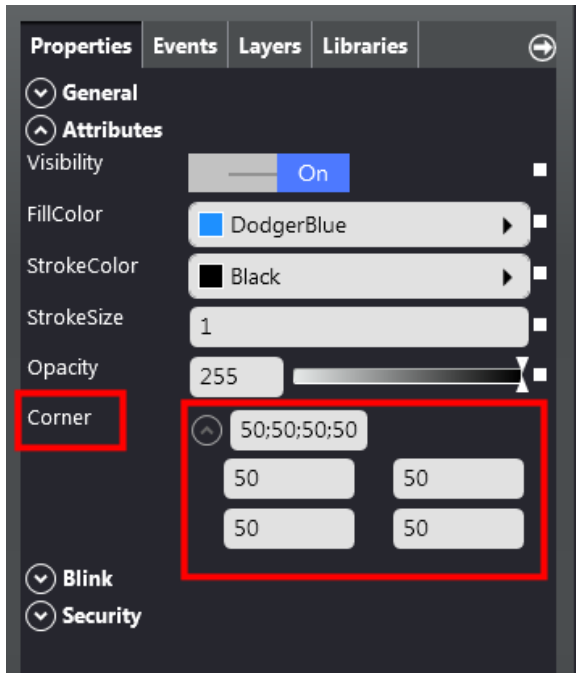


The “Rectangle” icon on the “Graphics” menu is used to place a rectangle on the page, and change its size on that page.

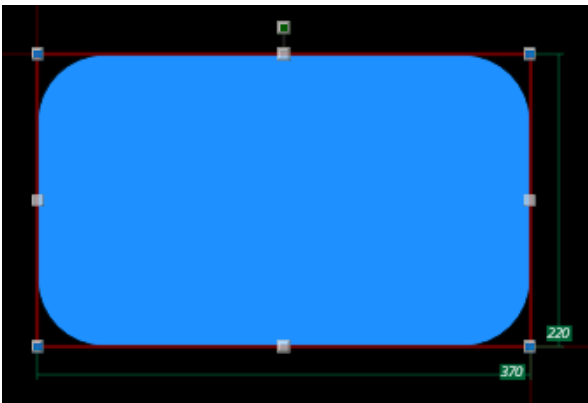


# CREW Manual

To add a rectangle with rounded corners it is necessary to assign values to the "Corner" property ("50" for example).



As you can see in the image, this gives you a rectangle with rounded corners.

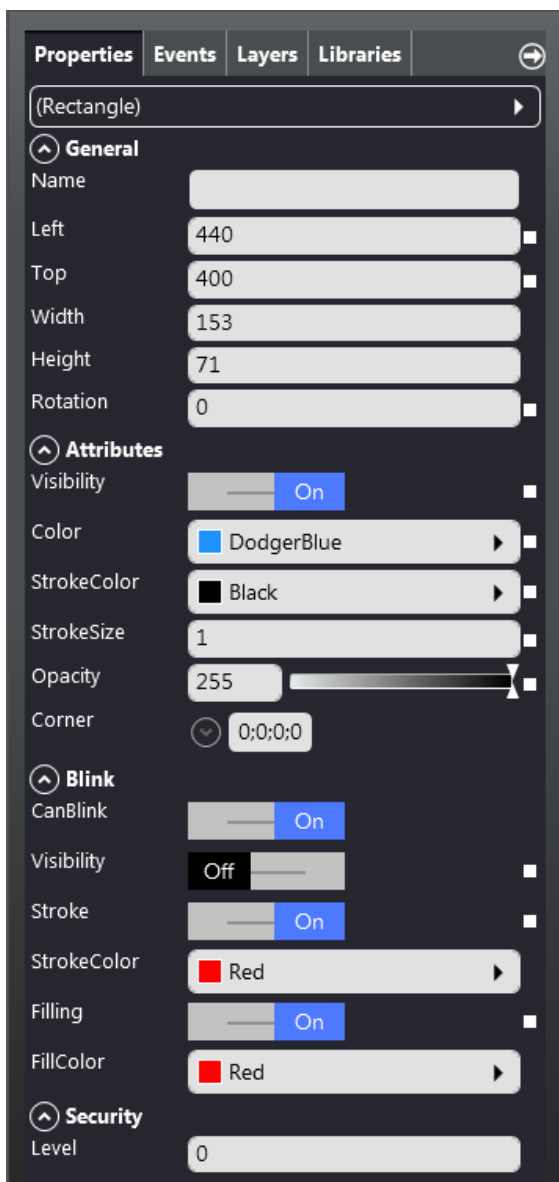


To determine the features of the "Rectangle", set them in the "Properties Editor", as shown in the section "[Rectangle Properties](#)".

# CREW Manual

## Rectangle Properties

The following image illustrates all the editable properties of the Rectangle. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



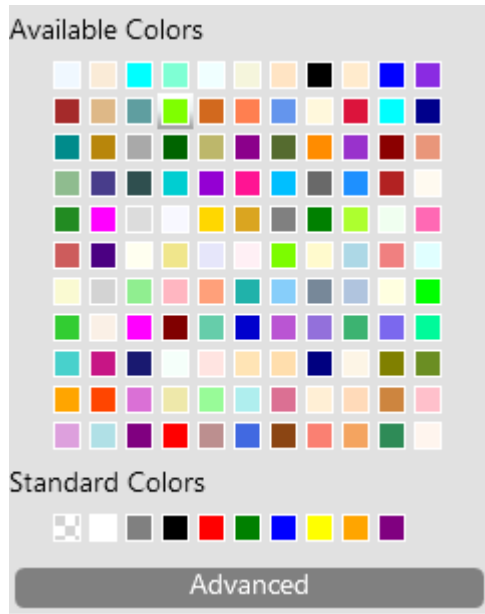
# CREW Manual

The following table describes all the editable properties of the Rectangle.

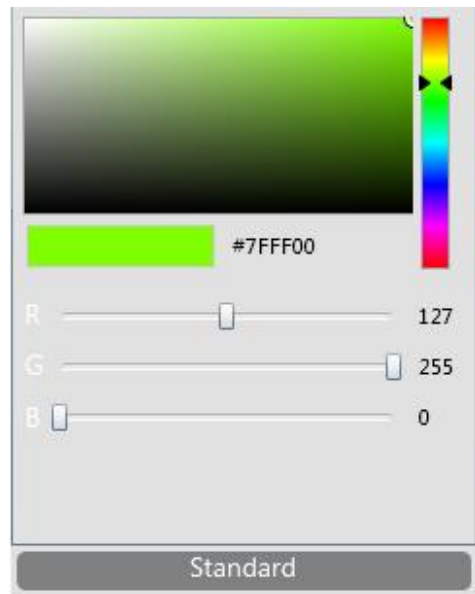
Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Rotation</b>	Determines the rotation degrees assigned to the object
<b>Attributes</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>Color</b>	Determines the fill colour of the object; the colour is selectable by RGB colour code or colour palette
<b>StrokeColor</b>	Determines the stroke colour (edges of the figure); the colour is selectable by RGB colour code or colour palette
<b>StrokeSize</b>	Determines the thickness of the stroke (edges of the figure)
<b>Opacity</b>	Determines the opacity of the object
<b>Corner</b>	Determines the roundness degree of the corners of the figure
<b>Blink</b>	
<b>CanBlink</b>	It allows you to enable or disable the Runtime blink
<b>Visibility</b>	Allows to display or hide the Runtime blink
<b>Stroke</b>	Determines the blinking of the object's edge
<b>StrokeColor</b>	Determines the blinking colour of the object's edge; it is selectable by RGB colour code or colour palette
<b>Filling</b>	Determines the blinking of the object's internal area
<b>FillColor</b>	Determines the blinking of the object's internal area; it is selectable by RGB colour code or colour palette
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project

# CREW Manual

The properties related to colours can be edited through the colour palette.

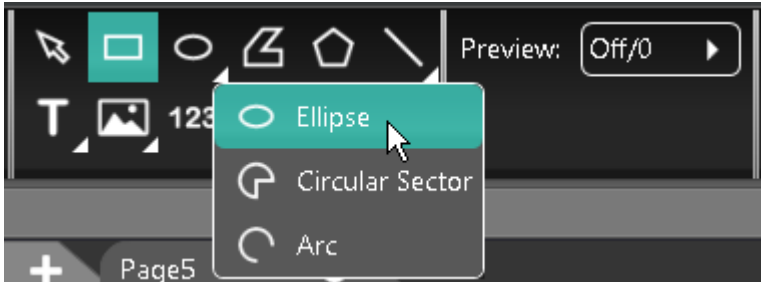


Click “Advanced” to select a colour using the RGB colour selection mask.

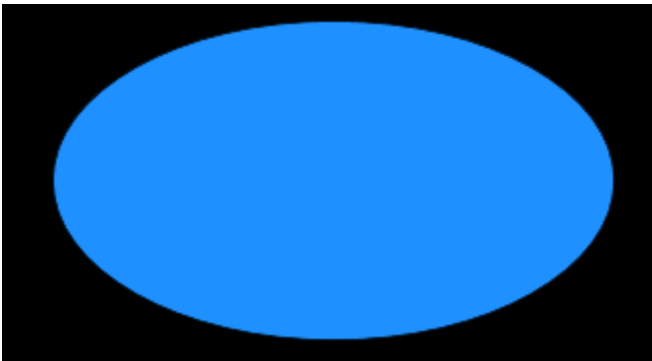


# CREW Manual

## Ellipse



The "Ellipse" icon on the "Graphics" menu is used to place an ellipse on the page, and change its size on that page.



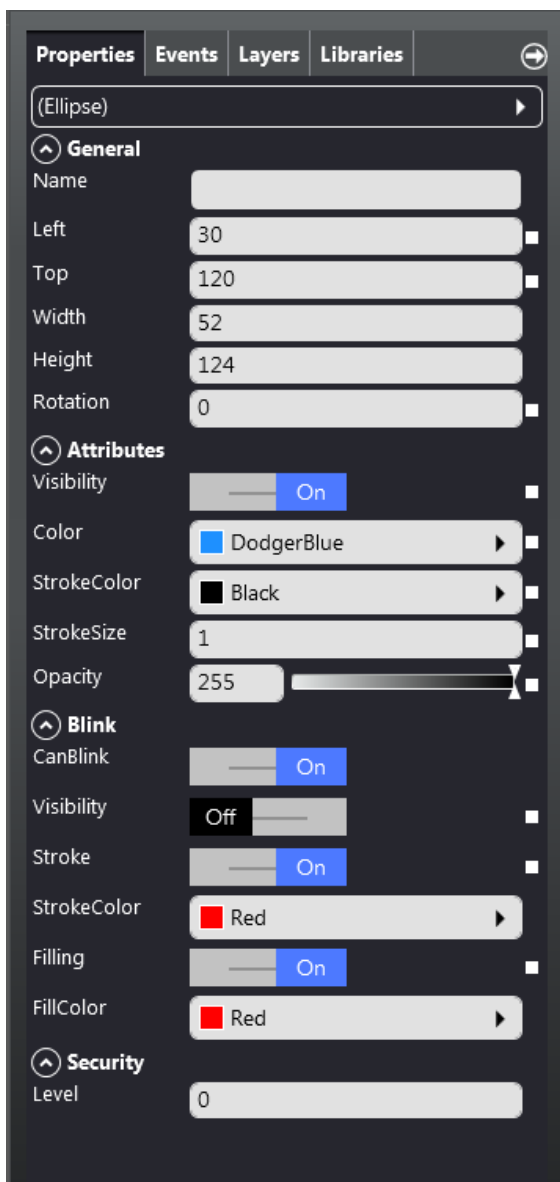
To determine the features of the Ellipse, set them in the "Properties Editor", as shown in the section "[Ellipse Properties](#)".



# CREW Manual

## Ellipse Properties

The following image illustrates all the editable properties of the Ellipse. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



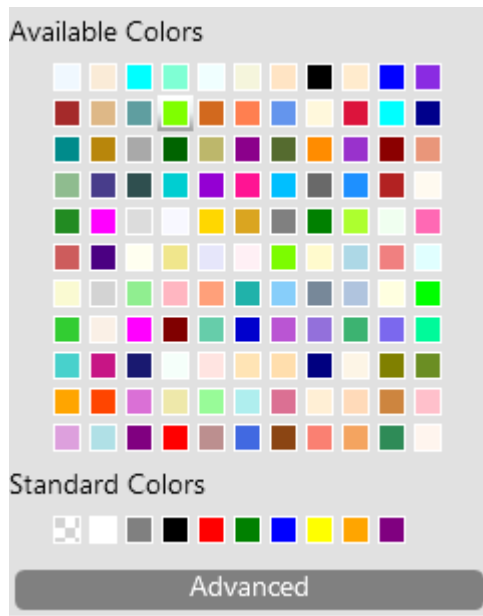
# CREW Manual

The following table describes all the editable properties of the Ellipse.

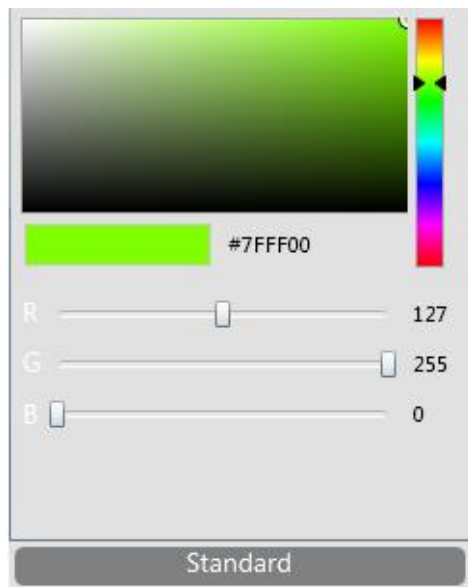
Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Rotation</b>	Determines the rotation degrees assigned to the object
<b>Attributes</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>Color</b>	Determines the fill colour of the object; the colour is selectable by RGB colour code or colour palette
<b>StrokeColor</b>	Determines the stroke colour (edges of the figure); the colour is selectable by RGB colour code or colour palette
<b>StrokeSize</b>	Determines the thickness of the stroke (edges of the figure)
<b>Opacity</b>	Determines the opacity of the object
<b>Blink</b>	
<b>CanBlink</b>	It allows you to enable or disable the Runtime blink
<b>Visibility</b>	Allows to display or hide the Runtime blin
<b>Stroke</b>	Determines the blinking of the object's edge
<b>StrokeColor</b>	Determines the blinking colour of the object's edge; it is selectable by RGB colour code or colour palette
<b>Filling</b>	Determines the blinking of the object's internal area
<b>FillColor</b>	Determines the blinking of the object's internal area; it is selectable by RGB colour code or colour palette
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project

# CREW Manual

The properties related to colours can be edited through the colour palette.



Click "Advanced" to select a colour using the RGB colour selection mask.



# CREW Manual

## Circular sector



The “Circular Sector” icon on the “Graphics” menu is used to place a circular sector on the page, and change its size on that page.

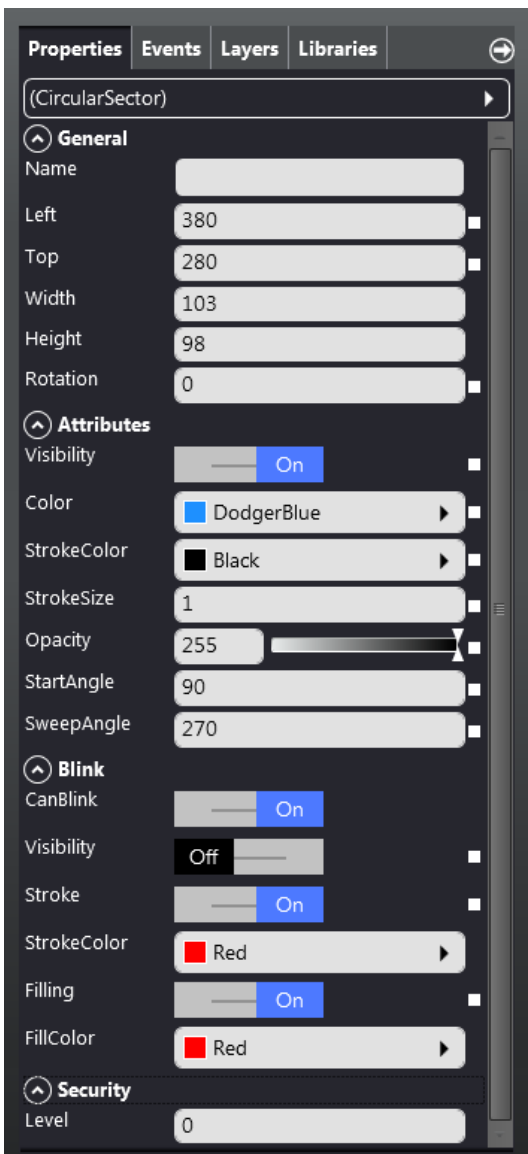


To determine the features of the Circular section, set them in the "Properties Editor", as shown in the section "[Circular sector Properties](#)".

# CREW Manual

## Circular sector Properties

The following image illustrates all the editable properties of the Circular Sector. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



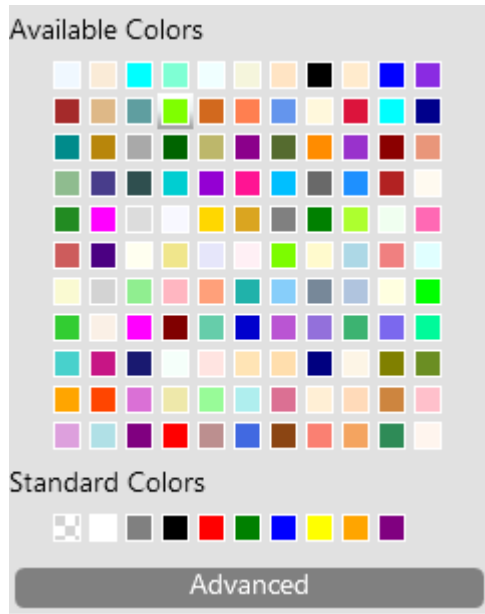
# CREW Manual

The following table describes all the editable properties of the Circular sector.

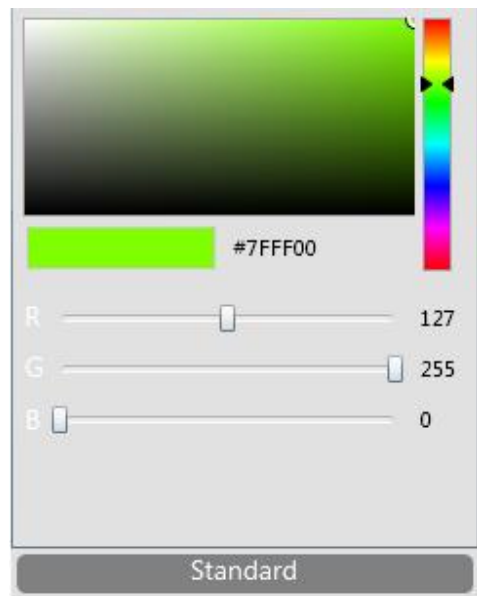
Properties	Description
<b>General</b>	
Name	Object ID
Left	Horizontal coordination of position
Top	Vertical coordination of position
Width	Width
Height	Height
Rotation	Determines the rotation degrees assigned to the object
<b>Attributes</b>	
Visibility	Determines whether the object should be displayed or not
Color	Determines the fill colour of the object; the colour is selectable by RGB colour code or colour palette
StrokeColor	Determines the stroke colour (edges of the figure); the colour is selectable by RGB colour code or colour palette
StrokeSize	Determines the thickness of the stroke (edges of the figure)
Opacity	Determines the opacity of the object
StartAngle	Determines, in degrees, the angle from which dimension begins the Section design
SweepAngle	Determines, in degrees, the opening angle of the Section
<b>Blink</b>	
CanBlink	It allows you to enable or disable the Runtime blink
Visibility	Allows to display or hide the Runtime blink
Stroke	Determines the blinking of the object's edge
StrokeColor	Determines the blinking colour of the object's edge; it is selectable by RGB colour code or colour palette
Filling	Determines the blinking of the object's internal area
FillColor	Determines the blinking of the object's internal area; it is selectable by RGB colour code or colour palette
<b>Security</b>	
Level	Allows to define the levels of authentication required to control access to specific areas of the project

# CREW Manual

The properties related to colours can be edited with the colour palette.

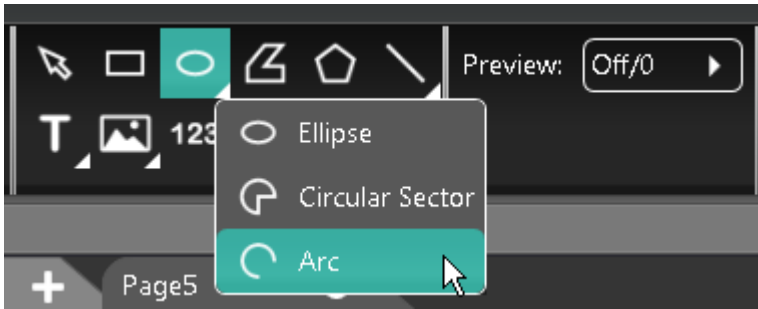


Click "Advanced" to select a colour using the RGB colour selection mask.

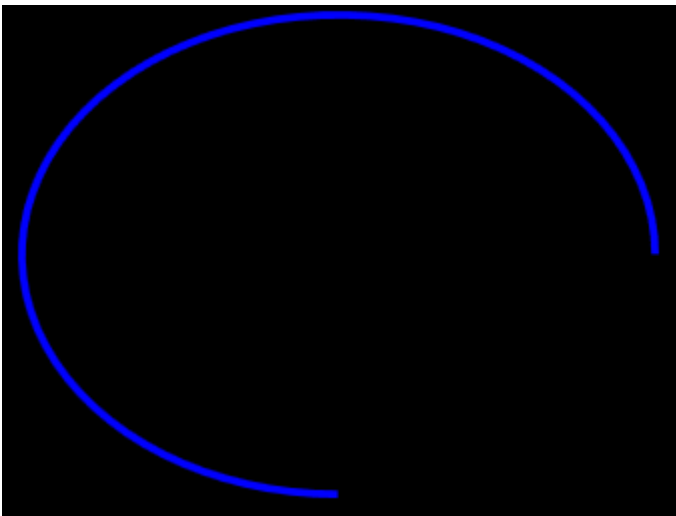


# CREW Manual

## Arch



The “Arch” icon on the “Graphics” menu is used to place an arch on the page, and change its size on that page.



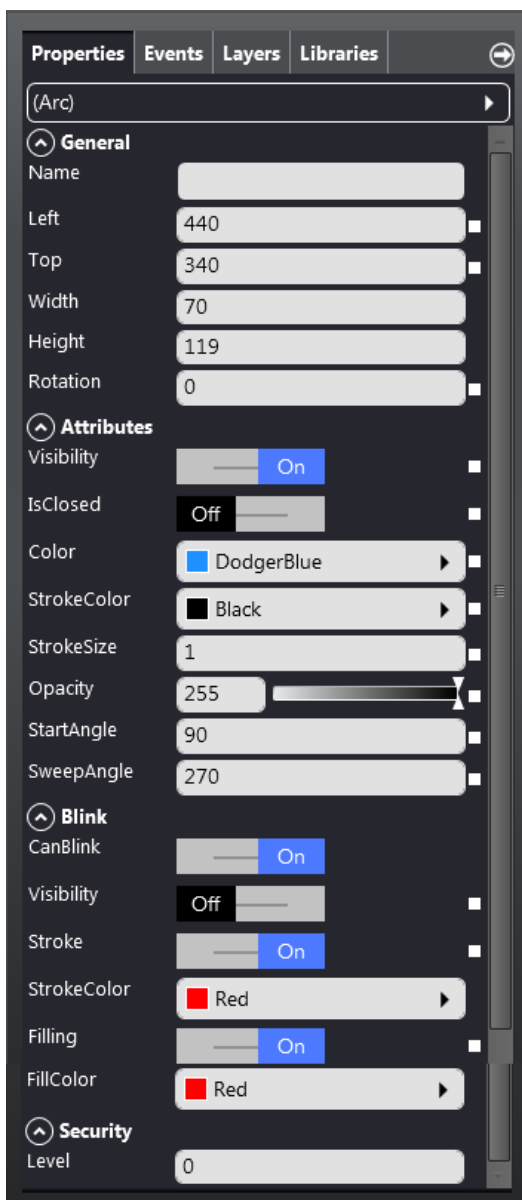
To determine the features of the Arch, set them in the "Properties Editor", as shown in the section "[Arch Properties](#)".



# CREW Manual

## Arch Properties

The following image illustrates all the editable properties of the Arch. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



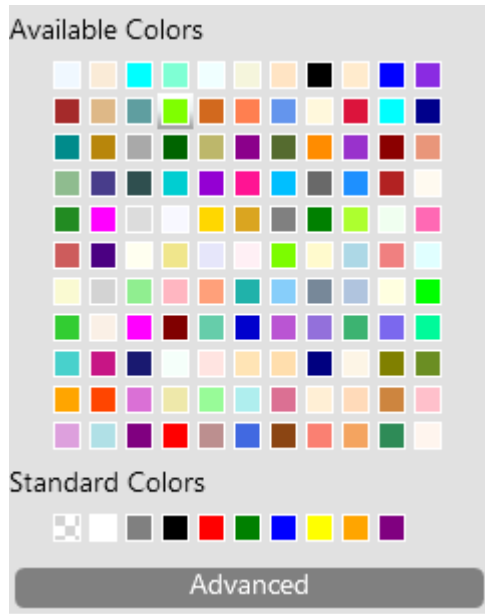
# CREW Manual

The following table describes all the editable properties of the Arch.

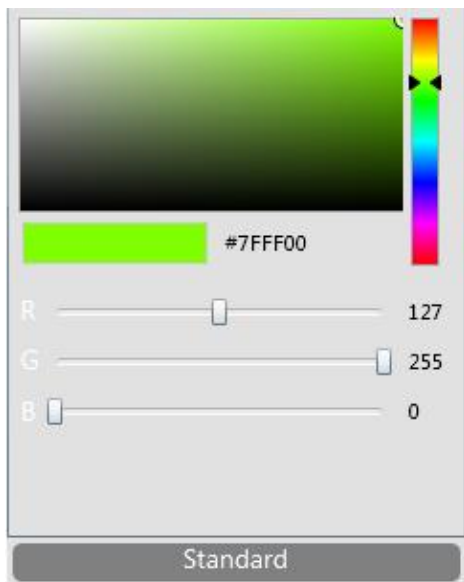
Properties	Description
<b>General</b>	
Name	Object ID
Left	Horizontal coordination of position
Top	Vertical coordination of position
Width	Width
Height	Height
Rotation	Determines the rotation degrees assigned to the object
<b>Attributes</b>	
Visibility	Determines whether the object should be displayed or not
IsClosed	Determines whether the arch should be closed or not
Color	Determines the fill colour of the object; the colour is selectable by RGB colour code or colour palette
StrokeColor	Determines the stroke colour (edges of the figure); the colour is selectable by RGB colour code or colour palette
StrokeSize	Determines the thickness of the stroke (edges of the figure)
Opacity	Determines the opacity of the object
StartAngle	Determines, in degrees, the angle from which dimension begins the Section design
SweepAngle	Determines, in degrees, the opening angle of the Section
<b>Blink</b>	
CanBlink	It allows you to enable or disable the Runtime blink
Visibility	Allows to display or hide the Runtime blink
Stroke	Determines the blinking of the object's edge
StrokeColor	Determines the blinking colour of the object's edge; it is selectable by RGB colour code or colour palette
Filling	Determines the blinking of the object's internal area
FillColor	Determines the blinking of the object's internal area; it is selectable by RGB colour code or colour palette
<b>Security</b>	
Level	Allows to define the levels of authentication required to control access to specific areas of the project

# CREW Manual

The properties related to colours can be edited through the colour palette.



Click “Advanced” to select a colour using the RGB colour selection mask.



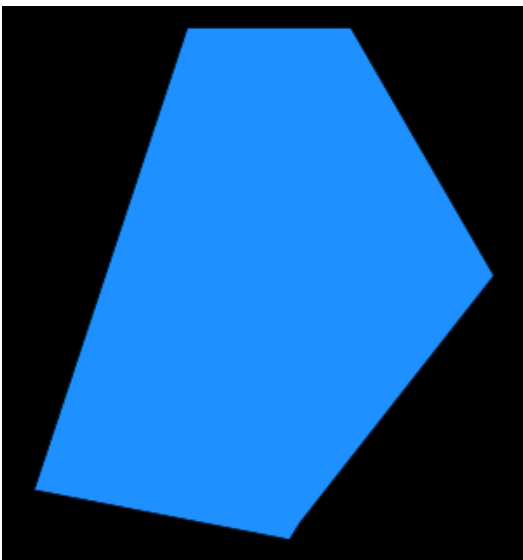
# CREW Manual

## Polygon



The "Polygon" icon in the "Graphics" menu is used to insert a circular sector on a page, drawing the sides within that page: after clicking the icon, click the points on the page where the vertices of the Polygon should appear. Every click will add a new vertex and Crew will show a preview of the Polygon as soon as the mouse is moved. Double click on the Polygon to confirm entry and finish editing it.

Once the Polygon is entered, its structure (namely its vertices) can be edited. To do so, select the Polygon and move one of its vertices: Crew also automatically moves the sides attached to the vertex. With this function you create an irregular "Polygon", i.e. with corners and sides that can be selected as needed.

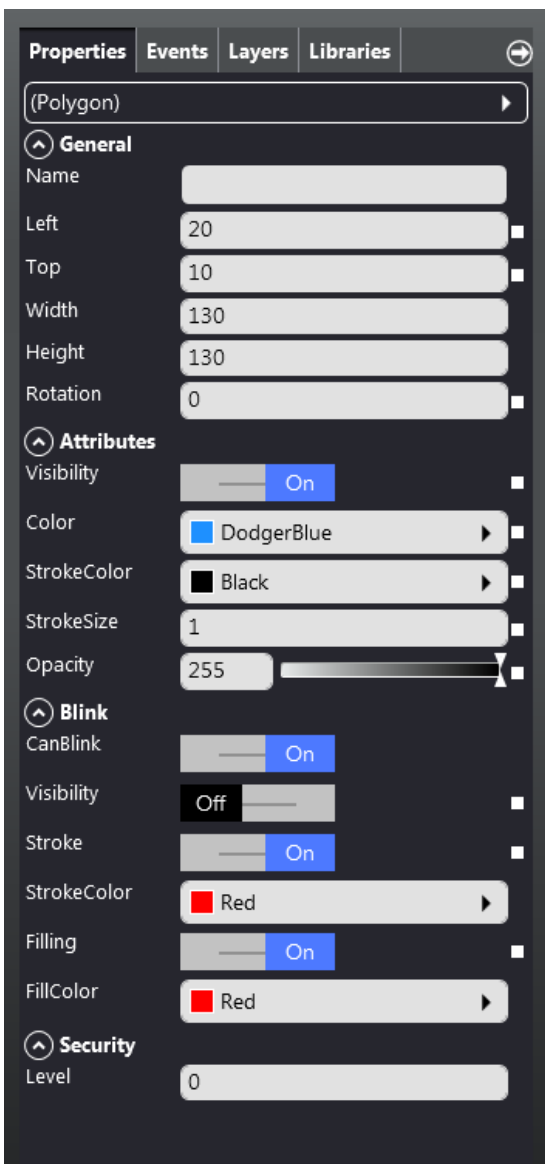


To determine the features of the Polygon, set them in the "Properties Editor", as shown in the section "[Polygon Properties](#)"

# CREW Manual

## Polygon Properties

The following image illustrates all the editable properties of the Polygon. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



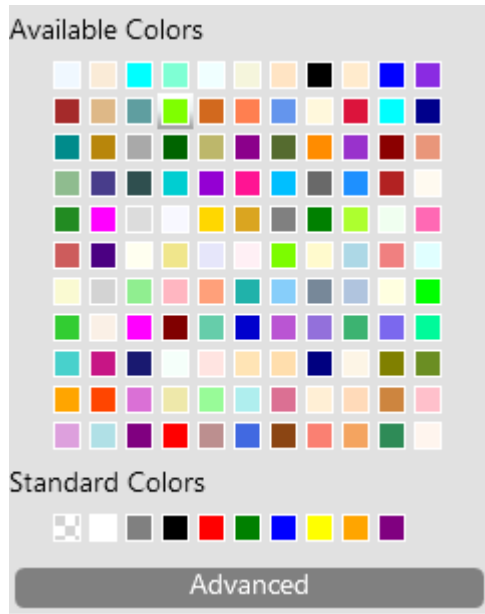
# CREW Manual

The following table describes all the editable properties of the Polygon.

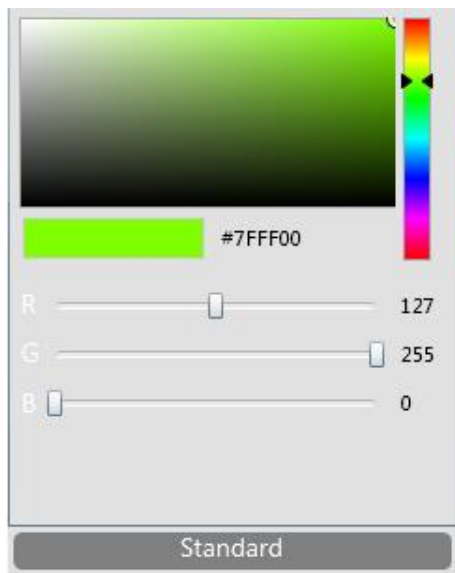
Properties	Description
<b>General</b>	
Name	Object ID
Left	Horizontal coordination of position
Top	Vertical coordination of position
Width	Width
Height	Height
<b>Attributes</b>	
Visibility	Determines whether the object should be displayed or not
Color	Determines the fill colour of the object; the colour is selectable by RGB colour code or colour palette
StrokeColor	Determines the stroke colour (edges of the figure); the colour is selectable by RGB colour code or colour palette
StrokeSize	Determines the thickness of the stroke (edges of the figure)
Opacity	Determines the opacity of the object
<b>Blink</b>	
CanBlink	It allows you to enable or disable the Runtime blink
Visibility	Allows to display or hide the Runtime blink
Stroke	Determines the blinking of the object's edge
StrokeColor	Determines the blinking colour of the object's edge; it is selectable by RGB colour code or colour palette
Filling	Determines the blinking of the object's internal area
FillColor	Determines the blinking of the object's internal area; it is selectable by RGB colour code or colour palette
<b>Security</b>	
Level	Allows to define the levels of authentication required to control access to specific areas of the project

# CREW Manual

The properties related to colours can be edited through the colour palette.



Click "Advanced" to select a colour using the RGB colour selection mask.

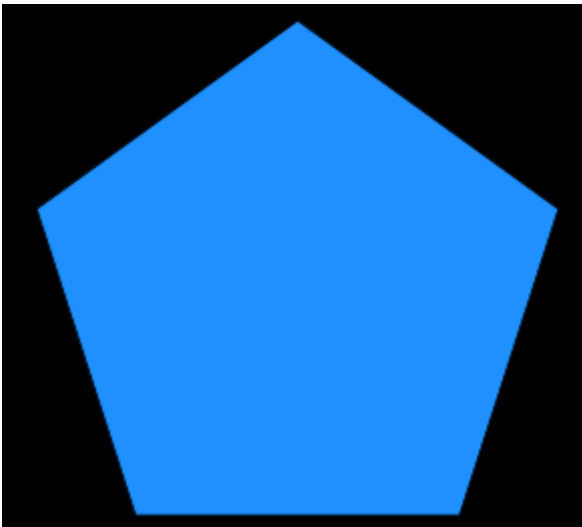


# CREW Manual

## Regular polygon



The “Regular Polygon” icon on the “Graphics” menu is used to place a regular polygon on the page, drawing its sides on that page. Double click to confirm.



By default a pentagon (5 sides) is drawn, but it is possible to change the number of sides (vertices). Simply change the number of Points in the “Regular polygon properties”. With this function only regular polygons are created, that is, with all corners and sides having the same measurement.

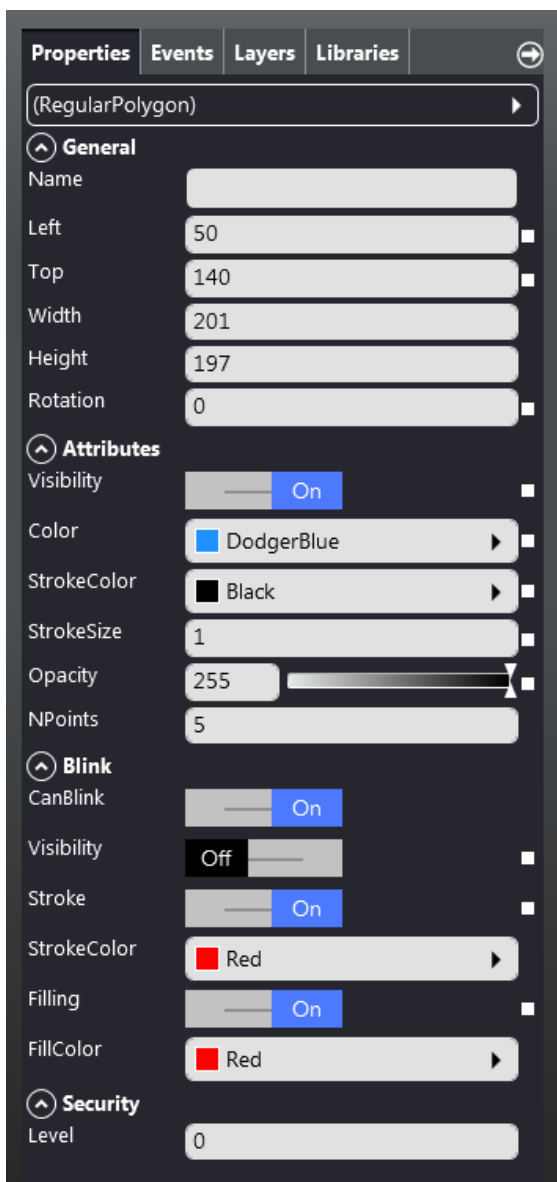
To determine the features of the Polygon, set them in the "Properties Editor", as shown in the section "[Regular Polygon Properties](#)".



# CREW Manual

## Regular Polygon Properties

The following image illustrates all the editable properties of the Regular Polygon. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



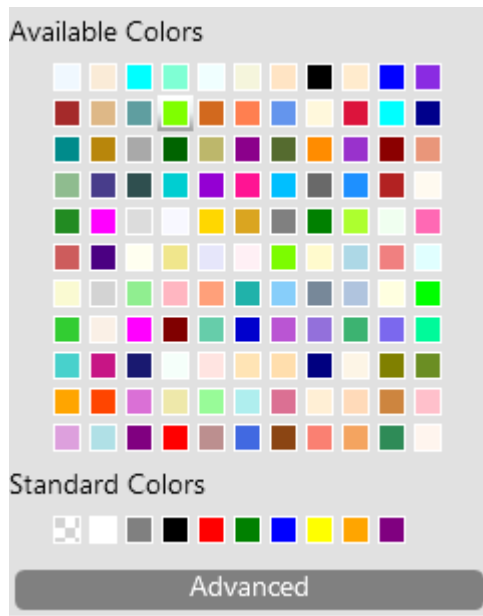
# CREW Manual

The following table describes all the editable properties of the Regular Polygon.

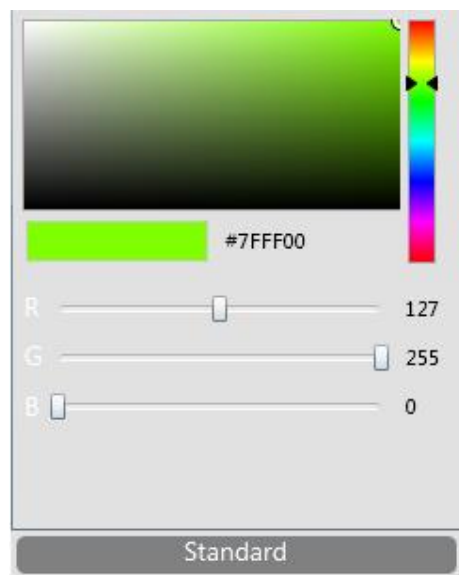
Properties	Description
<b>General</b>	
Name	Object ID
Left	Horizontal coordination of position
Top	Vertical coordination of position
Width	Width
Height	Height
<b>Attributes</b>	
Visibility	Determines whether the object should be displayed or not
Color	Determines the fill colour of the object; the colour is selectable by RGB colour code or colour palette
StrokeColor	Determines the stroke colour (edges of the figure); the colour is selectable by RGB colour code or colour palette
StrokeSize	Determines the thickness of the stroke (edges of the figure)
Opacity	Determines the opacity of the object
Npoints	Determines the number of sides of the Polygon
<b>Blink</b>	
CanBlink	It allows you to enable or disable the Runtime blink
Visibility	Allows to display or hide the Runtime blink
Stroke	Determines the blinking of the object's edge
StrokeColor	Determines the blinking colour of the object's edge; it is selectable by RGB colour code or colour palette
Filling	Determines the blinking of the object's internal area
FillColor	Determines the blinking of the object's internal area; it is selectable by RGB colour code or colour palette
<b>Security</b>	
Level	Allows to define the levels of authentication required to control access to specific areas of the project

# CREW Manual

The properties related to colours can be edited through the colour palette.

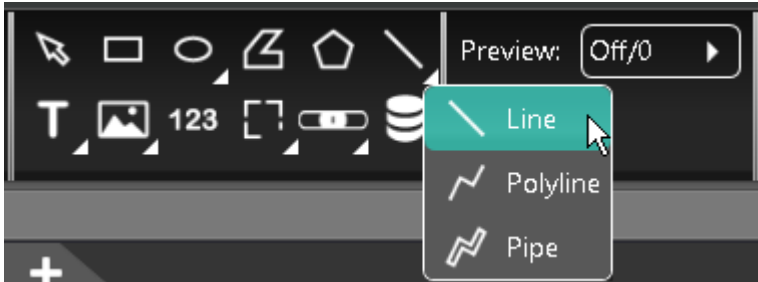


Click "Advanced" to select a colour using the RGB colour selection mask.

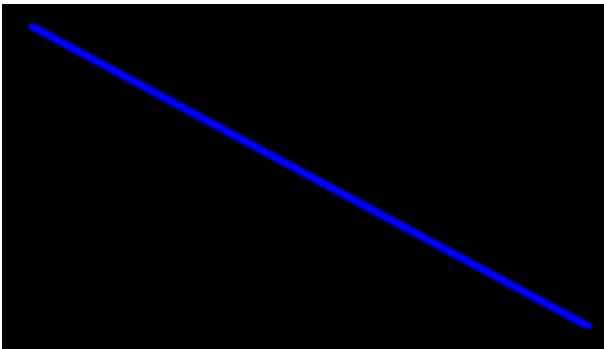


# CREW Manual

## Line



The “Line” icon on the “Graphics” menu is used to place a line on the page, drawing it with the mouse on the page.

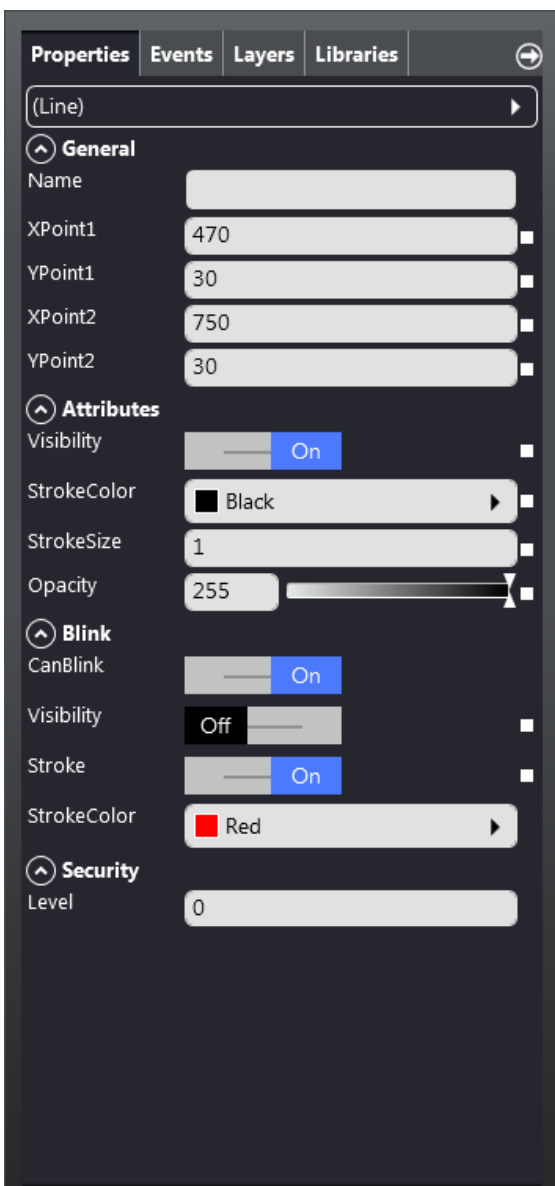


To determine the features of the Line, set them in the "Properties Editor", as shown in the section "[Line Properties](#)".

# CREW Manual

## Line Properties

The following image illustrates all the editable properties of the Line. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



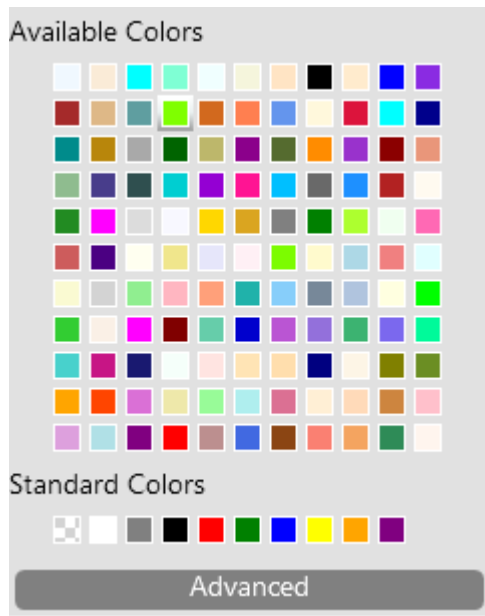
# CREW Manual

The following table describes all the editable properties of the Line.

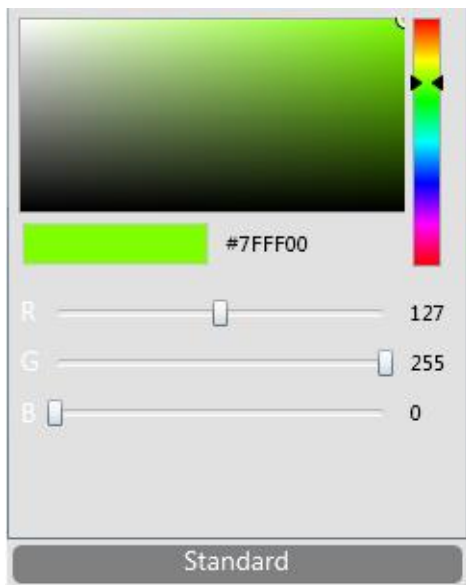
Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>XPoint1</b>	First horizontal coordination of position
<b>YPoint1</b>	First vertical coordination of position
<b>XPoint2</b>	Second horizontal coordination of position
<b>YPoint2</b>	Second vertical coordination of position
<b>Attributes</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>StrokeColor</b>	Determines the stroke colour (edges of the figure); the colour is selectable by RGB colour code or colour palette
<b>StrokeSize</b>	Determines the thickness of the stroke (edges of the figure)
<b>Opacity</b>	Determines the opacity of the object
<b>Blink</b>	
<b>CanBlink</b>	It allows you to enable or disable the Runtime blink
<b>Visibility</b>	Allows to display or hide the Runtime blink
<b>StrokeColor</b>	Determines the blinking colour of the object's edge; it is selectable by RGB colour code or colour palette
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project

# CREW Manual

The properties related to colours can be edited through the colour palette.

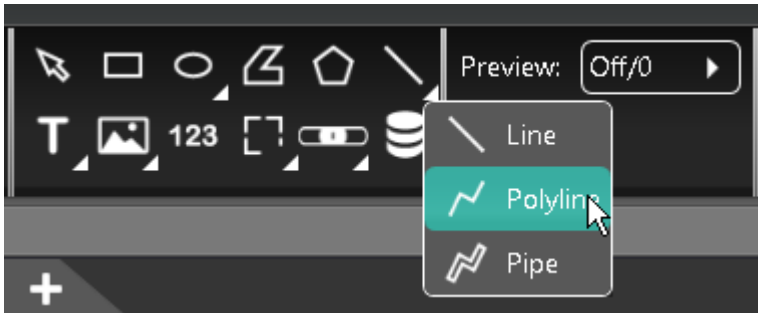


Click “Advanced” to select a colour using the RGB colour selection mask.

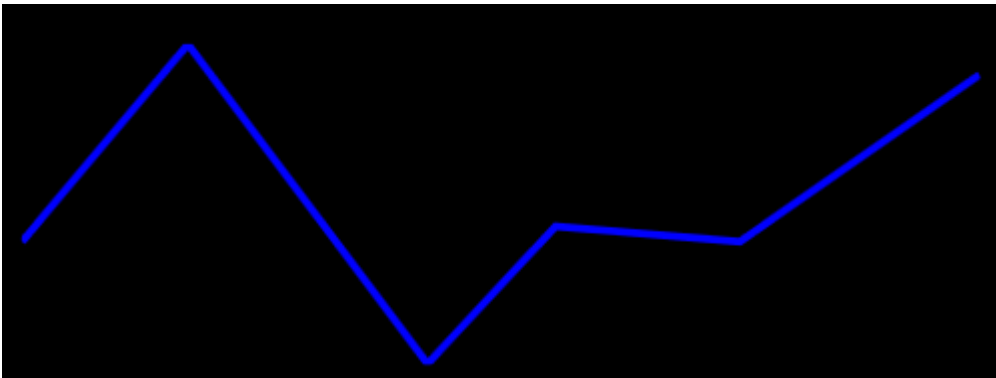


# CREW Manual

## Polygonal chain



The “Polygonal Chain” icon on the “Graphics” menu is used to place a polygonal chain on the page, drawing it with the mouse on the page. Double click to confirm.



After clicking the icon, click the points where the vertices of the figure should appear (the beginning and the end of the various sections). Every click will add a new section with the relative vertex and Crew will show a preview of the Polygon as soon as the mouse is moved. Double click on the Polygonal Chain to confirm entry and finish editing it.

With this function an open line is created which differs from the irregular Polygon because it does not necessarily form a closed geometric figure.

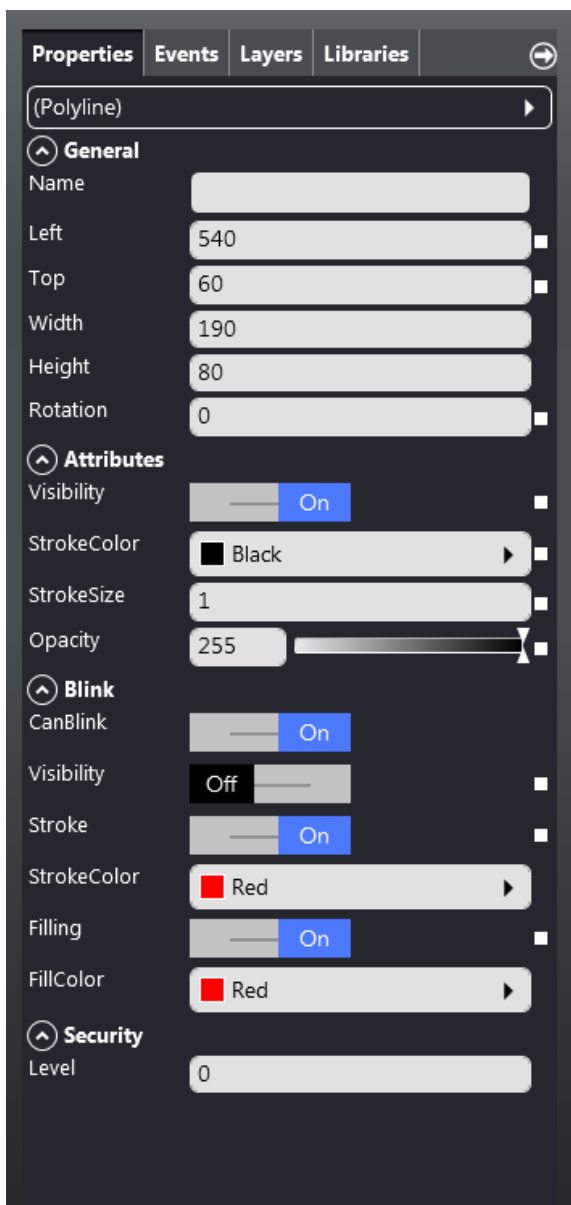
To determine the features of the Polygonal chain, set them in the "Properties Editor", as shown in the section "[Polygonal chain Properties](#)".



# CREW Manual

## Polygonal chain Properties

The following image illustrates all the editable properties of the Polygonal Chain. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



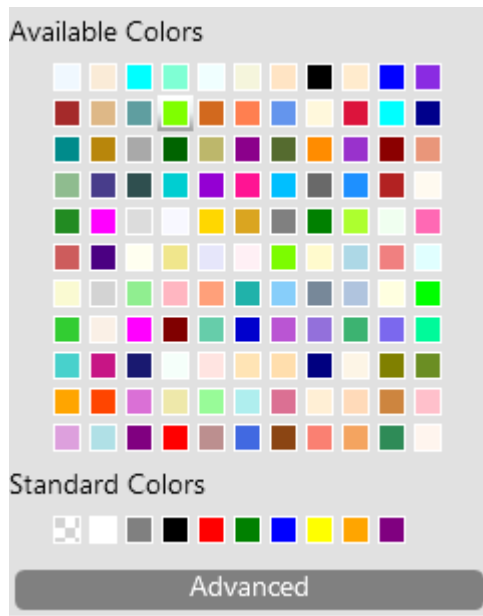
# CREW Manual

The following table describes all the editable properties of the Polygonal chain.

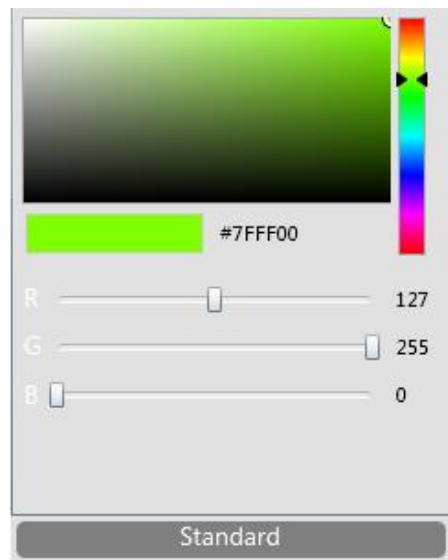
Properties	Description
<b>General</b>	
Name	Object ID
Left	Horizontal coordination of position
Top	Vertical coordination of position
Width	Width
Height	Height
<b>Attributes</b>	
Visibility	Determines whether the object should be displayed or not
StrokeColor	Determines the stroke colour (edges of the figure); the colour is selectable by RGB colour code or colour palette
StrokeSize	Determines the thickness of the stroke (edges of the figure)
Opacity	Determines the opacity of the object
<b>Blink</b>	
CanBlink	It allows you to enable or disable the Runtime blink
Visibility	Allows to display or hide the Runtime blink
Stroke	Determines the blinking of the object's edge
StrokeColor	Determines the blinking colour of the object's edge; it is selectable by RGB colour code or colour palette
Filling	Determines the blinking of the object's internal area
FillColor	Determines the blinking of the object's internal area; it is selectable by RGB colour code or colour palette
<b>Security</b>	
Level	Allows to define the levels of authentication required to control access to specific areas of the project

# CREW Manual

The properties related to colours can be edited through the colour palette.

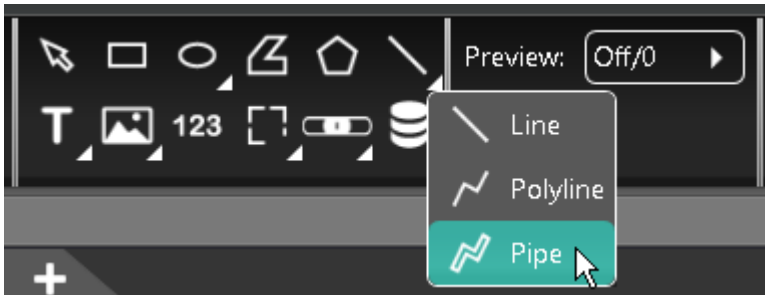


Click “Advanced” to select a colour using the RGB colour selection mask.

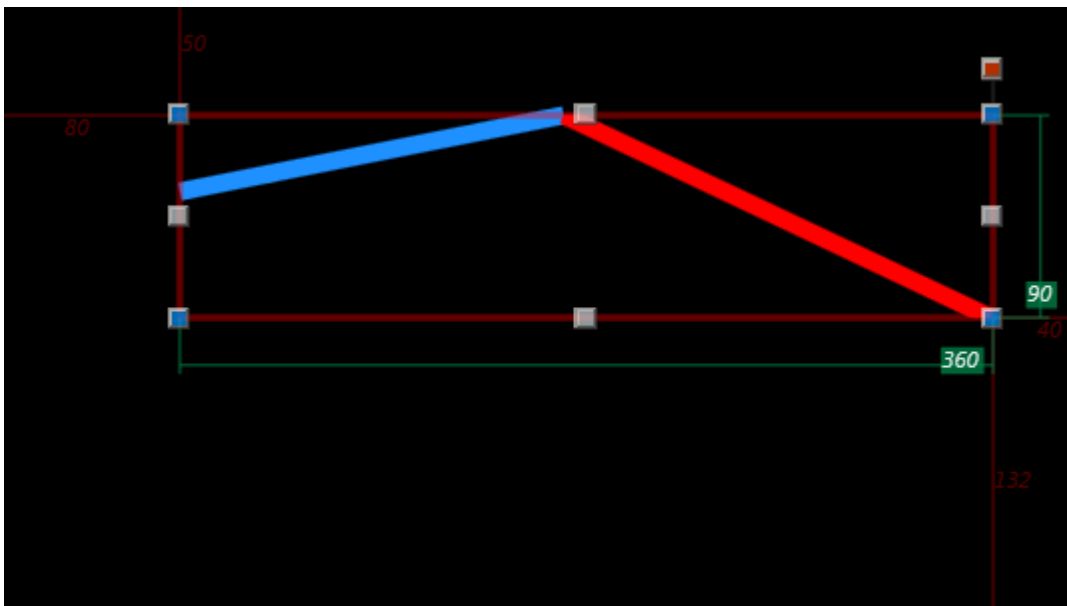


# CREW Manual

## Pipeline



The “Pipeline” icon on the “Graphics” menu is used to place a pipeline on the page, drawing it with the mouse on the page. Double click to confirm.

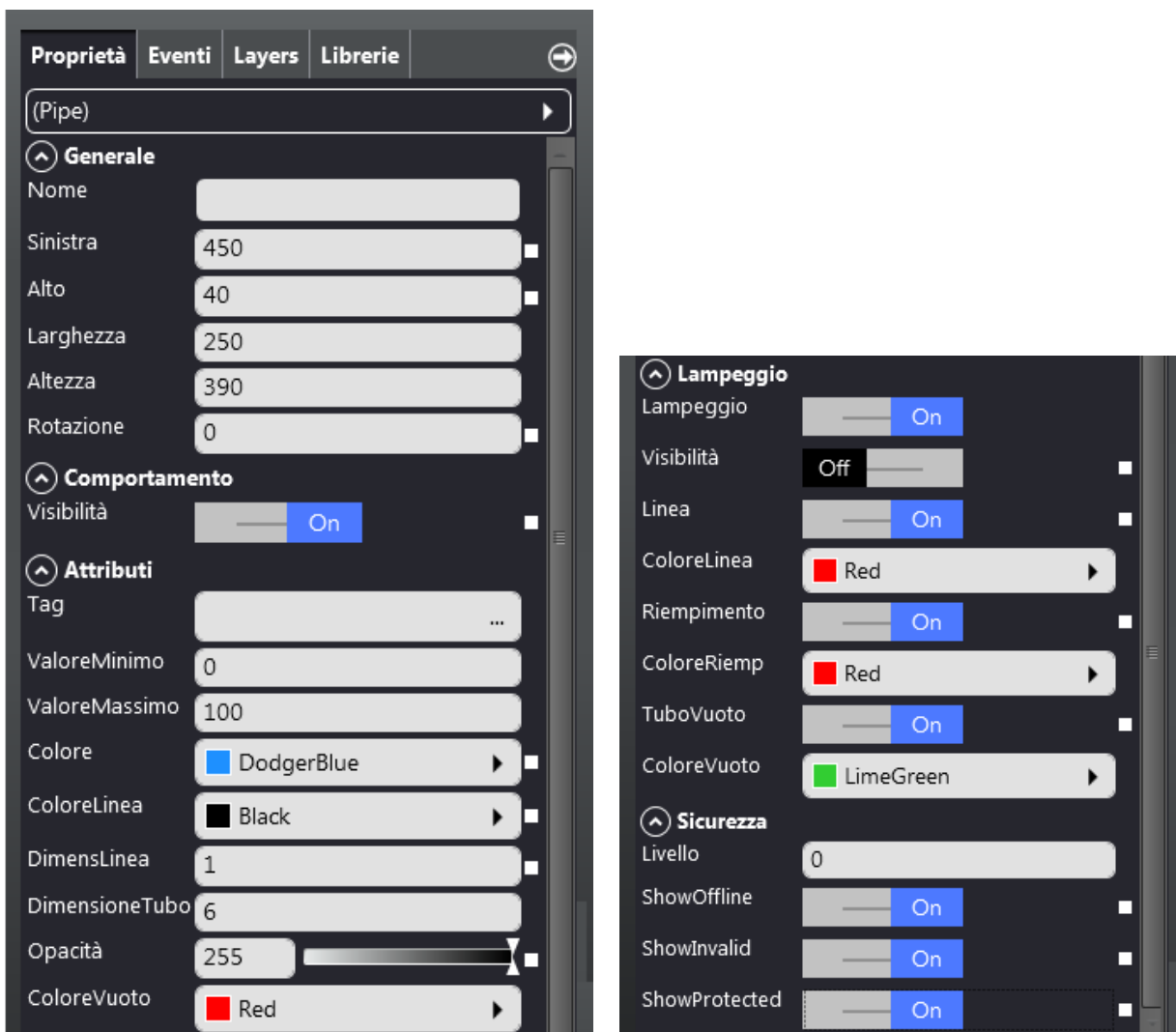


To determine the features of the Polygonal chain, set them in the "Properties Editor", as shown in the section "[Pipeline Properties](#)".

# CREW Manual

## Pipeline Properties

The following image illustrates all the editable properties of the Pipeline. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

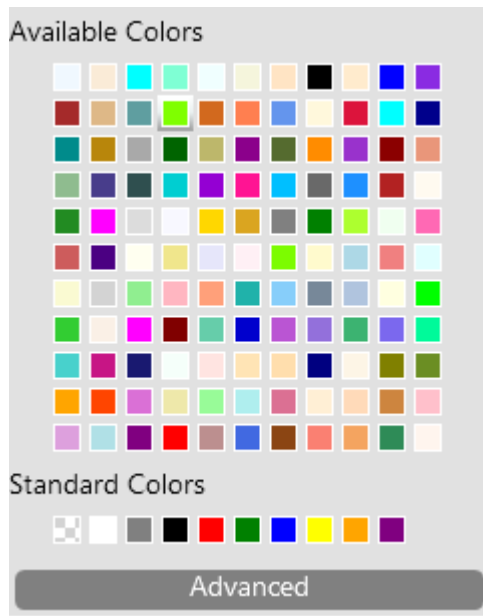
The following table describes all the editable properties of the Pipeline.

Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>Attributes</b>	
<b>Tag</b>	Tag linked to the value of the Pipe
<b>MinValue</b>	Represents the minimum value of the Pipe display scale
<b>MaxValue</b>	Represents the maximum value of the Pipe display scale
<b>Color</b>	Determines the colour used for filling the Pipe
<b>StrokeColor</b>	Determines the stroke colour (edges of the figure); the colour is selectable by RGB colour code or colour palette
<b>StrokeSize</b>	Determines the thickness of the stroke (edges of the figure)
<b>PipeSize</b>	Indicates the size of the Pipe

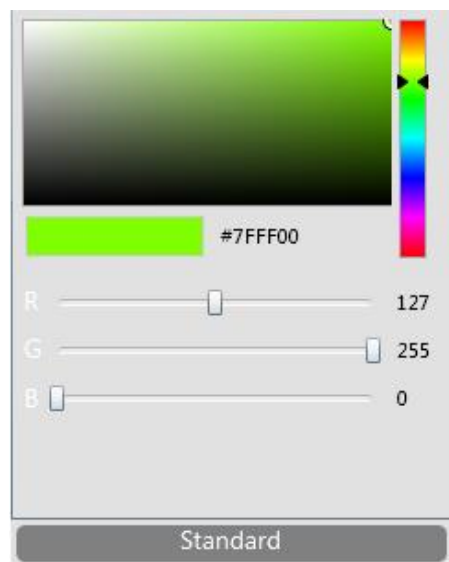
<b>Opacity</b>	Determines the opacity of the object
<b>VoidColor</b>	Determines the colour of the empty part of the Pipe
<b>Blink</b>	
<b>CanBlink</b>	It allows you to enable or disable the Runtime blink
<b>Visibility</b>	Allows to display or hide the Runtime blink
<b>Stroke</b>	Determines the blinking of the object's edge
<b>StrokeColor</b>	Determines the blinking colour of the object's edge; it is selectable by RGB colour code or colour palette
<b>Filling</b>	Determines the blinking of the object's internal area
<b>FillColor</b>	Determines the blinking of the object's internal area; it is selectable by RGB colour code or colour palette
<b>PipeVoid</b>	Determines the blinking of the Pipe's empty part
<b>EmptyColor</b>	Determines the blinking colour of the Pipe's empty part
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project
<b>ShowOffline</b>	Enables the displaying of a "status" icon in the object when it is in the offline condition
<b>ShowInvalid</b>	Enables the displaying of a "status" icon in the object when its current value is invalid
<b>ShowProtected</b>	Enables the displaying of a "status" icon in the object when it is protected for the current user

# CREW Manual

The properties related to colours can be edited through the colour palette.

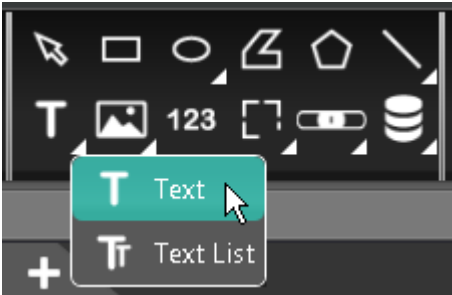


Click "Advanced" to select a colour using the RGB colour selection mask.

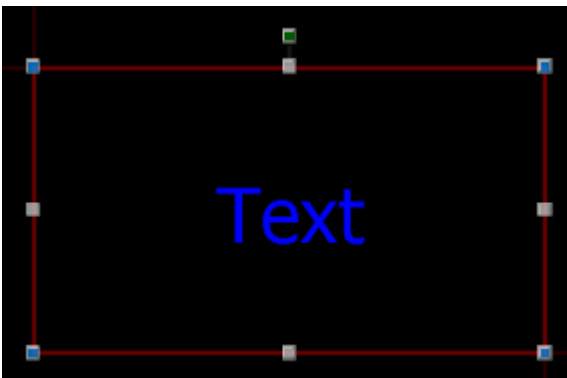


# CREW Manual

## Text - Label -



The "Text" (or "Label") icon in the "Graphics" menu is used to add text to a page, drawing the size of it on that page.



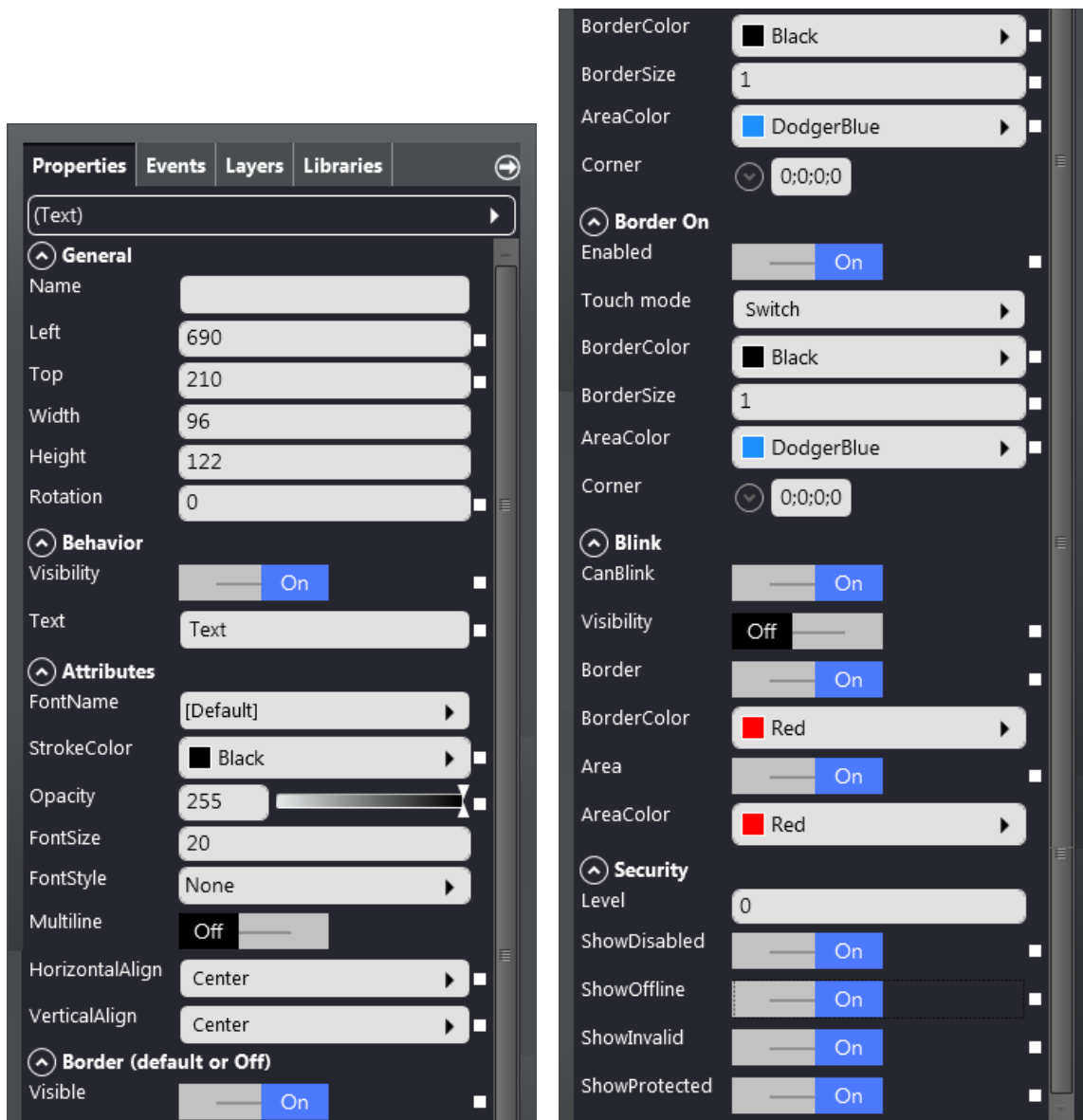
To determine the features of the "Text" field, set them in the "Properties Editor", as shown in the section "[Text - Label - Properties](#)".



# CREW Manual

## Text - Label - Properties

The following image illustrates all the editable properties of the “Text” box. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

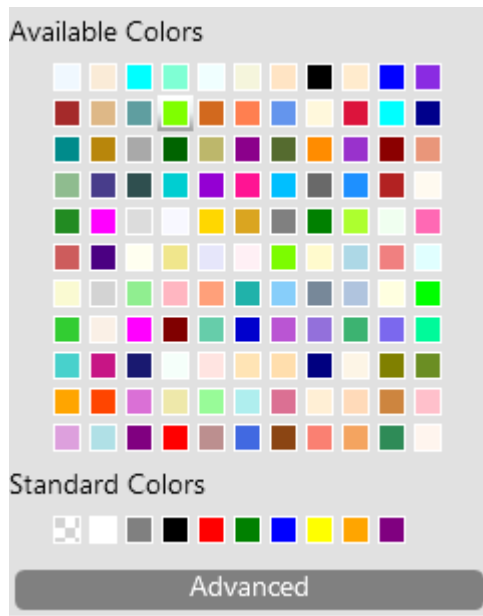
The following table describes all the editable properties of the Text box.

Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Rotation</b>	Determines the rotation degrees assigned to the object
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>Text</b>	Text displayed on the label
<b>Attributes</b>	
<b>FontName</b>	Text Font displayed on the label
<b>StrokeColor</b>	Determines the stroke colour (edges of the figure); the colour is selectable by RGB colour code or colour palette
<b>Opacity</b>	Determines the opacity of the object
<b>FontSize</b>	Establishes the Font size of the label
<b>FontStyle</b>	Font style. Any combination of the following features: - None - Italics - Bold - Underline
<b>Multiline</b>	If the text automatically starts a newline, it determines the number of lines needed with respect to the current width
<b>HorizontalAlign</b>	This function allows to position the image horizontally (Centred, Right or Left)
<b>VerticalAlign</b>	This function allows to position the image vertically (Centred, Top or Bottom)
<b>Border (default / Off)</b>	
<b>Visible</b>	Determines whether the border should be visible or not
<b>Border Color</b>	Determines the color of the border

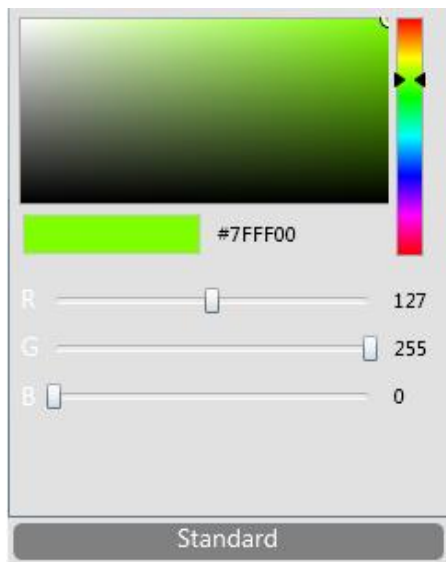
<b>Border Size</b>	Determines the border thickness
<b>Area Color</b>	Determines the area fill color
<b>Corner</b>	Determines the roundness of the angles
<b>Border On</b>	
<b>Enabled</b>	Determines whether the user can interact with the object
<b>TouchMode</b>	Determines the mode of interaction of the button: - SWITCH: The button acts as a normal button switch with on-pressed and on-released events generated when the user has released the button. - CONTACT: the button enters the "pressed" state as soon as the user touches the screen, and enters the state "released" when there is no more contact with the screen. - BUTTON: It is not necessary to associate a variable to the change of the button state, events associated to this are used
<b>BorderColor</b>	Determines the color of the border
<b>BorderSize</b>	Determines the border thickness
<b>AreaColor</b>	Determines the area fill color
<b>Corner</b>	Determines the roundness of the angles
<b>Blink</b>	
<b>CanBlink</b>	It allows you to enable or disable the Runtime blink
<b>Visibility</b>	Allows to display or hide the Runtime blink
<b>Border</b>	Determines the blinking or less of the border
<b>BorderColor</b>	Determines the color of the border during blinking
<b>Area</b>	Determines the blinking or less of the area
<b>AreaColor</b>	Determines the area fill color during blinking
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project
<b>ShowDisabled</b>	Enables the displaying of a "status" icon in the object when its use has been disabled
<b>ShowOffline</b>	Enables the displaying of a "status" icon in the object when it is in the offline condition
<b>ShowInvalid</b>	Enables the displaying of a "status" icon in the object when its current value is invalid
<b>ShowProtected</b>	Enables the displaying of a "status" icon in the object when it is protected for the current user

# CREW Manual

The properties related to colours can be edited through the colour palette.



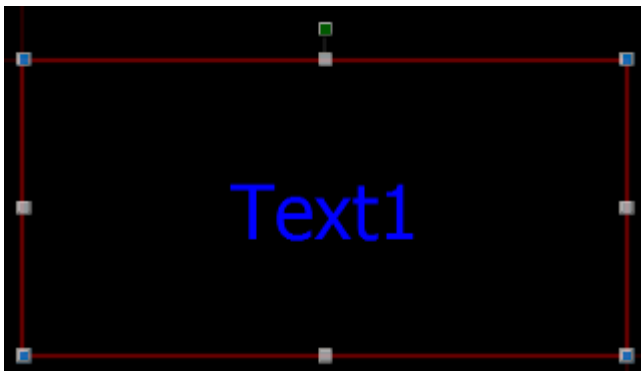
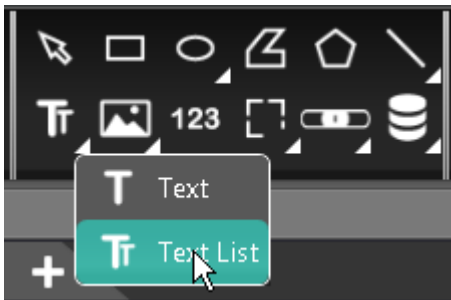
Click “Advanced” to select a colour using the RGB colour selection mask.



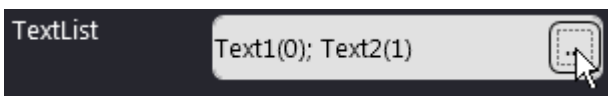
# CREW Manual

## Text list

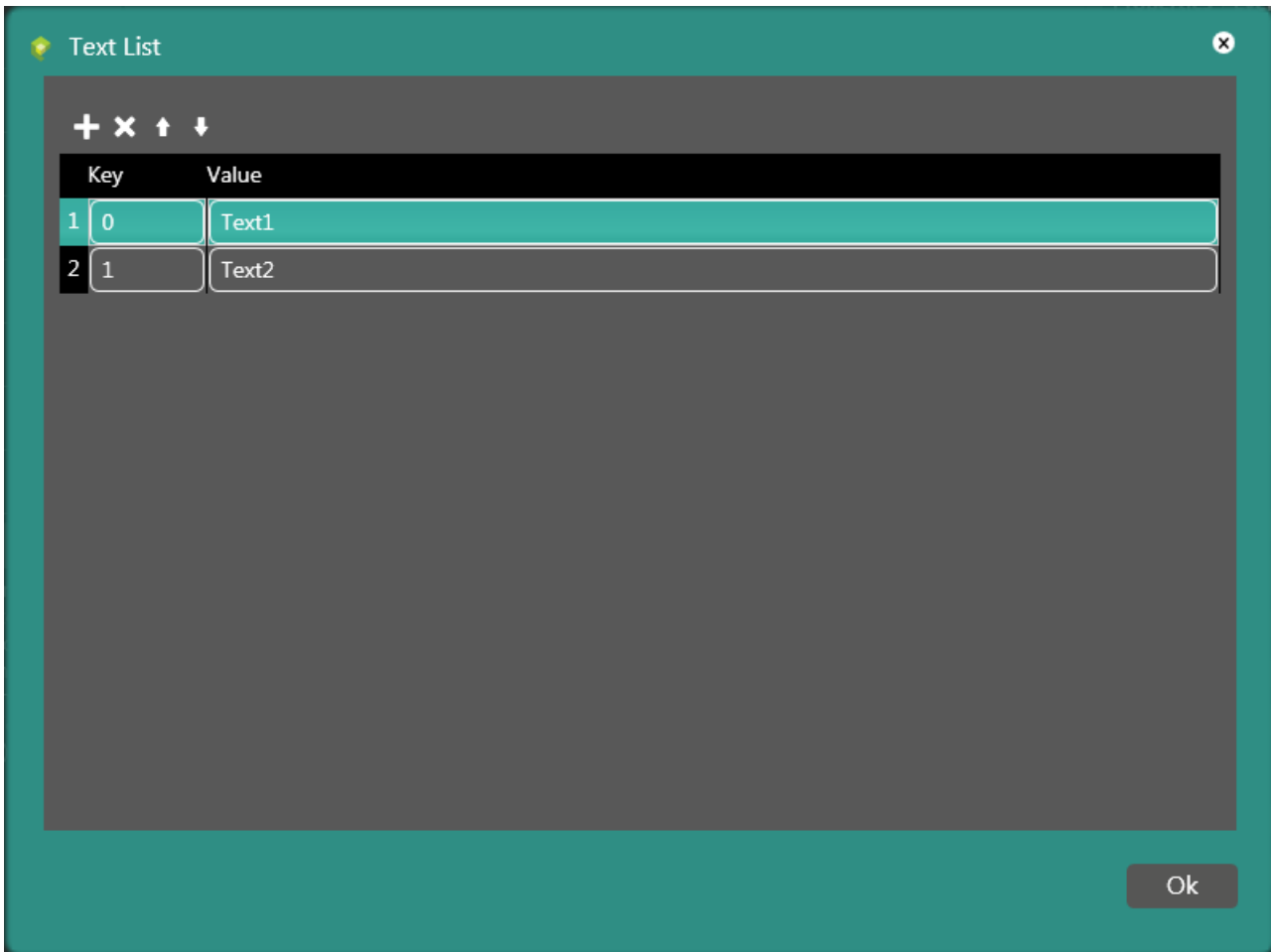
The Text List is a text container that is useful for creating value fields. Click the "Text List" icon in the "Graphics" menu to add a list of texts to a page, drawing the size of it on that page.



Press the relative key to open a window with the texts that can be linked to the object.



# CREW Manual



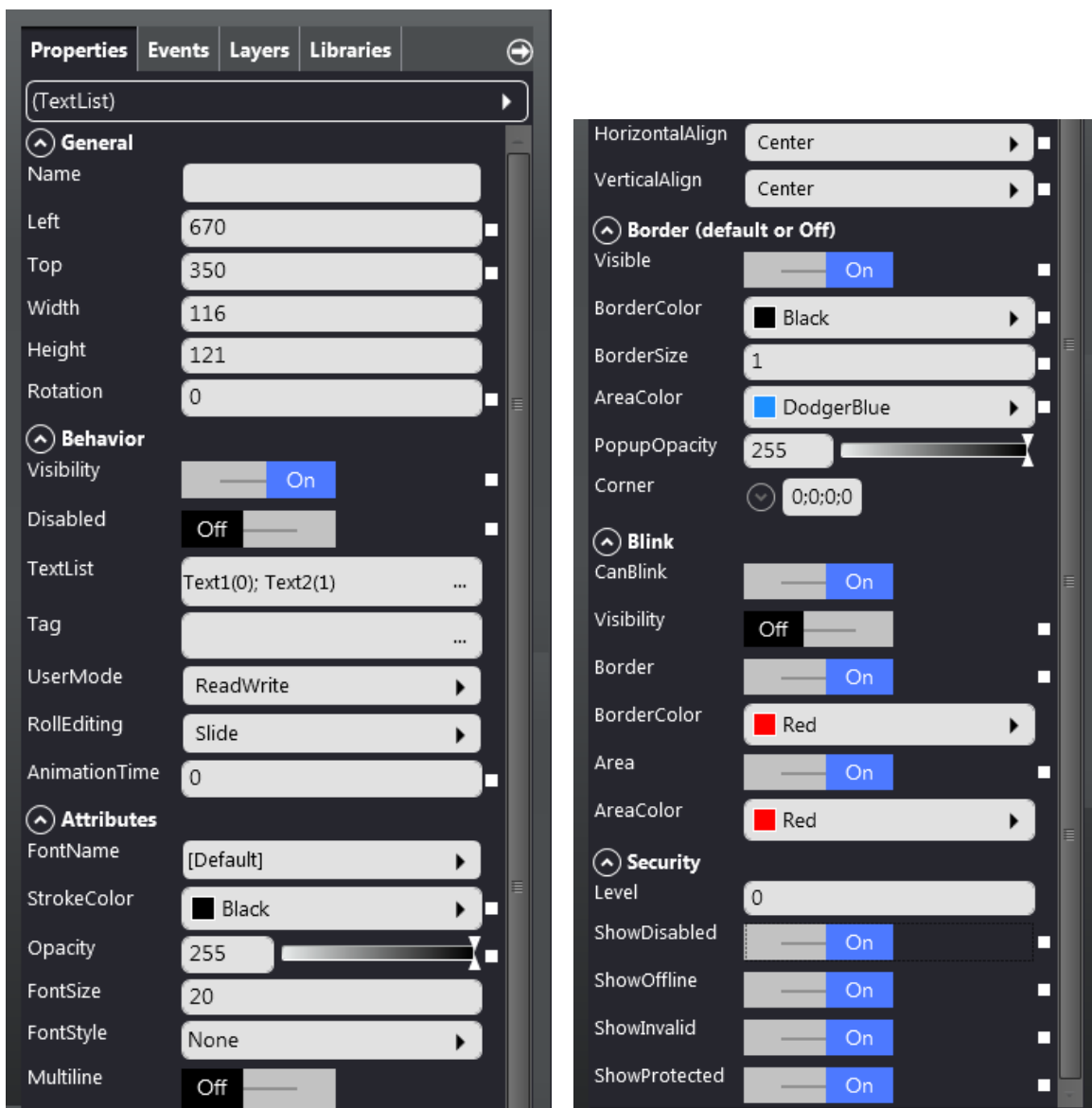
When the Tag linked to the "Text List" object has a value of 0, the text displayed in Runtime will be "Text1", whereas when the tag has a value of 1, the text displayed in Runtime will be "Text2". If the linked Tag has a "whole" data type, more than two texts may be entered.

To determine the features of the "Text List" field, set them in the "Properties Editor", as shown in the section "Text List Properties".

# CREW Manual

## Text list Properties

The following image illustrates all the editable properties of the Text List. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

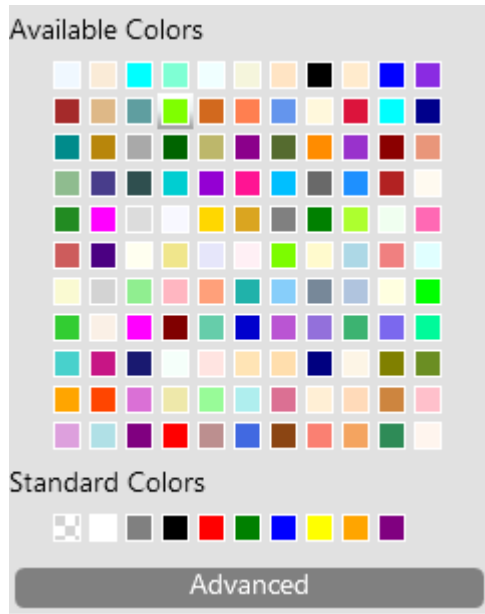
The following table describes all the editable properties of the Text List.

Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Rotation</b>	Determines the rotation degrees assigned to the object
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>Disabled</b>	Determines whether the user can interact with the object. This property is ignored when "UserMode" is not in read-write mode
<b>TextList</b>	List of available texts for the current field being edited
<b>Tag</b>	Tags associated with the value of the Text List
<b>UserMode</b>	Determines whether the user can interact with the object: - read only, the user cannot interact with the object; this value takes precedence over the Disabled property - read-write, the user can interact with the object; the effective interaction is validated by the Disabled property
<b>RollEditing</b>	It allows to change the sequence displaying method of images / texts
<b>AnimationTime</b>	Time range (in milliseconds) between the displaying of a text and the next of the list of available ones (TextList)
<b>Attributes</b>	
<b>FontName</b>	Text Font displayed on the text list
<b>StrokeColor</b>	Determines the fill colour of the object; the colour is selectable by RGB colour code or colour palette

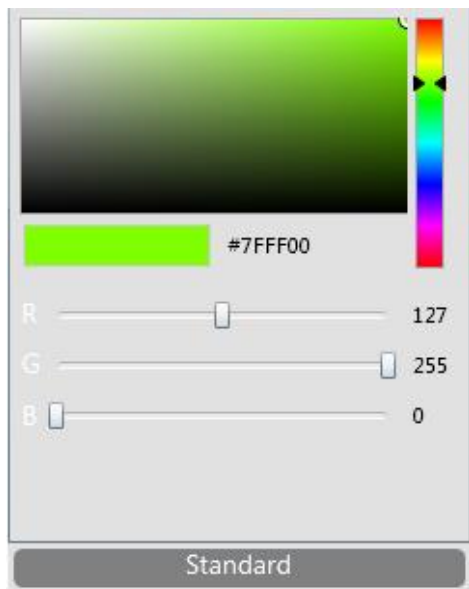
<b>Opacity</b>	Determines the opacity of the object
<b>FontSize</b>	Determines the size of the font of the object
<b>FontStyle</b>	Font style. Any combination of the following features: - None - Italics - Bold - Underline
<b>Multiline</b>	If the text automatically starts a newline, it determines the number of lines needed with respect to the current width
<b>HorizontalAlign</b>	This function allows to position the image horizontally (Centred, Right or Left)
<b>VerticalAlign</b>	This function allows to position the image vertically (Centred, Top or Bottom)
<b>Border (default / Off)</b>	
<b>Visible</b>	Determines whether the border should be visible or not
<b>Border Color</b>	Determines the color of the border
<b>Border Size</b>	Determines the border thickness
<b>Area Color</b>	Determines the area fill color
<b>PopupOpacity</b>	Determines the opacity of the object
<b>Corner</b>	Determines the roundness of the angles
<b>Blink</b>	
<b>CanBlink</b>	It allows you to enable or disable the Runtime blink
<b>Visibility</b>	Allows to display or hide the Runtime blink
<b>Border</b>	Determines the blinking or less of the border
<b>BorderColor</b>	Determines the color of the border during blinking
<b>Area</b>	Determines the blinking or less of the area
<b>AreaColor</b>	Determines the area fill color during blinking
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project

# CREW Manual

The properties related to colours can be edited through the colour palette.



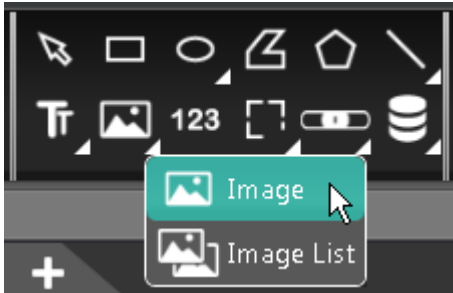
Click “Advanced” to select a colour using the RGB colour selection mask.





# CREW Manual

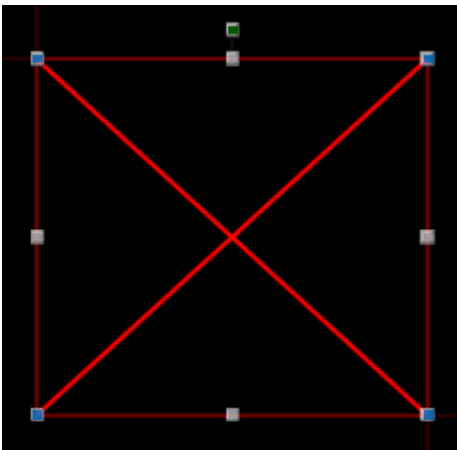
## Image



Crew offers the possibility of importing images contained on the programmer's machine to the project. For import, the images need to be saved in the most common graphic formats (PNG, JPEG, BITMAP, GIF).

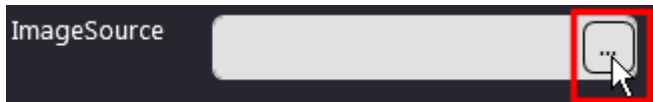
When a new project is created, Crew enters some default images intended for specific uses (alarm display, Pen Trend, etc.) which, however, can also be used in the project for other purposes.

The "Image Field" icon in the "Graphics" menu is used to add an image field to a page, drawing the size of it on that page.

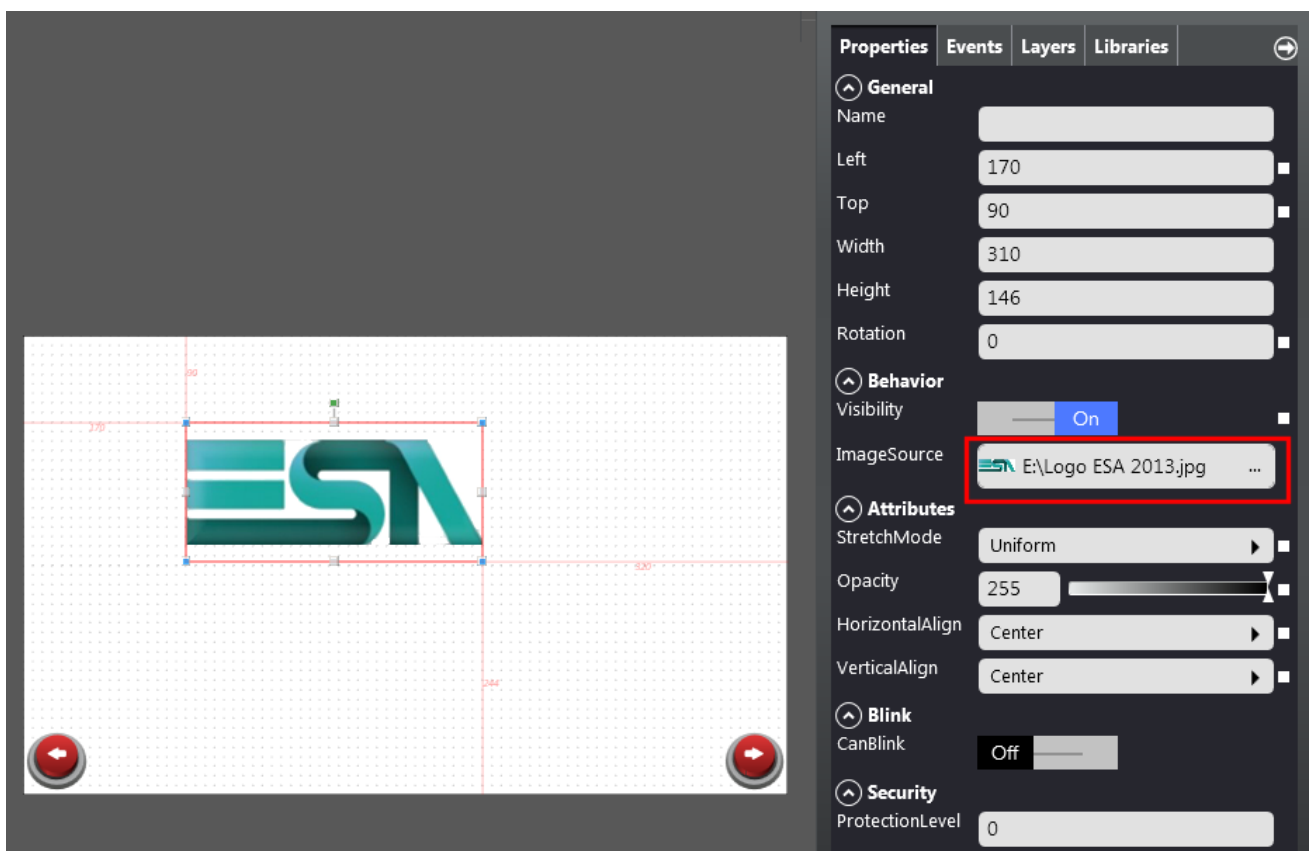


# CREW Manual

Click the "ImageSource" option to import an image to the image field.



Select the image to be added and confirm with "Ok".

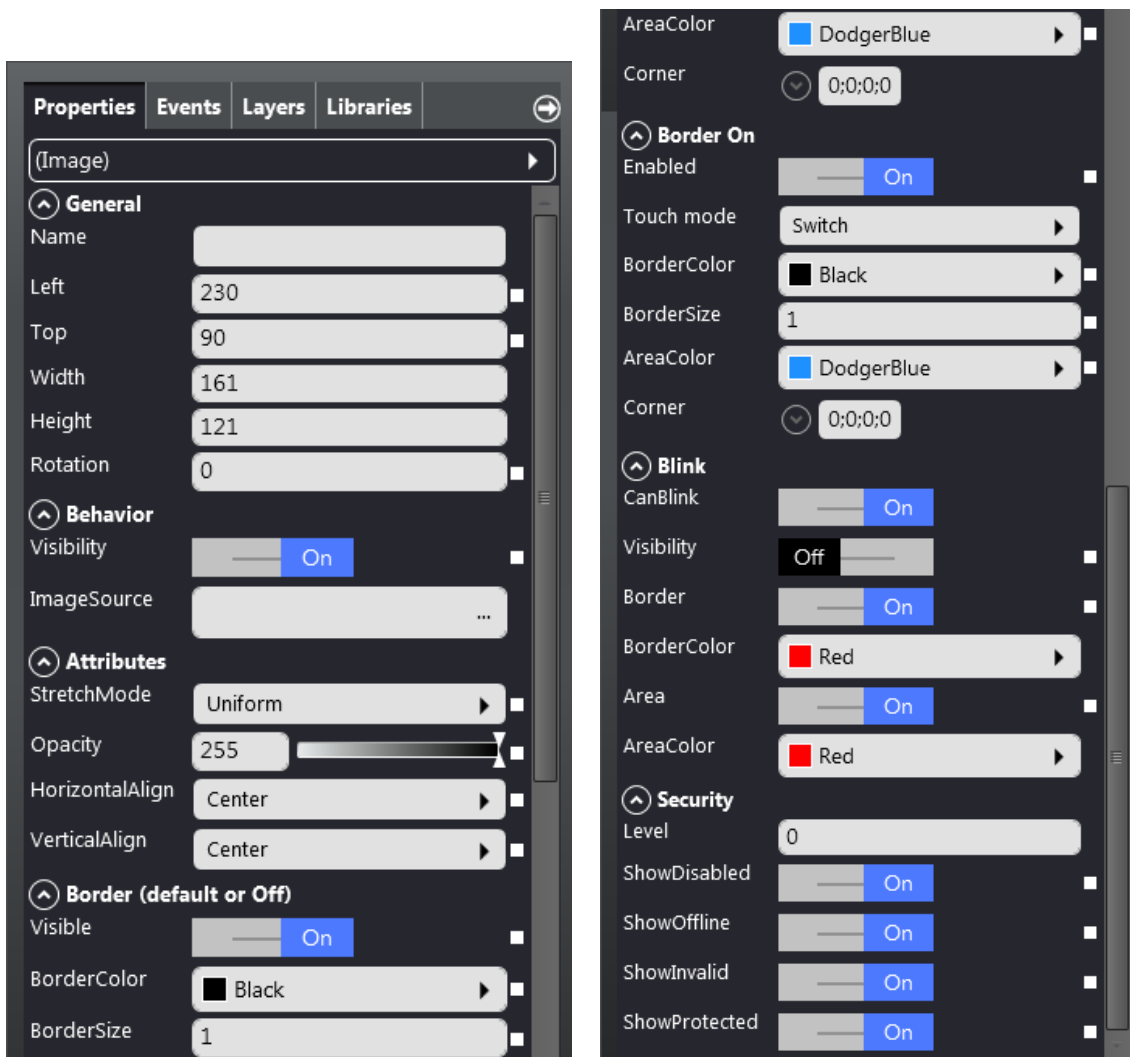


To determine the features of the "Image", set them in the "Properties Editor", as shown in the section "[Image Properties](#)".

# CREW Manual

## Image Properties

The following image illustrates all the editable properties of the “Image”. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

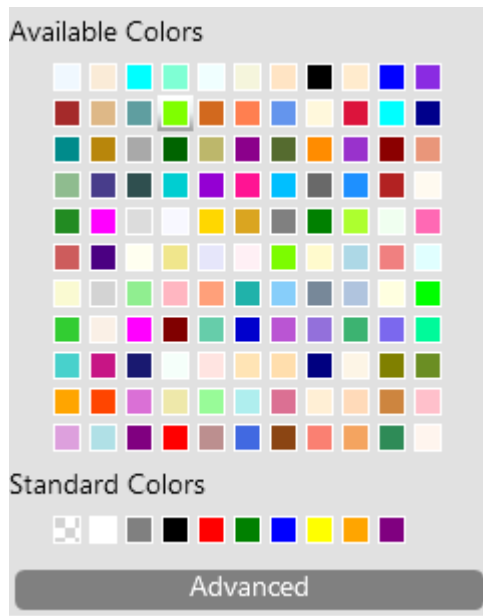
The following table describes all the editable properties of the Image object.

Properties	Description
<b>Security</b>	
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Rotation</b>	Determines the rotation degrees assigned to the object
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>ImageSource</b>	Indica il percorso dal quale viene importata l'immagine da inserire
<b>Attributes</b>	
<b>StretchMode</b>	Resize grouped elements maintaining the aspect
<b>Opacity</b>	Determines the opacity of the object
<b>HorizontalAlign</b>	This function allows to position the image horizontally (Centred, Right or Left)
<b>VerticalAlign</b>	This function allows to position the image vertically (Centred, Top or Bottom)
<b>Border (default / Off)</b>	
<b>Visible</b>	Determines whether the border should be visible or not
<b>Border Color</b>	Determines the color of the border
<b>Border Size</b>	Determines the border thickness
<b>Area Color</b>	Determines the area fill color
<b>Corner</b>	Determines the roundness of the angles

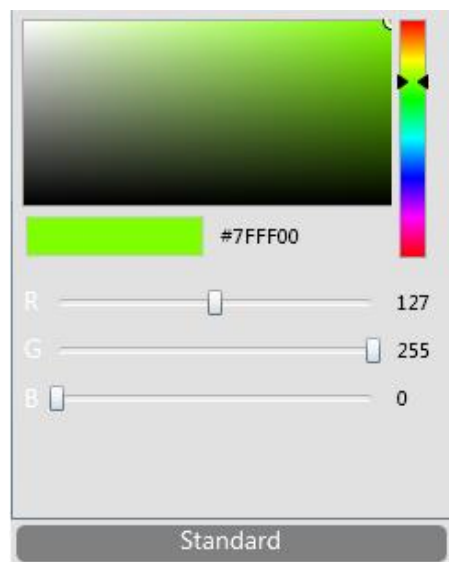
<b>Border On</b>	
<b>Enabled</b>	Determines whether the user can interact with the object
<b>TouchMode</b>	Determines the mode of interaction of the button: - SWITCH: The button acts as a normal button switch with on-pressed and on-released events generated when the user has released the button. - CONTACT: the button enters the "pressed" state as soon as the user touches the screen, and enters the state "released" when there is no more contact with the screen. - BUTTON: It is not necessary to associate a variable to the change of the button state, events associated to this are used
<b>BorderColor</b>	Determines the color of the border
<b>BorderSize</b>	Determines the border thickness
<b>AreaColor</b>	Determines the area fill color
<b>Corner</b>	Determines the roundness of the angles
<b>Blink</b>	
<b>CanBlink</b>	It allows you to enable or disable the Runtime blink
<b>Visibility</b>	Allows to display or hide the Runtime blink
<b>Border</b>	Determines the blinking or less of the border
<b>BorderColor</b>	Determines the color of the border during blinking
<b>Area</b>	Determines the blinking or less of the area
<b>AreaColor</b>	Determines the area fill color during blinking
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project
<b>ShowDisabled</b>	Enables the displaying of a "status" icon in the object when its use has been disabled
<b>ShowOffline</b>	Enables the displaying of a "status" icon in the object when it is in the offline condition
<b>ShowInvalid</b>	Enables the displaying of a "status" icon in the object when its current value is invalid
<b>ShowProtected</b>	Enables the displaying of a "status" icon in the object when it is protected for the current user

# CREW Manual

The properties related to colours can be edited through the colour palette.

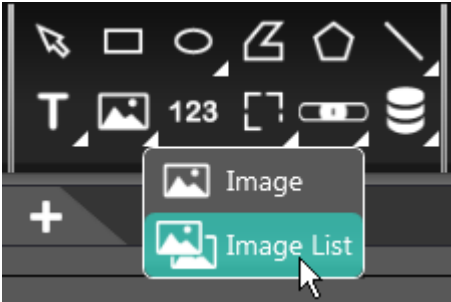


Click "Advanced" to select a colour using the RGB colour selection mask.

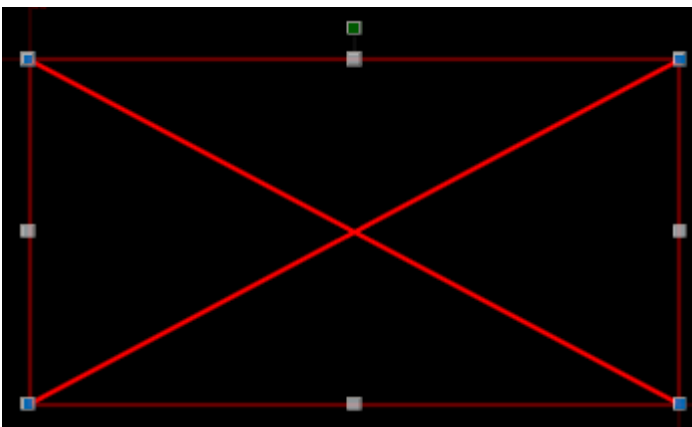


# CREW Manual

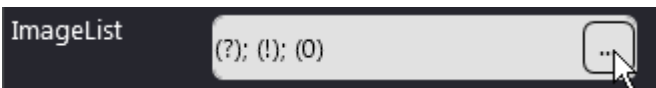
## Image list



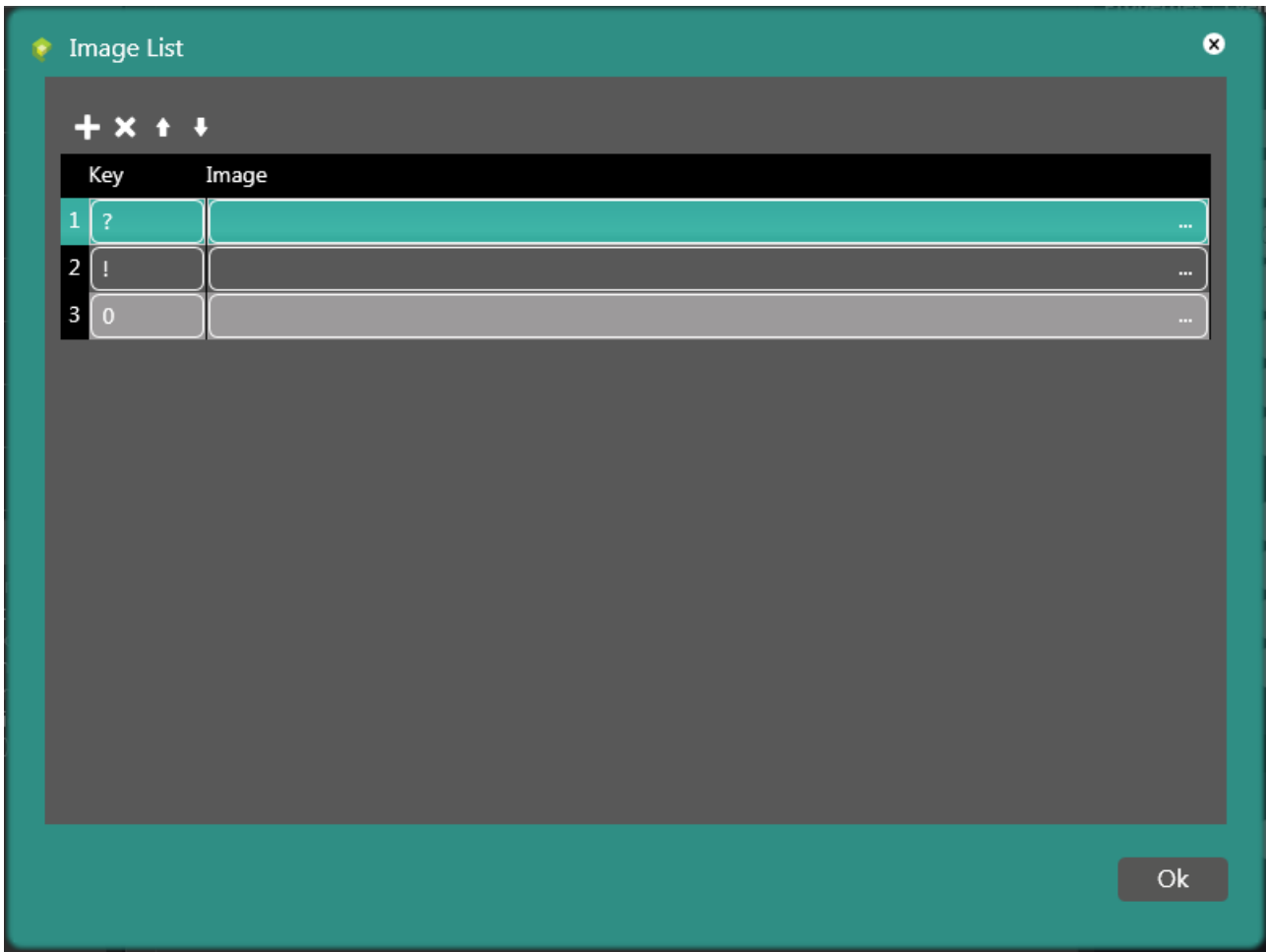
The "Image List" icon in the "Graphics" menu is used to add an image list to a page, drawing the size of it on that page.



Press the appropriate key to open a window listing the default images that can be linked to the object.

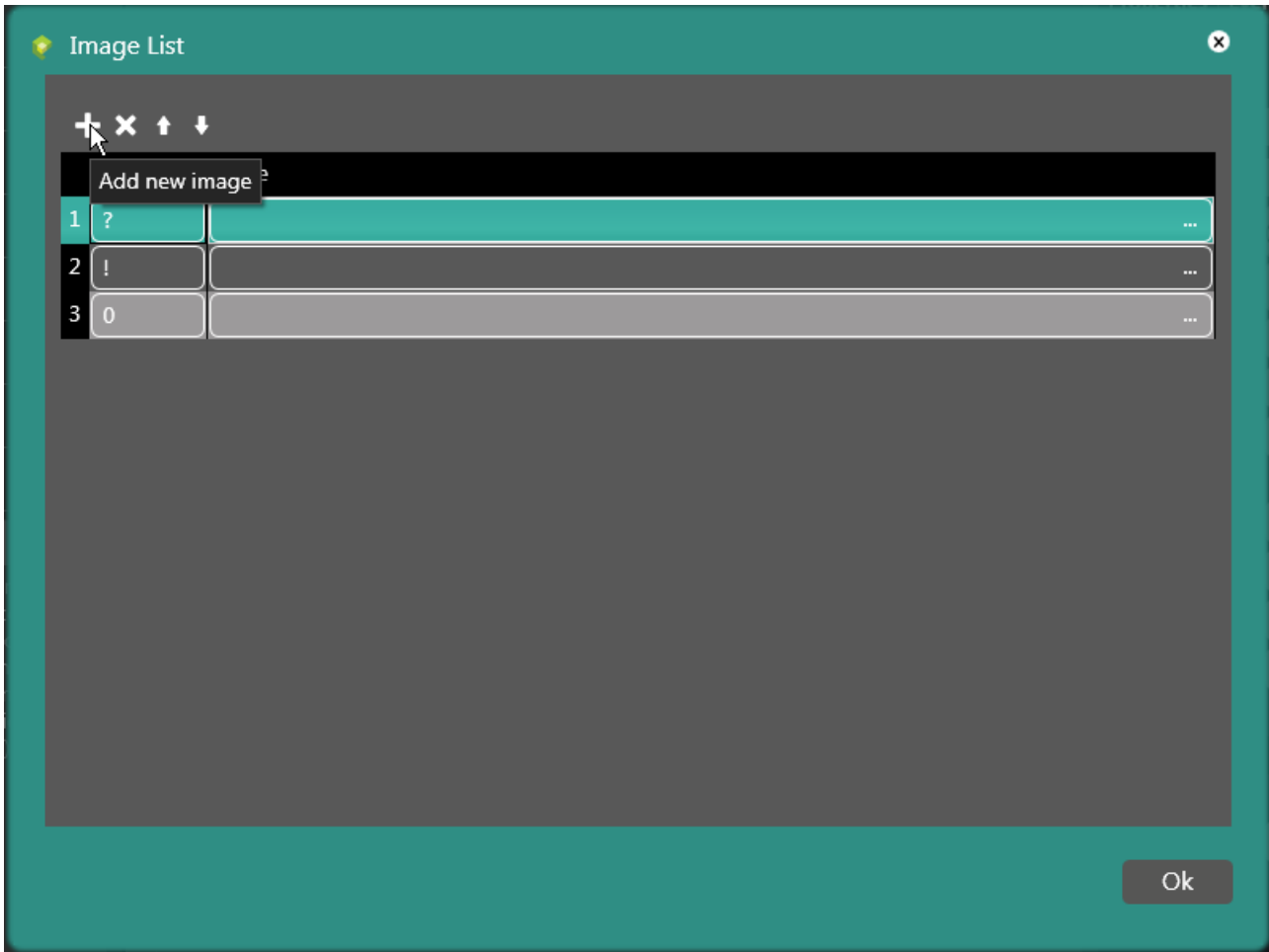


# CREW Manual



# CREW Manual

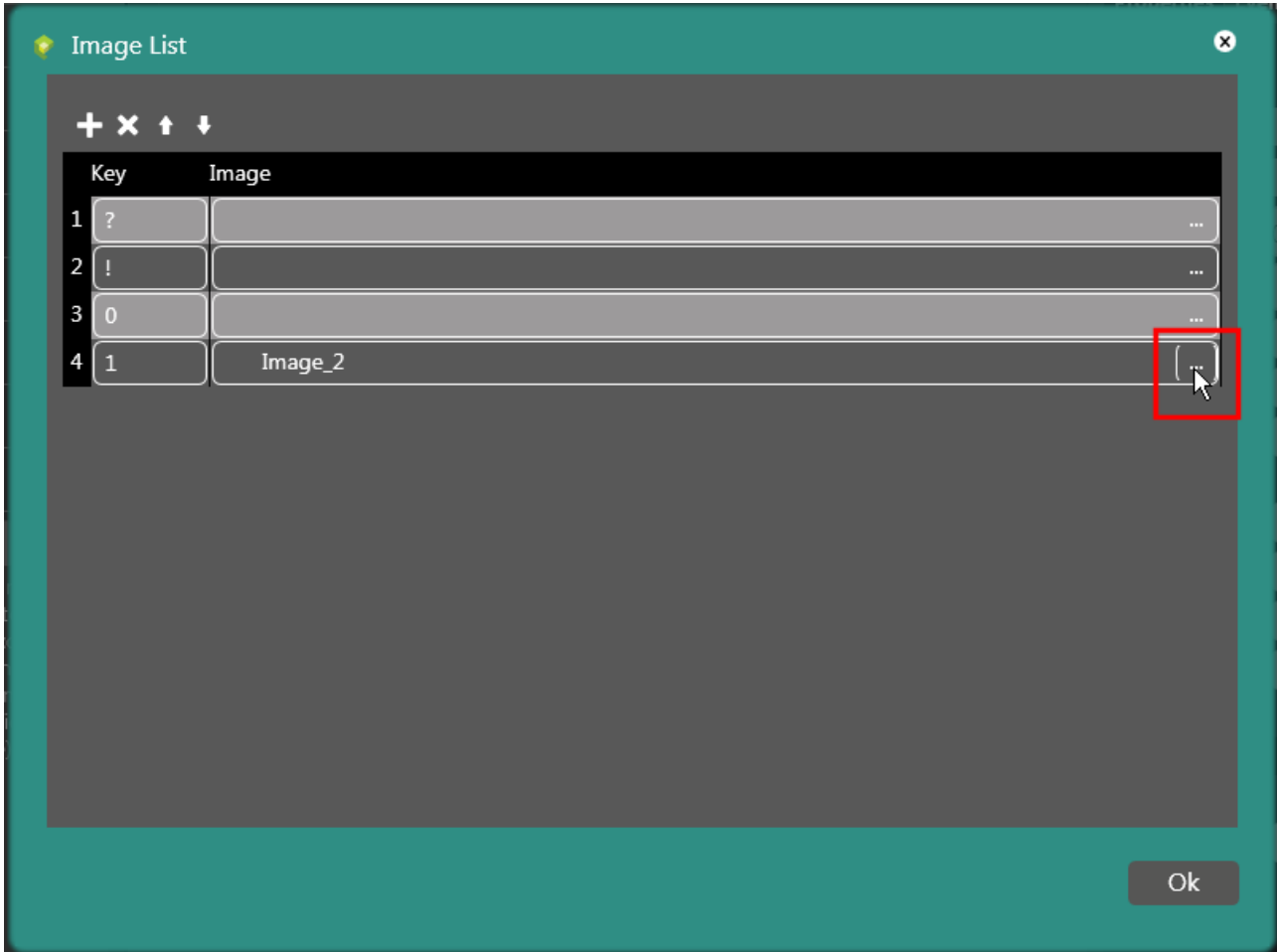
click the “+” icon to add new images to the list.





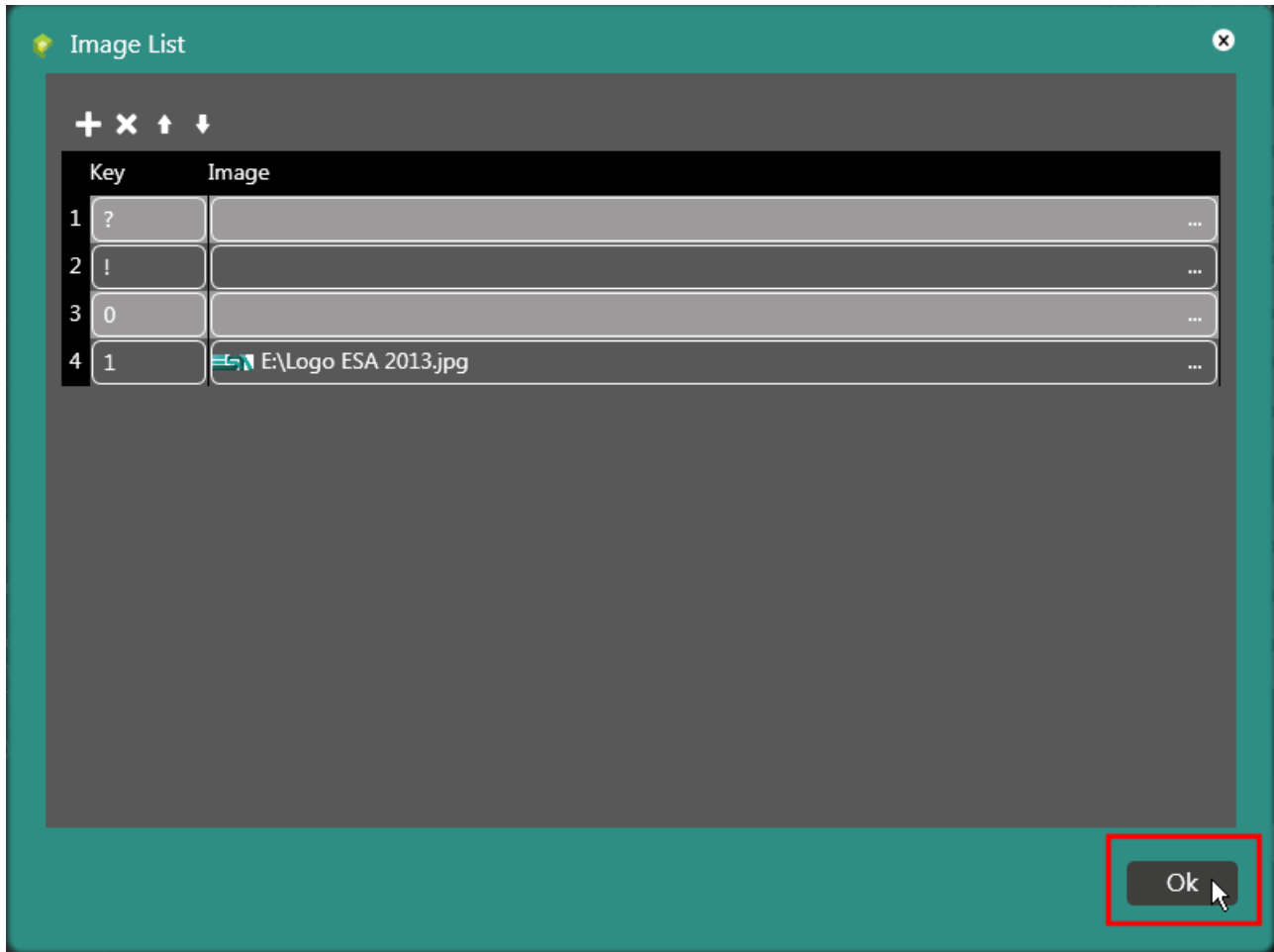
# CREW Manual

Click the “Browse” key.



# CREW Manual

Select the image and confirm with "Ok".

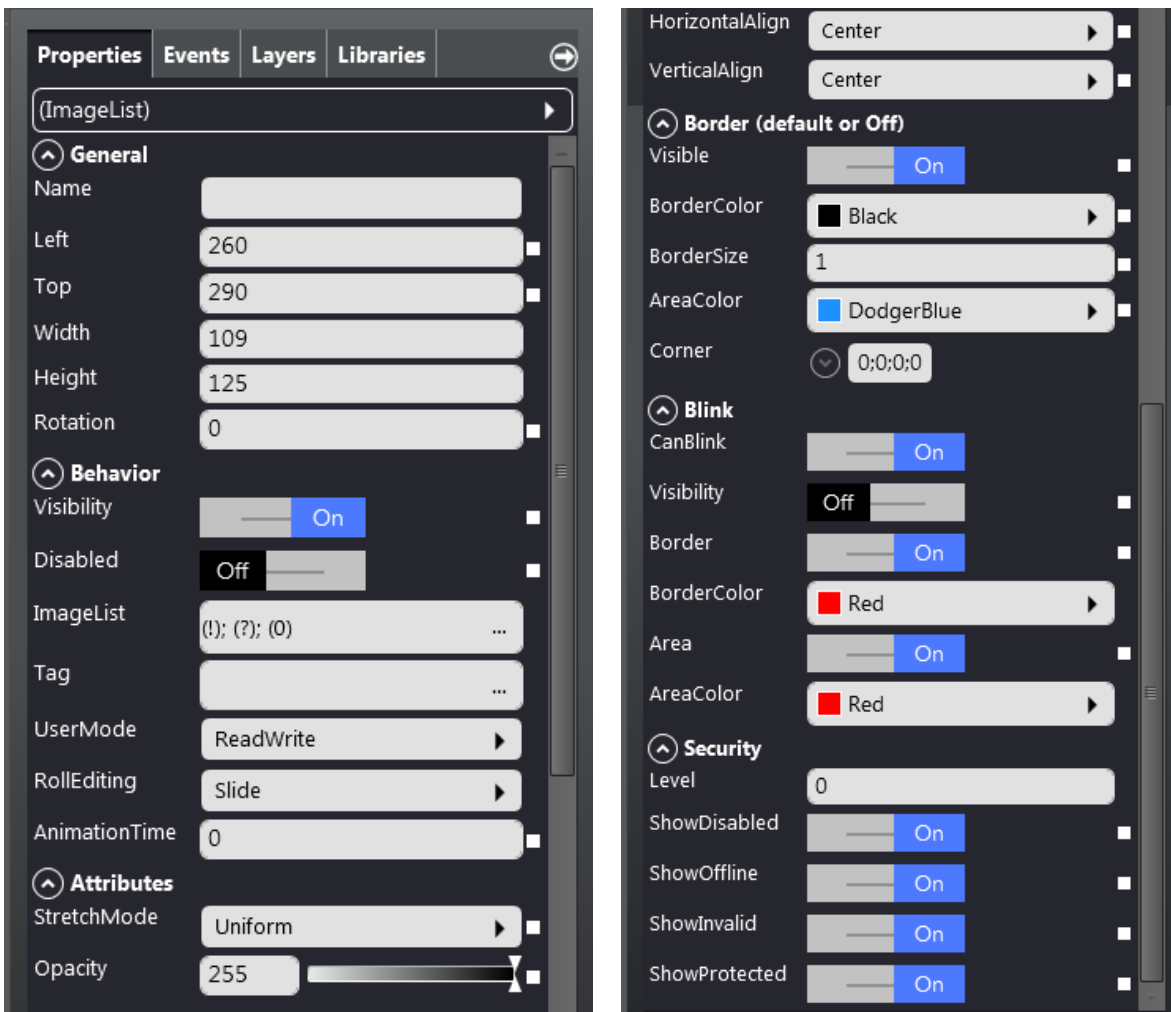


To determine the features of the "Image List" field, set them in the "Properties Editor", as shown in the section "Image List Properties".

# CREW Manual

## Image List Properties

The following image illustrates all the editable properties of the Image List. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

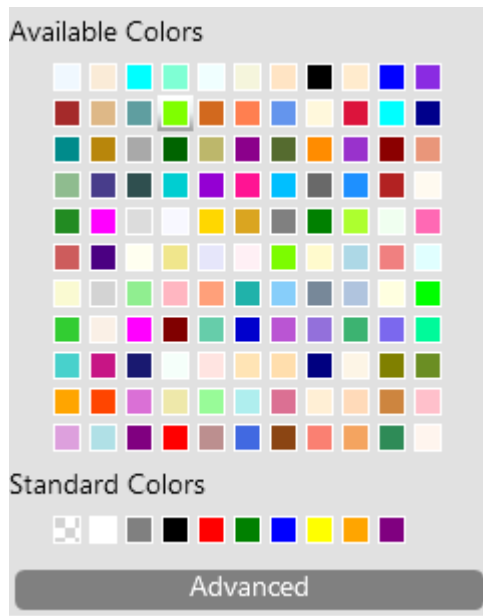
The following table describes all the editable properties of the Image List.

Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Rotation</b>	Determines the rotation degrees assigned to the object
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>Disabled</b>	Determines whether the user can interact with the object. This property is ignored when "UserMode" is not in read-write mode
<b>ImageList</b>	Indicates the image list from which the image to be displayed is selected at Runtime. By selecting the corresponding field you can add new images
<b>Tag</b>	Tags associated with the value of the Images List
<b>UserMode</b>	Determines whether the user can interact with the object: - read only, the user cannot interact with the object; this value takes precedence over the Disabled property - read-write, the user can interact with the object; the effective interaction is validated by the Disabled property
<b>RollEditing</b>	It allows to change the sequence displaying method of images / texts
<b>AnimationTime</b>	Time range (in milliseconds) between the displaying of a text and the next of the list of available ones (TextList)
<b>Attributes</b>	
<b>StrechMode</b>	Resize grouped elements maintaining the aspect
<b>Opacity</b>	Determines the opacity of the object

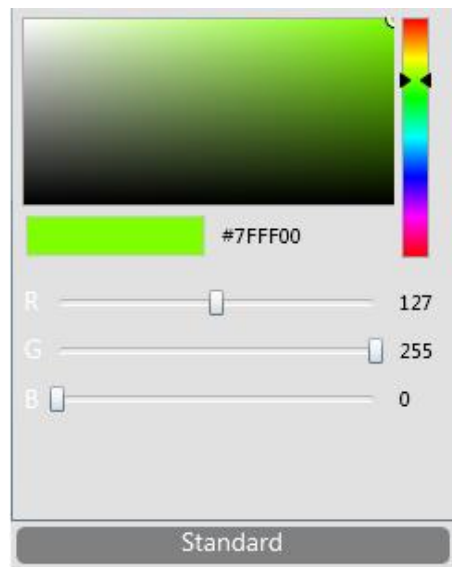
<b>HorizontalAlign</b>	This function allows to position the image horizontally (Centred, Right or Left)
<b>VerticalAlign</b>	This function allows to position the image vertically (Centred, Top or Bottom)
<b>Border (default / Off)</b>	
<b>Visible</b>	Determines whether the border should be visible or not
<b>Border Color</b>	Determines the color of the border
<b>Border Size</b>	Determines the border thickness
<b>Area Color</b>	Determines the area fill color
<b>Corner</b>	Determines the roundness of the angles
<b>Blink</b>	
<b>CanBlink</b>	It allows you to enable or disable the Runtime blink
<b>Visibility</b>	Allows to display or hide the Runtime blink
<b>Border</b>	Determines the blinking or less of the border
<b>BorderColor</b>	Determines the color of the border during blinking
<b>Area</b>	Determines the blinking or less of the area
<b>AreaColor</b>	Determines the area fill color during blinking
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project
<b>ShowDisabled</b>	Enables the displaying of a "status" icon in the object when its use has been disabled
<b>ShowOffline</b>	Enables the displaying of a "status" icon in the object when it is in the offline condition
<b>ShowInvalid</b>	Enables the displaying of a "status" icon in the object when its current value is invalid
<b>ShowProtected</b>	Enables the displaying of a "status" icon in the object when it is protected for the current user

# CREW Manual

The properties related to colours can be edited through the colour palette.

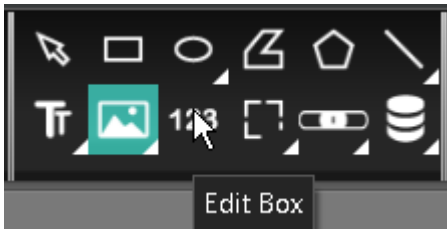


Click "Advanced" to select a colour using the RGB colour selection mask.

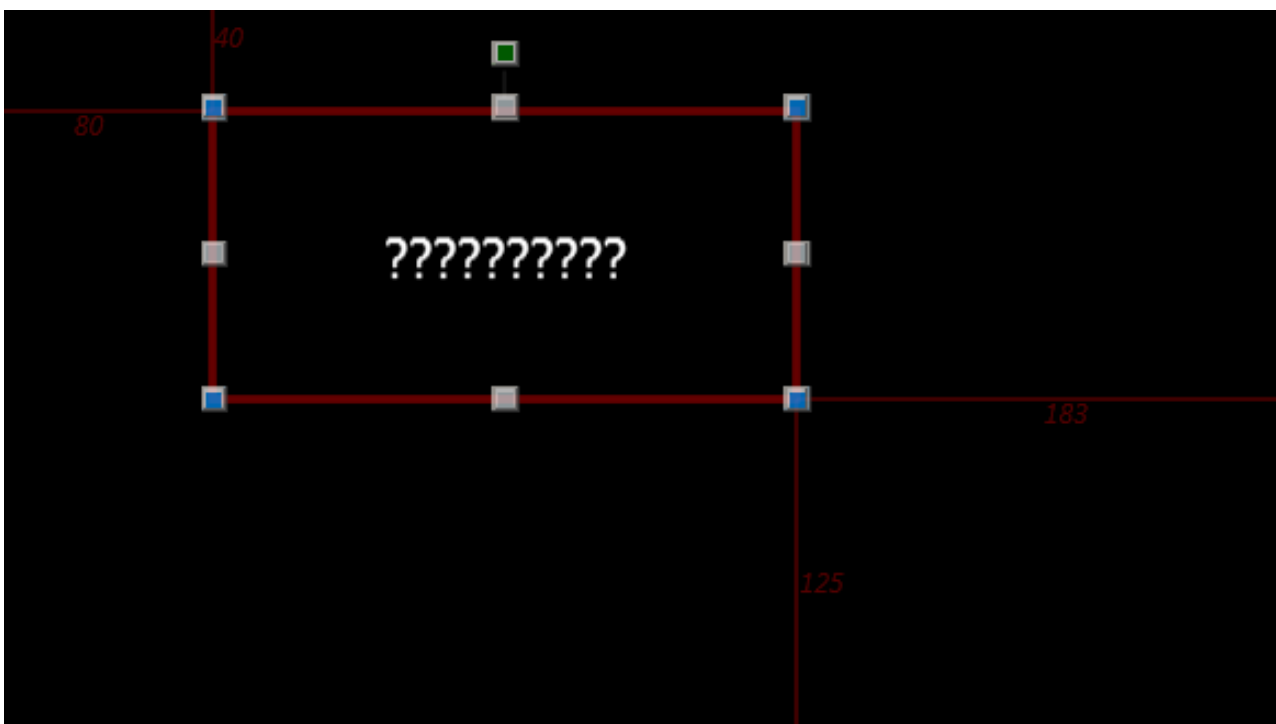


# CREW Manual

Edit Box - Numerical field -



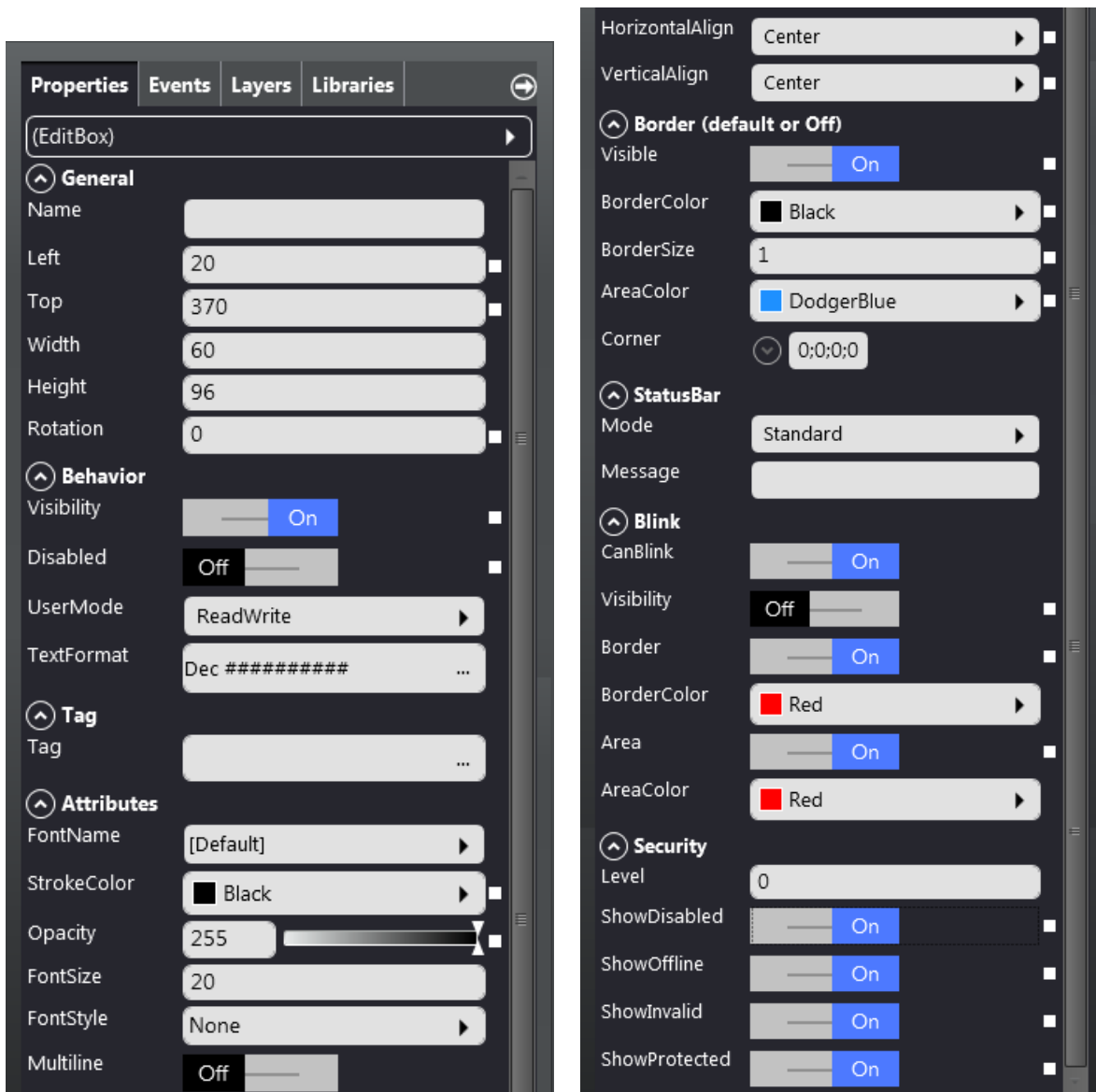
The "Edit Box" (or "Numerical field") icon in the "Graphics" menu is used to add an edit box to a page, drawing the size of it on that page.



# CREW Manual

## Edit Box Properties

The following image illustrates all the editable properties of the Edit Box. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

The following table describes all the editable properties of the Edit Box.

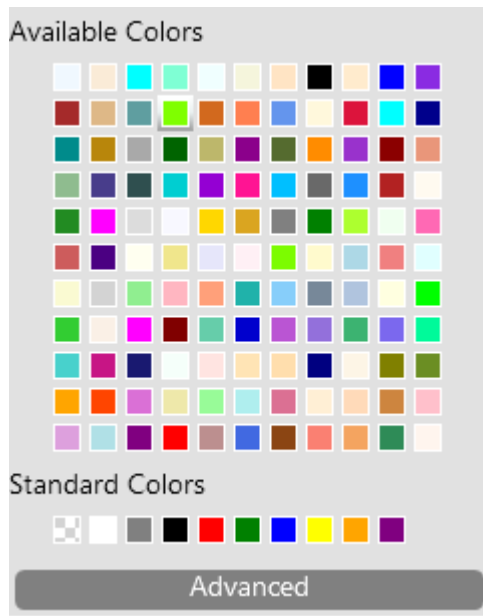
Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Rotation</b>	Determines the rotation degrees assigned to the object
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>Disabled</b>	Determines whether the user can interact with the object. This property is ignored when "UserMode" is not in read-write mode
<b>UserMode</b>	Determines whether the user can interact with the object: - read only, the user cannot interact with the object; this value takes precedence over the Disabled property - read-write, the user can interact with the object; the effective interaction is validated by the Disabled property
<b>TextFormat</b>	You can access all parameters of the object
<b>Tag</b>	
<b>Tag</b>	Tag associated with the value of the object
<b>Attributes</b>	
<b>FontName</b>	Font for the text shown in the object
<b>StrokeColor</b>	Determines the fill colour of the object; the colour is selectable by RGB colour code or colour palette
<b>Opacity</b>	Determines the opacity of the object
<b>FontSize</b>	Determines the size of the font of the object
<b>FontStyle</b>	Font style. Any combination of the following features: - None - Italics - Bold - Underline

<b>Multiline</b>	If the text automatically starts a newline, it determines the number of lines needed with respect to the current width
<b>HorizontalAlign</b>	This function allows to position the image horizontally (Centred, Right or Left)
<b>VerticalAlign</b>	This function allows to position the image vertically (Centred, Top or Bottom)
<b>Border (default / Off)</b>	
<b>Visible</b>	Determines whether the border should be visible or not
<b>Border Color</b>	Determines the color of the border
<b>Border Size</b>	Determines the border thickness
<b>Area Color</b>	Determines the area fill color
<b>Corner</b>	Determines the roundness of the angles
<b>StatusBar</b>	
<b>Mode</b>	It defines the behavior of the edit box keyboard Status Bar and the formatting type of the message
<b>Message</b>	It defines the text of the status message in the case of "custom bar" type
<b>Blink</b>	
<b>CanBlink</b>	It allows you to enable or disable the Runtime blink
<b>Visibility</b>	Allows to display or hide the Runtime blink
<b>Border</b>	Determines the blinking or less of the border
<b>BorderColor</b>	Determines the color of the border during blinking
<b>Area</b>	Determines the blinking or less of the area
<b>AreaColor</b>	Determines the area fill color during blinking
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project
<b>ShowDisabled</b>	Enables the displaying of a "status" icon in the object when its use has been disabled
<b>ShowOffline</b>	Enables the displaying of a "status" icon in the object when it is in the offline condition
<b>ShowInvalid</b>	Enables the displaying of a "status" icon in the object when its current value is invalid
<b>ShowProtected</b>	Enables the displaying of a "status" icon in the object when it is protected for the current user

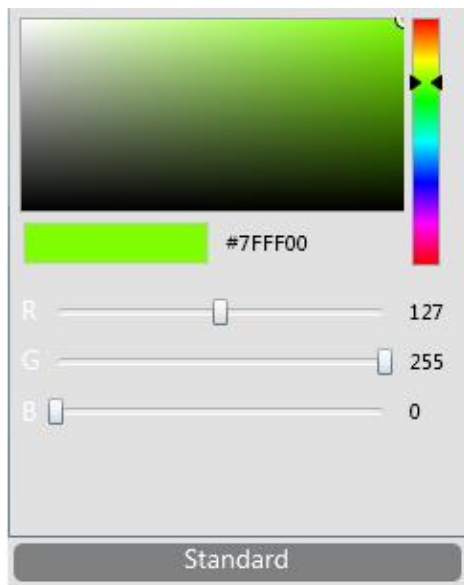


# CREW Manual

The properties related to colours can be edited through the colour palette.

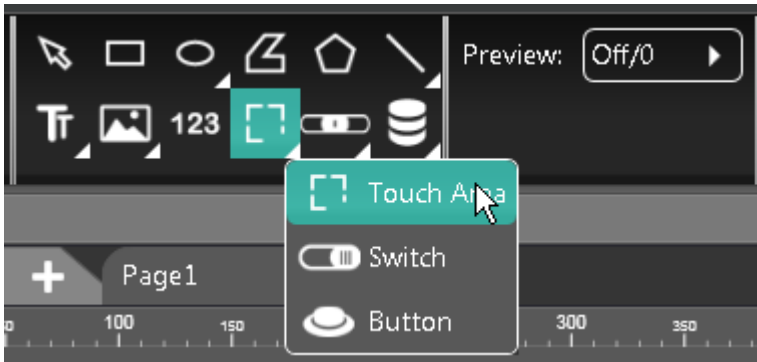


Click “Advanced” to select a colour using the RGB colour selection mask.

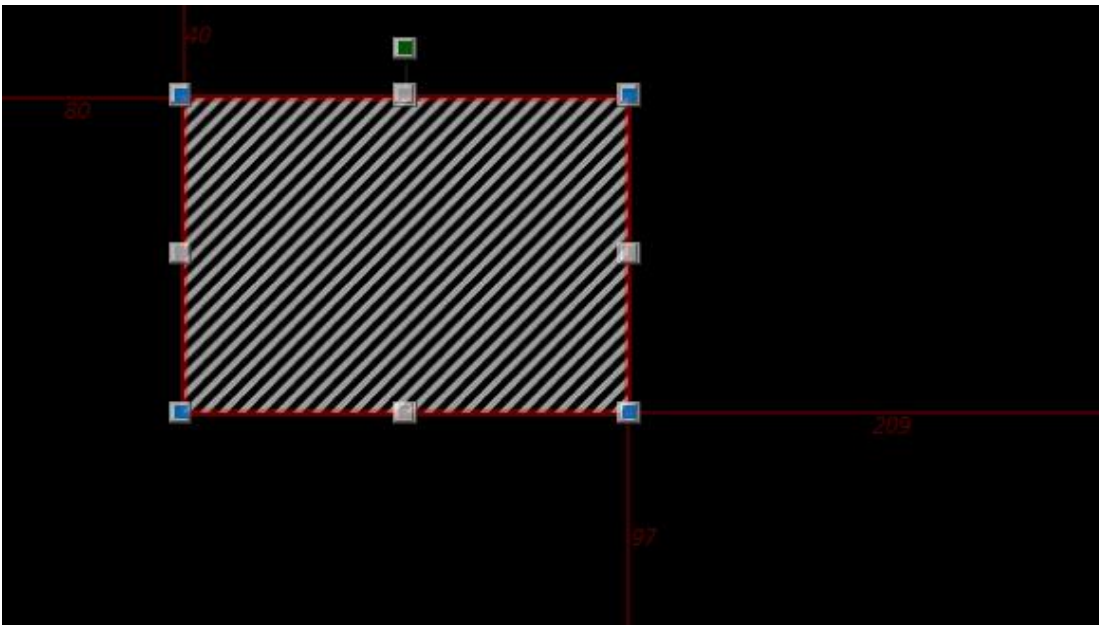


# CREW Manual

## Sensitive area



The "Sensitive area" icon on the "Graphics" menu is used to add a sensitive area on the page, drawing it with the mouse on the page.

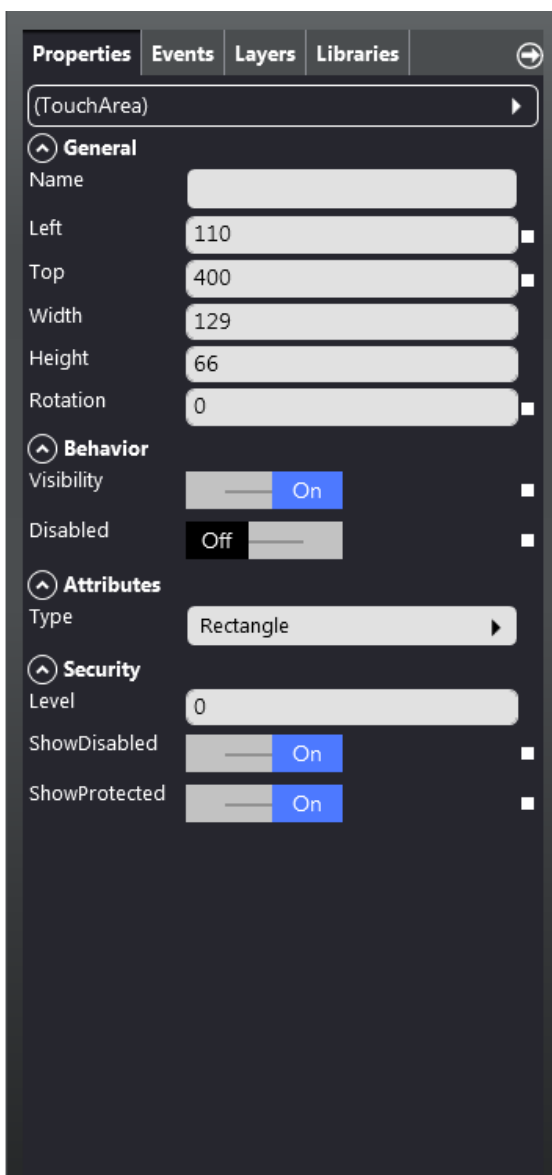


To determine the features of the "Sensitive area", set them in the "Properties Editor", as shown in the section "[Sensitive Area Properties](#)".

# CREW Manual

## Sensitive Area Properties

The following image illustrates all the editable properties of the Sensitive Area. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



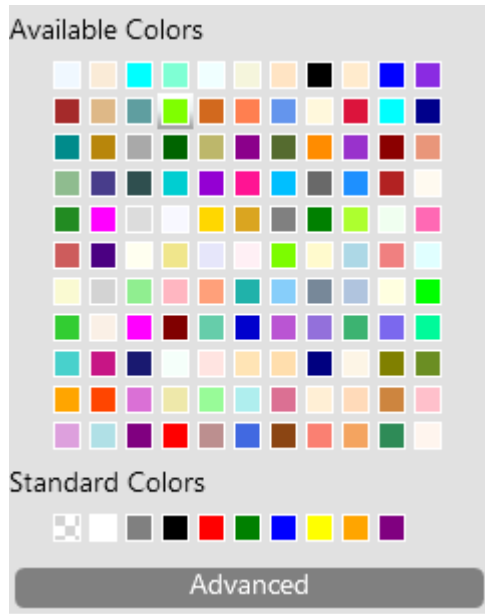
# CREW Manual

The following table describes all the editable properties of the Sensitive Area.

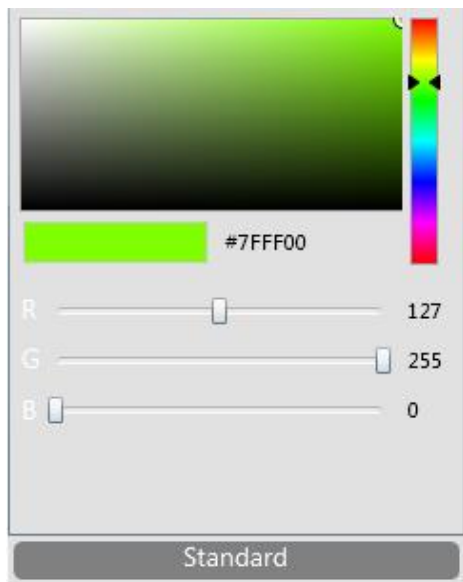
Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Rotation</b>	Determines the rotation degrees assigned to the object
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>Disabled</b>	Determines whether the user can interact with the object. This property is ignored when "UserMode" is not in read-write mode
<b>Attributes</b>	
<b>Type</b>	It defines the shape of the "Touch area"
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project
<b>ShowDisabled</b>	Enables the displaying of a "status" icon in the object when its use has been disabled
<b>ShowProtected</b>	Enables the displaying of a "status" icon in the object when it is protected for the current user

# CREW Manual

The properties related to colours can be edited through the colour palette.

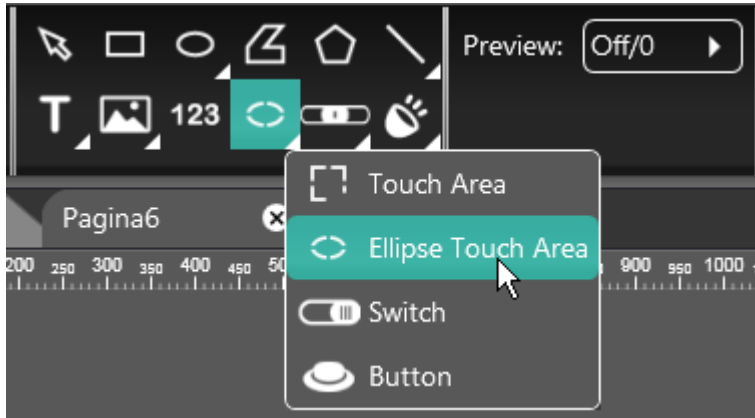


Click “Advanced” to select a colour using the RGB colour selection mask.

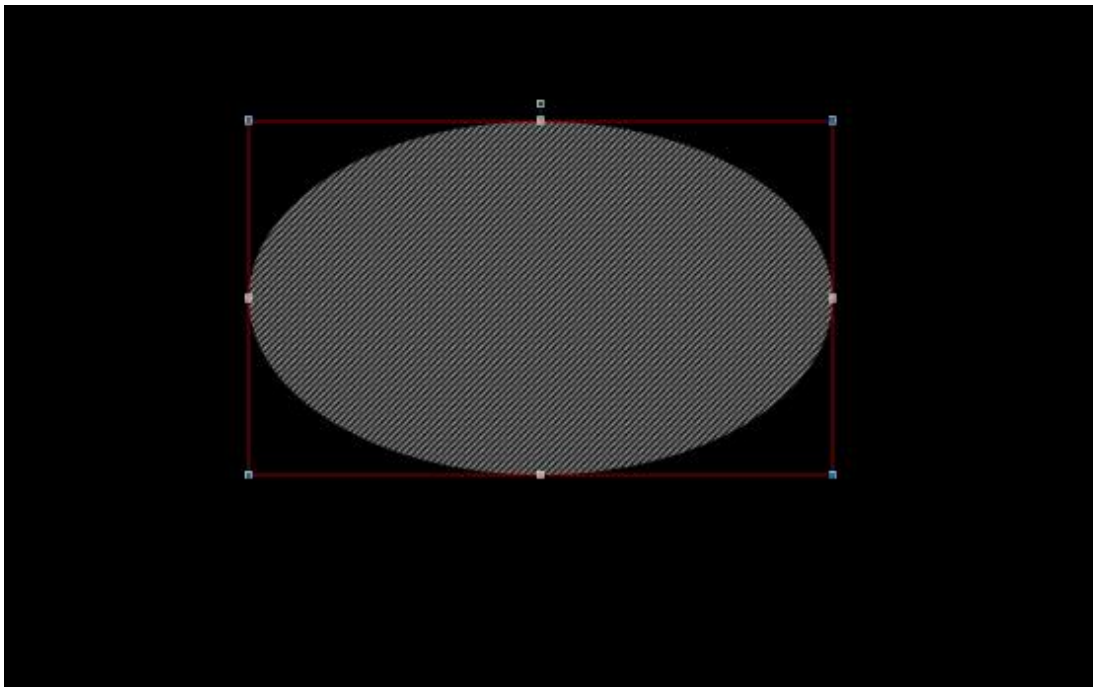


# CREW Manual

## Sensitive area - Ellipse -



The “Sensitive area - Ellipse -” icon on the “Graphics” menu is used to add an elliptical sensitive area on the page, drawing it with the mouse on the page.

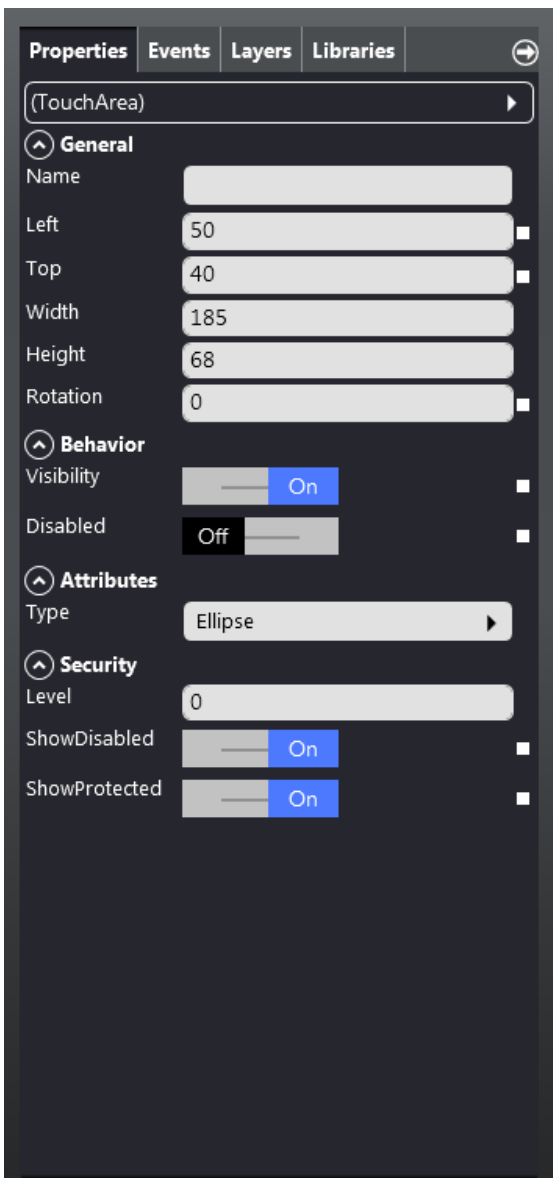


To determine the features of the "Sensitive area", set them in the "Properties Editor", as shown in the section "[Sensitive area - Ellipse - Properties](#)".

# CREW Manual

## Sensitive area - Ellipse - Properties

The following image illustrates all the editable properties of the Sensitive area - Ellipse -. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

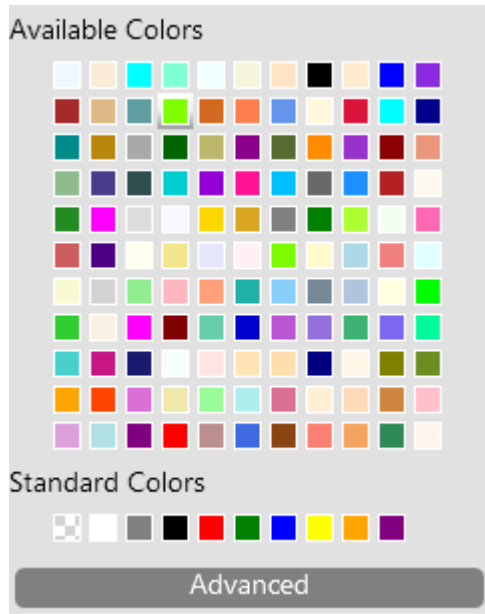
The following table describes all the editable properties of the Sensitive area - Ellipse -.

Properties	Descrizione
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Rotation</b>	Determines the rotation degrees assigned to the object
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>Disabled</b>	Determines whether the user can interact with the object. This property is ignored when "UserMode" is not in read-write mode
<b>Attributes</b>	
<b>Type</b>	It defines the shape of the "Touch area"
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project
<b>ShowDisabled</b>	Enables the displaying of a "status" icon in the object when its use has been disabled
<b>ShowProtected</b>	Enables the displaying of a "status" icon in the object when it is protected for the current user

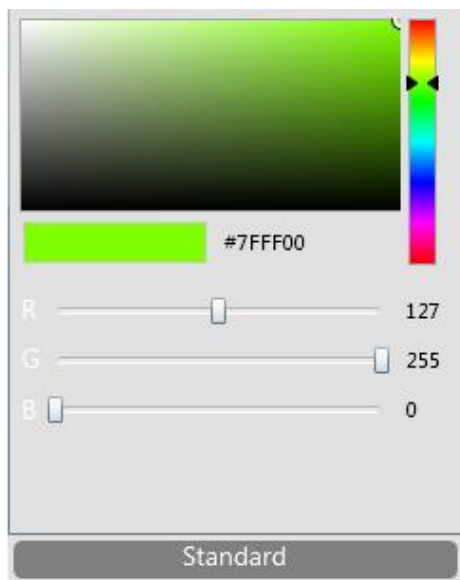


# CREW Manual

The properties related to colours can be edited through the colour palette.

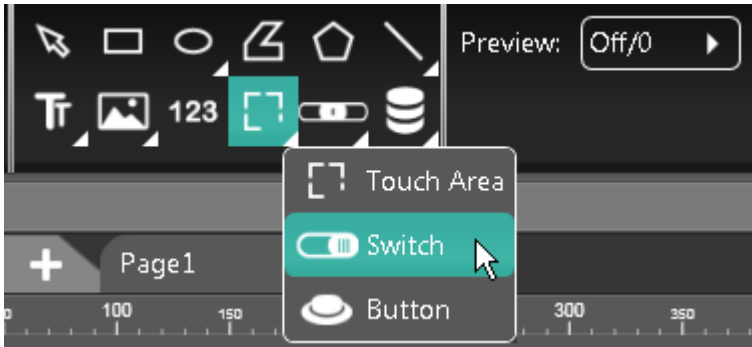


Click “Advanced” to select a colour using the RGB colour selection mask.

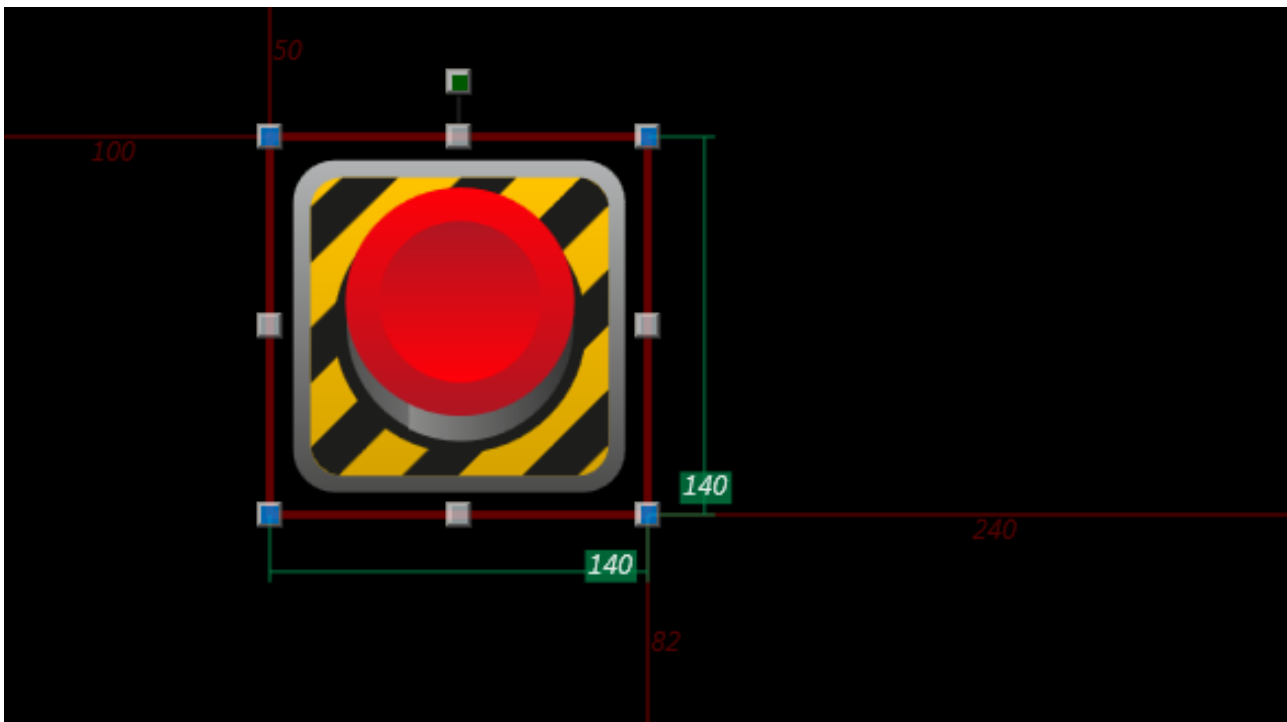


# CREW Manual

## Switch



The “Switch” icon on the “Graphics” menu is used to place a switch on the page, drawing it with the mouse on the page.



To determine the features of the “Switch” field it is necessary to set them in the “Properties Editor”, as shown in the "[Switch Properties](#)" section.

# CREW Manual

## Switch Properties

The following image illustrates all the editable properties of the Switch. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

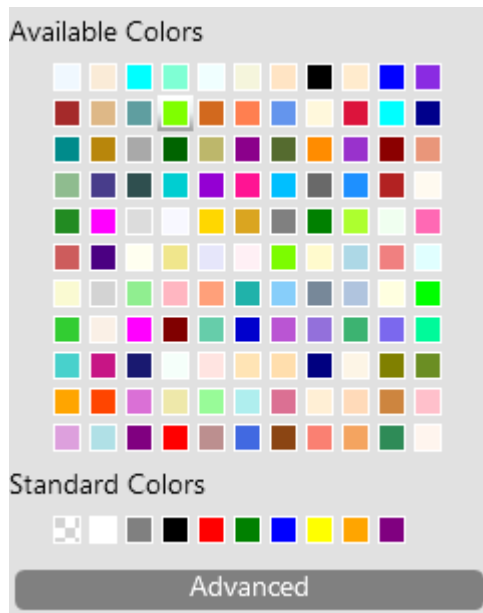
The following table describes all the editable properties of the Switch.

Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Rotation</b>	Determines the rotation degrees assigned to the object
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>Disabled</b>	Determines whether the user can interact with the object. This property is ignored when "UserMode" is not in read-write mode
<b>UserMode</b>	Determines whether the user can interact with the object: - read only, the user cannot interact with the object; this value takes precedence over the Disabled property - read-write, the user can interact with the object; the effective interaction is validated by the Disabled property
<b>TouchMode</b>	Determines the mode of interaction of the button: - SWITCH: The button acts as a normal button switch with on-pressed and on-released events generated when the user has released the button. - CONTACT: the button enters the "pressed" state as soon as the user touches the screen, and enters the state "released" when there is no more contact with the screen. - BUTTON: It is not necessary to associate a variable to the change of the button state, events associated to this are used
<b>Tag</b>	
<b>Tag</b>	Tag linked to the object
<b>ValueOn</b>	Represents the value of the "switch-on" threshold (default 1)
<b>ValueOff</b>	Represents the value of the "switch-off" threshold (default 1)

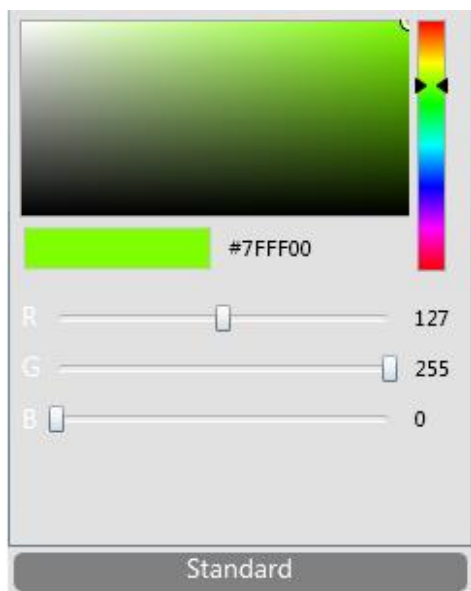
<b>Attributes</b>	
<b>Opacity</b>	Determines the opacity of the object
<b>On</b>	Allows you to assign the color of the object when it is "ON" using the color bar (Hue)
<b>SaturationOn</b>	It allows you to change the saturation of the color of the object in the "ON" state
<b>BrightnessOn</b>	It allows you to change the brightness of the color of the object in the "ON" state
<b>Off</b>	Allows you to assign the color of the object when it is "OFF" using the color bar (Hue)
<b>SaturationOff</b>	It allows you to change the saturation of the color of the object in the "OFF" state
<b>BrightnessOff</b>	It allows you to change the brightness of the color of the object in the "OFF" state
<b>LibraryVersion</b>	Allows you to change the default color of the object by choosing from the "Colors", "Black & White" and "Color Art" (Only for few switches in the library)
<b>Blink</b>	
<b>CanBlink</b>	It allows you to enable or disable the Runtime blink
<b>Visibility</b>	The "Visibility" property allows to display or hide the Runtime blink
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project
<b>ShowDisabled</b>	Enables the displaying of a "status" icon in the object when its use has been disabled
<b>ShowOffline</b>	Enables the displaying of a "status" icon in the object when it is in the offline condition
<b>ShowInvalid</b>	Enables the displaying of a "status" icon in the object when its current value is invalid
<b>ShowProtected</b>	Enables the displaying of a "status" icon in the object when it is protected for the current user

# CREW Manual

The properties related to colours can be edited through the colour palette.

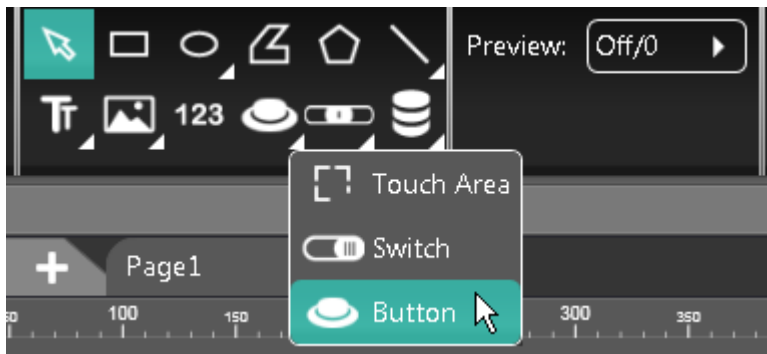


Click "Advanced" to select a colour using the RGB colour selection mask.

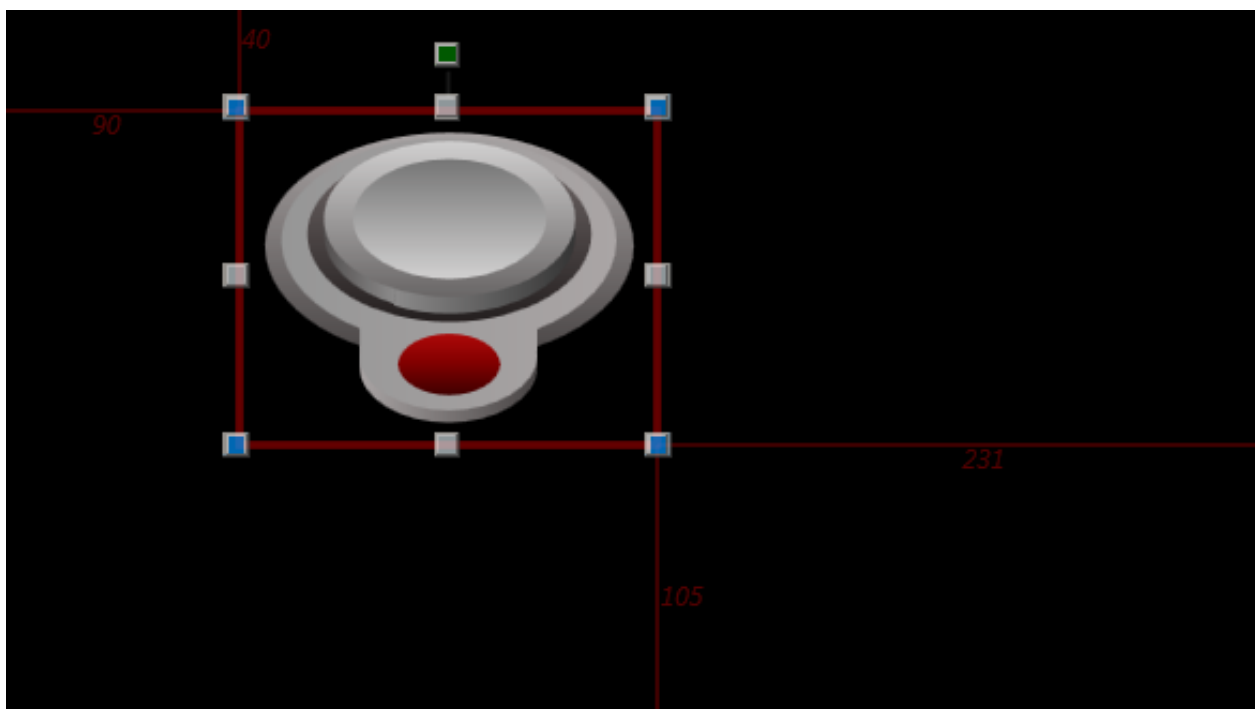


# CREW Manual

## Button



The “Button” icon on the “Graphics” menu is used to place a button on the page, drawing it with the mouse on the page.



To determine the features of the “Switch” field it is necessary to set them in the “Properties Editor”, as shown in the [“Button Properties”](#) section.

# CREW Manual

## Button Properties

The following image illustrates all the editable properties of the Button. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

The following table describes all the editable properties of the Button.

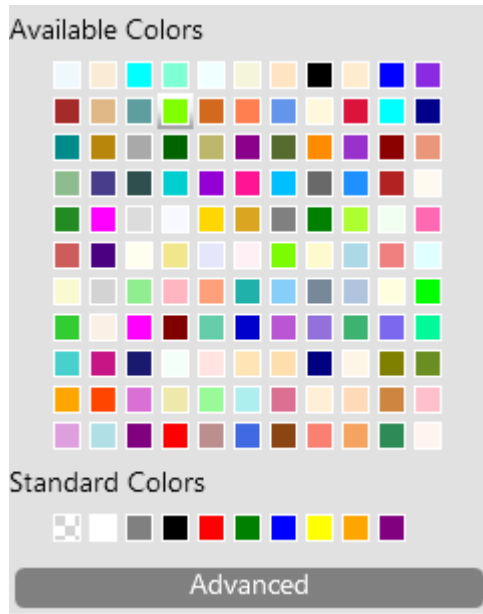
Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Rotation</b>	Determines the rotation degrees assigned to the object
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>Disabled</b>	Determines whether the user can interact with the object. This property is ignored when "UserMode" is not in read-write mode
<b>UserMode</b>	Determines whether the user can interact with the object: - read only, the user cannot interact with the object; this value takes precedence over the Disabled property - read-write, the user can interact with the object; the effective interaction is validated by the Disabled property
<b>TouchMode</b>	Determines the mode of interaction of the button: - SWITCH: The button acts as a normal button switch with on-pressed and on-released events generated when the user has released the button. - CONTACT: the button enters the "pressed" state as soon as the user touches the screen, and enters the state "released" when there is no more contact with the screen. - BUTTON: It is not necessary to associate a variable to the change of the button state, events associated to this are used
<b>Tag</b>	
<b>Tag</b>	Tag linked to the object
<b>ValueOn</b>	Represents the value of the "switch-on" threshold (default 1)
<b>ValueOff</b>	Represents the value of the "switch-off" threshold (default 1)

<b>Attributes</b>	
<b>Opacity</b>	Determines the opacity of the object
<b>On</b>	Allows you to assign the color of the object when it is "ON" using the color bar (Hue)
<b>SaturationOn</b>	It allows you to change the saturation of the color of the object in the "ON" state
<b>BrightnessOn</b>	It allows you to change the brightness of the color of the object in the "ON" state
<b>Off</b>	Allows you to assign the color of the object when it is "OFF" using the color bar (Hue)
<b>SaturationOff</b>	It allows you to change the saturation of the color of the object in the "OFF" state
<b>BrightnessOff</b>	It allows you to change the brightness of the color of the object in the "OFF" state
<b>LibraryVersion</b>	Allows you to change the default color of the object by choosing from the "Colors", "Black & White" and "Color Art" (Only for few switches in the library)
<b>Blink</b>	
<b>CanBlink</b>	It allows you to enable or disable the Runtime blink
<b>Visibility</b>	The "Visibility" property allows to display or hide the Runtime blink
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project
<b>ShowDisabled</b>	Enables the displaying of a "status" icon in the object when its use has been disabled
<b>ShowOffline</b>	Enables the displaying of a "status" icon in the object when it is in the offline condition
<b>ShowInvalid</b>	Enables the displaying of a "status" icon in the object when its current value is invalid
<b>ShowProtected</b>	Enables the displaying of a "status" icon in the object when it is protected for the current user

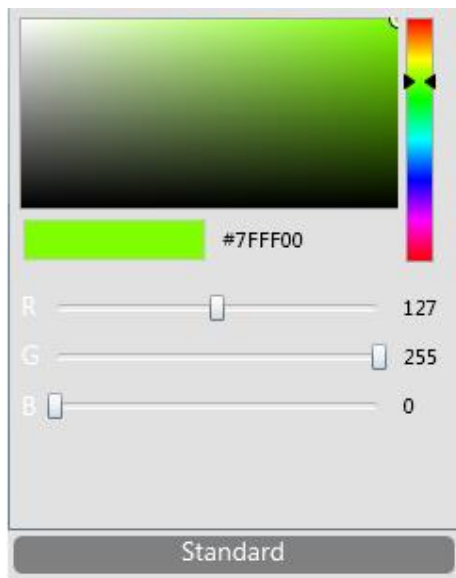


# CREW Manual

The properties related to colours can be edited through the colour palette.

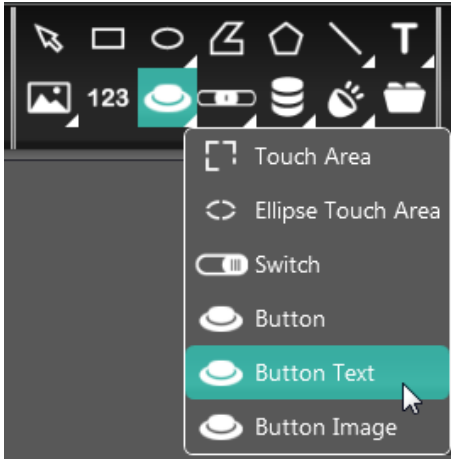


Click “Advanced” to select a colour using the RGB colour selection mask.

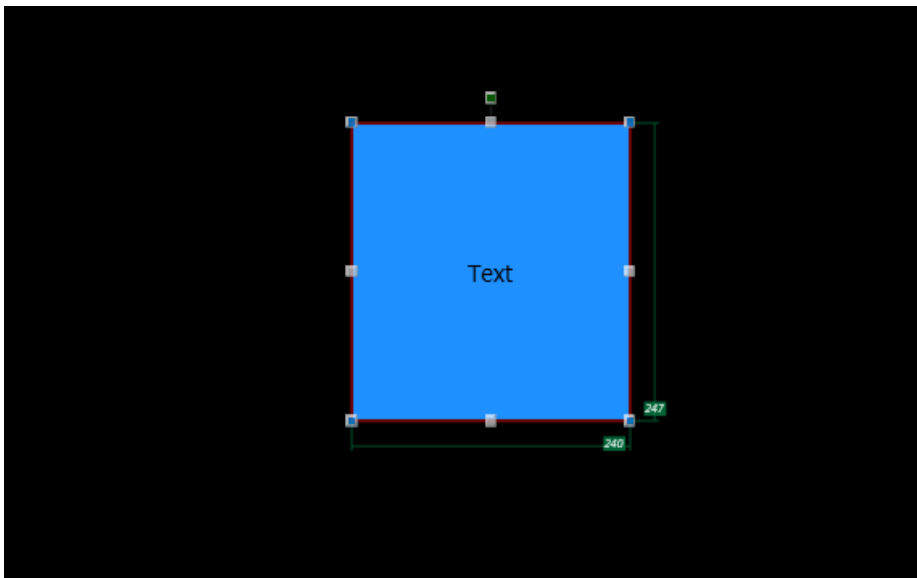


# CREW Manual

## Text Button



The "Text Button" icon on the "Graphics" menu is used to place a text button on the page, drawing it with the mouse on the page.

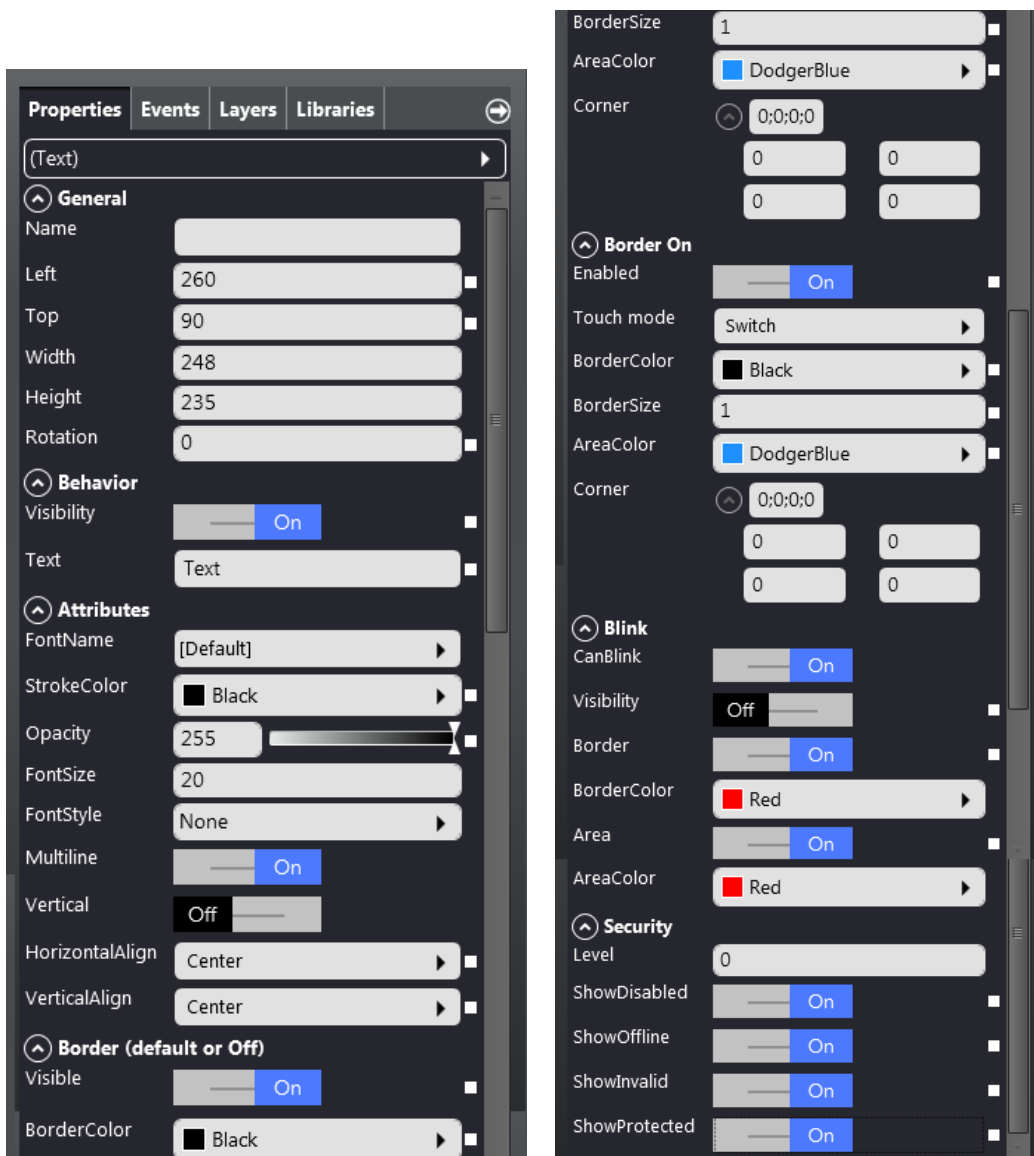


To determine the features of the "Text Button" field it is necessary to set them in the "Properties Editor", as shown in the [Text Button Properties](#) section.

# CREW Manual

## Text Button Properties

The following image illustrates all the editable properties of the Text Button. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

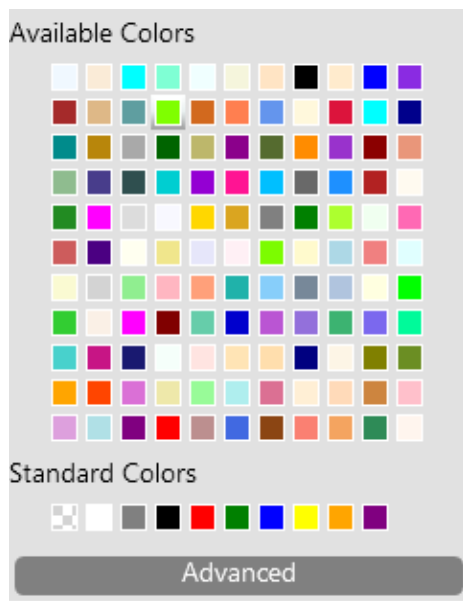
The following table describes all the editable properties of the Text Button.

Properties	Descrizione
<b>General</b>	
Name	Object ID
Left	Horizontal coordination of position
Top	Vertical coordination of position
Width	Width
Height	Height
Rotation	Determines the rotation degrees assigned to the object
<b>Behavior</b>	
Visibility	Determines whether the object should be displayed or not
Text	Text shown in the "button" area
<b>Attributes</b>	
FontName	Determines the font used for the items
StrokeColor	Determines the fill colour of the object; the colour is selectable by RGB colour code or colour palette
Opacity	Determines the opacity of the object
FontSize	Determines the size of the font of the object
FontStyle	Font style. Any combination of the following features: - None - Italics - Bold - Underline
Multiline	If the text automatically starts a newline, it determines the number of lines needed with respect to the current width
HorizontalAlign	This function allows to position the image horizontally (Centred, Right or Left)
VerticalAlign	This function allows to position the image vertically (Centred, Top or Bottom)
<b>Border (default / Off)</b>	
Visible	Determines whether the border should be visible or not
Border Color	Determines the color of the border
Border Size	Determines the border thickness
Area Color	Determines the area fill color
Corner	Determines the roundness of the angles

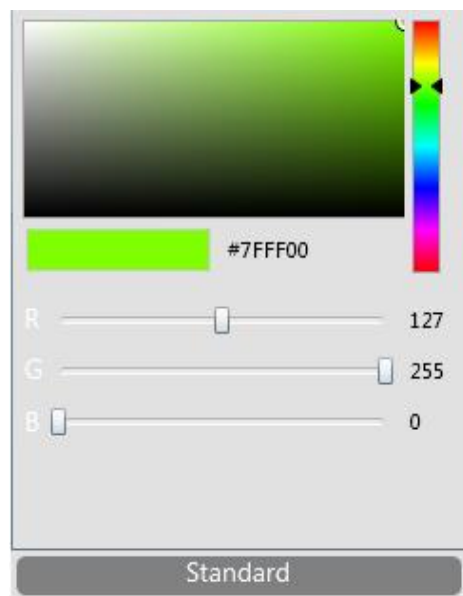
<b>Border On</b>	
Enabled	Determines whether the user can interact with the object
TouchMode	Determines the mode of interaction of the button: - SWITCH: The button acts as a normal button switch with on-pressed and on-released events generated when the user has released the button. - CONTACT: the button enters the "pressed" state as soon as the user touches the screen, and enters the state "released" when there is no more contact with the screen. - BUTTON: It is not necessary to associate a variable to the change of the button state, events associated to this are used
BorderColor	Determines the color of the border
BorderSize	Determines the border thickness
AreaColor	Determines the area fill color
Corner	Determines the roundness of the angles
<b>Blink</b>	
CanBlink	It allows you to enable or disable the Runtime blink
Visibility	Allows to display or hide the Runtime blink
Border	Determines the blinking or less of the border
BorderColor	Determines the color of the border during blinking
Area	Determines the blinking or less of the area
AreaColor	Determines the area fill color during blinking
<b>Security</b>	
Level	Allows to define the levels of authentication required to control access to specific areas of the project
ShowDisabled	Enables the displaying of a "status" icon in the object when its use has been disabled
ShowOffline	Enables the displaying of a "status" icon in the object when it is in the offline condition
ShowInvalid	Enables the displaying of a "status" icon in the object when its current value is invalid
ShowProtected	Enables the displaying of a "status" icon in the object when it is protected for the current user

# CREW Manual

The properties related to colours can be edited through the colour palette.

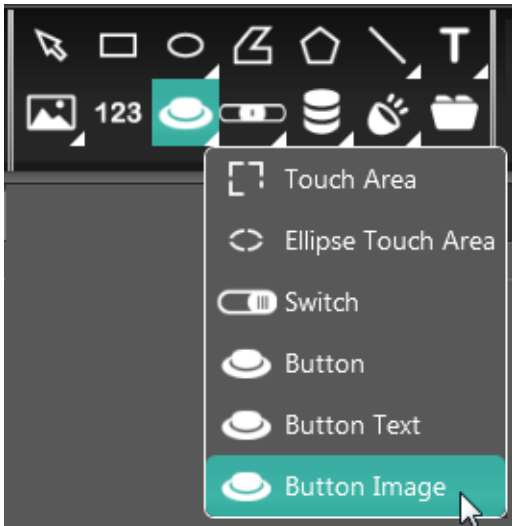


Click "Advanced" to select a colour using the RGB colour selection mask.

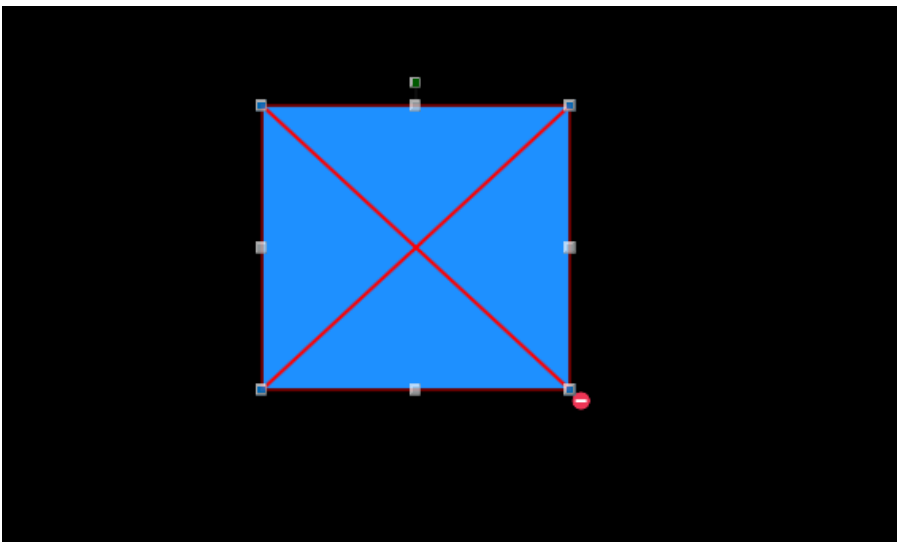


# CREW Manual

## Image Button



The “Image Button” icon on the “Graphics” menu is used to place an image button on the page, drawing it with the mouse on the page.

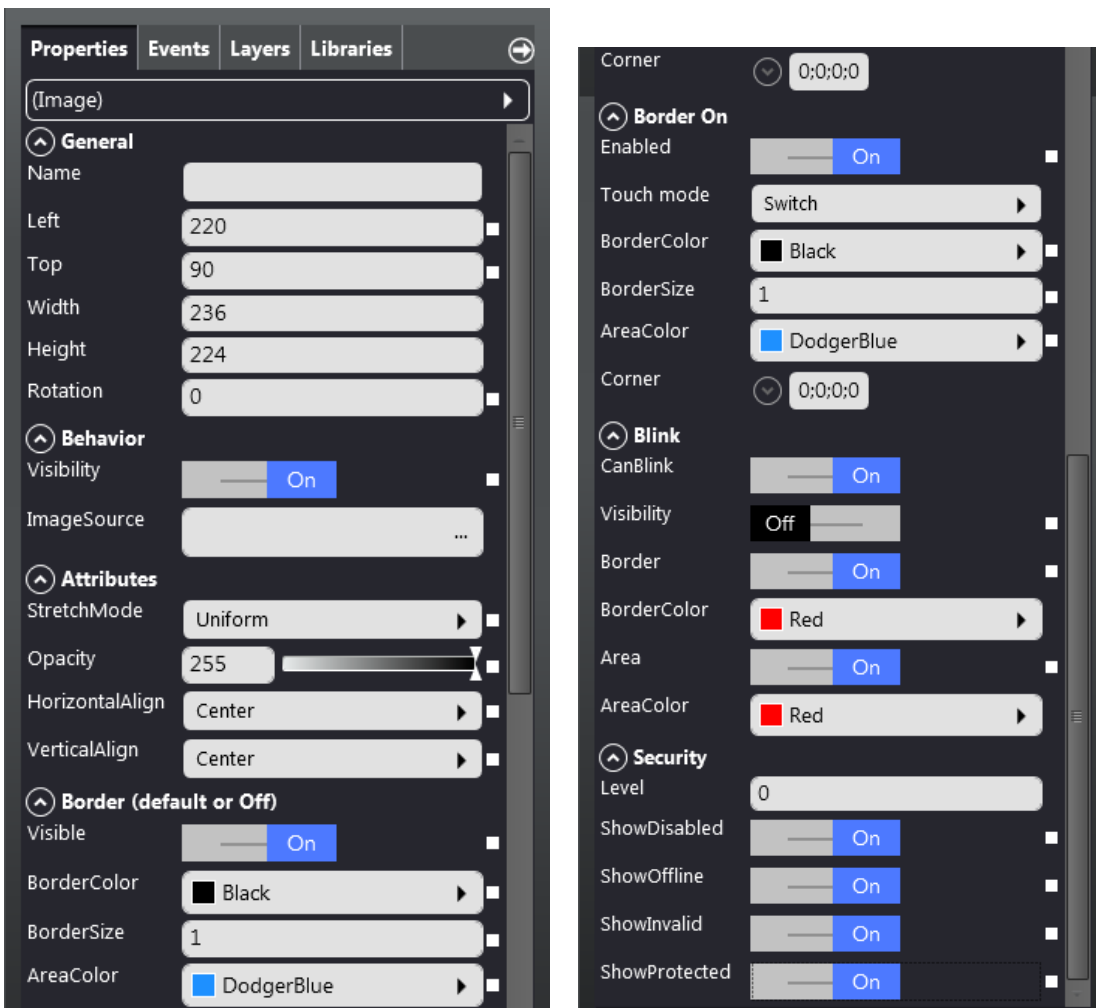


To determine the features of the “Image Button” field it is necessary to set them in the “Properties Editor”, as shown in the [Image Button Properties](#) section.

# CREW Manual

## Image Button Properties

The following image illustrates all the editable properties of the Image Button. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

The following table describes all the editable properties of the Image Button.

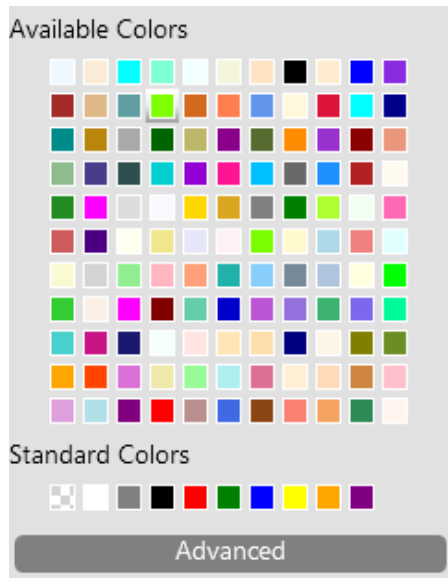
Properties	Descrizione
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Rotation</b>	Determines the rotation degrees assigned to the object
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>ImageSource</b>	Indicates the path from which the image to be inserted is imported
<b>Attributes</b>	
<b>StretchMode</b>	Resize grouped elements maintaining the aspect
<b>Opacity</b>	Determines the opacity of the object
<b>HorizontalAlign</b>	This function allows to position the image horizontally (Centred, Right or Left)
<b>VerticalAlign</b>	This function allows to position the image vertically (Centred, Top or Bottom)
<b>Border (default / Off)</b>	
<b>Visible</b>	Determines whether the border should be visible or not
<b>Border Color</b>	Determines the color of the border
<b>Border Size</b>	Determines the border thickness
<b>Area Color</b>	Determines the area fill color
<b>Corner</b>	Determines the roundness of the angles

<b>Border On</b>	
<b>Enabled</b>	Determines whether the user can interact with the object
<b>TouchMode</b>	Determines the mode of interaction of the button: - SWITCH: The button acts as a normal button switch with on-pressed and on-released events generated when the user has released the button. - CONTACT: the button enters the "pressed" state as soon as the user touches the screen, and enters the state "released" when there is no more contact with the screen. - BUTTON: It is not necessary to associate a variable to the change of the button state, events associated to this are used
<b>BorderColor</b>	Determines the color of the border
<b>BorderSize</b>	Determines the border thickness
<b>AreaColor</b>	Determines the area fill color
<b>Corner</b>	Determines the roundness of the angles
<b>Blink</b>	
<b>CanBlink</b>	It allows you to enable or disable the Runtime blink
<b>Visibility</b>	Allows to display or hide the Runtime blink
<b>Border</b>	Determines the blinking or less of the border
<b>BorderColor</b>	Determines the color of the border during blinking
<b>Area</b>	Determines the blinking or less of the area
<b>AreaColor</b>	Determines the area fill color during blinking
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project
<b>ShowDisabled</b>	Enables the displaying of a "status" icon in the object when its use has been disabled
<b>ShowOffline</b>	Enables the displaying of a "status" icon in the object when it is in the offline condition
<b>ShowInvalid</b>	Enables the displaying of a "status" icon in the object when its current value is invalid
<b>ShowProtected</b>	Enables the displaying of a "status" icon in the object when it is protected for the current user

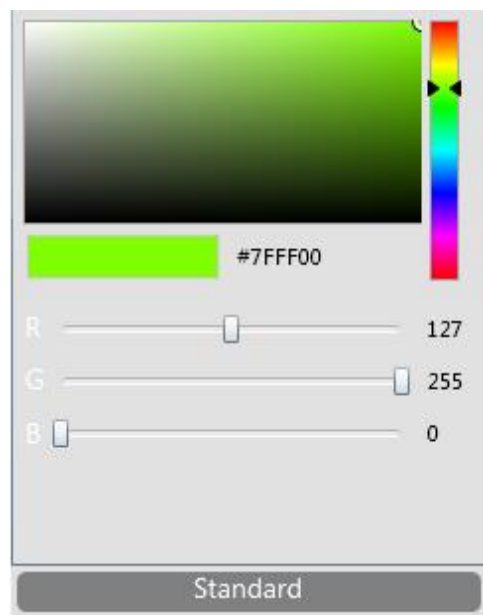


# CREW Manual

The properties related to colours can be edited through the colour palette.

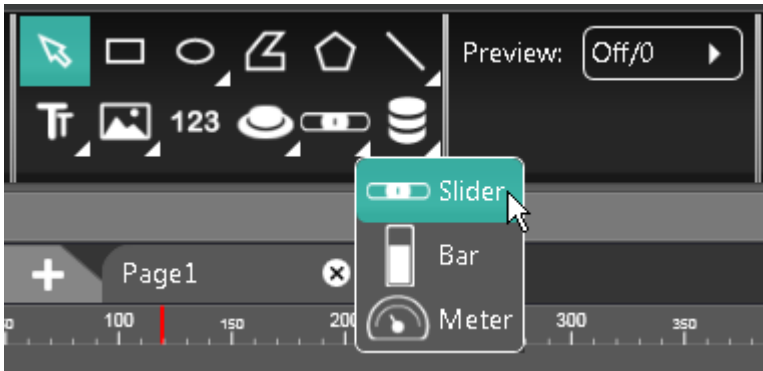


Click “Advanced” to select a colour using the RGB colour selection mask.

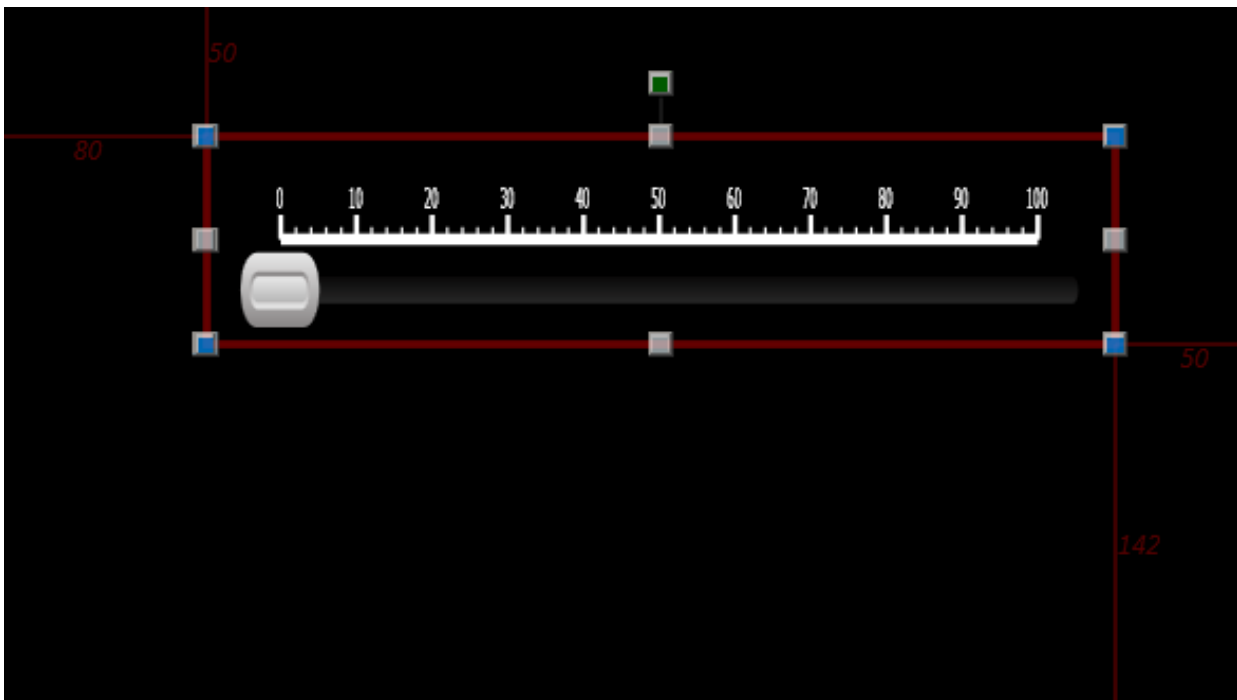


# CREW Manual

## Selector



The “Selector” icon on the “Graphics” menu is used to place a selector on the page, drawing it with the mouse on the page.

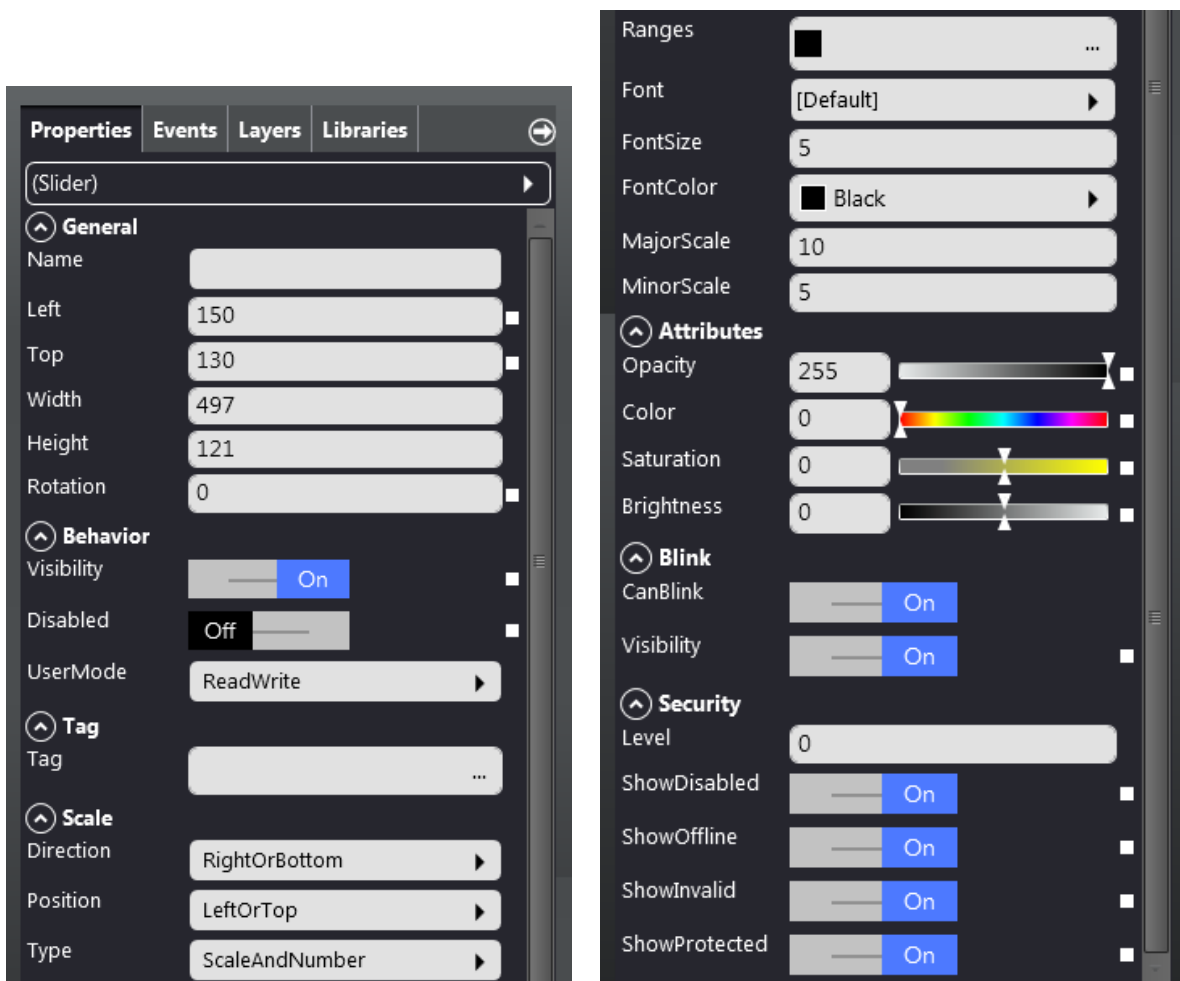


To determine the features of the “Selector” field it is necessary to set them in the “Properties Editor”, as shown in the "[Selector Properties](#)" section.

# CREW Manual

## Selector Properties

The following image illustrates all the editable properties of the Selector. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

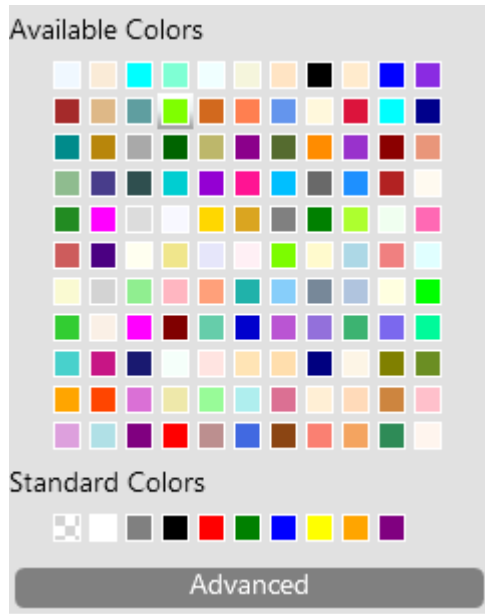
The following table describes all the editable properties of the Selector.

Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Rotation</b>	Determines the rotation degrees assigned to the object
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>Disabled</b>	Determines whether the user can interact with the object. This property is ignored when "UserMode" is not in read-write mode
<b>UserMode</b>	Determines whether the user can interact with the object: - read only, the user cannot interact with the object; this value takes precedence over the Disabled property - read-write, the user can interact with the object; the effective interaction is validated by the Disabled property
<b>Tag</b>	
<b>Tag</b>	Tag linked to the object
<b>Scale</b>	
<b>Direction</b>	Indicates the direction of the slider and can be from right to left or vice versa
<b>Position</b>	Tells you where to position the scale of values with respect to the selector. If the direction of the switch is vertical, the scale of values can be positioned to the left or right, if the direction is horizontal, the scale can be Top or Bottom
<b>Types</b>	Allows you to select the type of scale to display, the choices are as follows: - None : at runtime will be shown only the Slider value - Only Scale : both Slider indication and scale of values will be displayed - Just Numbers : numbers will be displayed along with Slider indication - Numbers and Scale : scale and the numbers will be displayed in addition to the Selector

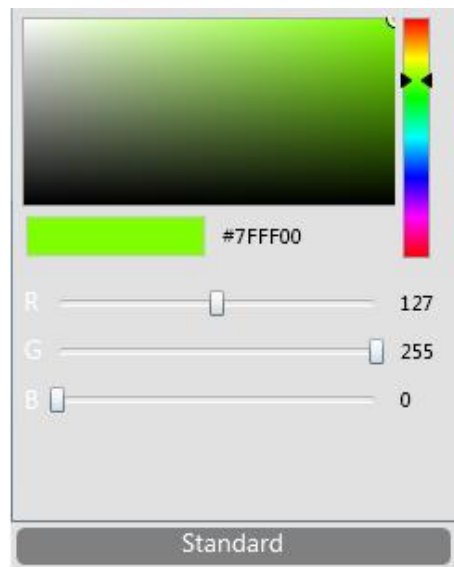
<b>Ranges</b>	Indicates groups of color can be assigned to particular ranges of values within the scale. By clicking you get into an editing window where you can specify the intervals of values and corresponding colors, from this window you can also specify the limits for the scale to display
<b>Font</b>	Indicates the font of the text of the Slider labels
<b>FontSize</b>	Indicates the font size of the numerical values written above the notches
<b>FontColor</b>	Represents the color of the notches of the slider
<b>MajorScale</b>	Indicates the number of the big notches of the scale of values. You will also see the digit of the value above the notch (calculated based on the number of notches)
<b>MinorScale</b>	Indicates the number of the small notches of the scale of values. You will also see the digit of the value above the notch (calculated based on the number of notches)
<b>Attributes</b>	
<b>Opacity</b>	Determines the opacity of the object
<b>Color</b>	Allows you to assign the color to the object using the color bar (Hue) (valid for objects in the Library)
<b>Saturation</b>	It allows you to change the saturation of the color of the object
<b>Brightness</b>	It allows you to change the brightness of the color of the object
<b>Blink</b>	
<b>CanBlink</b>	It allows you to enable or disable the Runtime blink
<b>Visibility</b>	The "Visibility" property allows to display or hide the Runtime blink
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project
<b>ShowDisabled</b>	Enables the displaying of a "status" icon in the object when its use has been disabled
<b>ShowOffline</b>	Enables the displaying of a "status" icon in the object when it is in the offline condition
<b>ShowInvalid</b>	Enables the displaying of a "status" icon in the object when its current value is invalid
<b>ShowProtected</b>	Enables the displaying of a "status" icon in the object when it is protected for the current user

# CREW Manual

The properties related to colours can be edited through the colour palette.



Click “Advanced” to select a colour using the RGB colour selection mask.

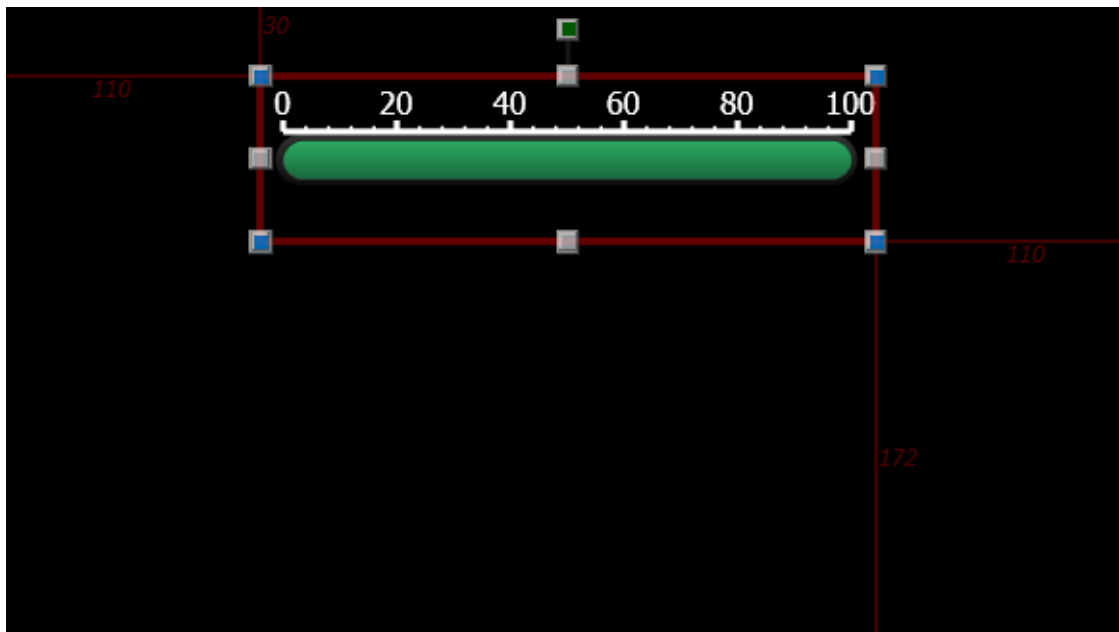


# CREW Manual

## Bar



The “Bar” icon on the “Graphics” menu is used to place a bar on the page, drawing it with the mouse on the page.

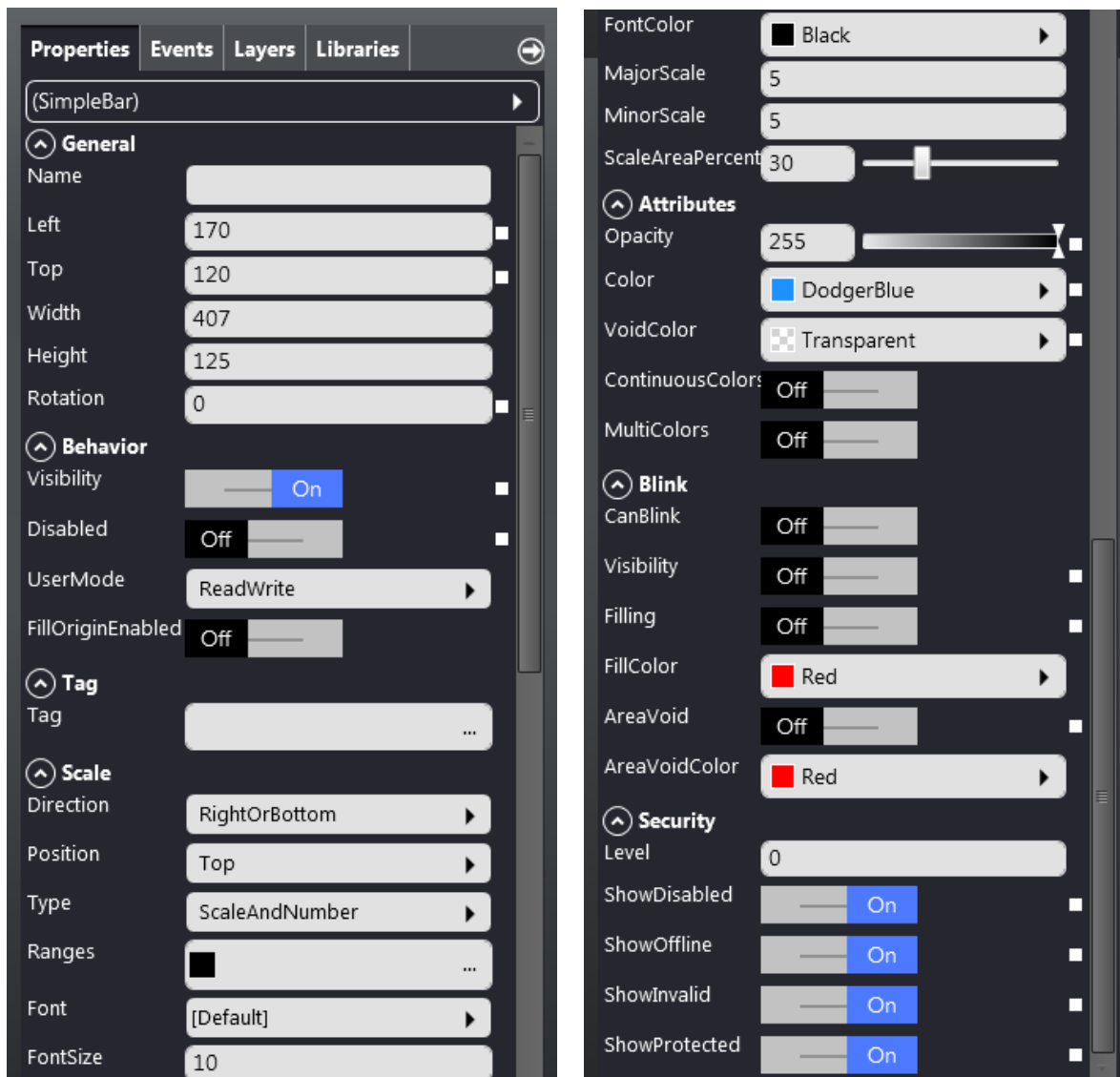


To determine the features of the “Bar” field it is necessary to set them in the “Properties Editor”, as shown in the "[Bar Properties](#)" section.

# CREW Manual

## Bar Properties

The following image illustrates all the editable properties of the Bar. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

The following table describes all the editable properties of the Bar.

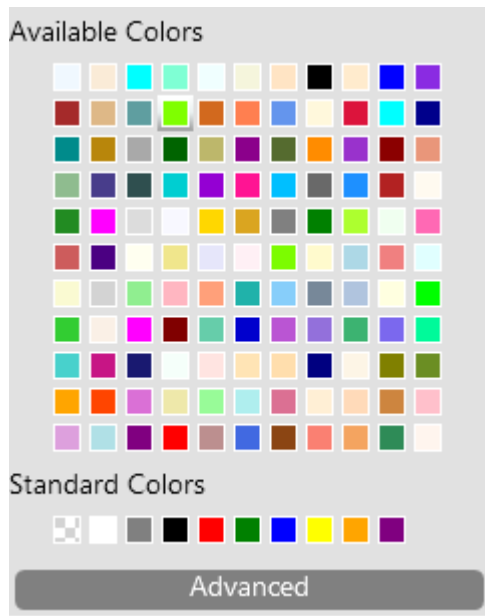
Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Rotation</b>	Determines the rotation degrees assigned to the object
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>Disabled</b>	Determines whether the user can interact with the object. This property is ignored when "UserMode" is not in read-write mode
<b>UserMode</b>	Determines whether the user can interact with the object: - read only, the user cannot interact with the object; this value takes precedence over the Disabled property - read-write, the user can interact with the object; the effective interaction is validated by the Disabled property
<b>Tag</b>	
<b>Tag</b>	Tag linked to the object
<b>Scale</b>	
<b>Color</b>	Color used to fill the area of the object
<b>Direction</b>	Indicates the direction of the Bar and can be from right to left or vice versa
<b>Position</b>	Tells you where to position the scale of values with respect to the Bar; if the direction of the Slider is vertical, the scale can be positioned to the left or right, if the direction is horizontal, the scale can be Top or Bottom
<b>Type</b>	Allows you to select the type of scale to display, the choices are as follows: - None : at runtime will be shown only the Bar value - Only Scale : both Bar indication and scale of values will be displayed - Just Numbers : numbers will be displayed along with Bar indication - Numbers and Scale : scale and the numbers will be displayed in addition to the Bar
<b>Ranges</b>	Indicates groups of color can be assigned to particular ranges of values within the scale. By clicking you get into an editing window where you can specify the intervals of values and corresponding colors, from this window you can also specify the limits for the scale to display
<b>Font</b>	Indicates the font of the text of the Bar labels

<b>FontSize</b>	Indicates the font size of the numerical values written above the notches
<b>FontColor</b>	Represents the color of the notches of the Bar
<b>MajorScale</b>	Indicates the number of the big notches of the scale of values. You will also see the digit of the value above the notch (calculated based on the number of notches)
<b>MinorScale</b>	Indicates the number of the small notches of the scale of values. You will also see the digit of the value above the notch (calculated based on the number of notches)
<b>ScaleAreaPercent</b>	It defines the percentage of space dedicated to the scale of the object
<b>Attributes</b>	
<b>Opacity</b>	Determines the opacity of the object
<b>Color</b>	Allows you to assign the color to the object using the color bar (Hue) (valid for objects in the Library)
<b>Blink</b>	
<b>CanBlink</b>	It allows you to enable or disable the Runtime blink
<b>Visibility</b>	Allows to display or hide the Runtime blink
<b>Filling</b>	Determines the blinking of the object's internal area
<b>FillColor</b>	Determines the blinking of the object's internal area; it is selectable by RGB colour code or colour palette
<b>AreaVoid</b>	Enables the blinking of the empty part
<b>AreaVoidColor</b>	Allows you to choose the color of the empty part of the object
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project
<b>ShowDisabled</b>	Enables the displaying of a "status" icon in the object when its use has been disabled
<b>ShowOffline</b>	Enables the displaying of a "status" icon in the object when it is in the offline condition
<b>ShowInvalid</b>	Enables the displaying of a "status" icon in the object when its current value is invalid
<b>ShowProtected</b>	Enables the displaying of a "status" icon in the object when it is protected for the current user

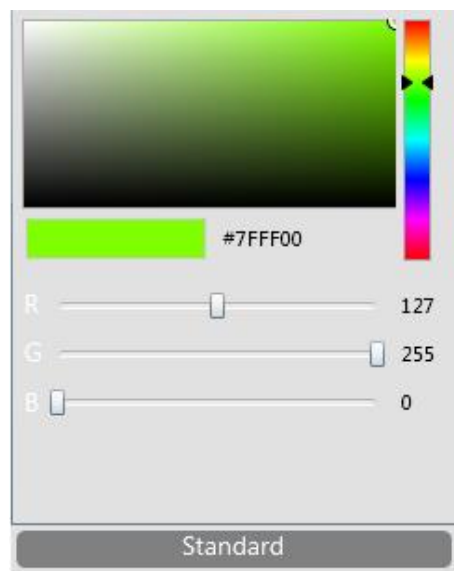


# CREW Manual

The properties related to colours can be edited through the colour palette.



Click "Advanced" to select a colour using the RGB colour selection mask.

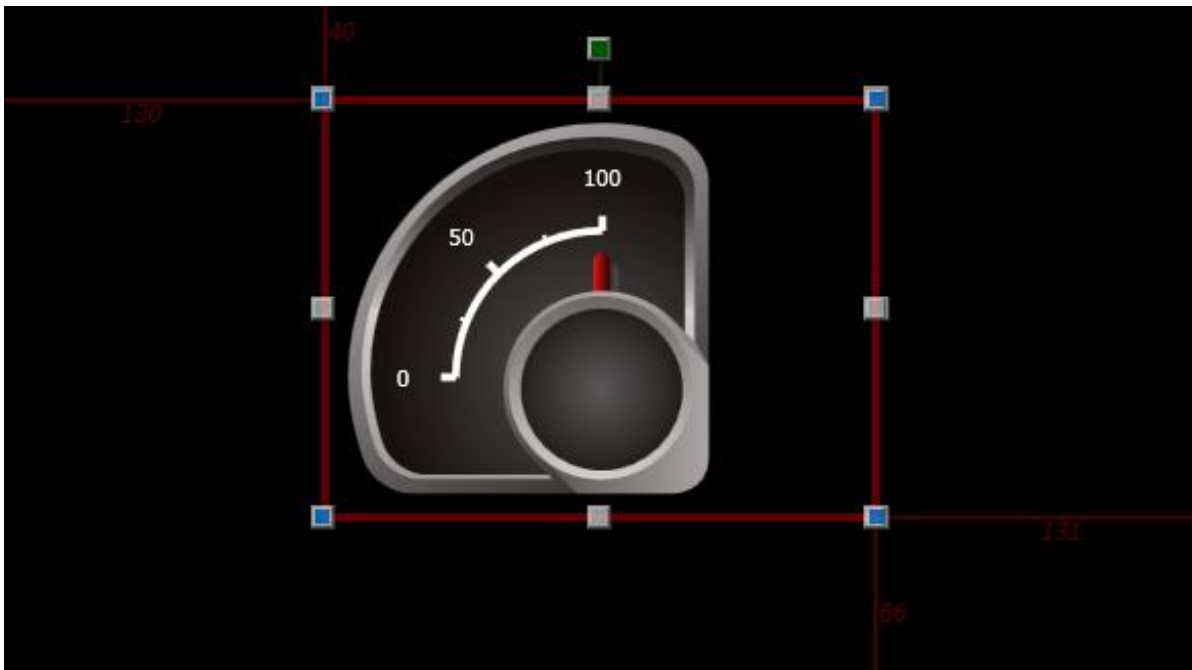


# CREW Manual

## Indicator



The “Indicator” icon on the “Graphics” menu is used to place an indicator on the page, drawing it with the mouse on the page.

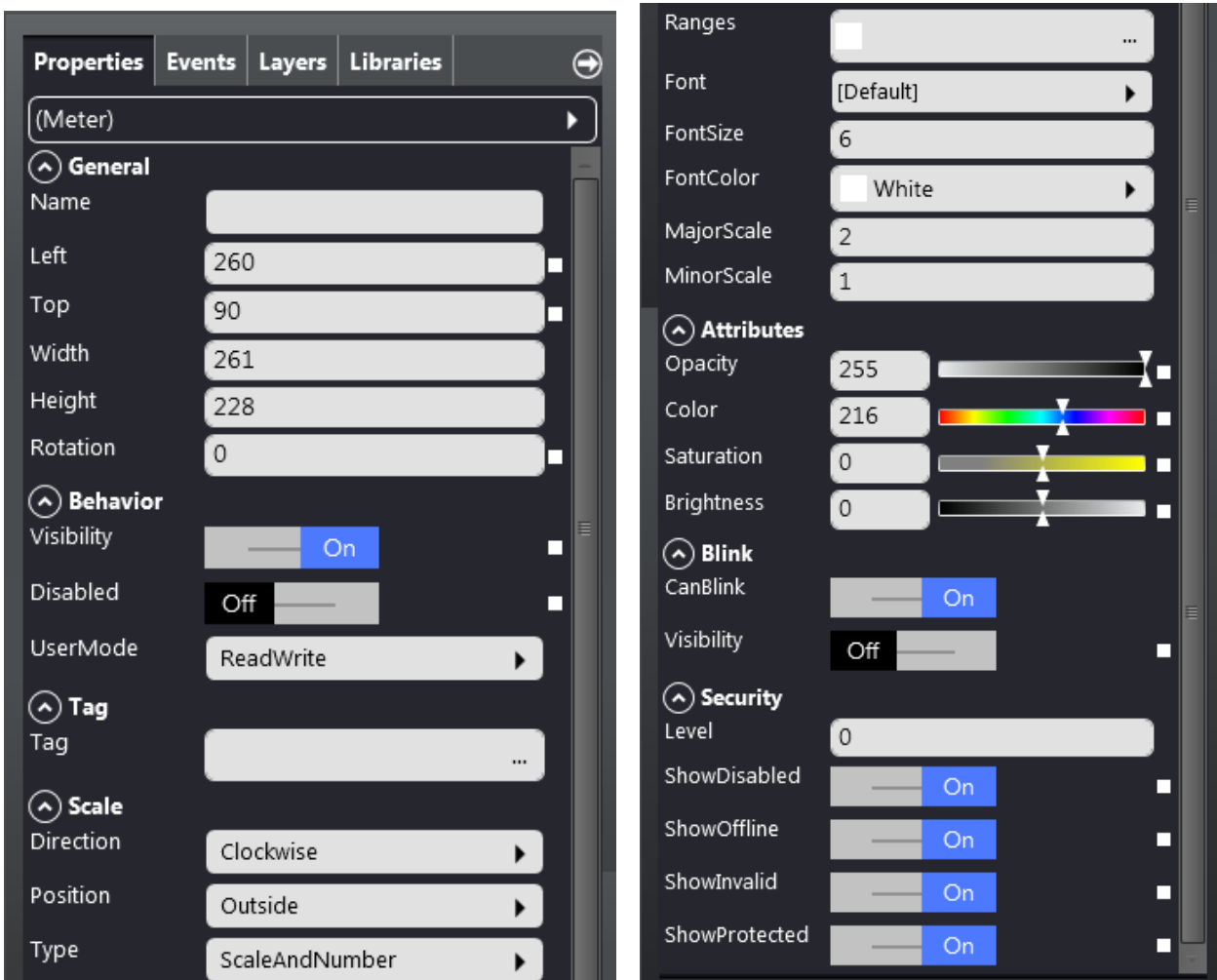


To determine the features of the “Indicator” field it is necessary to set them in the “Properties Editor”, as shown in the “[Indicator Properties](#)” section.

# CREW Manual

## Indicator Properties

The following image illustrates all the editable properties of the Indicator. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

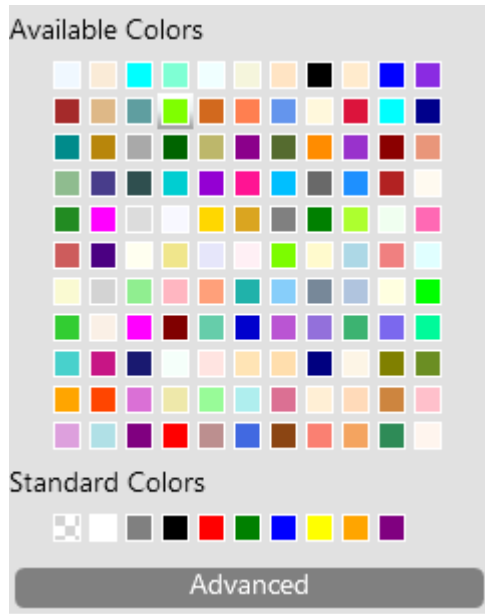
The following table describes all the editable properties of the Indicator object.

Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Rotation</b>	Determines the rotation degrees assigned to the object
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>Disabled</b>	Determines whether the user can interact with the object. This property is ignored when "UserMode" is not in read-write mode
<b>UserMode</b>	Determines whether the user can interact with the object: - read only, the user cannot interact with the object; this value takes precedence over the Disabled property - read-write, the user can interact with the object; the effective interaction is validated by the Disabled property
<b>Tag</b>	
<b>Tag</b>	Tag linked to the object
<b>Scale</b>	
<b>Direction</b>	Indicates the direction of the Bar and can be from right to left or vice versa
<b>Position</b>	Tells you where to position the scale of values with respect to the Bar; if the direction of the Slider is vertical, the scale can be positioned to the left or right, if the direction is horizontal, the scale can be Top or Bottom
<b>Type</b>	Allows you to select the type of scale to display, the choices are as follows: - None : at runtime will be shown only the Meter value - Only Scale : both Meter indication and scale of values will be displayed - Just Numbers : numbers will be displayed along with Meter indication - Numbers and Scale : scale and the numbers will be displayed in addition to the Meter

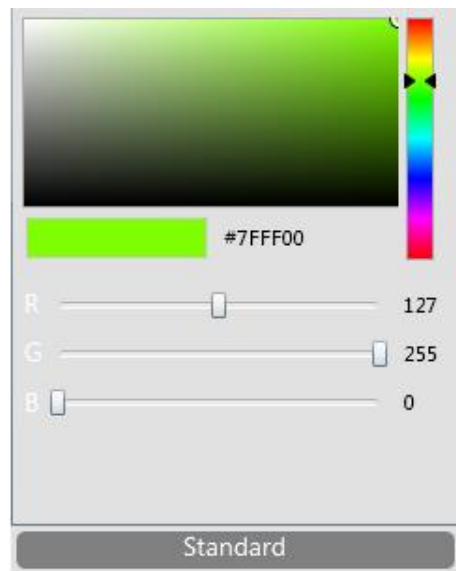
<b>Ranges</b>	Indicates groups of color can be assigned to particular ranges of values within the scale. By clicking you get into an editing window where you can specify the intervals of values and corresponding colors, from this window you can also specify the limits for the scale to display
<b>Font</b>	Indicates the font of the text of the Meter labels
<b>FontSize</b>	Indicates the font size of the numerical values written above the notches
<b>FontColor</b>	Represents the color of the notches of the Meter
<b>MajorScale</b>	Indicates the number of the big notches of the scale of values. You will also see the digit of the value above the notch (calculated based on the number of notches)
<b>MinorScale</b>	Indicates the number of the small notches of the scale of values. You will also see the digit of the value above the notch (calculated based on the number of notches)
<b>Attributes</b>	
<b>Opacity</b>	Determines the opacity of the object
<b>Color</b>	Allows you to assign the color to the object using the color bar (Hue) (valid for objects in the Library)
<b>Saturation</b>	It allows you to change the saturation of the color of the object
<b>Brightness</b>	It allows you to change the brightness of the color of the object
<b>Blink</b>	
<b>CanBlink</b>	It allows you to enable or disable the Runtime blink
<b>Visibility</b>	The "Visibility" property allows to display or hide the Runtime blink
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project
<b>ShowDisabled</b>	Enables the displaying of a "status" icon in the object when its use has been disabled
<b>ShowOffline</b>	Enables the displaying of a "status" icon in the object when it is in the offline condition
<b>ShowInvalid</b>	Enables the displaying of a "status" icon in the object when its current value is invalid
<b>ShowProtected</b>	Enables the displaying of a "status" icon in the object when it is protected for the current user

# CREW Manual

The properties related to colours can be edited through the colour palette.

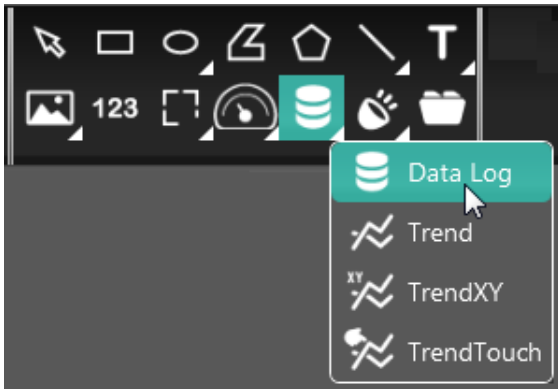


Click "Advanced" to select a colour using the RGB colour selection mask.

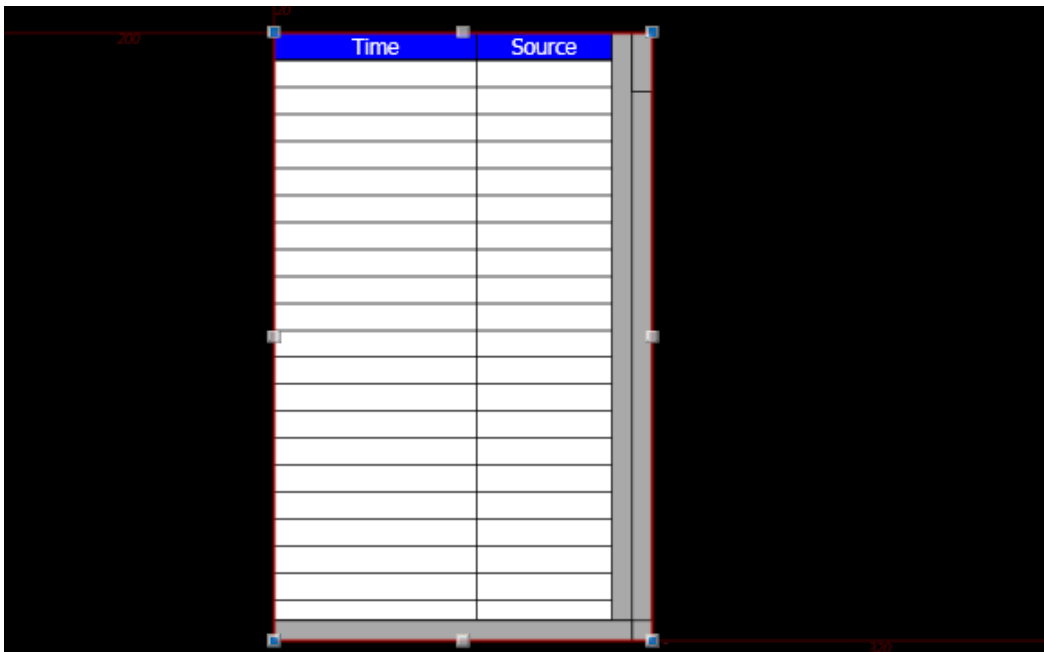


# CREW Manual

## Data Log



The “Data Log” icon on the “Graphics” menu is used to place a data log table on the page, drawing it with the mouse on the page.

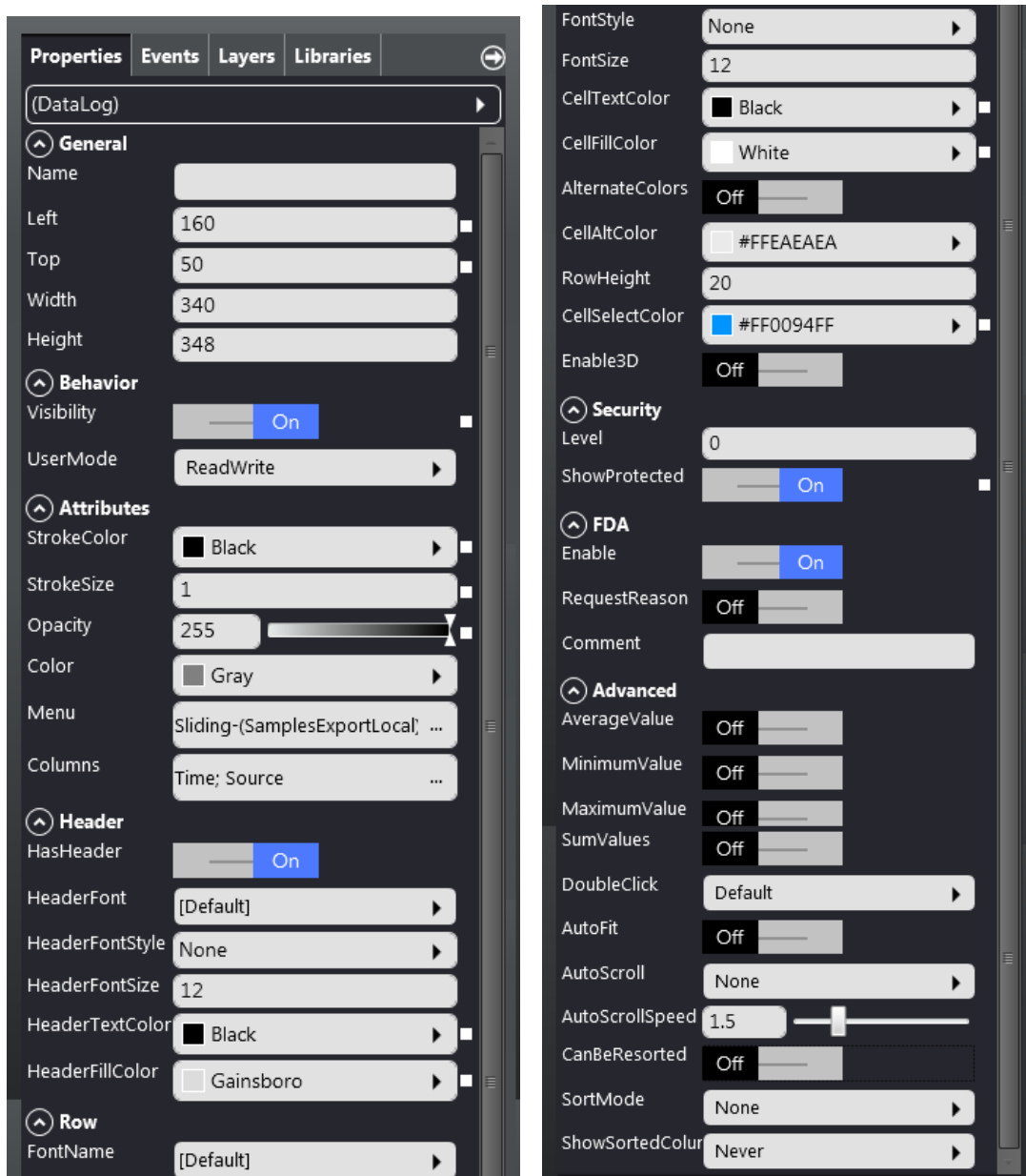


To determine the features of the “Data Log” table, set them in the “Properties Editor”, as shown in the section "[Data Log Properties](#)".

# CREW Manual

## Data Log Properties

The following image illustrates all the editable properties of the Data Log table. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

The following table describes all the editable properties of the Data Log table.

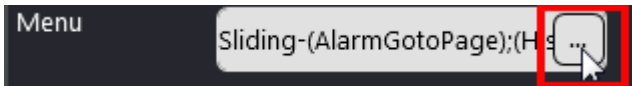
Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>UserMode</b>	Determines whether the user can interact with the object: - read only, the user cannot interact with the object; this value takes precedence over the Disabled property - read-write, the user can interact with the object; the effective interaction is validated by the Disabled property
<b>Attributes</b>	
<b>StrokeColor</b>	Determines the fill colour of the object; the colour is selectable by RGB colour code or colour palette
<b>StrokeSize</b>	Determines the stroke colour (edges of the figure); the colour is selectable by RGB colour code or colour palette
<b>Opacity</b>	Determines the opacity of the object
<b>Color</b>	Allows you to assign the color to the object using the color bar (Hue) (valid for objects in the Library)
<b>Menu</b>	By clicking on the "Browse" menu option, you can make a Data Log management as described in this section
<b>Columns</b>	Clicking on the "Browse" button of the "Columns" option, you can determine which columns to appear in the table and define their properties (as described in this section)
<b>Header</b>	
<b>HasHeader</b>	Determines whether the view of the Data Log must have the header or not
<b>HeaderFont</b>	Determines the font type used to display the header text
<b>HeaderFontStyle</b>	Header font style. Any combination of the following features: - None - Italics - Bold - Underline
<b>HeaderFontSize</b>	Indicates the font size of the values written in the headers
<b>HeaderTextColor</b>	Determines the color of the header text of Data Log
<b>HeaderFillColor</b>	Determines the color of the cell that contains the header of Data Log
<b>Row</b>	
<b>FontName</b>	Determines the font used for the items of Data Log
<b>FontStyle</b>	Font style. Any combination of the following features: - None - Italics - Bold - Underline
<b>FontSize</b>	Determines the stroke of Data Log view
<b>CellTextColor</b>	Represents the color of the Data Log writing cells
<b>CellFillColor</b>	Determines the color of the cells of table columns
<b>AlternateColors</b>	Allows you to assign two alternating colors for each row in the table
<b>CelAltCol</b>	Determines the alternate color (active when the option "Alternate Color" is activated)
<b>RowHeight</b>	Determines the height of the row of the table (pixel)
<b>CellSelectColor</b>	Determines the color of the selected cell
<b>Enable3D</b>	Enable the 3D view ("embossed" view) of the table
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project
<b>ShowProtected</b>	Enables the displaying of a "status" icon in the object when it is protected for the current user
<b>FDA</b>	
<b>Enable</b>	Enables tracking of events related to the functions controlled through the grid menu
<b>RequestReason</b>	In each event to be recorded you have to insert the text that will be logged together with the time and date.
<b>Comment</b>	It defines a static comments to be tracked along with the execution events of the grid menu functions
<b>Advanced</b>	
<b>AverageValues</b>	Enable (in Runtime mode) a field where you can see the average between the values of Log Data
<b>MinimumValues</b>	Enable (in Runtime mode) a field where you can see the minimum value of Log Data
<b>MaximumValues</b>	Enable (in Runtime mode) a field where you can see the maximum value of Log Data
<b>SumValues</b>	Runs for n times (100 times by default) the sum of the values of Log Data
<b>DoubleClick</b>	It allows you to assign a function to "double click" on each row of the table selecting it from the available ones
<b>AutoFit</b>	It determines whether the size of the columns of the grid are automatically calculated to fit precisely in the total available size
<b>AutoScroll</b>	It determines whether the content of the grid should automatically scroll
<b>AutoScrollSpeed</b>	It defines the scroll speed (when "AutoScroll" is enabled)
<b>CanBeResorted</b>	It determines whether the grid lines can be ordered at runtime
<b>SortMode</b>	It defines the policy of the default grid sorting
<b>ShowSortedColumn</b>	It determines whether the column identified as sorting order must be highlighted in the grid



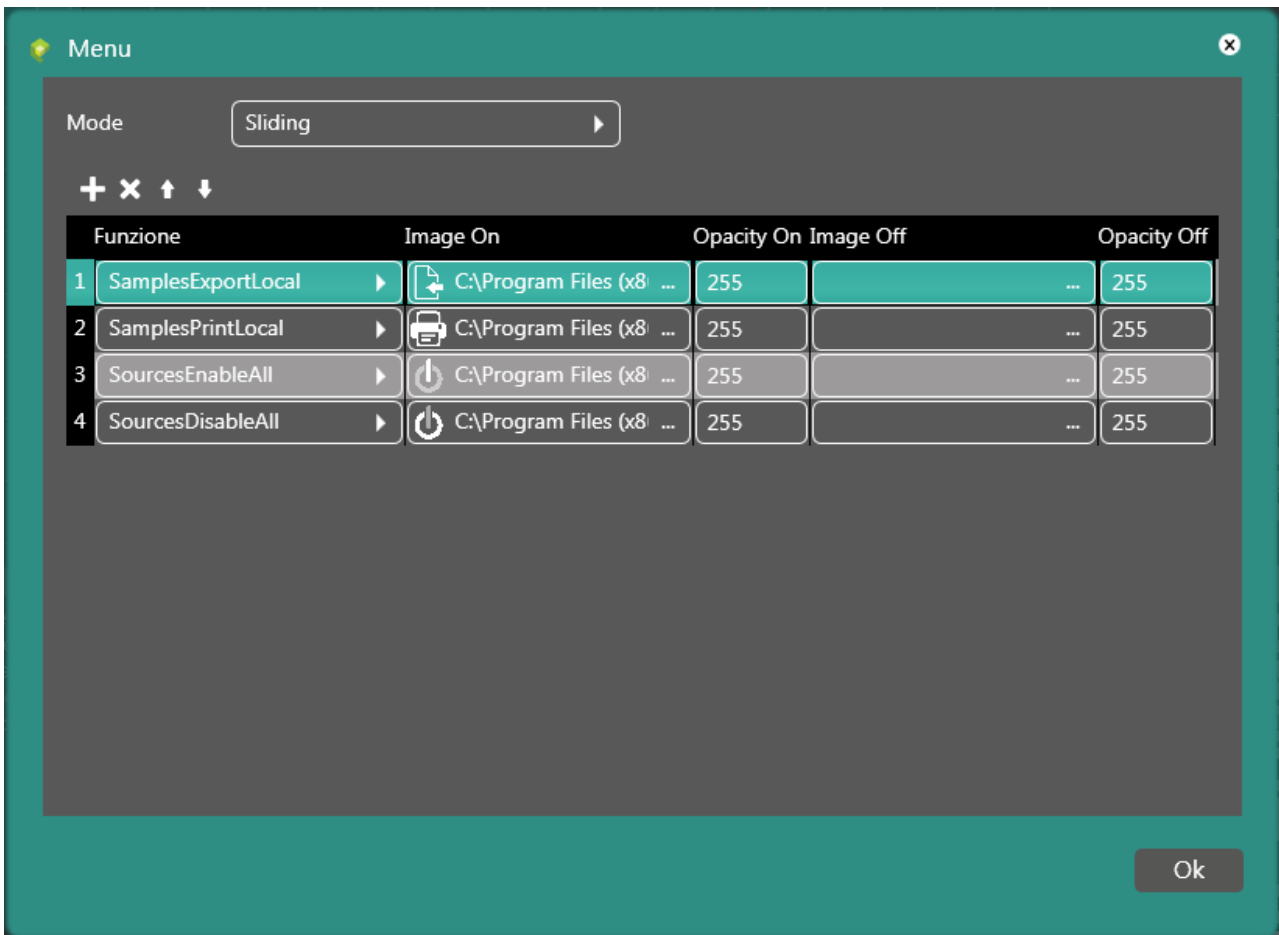
# CREW Manual

“Menu” option

Click the “Browse” key.

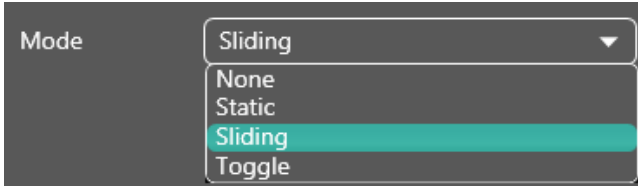


On the window that appears, it is possible to decide how to set the Runtime menu of the Data Log table.

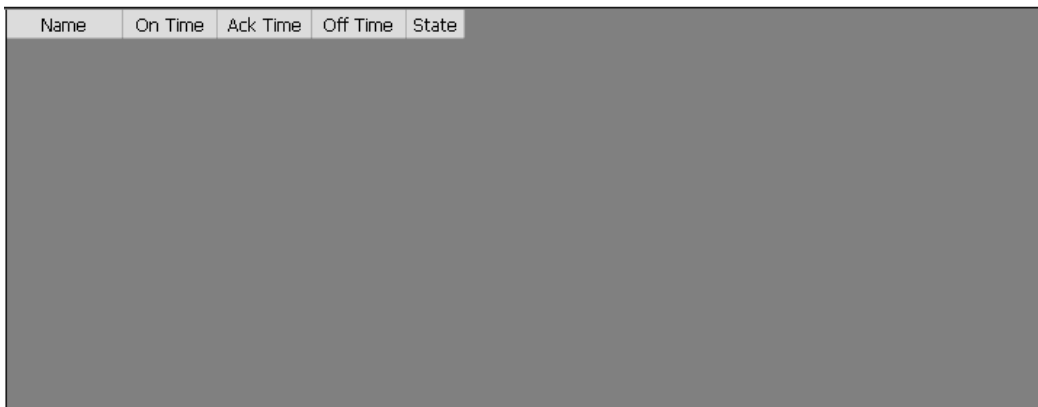


# CREW Manual

Mode: it is possible to customise the Runtime menu through one of the following options.



None: no Runtime menu. There are only the default columns.

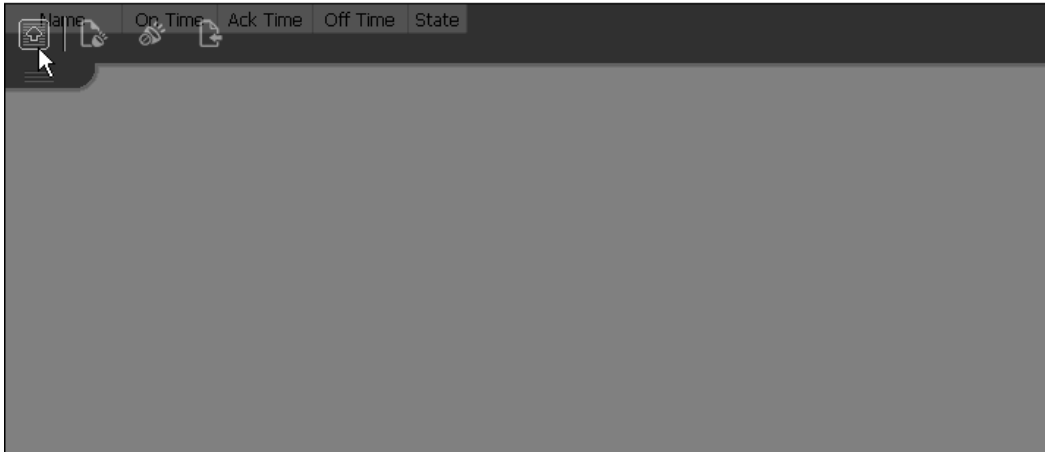


Static: static menu, namely fixed and always there.



# CREW Manual

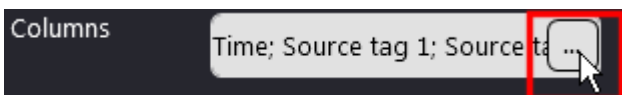
Sliding: floating menu, which appears at the user's discretion.



Toggle: floating menu (similar to the “Sliding” option), which appears at the user's discretion.

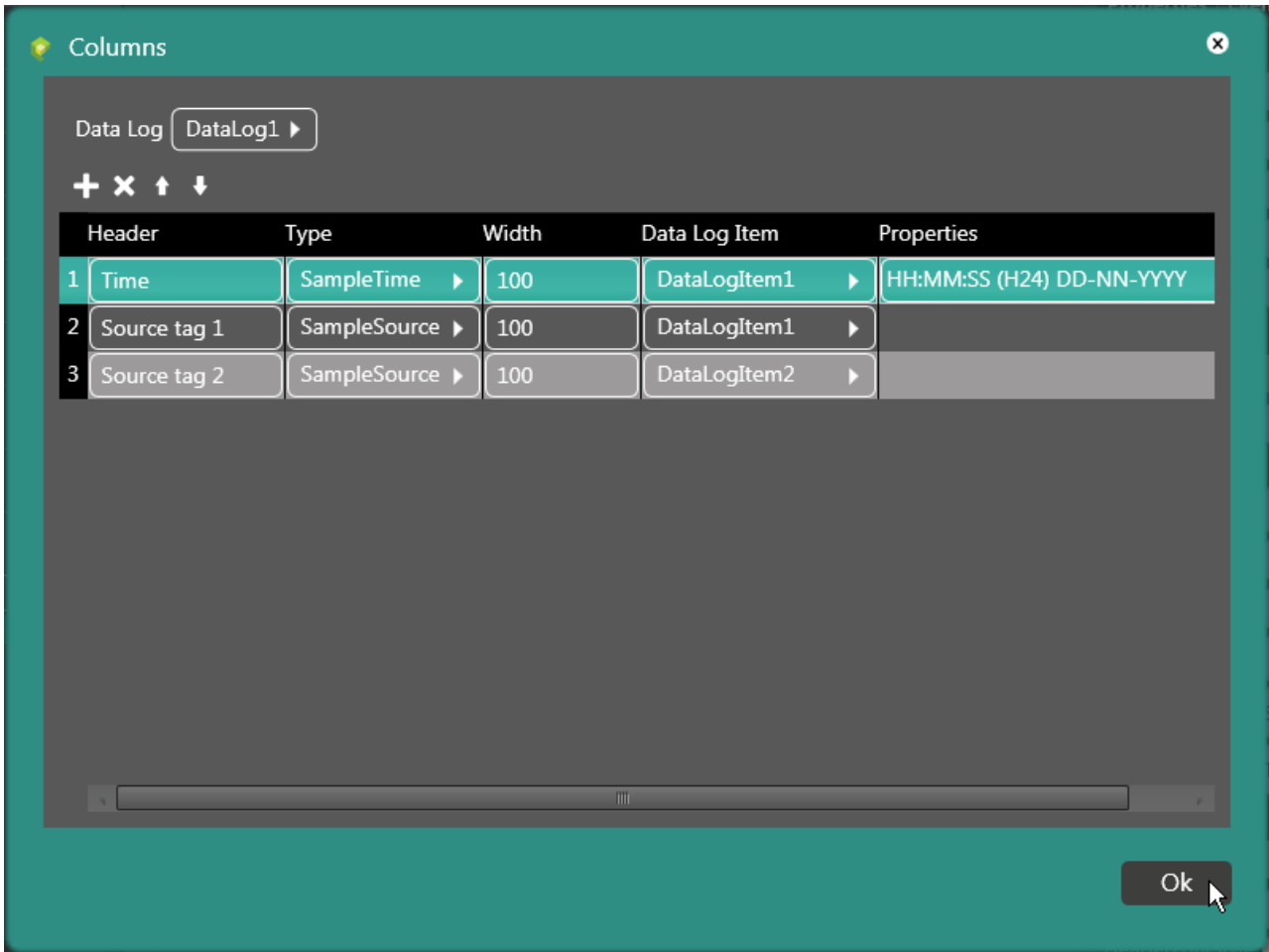
“Columns” option

Click the “Browse” key.

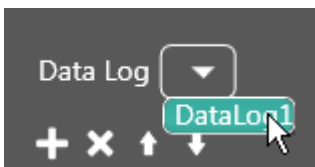


# CREW Manual

From the window, it is possible to decide how to set the Data Log table.



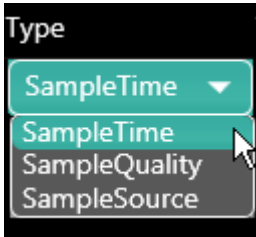
Data Log: the Data Log shown in the table, it is possible to select one of the ones contained in the project.



# CREW Manual

Heading: the title of the columns that comprise the table.

Type: the type of data sampling, which can be selected from the options shown in the image.



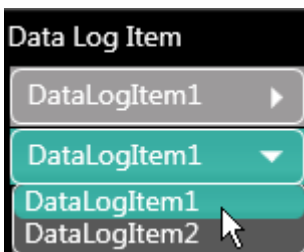
Time: the sampling date is shown (hour, minutes, seconds, year, month, day, customisable according to the “Properties” option described below).

Quality: the quality of the sampled data. With every correct sample taking, the value “1” is displayed, while a value of “0” is displayed if the datum is not sampled correctly.

Source: the source of the sampled datum. In the example, this is the value of the associated Tag (Data Log Item).

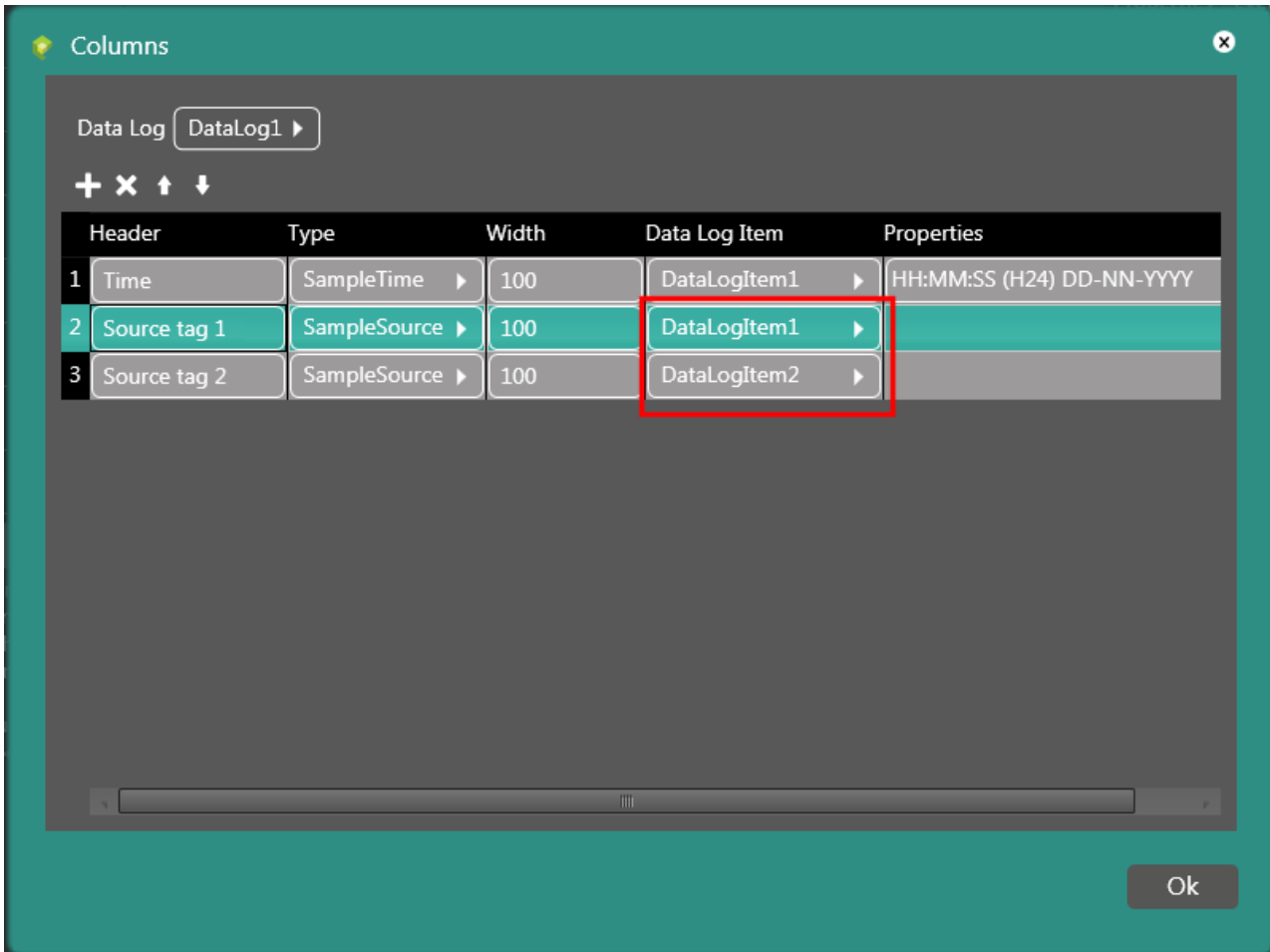
Width: the width of the table columns.

Data Log Item: the value associated to the Data Log.



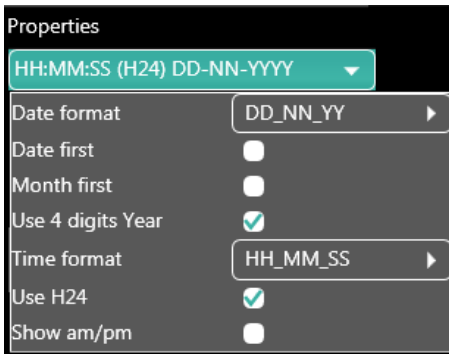
# CREW Manual

In the example the values of two Tags have been associated (Tag1 and Tag2), to be displayed in the respective "Source tag 1" and "Source tag 2" columns .



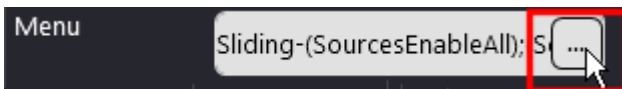
# CREW Manual

Property: to customise how datum acquisition is displayed (hour, minutes, seconds, etc.).



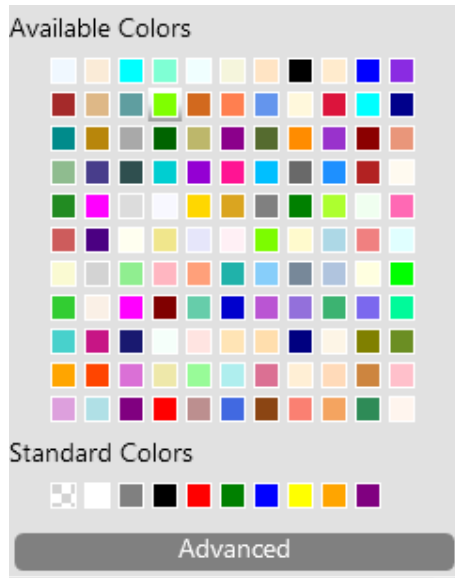
“Menu” option

Click the “Browse” key.

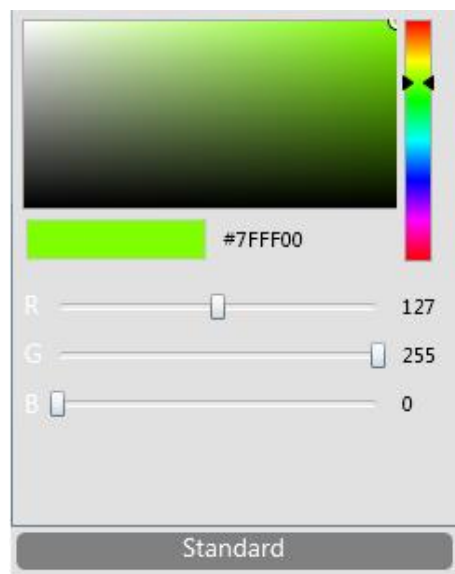


# CREW Manual

The properties related to colours can be edited through the colour palette.



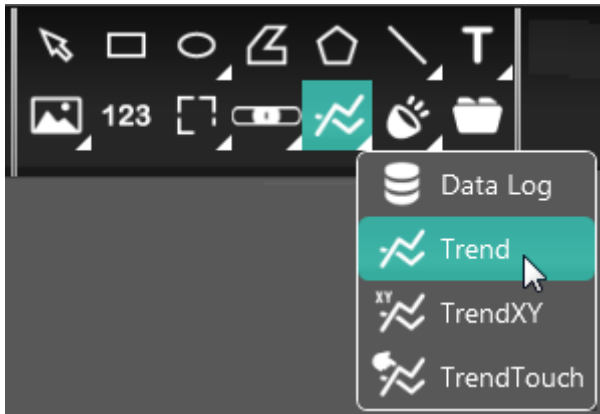
Click “Advanced” to select a colour using the RGB colour selection mask.



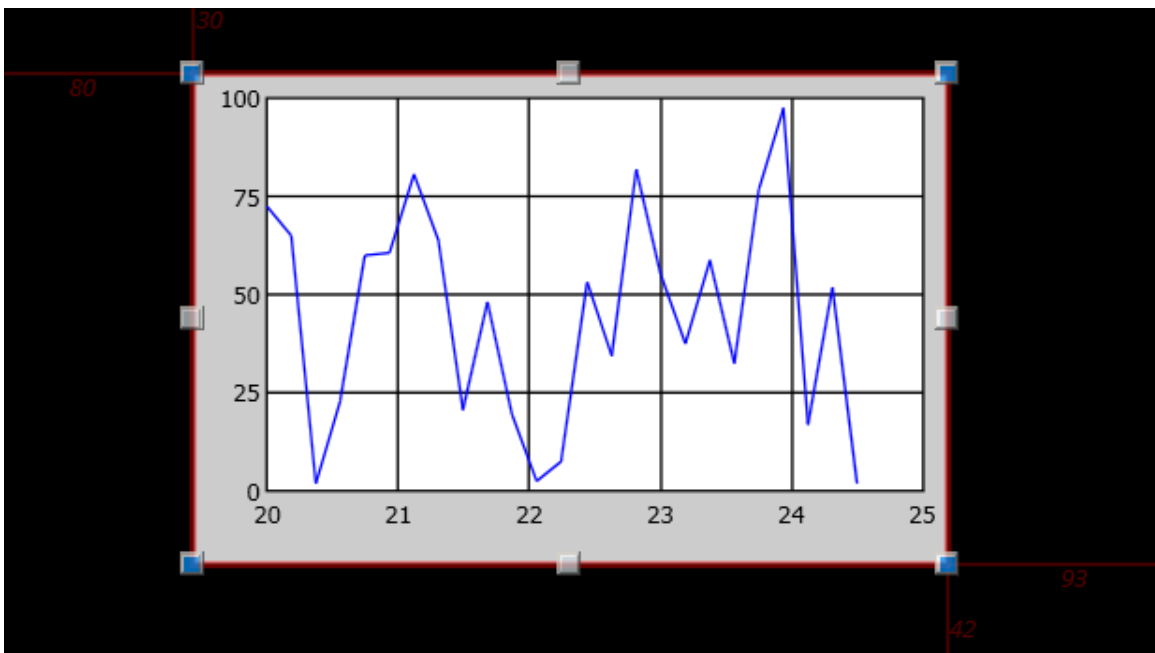


# CREW Manual

## Trend



The "Trend" icon on the "Graphics" menu is used to place a trend (or trend graph) on the page, drawing it with the mouse on the page.



To determine the features of the "Trend" field it is necessary to set them in the "Properties Editor", as shown in the "[Trend Properties](#)" section.

# CREW Manual

## Trend Properties

The following image illustrates all the editable properties of the Trend. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

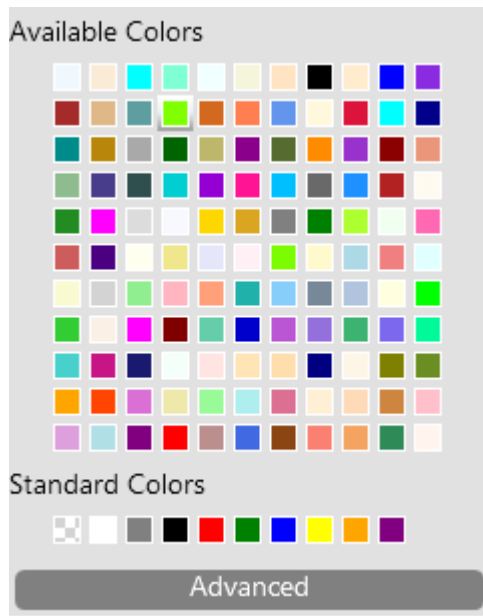
The following table describes all the editable properties of the Trend.

Properties	Description
<b>General</b>	
Name	Object ID
Left	Horizontal coordination of position
Top	Vertical coordination of position
Width	Width
Height	Height
<b>Behavior</b>	
Visibility	Determines whether the object should be displayed or not
Pens	It allows you to enter the page editor of "Trend pens" used to represent the chart.
TimeSpan	Length of time periods expressed in thousandths of a second. If the value 10000 is entered, for example, at any point the trend table will display the values gathered in 10 seconds
<b>Scale</b>	
FontName	Determines the font type used to represent the values of the X and Y graph
FontSize	Determines the font size used to represent the values of the X and Y graph
TextColor	Determines the color of the text used to represent the values of the X and Y graph
<b>Attributes</b>	
StrokeColor	Determines the stroke colour (edges of the chart); the colour is selectable by RGB colour code or colour palette
StrokeSize	Determines the stroke weight of the chart
Opacity	Determines the opacity of the chart
Menu	By clicking on the "Browse" menu option, you can make a Trend chart menu management
Color	Color of the Trend area
ChartFillColor	Background color of Trend chart
InternalBorder	It indicates the margin over the four sides of the page
MarginsAuto	It determines whether the size of the margins around the area of the chart is to be automatically calculated based on the characteristics of the enabled elements
ChartMargin	Indicates the margin respect on the four sides of the page (when "MarginsAuto" is disabled)
<b>Security</b>	
Level	Allows to define the levels of authentication required to control access to specific areas of the project
ShowProtected	Enables the displaying of a "status" icon in the object when it is protected for the current user

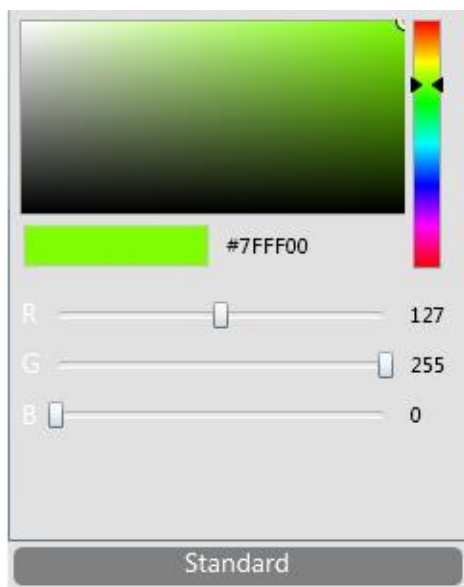
<b>FDA</b>	
Enable	Enables tracking of events related to the functions controlled through the grid menu
RequestReason	In each event to be recorded you have to insert the text that will be logged together with the time and date.
Comment	It defines a static comments to be tracked along with the execution events of the grid menu functions
<b>Advanced</b>	
VerticalScale	Allows to enable or not the presence in Runtime vertical scale of values
HorizontalScale	Allows to enable or not the presence in Runtime horizontal scale of values
VerticalBar	Allows to enable or disable the displaying in Runtime of a vertical bar of Reference "Range" values
HorizontalBar	Allows to enable or disable the displaying in Runtime of a horizontal bar of Reference "Range" values
OriginLabel	Allows to enable or not the presence in Runtime of the Date and Time field placed on the bottom side of the Trend
TraceName	Allows to enable or disable in runtime the displaying of the name of the data log associated with the pen
HasMarkers	Allows to enable or not the presence in Runtime of a Markers displaying line
HasLines	Allows to enable or not the presence in Runtime of a Trend line
HasArea	Allows to enable or not the presence in Runtime of a selection under the Trend line
IsDigitalLine	Allows to enable or not the presence in Runtime of the digital visualization of variations of the Trend chart line (On: immediate digital variation, OFF: gradual variation)
IsHistogram	Allows you to display a representation (weekly, monthly, yearly) on the chart of the average of the recorded values
GridMode	Allows to display the chart grid in the following ways: - None (the chart is not divided into cells) - Normal (the chart is divided into normal cells) - Thick: (each cell of the chart is divided into smaller cells)
TimeMode	Allows to change the viewing of time on the horizontal axis of the chart : - Normal: In runtime, it divides the scale according to the set values - Day: In runtime, it divides the scale into 24 hours - Week: In runtime, it divides the scale into 7 days - Month: In Runtime, it divides the scale into 31 days - Year: In Runtime, it divides the scale into 12 months
IndexedMode	It define a trend with indexed samples: the samples are equally spaced on the horizontal axis, which only shows the samples order number
CursorMode	It defines the type of measurements taken from the trend "slider" (whether single or multiple) and the displaying mode of the results

# CREW Manual

The properties related to colours can be edited through the colour palette.



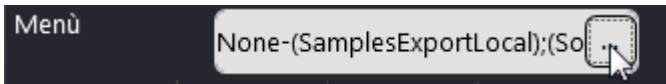
Click “Advanced” to select a colour using the RGB colour selection mask.



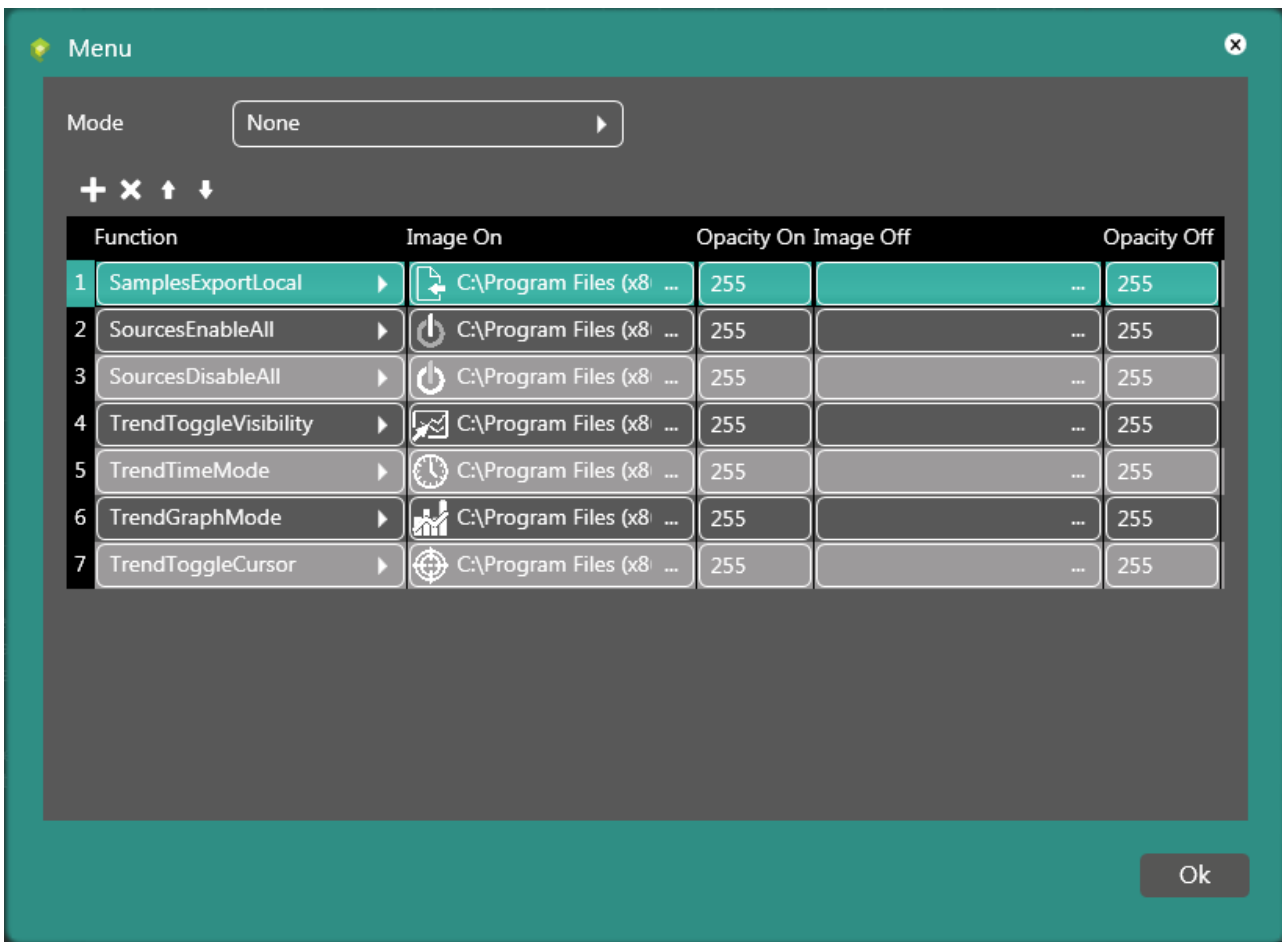
# CREW Manual

“Menu” option

Click the “Browse” key.

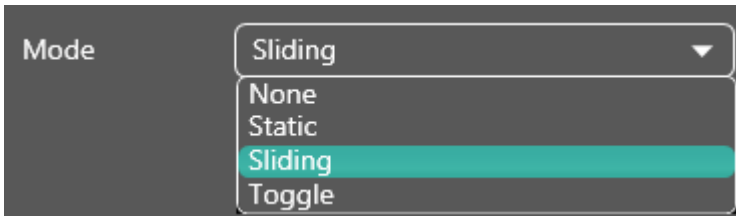


On the window that appears, it is possible to decide how to set the Runtime menu of the Trend.

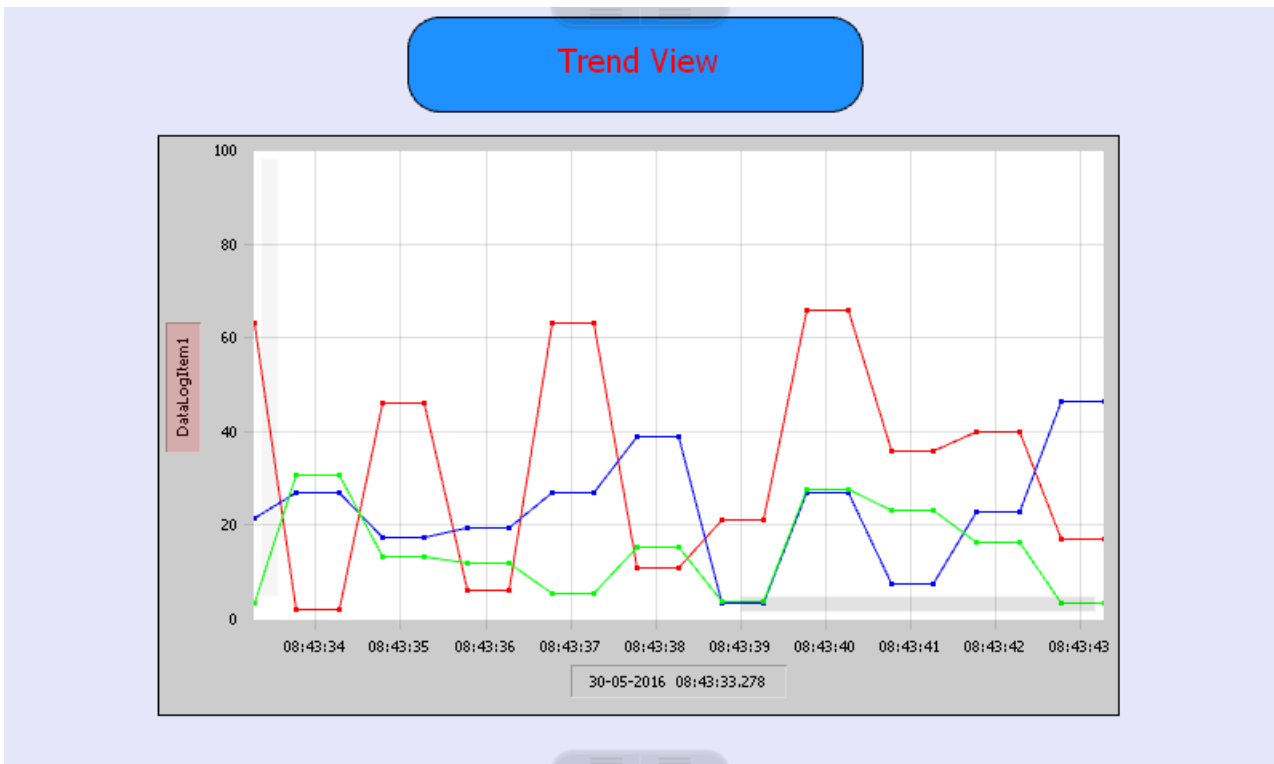


# CREW Manual

Mode: it is possible to customise the Runtime menu through one of the following options.

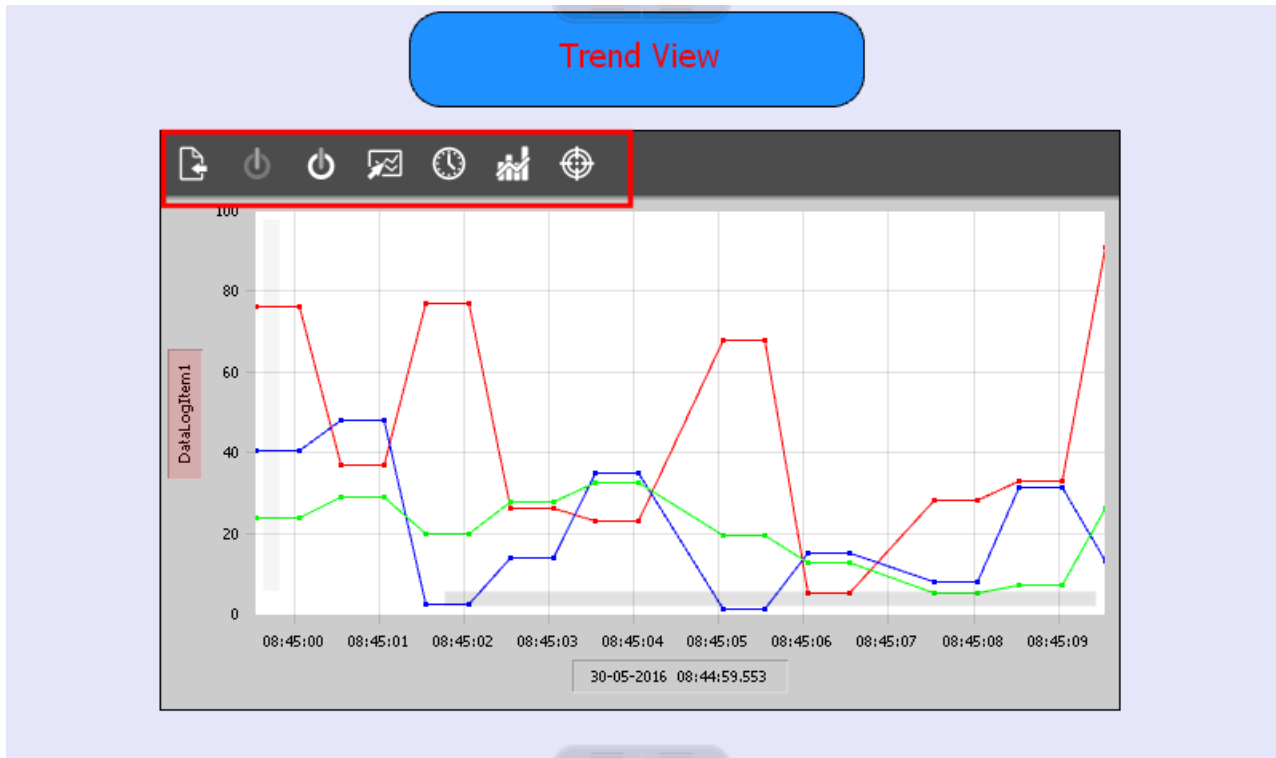


None: no Runtime menu. There are only the default columns.



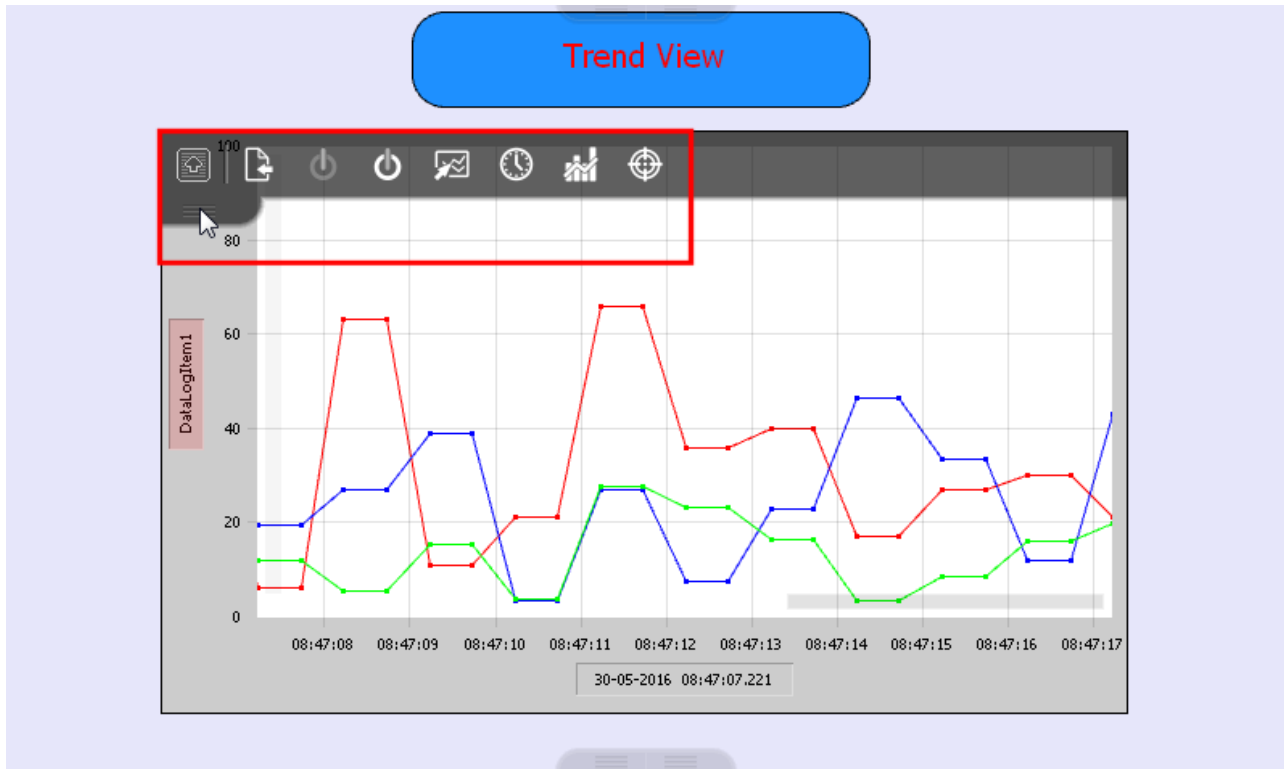
# CREW Manual

Static: static menu, namely fixed and always there.



# CREW Manual

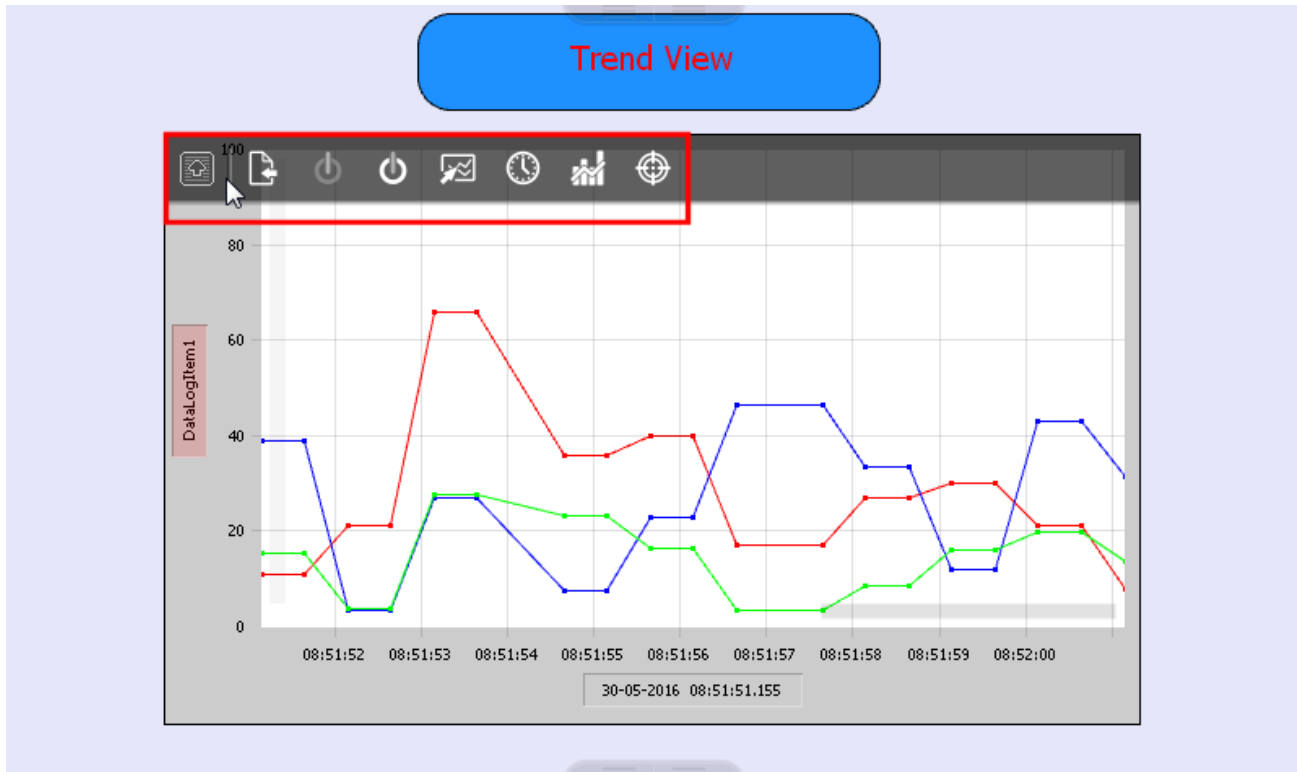
Sliding: floating menu, which appears at the user's discretion.





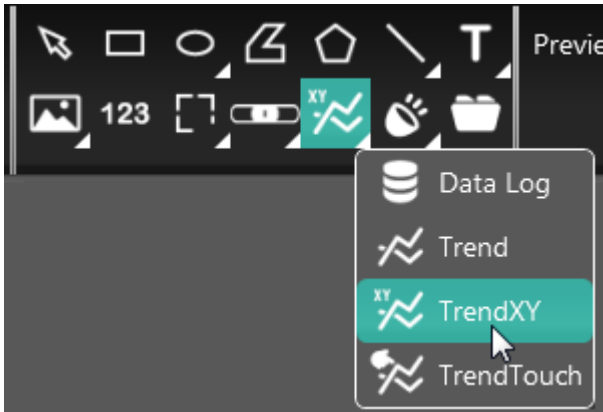
# CREW Manual

Toggle: floating menu (similar to the “Sliding” option), which appears at the user’s discretion.

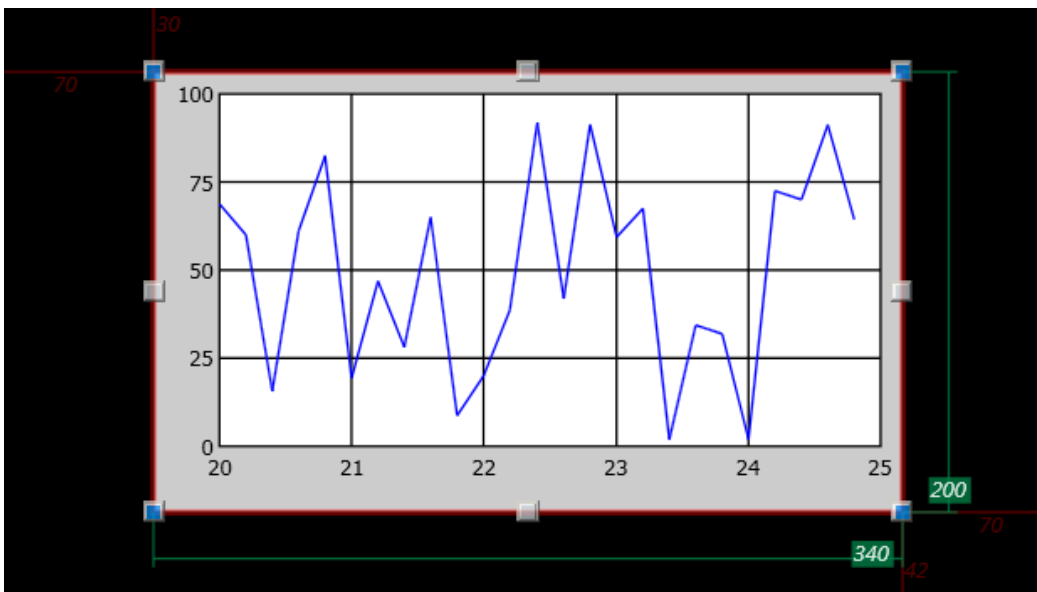


# CREW Manual

## Trend XY



The “TrendXY” icon on the “Graphics” menu is used to place a trendXY (or trendXY graph) on the page, drawing it with the mouse on the page.

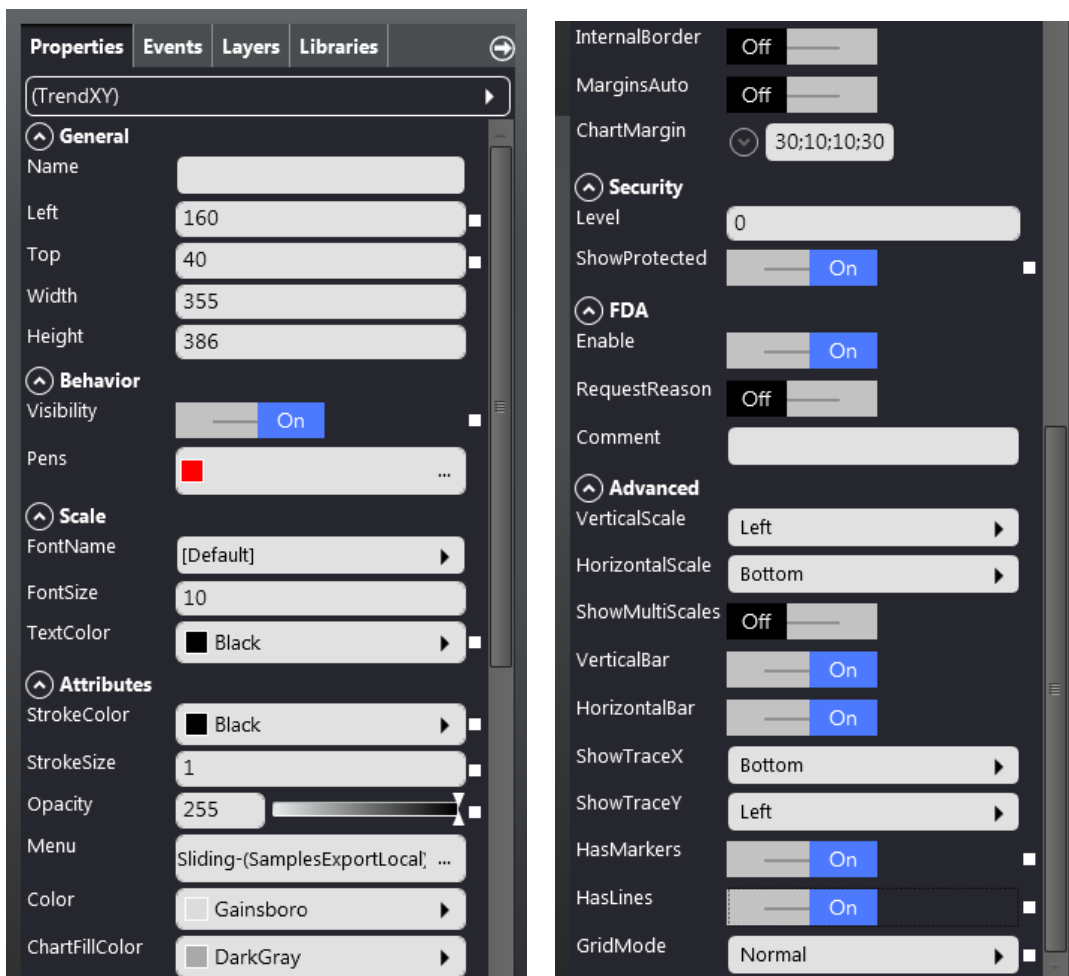


To determine the features of the “TrendXY” field it is necessary to set them in the “Properties Editor”, as shown in the “[TrendXY Properties](#)” section.

# CREW Manual

## TrendXY Properties

The following image illustrates all the editable properties of the TrendXY. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

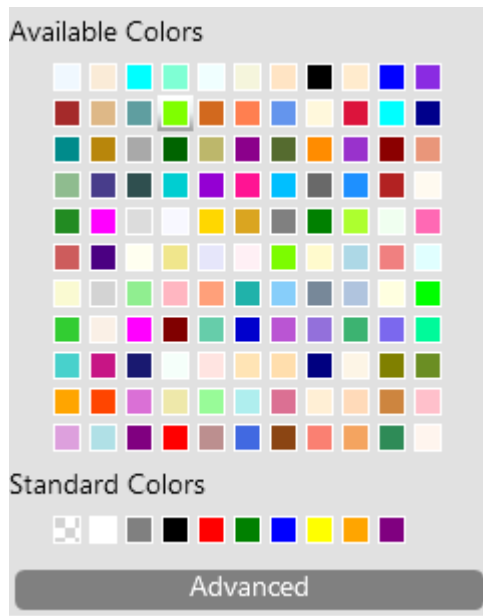
The following table describes all the editable properties of the Trend XY.

Properties	Description
<b>General</b>	
Name	Object ID
Left	Horizontal coordination of position
Top	Vertical coordination of position
Width	Width
Height	Height
<b>Behavior</b>	
Visibility	Determines whether the object should be displayed or not
Pens	It allows you to enter the page editor of "Trend pens" used to represent the chart.
<b>Scale</b>	
FontName	Determines the font type used to represent the values of the X and Y graph
FontSize	Determines the font size used to represent the values of the X and Y graph
TextColor	Determines the color of the text used to represent the values of the X and Y graph
<b>Attributes</b>	
StrokeColor	Determines the stroke colour (edges of the chart); the colour is selectable by RGB colour code or colour palette
StrokeSize	Determines the stroke weight of the chart
Opacity	Determines the opacity of the chart
Menu	By clicking on the "Browse" menu option, you can make a Trend chart menu management
Color	Color of the Trend area
ChartFillColor	Background color of Trend chart
MarginsAuto	It determines whether the size of the margins around the area of the chart is to be automatically calculated based on the characteristics of the enabled elements
ChartMargin	Indicates the margin respect on the four sides of the page (when "MarginsAuto" is disabled)

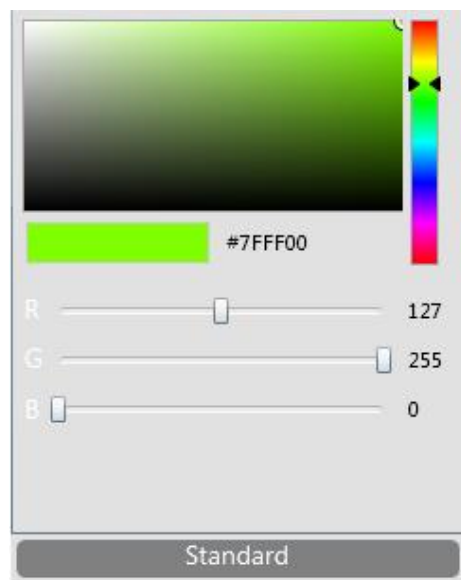
<b>Security</b>	
Level	Allows to define the levels of authentication required to control access to specific areas of the project
ShowProtected	Enables the displaying of a "status" icon in the object when it is protected for the current user
<b>FDA</b>	
Enable	Enables tracking of events related to the functions controlled through the grid menu
RequestReason	In each event to be recorded you have to insert the text that will be logged together with the time and date.
Comment	It defines a static comments to be tracked along with the execution events of the grid menu functions
<b>Advanced</b>	
VerticalScale	Allows to enable or not the presence in Runtime vertical scale of values
HorizontalScale	Allows to enable or not the presence in Runtime horizontal scale of values
ShowMultiscalses	Determines whether more pens scales to be displayed at the same time around the chart, instead of just the active pen scale
VerticalBar	Allows to enable or disable the displaying in Runtime of a vertical bar of Reference "Range" values
HorizontalBar	Allows to enable or disable the displaying in Runtime of a horizontal bar of Reference "Range" values
ShowTraceX	Allows to enable or disable the Runtime displaying of the data log name associated with the pen of the X axis
ShowTraceY	Allows to enable or disable the Runtime displaying of the data log name associated with the pen of the Y axis
HasMarkers	Allows to enable or not the presence in Runtime of a Markers displaying line
HasLines	Allows to enable or not the presence in Runtime of a XY Trend line
GridMode	Allows to display the chart grid in the following ways: - None (the chart is not divided into cells) - Normal (the chart is divided into normal cells) - Thick; (each cell of the chart is divided into smaller cells)

# CREW Manual

The properties related to colours can be edited through the colour palette.

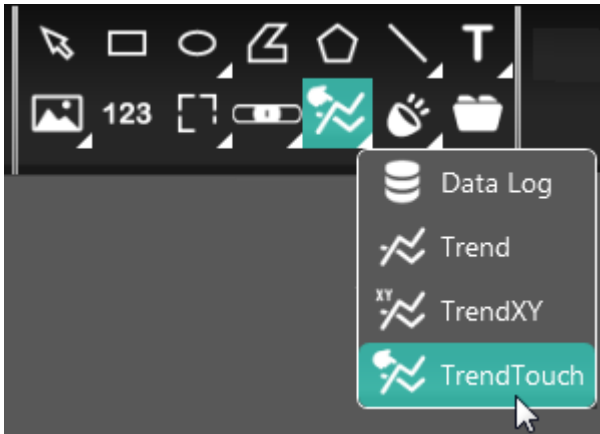


Click “Advanced” to select a colour using the RGB colour selection mask.

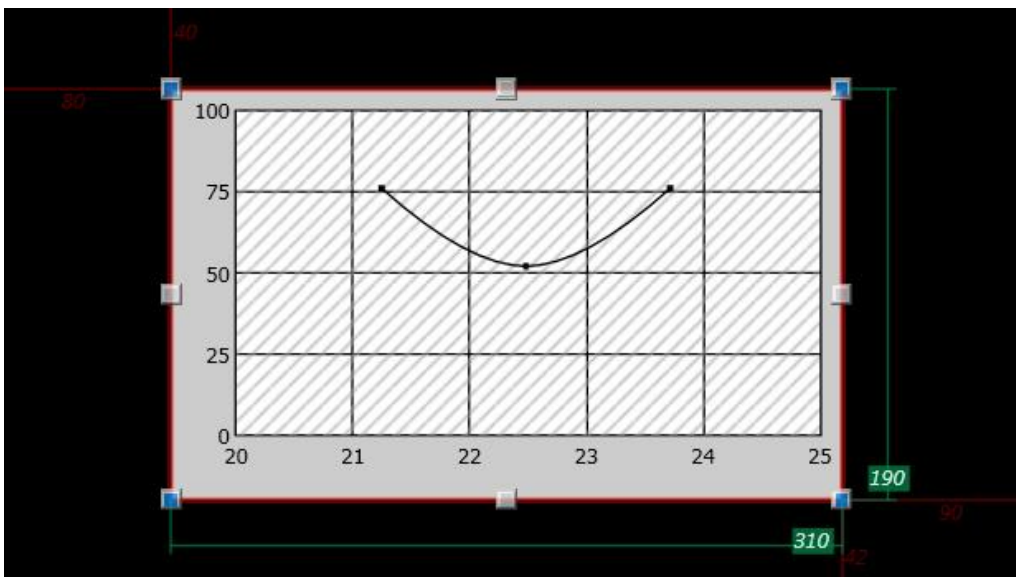


# CREW Manual

## Touch Trend



The “TouchTrend” icon on the “Graphics” menu is used to place a touch trend (or touch trend graph) on the page, drawing it with the mouse on the page.



To determine the features of the “Touch Trend” field it is necessary to set them in the “Properties Editor”, as shown in the “Touch Trend Properties” section.

# CREW Manual

## Touch Trend Properties

The following image illustrates all the editable properties of the Touch Trend. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

The following table describes all the editable properties of the Touch Trend.

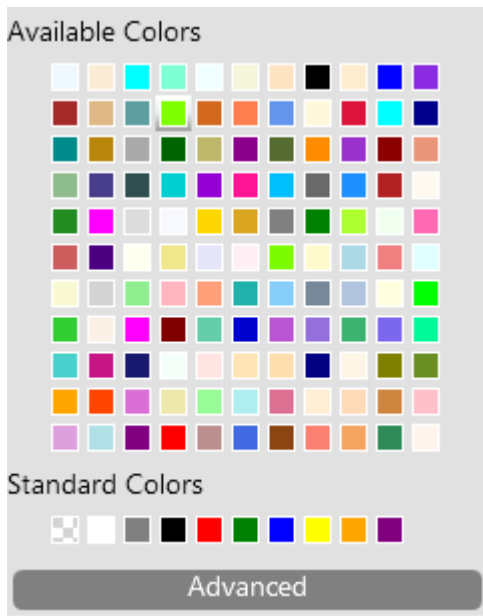
Properties	Description
<b>General</b>	
Name	Object ID
Left	Horizontal coordination of position
Top	Vertical coordination of position
Width	Width
Height	Height
<b>Behavior</b>	
Visibility	Determines whether the object should be displayed or not
Pens	It allows you to enter the page editor of "Trend TOUCH pens" used to represent the chart.
<b>Scale</b>	
FontName	Determines the font type used to represent the values of the X and Y graph
FontSize	Determines the font size used to represent the values of the X and Y graph
TextColor	Determines the color of the text used to represent the values of the X and Y graph
<b>Attributes</b>	
StrokeColor	Determines the stroke colour (edges of the chart); the colour is selectable by RGB colour code or colour palette
StrokeSize	Determines the stroke weight of the chart
Opacity	Determines the opacity of the chart
Menu	By clicking on the "Browse" menu option, you can make a Trend chart menu management
Color	Color of the Trend Touch area
ChartFillColor	Background color of Trend Touch chart
MarginsAuto	It determines whether the size of the margins around the area of the chart is to be automatically calculated based on the characteristics of the enabled elements
ChartMargin	Indicates the margin respect on the four sides of the page (when "MarginsAuto" is disabled)
<b>TrendTouch</b>	
TouchStrokeColor	Determines the stroke colour of the Trend Touch line, the colour is selectable by RGB colour code or colour palette
TouchStrokeSize	Determines the stroke weight of the Trend Touch line
LinkMode	Allows to change the displaying mode of the connection points on the chart in the following ways: - None: the points of the chart are not linked together - Segments : the points of the chart are linked together by segments - Bezier : the points of the chart are linked together by a continuous curved line named "Bezier curve" (see "Insights on Bezier Curve" section) - HorLevels : the points of the chart are linked together by a continuous curved line that becomes horizontal close to the points themselves - VertLevels : the points of the chart are linked together by a continuous curved line that becomes vertical close to the points themselves

GridThicknessX	Indicates, at runtime, the step (value expressed in pixels) of the shift along the X axis for each point of the chart line
GridThicknessY	Indicates, at runtime, the step (value expressed in pixels) of the shift along the Y axis for each point of the chart line
EnableDragX	Allows the enabling or less, in Runtime, the manual shifting of the points along the X axis
EnableDragY	Allows the enabling or less, in Runtime, the manual shifting of the points along the Y axis
Points	Default number of dots from which passes the line of the chart; they can be manually moved in the runtime obtaining the desired graph, to do this, the "EnableDragX" and "EnableDragY" properties must be enabled
<b>Blink</b>	
CanBlink	It allows you to enable or disable the Runtime blink
Visibility	Allows to display or hide the blinking in the Runtime
TouchStroke	Allows to displaying or not the blinking of the trace colour in the Runtime
TouchStrokeColor	Allows to define the colour of the trace, it is selectable by RGB colour code or colour palette
<b>Security</b>	
Level	Allows to define the levels of authentication required to control access to specific areas of the project
ShowDisabled	Enables the displaying of a "status" icon in the object when its use has been disabled
ShowProtected	Enables the displaying of a "status" icon in the object when it is protected for the current user
<b>FDA</b>	
Enable	Enables tracking of events related to the functions controlled through the grid menu
RequestReason	In each event to be recorded you have to insert the text that will be logged together with the time and date.
Comment	It defines a static comments to be tracked along with the execution events of the grid menu functions
<b>Advanced</b>	
VerticalScale	Allows to enable or not the presence of the vertical scale of values in the Runtime
HorizontalScale	Allows to enable or not the presence of the horizontal scale of values in the Runtime
VerticalBar	Allows to enable or disable the displaying in Runtime of a vertical bar of Reference "Range" values
HorizontalBar	Allows to enable or disable the displaying in Runtime of a horizontal bar of Reference "Range" values
ShowTraceX	Allows to enable or disable the Runtime displaying of the data log name associated with the pen of the X axis
ShowTraceY	Allows to enable or disable the Runtime displaying of the data log name associated with the pen of the Y axis
HasMarkers	Allows to enable or not the presence in Runtime of a Markers displaying line
HasLines	Allows to enable or not the presence in Runtime of a Trend Touch line
GridMode	Allows to display the chart grid in the following ways: - None (the chart is not divided into cells) - Normal (the chart is divided into normal cells) - Thick: (each cell of the chart is divided into smaller cells)

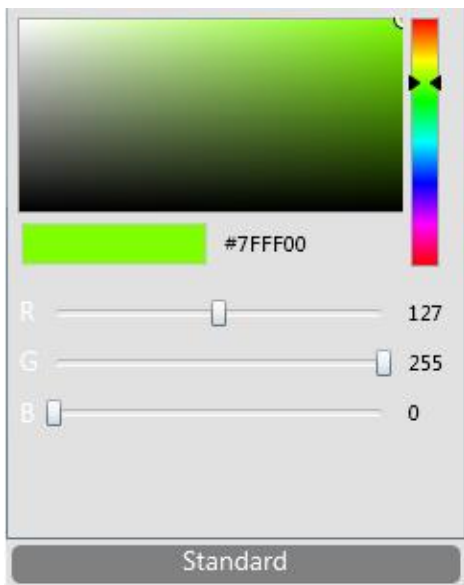


# CREW Manual

The properties related to colours can be edited through the colour palette.



Click "Advanced" to select a colour using the RGB colour selection mask.



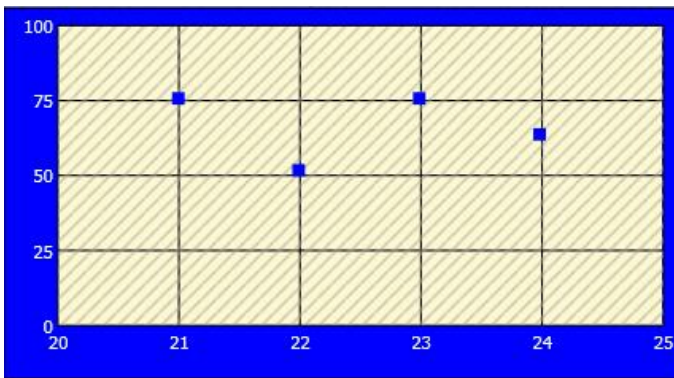
# CREW Manual

## LinkMode

The "LinkMode" property can be set as shown in the image.



- None: the points on the graph are not joined.

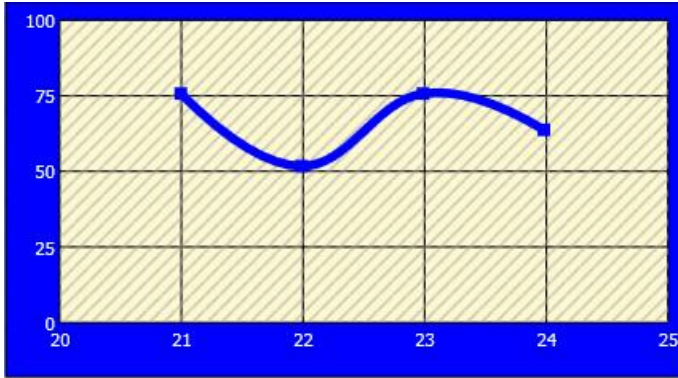


- Segments: the points on the graph are joined by segments.

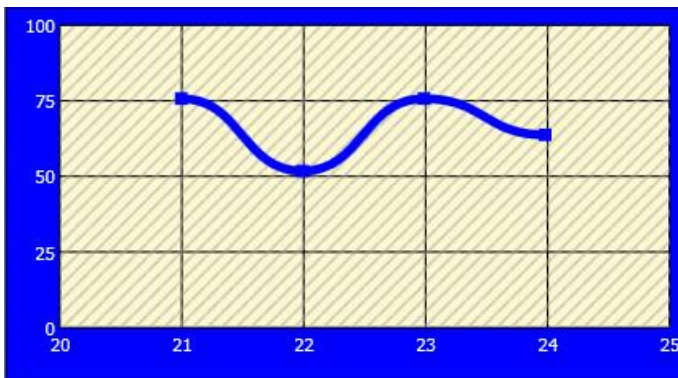


# CREW Manual

- Bezier: the points on the graph are joined by a continuous curved line called "Bezier Curve" (see "[Details on Bezier Curves](#)" section).



- HorLevels: the points on the graph are joined by a continuous curved line which flattens horizontally around the points.



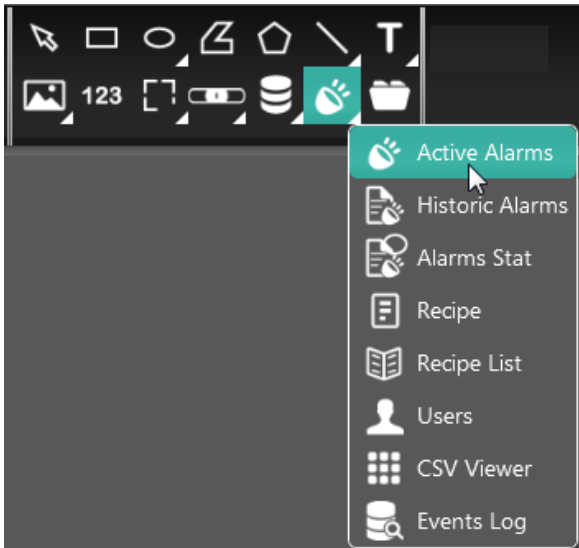
# CREW Manual

- VertLevels: the points on the graph are joined by a continuous curved line which flattens vertically around the points.

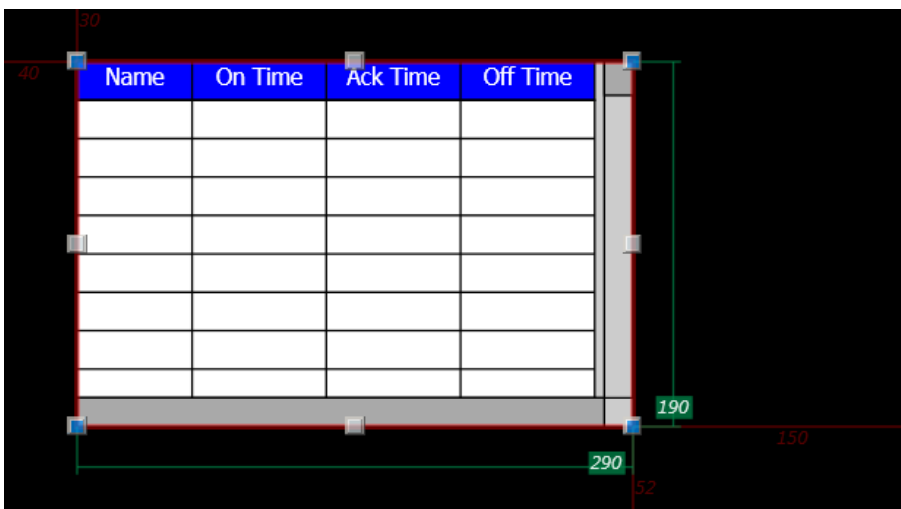


# CREW Manual

## Active Alarms



The “Active Alarms” icon on the “Graphics” menu is used to place an active alarms table on the page, drawing it with the mouse on the page.



The screenshot shows a table with a header row and several empty rows. The header row has four columns: 'Name', 'On Time', 'Ack Time', and 'Off Time'. The table is overlaid with a red border and green dimension lines. The dimensions are: 40 (top-left corner), 130 (top edge), 150 (right edge), 190 (bottom-right corner), 290 (bottom edge), and 52 (bottom-right corner).

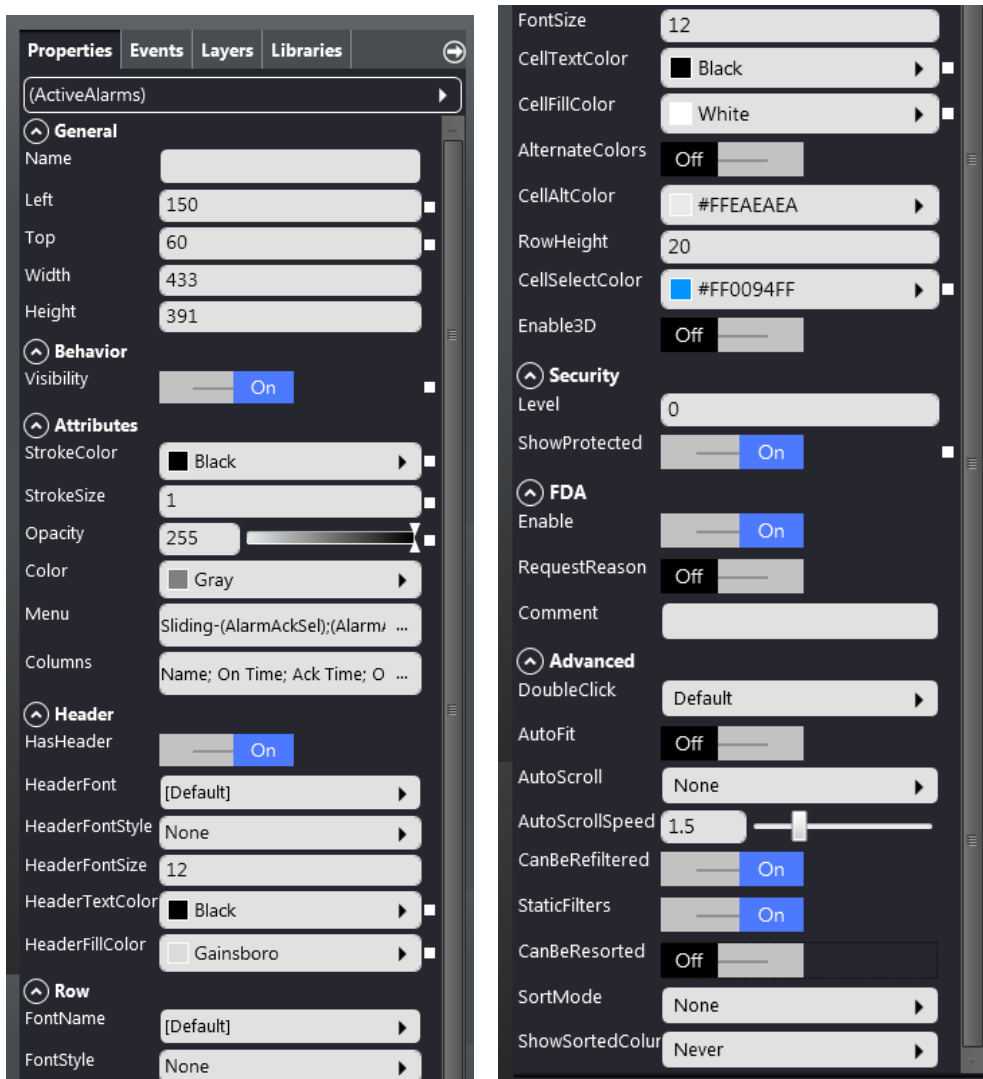
Name	On Time	Ack Time	Off Time

To determine the features of the “Active Alarms” table, set them in the “Properties Editor”, as shown in the section [“Active Alarms Properties”](#).

# CREW Manual

## Active Alarms Properties

The following image illustrates all the editable properties of the Active Alarms table. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



The screenshot displays the configuration interface for the Active Alarms table, organized into several sections:

- General:** Name, Left (150), Top (60), Width (433), Height (391).
- Behavior:** Visibility (On).
- Attributes:** StrokeColor (Black), StrokeSize (1), Opacity (255), Color (Gray), Menu (Sliding-(AlarmAckSel);(Alarmv ...), Columns (Name; On Time; Ack Time; O ...).
- Header:** HasHeader (On), HeaderFont ([Default]), HeaderFontStyle (None), HeaderFontSize (12), HeaderTextColor (Black), HeaderFillColor (Gainsboro).
- Row:** FontName ([Default]), FontStyle (None).
- Appearance:** FontSize (12), CellTextColor (Black), CellFillColor (White), AlternateColors (Off), CellAltColor (#FFEAEEA), RowHeight (20), CellSelectColor (#FF0094FF), Enable3D (Off).
- Security:** Level (0), ShowProtected (On).
- FDA:** Enable (On), RequestReason (Off), Comment.
- Advanced:** DoubleClick (Default), AutoFit (Off), AutoScroll (None), AutoScrollSpeed (1.5), CanBeRefiltered (On), StaticFilters (On), CanBeResorted (Off), SortMode (None), ShowSortedColour (Never).

# CREW Manual

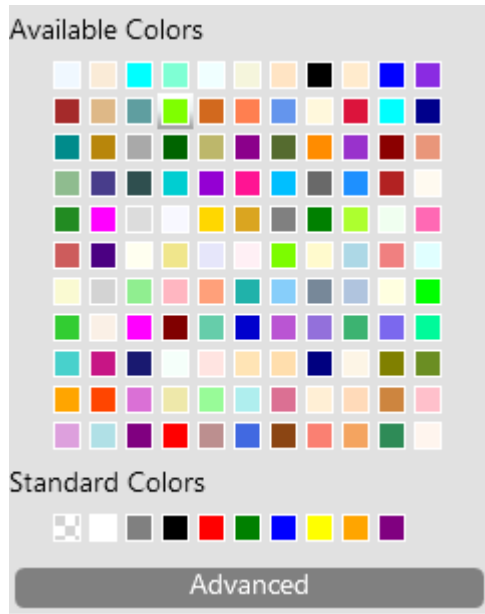
The following table describes all the editable properties of the Active Alarms table.

Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>UserMode</b>	Determines whether the user can interact with the object: - read only, the user cannot interact with the object; this value takes precedence over the Disabled property - read-write, the user can interact with the object; the effective interaction is validated by the Disabled property
<b>Attributes</b>	
<b>StrokeColor</b>	Determines the fill colour of the object; the colour is selectable by RGB colour code or colour palette
<b>StrokeSize</b>	Determines the stroke colour (edges of the figure); the colour is selectable by RGB colour code or colour palette
<b>Opacity</b>	Determines the opacity of the object
<b>Color</b>	Allows you to assign the color to the object using the color bar (Hue) (valid for objects in the Library)
<b>Menu</b>	By clicking on the "Browse" menu option, you can make an Active Alarms menu management of the table in Runtime as described in this section
<b>Columns</b>	Clicking on the "Browse" button of the "Columns" option, you can determine which columns to appear in the table and define their properties (as described in this section)
<b>Header</b>	
<b>HasHeader</b>	Determines whether the view of the Active Alarm View must have the header or not
<b>HeaderFont</b>	Determines the font type used to display the header text
<b>HeaderFontStyle</b>	Header font style. Any combination of the following features: - None - Italic - Bold - Underline
<b>HeaderFontSize</b>	Indicates the font size of the values written in the headers
<b>HeaderTextColor</b>	Determines the color of the header text
<b>HeaderFillColor</b>	Determines the color of the cell that contains the header
<b>Row</b>	
<b>FontName</b>	Determines the font used for the items of Active Alarms

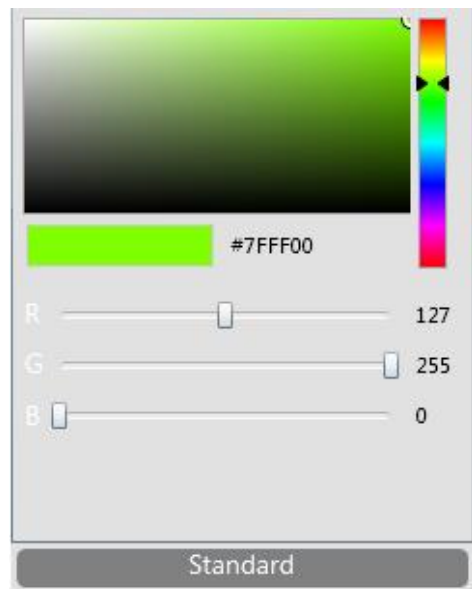
<b>FontStyle</b>	Font style. Any combination of the following features: - None - Italic - Bold - Underline
<b>FontSize</b>	Determines the stroke of Active Alarms view
<b>CellTextColor</b>	Represents the color of the writing cells
<b>CellFillColor</b>	Determines the color of the cells of table columns
<b>AlternateColors</b>	Allows you to assign two alternating colors for each row in the table
<b>CellAltColor</b>	Determines the alternative color (active if the option "AlternateColors" is "ON")
<b>RowHeight</b>	Determines the height of the row of the table (pixel)
<b>CellSelectColor</b>	Determines the color of the selected cell
<b>Enable3D</b>	Enable the 3D view ("embossed" view) of the table
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project
<b>ShowProtected</b>	Enables the displaying of a "status" icon in the object when it is protected for the current user
<b>FDA</b>	
<b>Enable</b>	Enables tracking of events related to the functions controlled through the grid menu
<b>RequestReason</b>	In each event to be recorded you have to insert the text that will be logged together with the time and date.
<b>Comment</b>	It defines a static comments to be tracked along with the execution events of the grid menu functions
<b>Advanced</b>	
<b>DoubleClick</b>	It allows you to assign a function to "double click" on each row of the table selecting it from the available ones
<b>AutoFit</b>	It determines whether the size of the columns of the grid are automatically calculated to fit precisely in the total available size
<b>AutoScroll</b>	It determines whether the content of the grid should automatically scroll
<b>AutoScrollSpeed</b>	It defines the scroll speed (when "AutoScroll" is enabled)
<b>CanBeRefiltered</b>	It determines whether the grid filters can be changed at runtime
<b>StaticFilters</b>	It defines the mode of interaction with the grid for access to the filter definition
<b>CanBeResorted</b>	It determines whether the grid lines can be ordered at runtime
<b>SortMode</b>	It defines the policy of the default grid sorting
<b>ShowSortedColumn</b>	It determines whether the column identified as sorting order must be highlighted in the grid

# CREW Manual

The properties related to colours can be edited through the colour palette.



Click “Advanced” to select a colour using the RGB colour selection mask.

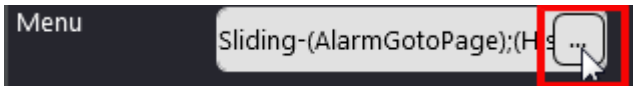




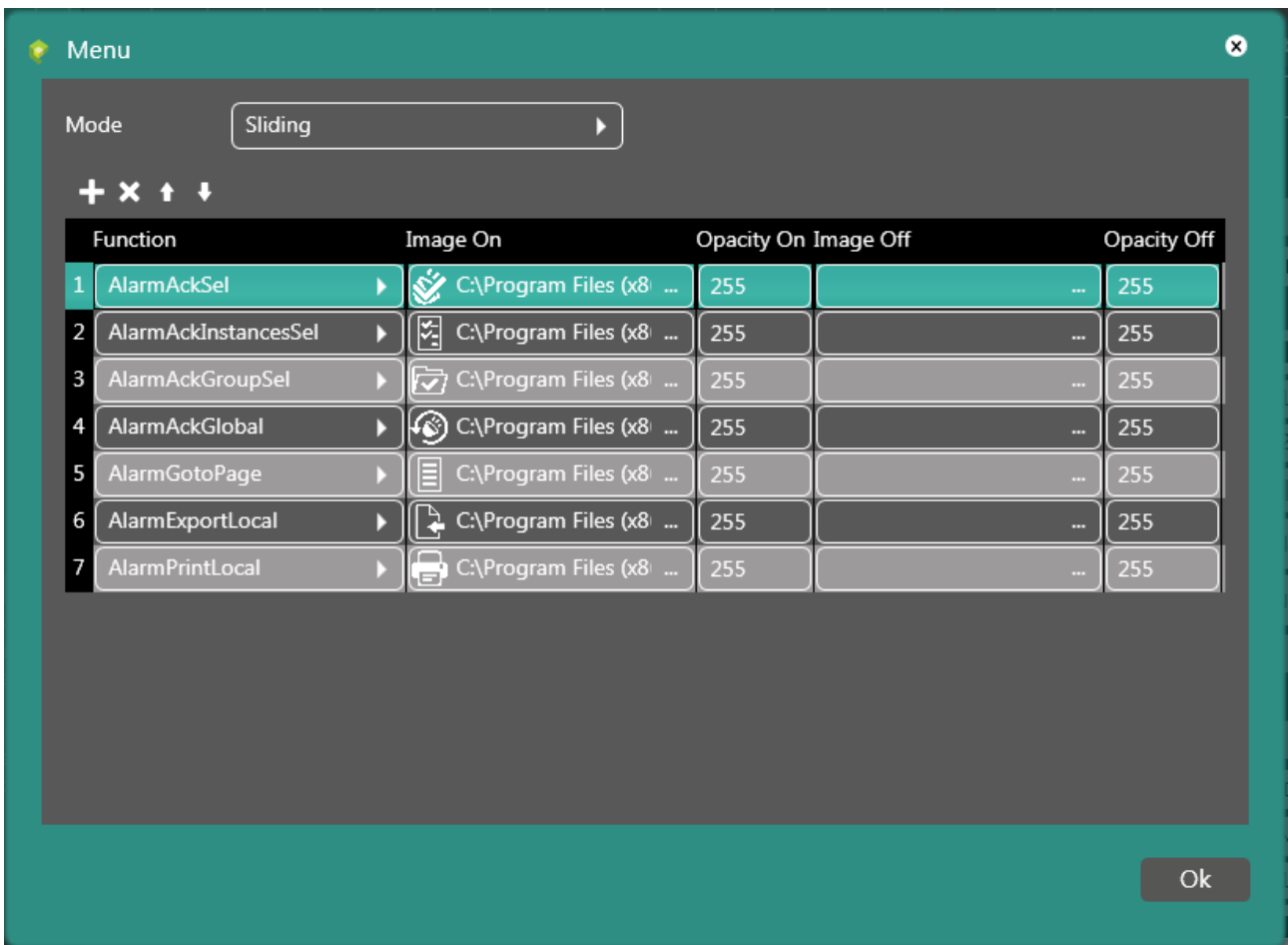
# CREW Manual

“Menu” option

Click the “Browse” key.

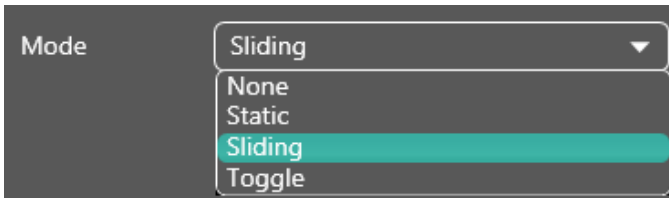


On the window that appears, it is possible to decide how to set the Runtime menu of the Active Alarms table.



# CREW Manual

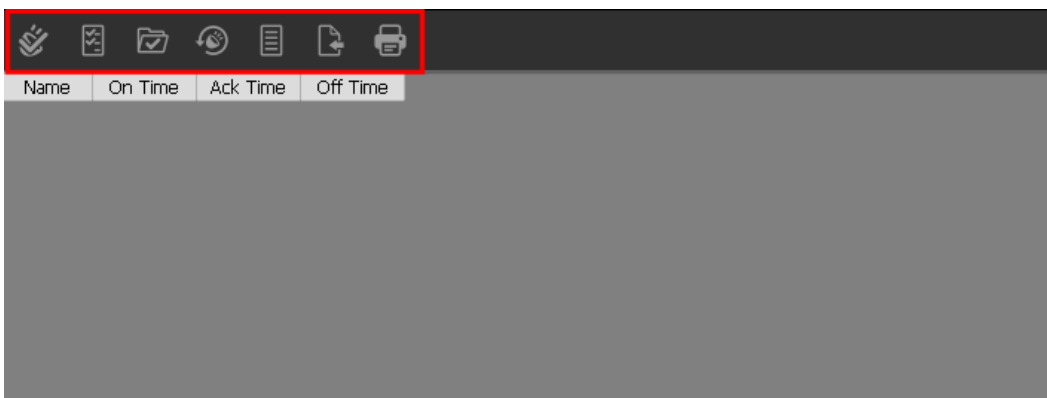
Mode: it is possible to customise the Runtime menu through one of the following options.



None: no Runtime menu. There are only the default columns.



Static: static menu, namely fixed and always there.



# CREW Manual

Sliding: floating menu, which appears at the user's discretion.



Toggle: floating menu (similar to the “Sliding” option), which appears at the user's discretion.



Using the icons displayed in the image to add (+) or eliminate (X) new functions and move them up or down.



# CREW Manual

Function: this is used to select a function (from those shown in the image) for each of the icons that compose the Alarms view menu.

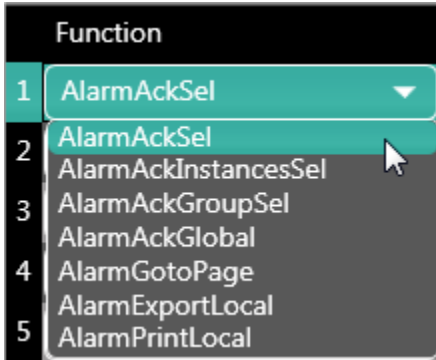
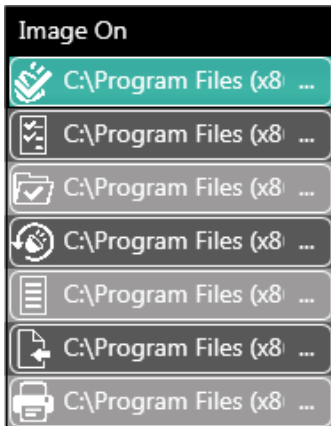


Table of functions that can be associated to the Active Alarms view.

Function	Description
<b>AlarmAckSel</b>	Acknowledge the alarm instance selected in grid
<b>AlarmAckInstancesSel</b>	Acknowledge all the instances of the same alarm selected in grid
<b>AlarmAckGroupSel</b>	Acknowledge all the instances of alarms of the same group of the alarm selected in grid
<b>AlarmAckGlobal</b>	Acknowledge all existing active alarms
<b>AlarmGotoPage</b>	Show the page associated to the alarm selected in grid (works for both active and historical alarms)
<b>AlarmExportLocal</b>	Export all active alarms on file at RUNTIME; the name of the destination file (CSV format) is needed. The export is performed on the machine where the user interface is active.
<b>AlarmPrint</b>	Print all alarm records
<b>AlarmPrintLocal</b>	Print all alarm records (a dialog box allows the selection of the target printer)

# CREW Manual

Image On: to associate an image to the icon when it is active.

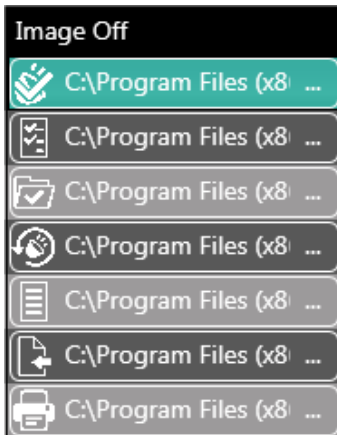


Opacity On: to set the level of opacity of the image to be associated with the icon when it is active.



# CREW Manual

Image Off: to associate an image to the icon when it is not active.

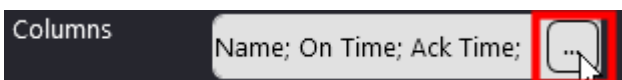


Opacity Off: to set the level of opacity of the image to be associated with the icon when it is not active.



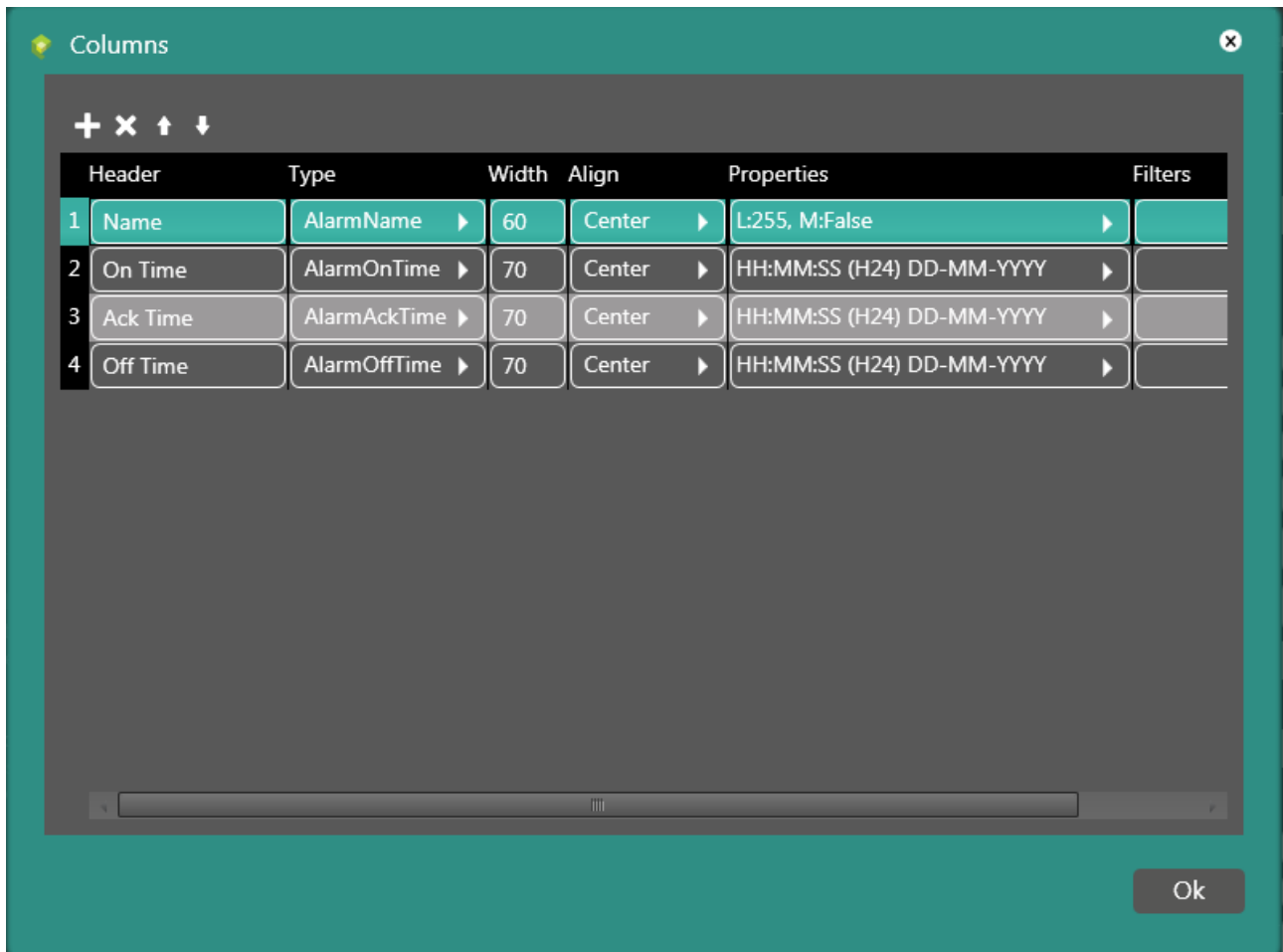
“Columns” option

Click the “Browse” key.



# CREW Manual

On the window that appears, it is possible to decide how to set the Active Alarms table.



Using the icons displayed in the image to add (+) or eliminate (X) new functions and move them up or down.



Heading: the title of the columns that comprise the table.

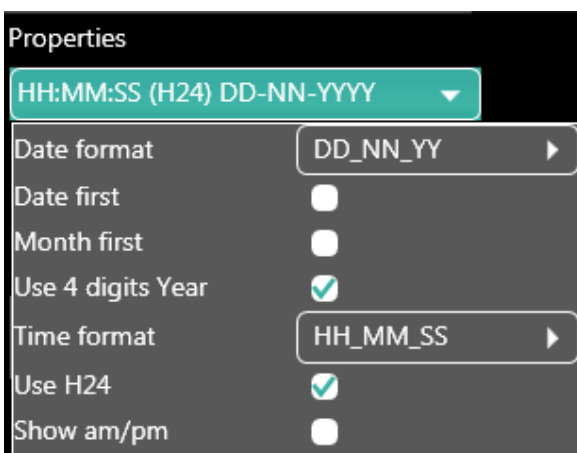
# CREW Manual

Type: this identifies the type of Alarm, which can be selected from the options shown in the image.



Width: the width of the table columns.

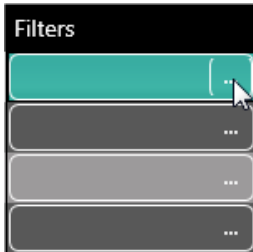
Property: to customise how datum acquisition is displayed (hour, minutes, seconds, etc.).





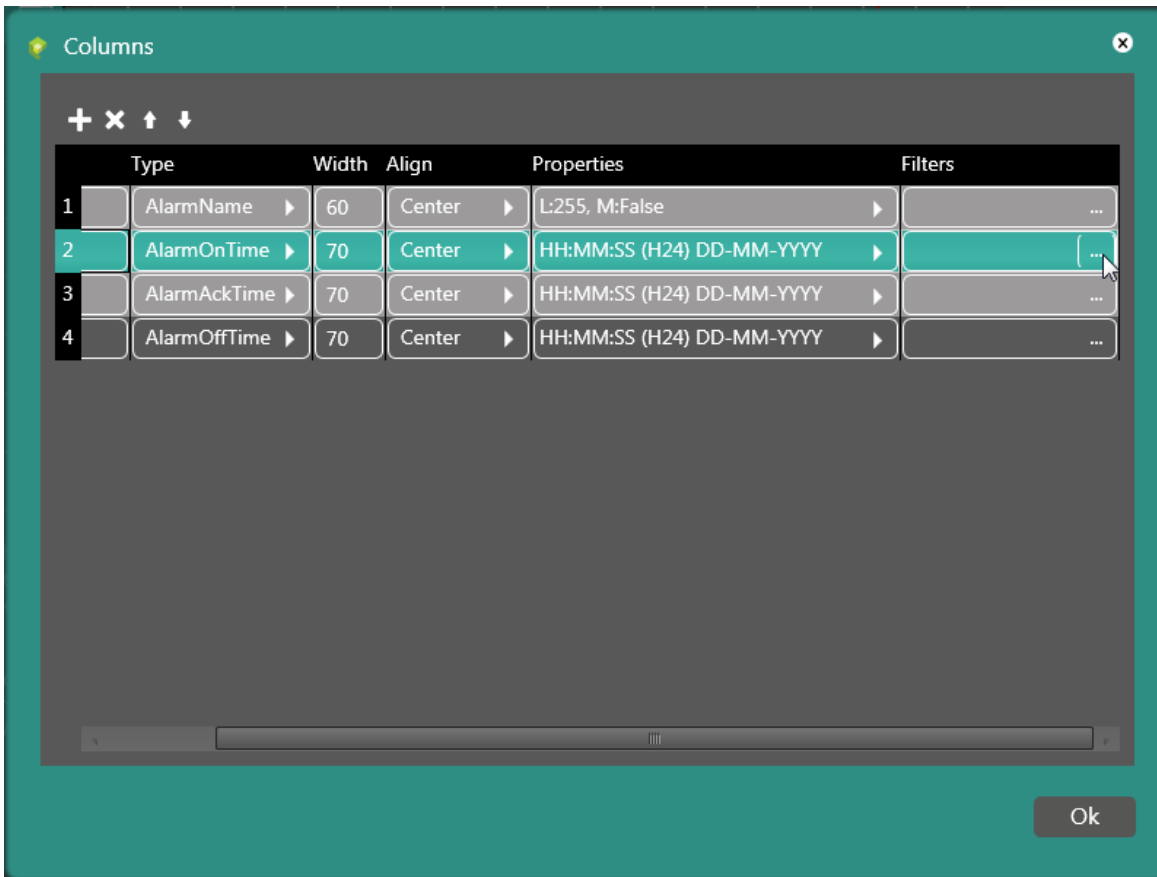
# CREW Manual

## Filters



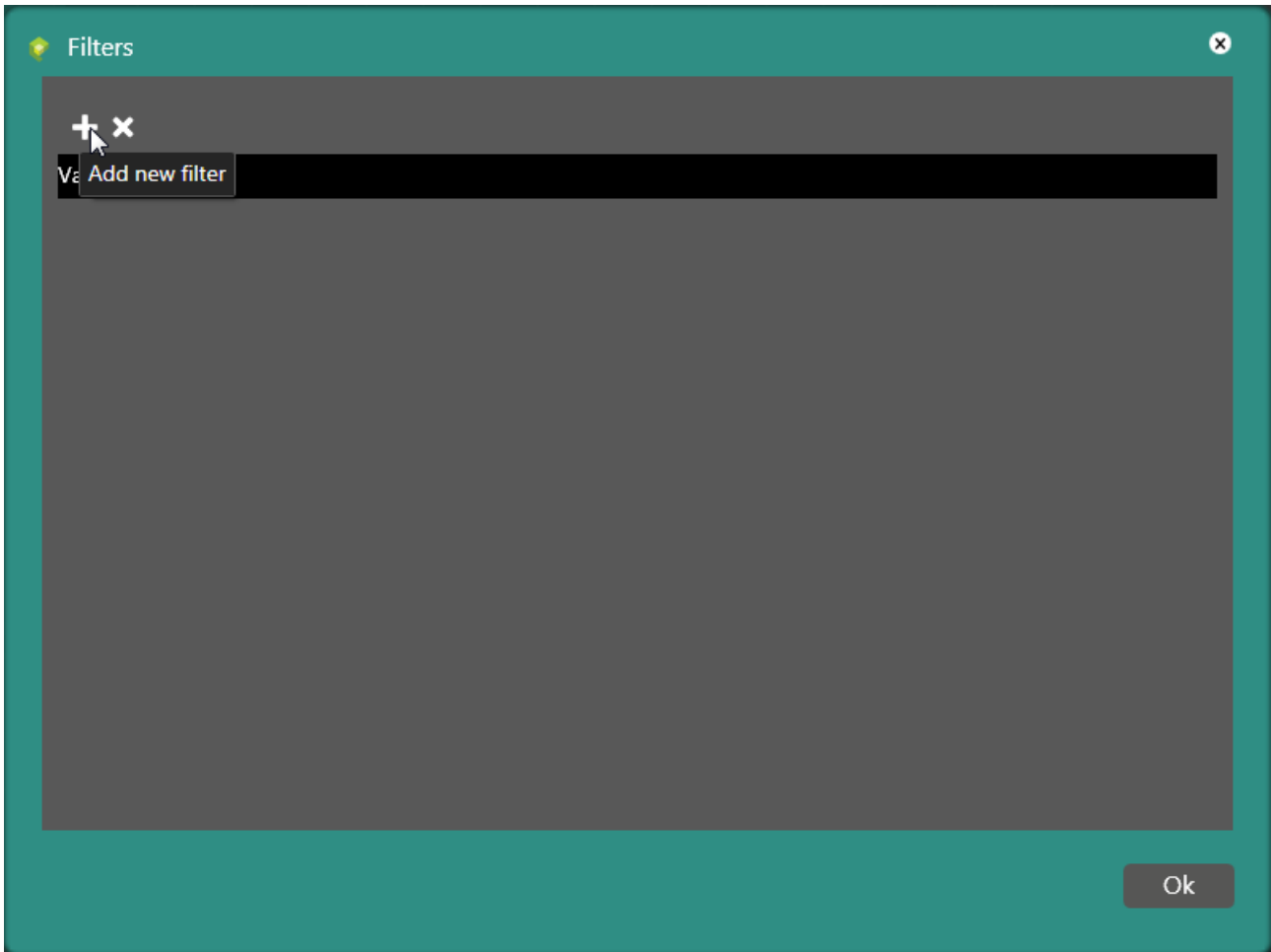
With the Filters option it is possible to enable Runtime display filters for the alarm log. For example, it is possible to associate a filter to the "AlarmOnTime" function to display only the alarms within a given period of time (chosen by the user).

To enable a filter, click "Browse" in the "Filters" column.



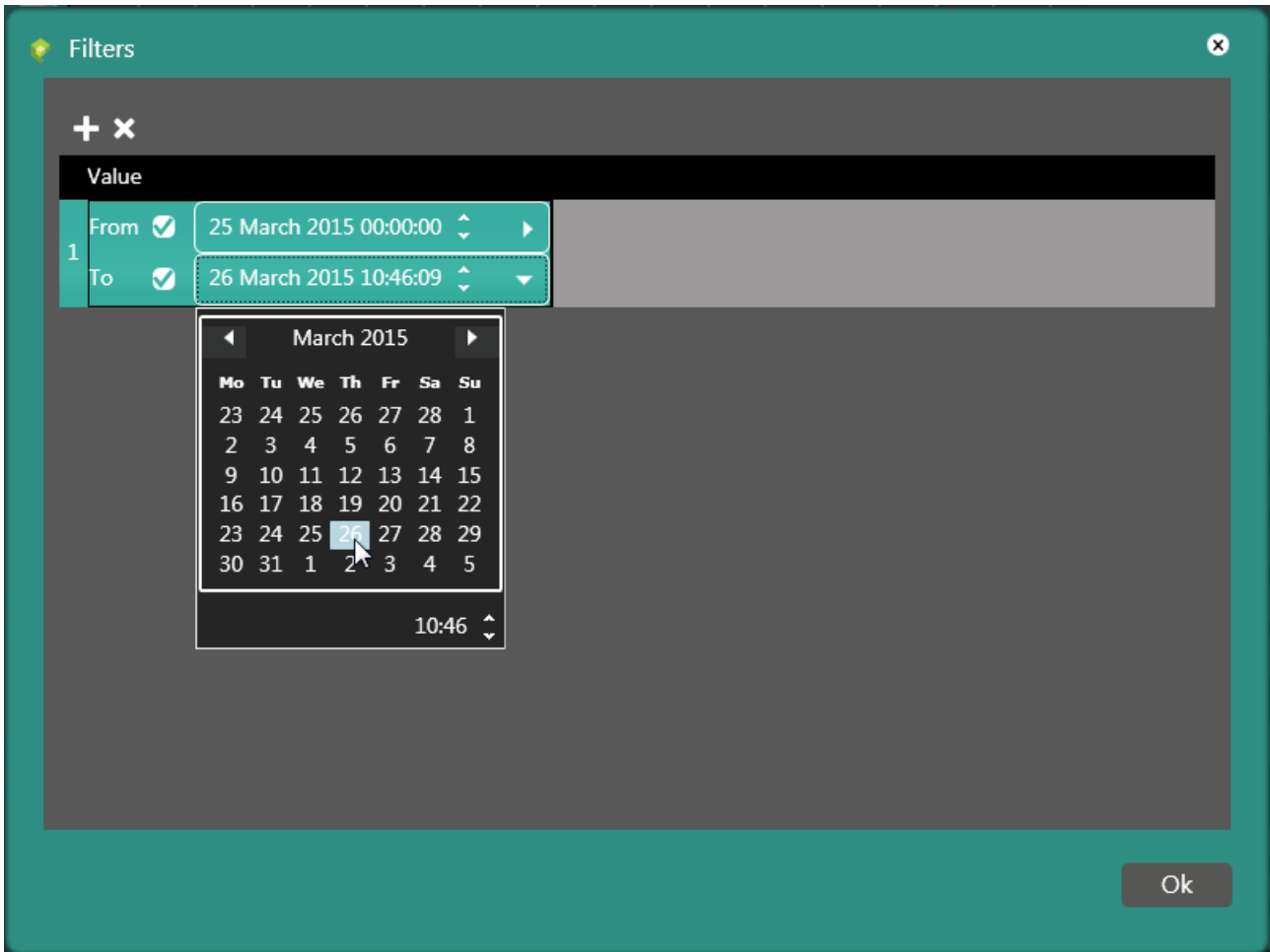
# CREW Manual

Click “Add filter”.



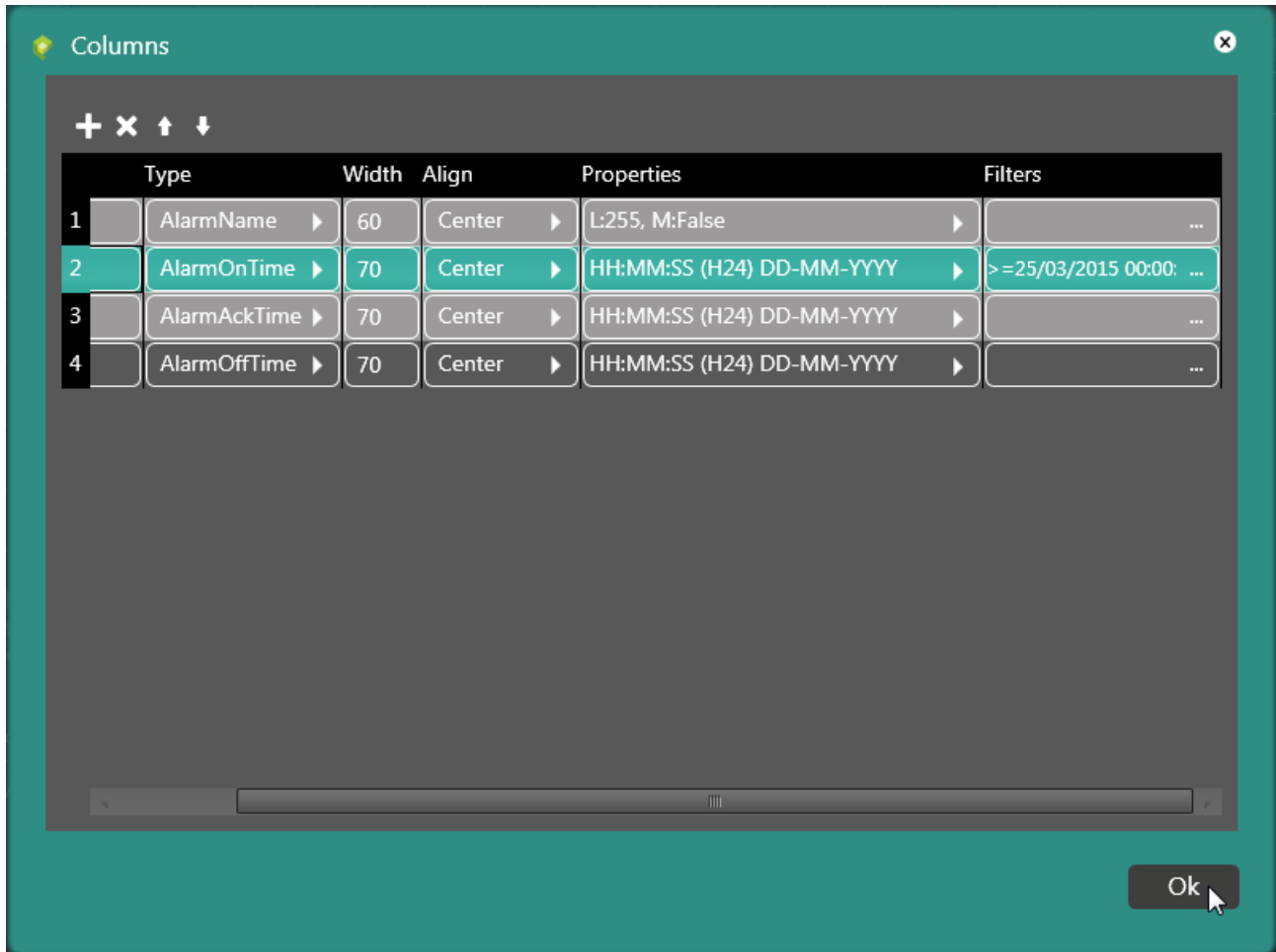
# CREW Manual

Select the time period that you wish to see the alarms associated to the predefined "AlarmOnTime" function for (for example, from March 24-25 2015).



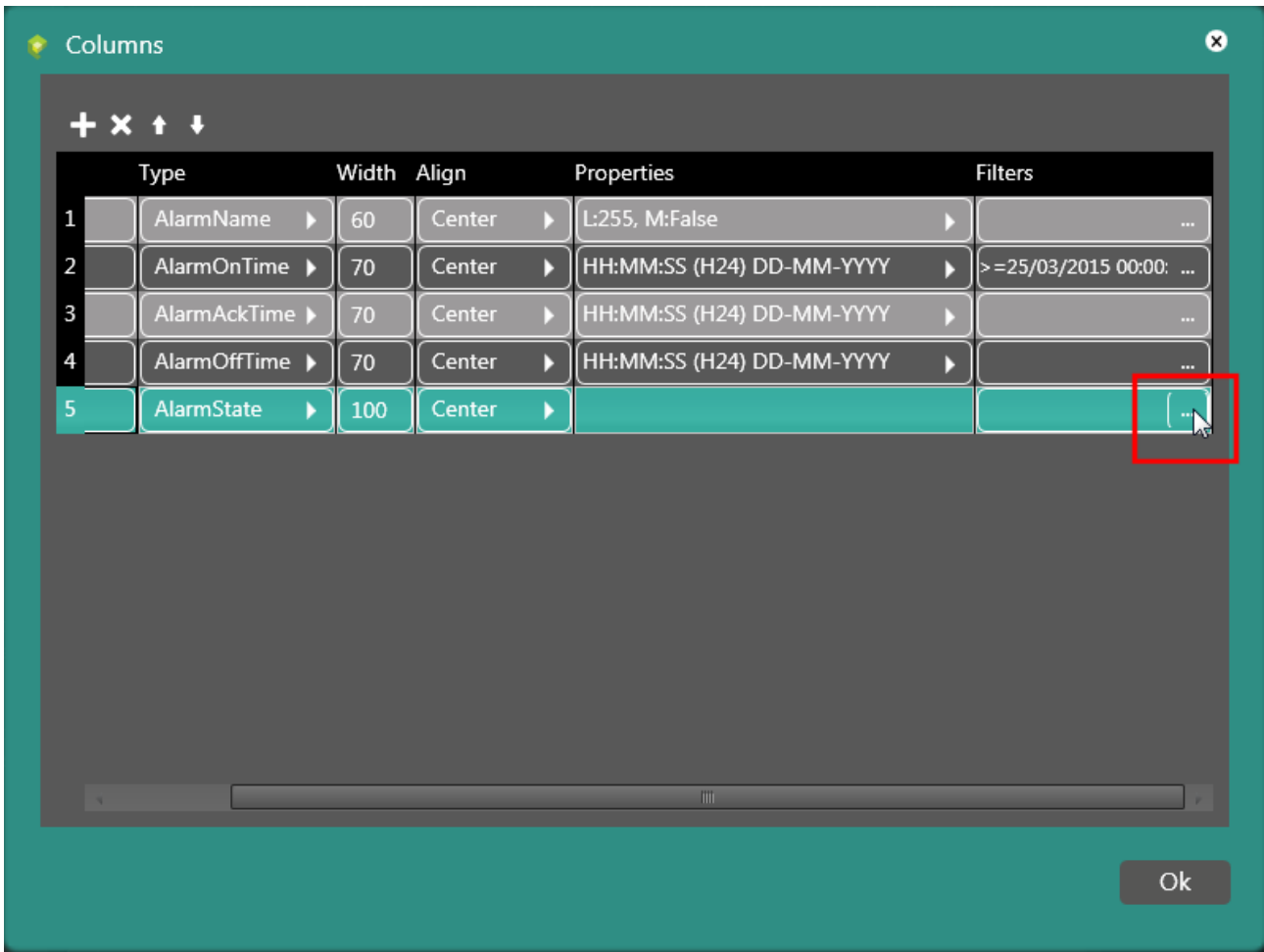
# CREW Manual

The filter will now be displayed in the “Filters” column.

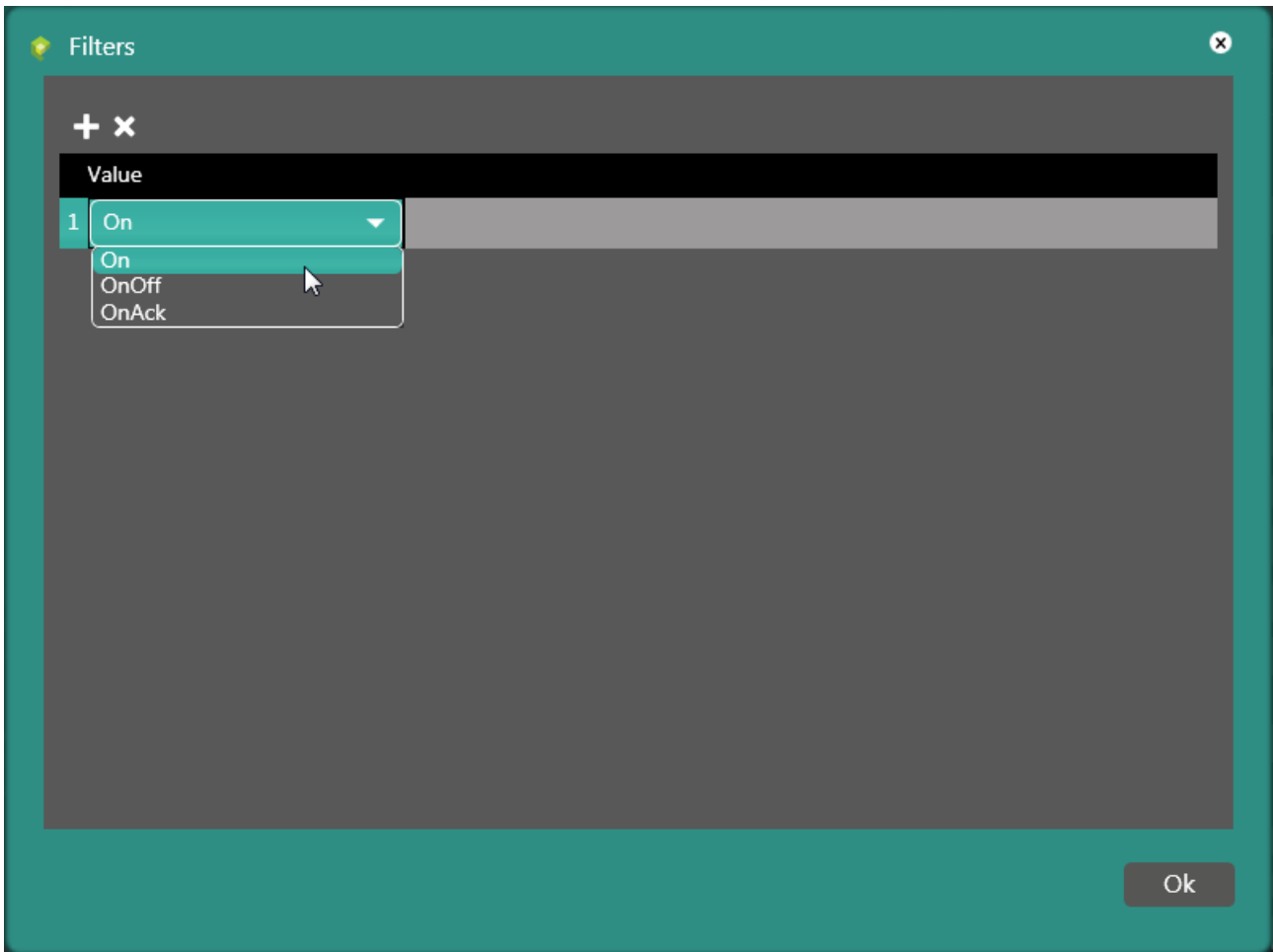


# CREW Manual

Based on the type of function, it is possible to chose from various filters. For example, it is possible to associate the viewing mode of the "AlarmState" function based on the alarm state.

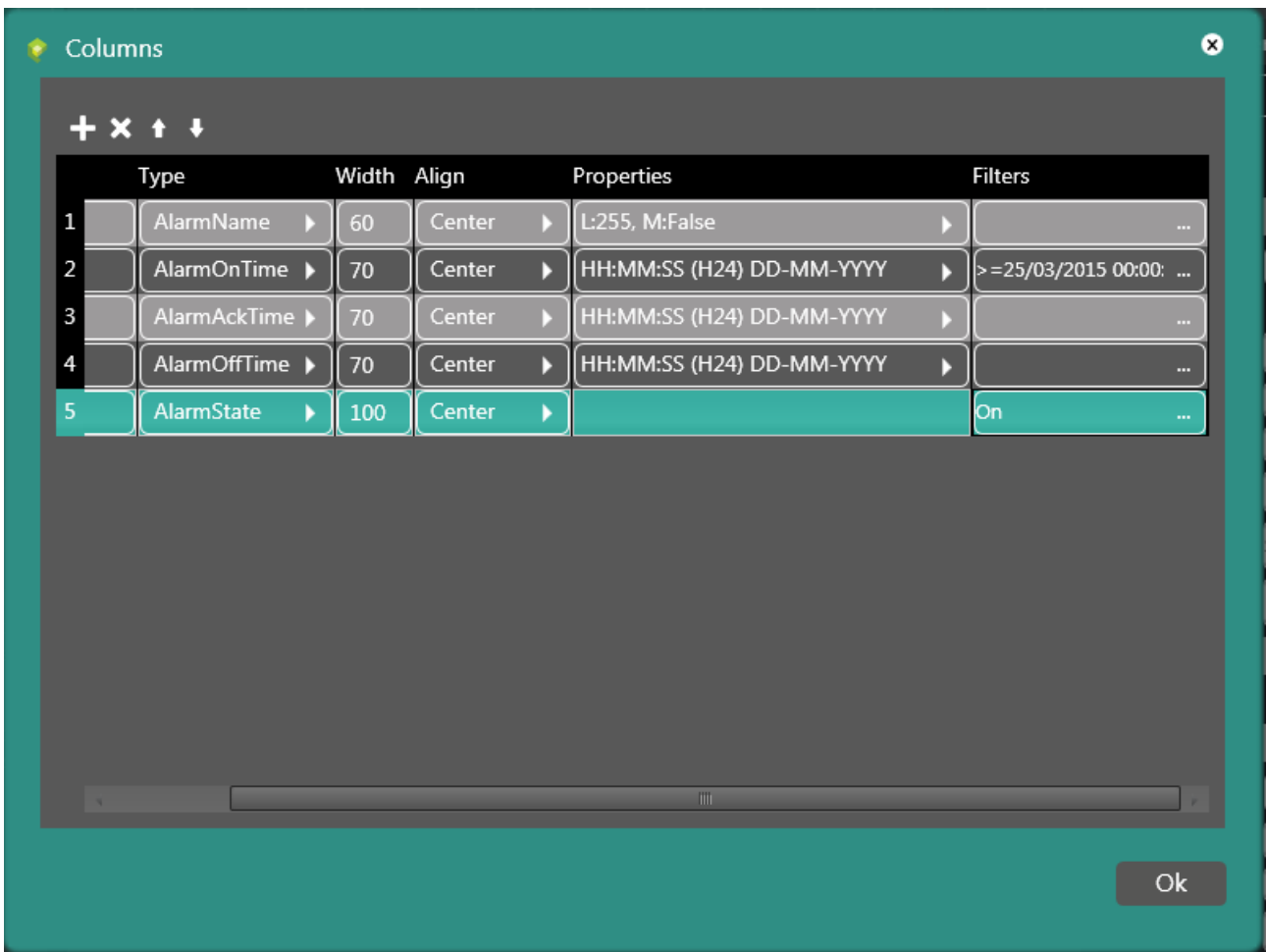


# CREW Manual



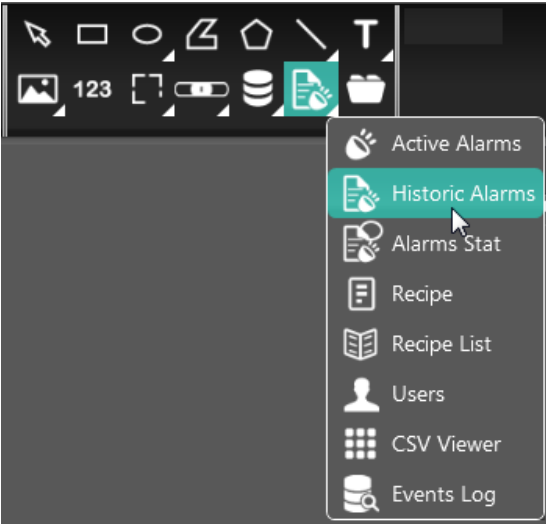
# CREW Manual

Choose the type of filter to view only the alarms assigned to that filter. For example, if you choose the “On” filter, only the alarms with “On” state will be displayed.

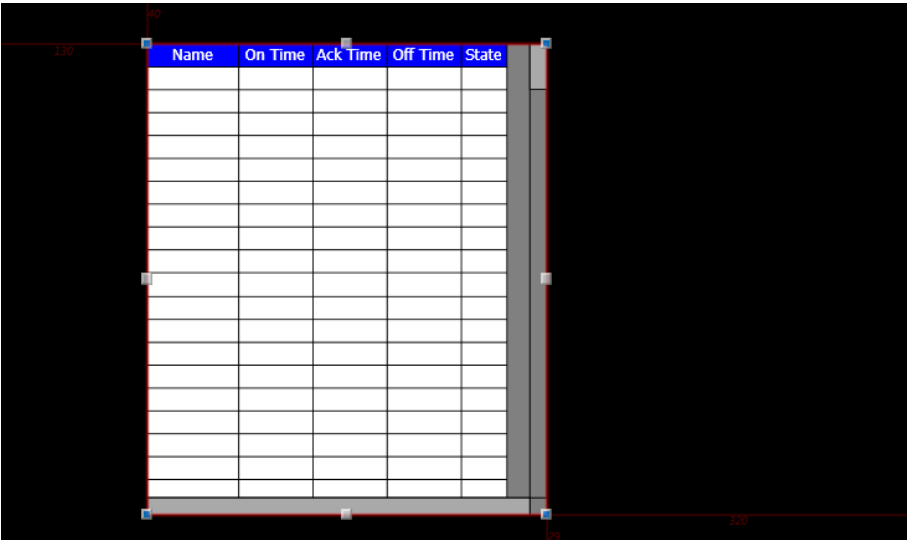


# CREW Manual

## Alarm Log



The "Alarm Log" icon on the "Graphics" menu is used to place an alarm log table on the page, drawing it with the mouse on the page.



A screenshot of a table being drawn on a black background. The table has a header row with columns: Name, On Time, Ack Time, Off Time, and State. The table is currently empty of data. Red dimension lines are visible around the table, indicating its size.

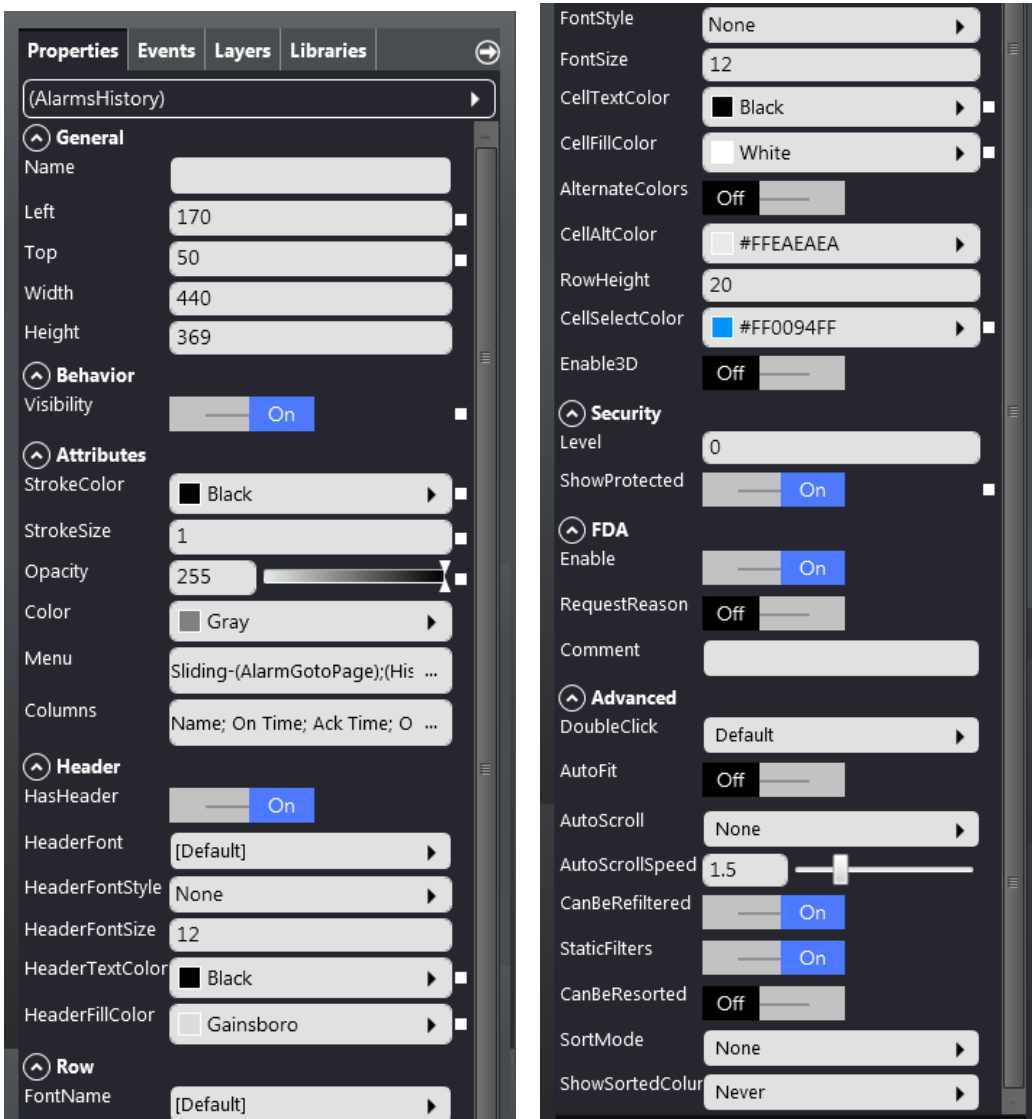
To determine the features of the "Alarm Log" table, set them in the "Properties Editor", as shown in the section "[Alarm Log Properties](#)".



# CREW Manual

## Alarm Log Properties

The following image illustrates all the editable properties of the Alarm Log table. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

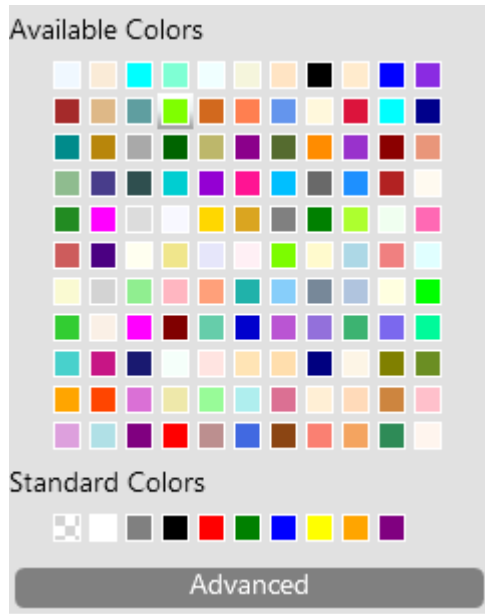
The following table describes all the editable properties of the Alarm Log table.

Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>UserMode</b>	Determines whether the user can interact with the object: - read only, the user cannot interact with the object; this value takes precedence over the Disabled property - read-write, the user can interact with the object; the effective interaction is validated by the Disabled property
<b>Attributes</b>	
<b>StrokeColor</b>	Determines the fill colour of the object; the colour is selectable by RGB colour code or colour palette
<b>StrokeSize</b>	Determines the stroke colour (edges of the figure); the colour is selectable by RGB colour code or colour palette
<b>Opacity</b>	Determines the opacity of the object
<b>Color</b>	Allows you to assign the color to the object using the color bar (Hue) (valid for objects in the Library)
<b>Menu</b>	By clicking on the "Browse" menu option, you can make an Active Alarms menu management of the table in Runtime as described in this section
<b>Columns</b>	Clicking on the "Browse" button of the "Columns" option, you can determine which columns to appear in the table and define their properties (as described in this section)
<b>Header</b>	
<b>HasHeader</b>	Determines whether the view of the Active Alarm View must have the header or not
<b>HeaderFont</b>	Determines the font type used to display the header text
<b>HeaderFontStyle</b>	Header font style. Any combination of the following features: - None - Italics - Bold - Underline
<b>HeaderFontSize</b>	Indicates the font size of the values written in the headers
<b>HeaderTextColor</b>	Determines the color of the header text
<b>HeaderFillColor</b>	Determines the color of the cell that contains the header
<b>Row</b>	
<b>FontName</b>	Determines the font used for the items of Active Alarms

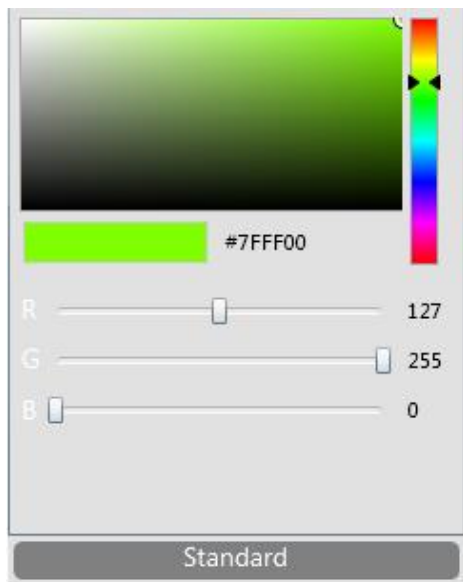
<b>FontStyle</b>	Font style. Any combination of the following features: - None - Italics - Bold - Underline
<b>FontSize</b>	Determines the stroke of Active Alarms view
<b>CellTextColor</b>	Represents the color of the writing cells
<b>CellFillColor</b>	Determines the color of the cells of table columns
<b>AlternateColors</b>	Allows you to assign two alternating colors for each row in the table
<b>CellAltColor</b>	Determines the alternative color (active if the option "AlternateColors" is "ON")
<b>RowHeight</b>	Determines the height of the row of the table (pixel)
<b>CellSelectColor</b>	Determines the color of the selected cell
<b>Enable3D</b>	Enable the 3D view ("embossed" view) of the table
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project
<b>ShowProtected</b>	Enables the displaying of a "status" icon in the object when it is protected for the current user
<b>FDA</b>	
<b>Enable</b>	Enables tracking of events related to the functions controlled through the grid menu
<b>RequestReason</b>	In each event to be recorded you have to insert the text that will be logged together with the time and date.
<b>Comment</b>	It defines a static comments to be tracked along with the execution events of the grid menu functions
<b>Advanced</b>	
<b>DoubleClick</b>	It allows you to assign a function to "double click" on each row of the table selecting it from the available ones
<b>AutoFit</b>	It determines whether the size of the columns of the grid are automatically calculated to fit precisely in the total available size
<b>AutoScroll</b>	It determines whether the content of the grid should automatically scroll
<b>AutoScrollSpeed</b>	It defines the scroll speed (when "AutoScroll" is enabled)
<b>CanBeRefiltered</b>	It determines whether the grid filters can be changed at runtime
<b>StaticFilters</b>	It defines the mode of interaction with the grid for access to the filter definition
<b>CanBeResorted</b>	It determines whether the grid lines can be ordered at runtime
<b>SortMode</b>	It defines the policy of the default grid sorting
<b>ShowSortedColumn</b>	It determines whether the column identified as sorting order must be highlighted in the grid

# CREW Manual

The properties related to colours can be edited through the colour palette.



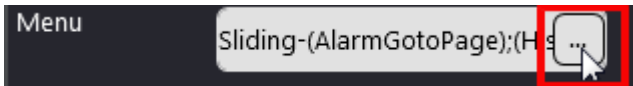
Click "Advanced" to select a colour using the RGB colour selection mask.



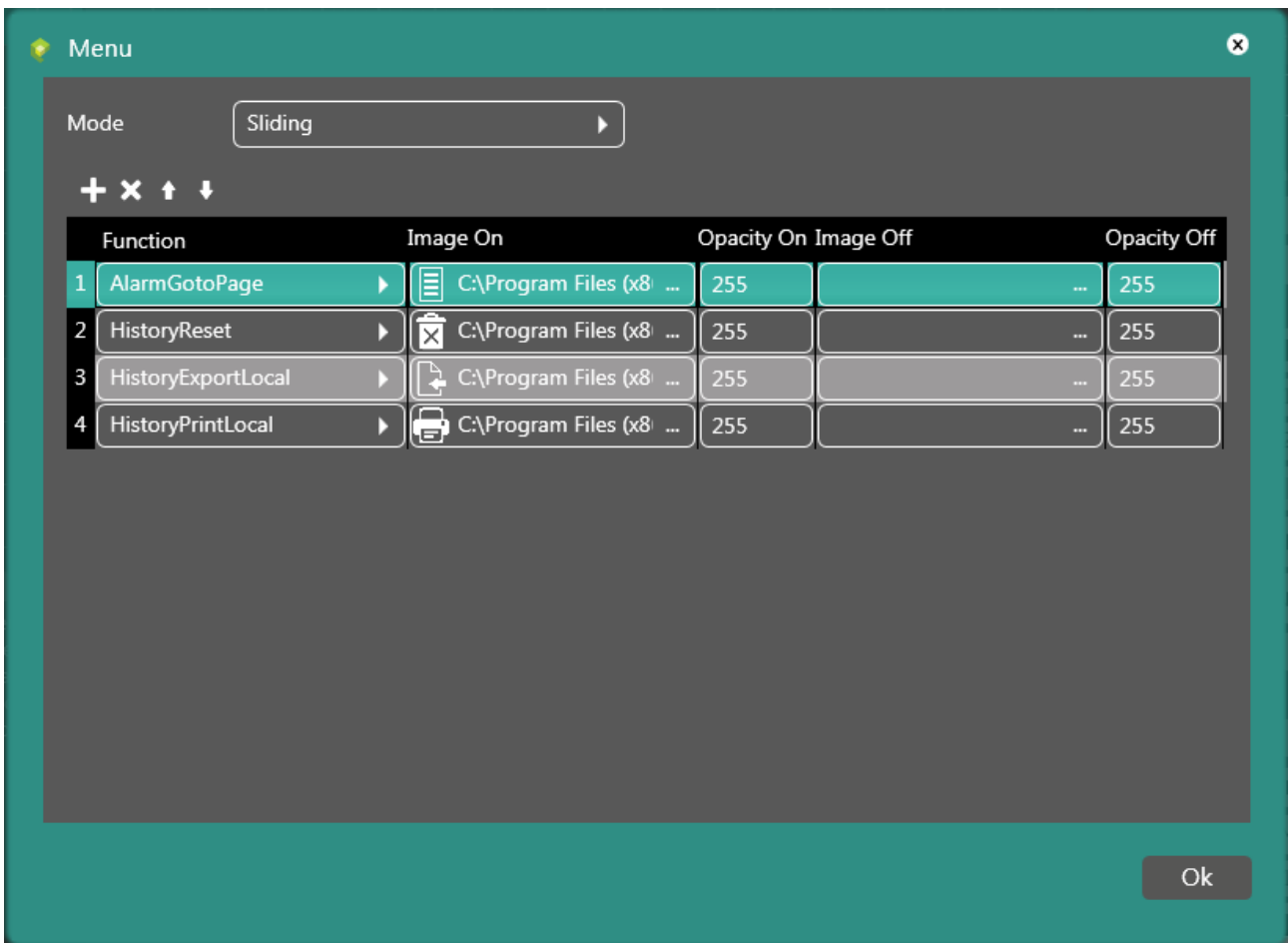
# CREW Manual

“Menu” option

Click the “Browse” key.

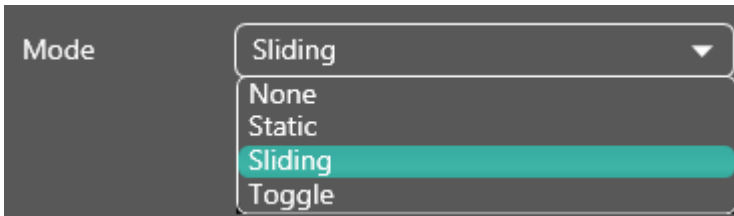


On the window that appears, it is possible to decide how to set the Runtime menu of the Alarm Log table.

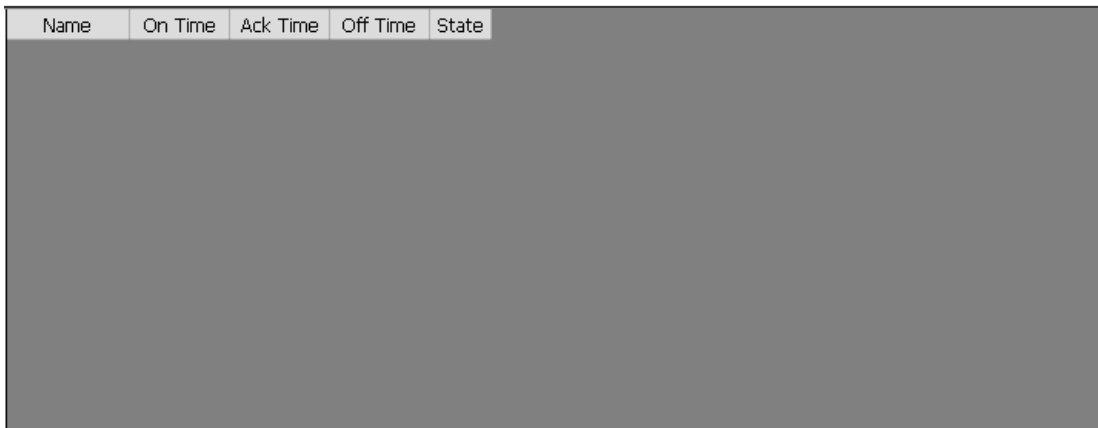


# CREW Manual

Mode: it is possible to customise the Runtime menu through one of the following options.



None: no Runtime menu. There are only the default columns.

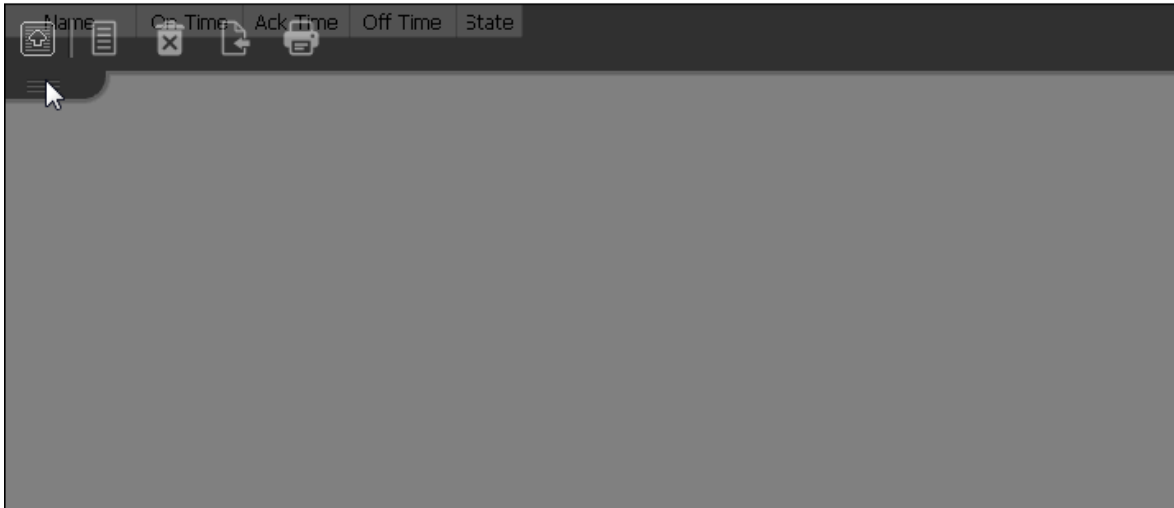


Static: static menu, namely fixed and always there.



# CREW Manual

Sliding: floating menu, which appears at the user's discretion.



Toggle: floating menu (similar to the “Sliding” option), which appears at the user's discretion.



# CREW Manual

Using the icons displayed in the image to add (+) or eliminate (X) new functions and move them up or down.



Function: this is used to assign a function from those shown in the image, to each of the icons that compose the Alarm Log view menu.

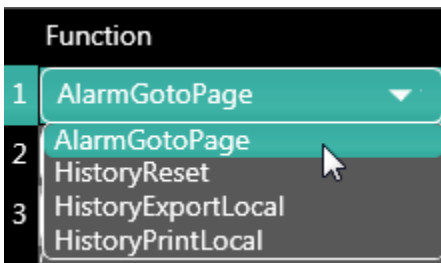
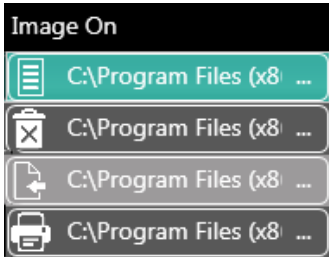


Table of functions that can be associated to the Alarm Log view.

Function	Description
<b>AlarmGotoPage</b>	Show the page associated to the alarm selected in grid (works for both active and historical alarms)
<b>HistoryReset</b>	Allows to reset all the contents of the alarm log
<b>HistoryExportBox</b>	Export all the history records, a dialog box will ask for the exported file name
<b>HistoryExportLocal</b>	Export all the history records; if no FileName is provided a request dialog box is visualized and will ask for the exported file name
<b>HistoryPrint</b>	Print all records alarm present in the alarm historical
<b>HistoryPrintLocal</b>	Print all records alarm present in the alarm historical, (a dialog box allows the selection of the target printer)

# CREW Manual

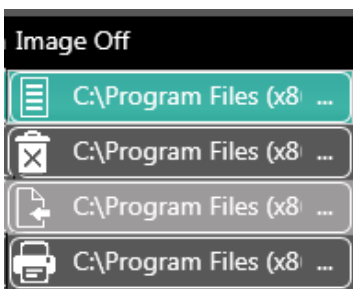
Image On: to associate an image to the icon when it is active.



Opacity On: to set the level of opacity of the image to be associated with the icon when it is active.



Image Off: to associate an image to the icon when it is not active.





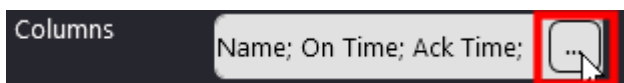
# CREW Manual

Opacity Off: to set the level of opacity of the image to be associated with the icon when it is not active.



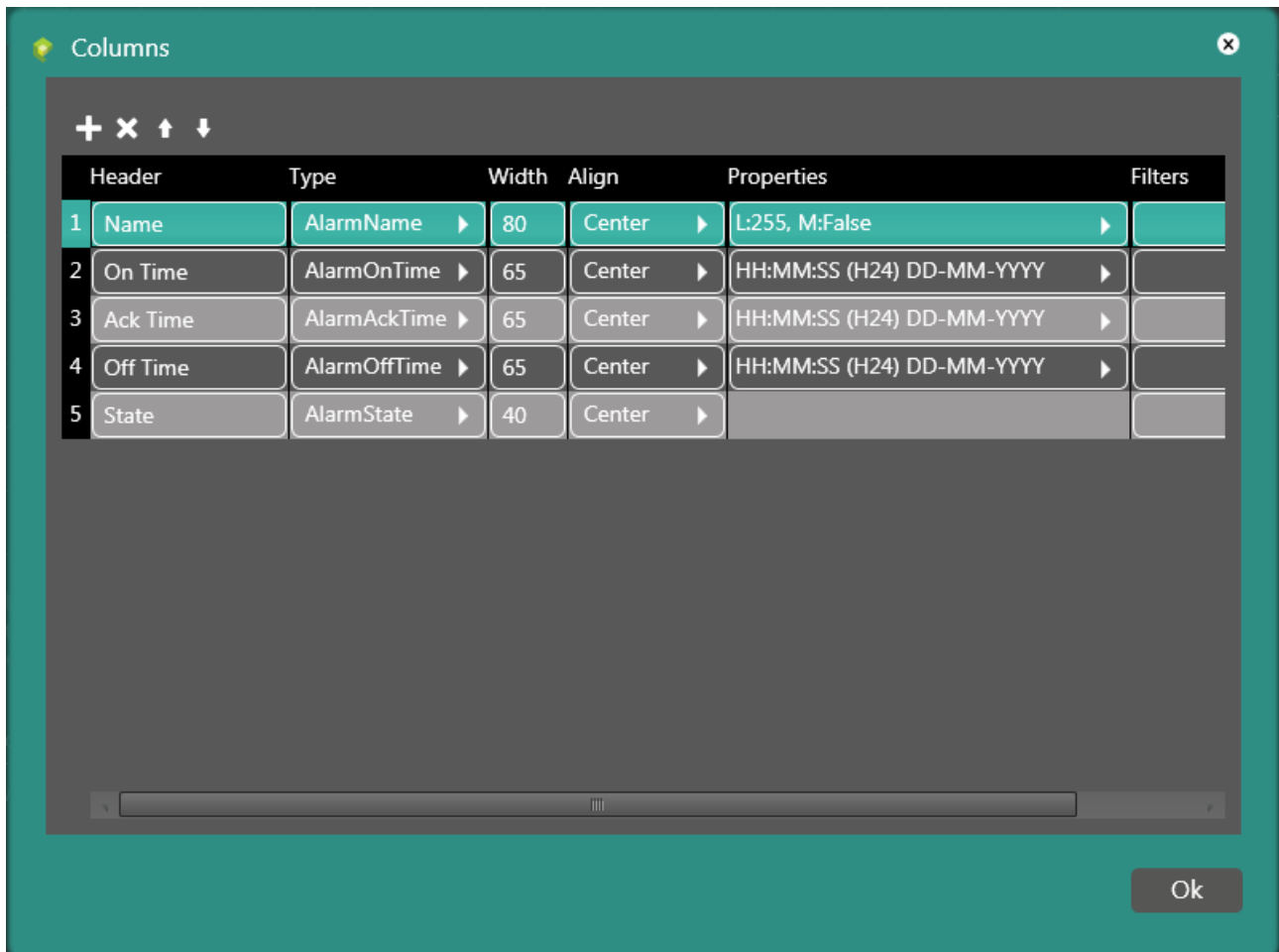
“Columns” option

Click the “Browse” key.



# CREW Manual

On the window that appears, it is possible to decide how to set the Alarm Log table.



Using the icons displayed in the image to add (+) or eliminate (X) new functions and move them up or down.



Heading: the title of the columns that comprise the table.

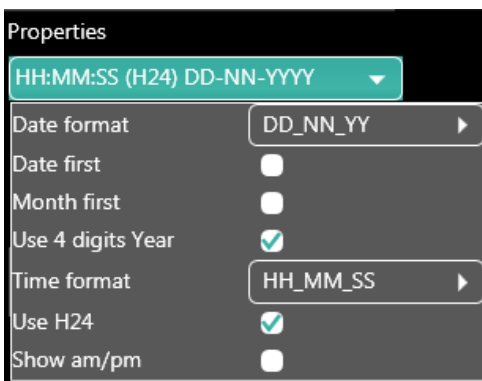
# CREW Manual

Type: the type of Alarm, which can be selected from the options shown in the image.



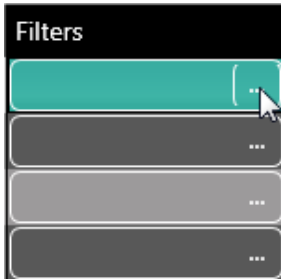
Width: the width of the table columns.

Property: to customise how datum acquisition is displayed (hour, minutes, seconds, etc.).



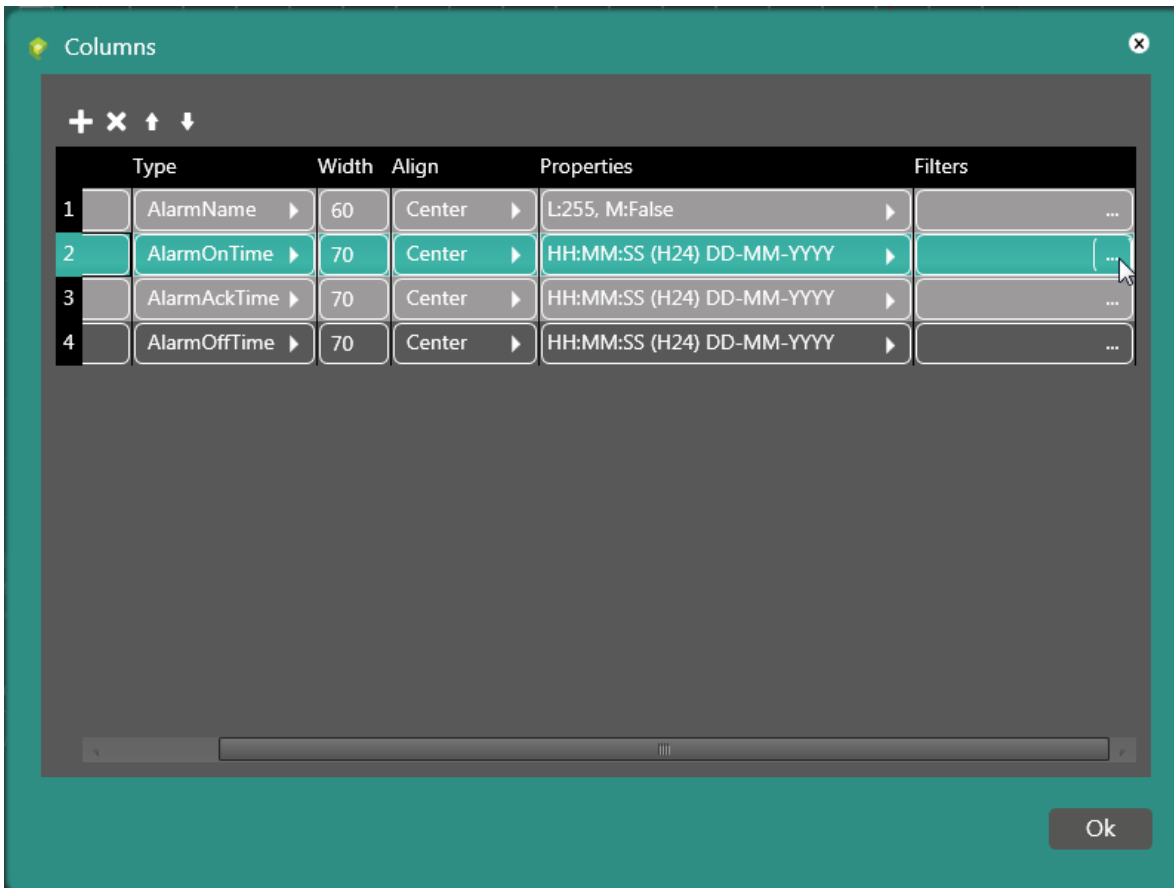
# CREW Manual

## Filters



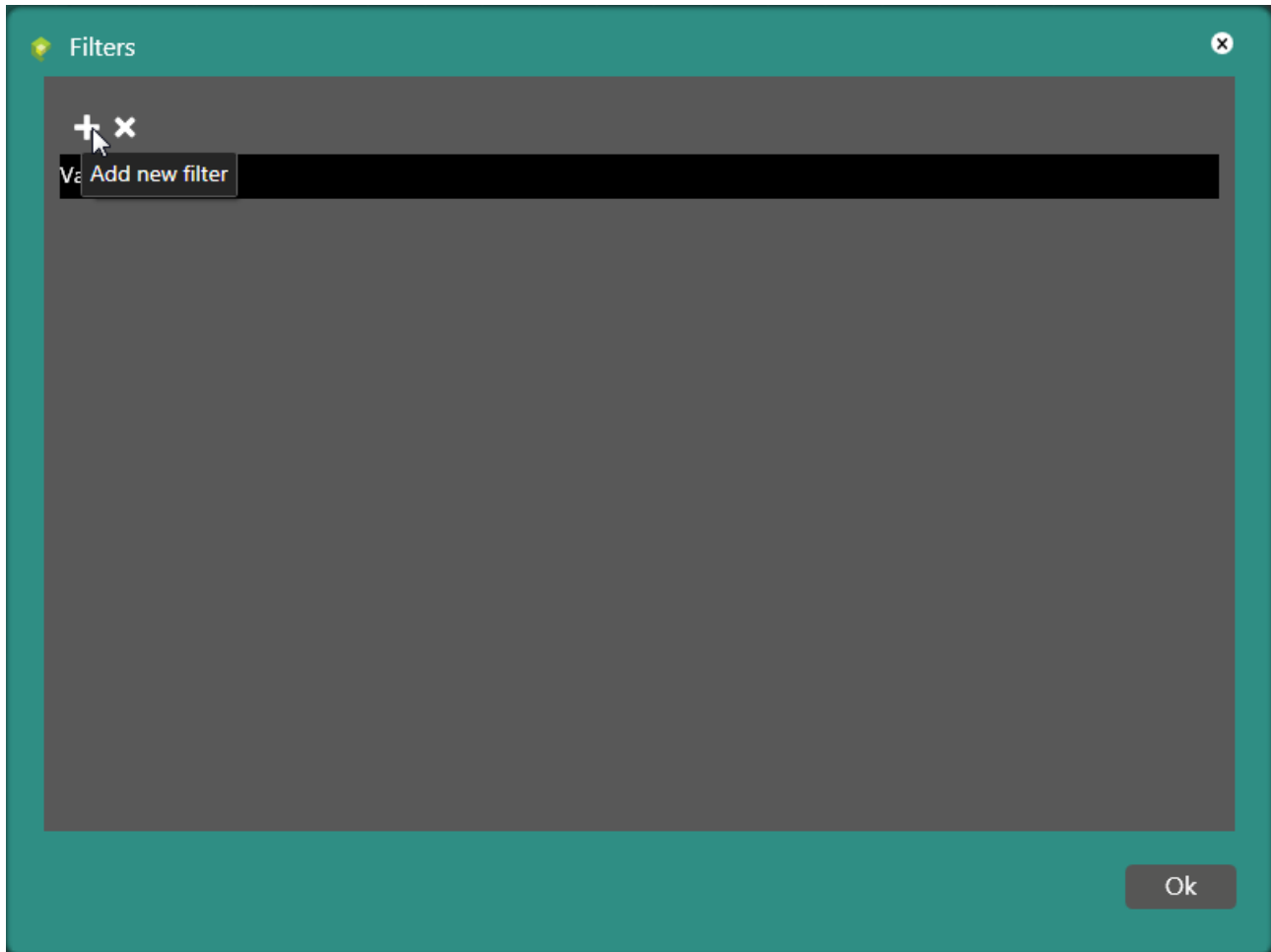
With the Filters option it is possible to enable Runtime display filters for the alarm log. For example, it is possible to associate a filter to the "AlarmOnTime" function to display only the alarms within a given period of time (chosen by the user).

To enable a filter, click "Browse" in the "Filters" column.



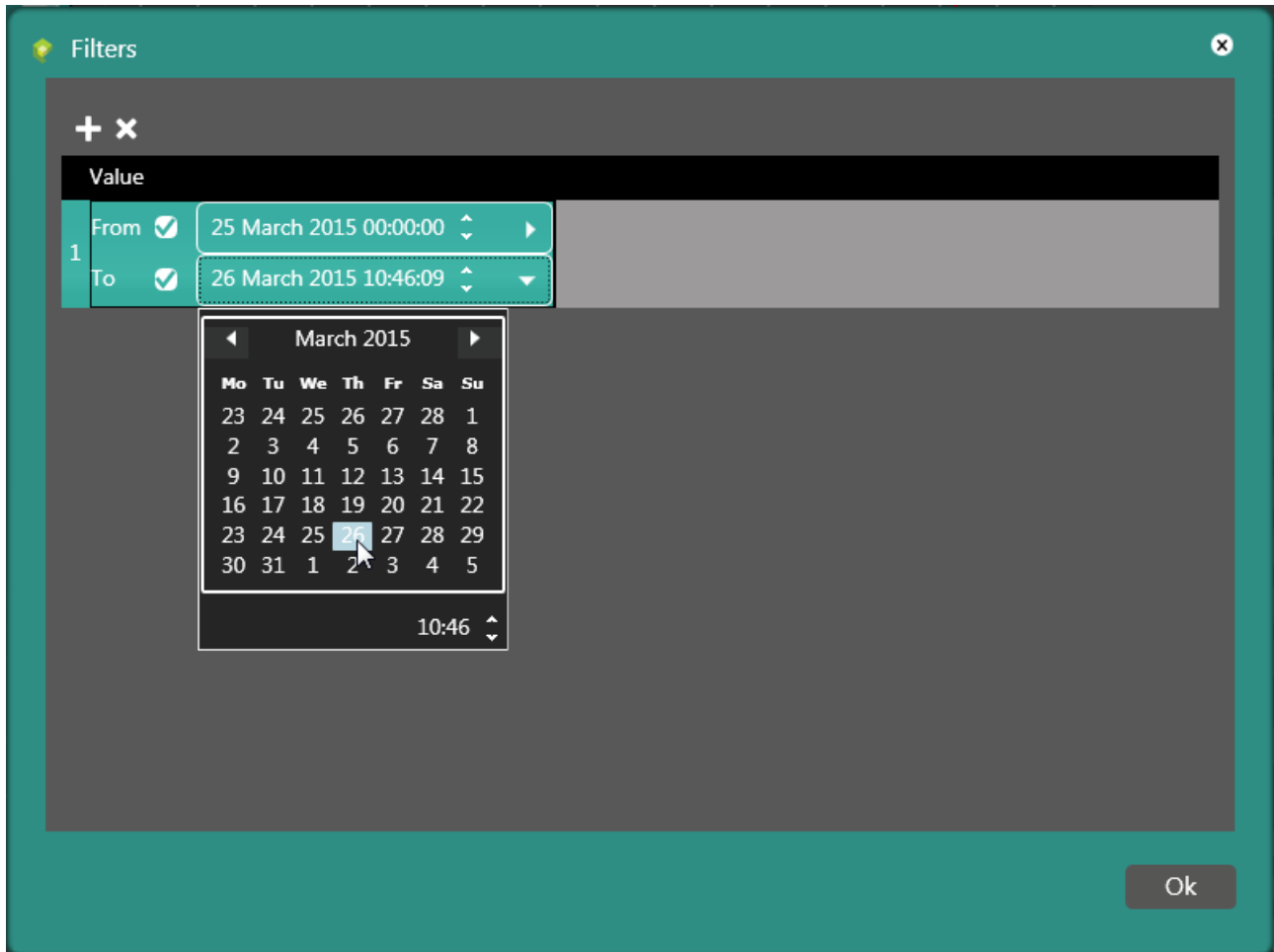
# CREW Manual

Click “Add filter”.



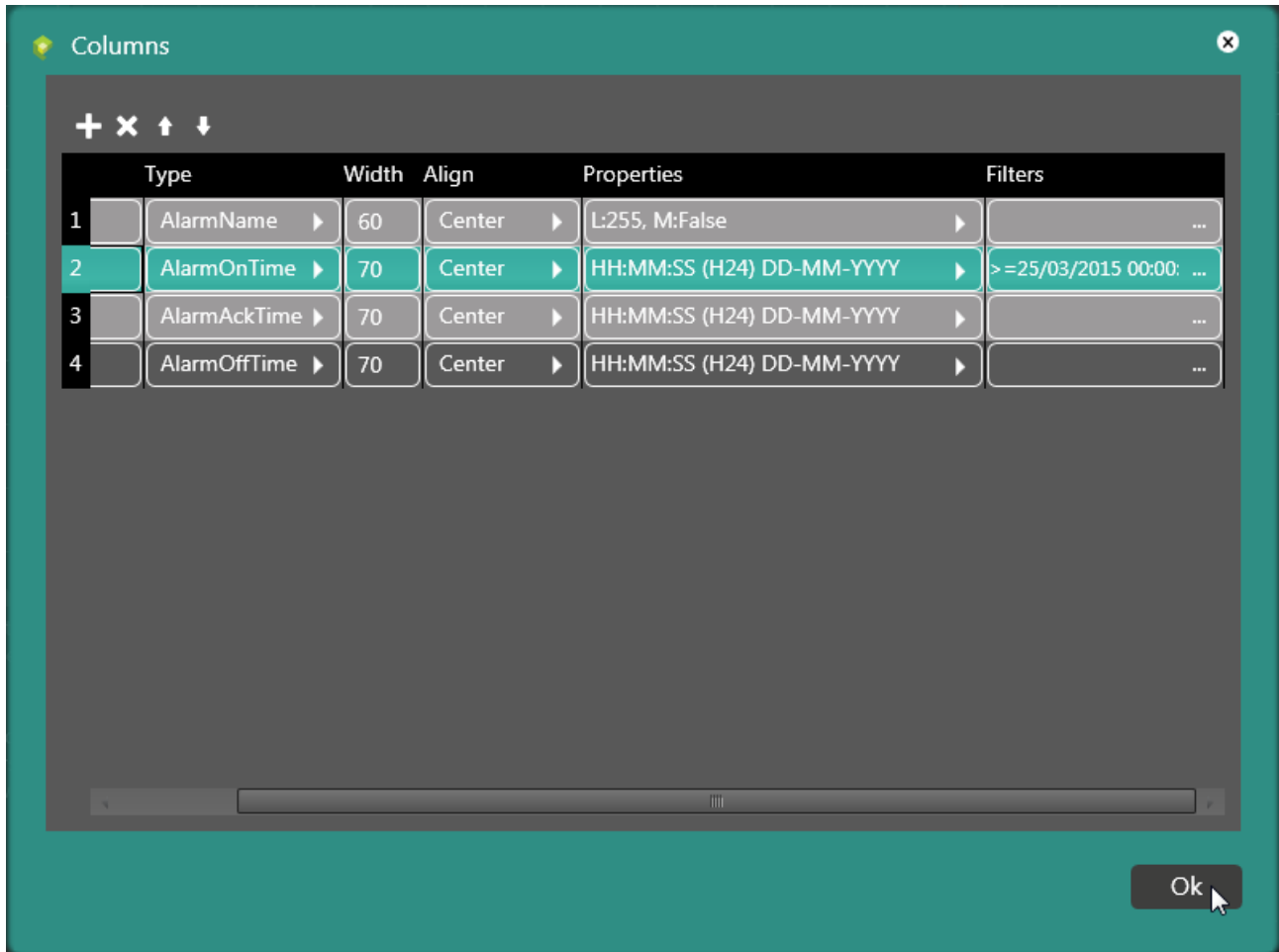
# CREW Manual

Select the time period that you wish to see the alarms associated to the predefined "AlarmOnTime" function for (for example, from March 24-25 2015).



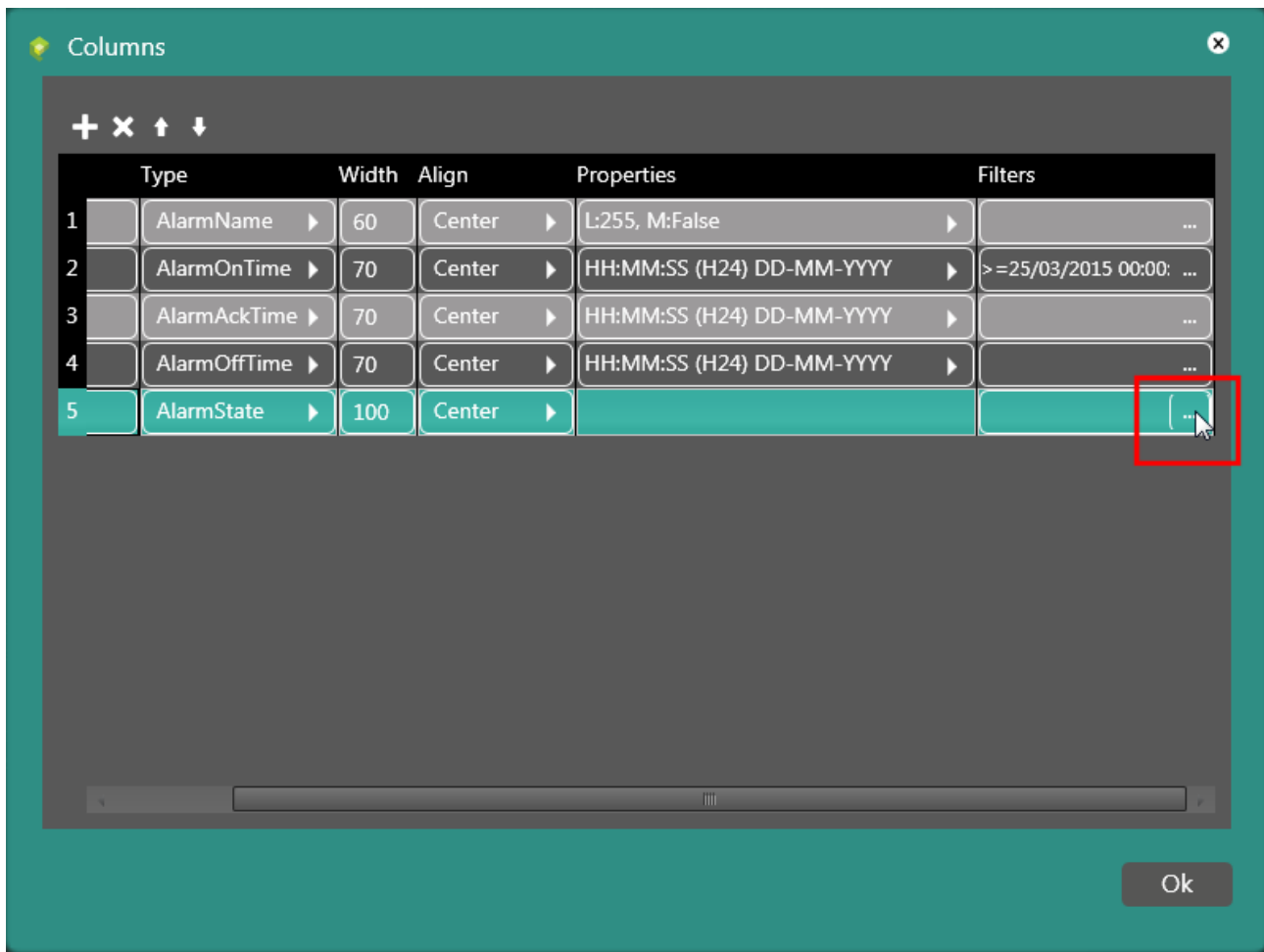
# CREW Manual

The filter will now be displayed in the “Filters” column.



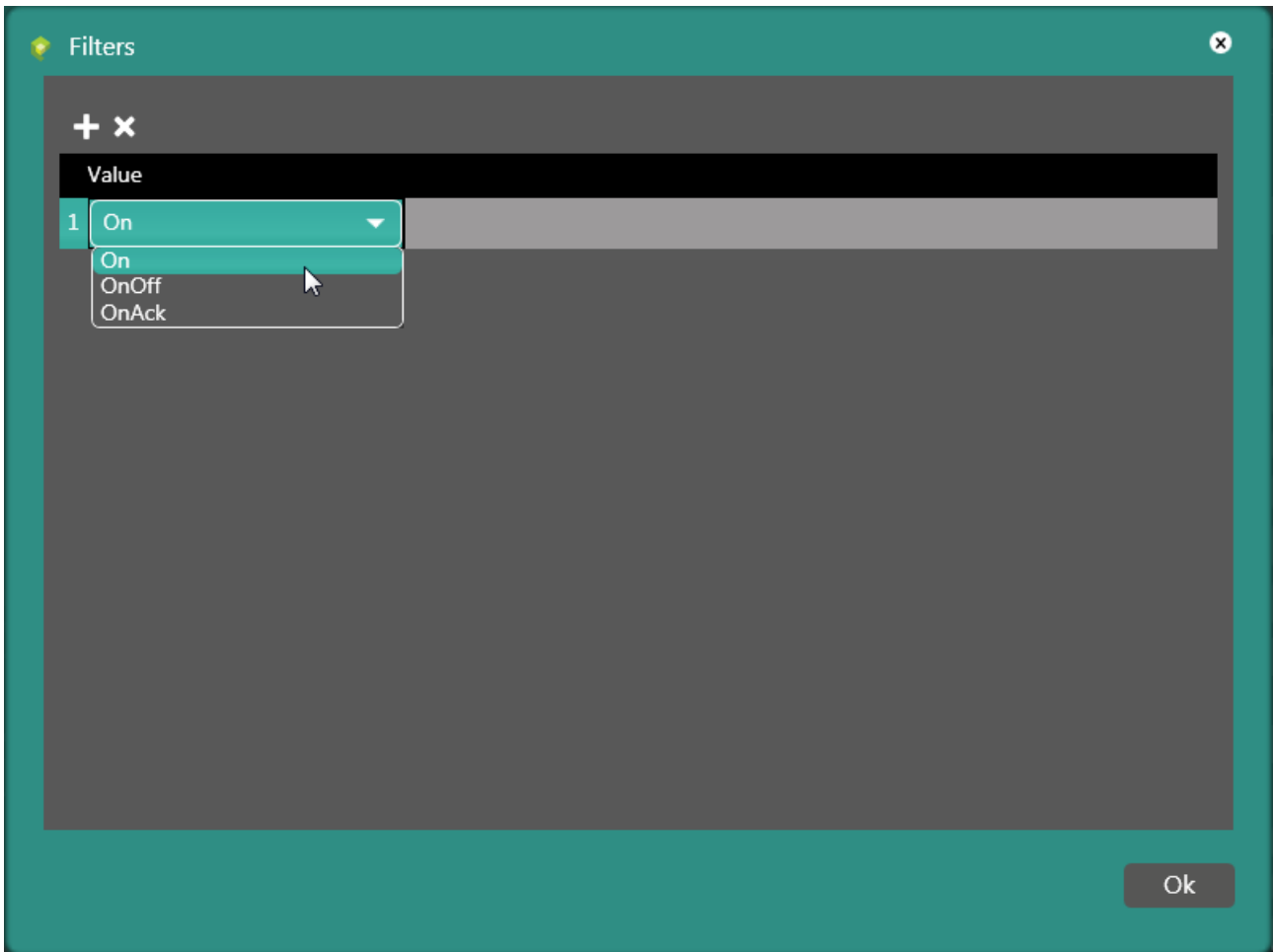
# CREW Manual

Based on the type of function, it is possible to choose from various filters. For example, it is possible to associate the viewing mode of the "AlarmState" function based on the alarm state.



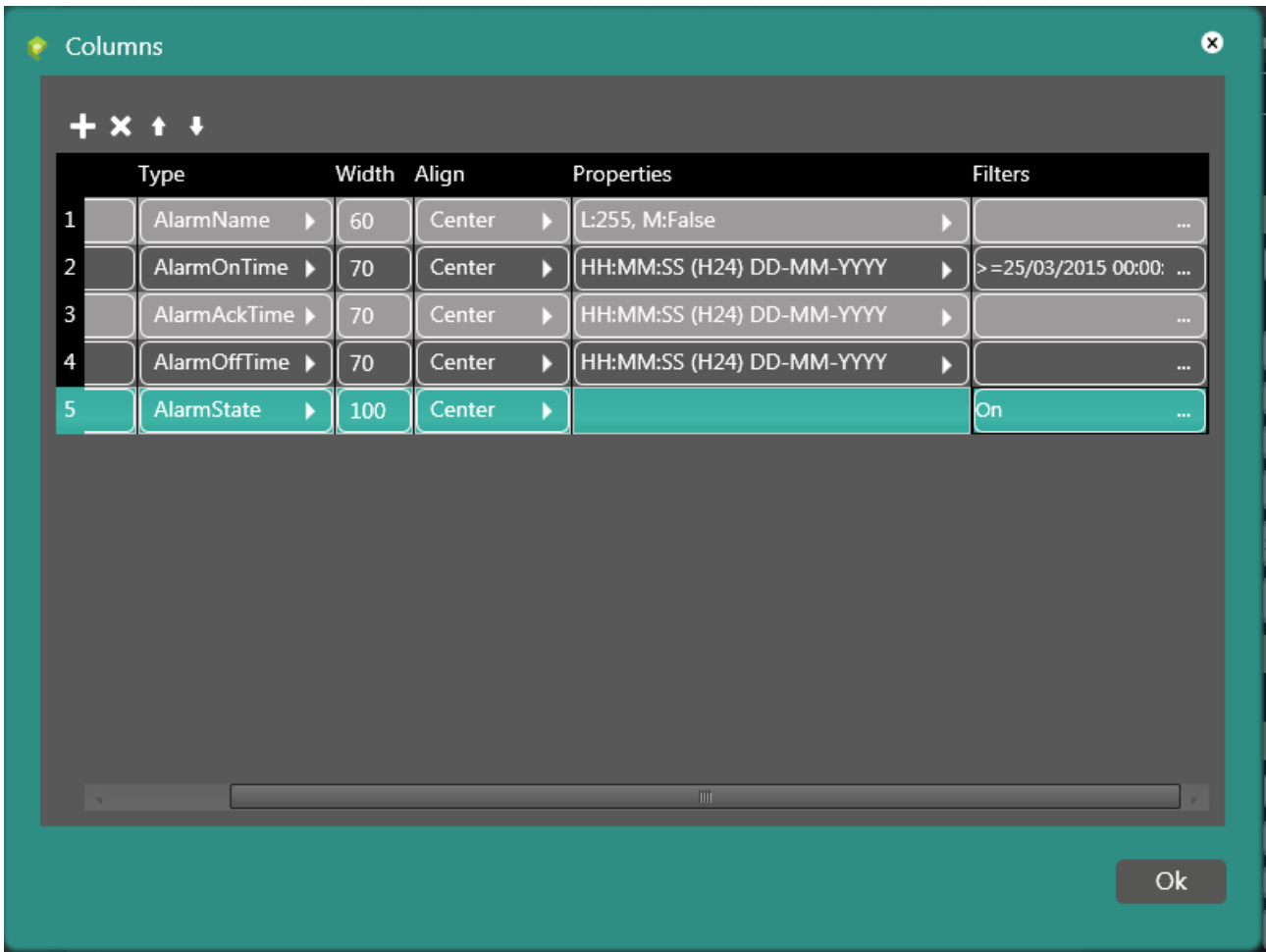


# CREW Manual



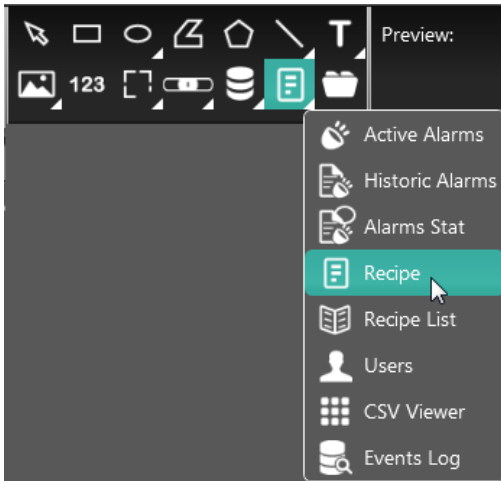
# CREW Manual

Choose the type of filter to view only the alarms assigned to that filter. For example, if you choose the “On” filter, only the alarms with “On” state will be displayed.

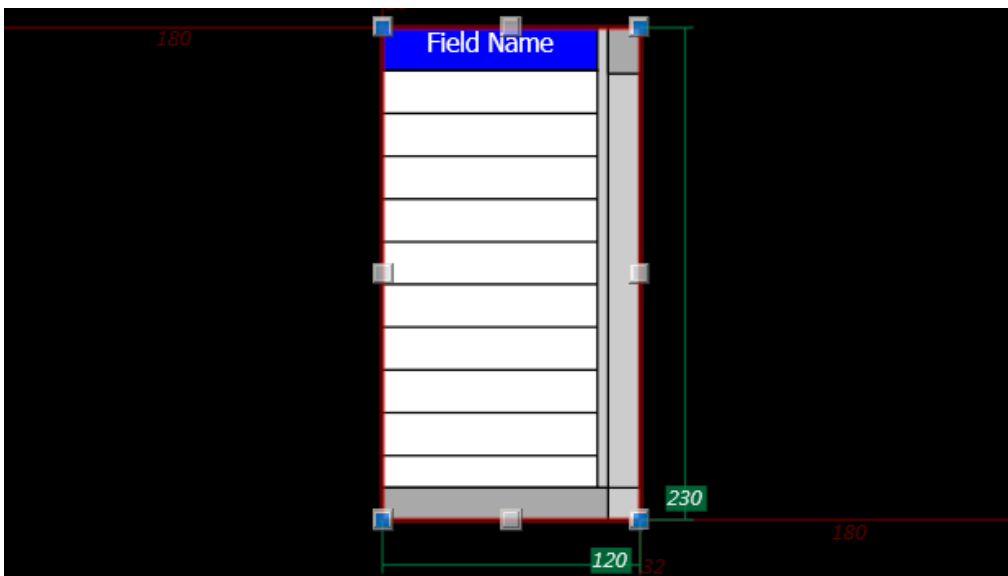


# CREW Manual

## Recipe



The “Recipe” icon on the “Graphics” menu is used to place a recipe table on the page, drawing it with the mouse on the page.

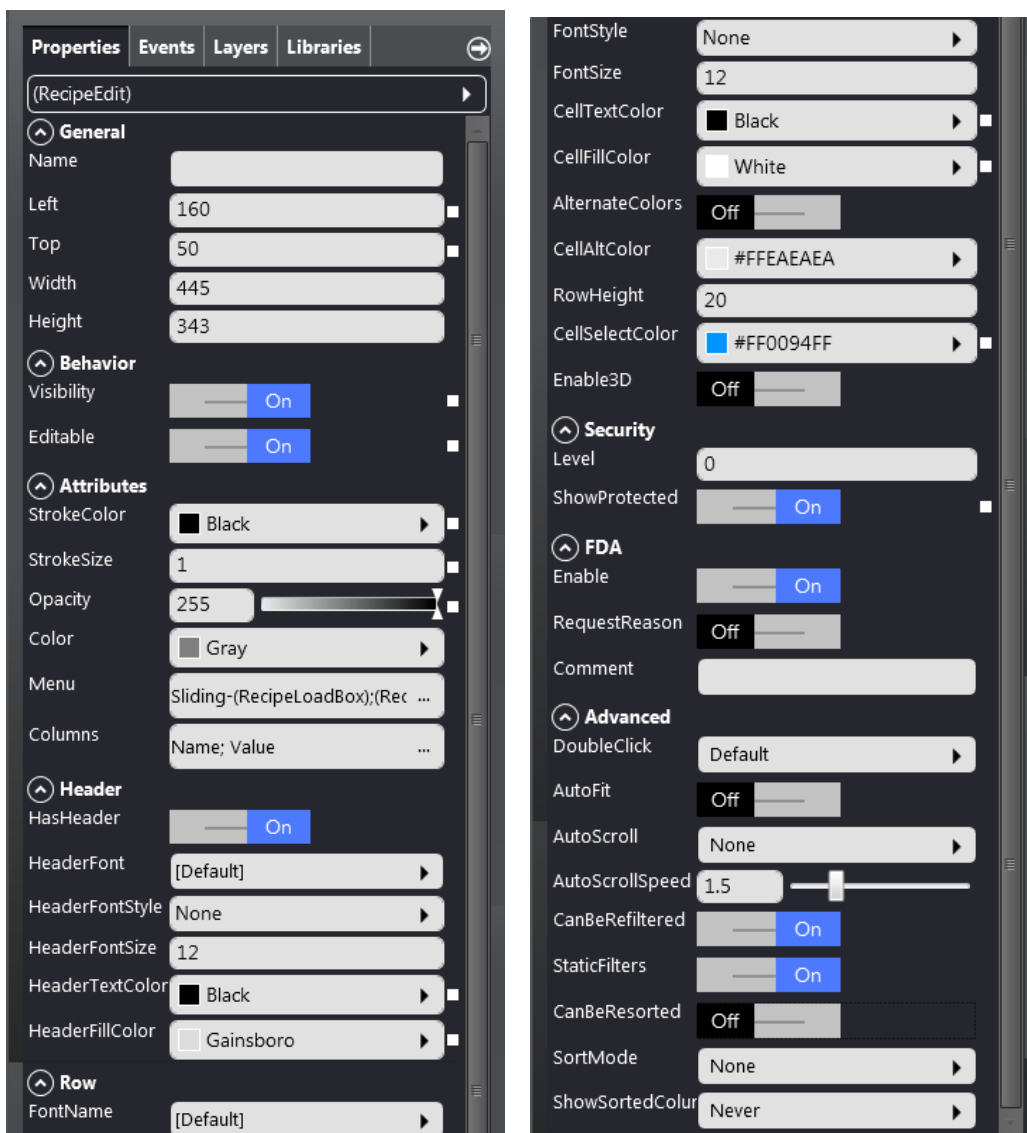


To determine the features of the “Recipe”, set them in the “Properties Editor”, as shown in the section "[Recipe Properties](#)".

# CREW Manual

## Recipe Properties

The following image illustrates all the editable properties of the Recipe. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

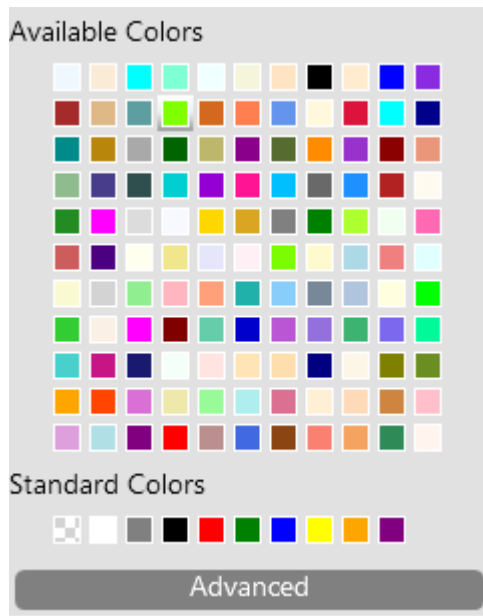
The following table describes all the editable properties of the Recipe view.

Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>UserMode</b>	Determines whether the user can interact with the object: - read only, the user cannot interact with the object; this value takes precedence over the Disabled property - read-write, the user can interact with the object; the effective interaction is validated by the Disabled property
<b>Attributes</b>	
<b>StrokeColor</b>	Determines the fill colour of the object; the colour is selectable by RGB colour code or colour palette
<b>StrokeSize</b>	Determines the stroke colour (edges of the figure); the colour is selectable by RGB colour code or colour palette
<b>Opacity</b>	Determines the opacity of the object
<b>Color</b>	Allows you to assign the color to the object using the color bar (Hue) (valid for objects in the Library)
<b>Menu</b>	By clicking on the "Browse" menu option, you can make a Recipe view management as described in this section
<b>Columns</b>	Clicking on the "Browse" button of the "Columns" option, you can determine which columns to appear in the table and define their properties (as described in this section)
<b>Header</b>	
<b>HasHeader</b>	Determines whether the Recipe view must have the header or not
<b>HeaderFont</b>	Determines the font type used to display the header text
<b>HeaderFontStyle</b>	Header font style. Any combination of the following features: - None - Italics - Bold - Underline
<b>HeaderFontSize</b>	Indicates the font size of the values written in the headers
<b>HeaderTextColor</b>	Determines the color of the header text of Recipe view
<b>HeaderFillColor</b>	Determines the color of the cell that contains the header of Recipe view

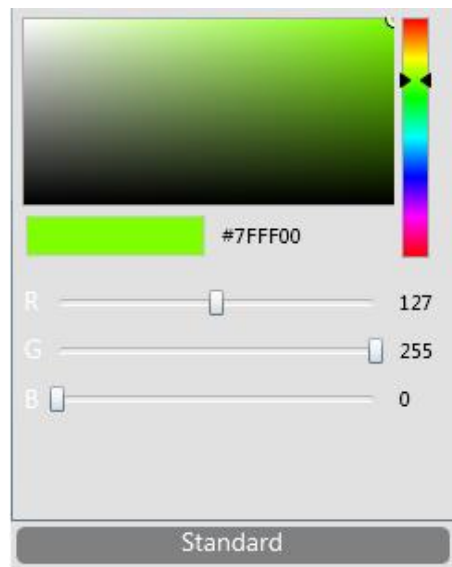
<b>Row</b>	
<b>FontName</b>	Determines the font used for the items of Recipe view
<b>FontStyle</b>	Font style. Any combination of the following features: - None - Italics - Bold - Underline
<b>FontSize</b>	Determines the stroke of Recipe view
<b>CellTextColor</b>	Represents the color of the Recipe view writing cells
<b>CellFillColor</b>	Determines the color of the cells of table columns
<b>AlternateColors</b>	Allows you to assign two alternating colors for each row in the table
<b>CellAltColor</b>	Determines the alternative color (active if the option "AlternateColors" is "ON")
<b>RowHeight</b>	Determines the height of the row of the table (pixel)
<b>CellSelectColor</b>	Determines the color of the selected cell
<b>Enable3D</b>	Enable the 3D view ("embossed" view) of the table
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project
<b>ShowProtected</b>	Enables the displaying of a "status" icon in the object when it is protected for the current user
<b>FDA</b>	
<b>Enable</b>	Enables tracking of events related to the functions controlled through the grid menu
<b>RequestReason</b>	In each event to be recorded you have to insert the text that will be logged together with the time and date.
<b>Comment</b>	It defines a static comments to be tracked along with the execution events of the grid menu functions
<b>Advanced</b>	
<b>DoubleClick</b>	It allows you to assign a function to "double click" on each row of the table selecting it from the available ones
<b>AutoFit</b>	It determines whether the size of the columns of the grid are automatically calculated to fit precisely in the total available size
<b>AutoScroll</b>	It determines whether the content of the grid should automatically scroll
<b>AutoScrollSpeed</b>	It defines the scroll speed (when "AutoScroll" is enabled)
<b>CanBeRefiltered</b>	It determines whether the grid filters can be changed at runtime
<b>StaticFilters</b>	It defines the mode of interaction with the grid for access to the filter definition
<b>CanBeResorted</b>	It determines whether the grid lines can be ordered at runtime
<b>SortMode</b>	It defines the policy of the default grid sorting
<b>ShowSortedColumn</b>	It determines whether the column identified as sorting order must be highlighted in the grid

# CREW Manual

The properties related to colours can be edited through the colour palette.



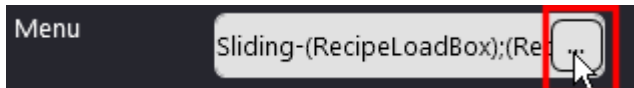
Click “Advanced” to select a colour using the RGB colour selection mask.



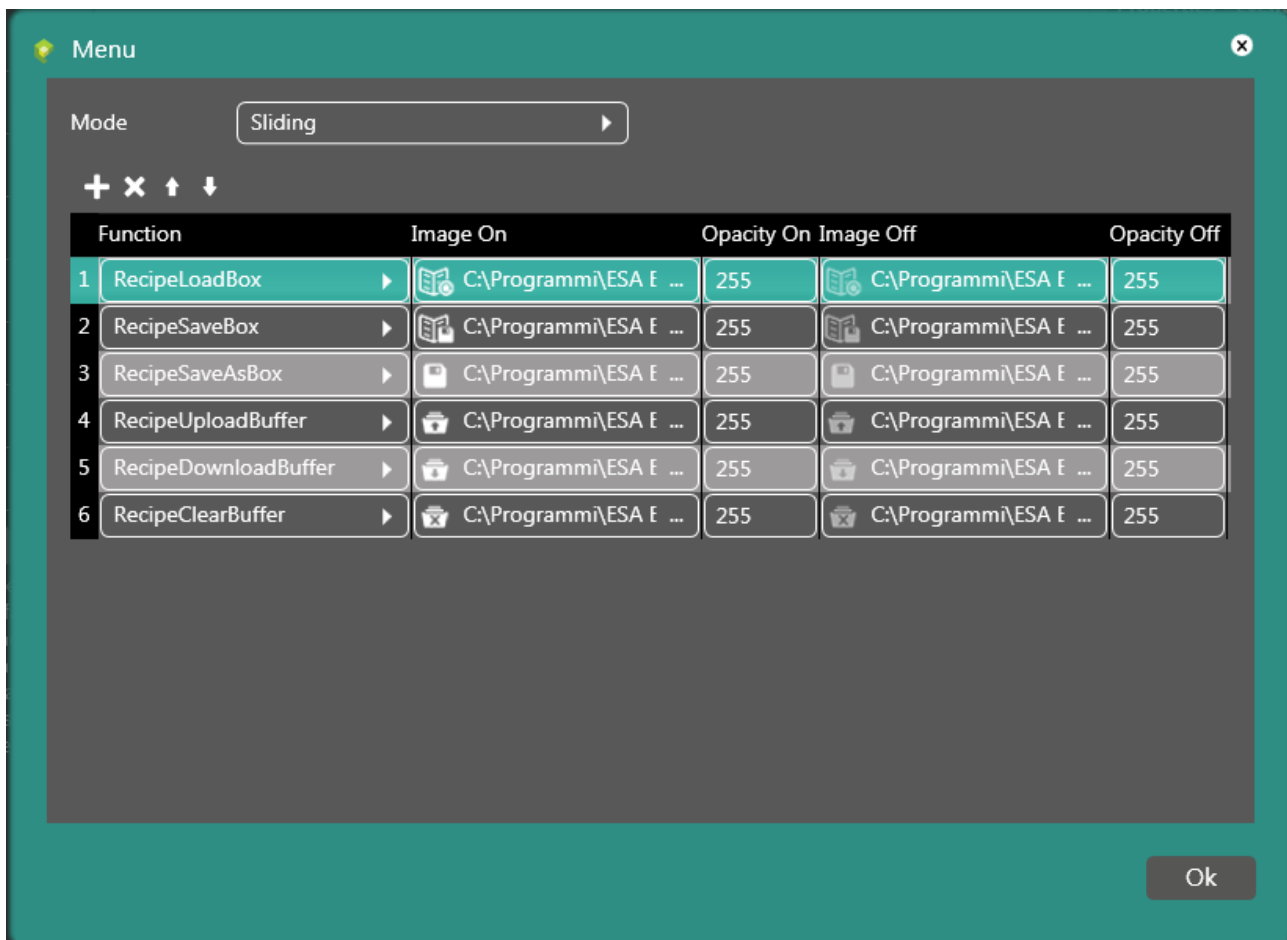
# CREW Manual

“Menu” option

Click the “Browse” key.

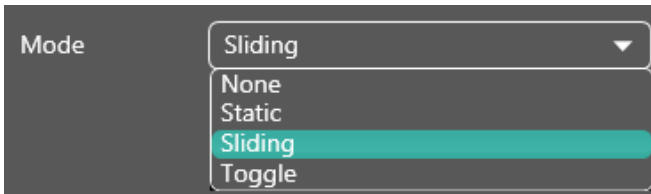


On the window that appears, it is possible to decide how to set the Runtime menu of the Recipe view.



# CREW Manual

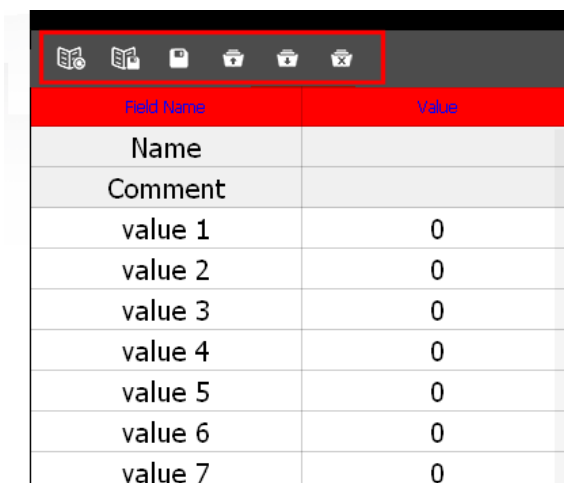
Mode: it is possible to customise the Runtime menu through one of the following options.



None: no Runtime menu. There are only the default columns.

Field Name	Value
Name	
Comment	
value 1	0
value 2	0
value 3	0
value 4	0
value 5	0
value 6	0
value 7	0

Static: static menu, namely fixed and always there.



Field Name	Value
Name	
Comment	
value 1	0
value 2	0
value 3	0
value 4	0
value 5	0
value 6	0
value 7	0



# CREW Manual

Sliding: floating menu, which appears at the user's discretion.

Field Name	Value
Name	
Comment	
value 1	0
value 2	0
value 3	0
value 4	0
value 5	0
value 6	0
value 7	0

Toggle: floating menu (similar to the "Sliding" option), which appears at the user's discretion.

Field Name	Value
Name	
Comment	
value 1	0
value 2	0
value 3	0
value 4	0
value 5	0
value 6	0
value 7	0

Using the icons displayed in the image to add (+) or eliminate (X) new functions and move them up or down.



# CREW Manual

Function: this is used to assign a function from those shown in the image, to each of the icons that compose the Alarms view menu.

Function	
1	RecipeLoadBox ▶
2	RecipeSaveBox ▶
3	RecipeSaveAsBox ▶
4	RecipeUploadBuffer ▶
5	RecipeDownloadBuffer ▶
6	RecipeClearBuffer ▶

Table of functions that can be associated to the Recipe view.

Function	Description
<b>RecipeLoadBox</b>	Uploads a recipe from the archive to the buffer; a window is displayed to allow selection of the recipe to be uploaded
<b>RecipeSaveBox</b>	Saves a recipe from the buffer to the archive; if the valid name of a recipe is in the buffer, then the function can be executed, otherwise the function acts as "Recipe SaveAs"; in case there is already a recipe with the same name, a window is displayed requesting confirmation, otherwise saving is performed immediately
<b>RecipeSaveAsBox</b>	Saves a recipe from the buffer to the archive; a window is displayed to allow a name for the recipe to be entered; the name of the recipe is initially assigned to the dedicated buffer, therefore backup is performed; in case of overwriting, a window is displayed requesting confirmation
<b>RecipeUploadBuffer</b>	Uploads the recipe indicated by the input parameters to the buffer from the device. The ID of the recipe type must be supplied
<b>RecipeDownloadBuffer</b>	Allows to download a recipe from the buffer to the device (PLC)
<b>RecipeClearBuffer</b>	Deletes all tag buffer contents of the data structure

# CREW Manual

Image On: to associate an image to the icon when it is active.



Opacity On: to set the level of opacity of the image to be associated with the icon when it is active.



Image Off: to associate an image to the icon when it is not active.



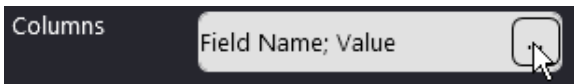
# CREW Manual

Opacity Off: to set the level of opacity of the image to be associated with the icon when it is not active.

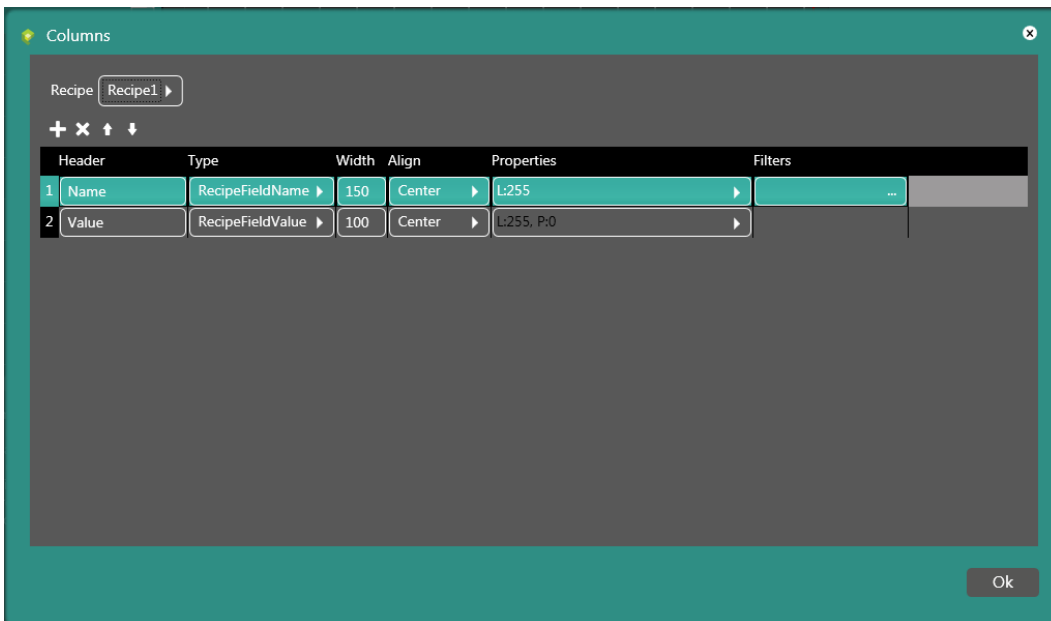


“Columns” option

Click the “Browse” key.



On the window that appears, it is possible to decide how to set the Recipe view.



# CREW Manual

Using the icons displayed in the image to add (+) or eliminate (X) new functions and move them up or down.



Heading: the title of the columns that comprise the table.

Type: the type of Recipe field, which can be selected from the options shown in the image.



Width: the width of the table columns.

Properties: to customise how the type of recipe is viewed. If "RecipeFieldName" is the selected "Type" field, the relative pages are the ones shown in the image.

Type	Width	Properties
RecipeFieldName ▶	200	L:255, M:False
RecipeFieldValue ▶	200	Max Length <input type="text" value="255"/> Multiline <input type="checkbox"/>

# CREW Manual

From the properties it is possible to define the maximum length of the recipe name (255 characters by default) and activate or deactivate the “Multiline” option.

If, on the other hand, the “RecipeFieldValue” is the selected “Type” field, the relative properties are the ones shown in this image.

Type	Width	Properties						
RecipeFieldValue ▶	200	L:255, M:False, P:0						
RecipeFieldValue ▶	200	<table border="1"> <tr> <td>Max Length</td> <td>255</td> </tr> <tr> <td>Multiline</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Precision</td> <td>0</td> </tr> </table>	Max Length	255	Multiline	<input type="checkbox"/>	Precision	0
Max Length	255							
Multiline	<input type="checkbox"/>							
Precision	0							

From the properties it is possible to define the maximum length of the recipe name (255 characters by default) and activate or deactivate the “Multiline” option, and establish the decimal precision of the recipe field, namely the number of digits after the decimal point.

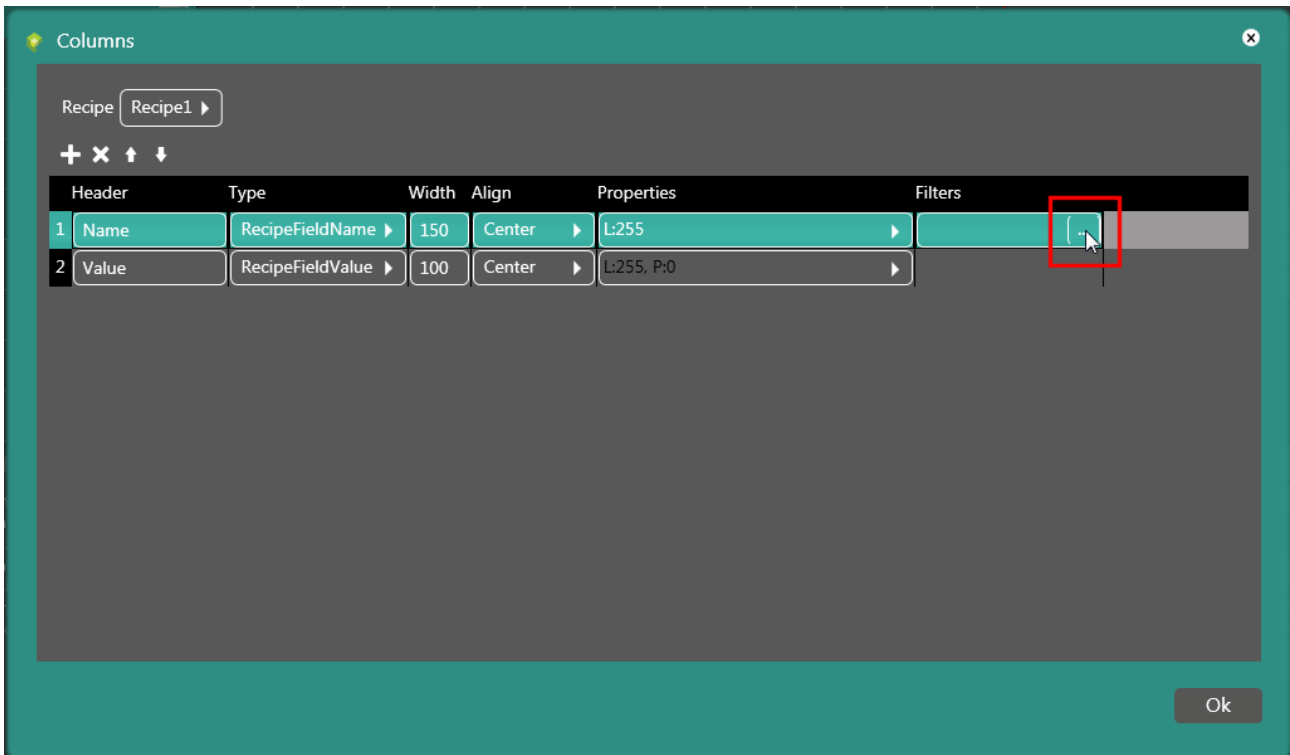
## Filters



With the Filters option it is possible to enable Runtime display filters for the alarm log. For example, it is possible to associate a filter to the “RecipeFieldName” function to display only the alarms within a given period of time (chosen by the user).

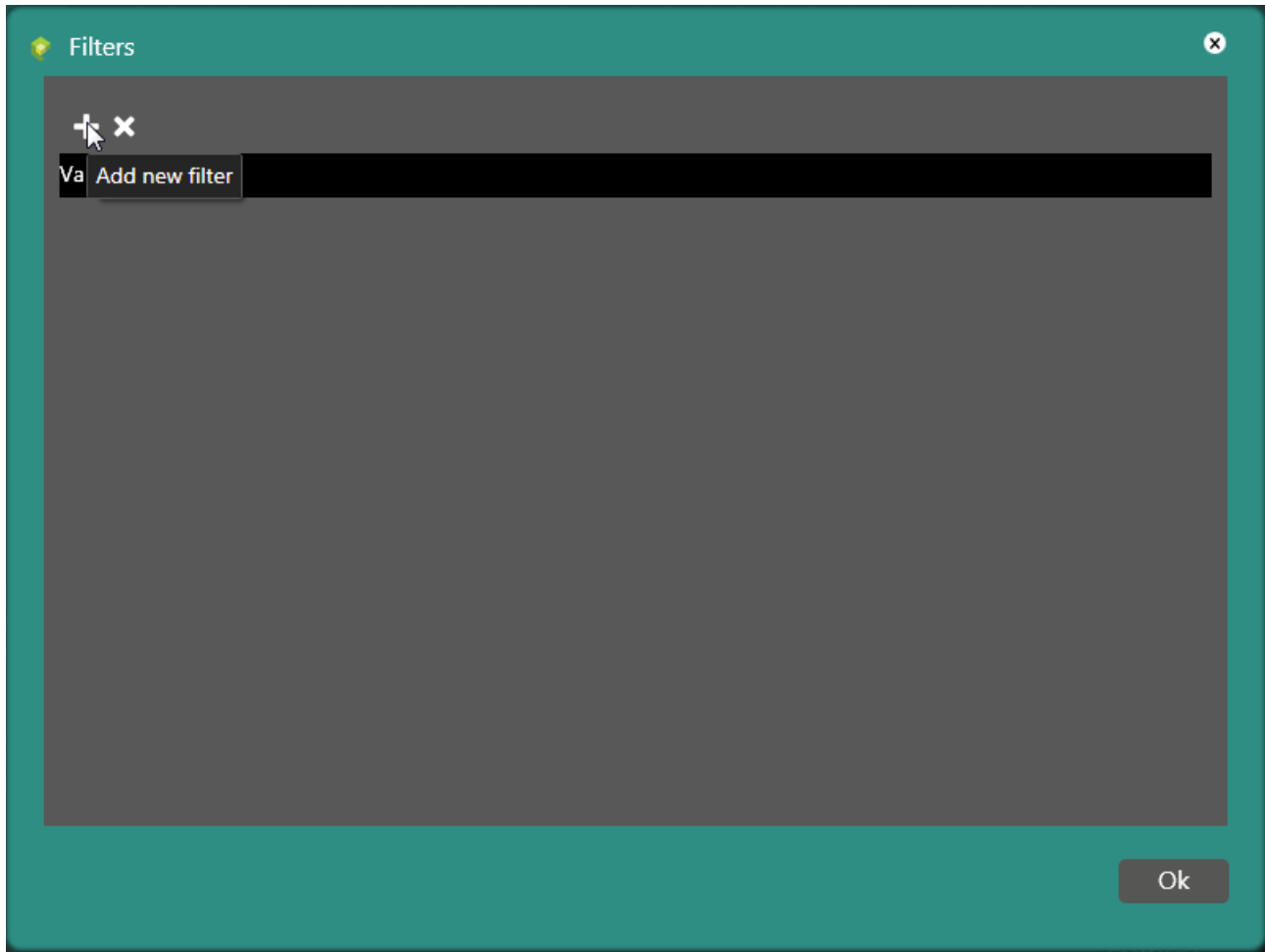
# CREW Manual

To enable a filter, click “Browse” in the “Filters” column.



# CREW Manual

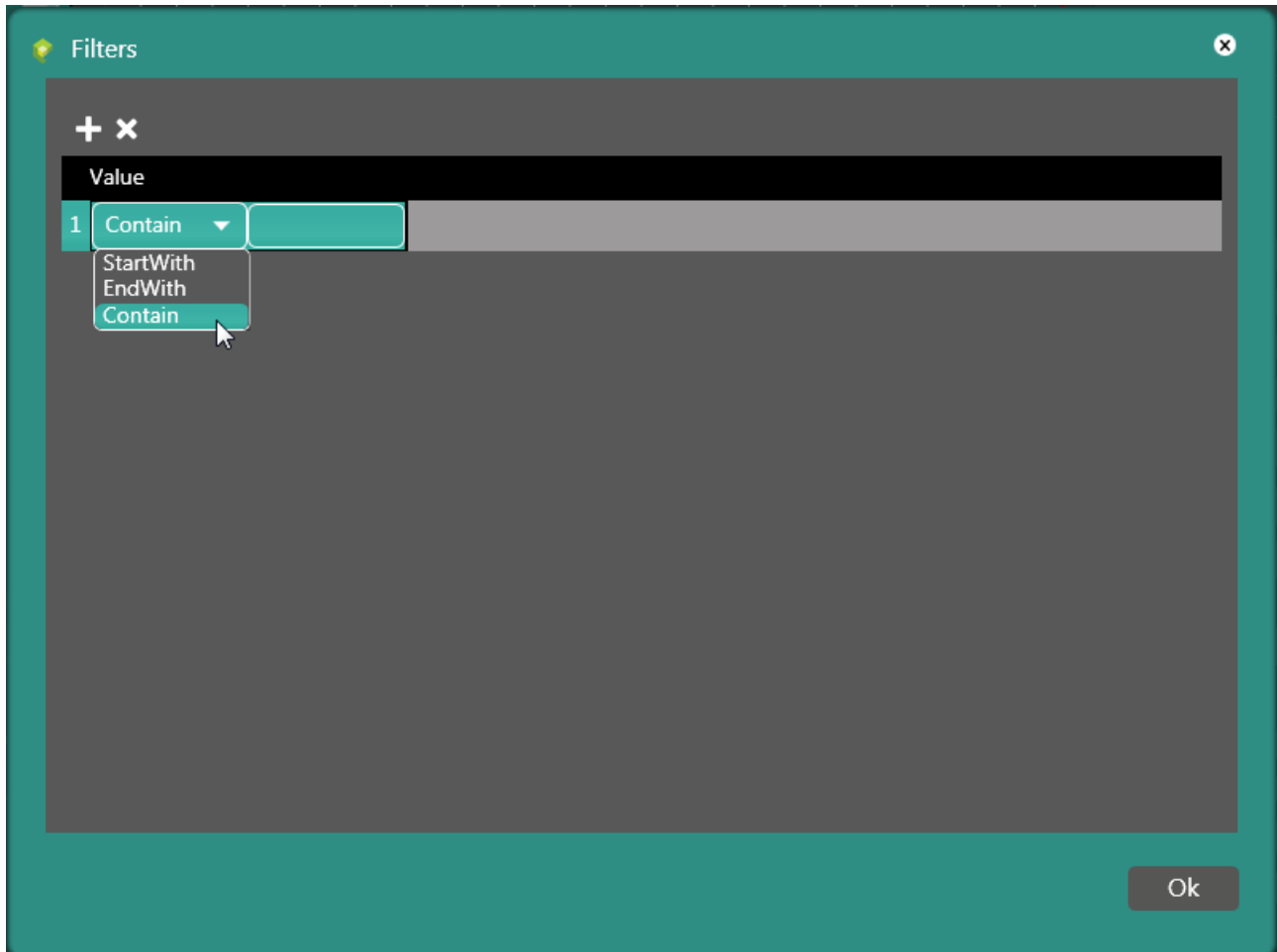
Click “Add filter”.





# CREW Manual

Select the “Contain” option.



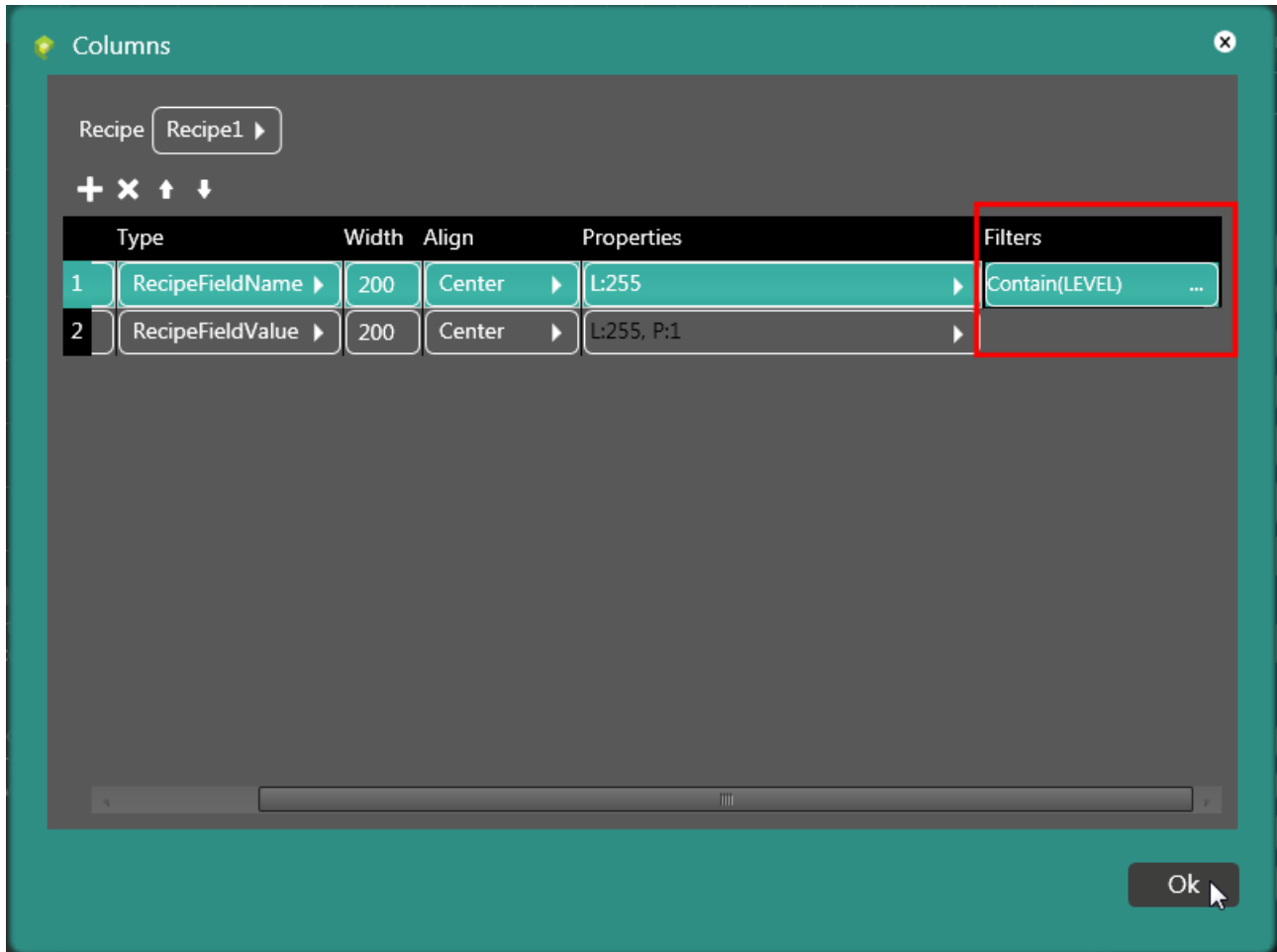
# CREW Manual

Enter the word “LEVEL” to apply the filter to all of the recipe elements containing the word “LEVEL”.



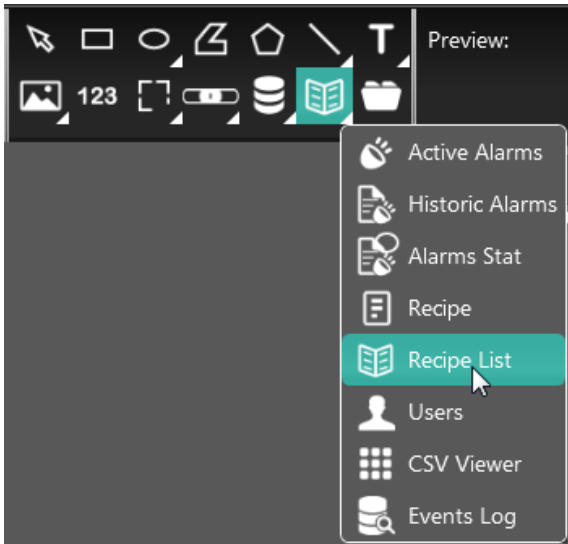
# CREW Manual

The filter will now be displayed in the “Filters” column.

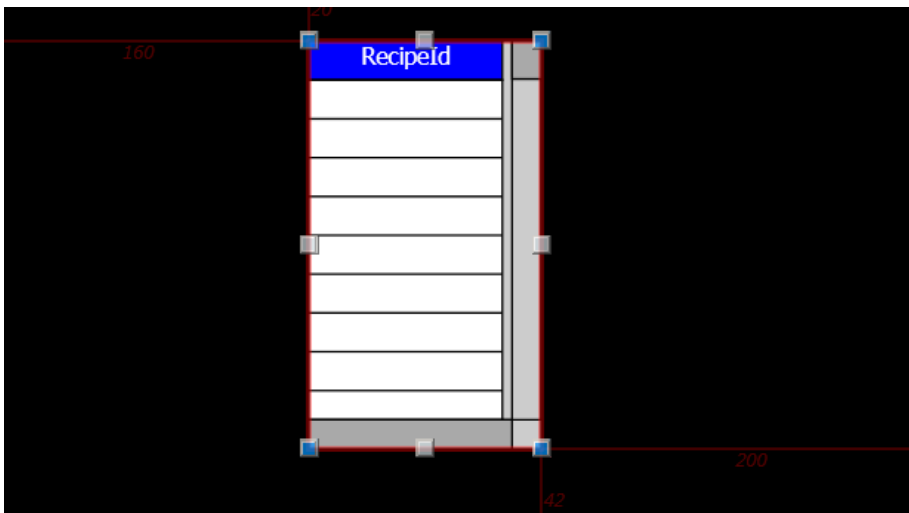


# CREW Manual

## Recipe List



The "Recipe List" icon on the "Graphics" menu is used to place a recipe list on the page, drawing it with the mouse on the page.

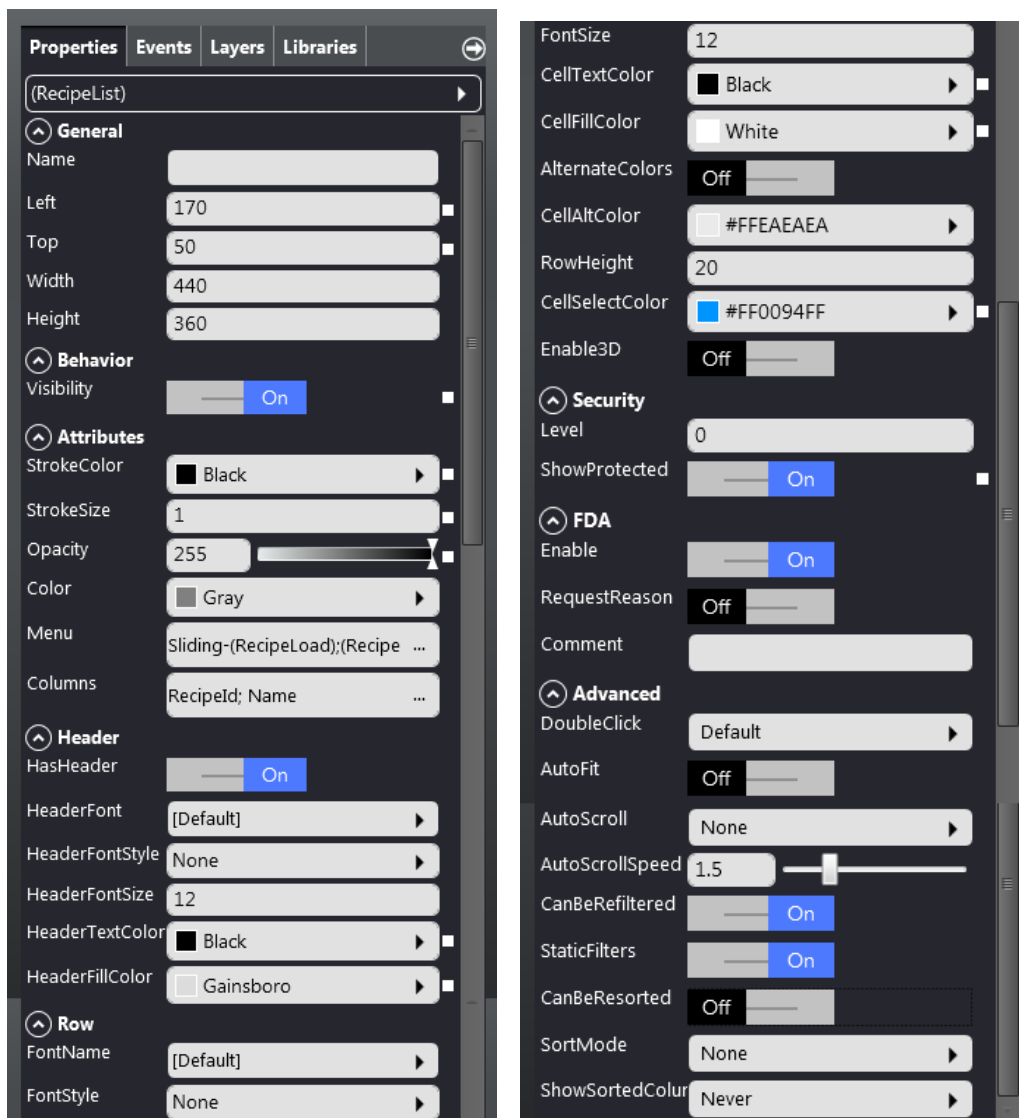


To determine the features of the "Recipe List" field, set them in the "Properties Editor", as shown in the section "[Recipe List Properties](#)".

# CREW Manual

## Recipe List Properties

The following image illustrates all the editable properties of the Recipe List. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

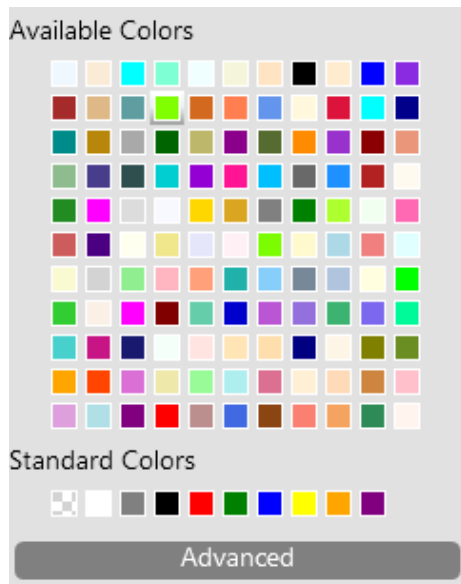
The following table describes all the editable properties of the “Recipe List”.

Properties	Description
<b>General</b>	
<b>Name</b>	Object ID
<b>Left</b>	Horizontal coordination of position
<b>Top</b>	Vertical coordination of position
<b>Width</b>	Width
<b>Height</b>	Height
<b>Behavior</b>	
<b>Visibility</b>	Determines whether the object should be displayed or not
<b>UserMode</b>	Determines whether the user can interact with the object: - read only, the user cannot interact with the object; this value takes precedence over the Disabled property - read-write, the user can interact with the object; the effective interaction is validated by the Disabled property
<b>Attributes</b>	
<b>StrokeColor</b>	Determines the fill colour of the object; the colour is selectable by RGB colour code or colour palette
<b>StrokeSize</b>	Determines the stroke colour (edges of the figure); the colour is selectable by RGB colour code or colour palette
<b>Opacity</b>	Determines the opacity of the object
<b>Color</b>	Allows you to assign the color to the object using the color bar (Hue) (valid for objects in the Library)
<b>Menu</b>	By clicking on the “Browse” menu option, you can make a Recipe view management as described in this section
<b>Columns</b>	Clicking on the “Browse” button of the “Columns” option, you can determine which columns to appear in the table and define their properties (as described in this section)
<b>Header</b>	
<b>HasHeader</b>	Determines whether the Recipe view must have the header or not
<b>HeaderFont</b>	Determines the font type used to display the header text
<b>HeaderFontStyle</b>	Header font style. Any combination of the following features: - None - Italics - Bold - Underline
<b>HeaderFontSize</b>	Indicates the font size of the values written in the headers
<b>HeaderTextColor</b>	Determines the color of the header text of Recipe view
<b>HeaderFillColor</b>	Determines the color of the cell that contains the header of Recipe view
<b>Row</b>	
<b>FontName</b>	Determines the font used for the items of Recipe view

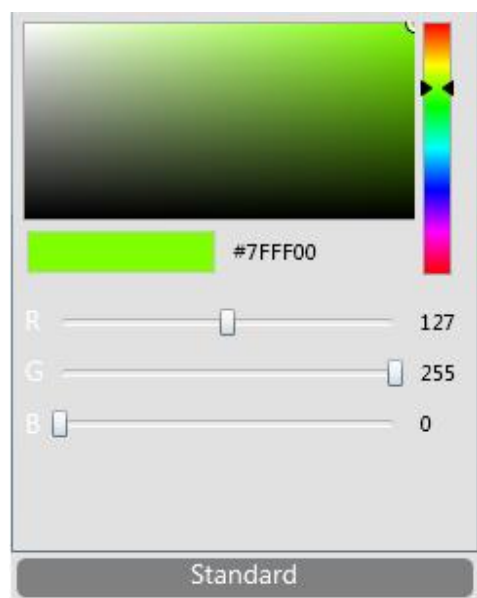
<b>FontStyle</b>	Font style. Any combination of the following features: - None - Italics - Bold - Underline
<b>FontSize</b>	Determines the stroke of Recipe view
<b>CellTextColor</b>	Represents the color of the Recipe view writing cells
<b>CellFillColor</b>	Determines the color of the cells of table columns
<b>AlternateColors</b>	Allows you to assign two alternating colors for each row in the table
<b>CellAltColor</b>	Determines the alternative color (active if the option “AlternateColors” is “ON”)
<b>RowHeight</b>	Determines the height of the row of the table (pixel)
<b>CellSelectColor</b>	Determines the color of the selected cell
<b>Enable3D</b>	Enable the 3D view (“embossed” view) of the table
<b>Security</b>	
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project
<b>ShowProtected</b>	Enables the displaying of a “status” icon in the object when it is protected for the current user
<b>FDA</b>	
<b>Enable</b>	Enables tracking of events related to the functions controlled through the grid menu
<b>RequestReason</b>	In each event to be recorded you have to insert the text that will be logged together with the time and date.
<b>Comment</b>	It defines a static comments to be tracked along with the execution events of the grid menu functions
<b>Advanced</b>	
<b>DoubleClick</b>	It allows you to assign a function to “double click” on each row of the table selecting it from the available ones
<b>AutoFit</b>	It determines whether the size of the columns of the grid are automatically calculated to fit precisely in the total available size
<b>AutoScroll</b>	It determines whether the content of the grid should automatically scroll
<b>AutoScrollSpeed</b>	It defines the scroll speed (when “AutoScroll” is enabled)
<b>CanBeRefiltered</b>	It determines whether the grid filters can be changed at runtime
<b>StaticFilters</b>	It defines the mode of interaction with the grid for access to the filter definition
<b>CanBeResorted</b>	It determines whether the grid lines can be ordered at runtime
<b>SortMode</b>	It defines the policy of the default grid sorting
<b>ShowSortedColumn</b>	It determines whether the column identified as sorting order must be highlighted in the grid

# CREW Manual

The properties related to colours can be edited through the colour palette.



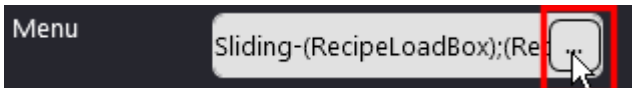
Click "Advanced" to select a colour using the RGB colour selection mask.



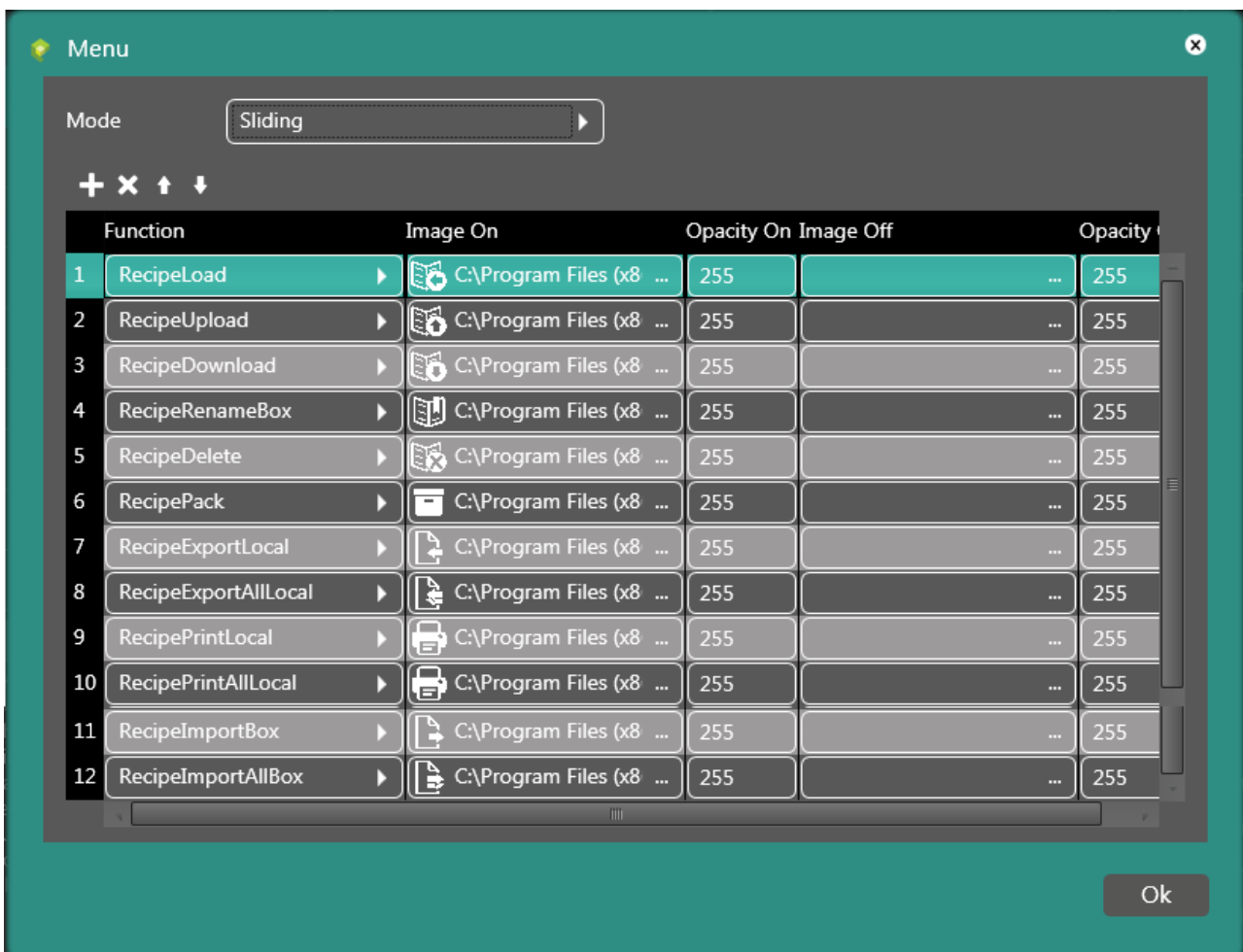
# CREW Manual

“Menu” option

Click the “Browse” key.



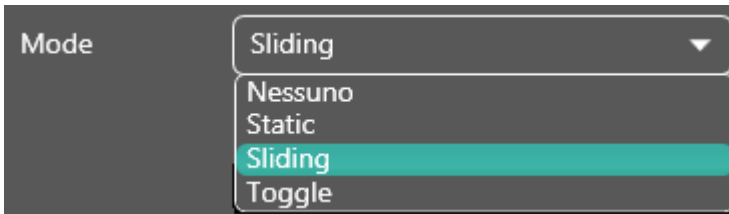
On the window that appears, it is possible to decide how to set the Runtime menu of the Recipe List view.





## CREW Manual

Mode: it is possible to customise the Runtime menu through one of the following options.

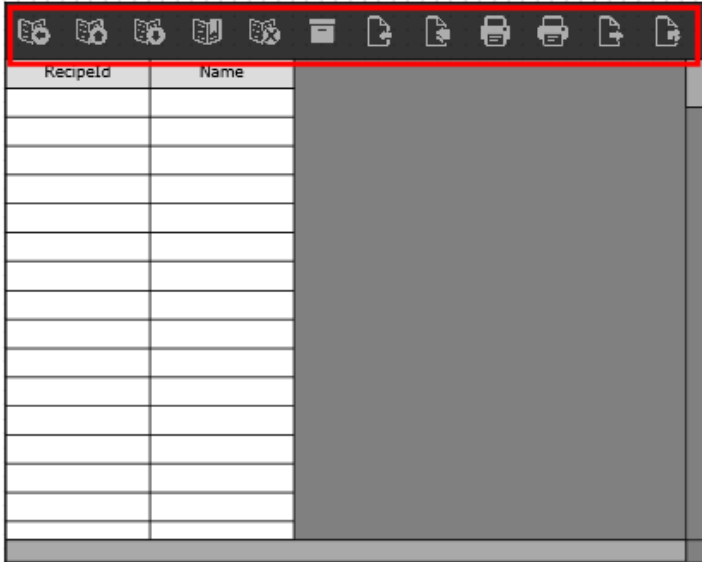


None: no Runtime menu. There are only the default columns.

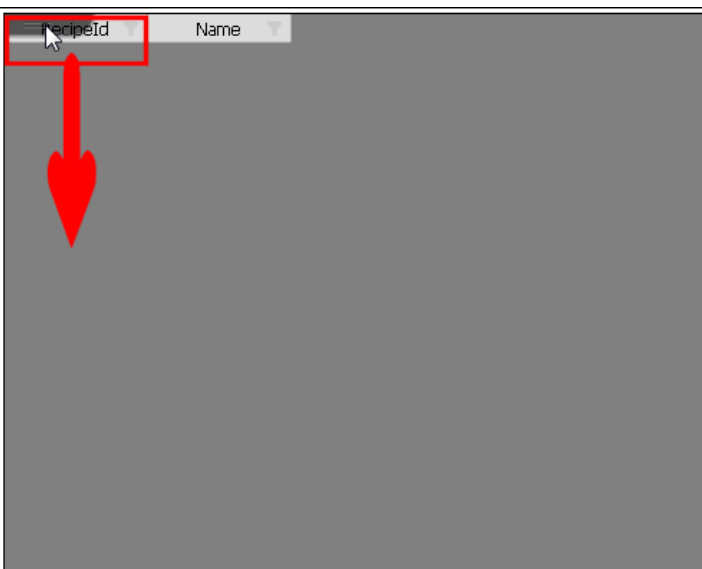
RecipeId	Name	

# CREW Manual

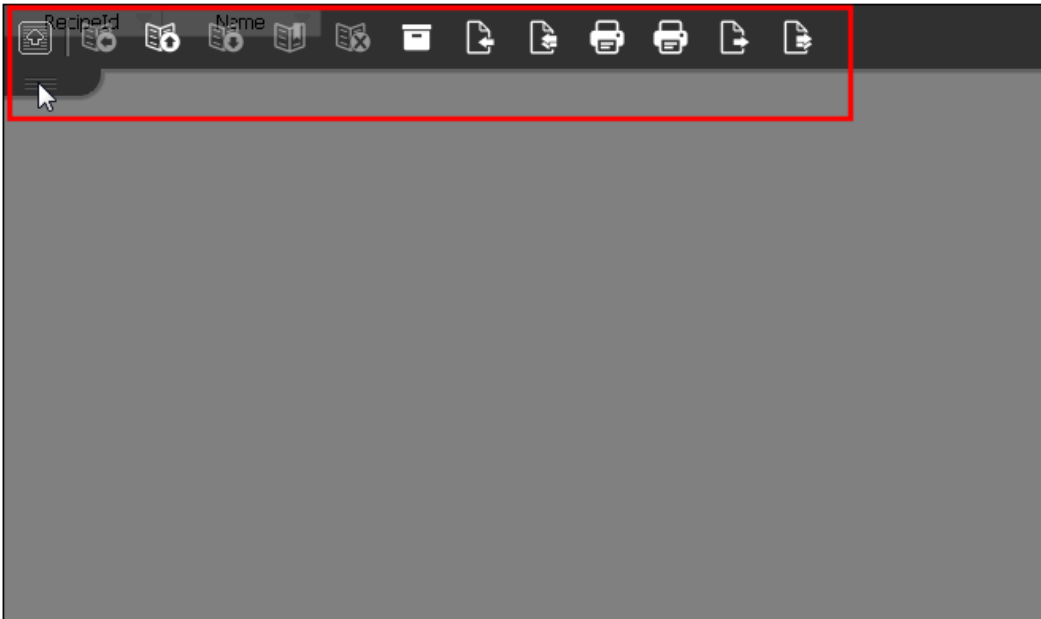
Static: static menu, namely fixed and always there.



Sliding: floating menu, which appears at the user's discretion.



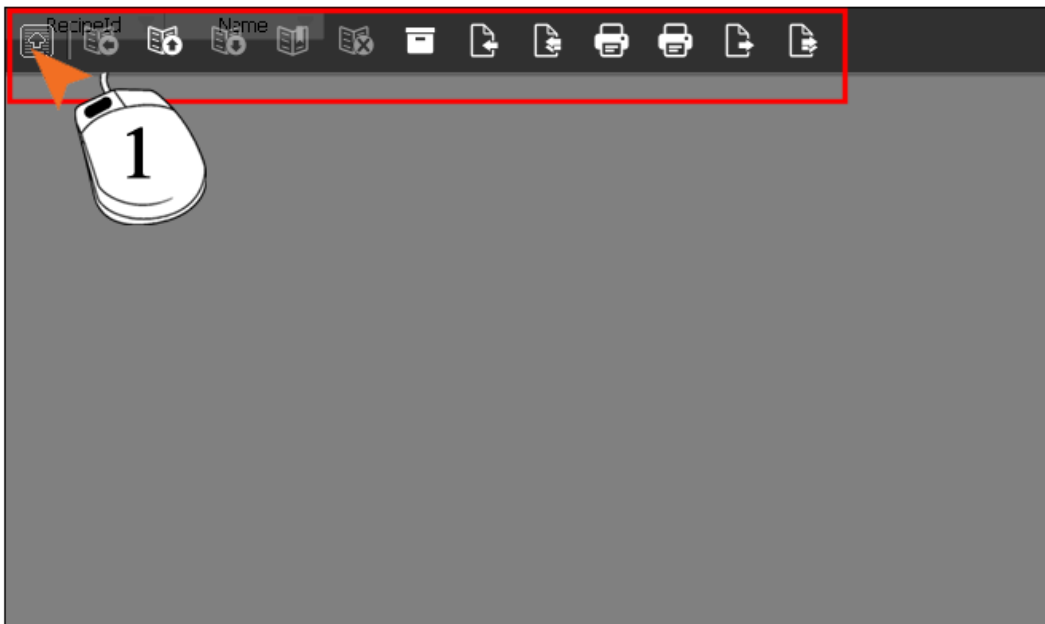
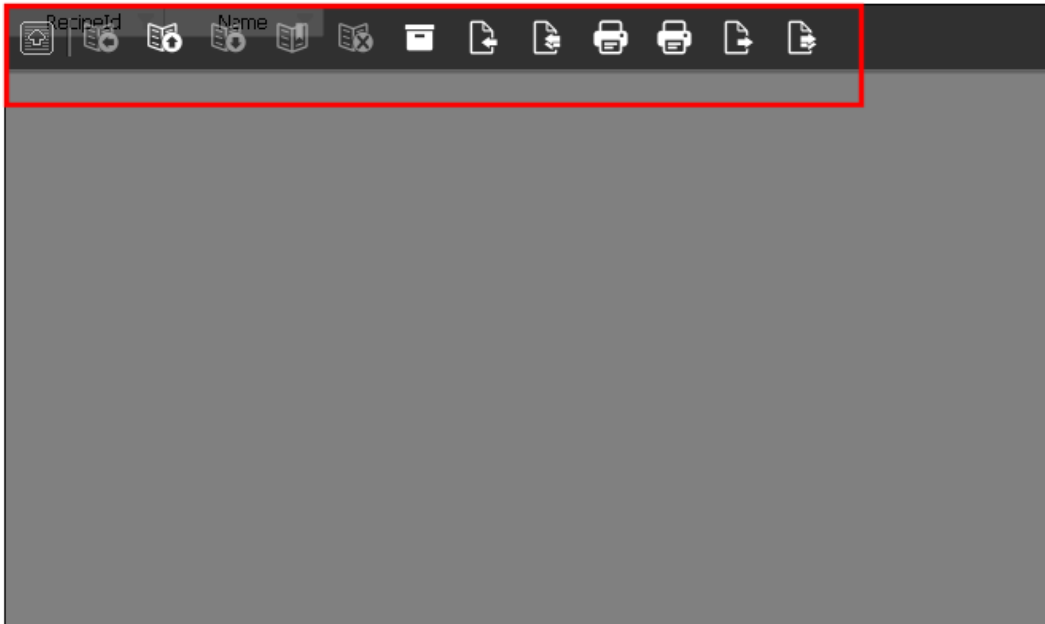
# CREW Manual



Toggle: floating menu (similar to the “Sliding” option), which appears and disappears at the user’s discretion with a click or double click.



# CREW Manual



# CREW Manual



Using the icons displayed in the image to add (+) or eliminate (X) new functions and move them up or down.



Function: this is used to assign a function from those shown in the image, to each of the icons that compose the Recipe List menu.

Function	
1	RecipeLoad ▶
2	RecipeUpload ▶
3	RecipeDownload ▶
4	RecipeRenameBox ▶
5	RecipeDelete ▶
6	RecipePack ▶
7	RecipeExportLocal ▶
8	RecipeExportAllLocal ▶
9	RecipePrintLocal ▶
10	RecipePrintAllLocal ▶
11	RecipeImportBox ▶
12	RecipeImportAllBox ▶

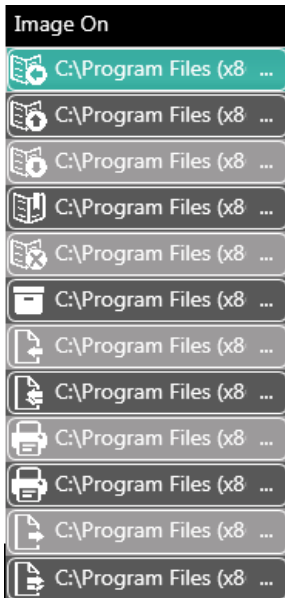
# CREW Manual

Table of functions that can be associated to the Recipe view.

Function	Description
<b>RecipeLoad</b>	Uploads a specific type of recipe; in CREW, recipe type to which this command refers must be specified
<b>RecipeUpload</b>	Uploads a recipe from the device to the archive (the buffer is not influenced during the operation); the device -PLC- variables must include the name of the recipe
<b>RecipeDownload</b>	Allows to download a recipe to the device; recipe type and name are required (the buffer is not influenced during this operation)
<b>RecipeRenameBox</b>	Changes the name of a recipe in the archive; a window is displayed to allow selection of the recipe to be renamed and editing of its new name
<b>RecipeDelete</b>	Allows to delete a recipe; recipe type and name are required
<b>RecipePack</b>	Compresses recipes contained in one archive; the operation may result in changing the ID of the existing recipes
<b>RecipeExportLocal</b>	Allows to export to a file (CSV) a recipe to the terminal; in CREW, recipe type to which this command refers must be specified; export is performed on the machine on which the user interface is enabled
<b>RecipeExportAllLocal</b>	Allows to export to a file (CSV) all present recipes, regardless of the type; export is performed on the machine on which the user interface is enabled
<b>RecipePrintLocal</b>	Print all the recipes of a given structure; a dialog box allows the selection of the target printer
<b>RecipePrintAllLocal</b>	Print all the recipes of all the existing structures; a dialog box allows the selection of the target printer
<b>RecipeImportBox</b>	Import the recipes contained in the given file, into the archive of the given structure, in this case the box allows the selection
<b>RecipeImportAllBox</b>	Import all the recipes contained in the given file, into the needed archives of the existing structures, in this case the box allows the selection

# CREW Manual

Image On: to associate an image to the icon when it is active.

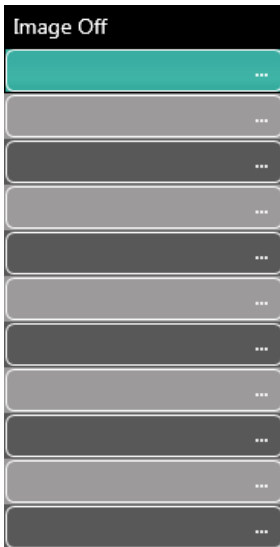


Opacity On: to set the level of opacity of the image to be associated with the icon when it is active.



# CREW Manual

Image Off: to associate an image to the icon when it is not active.

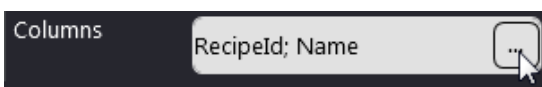


Opacity Off: to set the level of opacity of the image to be associated with the icon when it is not active.



“Columns” option

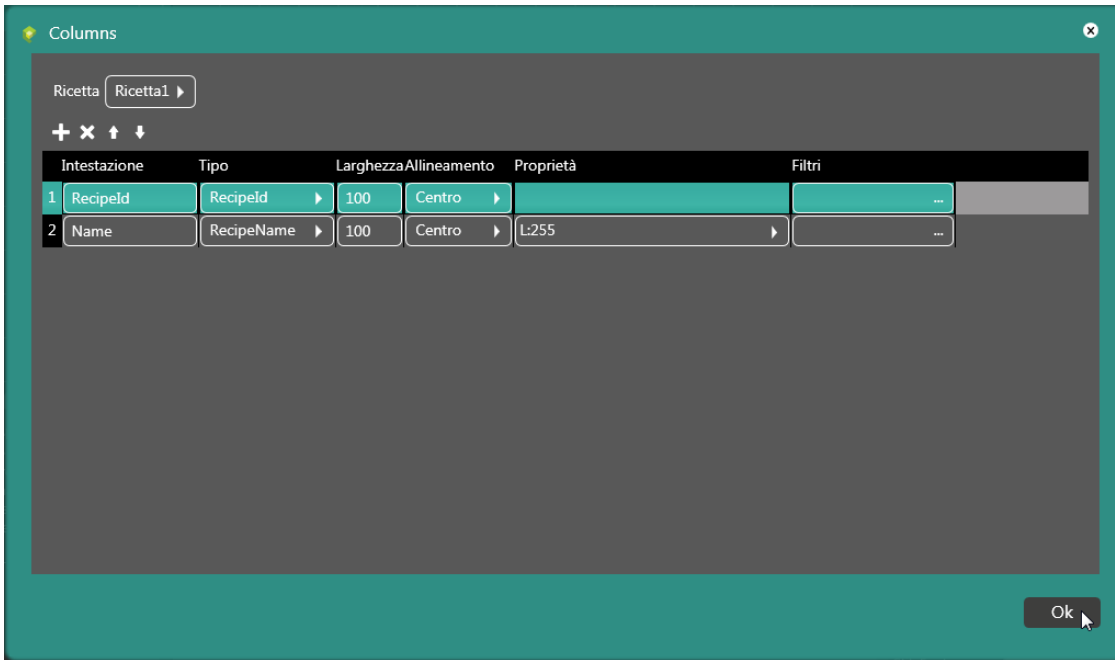
Click the “Browse” key.





# CREW Manual

On the window that appears, it is possible to decide how to set the Recipe List view.



Using the icons displayed in the image to add (+) or eliminate (X) new functions and move them up or down.



Heading: the title of the columns that comprise the table.

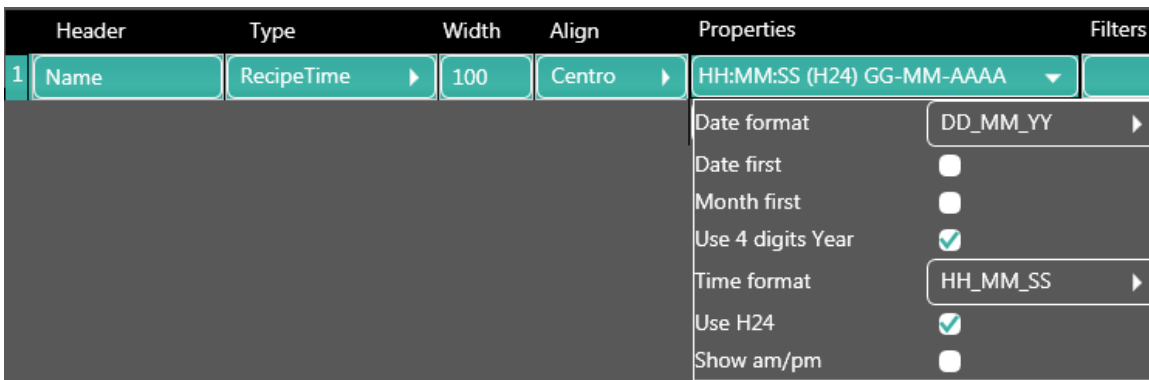
Type: to choose the type of Recipe field, which can be selected from the options shown in the image.



# CREW Manual

Width: the width of the table columns.

Properties: to customise the recipe type view. If "RecipeTime" is the selected "Type" field, the relative pages are the ones shown in the image.



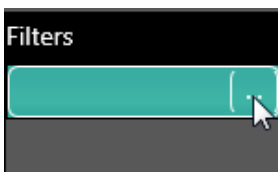
From the properties it is possible to define the format for date and time.

On the other hand, if the "RecipeName" is the selected "Type" field, the relative properties are the ones shown in this image.



From these properties it is possible to define the maximum length of the recipe name (255 characters by default).

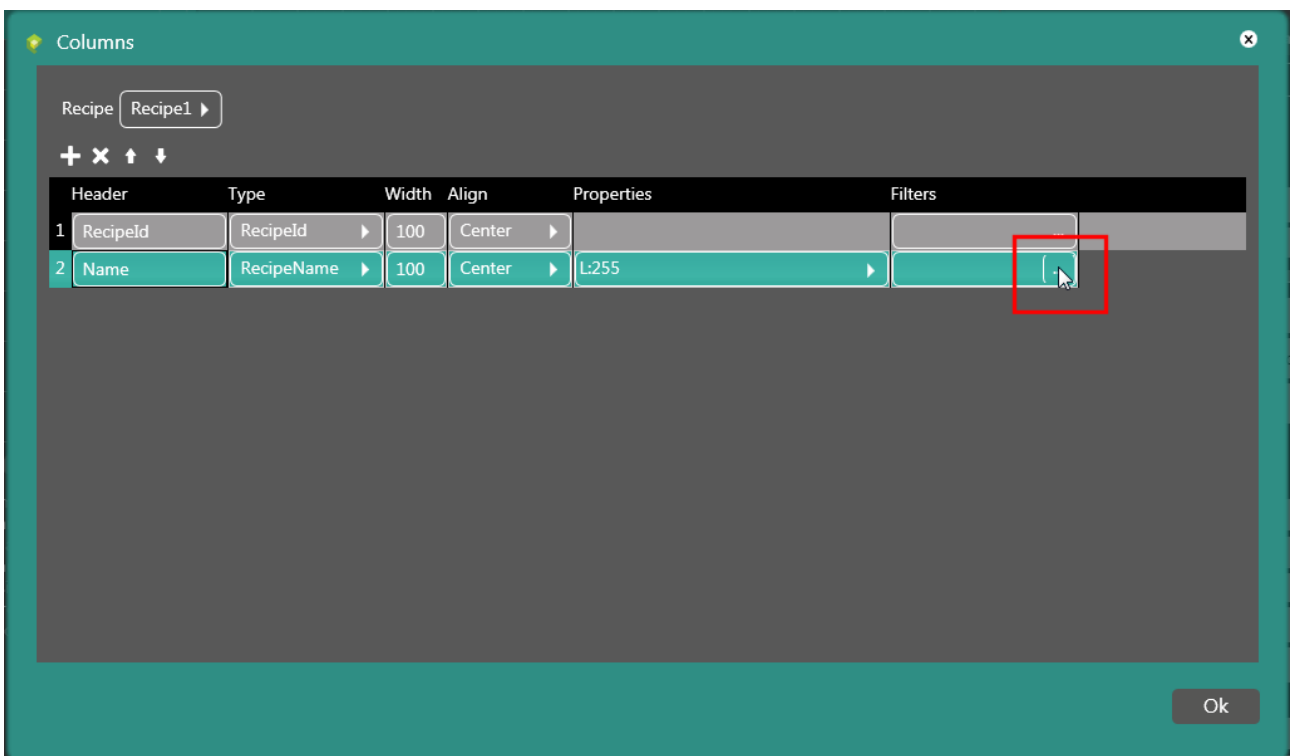
Filters



# CREW Manual

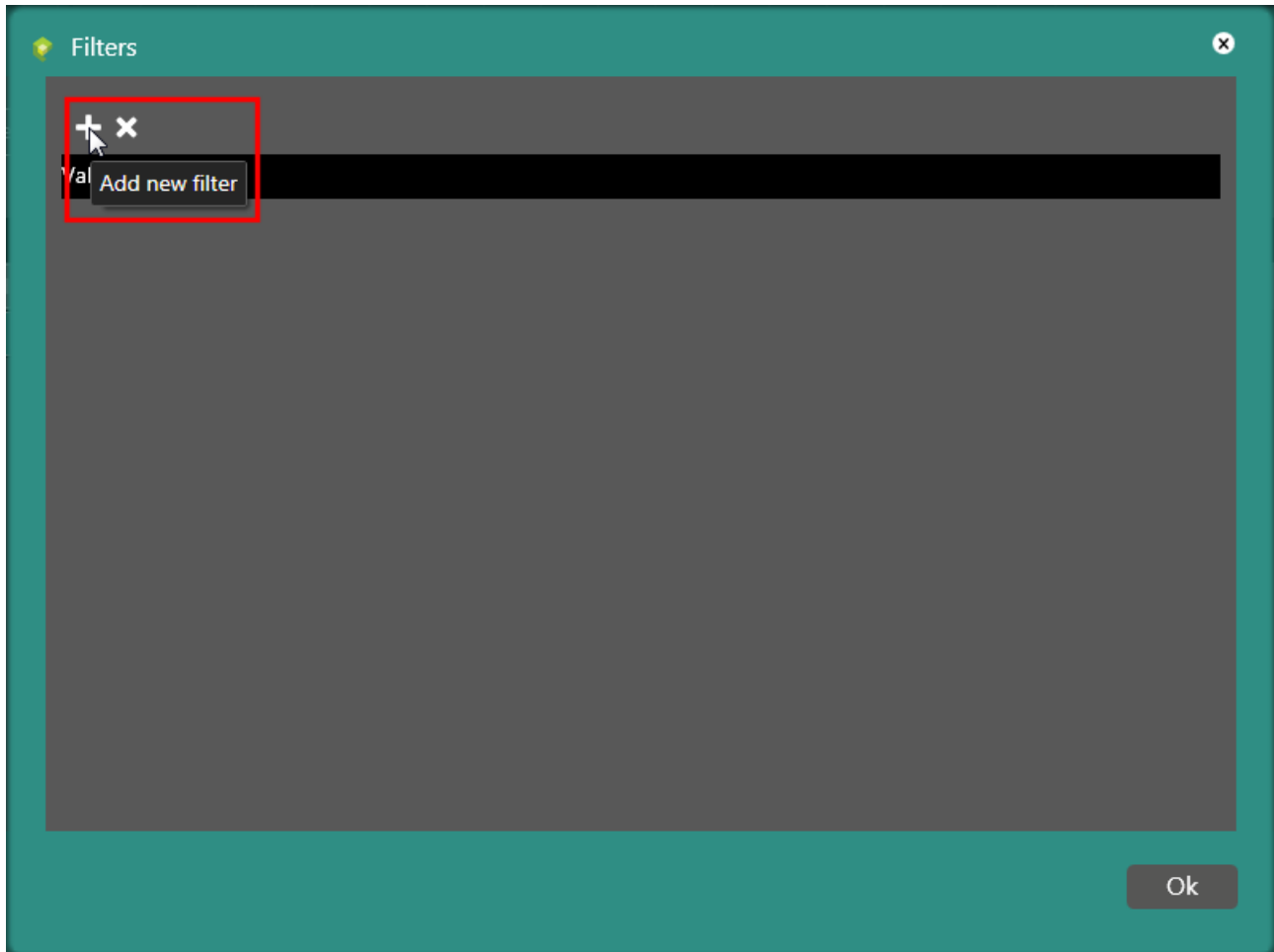
With the Filters option it is possible to enable Runtime display filters for the Recipe List view. For example, you can associate a filter to the "RecipeName" function to display, in Runtime, only recipes with names containing a specific word or a series of characters.

To enable a filter, click "Browse" in the "Filters" column.



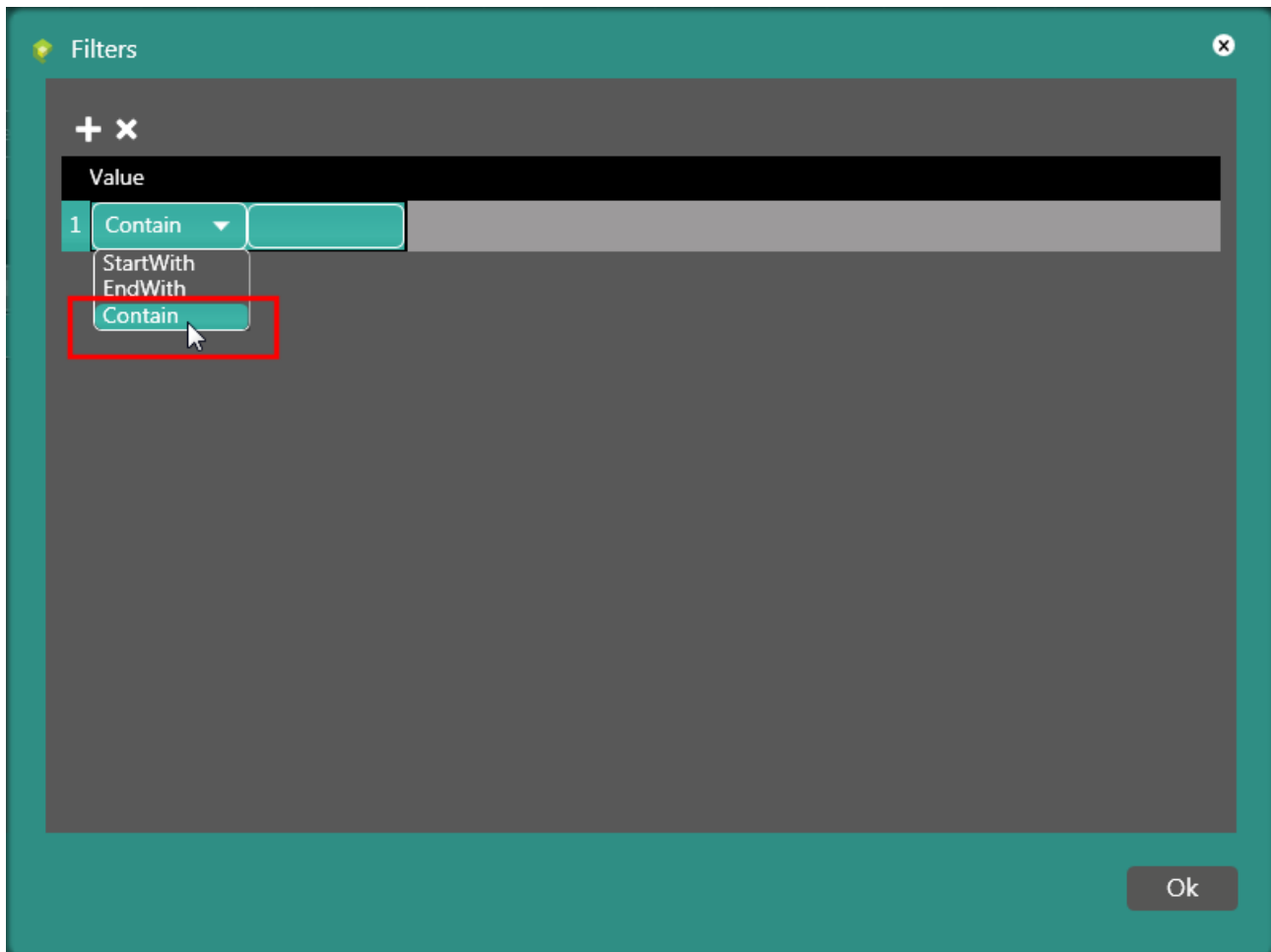
# CREW Manual

Click “Add filter”.



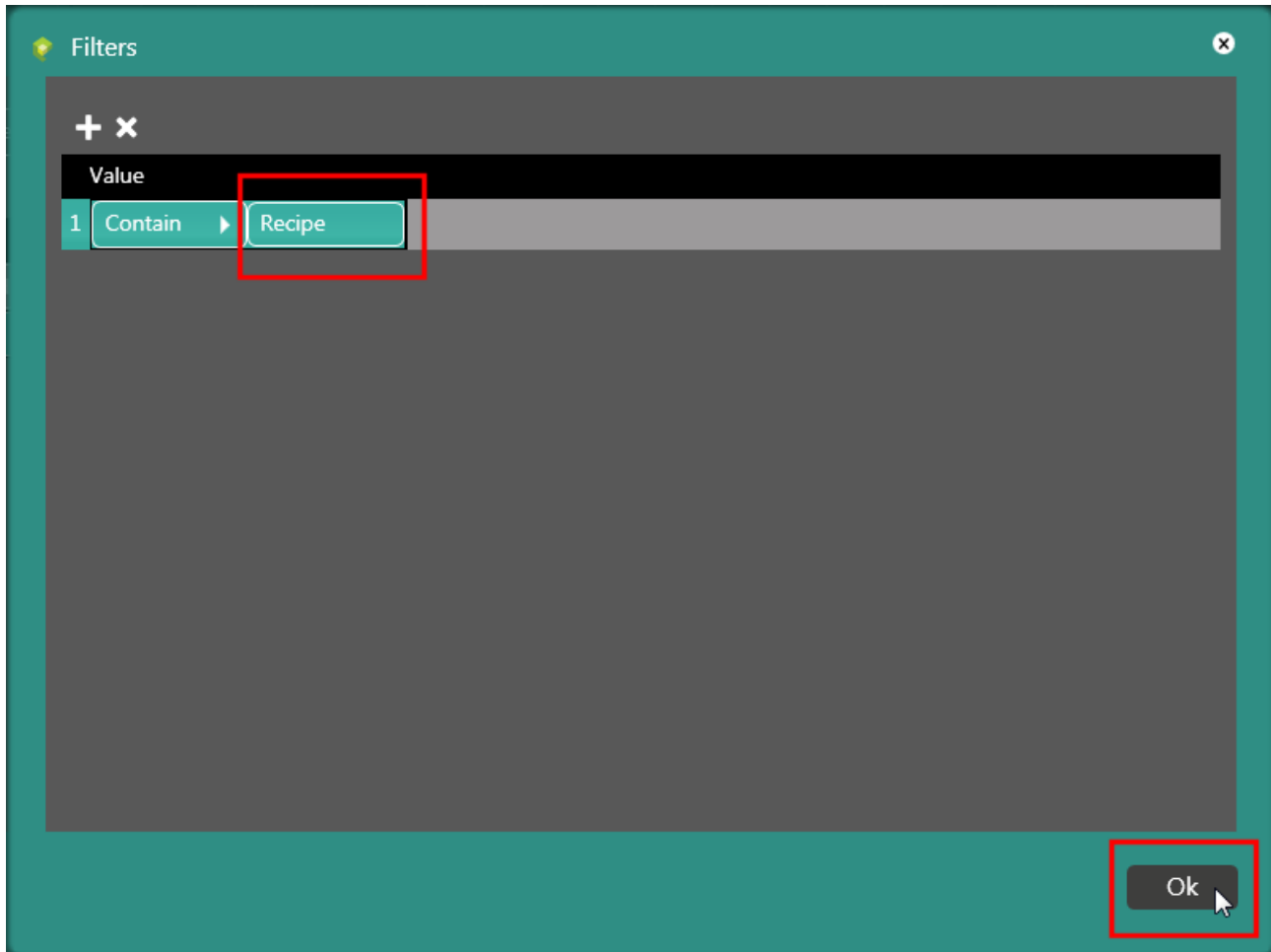
# CREW Manual

Select the “Contain” option.



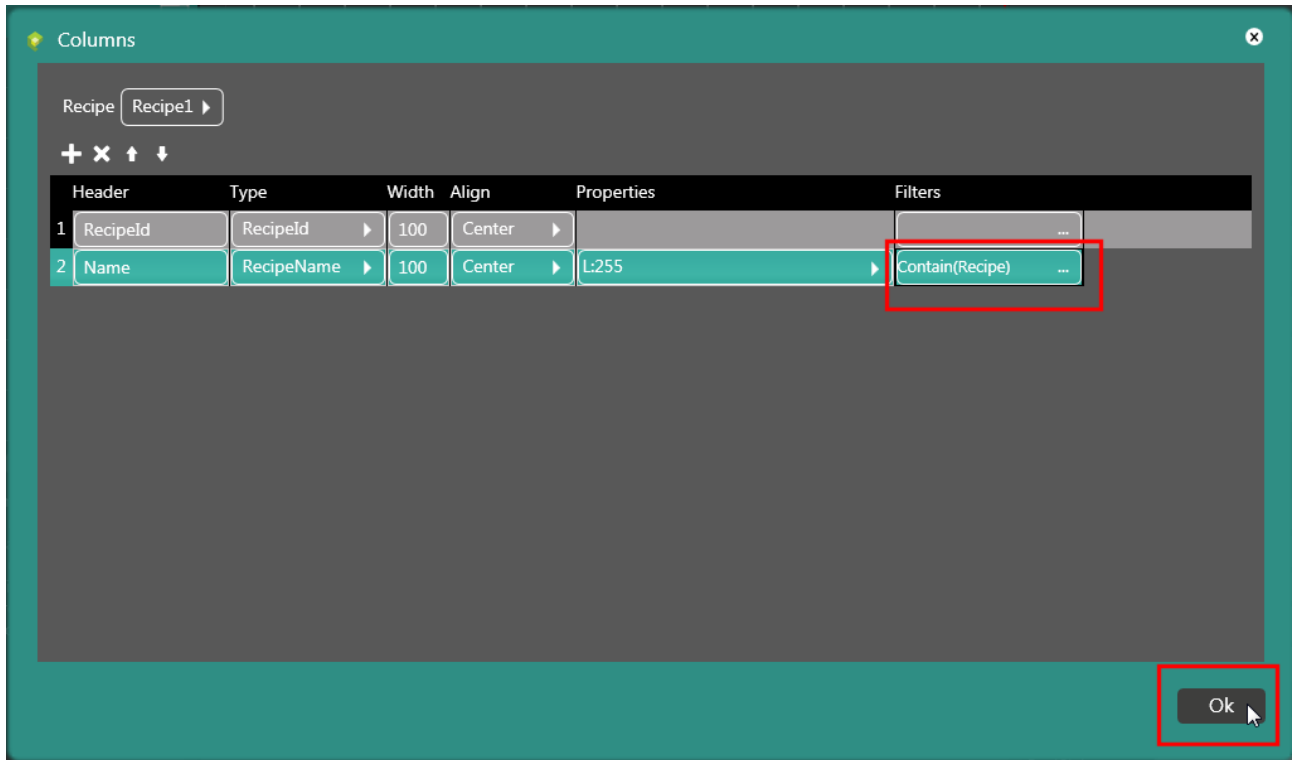
# CREW Manual

Enter the word “Recipe” to apply the filter to all of the recipe elements containing the word “Recipe”.



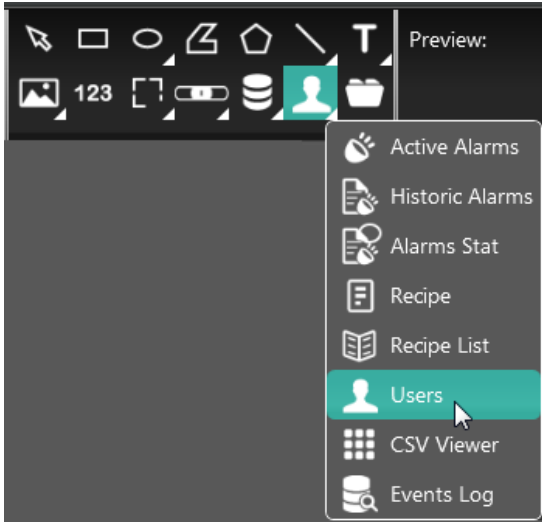
# CREW Manual

The filter will now be displayed in the “Filters” column.

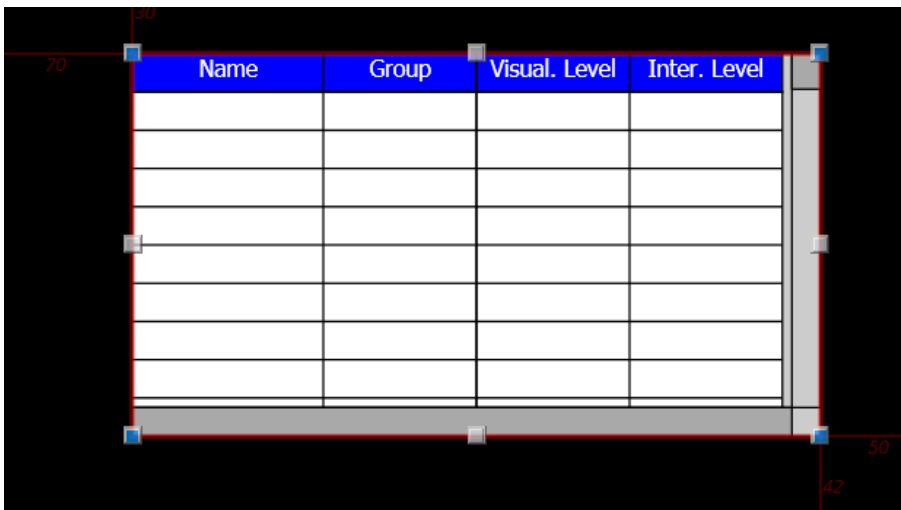


# CREW Manual

## Users



The “Users” icon on the “Graphics” menu is used to place a user table on the page, drawing it with the mouse on the page.



The screenshot shows a table being drawn on a page. The table has four columns: Name, Group, Visual. Level, and Inter. Level. The table is currently empty, with only the header row visible. The table is being drawn with a mouse, as indicated by the red lines and handles around the table.

Name	Group	Visual. Level	Inter. Level

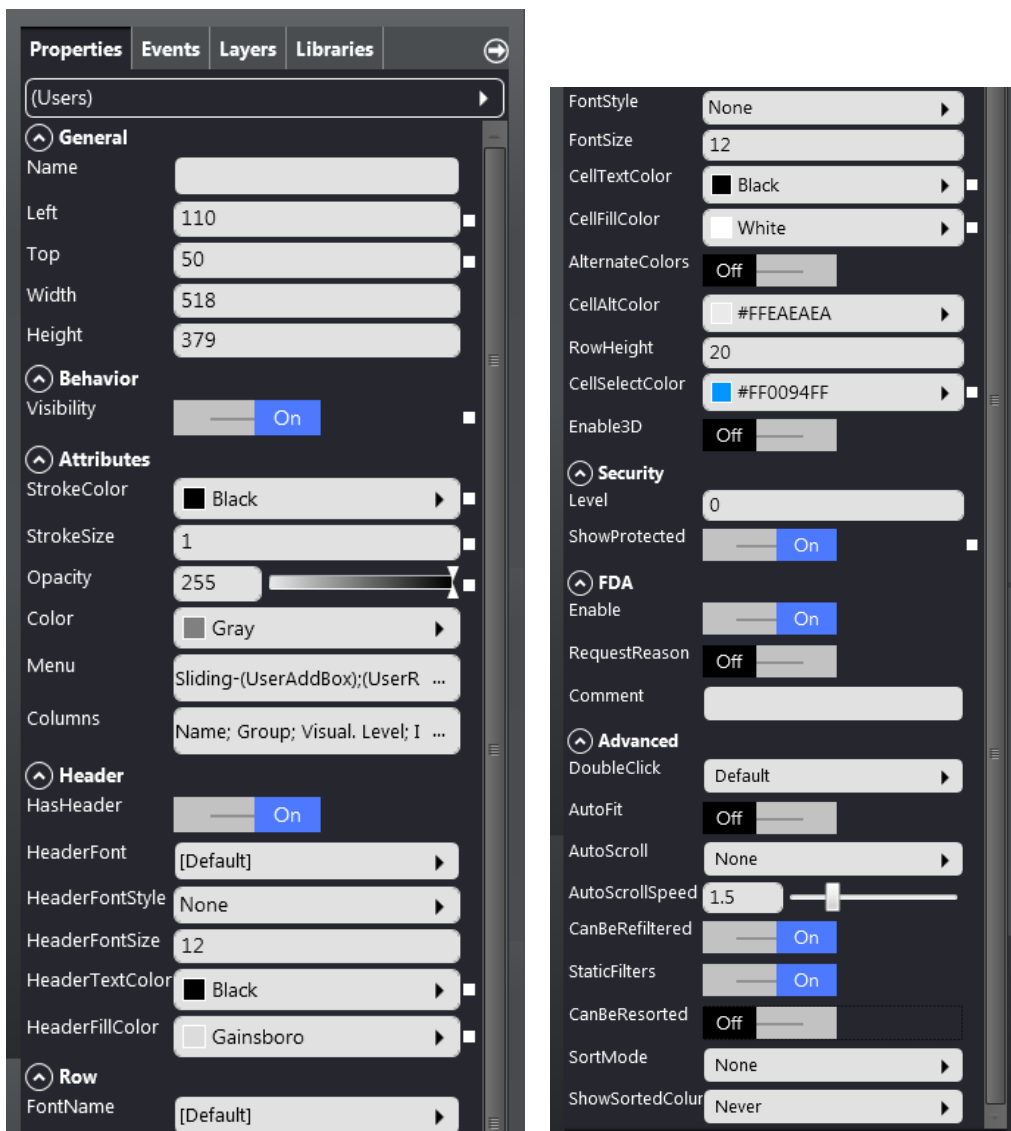
To determine the features of the “User” field it is necessary to set them in the “Properties Editor”, as shown in the “User Properties” section.



# CREW Manual

## User Properties

The following image illustrates all the editable properties of the User table. The properties with the white square next to them make it possible to link both constant values (by entering the value directly in the edit field) and dynamic values (see "[Dynamic assignment of values to the properties](#)").



# CREW Manual

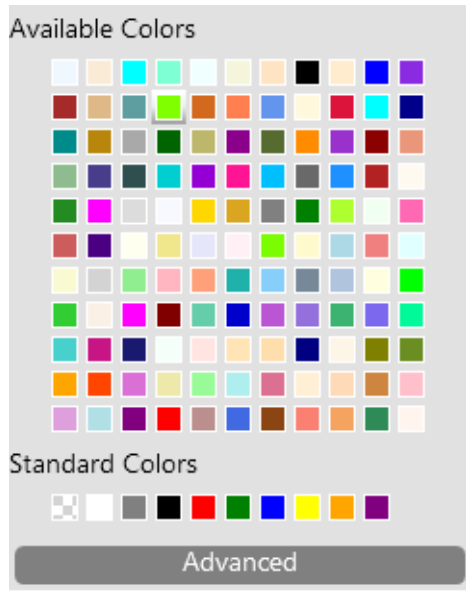
The following table describes all the editable properties of the User field.

Properties	Description
<b>General</b>	
Name	Object ID
Left	Horizontal coordination of position
Top	Vertical coordination of position
Width	Width
Height	Height
<b>Behavior</b>	
Visibility	Determines whether the object should be displayed or not
UserMode	Determines whether the user can interact with the object: - read only, the user cannot interact with the object; this value takes precedence over the Disabled property - read-write, the user can interact with the object; the effective interaction is validated by the Disabled property
<b>Attributes</b>	
StrokeColor	Determines the fill colour of the object; the colour is selectable by RGB colour code or colour palette
StrokeSize	Determines the stroke colour (edges of the figure); the colour is selectable by RGB colour code or colour palette
Opacity	Determines the opacity of the object
Color	Allows you to assign the color to the object using the color bar (Hue) (valid for objects in the Library)
Menu	By clicking on the "Browse" menu option, you can make a Users view management
Columns	Clicking on the "Browse" button of the "Columns" option, you can determine which columns to appear in the table and define their properties (as described in this section)
<b>Header</b>	
HasHeader	Determines whether the Users view must have the header or not
HeaderFont	Determines the font type used to display the header text
HeaderFontStyle	Header font style. Any combination of the following features: - None - Italics - Bold - Underline
HeaderFontSize	Indicates the font size of the values written in the headers
HeaderTextColor	Determines the color of the header text
HeaderFillColor	Determines the color of the cell that contains the header

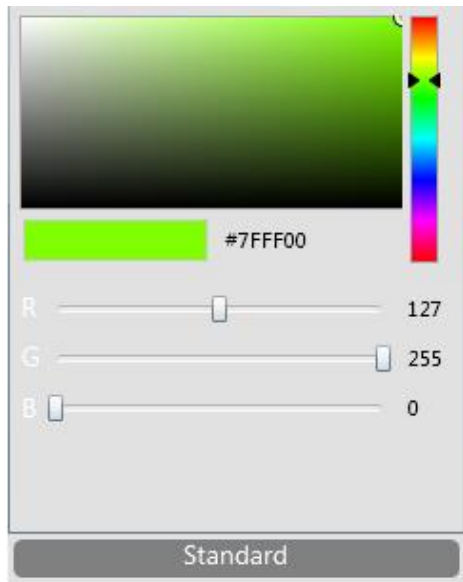
<b>Row</b>	
FontName	Determines the font used for the items
FontStyle	Font style. Any combination of the following features: - None - Italics - Bold - Underline
FontSize	Determines the font size of the text in the cells of the table columns of the User List
CellTextColor	Represents the writing color of the User List view cells
CellFillColor	Determines the color of the cells columns of table
RowHeight	Determines the height of the row of the table (pixel)
CellSelectColor	Determines the color of the selected cell
<b>Security</b>	
Level	Allows to define the levels of authentication required to control access to specific areas of the project
ShowProtected	Enables the displaying of a "status" icon in the object when it is protected for the current user
<b>FDA</b>	
Enable	Enables tracking of events related to the functions controlled through the grid menu
RequestReason	In each event to be recorded you have to insert the text that will be logged together with the time and date.
Comment	It defines a static comments to be tracked along with the execution events of the grid menu functions
<b>Advanced</b>	
DoubleClick	It allows you to assign a function to "double click" on each row of the table selecting it from the available ones
AutoFit	It determines whether the size of the columns of the grid are automatically calculated to fit precisely in the total available size
AutoScroll	It determines whether the content of the grid should automatically scroll
AutoScrollSpeed	It defines the scroll speed (when "AutoScroll" is enabled)
CanBeRefiltered	It determines whether the grid filters can be changed at runtime
StaticFilters	It defines the mode of interaction with the grid for access to the filter definition
CanBeResorted	It determines whether the grid lines can be ordered at runtime
SortMode	It defines the policy of the default grid sorting
ShowSortedColumn	It determines whether the column identified as sorting order must be highlighted in the grid

# CREW Manual

The properties related to colours can be edited through the colour palette.



Click "Advanced" to select a colour using the RGB colour selection mask.



# CREW Manual

## Details on Bézier Curves

A Bézier curve is a special parametric curve frequently used in computer graphics. The most important Bézier curves are the quadratic and cubic ones. When it is necessary to draw more complex shapes, however, multiple second and third order curves are joined as a Bézier spline.

### Linear Bézier curves:

Given points  $P_0$  and  $P_1$ , a linear Bézier curve is simply a straight line between those two points. This curve is given by:

$$\mathbf{B}(t) = (1 - t)\mathbf{P}_0 + t\mathbf{P}_1, t \in [0, 1].$$

### Quadratic Bézier curves:

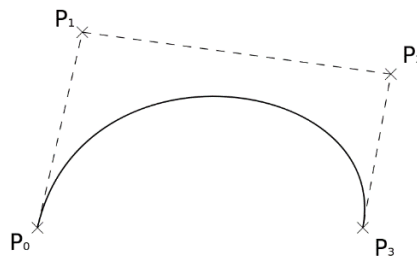
A quadratic Bézier curve is the path traced by the function  $B(t)$ , given points  $P_0$ ,  $P_1$ , and  $P_2$ .

$$\mathbf{B}(t) = (1 - t)^2\mathbf{P}_0 + 2t(1 - t)\mathbf{P}_1 + t^2\mathbf{P}_2, t \in [0, 1].$$

### Cubic Bézier curves:

Four points  $P_0$ ,  $P_1$ ,  $P_2$  and  $P_3$  in the plane or in three-dimensional space define a cubic Bézier curve. The curve starts at  $P_0$  going toward  $P_1$  and arrives at  $P_3$  coming from the direction of  $P_2$ . Usually, it will not pass through  $P_1$  or  $P_2$ ; these points are only there to provide directional information. The distance between  $P_0$  and  $P_1$  determines how long the curve moves into direction  $P_2$  before turning towards  $P_3$ . The parametric form of the curve is:

$$\mathbf{B}(t) = \mathbf{P}_0(1 - t)^3 + 3\mathbf{P}_1t(1 - t)^2 + 3\mathbf{P}_2t^2(1 - t) + \mathbf{P}_3t^3, t \in [0, 1].$$



See Note 1

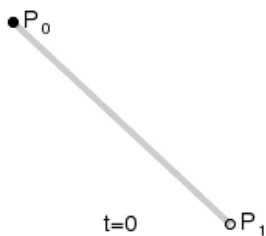
# CREW Manual

## Constructing Bézier curves

### Linear curves:

The  $t$  in the function for a linear Bézier curve can be thought of as describing how far  $B(t)$  is from  $P_0$  to  $P_1$ . For example when  $t=0.25$ ,  $B(t)$  is one quarter of the way from point  $P_0$  to  $P_1$ . As  $t$  varies from 0 to 1,  $B(t)$  describes a straight line from  $P_0$  to  $P_1$ .

Animation of a linear Bézier curve,  $t$  in  $[0,1]$ :

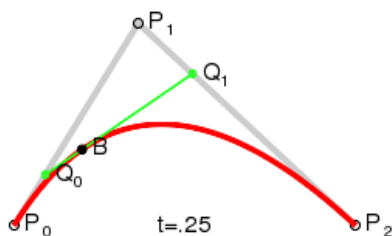


See Note 2

### Quadratic curves:

- For quadratic Bézier curves one can construct intermediate points  $Q_0$  and  $Q_1$  as  $t$  varies from 0 to 1;
- Point  $Q_0$  varies from  $P_0$  to  $P_1$  and describes a linear Bézier curve.
- Point  $Q_1$  varies from  $P_1$  to  $P_2$  and describes a linear Bézier curve.
- Point  $B(t)$  varies from  $Q_0$  to  $Q_1$  and describes a quadratic Bézier curve.

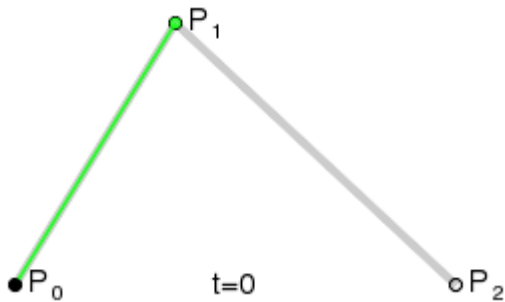
Construction of a quadratic Bézier curve:



See Note 3

# CREW Manual

Animation of a quadratic Bézier curve,  $t$  in  $[0,1]$ :



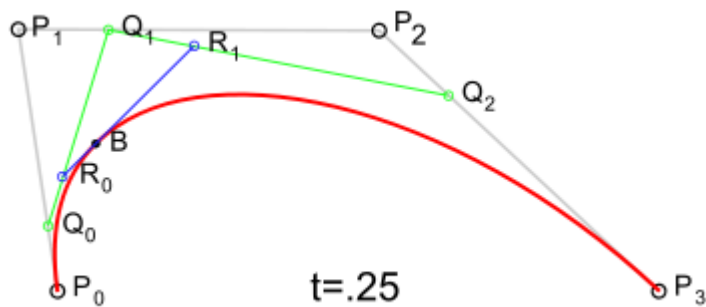
See Note 4

## Cubic and higher-order curves

For higher-order curves more intermediate points are required.

For cubic curves one can construct intermediate points  $Q_0$ ,  $Q_1$ , and  $Q_2$  that describe a linear Bézier curve, and points  $R_0$  &  $R_1$  that describe a quadratic Bézier curve.

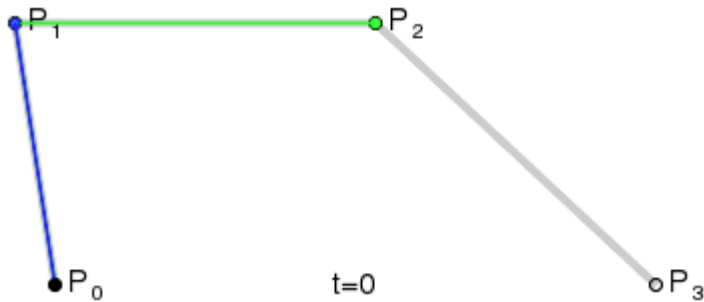
Construction of a cubic Bézier curve:



See Note 5

# CREW Manual

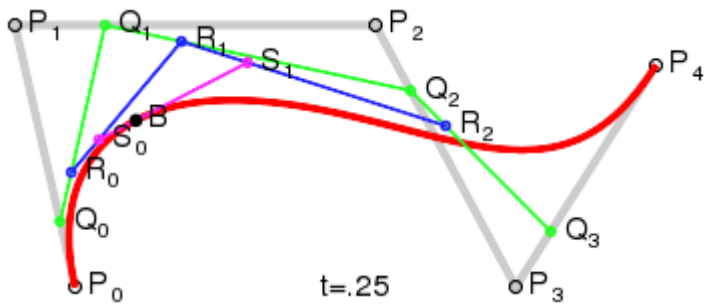
Animation of a cubic Bézier curve,  $t$  in  $[0,1]$ :



See Note 6

For fourth-order curves, one can construct intermediate points  $Q_0, Q_1, Q_2$  &  $Q_3$  that describe linear Bézier curves, points  $R_0, R_1$  &  $R_2$  that describe quadratic Bézier curves, and points  $S_0$  &  $S_1$  that describe a cubic Bézier curve.

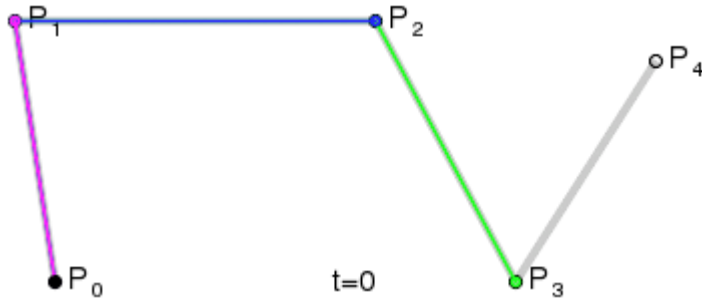
Construction of a quartic Bézier curve:



See Note 7

# CREW Manual

Animation of a quartic Bézier curve,  $t$  in  $[0,1]$ :



See Note 8

## Application in computer graphics:

Bézier curves are broadly used in computer graphics to model smooth curves. As the curve is completely contained in the convex hull of its control points, the points can be graphically displayed and can be used to manipulate the curve intuitively. Affine transformations such as translation and rotation can be applied on the curve by applying the respective transformation on the control points of the curve.



Notes:

### Note 1

- Source: Self-drawn using gedit, based on Image:Bezier.png
- Date: 12:48, 13 May 2006 (UTC)
- Author: Marian Sigler
- Licence: Public Domain

### Note 2

- Description: Bezier Curves Linear Bezier Curve
- Date: 28 February 2007
- Source: Created using ImageMagick
- Author: Phil Tregoning
- Licence: Own work, all rights released (Public domain)



# CREW Manual

## Note 3

- Description: Bezier Curves Quadratic Bezier Curve
- Date: 28 February 2007
- Source: Created using ImageMagick
- Author: Phil Tregoning
- Licence: Own work, all rights released (Public domain)

## Note 4

- Description: Bezier Curves Animated Quadratic Bezier Curve
- Date: 28 February 2007
- Source: Created using ImageMagick
- Author: Phil Tregoning
- Licence: Own work, all rights released (Public domain)

## Note 5

- Description: Cubic Bezier Curve
- Date: 22:39, 29 May 2009 (UTC)
- Source: Source: file Bezier\_3\_big.png
- Author: Author: Chris828
- Licence: Own work, all rights released (Public domain)

## Note 6

- Description: Bezier Curves Animated Cubic Bezier Curve
- Date: 28 February 2007
- Source: Created using ImageMagick
- Author: Phil Tregoning
- Licence: Own work, all rights released (Public domain)

## Note 7

- Description: Quartic Bezier Curve
- Date: 28 February 2007

# CREW Manual

- Source: Created using ImageMagick
- Author: Phil Tregoning
- Licence: Own work, all rights released (Public domain)

## Note 8

- Description: Quartic Bezier Curve
- Date: 28 February 2007
- Source: Created using ImageMagick
- Author: Phil Tregoning
- Licence: Own work, all rights released (Public domain)

Licence relative to the images of the "Bézier Curves".



I, the copyright holder of this work, release this work into the **public domain**. This applies worldwide.

In some countries this may not be legally possible; if so:

I grant anyone the right to use this work **for any purpose**, without any conditions, unless such conditions are required by law.

# CREW Manual

---

## FDA 21 CFR Part 11

FDA Regulation 21 CFR Part 11 establishes the guide lines for executing some the procedures of the production process.

It is advisable to underline that to validate environments according to these guide lines, the regulation does not necessarily involve the acquisition of specialised tools (analytical tools, document management systems, reporting tools, etc.), but verifying that all of the procedures adopted in the process observe certain requirements.

This is a crucial distinction, as the regulation itself often creates confusion among production administrators. Essentially, when 21 CFR part 11 says:

“The system must have several security levels. Each user must be assigned an account with unique username and password, both required to access the system. The identity and role of the user are therefore combined with the attributes of the control system for access to one or more documents, to determine whether they have access rights, or not, to a specific procedure”,

it is clear that the company must equip itself with:

- (a) A centralised account management system.
- (b) A security system.
- (c) Automatic role differentiation (administrator, analyst technician, operator, etc.).

The regulation does not suggest how to deal with these issues, however, it is clear that the production administrator is required to equip the production environment with due company process control tools (analytical tools, document management systems, reporting tools, etc.). Only when the administrator has all of the essential means can he/she request 21 CFR Part 11 validation. This validation consists in checking that all requested requirements are observed (from tracking operator access to the work bench, to the production of standardised reports for the administrators).

In sum, we can say that regulation 21 CFR Part 11 has four founding “pillars”:

- **Centralised security system:** Compulsory access by logging in with personal username and password.

# CREW Manual

- **Automatic differentiation of the roles:** For example, an operator cannot have access to the same analytical functions and data that a technical analyst has. In simple terms, an operator may view a Quality Control Card (updatable in real time), but may not view a series of other, more detailed information (statistics), such as process capacity indices, specific violations, log trend, etc.
- **Standardised reporting system:** The regulation explicitly requires “The possibility of generating legible, accurate and complete copies of records and in electronic format suitable for controls/inspections, revisitation, and copying for the agency”. It is therefore crucial to have software tools that are not only able to create modifiable and explorable tables and graphs, but also contain management information that does not necessarily pertain to the analysis phase.
- **Traceability of each single operation (Audit Trail):** The cornerstone of the regulation is that EVERY SINGLE OPERATION must be traced and belong to a log that can be consulted at any time, in the form of standard reports from system administrators or certifying agencies. In other words, the administrator must be able to establish “who did what, when, from what machine (and possibly why)” in a legible and editable form.

## Editor Properties

This section describes all the functions offered by Crew for editing the graphics and accessibility of the project apps.

The concept is that all executable procedures, any displayed data (editable and not), any link between the pages, and any function button, should appear to the operator within a page opens on the terminal.

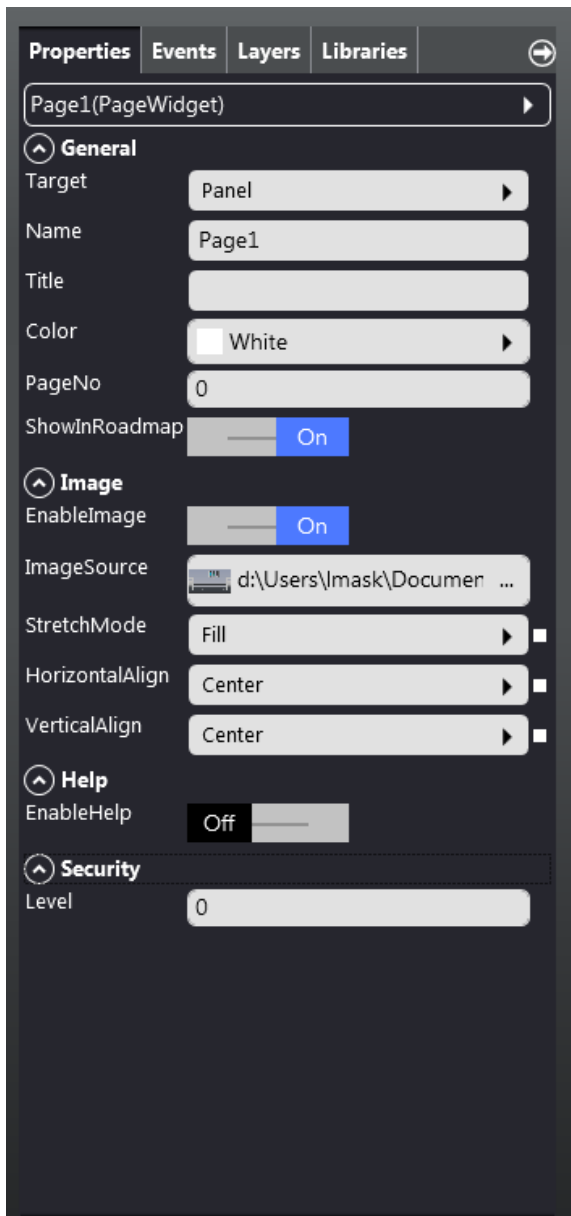
We will begin by mentioning the general organisation of the pages, then go into detail of all the elements to be entered and their features. For each graphic element that can be entered on a page (and for the pages themselves) a set of properties is defined that describes the appearance of the object at Runtime. Additionally, for many objects there are also functions or scripts that can apply when certain events occur.

The reference windows of the Editor Properties are as follows:

- Properties
- Events
- Layers
- Libraries

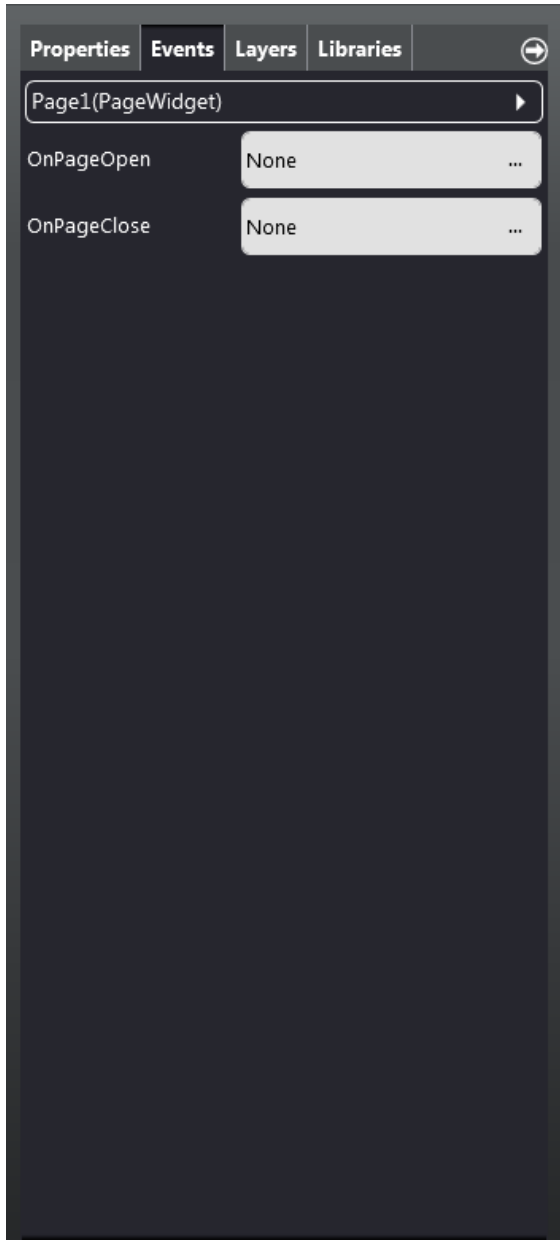
# CREW Manual

## Properties



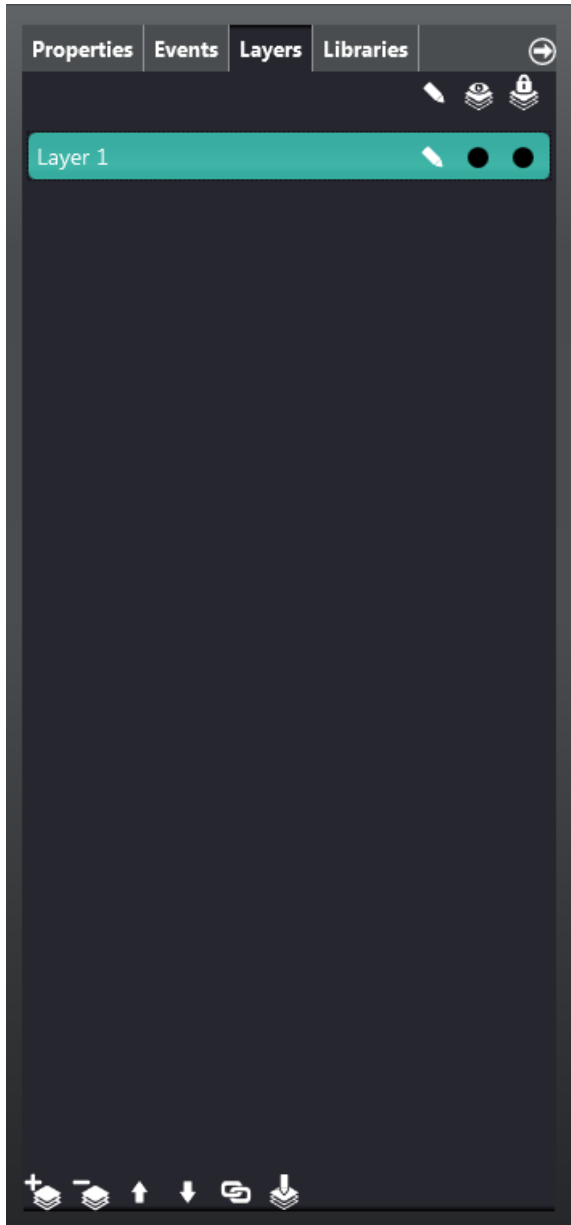
# CREW Manual

## Events



# CREW Manual

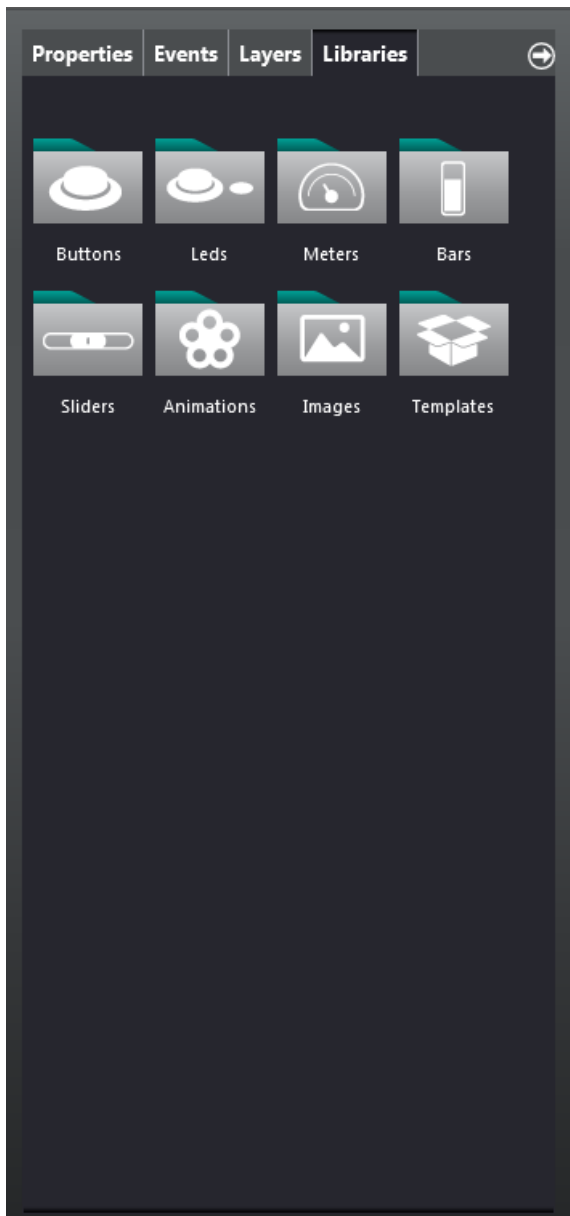
## Layers





# CREW Manual

## Libraries



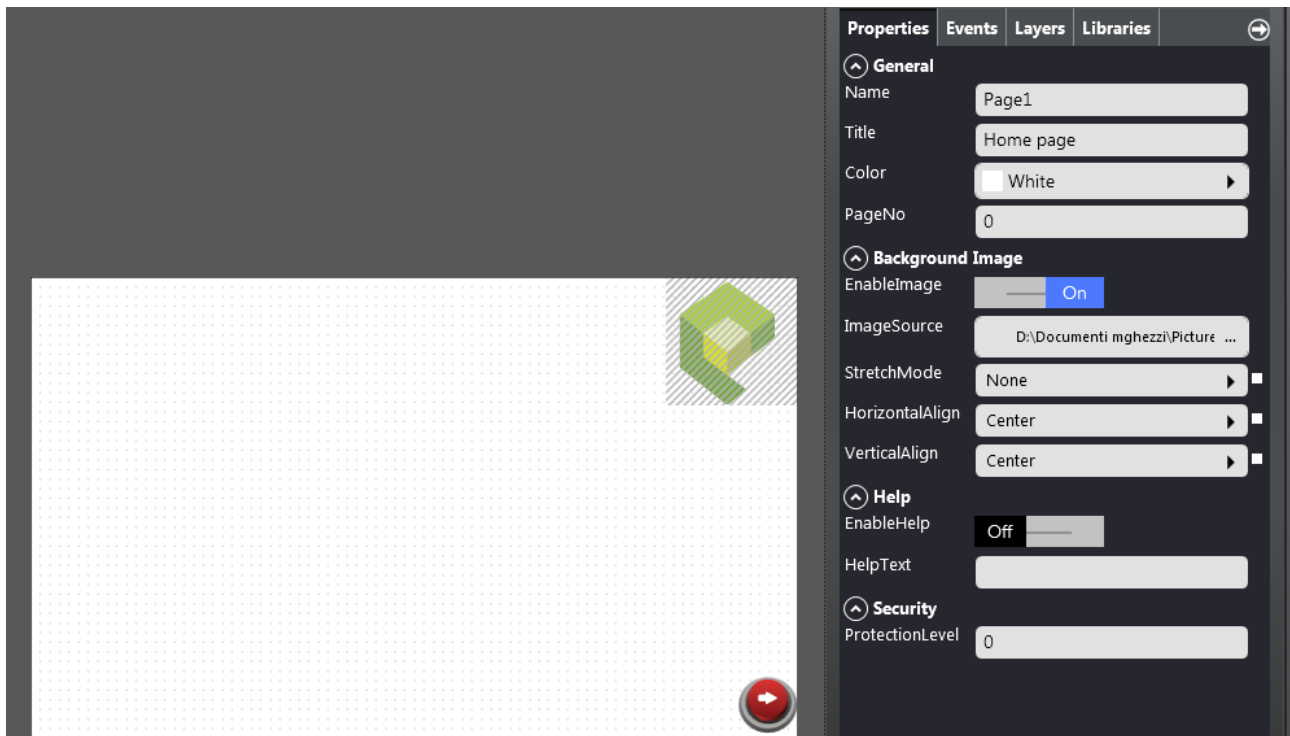
# CREW Manual

## Properties

The “Properties Editor” is comprised of a list of properties and their relative values in editable fields. The property fields of the single object vary according to the type of object entered on the page. Also, the page itself features a series of editable properties.

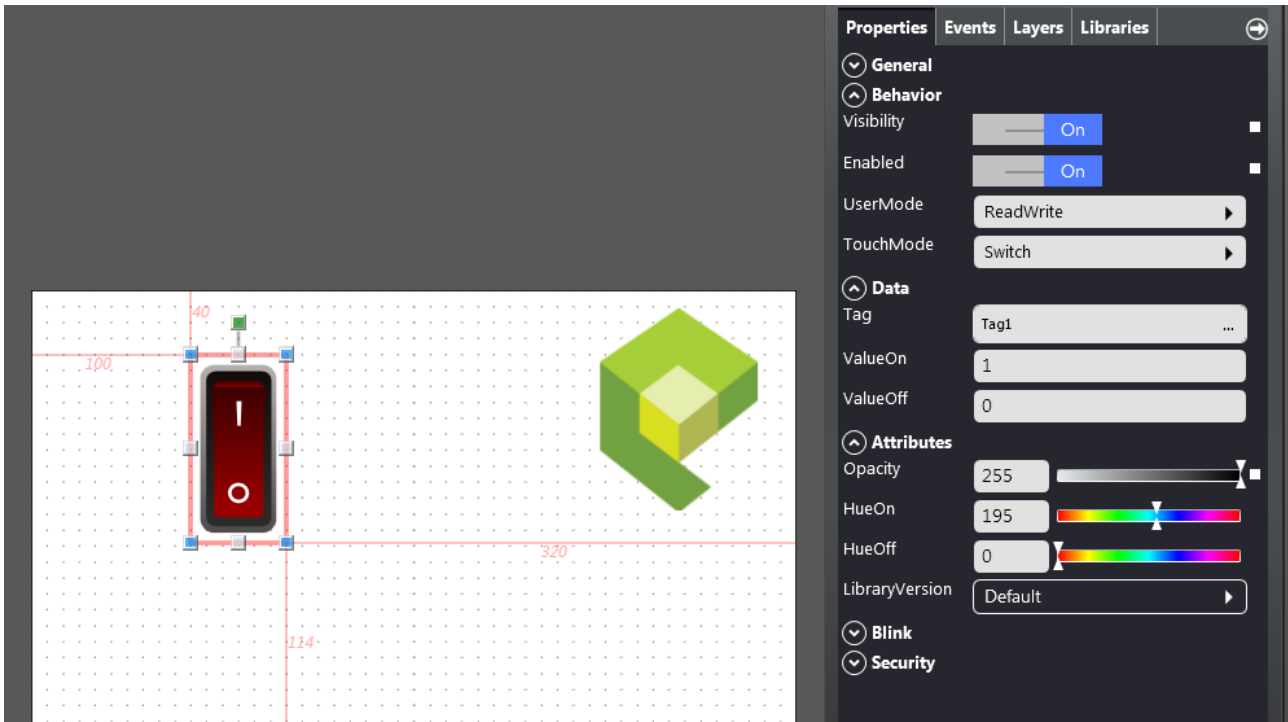
Below it is possible to look at an example relative to the properties of the page, and another referring to the properties of a switch.

### Page properties



# CREW Manual

## Switch properties

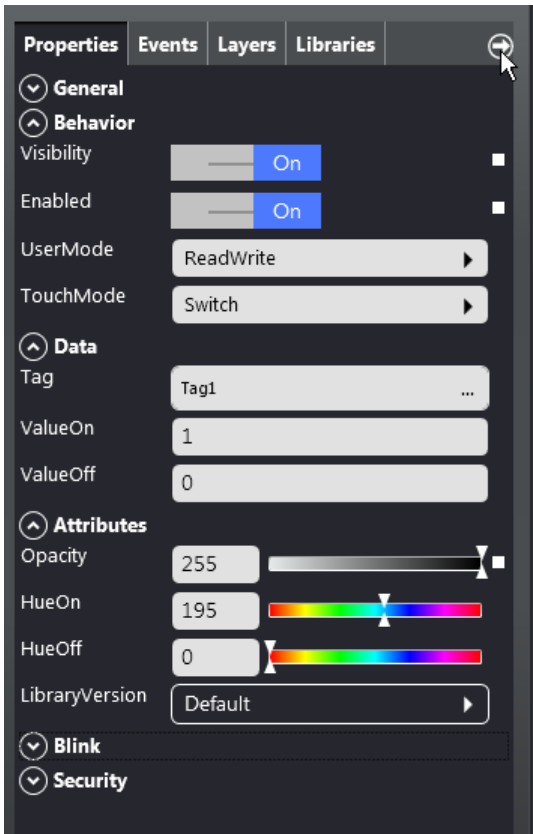


As you can see, the editable fields of the page differ from those of the switch. If it is not possible to edit the values of an element, this means that its current configuration does not allow it to be edited. In these cases it is only possible to enable editing for these fields if the correlated attributes allow for it.

The variations of the graphic properties of an object are displayed in real-time on the page so that the user can immediately see the changes.

# CREW Manual

If the “Event Editor” does not appear because previously closed, click the relative icon to open it back up again.



The following sections illustrate the editable properties for each object, and their meanings.

# CREW Manual

## Dynamic assignment of values to the properties

Some properties can be linked to a variable instead of entering a constant value. The value of the property can change in Runtime according to the changes to the associated variable.

To switch from constant value assignment mode to variable assignment mode, simply click on the red square on the right of the editable field. When you switch to variable assignment mode, it is possible to choose whether to associate a variable or whether to manage the value through the “Thresholds” function (refer to “[Thresholds Management Feature](#)”).

The associated variable type be consistent with the values required by the property:

- for True/False properties, the variable must assume Boolean values.
- for Date/Time properties, the variable must assume Long values.
- for colour properties (for example, BorderColor or AreaColor), the variable must assume valid RGB (Long) values as indicated in the table.

Colour	RGB	Hexadecimal value
Red	255,0,0	00 00 00 FF
Green	0,255,0	00 00 FF 00
Blue	0,0,255	00 FF 00 00

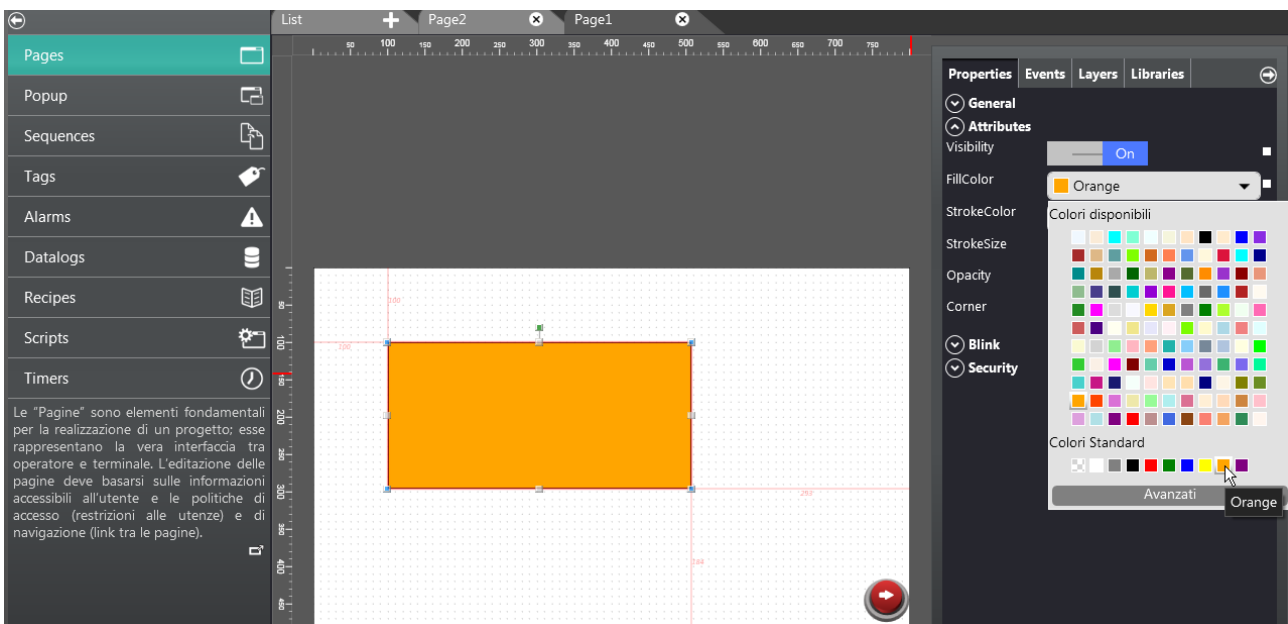
# CREW Manual

## Thresholds Management Feature

The "Thresholds" function in the Editor Properties can be used to manage "colour change", "blinking", "hide" and "disable" and other properties of the objects.

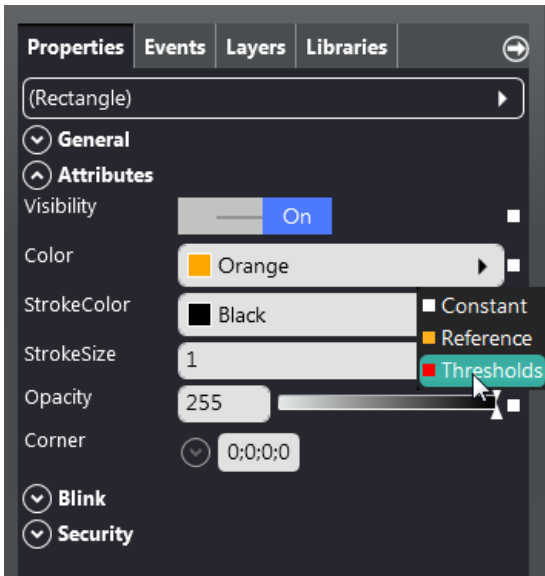
## Thresholds Management Operation

To explain how the "Thresholds" option works, the example of the "FillColor" property of a Rectangle is given, which has the colour orange attributed to it.

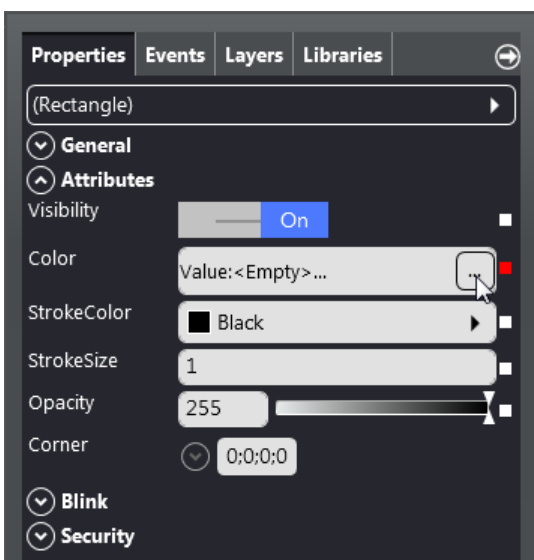


# CREW Manual

Click the white square on the right of the "FillColor" property and select the "Thresholds" option.



Then click "Browse" to attribute the variable and the values of the thresholds connected to the property that you wish to manage.

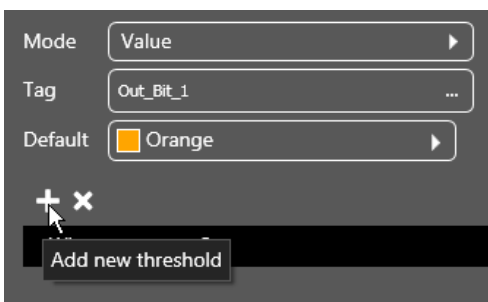


# CREW Manual

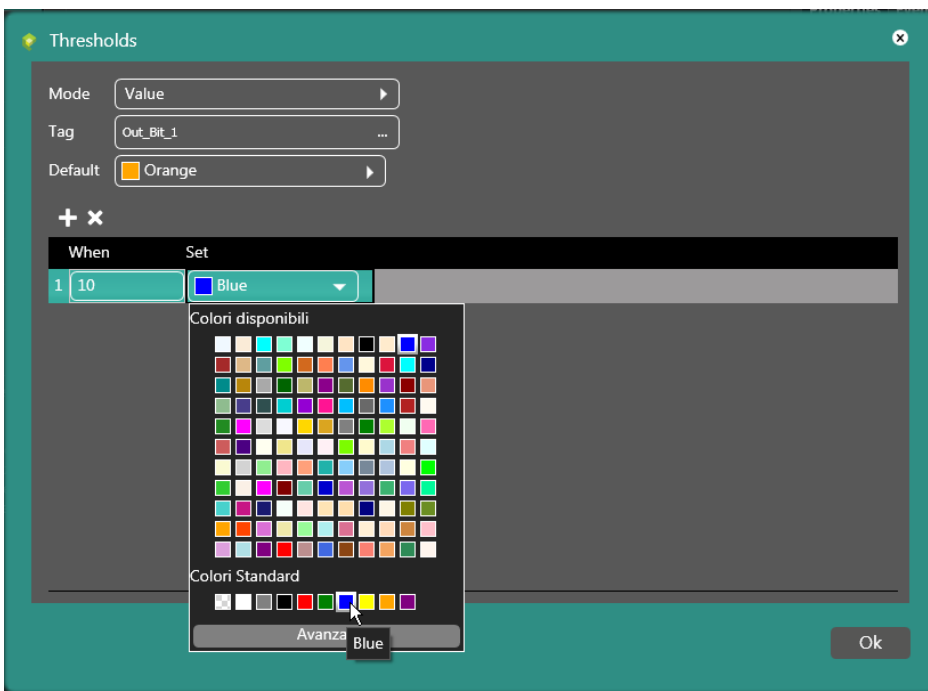
The window shown in the image will now appear, where it is possible to decide whether to manage the “FillColor” property in “Value” or in “Bits”. It is also necessary to associate the rectangle’s “FillColor” property to a “Tag”.

In the example we chose “Value” management, which allows the user to add as many values as he/she wishes without any limit, and the “Out\_Bit\_1” Tag has been associated.

Then click on the “+” icon to add a new threshold.



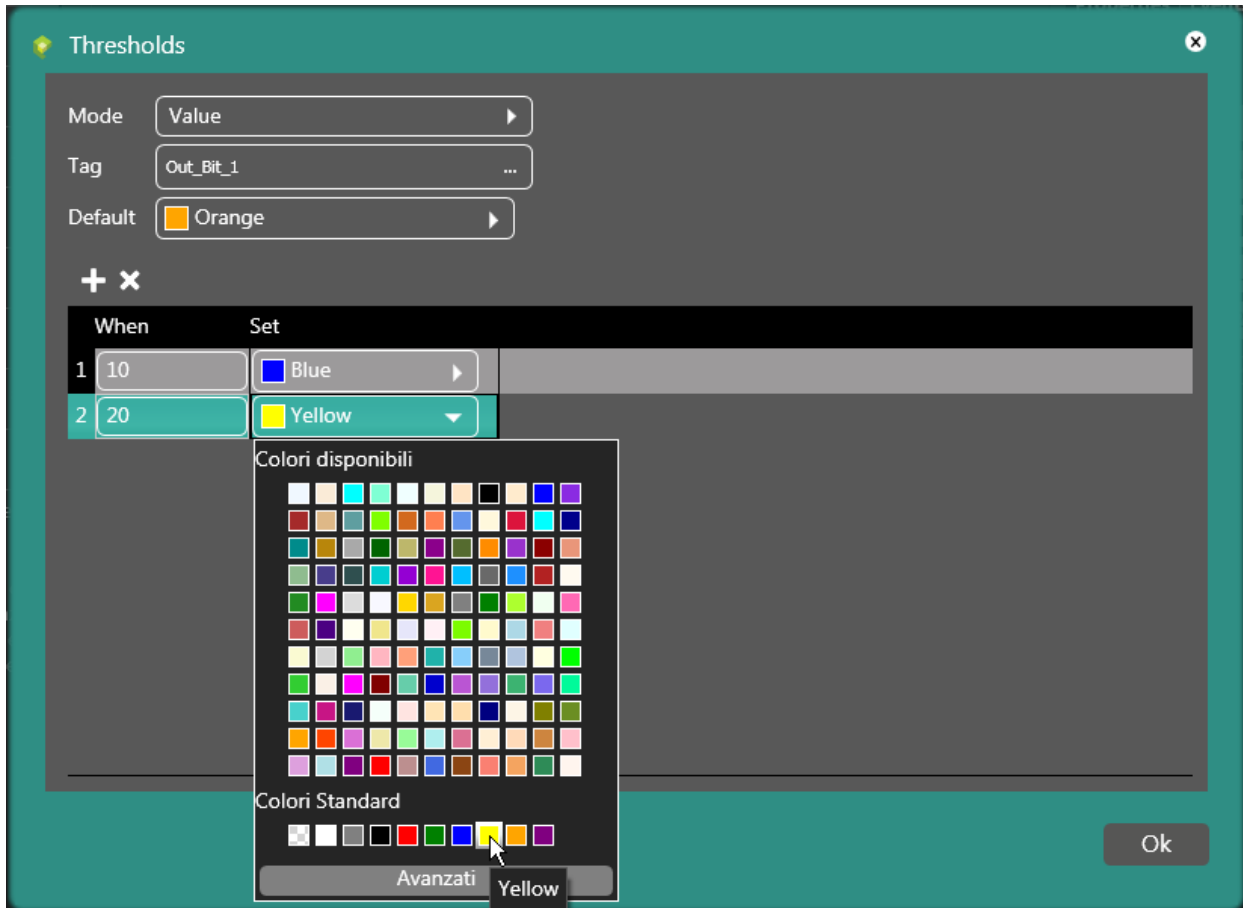
Enter the value "10" and link it to the colour blue.





# CREW Manual

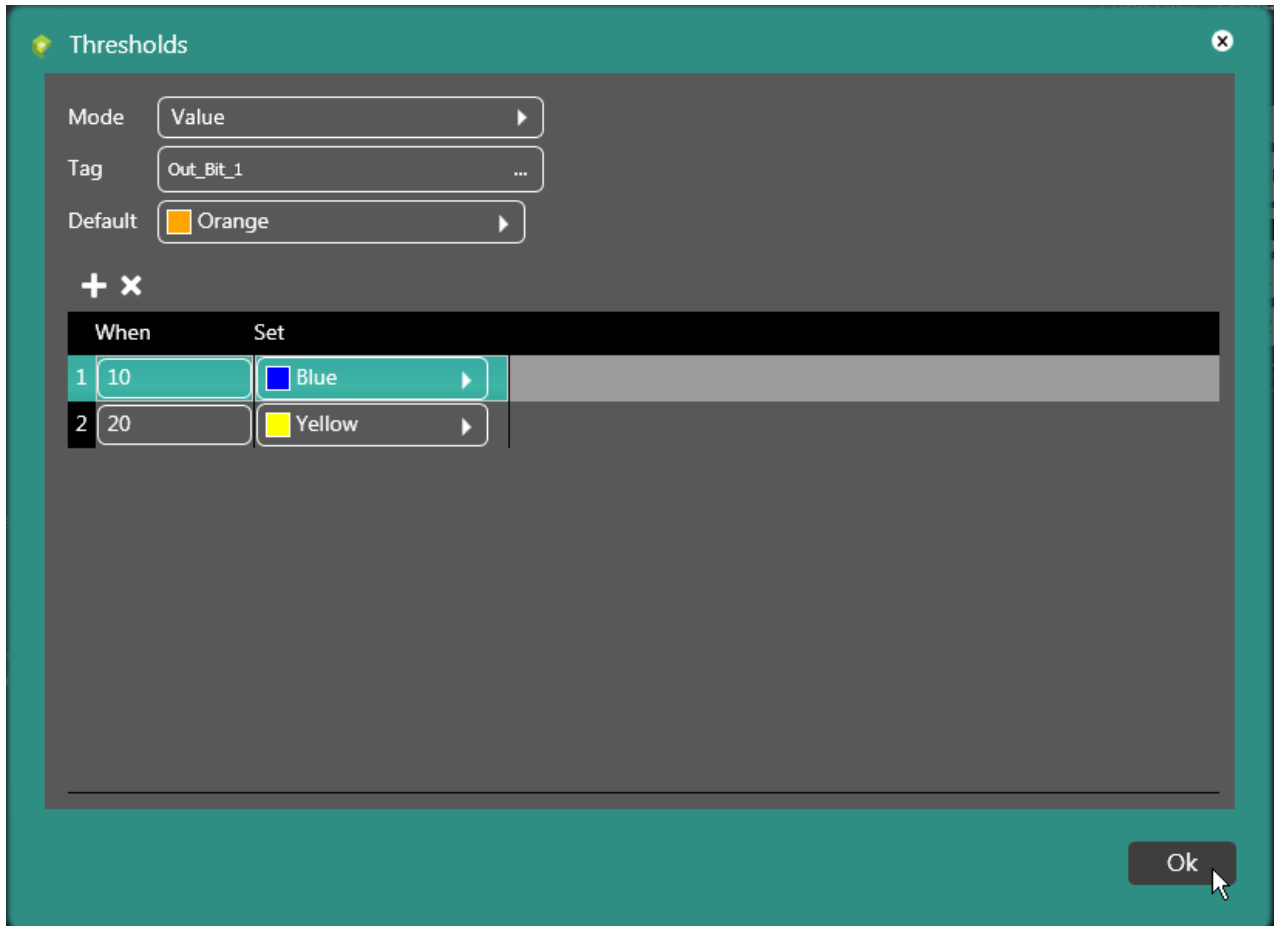
Enter the value "20" and link it to the colour yellow.



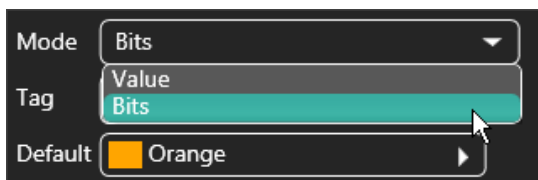
Accordingly, the colour of the rectangle will turn blue when the "Out\_Bit\_1" variable (in our example) has a value of "10", while the rectangle turns yellow when the variable has a value of "20".

# CREW Manual

Click "Ok" to confirm.

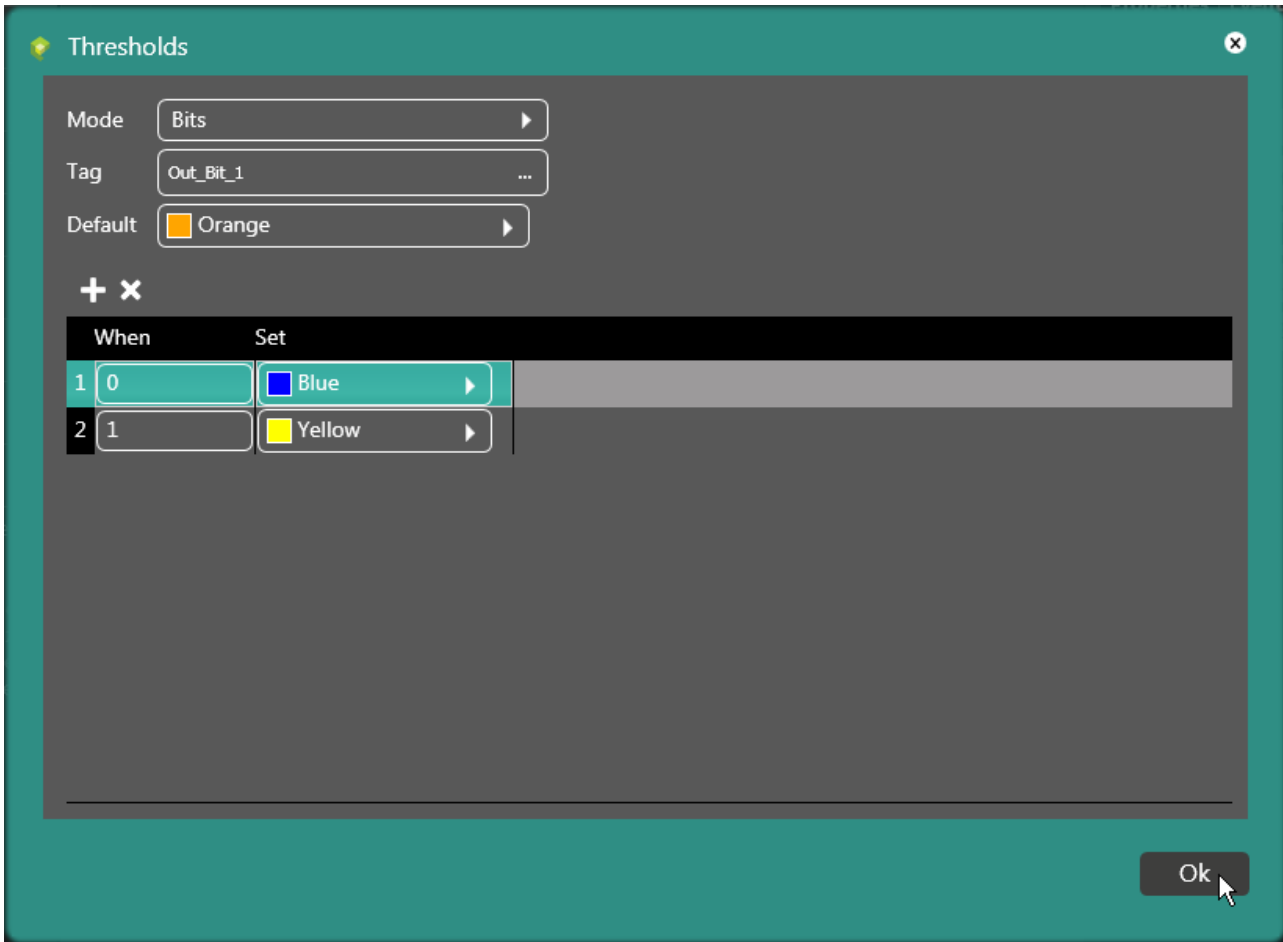


Otherwise, for "Bits" management



# CREW Manual

it is necessary to enter the same number as values as defined Bits to associate the necessary settings to.

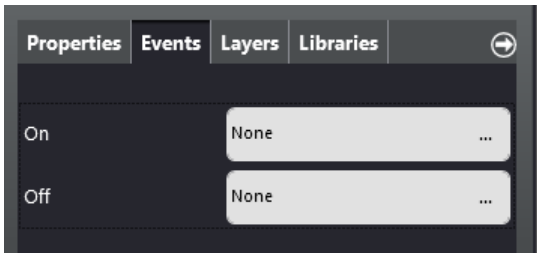


In this case it is possible to assign a different colour to each Bit. For example, the first Bit is on 1, the object is blue, when the second bit is on 1, the object is yellow, and so forth. If there are more bits on 1, the lowest one will be used. The "Bits" applied by the user may also be non contiguous. The least significant Bit must be Bit "1" while the most significant depends on the length of the associated type of Tag. For example, if the Tag is at 16 Bits, the user can enter Bits 0 to 15.

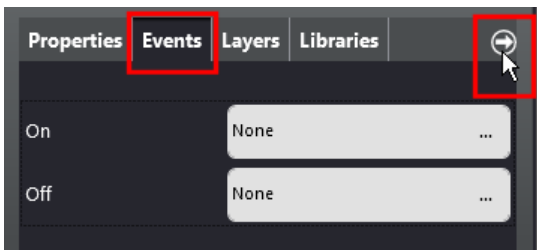
# CREW Manual

## Event Editor

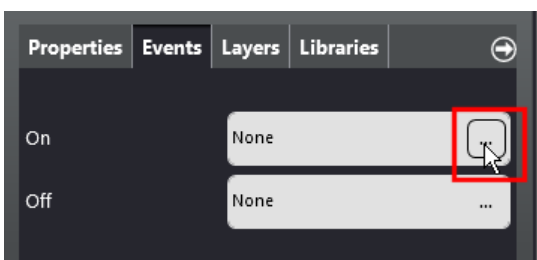
The "Event Editor" consists of a list of events linked to the element in question.



If the "Event Editor" does not appear because previously closed, click the relative icon to open it back up again.

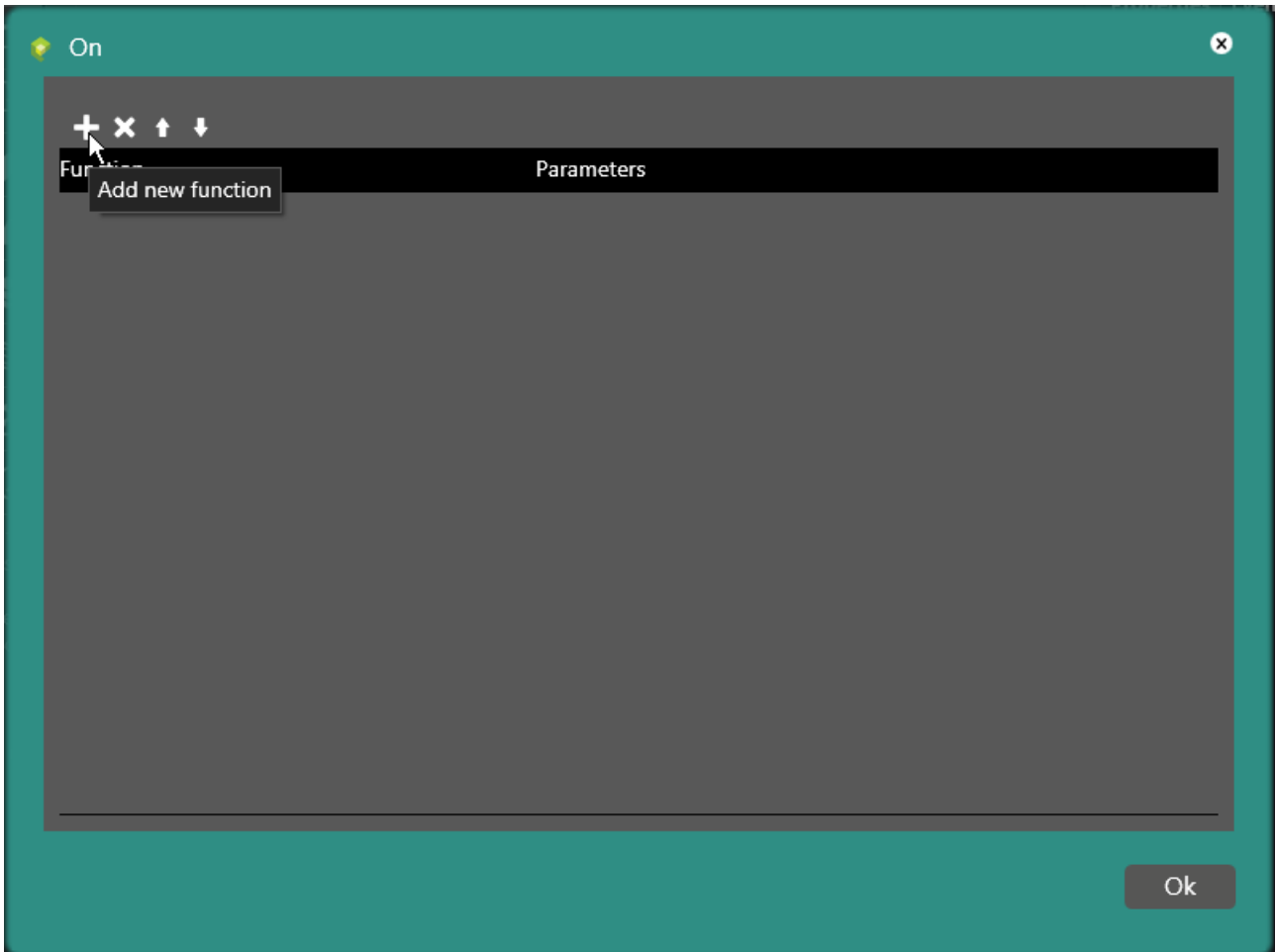


Use this window to link a predefined function (or a script, after having created it) to each event. To do so, simply click the relative key, as shown in the image.

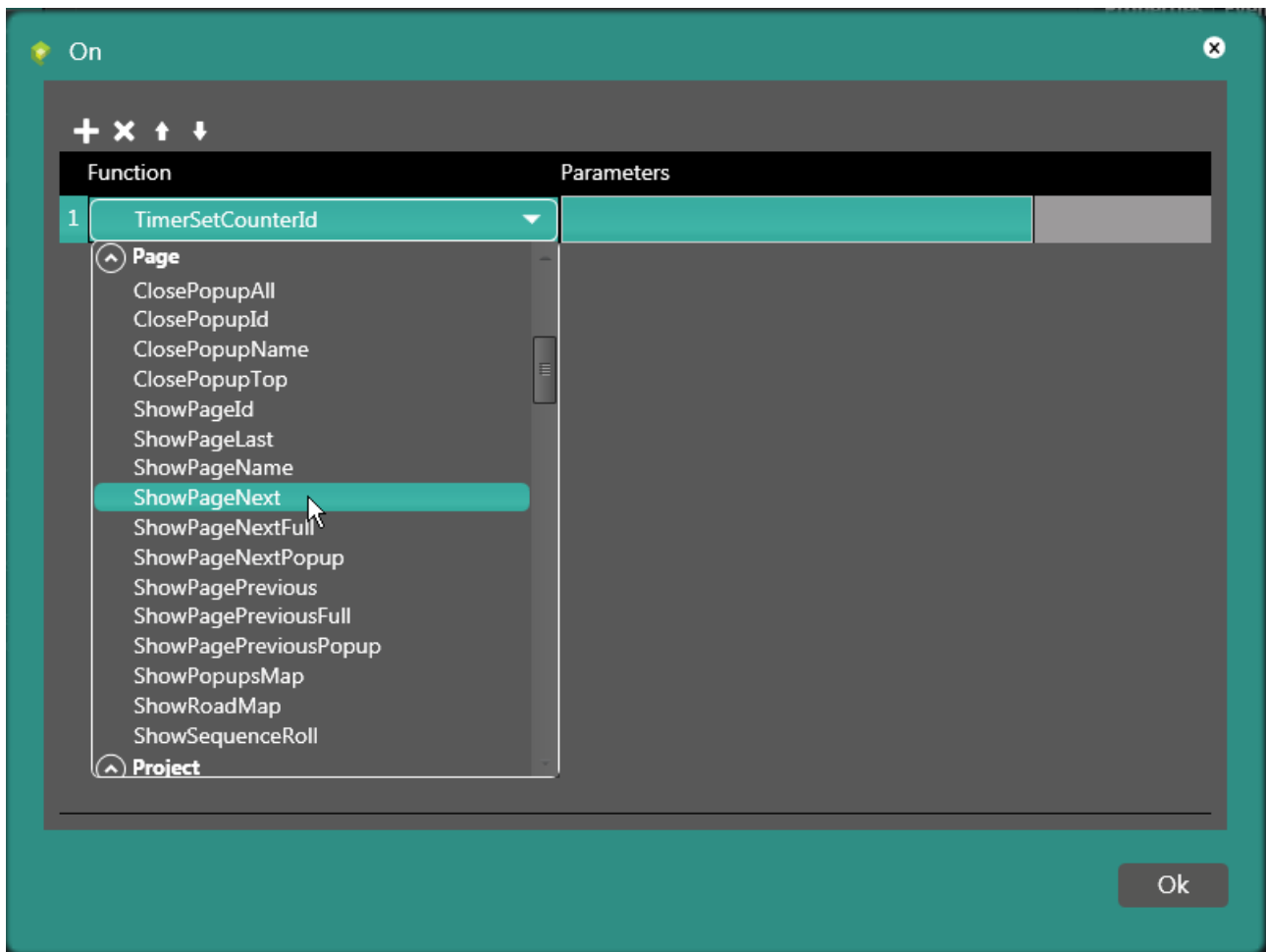


# CREW Manual

On the mask that appears, click the “+” icon to add one of the predefined functions available for the used object.



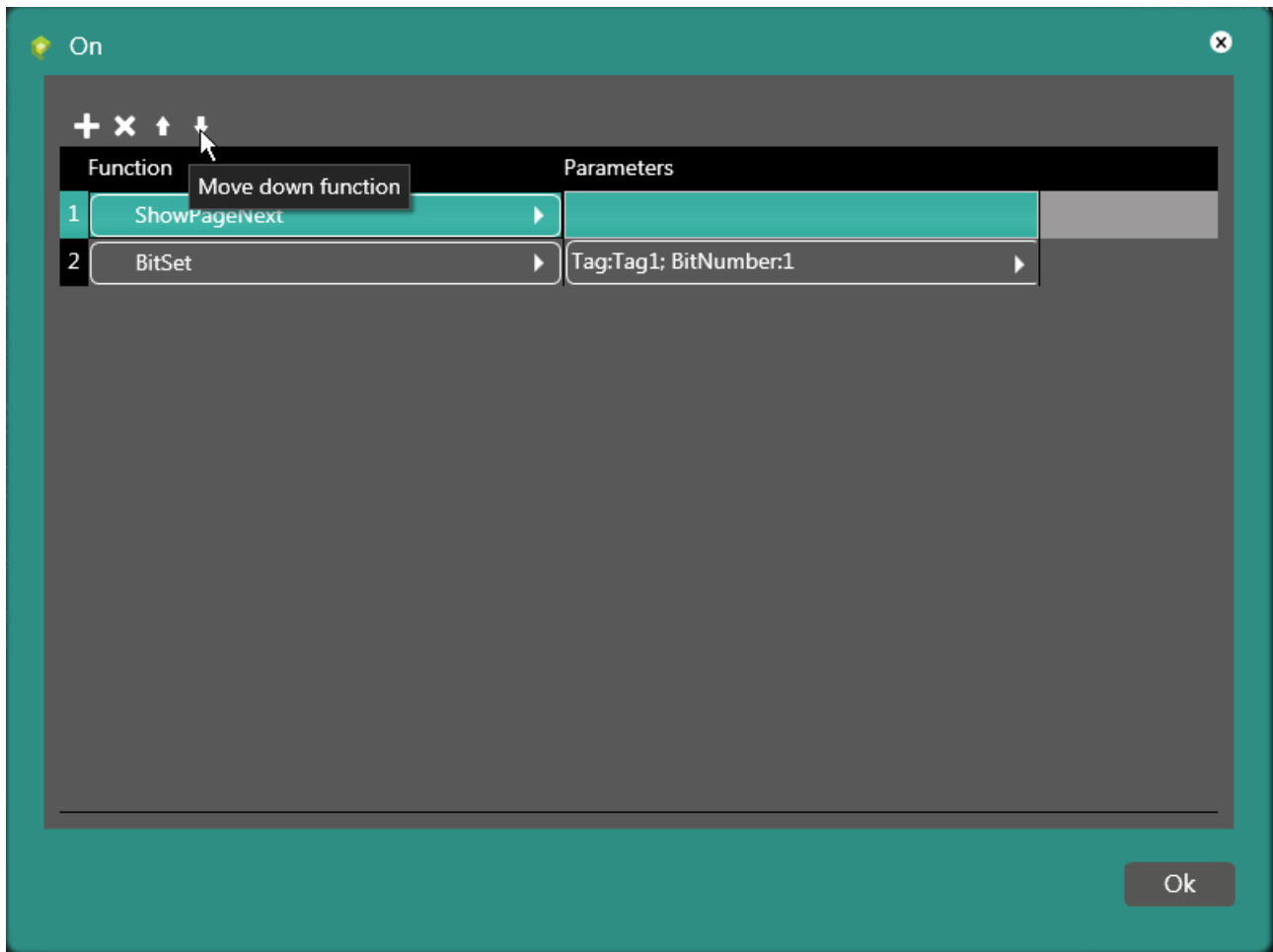
# CREW Manual



# CREW Manual

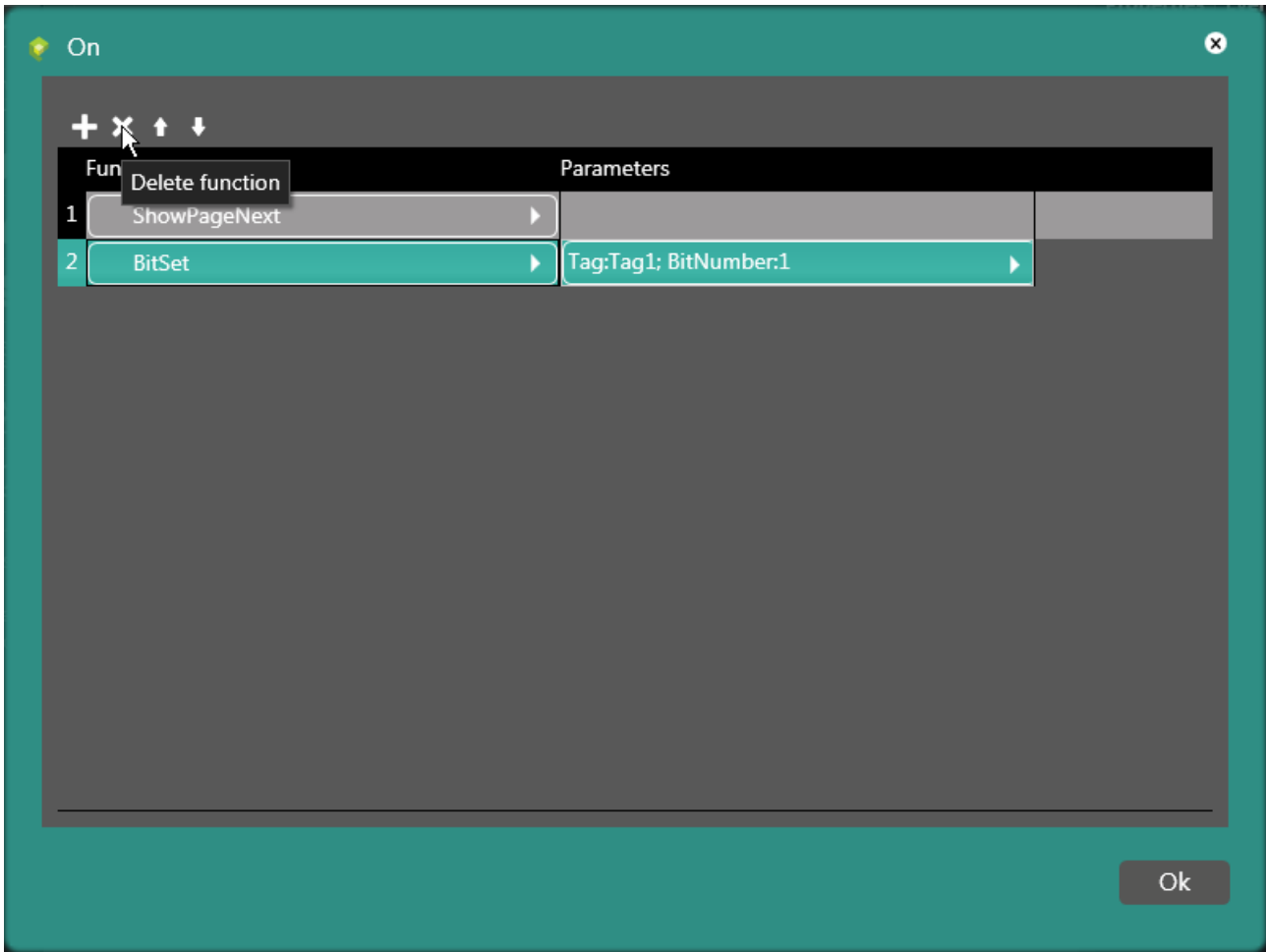
Choose and click “Ok” to confirm.

If multiple functions are associated to an event, it is possible to establish the order of execution (provided by the numbers). When a function has been selected, click the arrows to move it up or down.



# CREW Manual

To delete a function, select it and click "Delete function" ("X").



The following sections illustrate, for each object, the events that the functions and scripts can be associated to (see "[Predefined Functions](#)" and "[Scripts](#)").



# CREW Manual

## Page Events

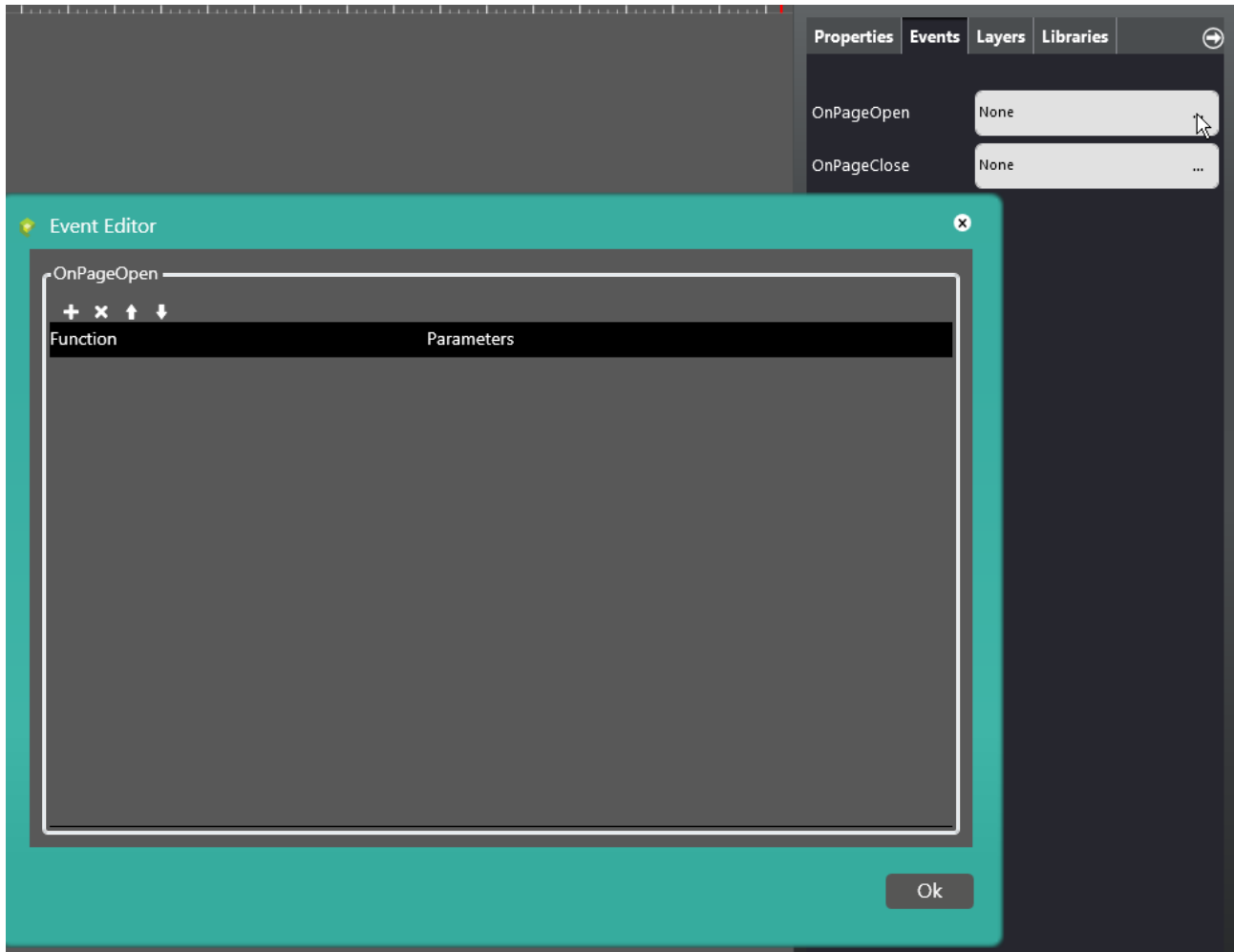
Event	Description
<b>OnPageOpen</b>	Enabled when a page is opened
<b>OnPageClose</b>	Enabled when a page is closed

An event (function or script) can be linked to each previously created Page. The event is activated based on whether the page is open or closed:

- **OnPageOpen:** when the page opens. For example, it is possible to bring up a "Message Box" when the page opens with the words "Page 1 Open".
- **OnPageClose:** when the page closes. For example, it is possible to bring up a "Message Box" when the page closes with the words "Page 1 Closed".

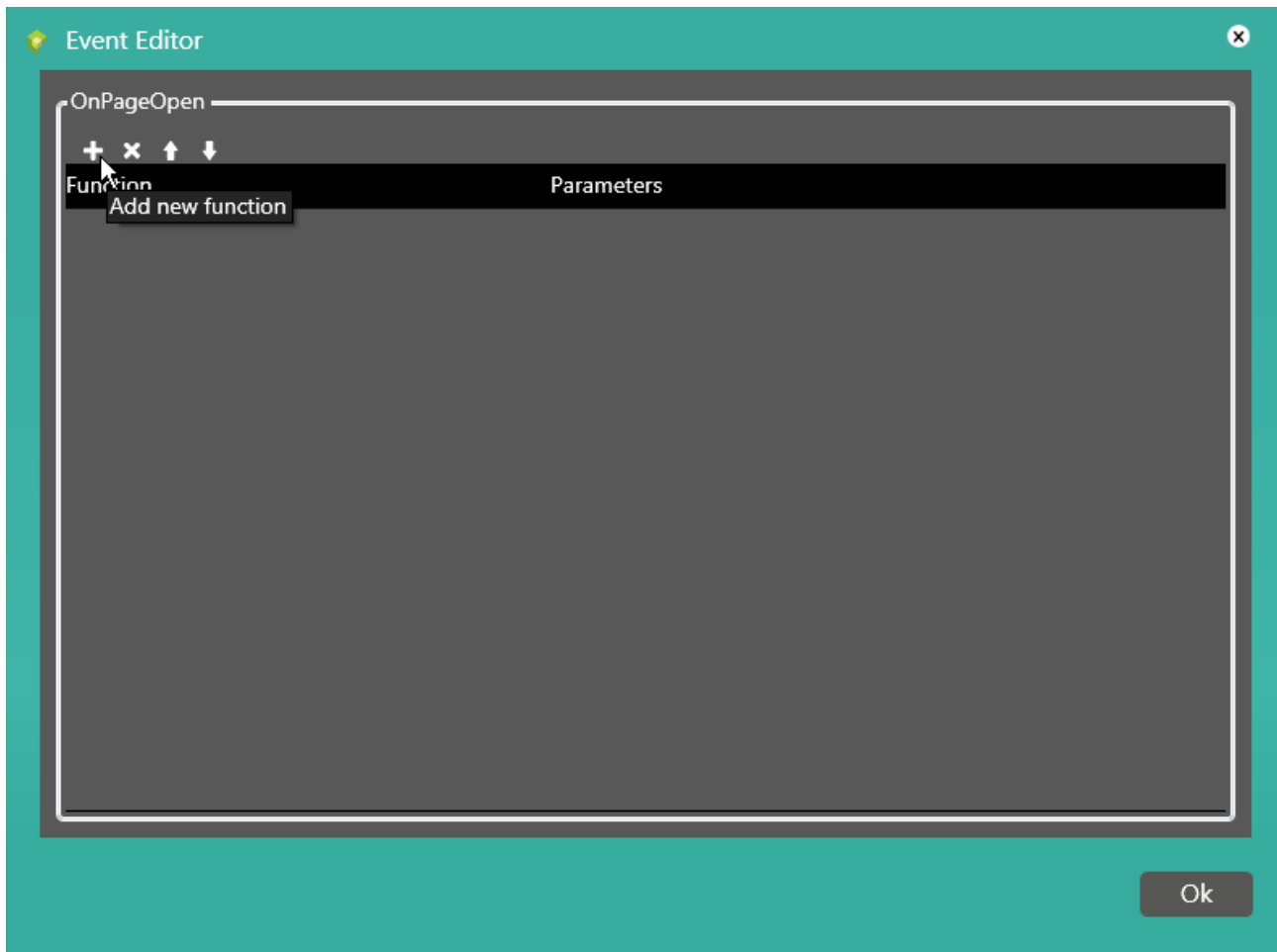
# CREW Manual

To assign an event to the page, select “OnPageOpen” or “OnPageClose” and click the browse key. The window illustrated in the image will now appear.

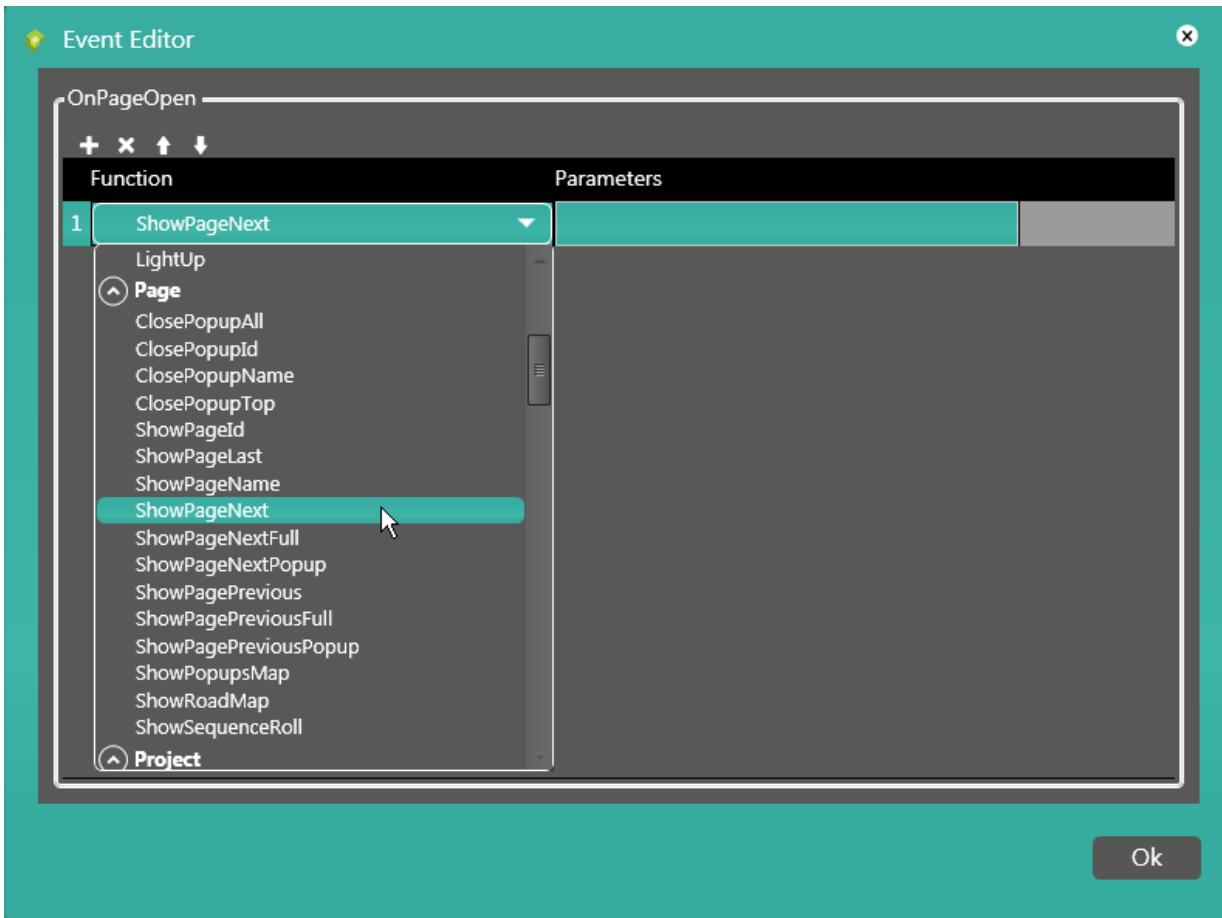


# CREW Manual

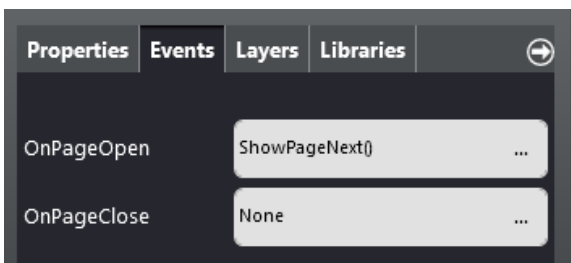
Click the “+” icon to add a function to be associated to the available ones.



# CREW Manual



Click “Ok” to associate the function. For example, if the function has been associated to “OnPageOpen”, it will be linked to the opening of the page in Event Editor.



# CREW Manual

## Pop-up page Events

Event	Description
OnPageOpen	Enabled when a page is opened
OnPageClose	Enabled when a page is closed

## Variable Events

Event	Description
OnValueChange	Enabled when the variable value has been edited
OnValueRead	Enabled when the Tag has been read from the device (PLC)
OnValueWrite	Enabled when the Tag has been written on the device (PLC)
OnValueSet	Enabled when the Tag value has been set by the Runtime
OnOnLine	Enabled when the Tag is no longer "offline" (once again accessible)
OnOffLine	Enabled when the Tag is "offline" (inaccessible)

# CREW Manual

## Alarm Events

Event	Description
<b>OnAnyAlarmOn</b>	Enabled when an alarm has been raised
<b>OnAnyAlarmOff</b>	Enabled when an alarm has ended
<b>OnAnyAlarmAck</b>	Enabled when an alarm has been acknowledged
<b>OnHistoryFull</b>	Enabled when the alarm log has reached its maximum capacity
<b>OnHistoryWarning</b>	Enabled when the memory capacity of the alarm log reaches the set warning percentage (by default the warning is triggered after 75% of the entire memory)
<b>OnAlarmOn</b>	Enabled when a specific alarm has been raised
<b>OnAlarmOff</b>	Enabled when a specific alarm has ended
<b>OnAlarmAck</b>	Enabled when a specific alarm has been acknowledged

# CREW Manual

## Datalog Events

Event	Description
<b>OnSamplesFull</b>	Enabled when the samples buffer has reached full capacity
<b>OnSamplesWarning</b>	Enabled when the samples buffer has reached warning level
<b>OnSamplesEnabled</b>	Enabled when the samples buffer has been enabled
<b>OnSamplesDisabled</b>	Enabled when the samples buffer has been disabled
<b>OnSamplesReset</b>	Enabled when the samples buffer has been reset
<b>OnSamplesStart</b>	Enabled when new samples logging has been started
<b>OnSamplesComplete</b>	Enabled when new samples logging has been completed
<b>OnSamplesSuccess</b>	Enabled when new samples have been logged successfully
<b>OnSamplesError</b>	Enabled when new samples have been logged with errors
<b>OnSamplesExportStart</b>	Enabled when new samples export has been started
<b>OnSamplesExportComplete</b>	Enabled when new samples export has been completed
<b>OnSamplesPrintStart</b>	Enabled when new print of samples buffer has been started
<b>OnSamplesPrintComplete</b>	Enabled when new print of samples buffer has been completed

# CREW Manual

## Recipe Events

Event	Description
<b>OnRecipeSaved</b>	Enabled when a recipe (from the buffer) has been saved in the archives
<b>OnRecipeLoaded</b>	Enabled when a recipe (from the archive) has been uploaded to the buffer
<b>OnRecipeDeleted</b>	Enabled when a recipe has been deleted from the archive
<b>OnRecipeRenamed</b>	Enabled when a recipe in the archive has been renamed
<b>OnDownloadStart</b>	Enabled when transference to the device has been started (from either the buffer or the archive)
<b>OnDownloadComplete</b>	Enabled upon download completion from the terminal to the device
<b>OnDownloadError</b>	Enabled when errors appear while downloading from the terminal to the device
<b>OnUploadStart</b>	A transfer from the device is started (directly for storage or archive)
<b>OnUploadComplete</b>	Transfer from the device has been successfully completed
<b>OnUploadError</b>	Transfer from the device has ended with errors



# CREW Manual

## Timer Events

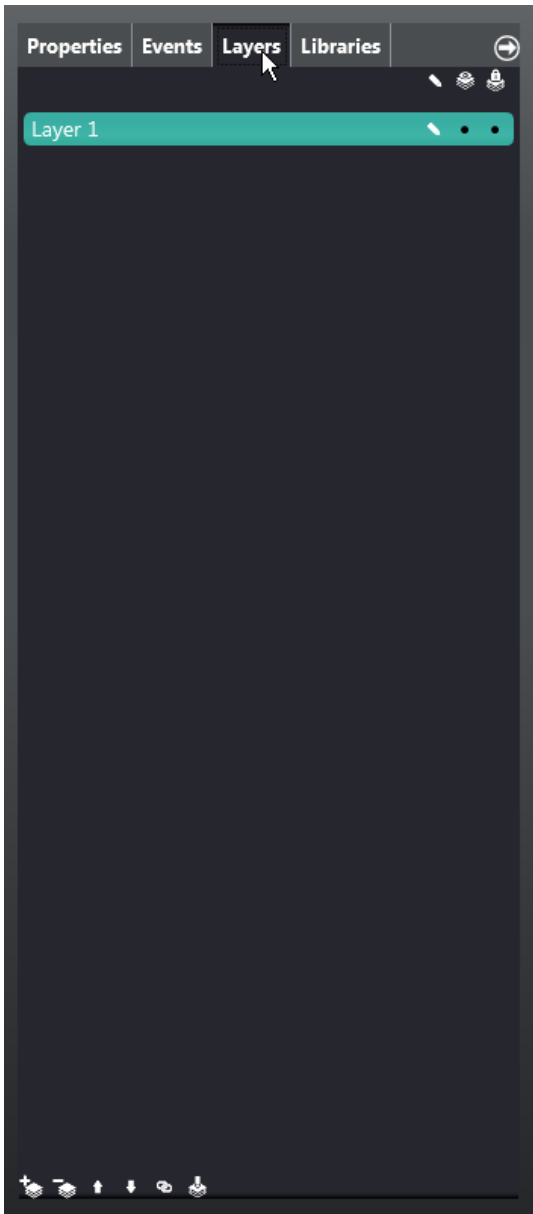
Event	Description
<b>OnTimerFired</b>	Enabled, upon completion, when timer countdown is started
<b>OnTimerStarted</b>	Enabled when timer countdown is started
<b>OnTimerStopped</b>	Enabled following the stop command impartito al timer
<b>OnTimerSuspended</b>	Enabled when the timer is suspended by command

## Layers

In Crew, the individually displayed pages are divided into layers. In the beginning, all of the objects are placed on layer 1, which is provided by default and can never be deleted.

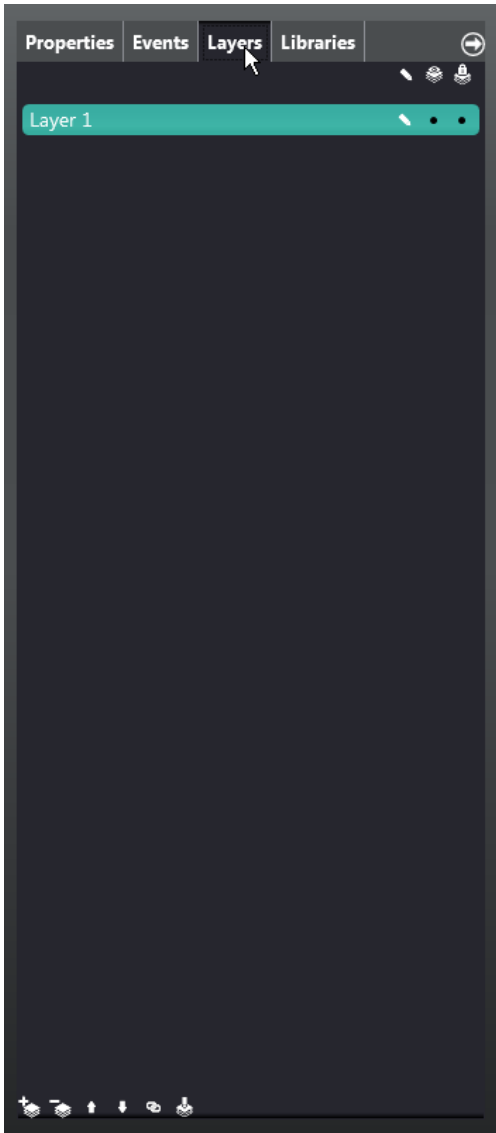
# CREW Manual

To access the Layers function, from “Property Editor” click on “Layers”.



# CREW Manual

## Operations on Layers



The bottom of the window contains the icons shown in the image:



Each icon makes it possible to run an operation (from left to right):

# CREW Manual

- Add one or more layers.
- Delete the selected layer.
- Move the selected Layer up one level (one click per level).
- Move the selected Layer down one level (one click per level).
- Join multiple layers.

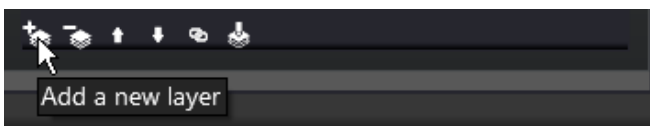
The top of the window contains the icons shown in this image.



Each icon makes it possible to run an operation (from left to right):

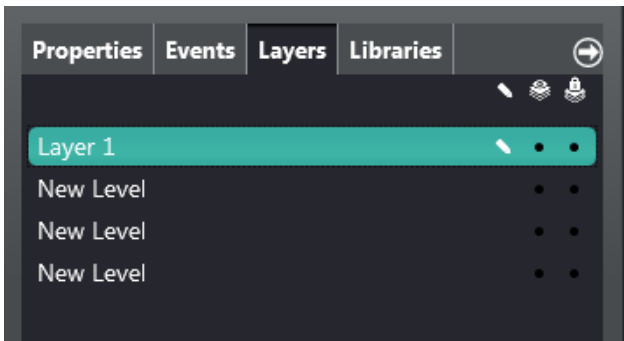
- Select a Layer
- Show-Hide Layers
- Block-Unlock Layers

## Add one or more Layers

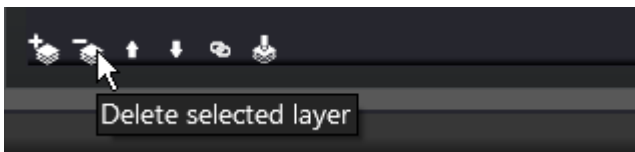


# CREW Manual

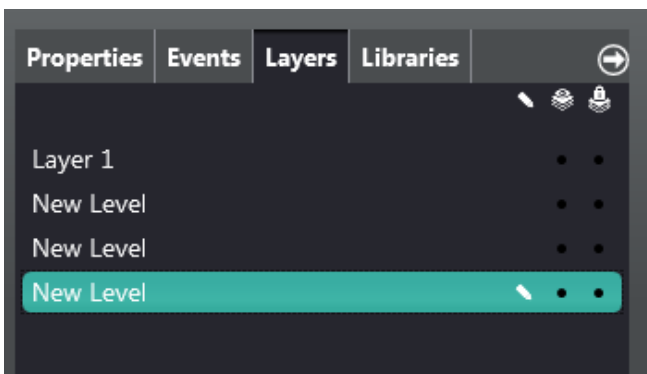
Click the “Add a new layer” icon every time you wish to add a new layer. For example, click the icon three times to add three new layers to the default one (four in total).



Delete the selected Layer

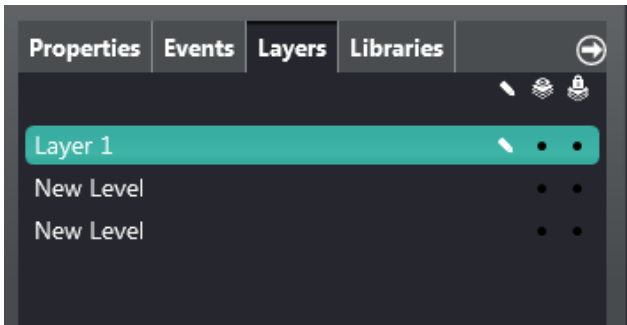


Select the layer you wish to delete



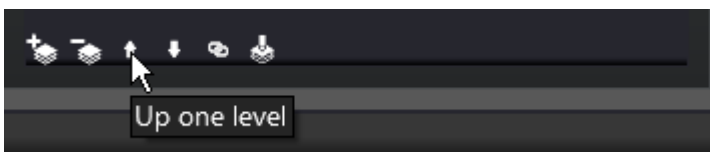
# CREW Manual

and click on the “Delete selected layer” icon.

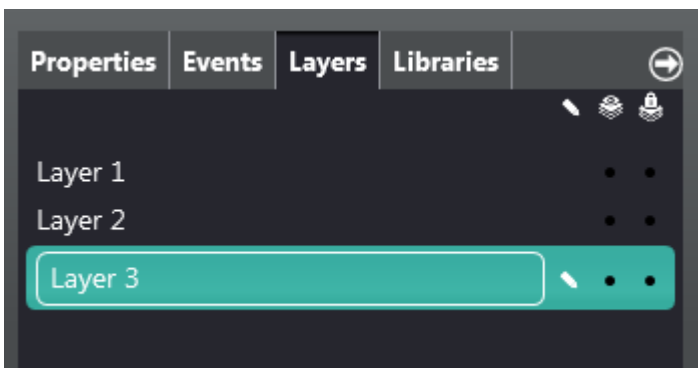


Move the selected Layer up one level

The “Up one level” icon (the arrow pointing up) moves the selected layer up every time it is clicked.

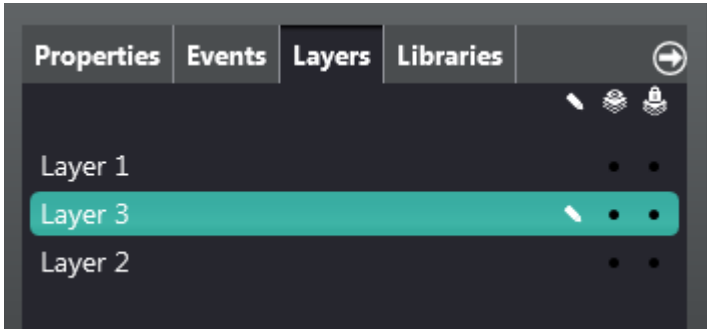


Rename the two remaining layers (double click the layer, enter the name and confirm with “Enter”) in Layer 2 and Layer 3.

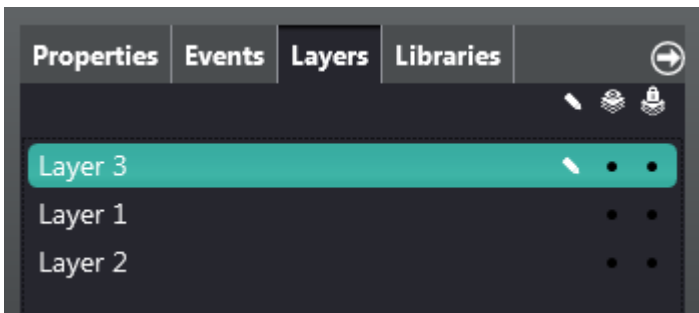


# CREW Manual

Press the “Up one level” icon once to make Layer 3 go up one position.

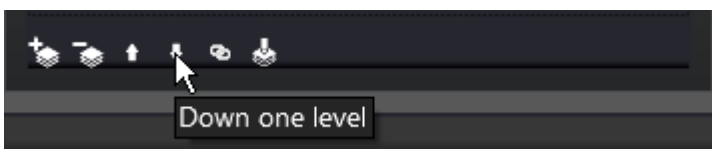


Press the icon again to make Layer 3 go up one more position.



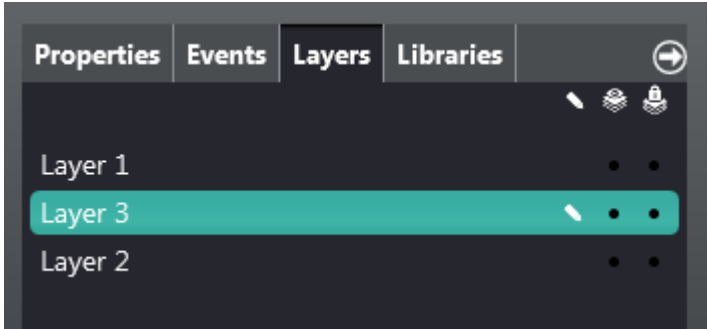
Move the selected Layer down one level

The “Down one level” icon (the arrow pointing down) moves the selected layer down every time it is clicked.

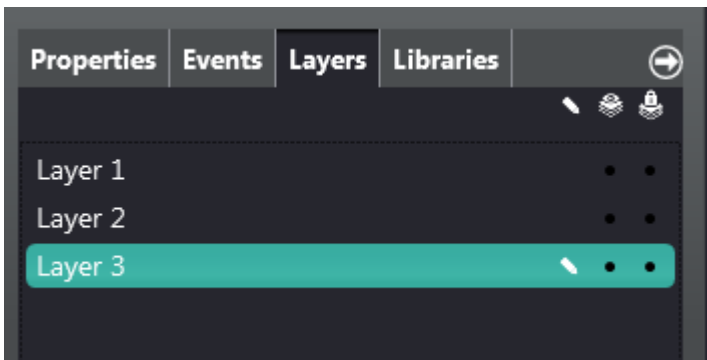


# CREW Manual

Select Layer 3 and press the “Down one level” icon once to make it go down one position.



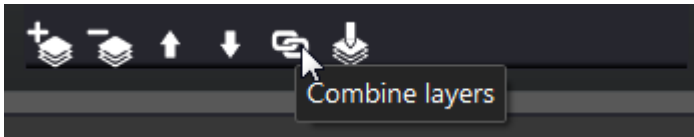
Press the icon again to make Layer 3 go down one more position.



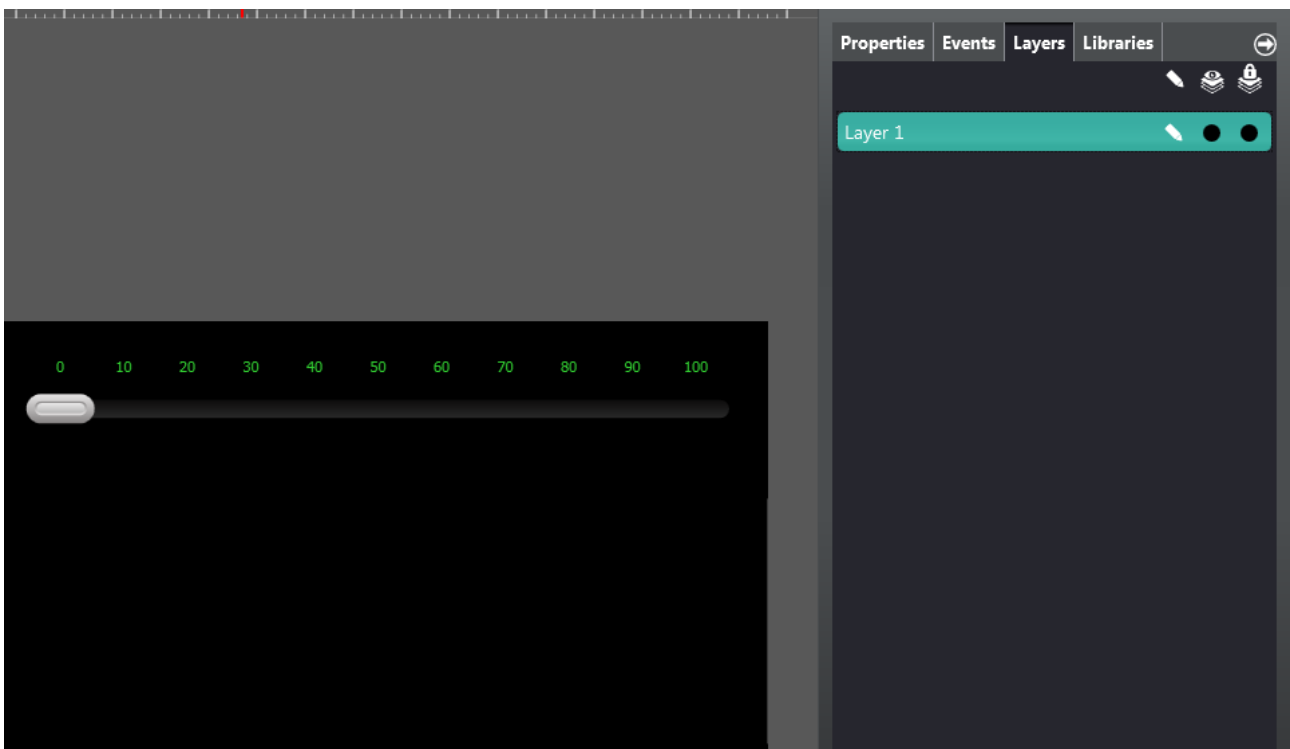


# CREW Manual

## Merge Layers

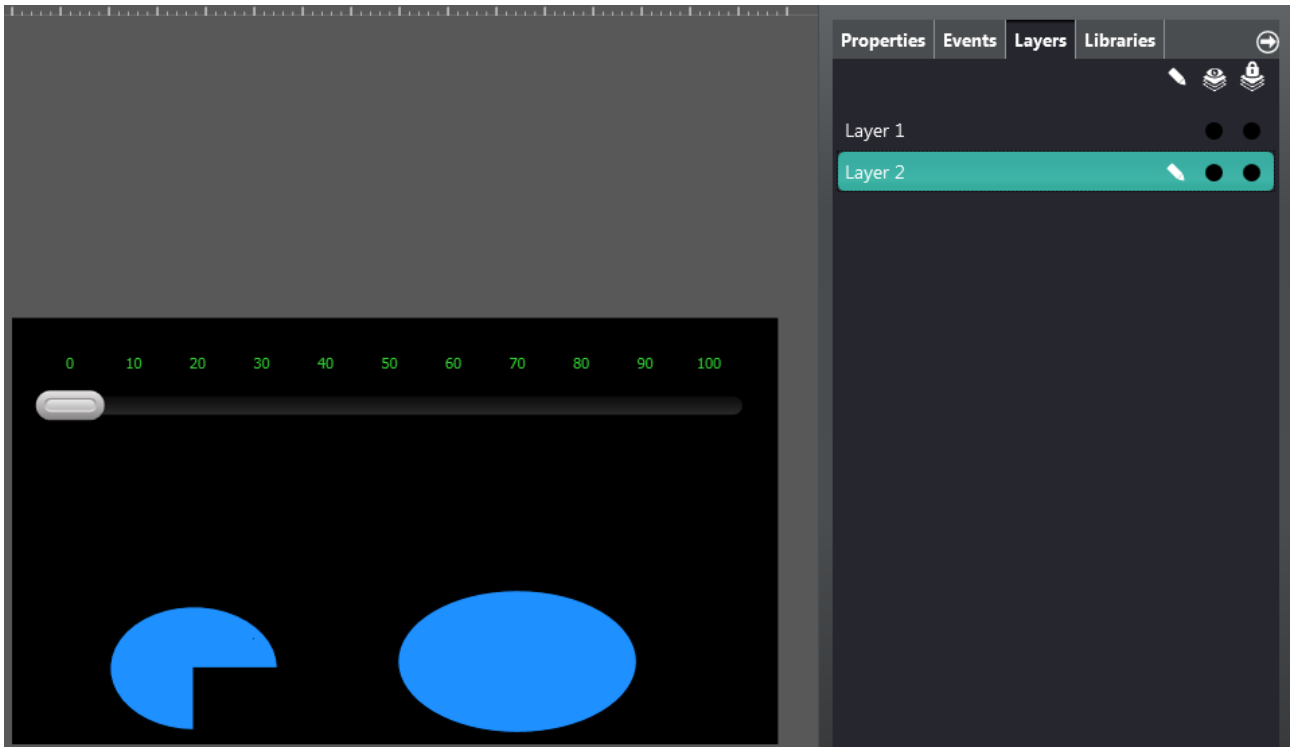


The “Merge Layers” key merges the selected Layer with the next one. For example, enter a selector in Layer 1,



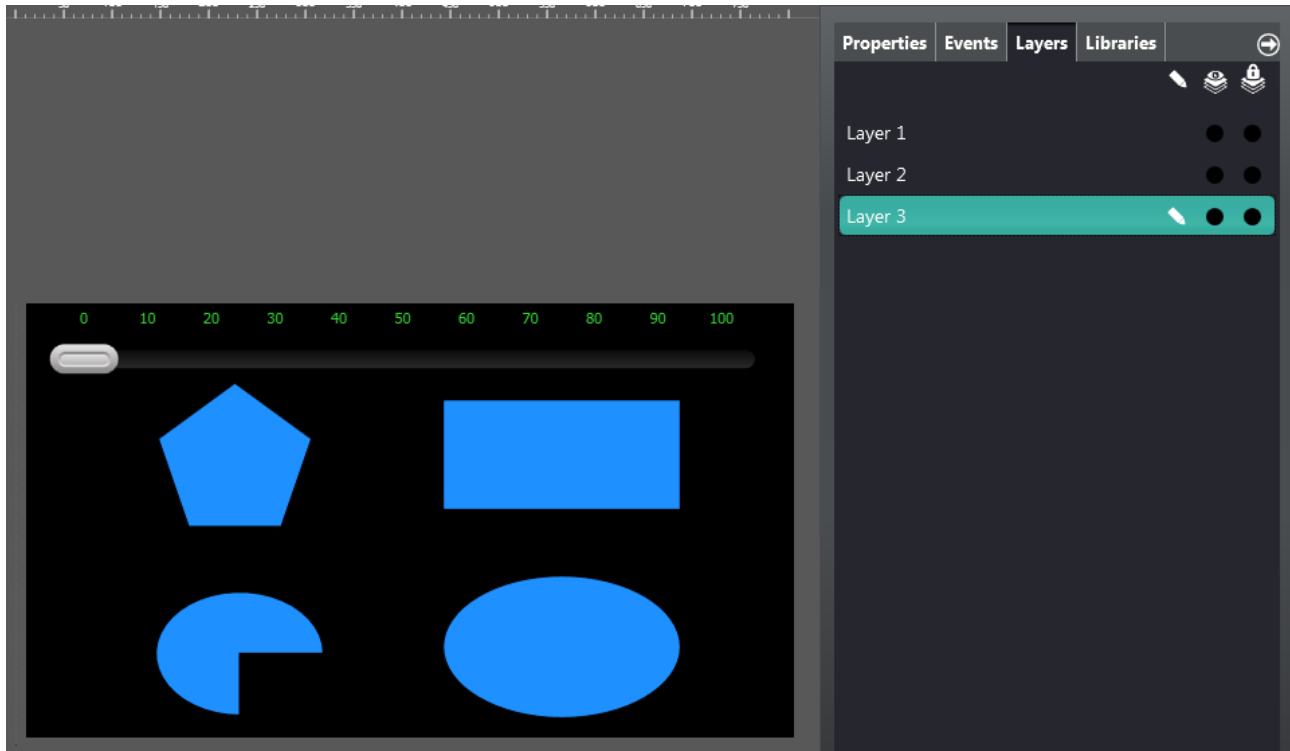
# CREW Manual

a circular sector and an ellipse in Layer 2,



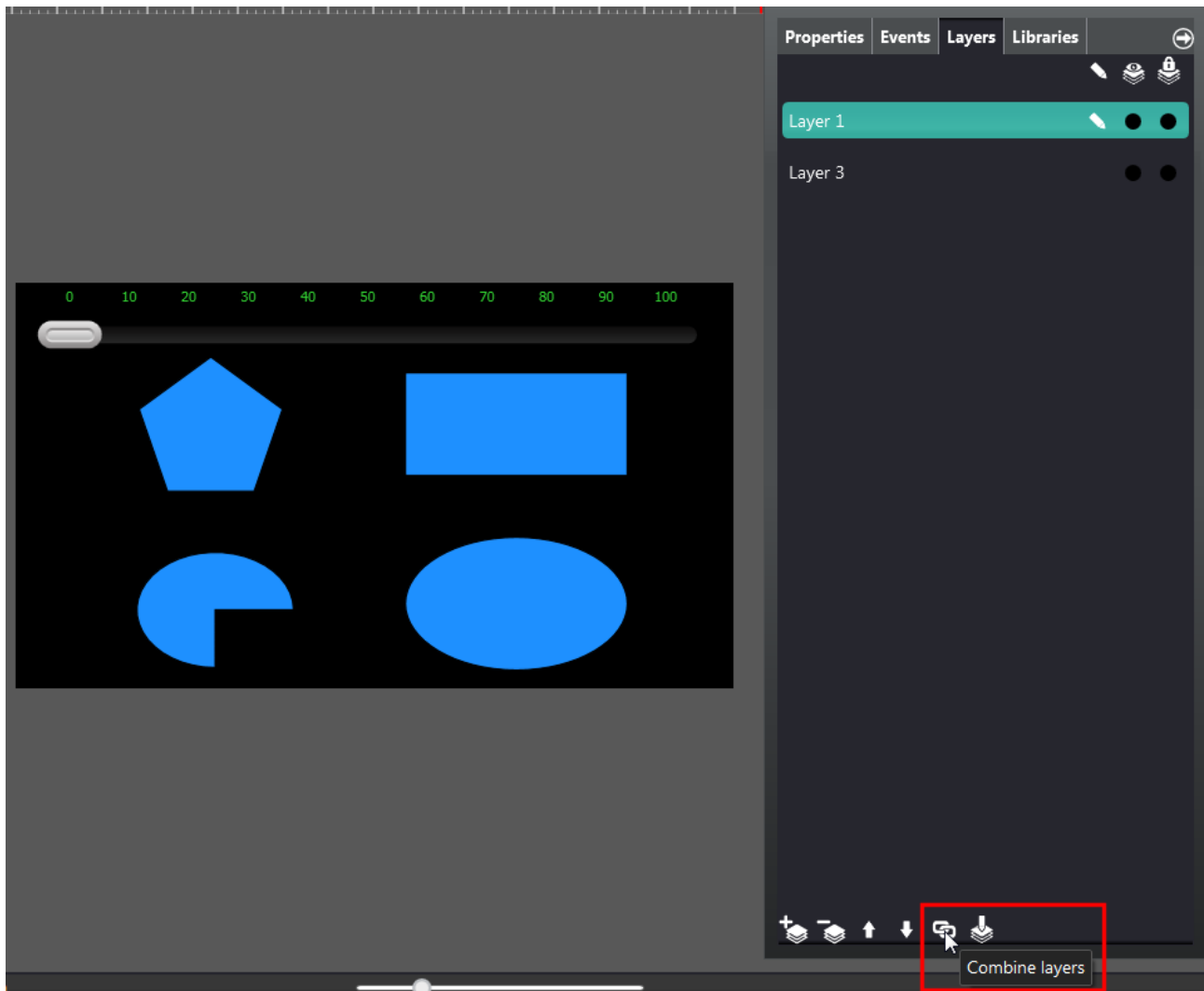
# CREW Manual

and a regular polygon and a rectangle in Layer 3.



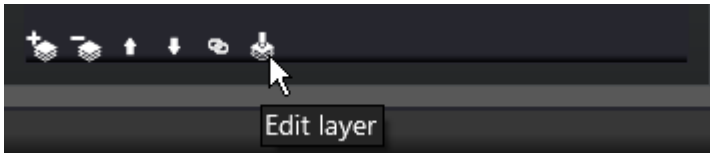
# CREW Manual

Select layer 1 and click on the “Merge Layers” icon, to merge Layer 1 and Layer 2.

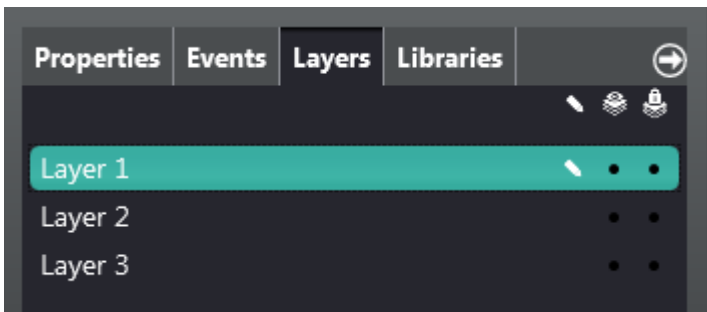


# CREW Manual

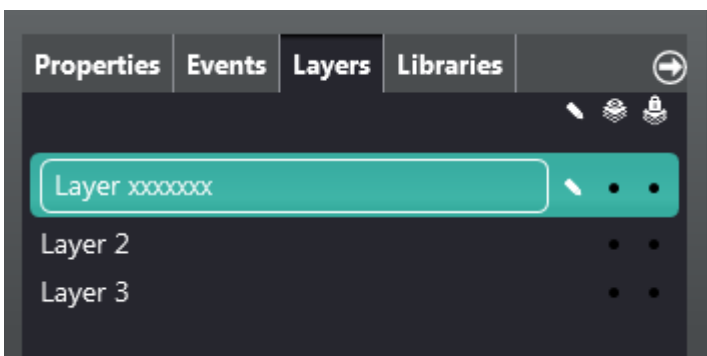
## Edit a Layer



Select the Layer that you wish to change and click the “Edit Layer” icon.



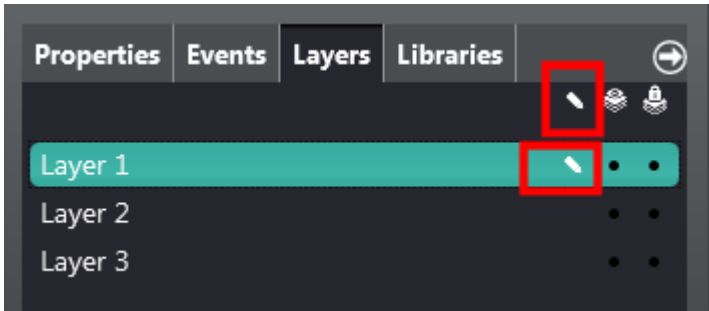
It is now possible to proceed as shown in the image.



Confirm by clicking "Enter" to apply changes.

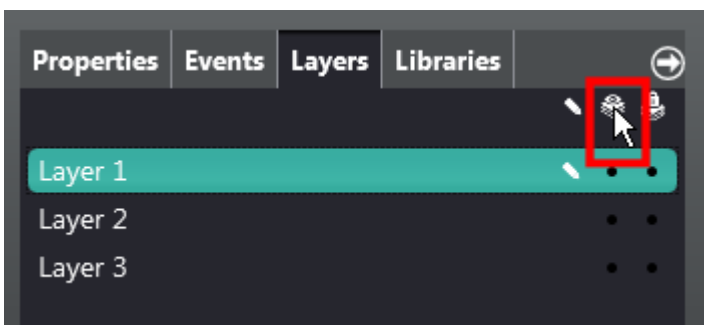
# CREW Manual

## Select a Layer



The "Select a Layer" option is automatically enabled when a Layer is selected with the mouse.

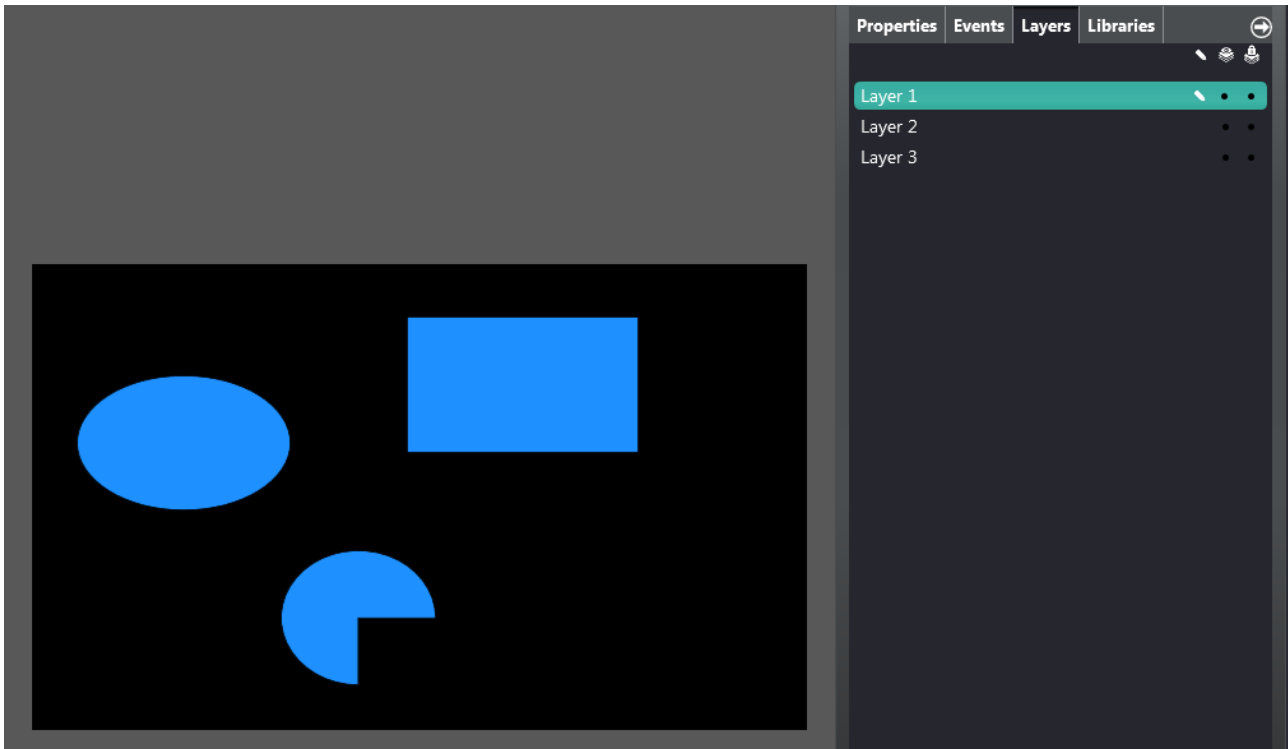
## Show-Hide Layers



The "Show - Hide Layers" icon is used to hide all of the created layers.

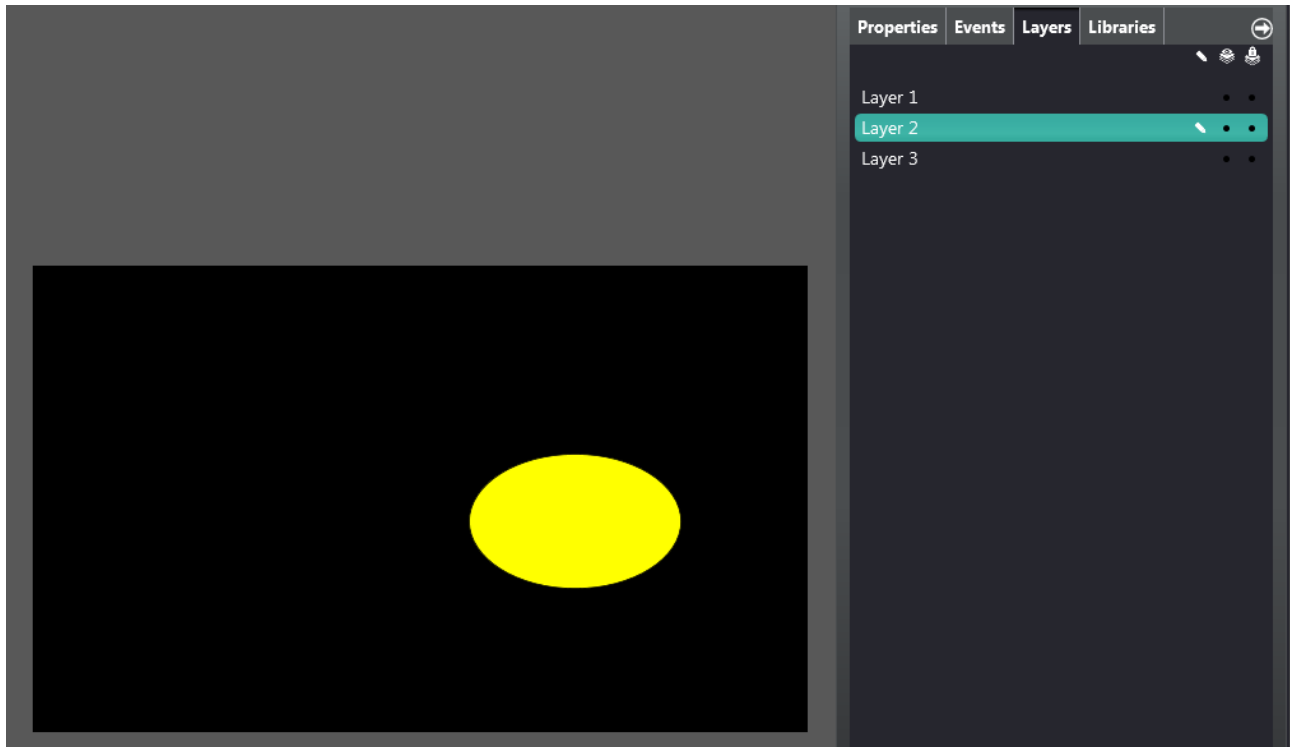
# CREW Manual

For example, with three different layers where: the first contains an ellipse, a rectangle and a circular sector;



# CREW Manual

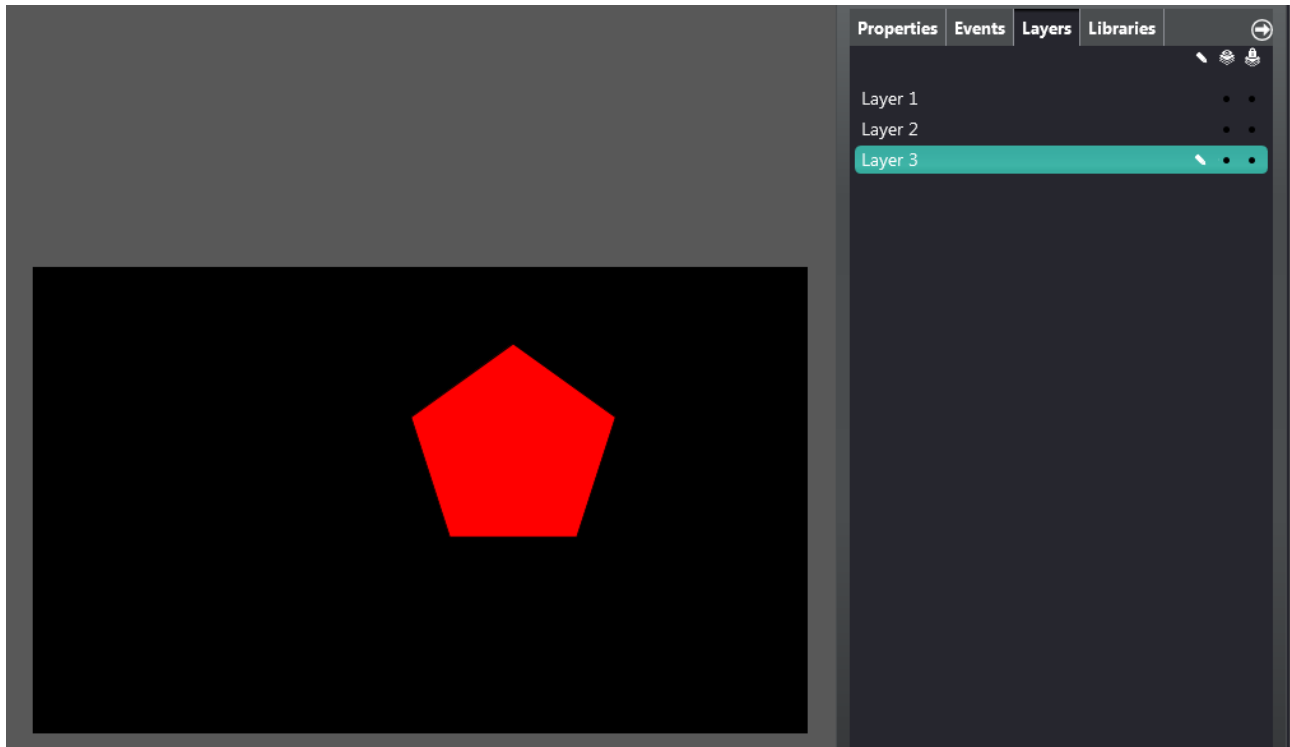
the second has a yellow ellipse;





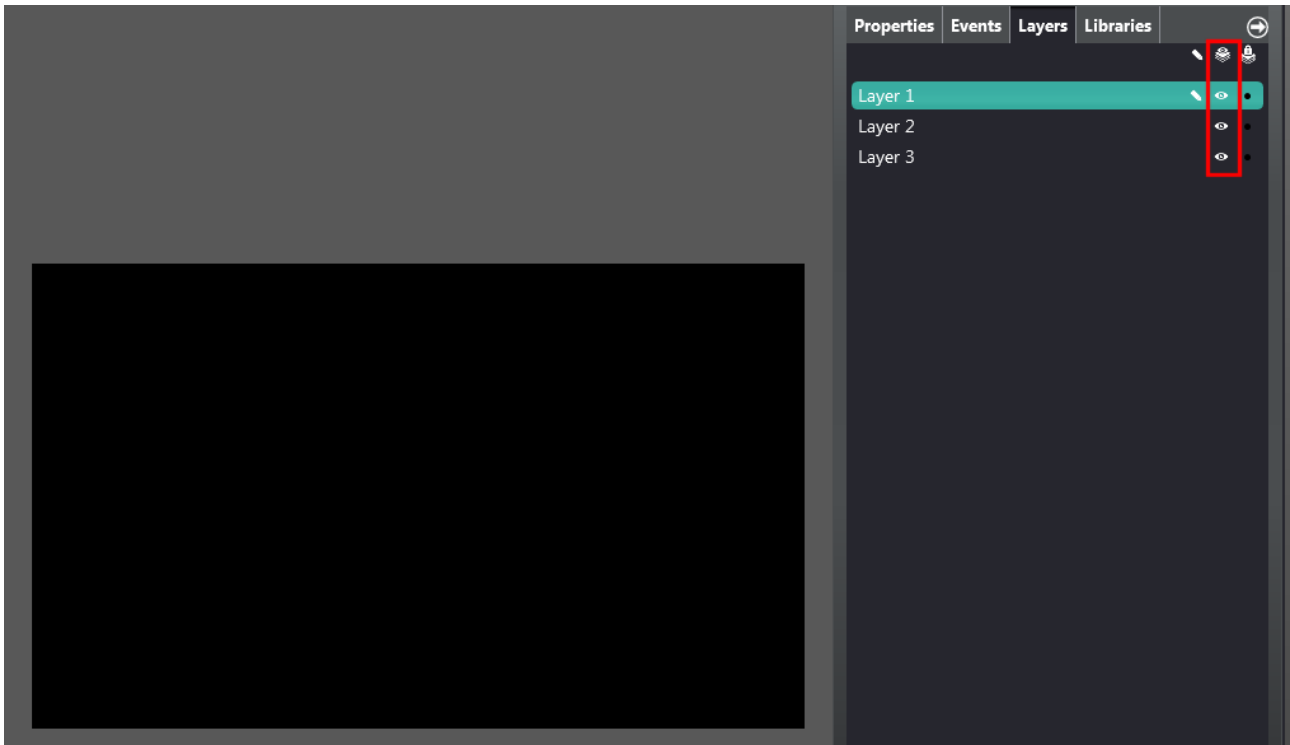
# CREW Manual

the third has a red polygon.



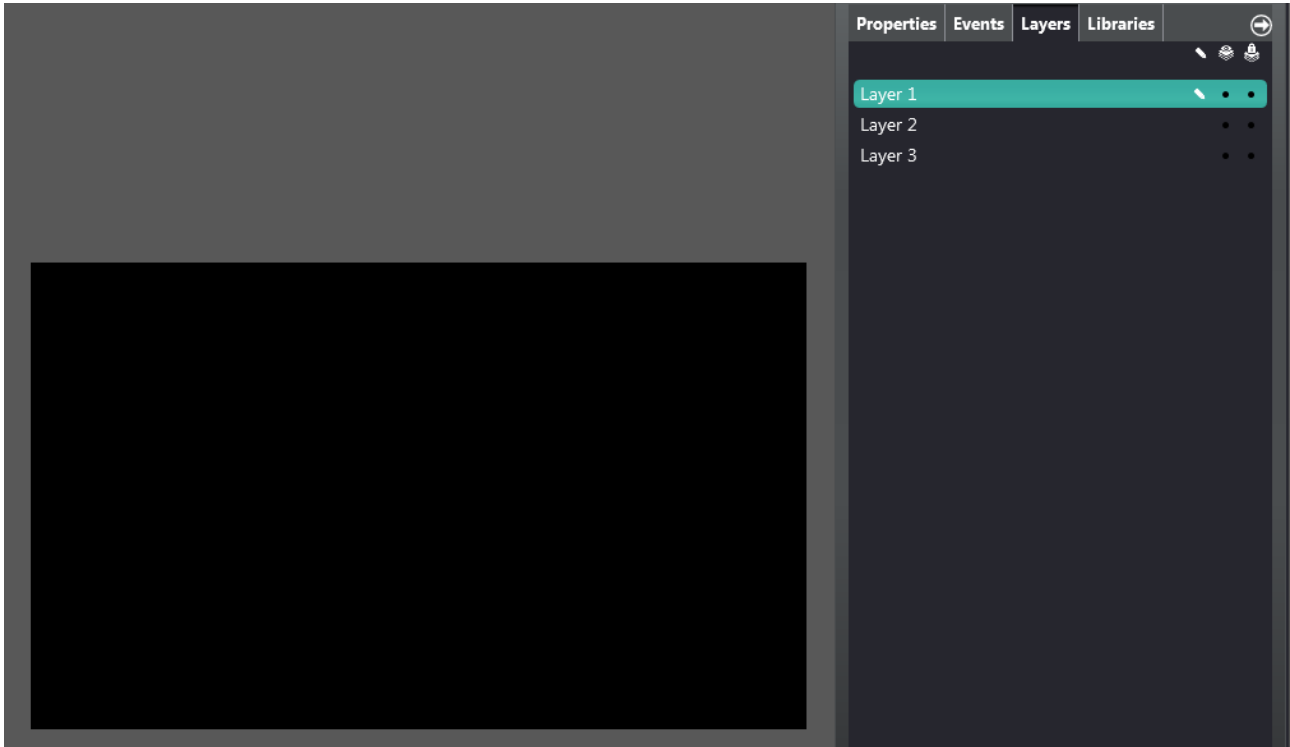
# CREW Manual

Click the “Show- Hide Layers” icon to hide all three layers.

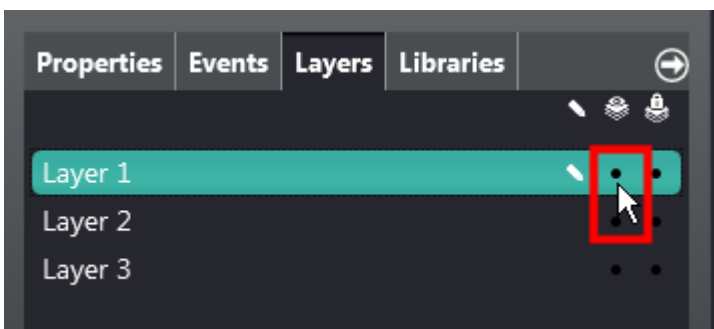


# CREW Manual

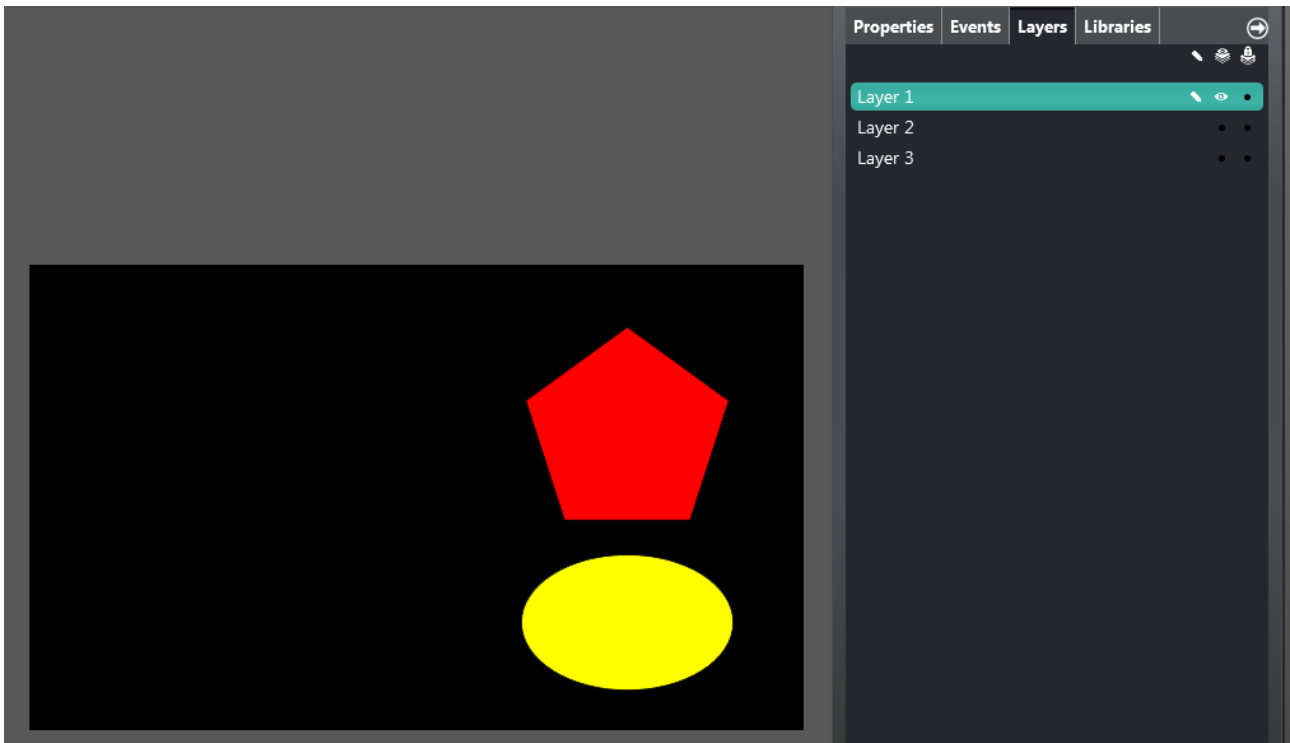
Click the “Show- Hide Layers” icon again to show all three layers.



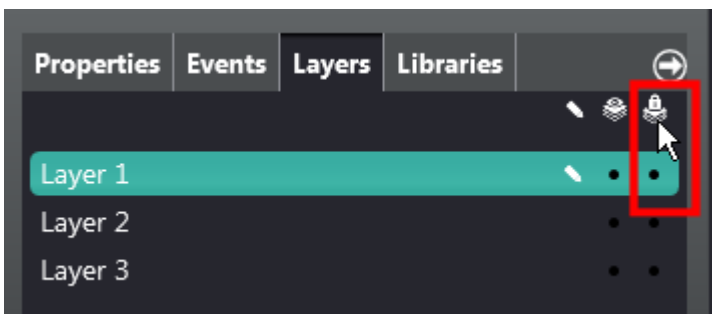
Plus, it is possible to show or hide a single layer at a time. Click the icon next to the name of each layer.



# CREW Manual



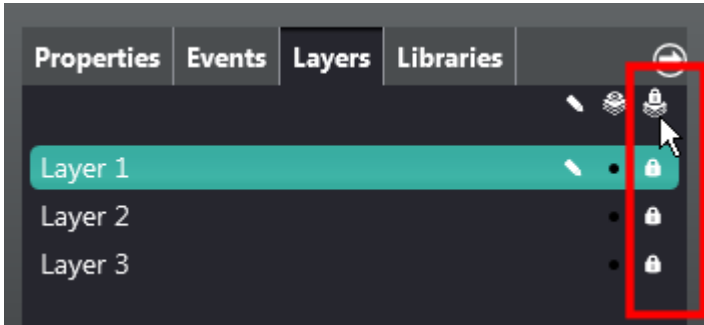
## Block-Unblock Layers



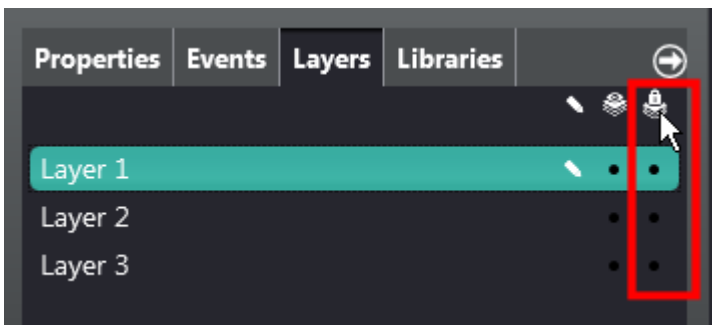
The “Block-Unblock Layers” icon blocks or enables editing to all of the layers on the page.

# CREW Manual

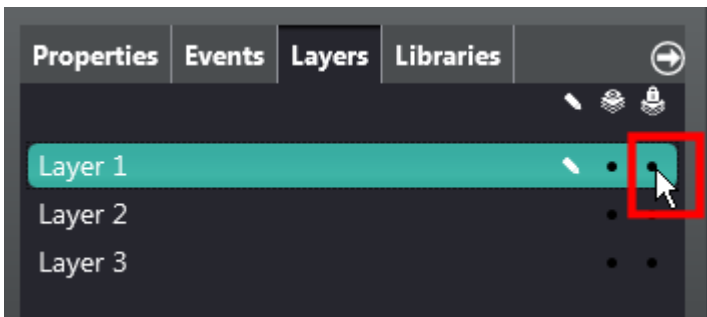
Click the icon once to block editing for all of the active layers.



Click the "Block - Unblock Layers" icon again to unblock the possibility to edit the layers.

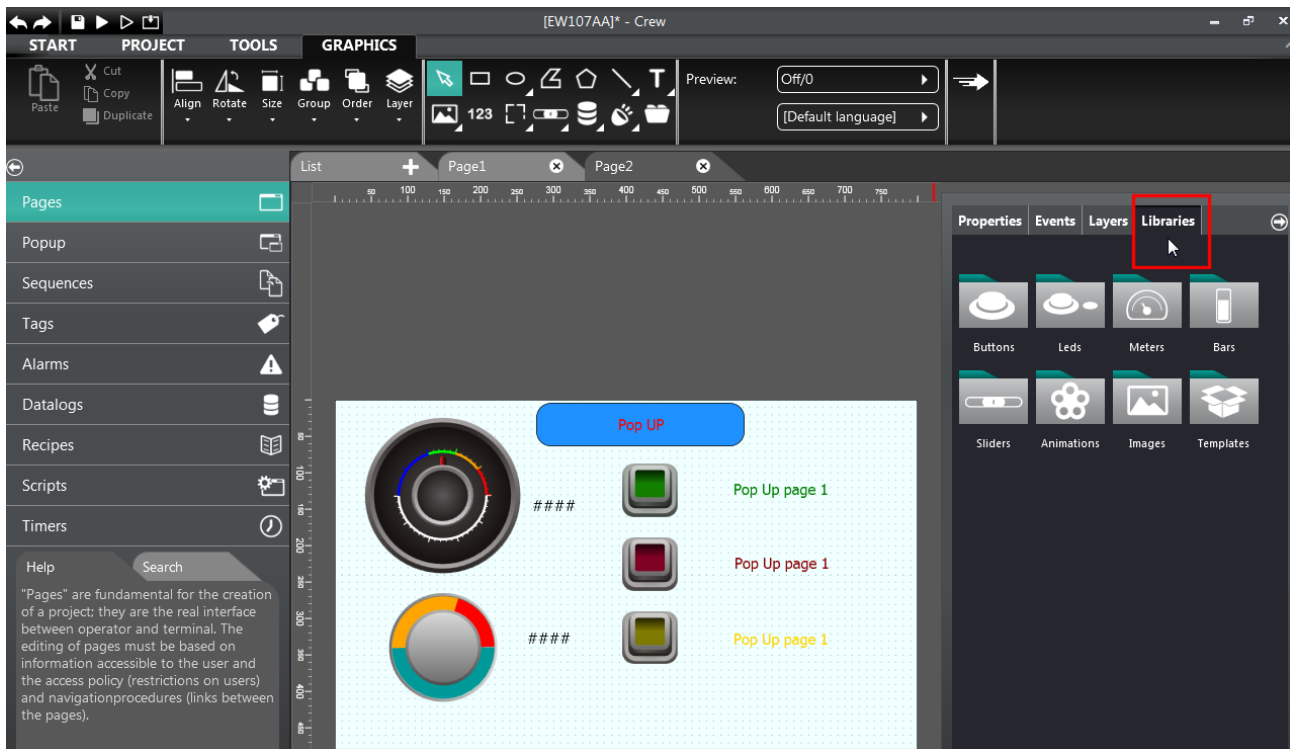


Click the icon next to the name of each layer to block or unblock editing a single layer at a time.



# CREW Manual

## Libraries



The “Libraries” section is the last window of the “Property Editor”. This is where the predefined images are stored (divided by categories) to be used in the pages of the project.

To enter an image on the project page simply use the mouse to select the object that you wish to add and drag it to the page.

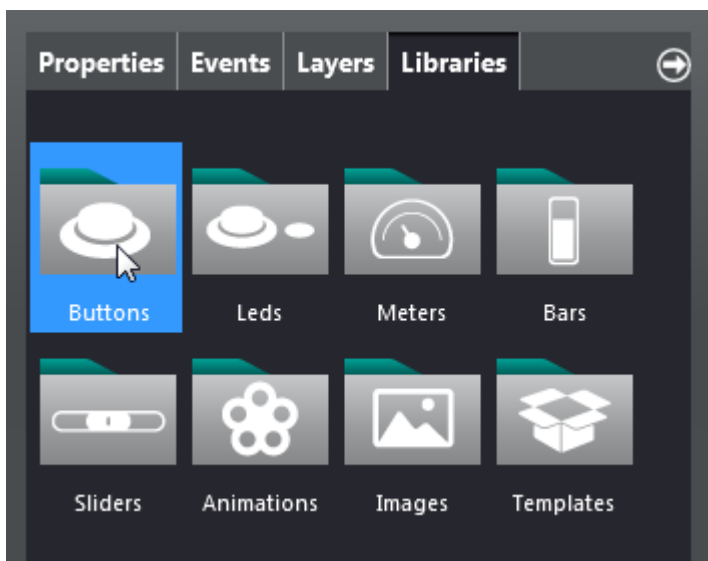
Images belong to the following categories:

- Buttons (buttons and switches)
- Led lights
- Indicators (speedometers, graduated scale indicators, etc.)
- Bars (images with graduated scales)
- Selectors (graduated scale selectors)
- Animations (animated images)

# CREW Manual

- Images (industrial images such as motors and pipelines, but also Esaware logos and flags)
- Templates (customised images created by the user)

## Buttons







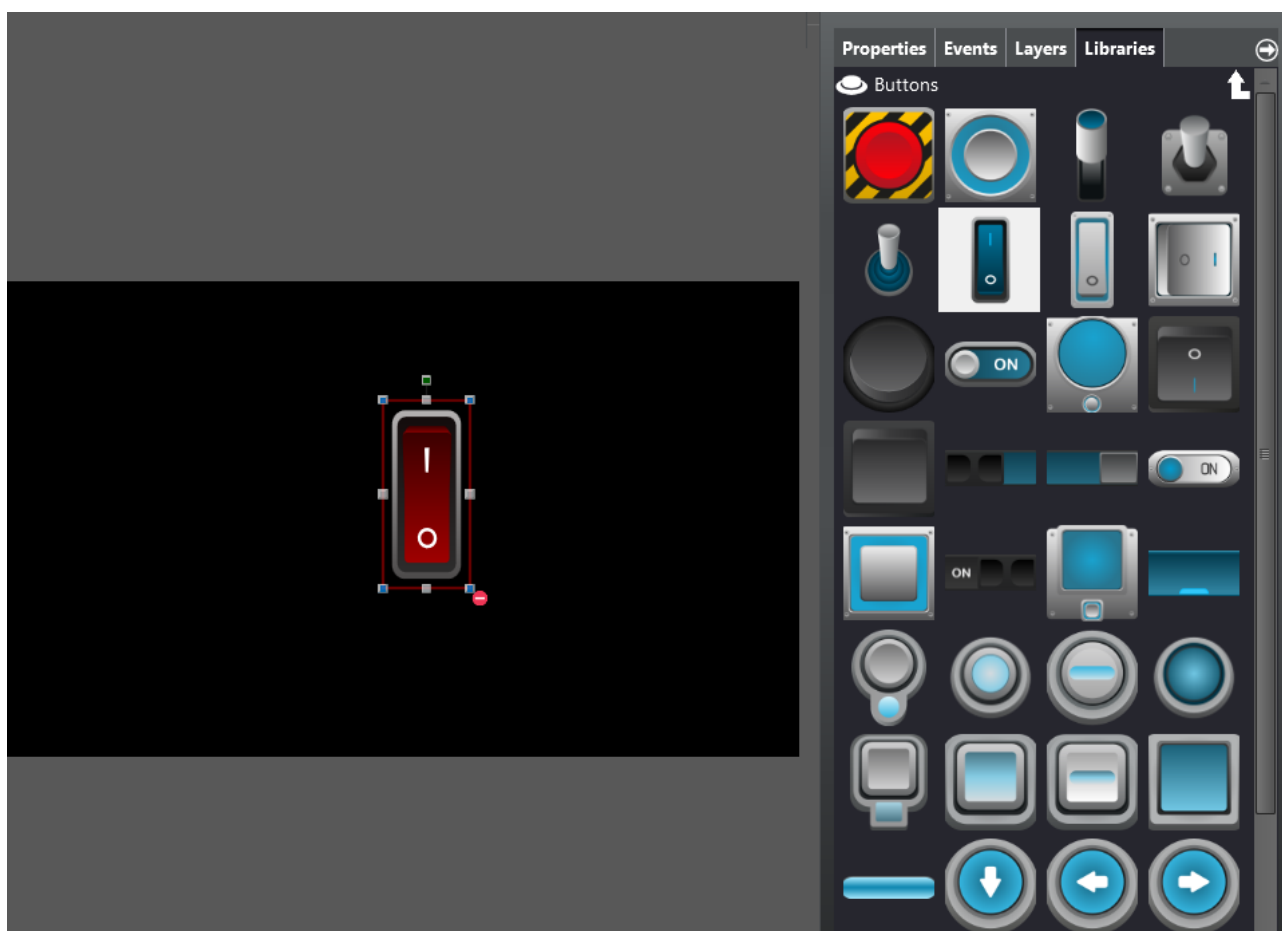
# CREW Manual

Use the mouse to select a button and drag it into the page to enter it into the project.



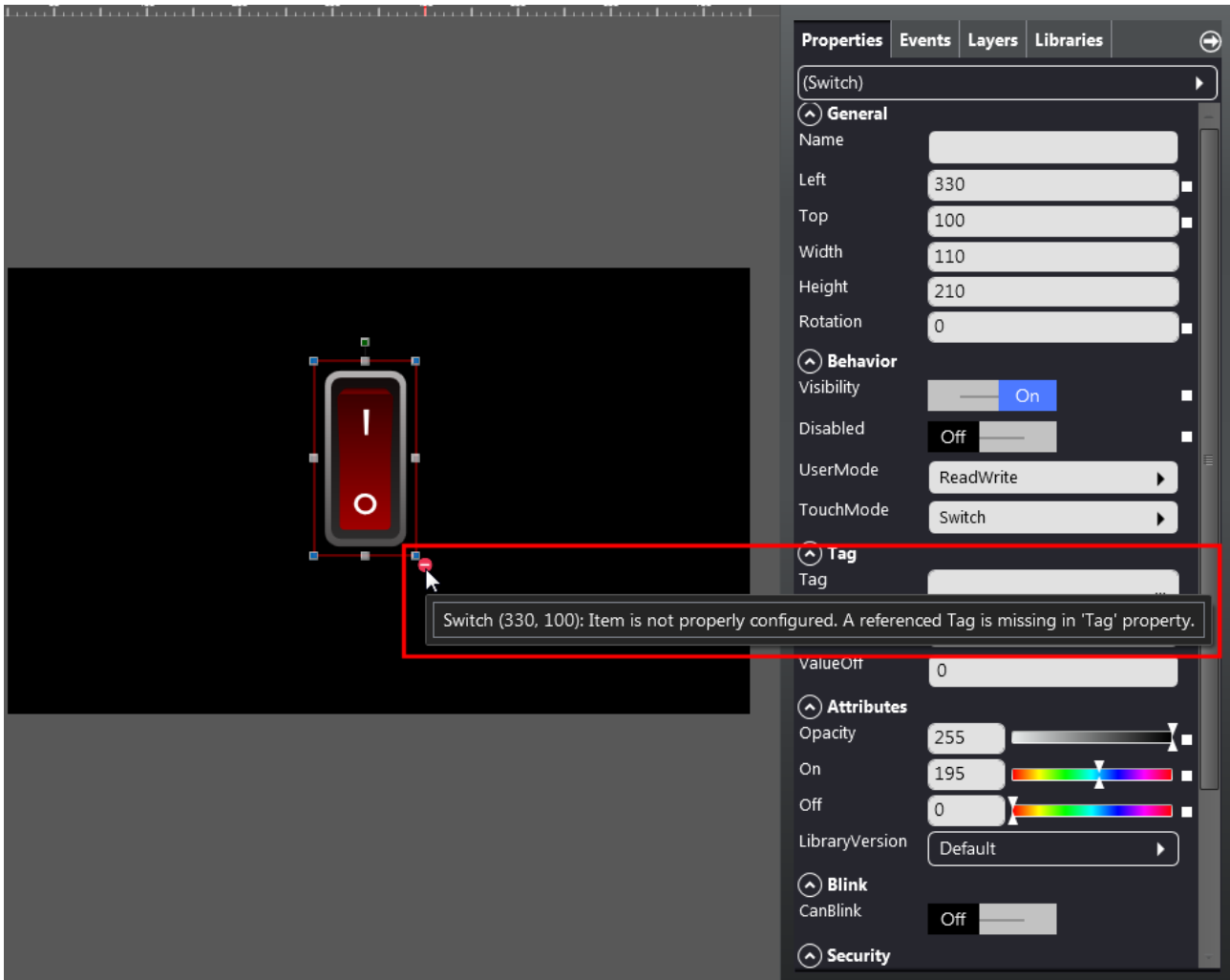
# CREW Manual

The object will now appear in the page.



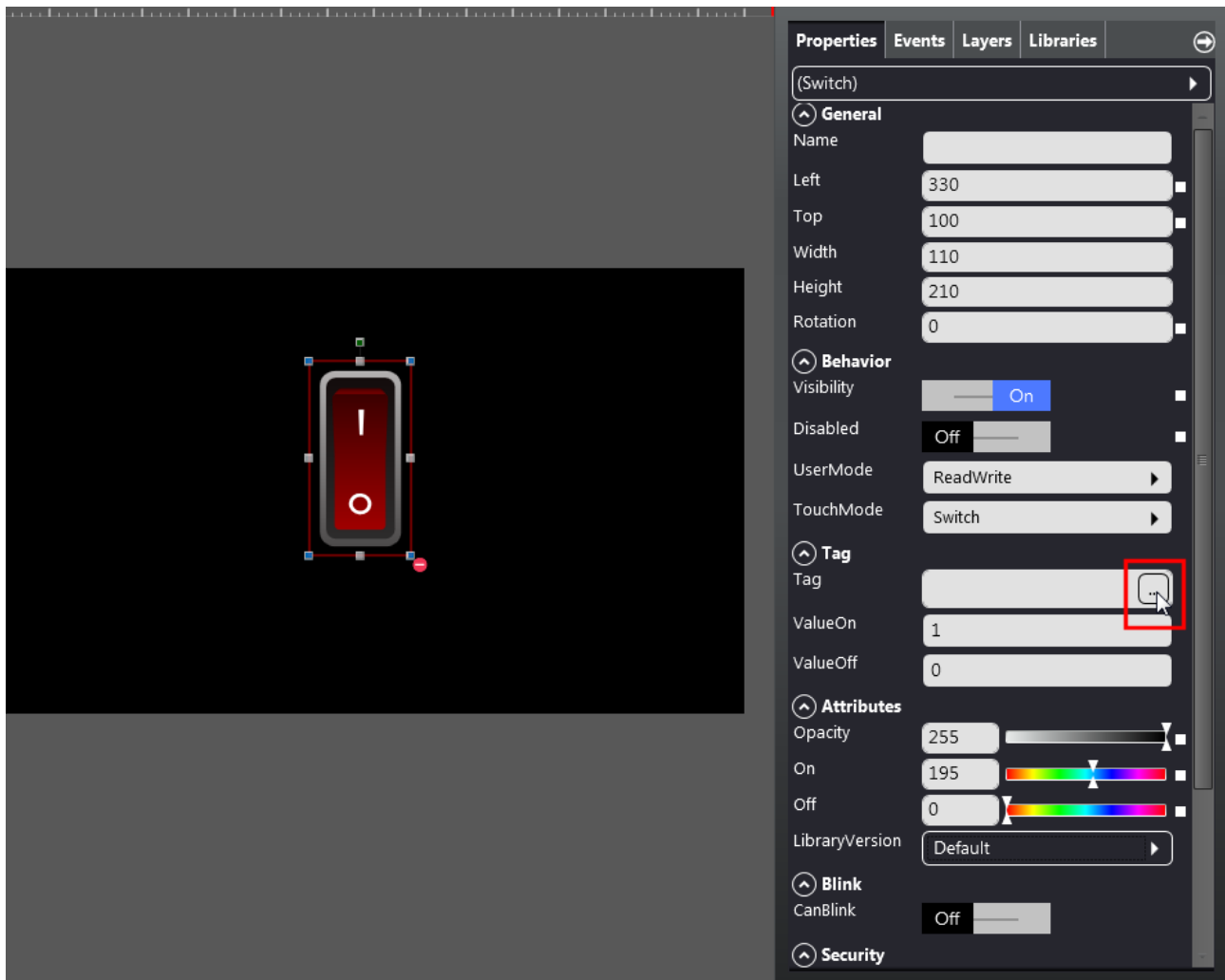
# CREW Manual

Next to the object there is an error alert that disappears when a reference Tag is associated.



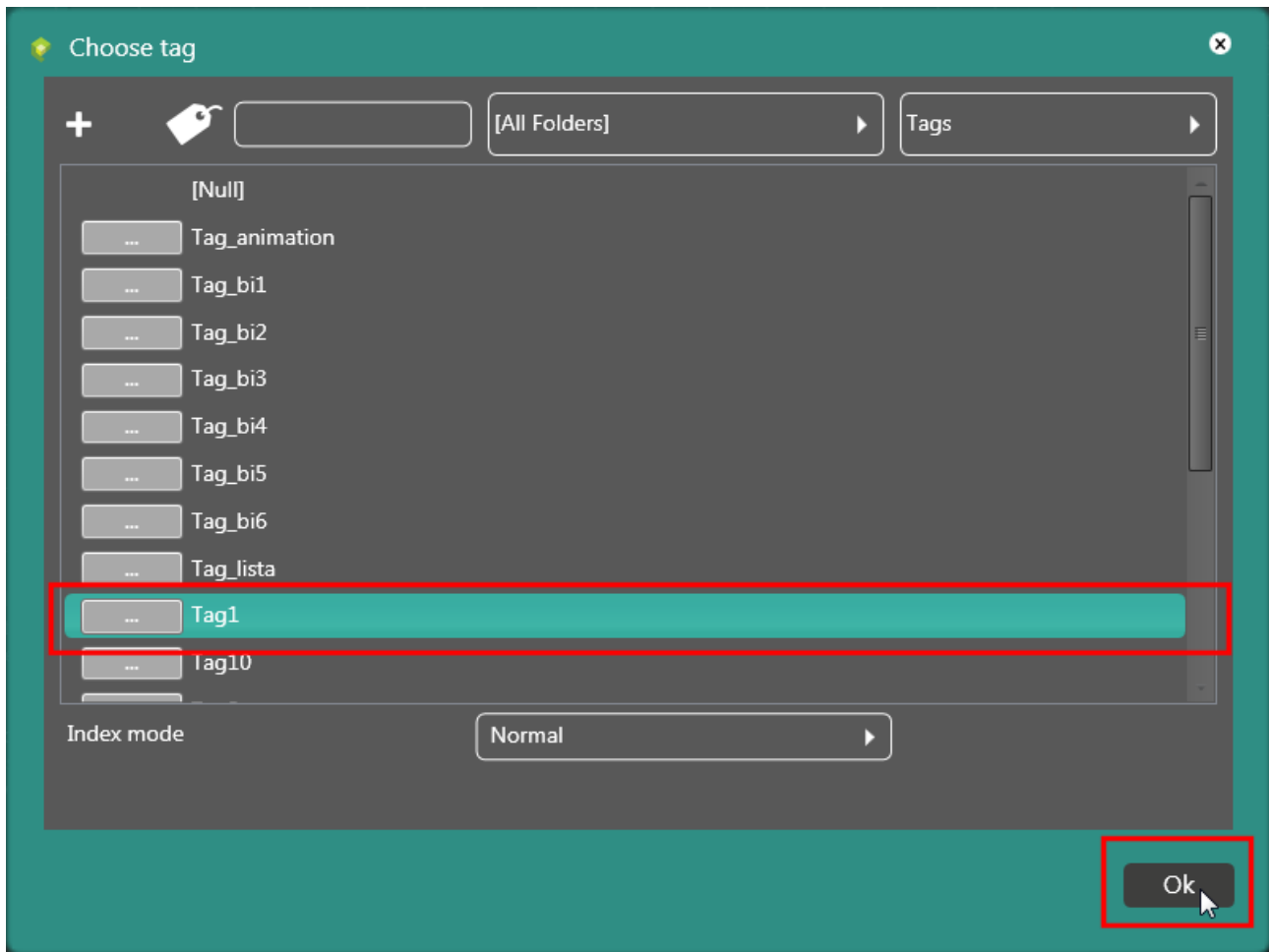
# CREW Manual

Select the drop down menu in the “Property Editor” to choose the variable to be associated.



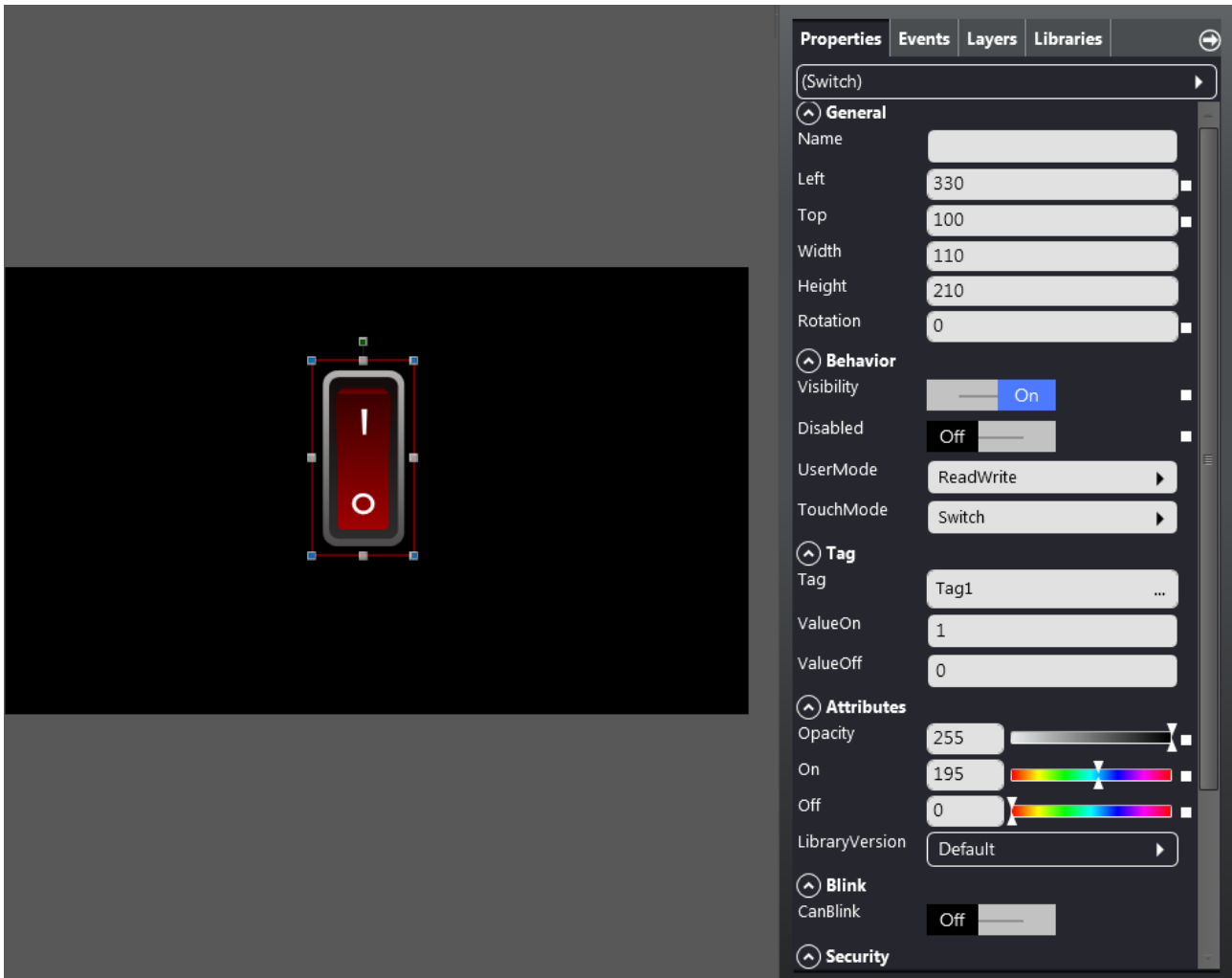
# CREW Manual

Associate a variable (for example “Tag1”) and confirm with “Ok”.



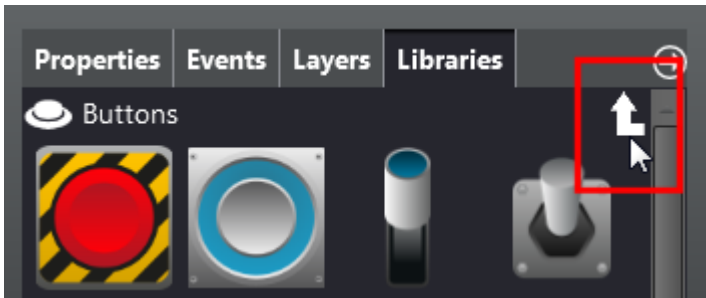
# CREW Manual

Now the error alert next to the object has disappeared.



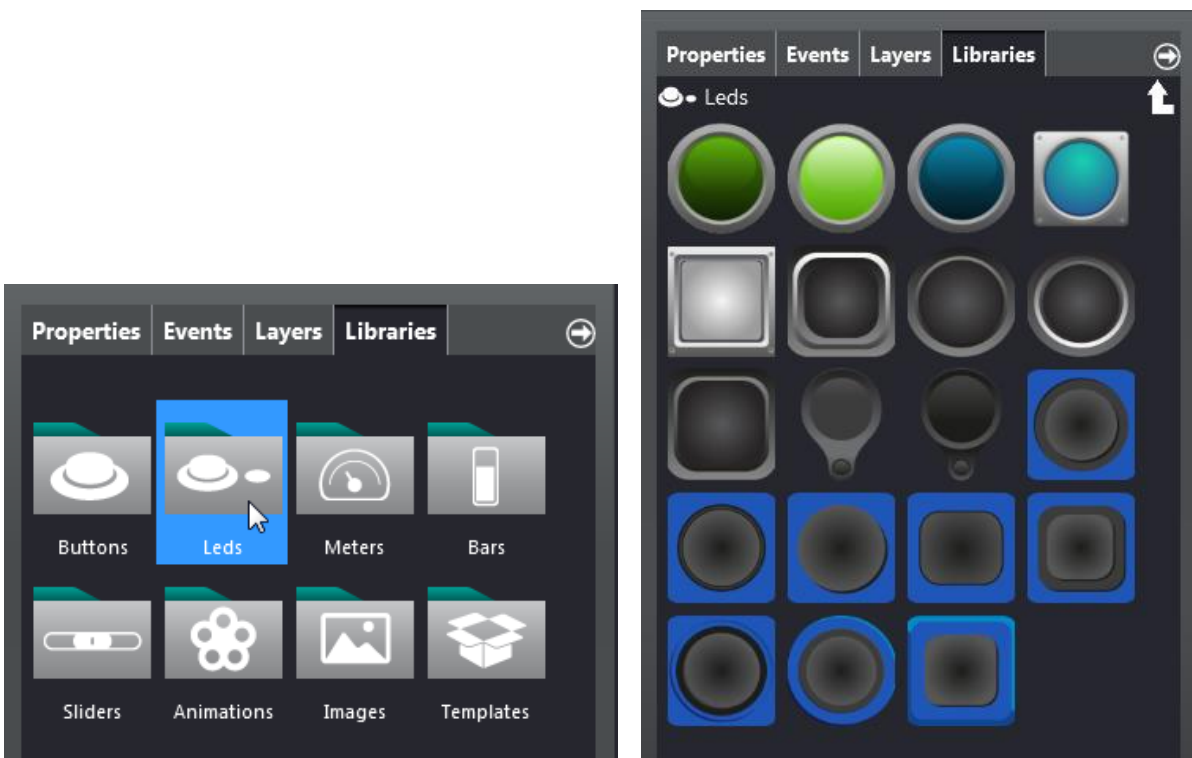
# CREW Manual

Use the vertical arrow shown in the image to go back to the main menu of the “Libraries” property.



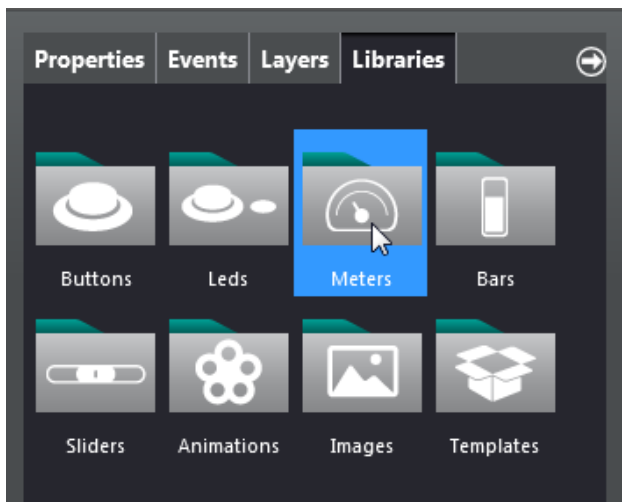
Here it is possible to select the folder that you wish to use, from:

Led lights



# CREW Manual

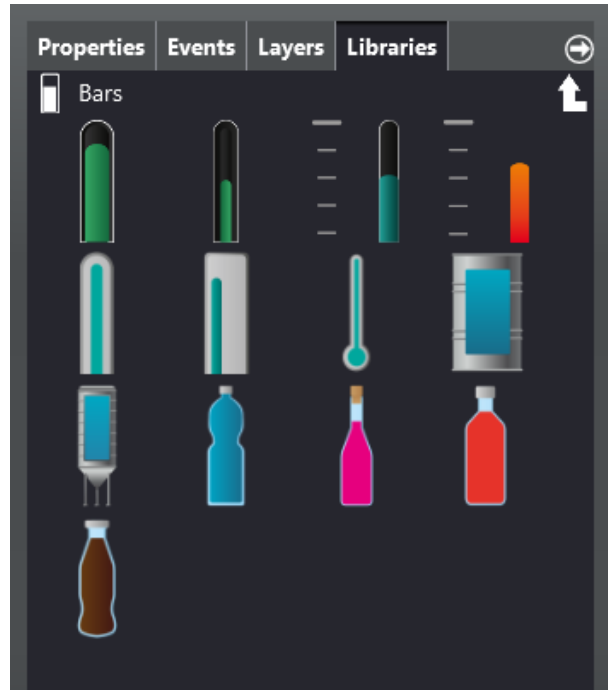
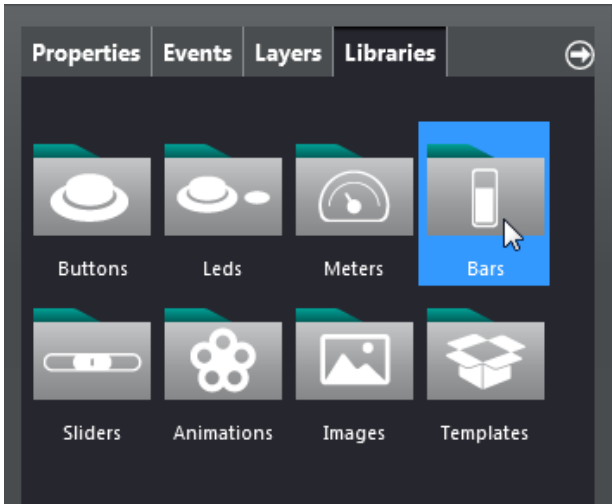
## Indicators





# CREW Manual

## Bars



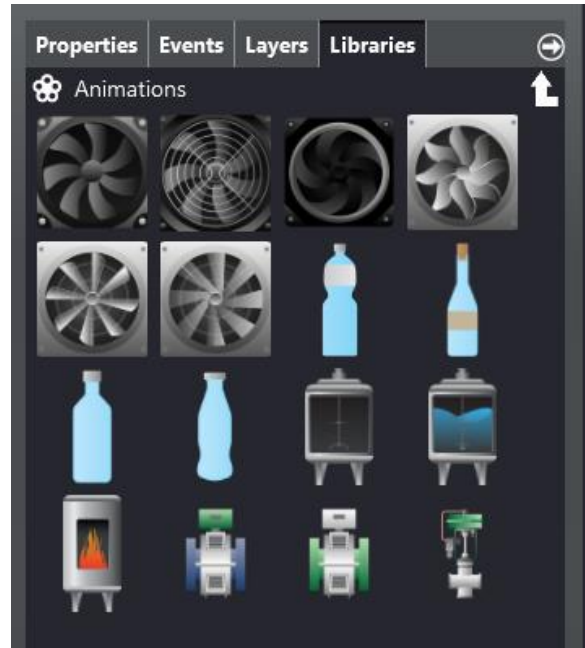
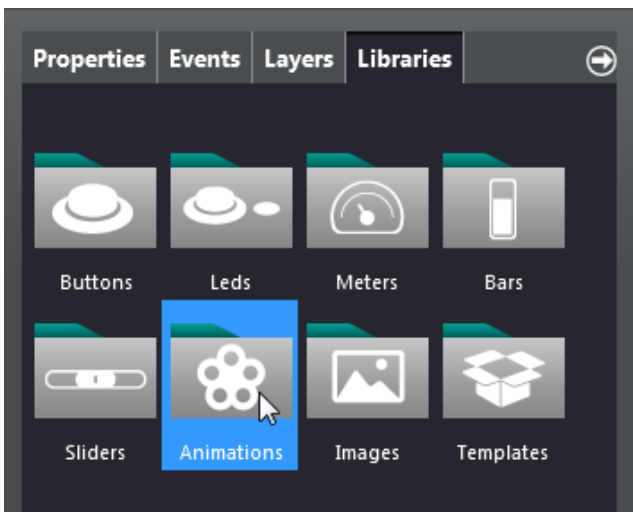
# CREW Manual

## Selectors



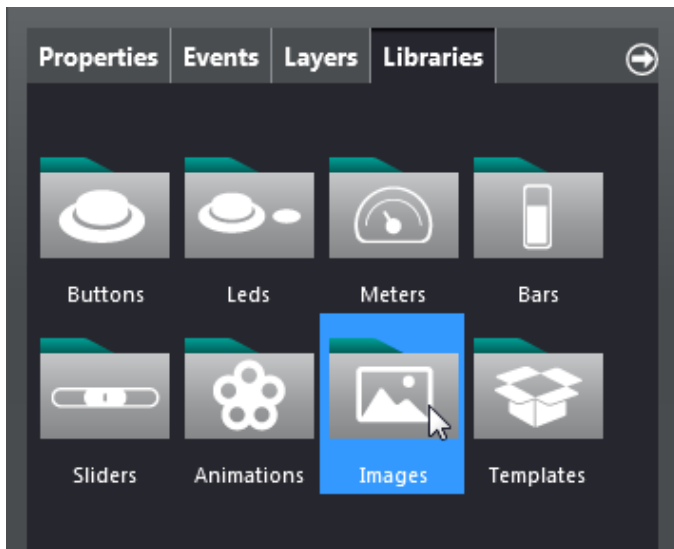
# CREW Manual

## Animations



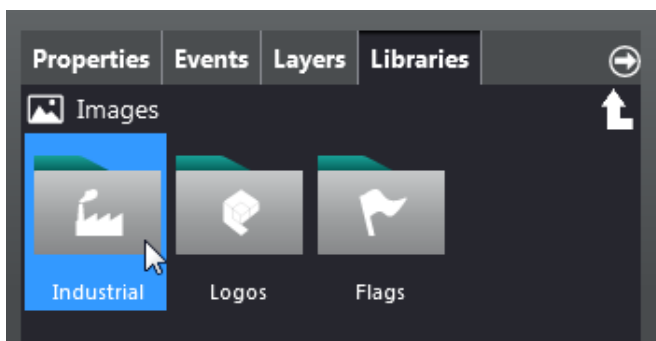
# CREW Manual

## Images



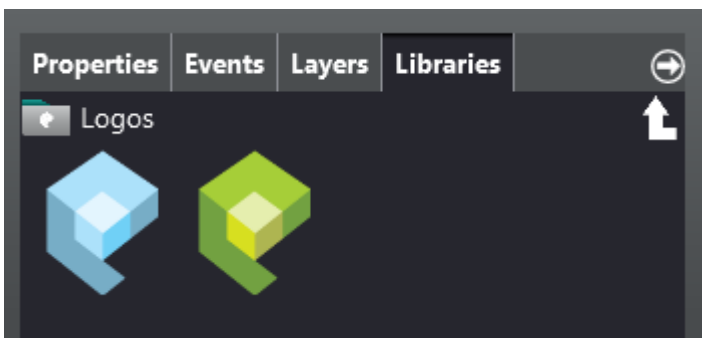
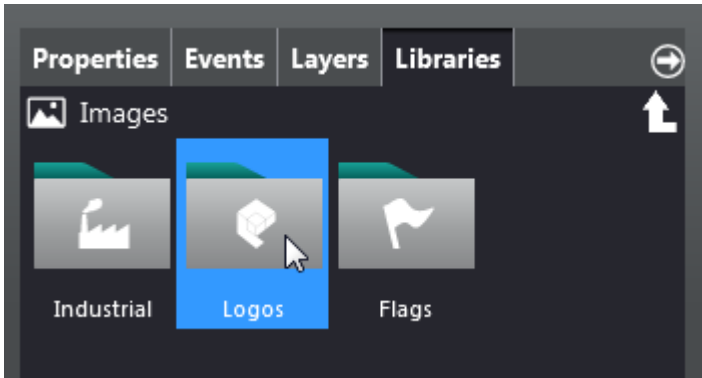
# CREW Manual

## Images (Industrial)



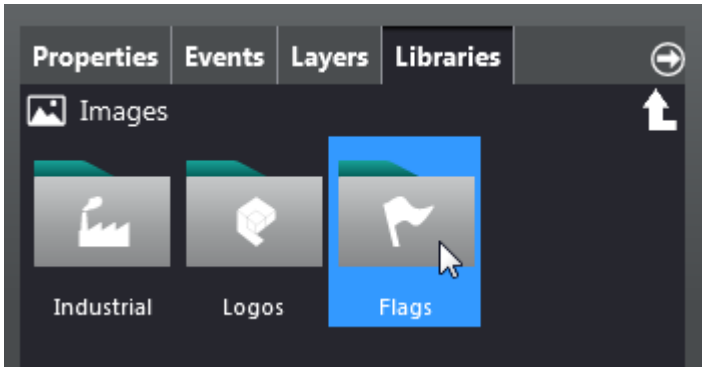
# CREW Manual

## Images (Logos)

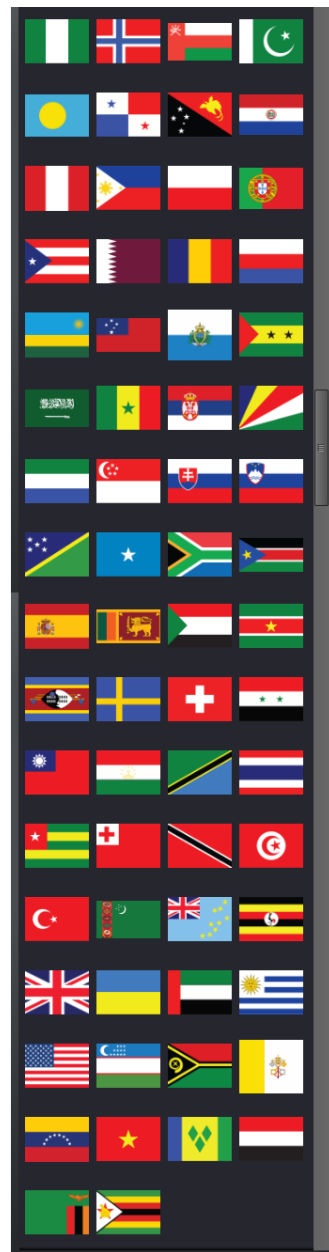
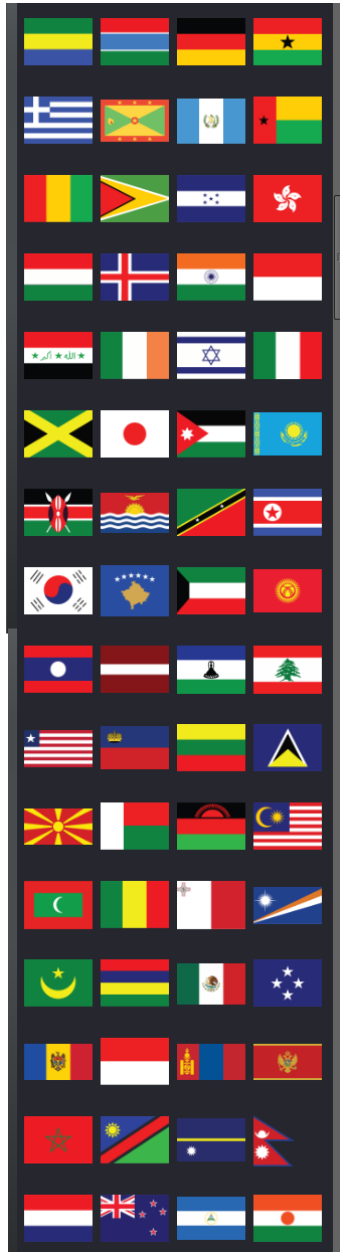
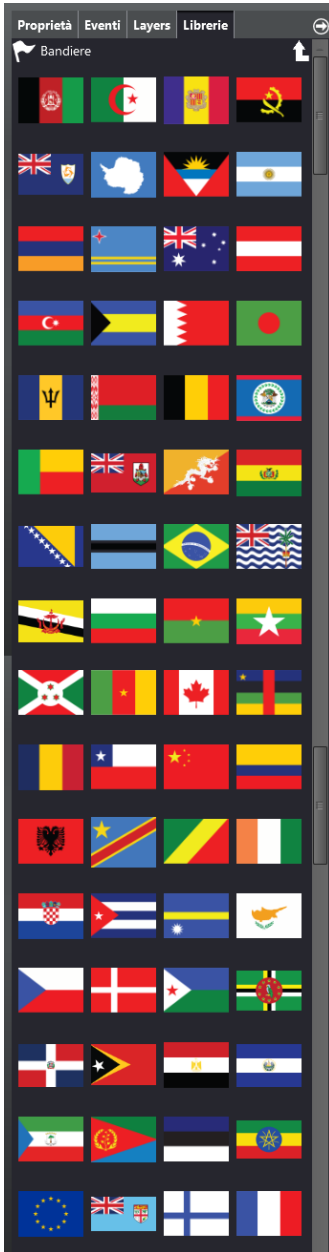


# CREW Manual

## Images (Flags)



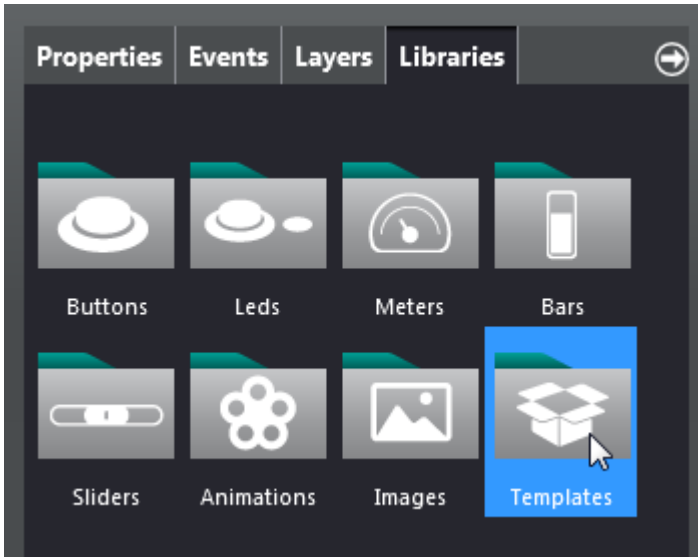
# CREW Manual



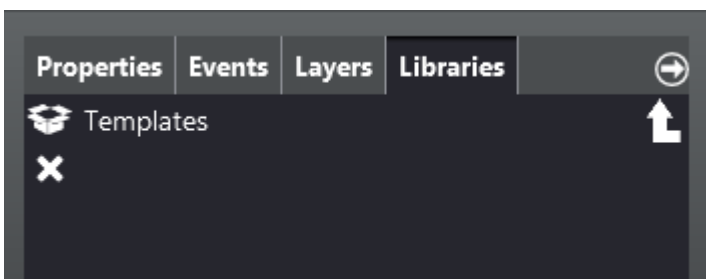


# CREW Manual

## Templates

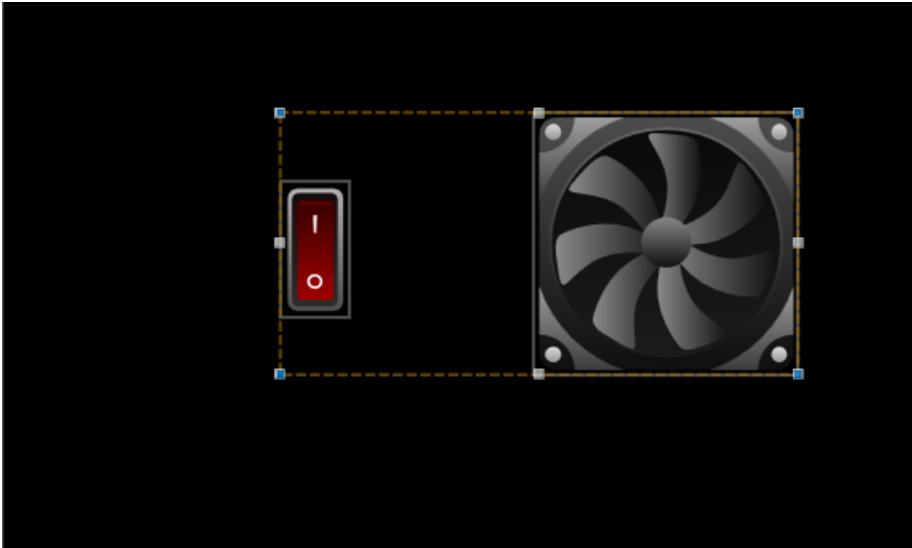


The “Templates” folder offers the user-created objects.

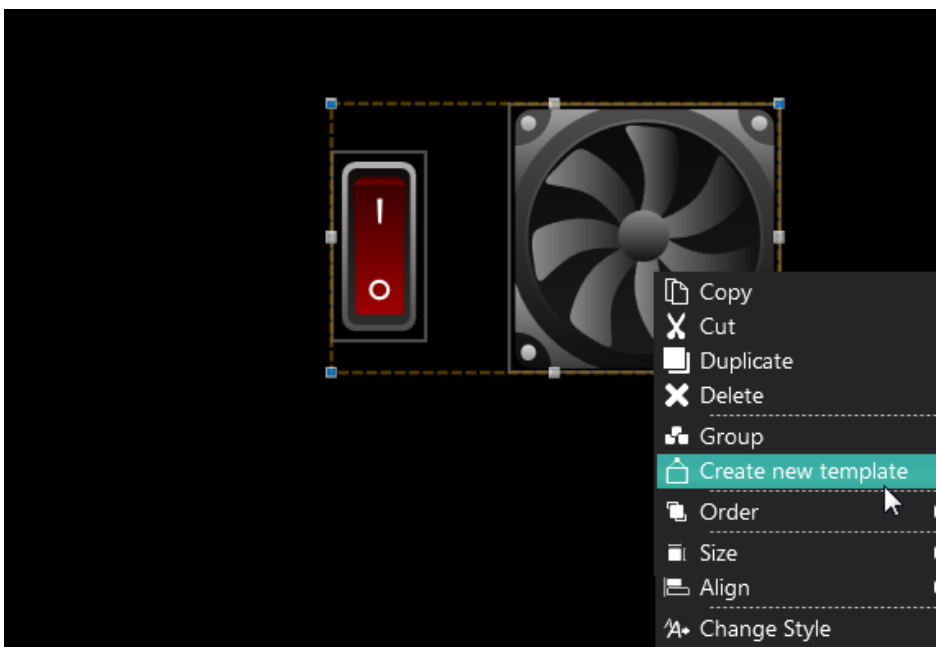


# CREW Manual

Place a customised object on the page (for example, an object composed of a switch and a fan) and select it with the mouse.

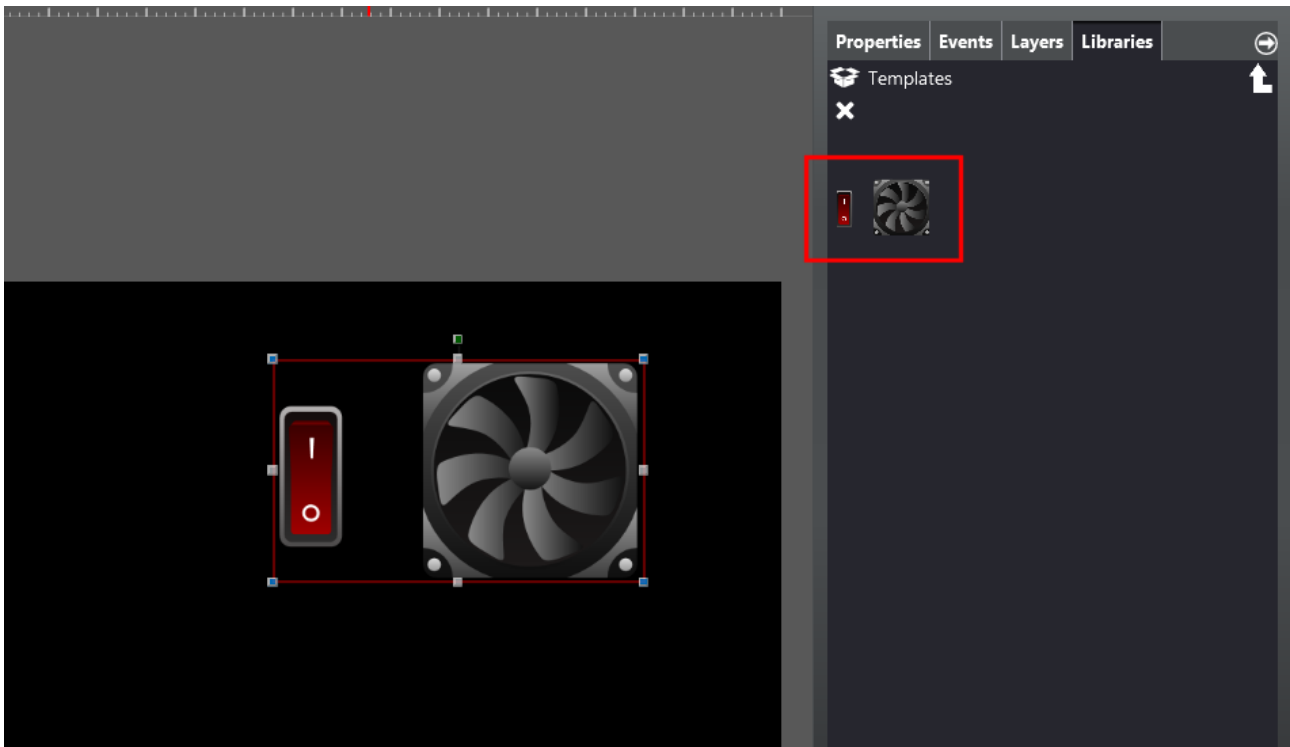


Right click and choose the “Create new template” option.



The customised object now appears under the “Templates” folder.

# CREW Manual



# Explore Project

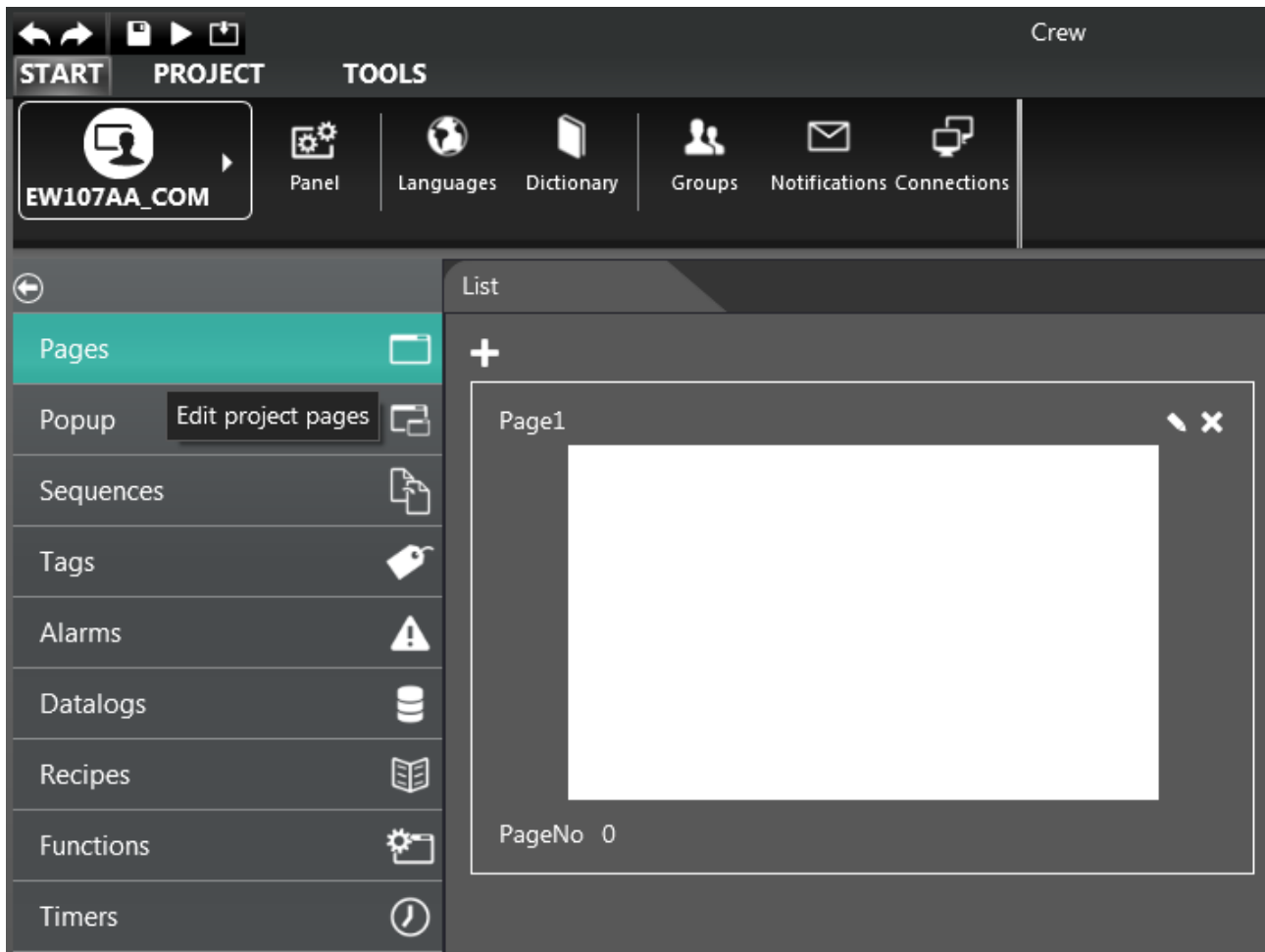
---

## Pages

The "Pages" are key to realising a project, as they represent the true interface between operator and terminal. Page editing must be based on the information accessible to the user and on access (user restrictions) and navigation (links between pages) policies.

# CREW Manual

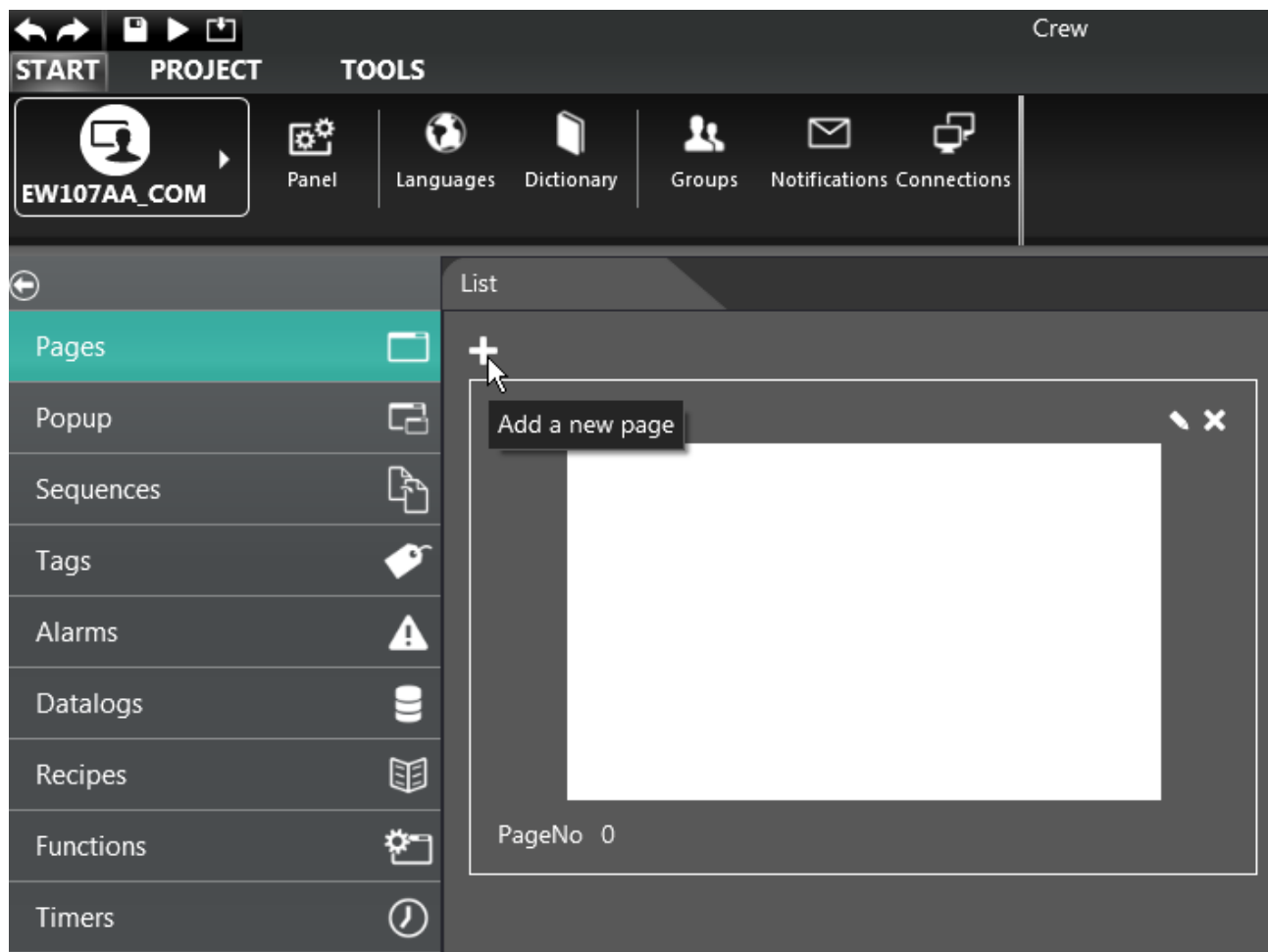
Clicking the "Pages" menu to view a list of pages included in the project.



# CREW Manual

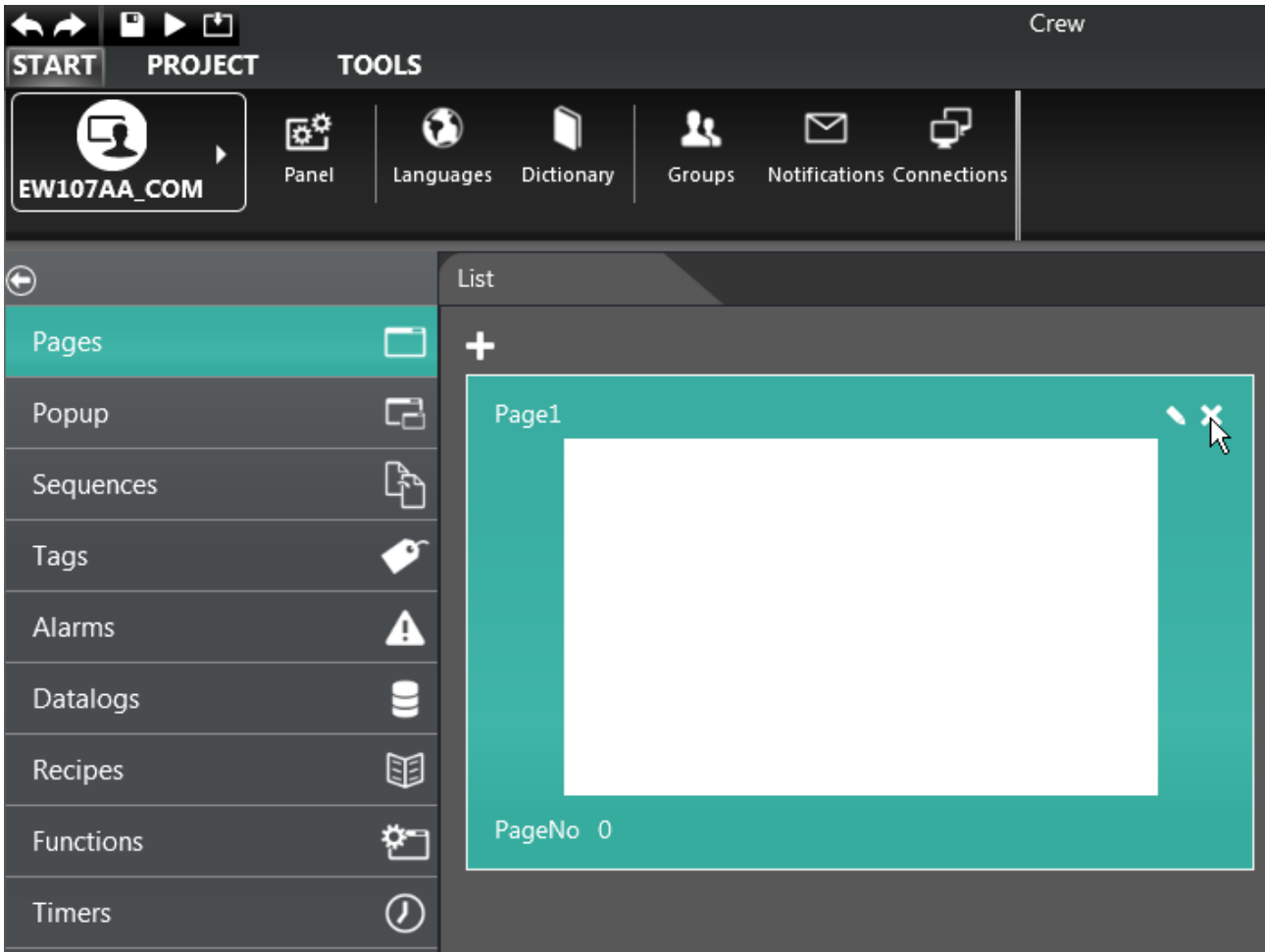
From this list it is possible to do the following operations.

Enter new pages.



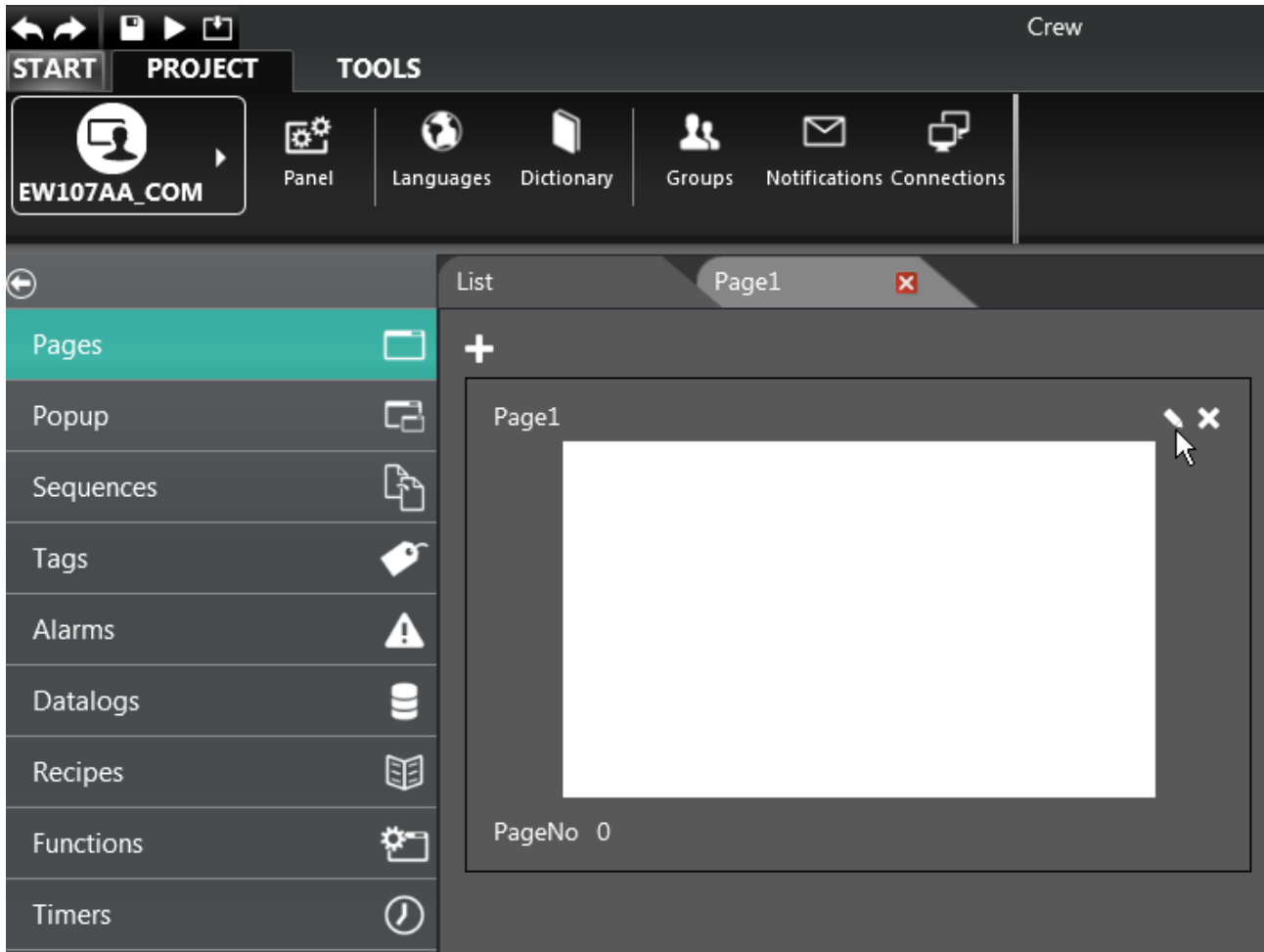
# CREW Manual

Delete the selected page.



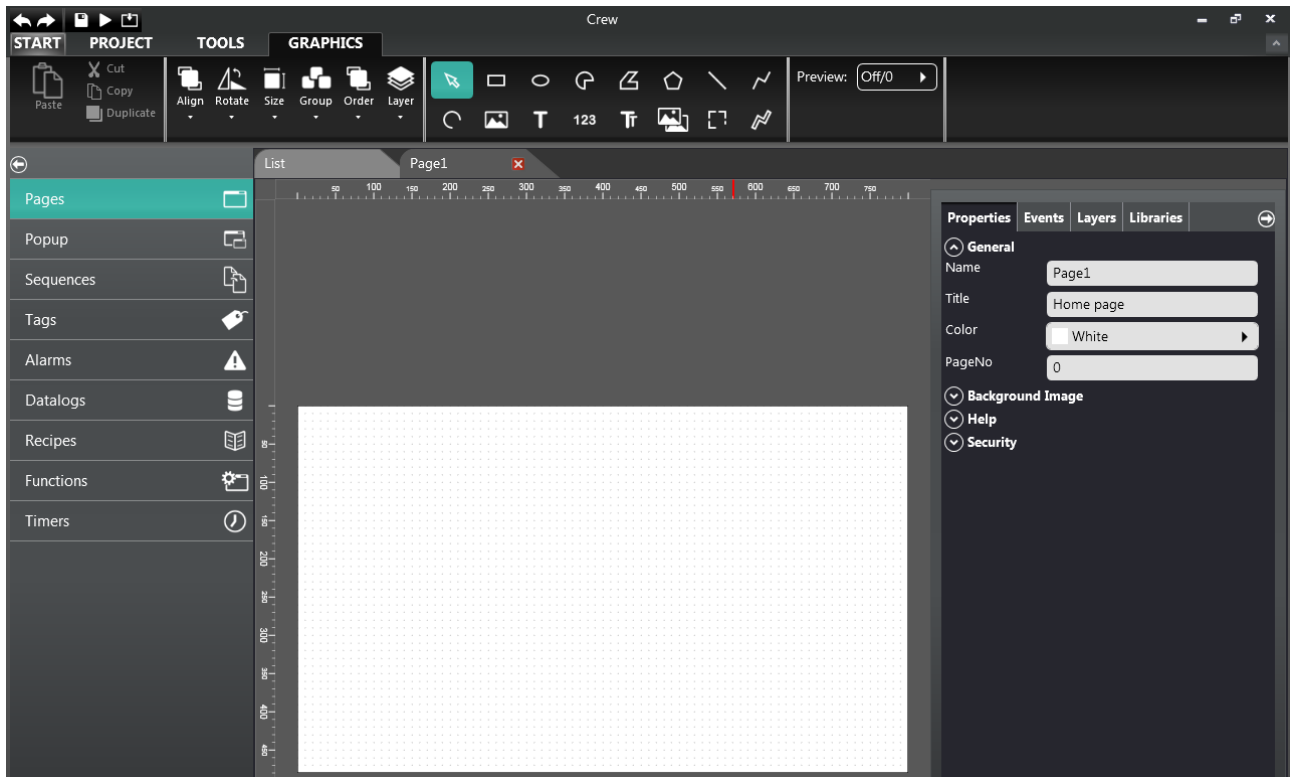
# CREW Manual

Enter the page editor.



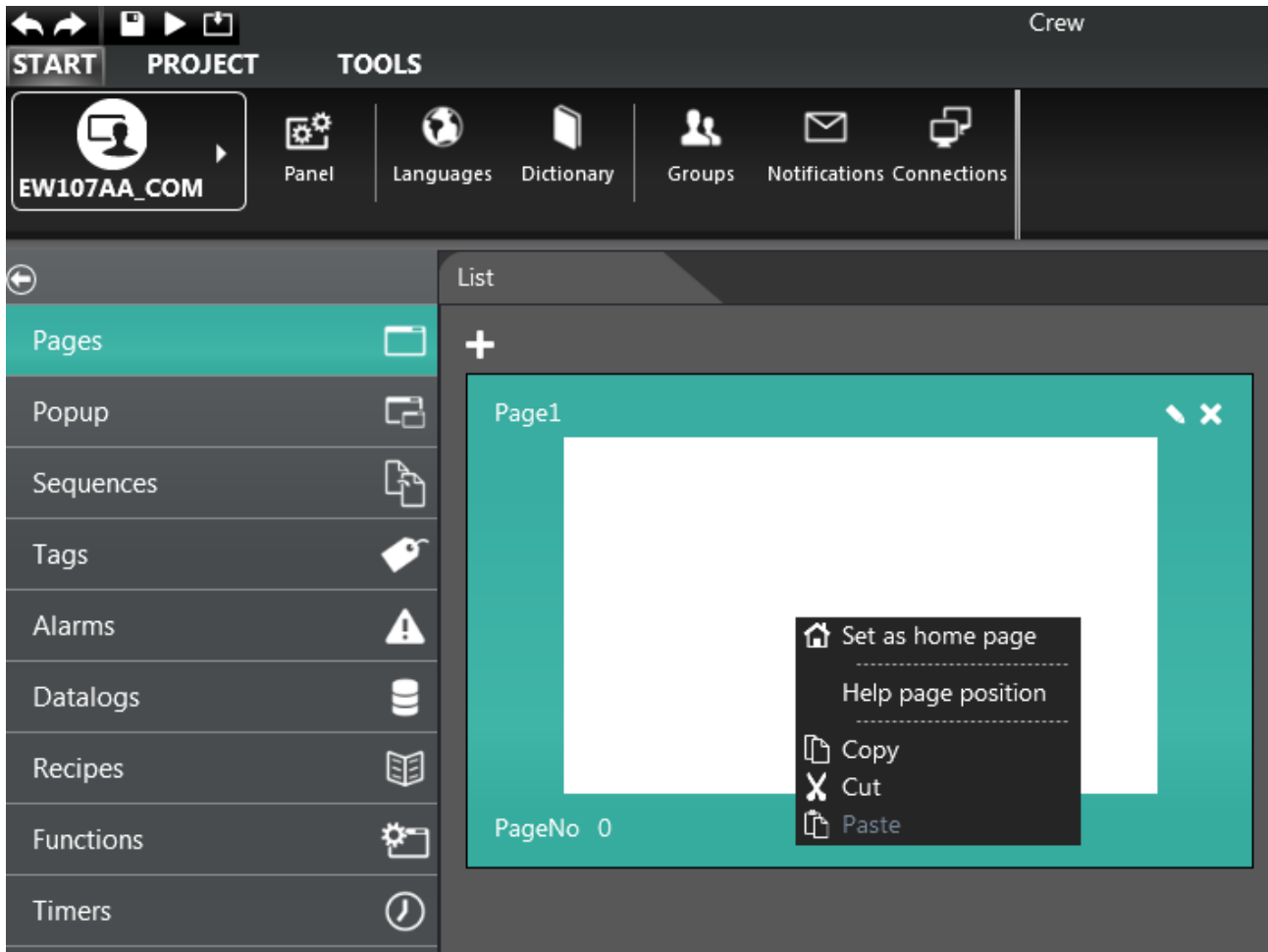


# CREW Manual



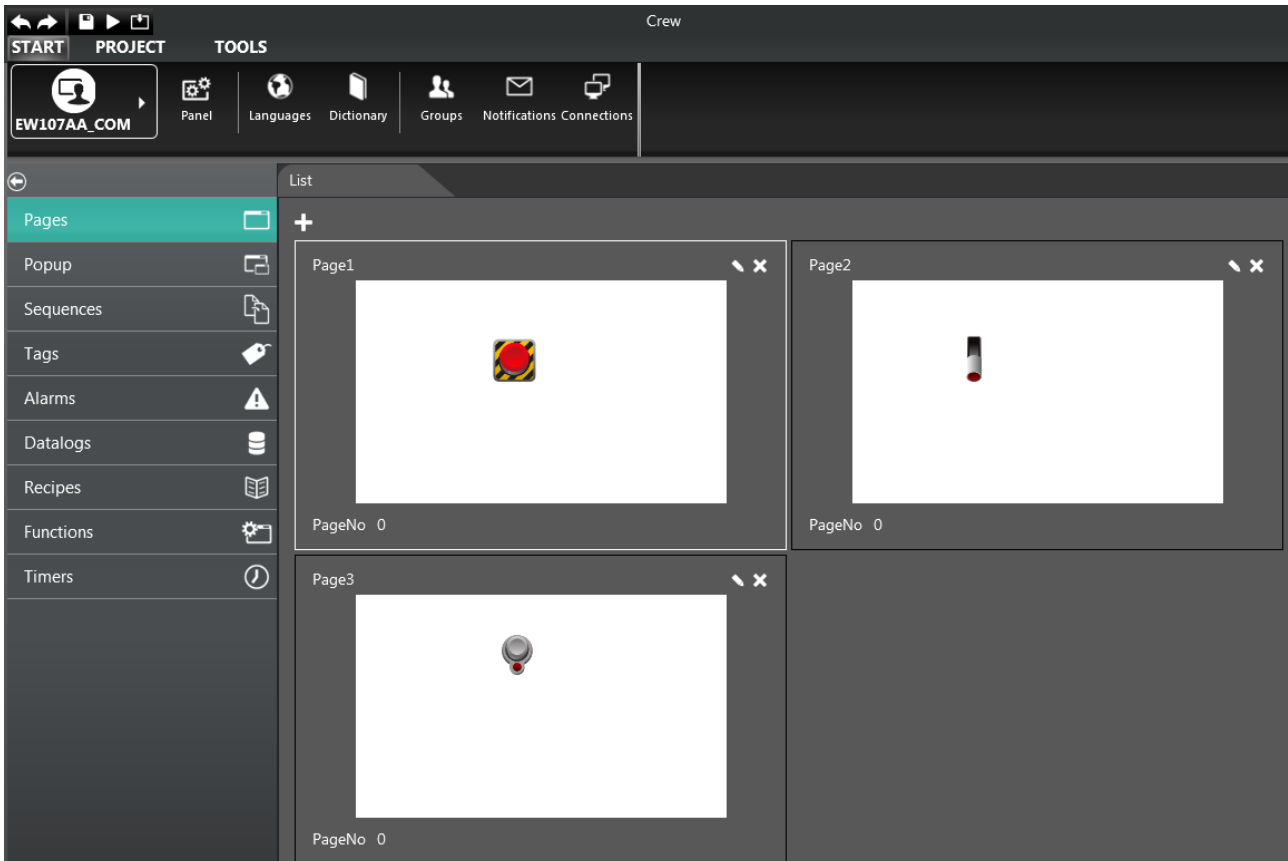
- Plus, after right clicking the mouse you can:
- Set a page as the “Home page” of the project.
- Define the position of the “Help Page”.
- Duplicate the existing pages (cut, copy or paste).

# CREW Manual



# CREW Manual

During the design phase, the "Home page" is identified by a white edge (as shown in the image).



Once a page has been created, double-click it (or click the aforementioned icon), to edit it in the work area.

The page editor is described in the "[Page management](#)" section. It is also possible to edit the properties of the page (see "[Page Properties](#)" section) and link events (see "[Page Events](#)" section).

# CREW Manual

## Page management

All of the graphic and visual characteristics of the project are illustrated on the basic "Page" element.

In order for a graphic element to be visible by the operator in Runtime, it needs to be placed on a page.

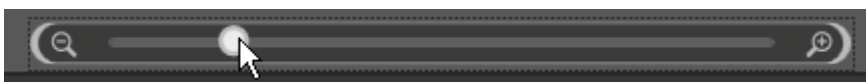
The "Zoom" commands are at the bottom of the screen.



Click the drop down menu and choose from the available Zoom options to change the size of the displayed page.

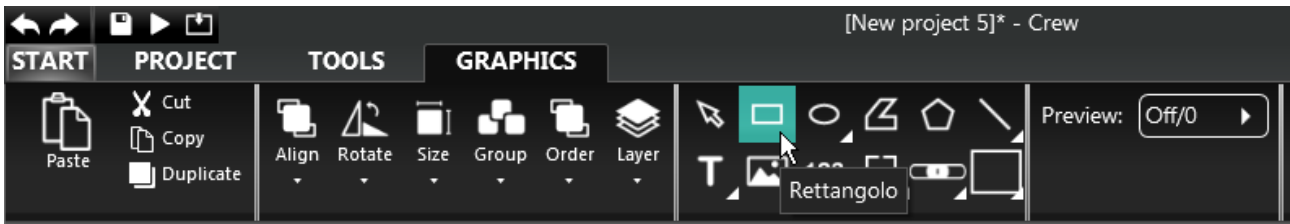


Or you can zoom with the zoom slide.

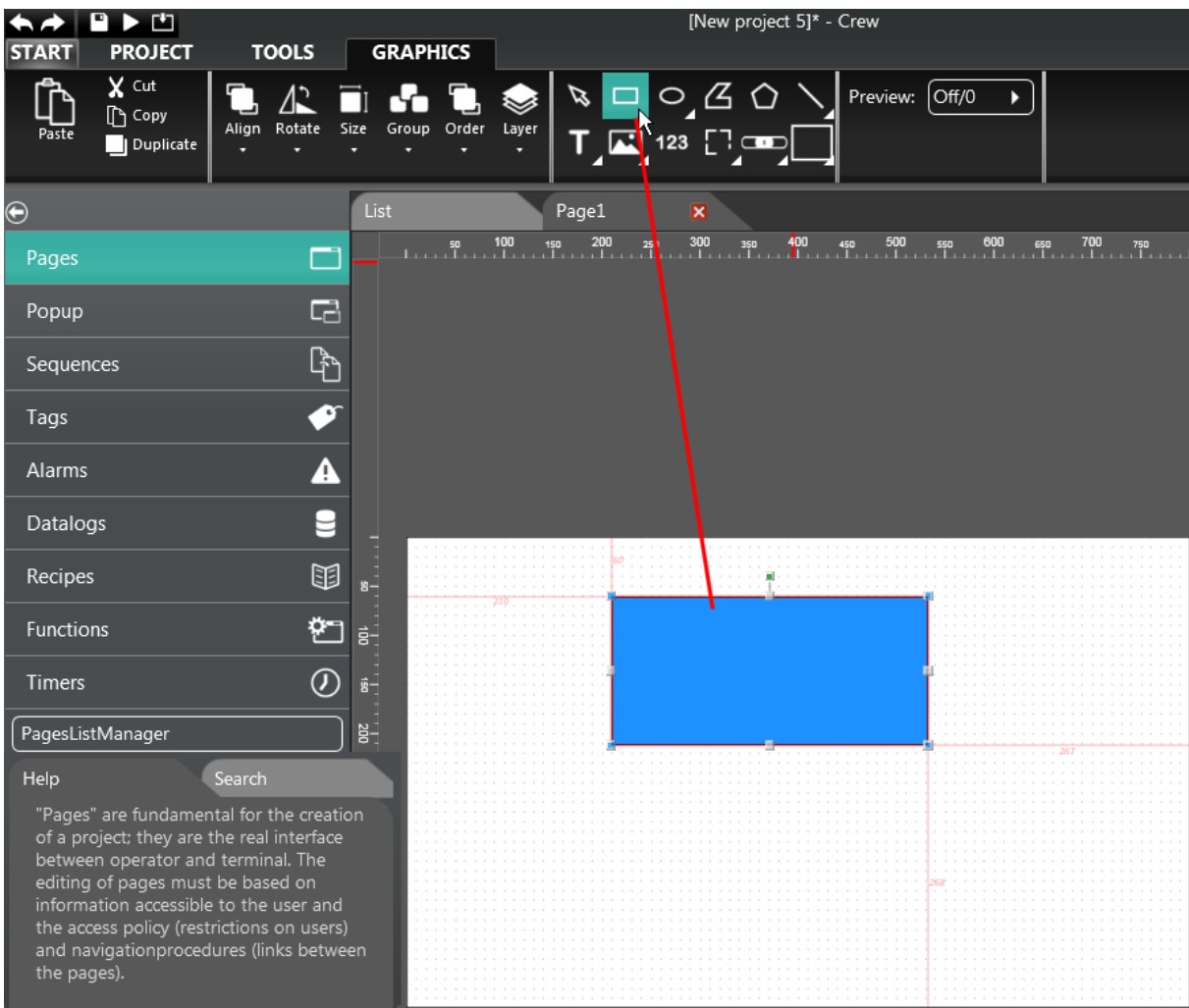


To enter an object on the page click the relative icon in the toolbar.

# CREW Manual



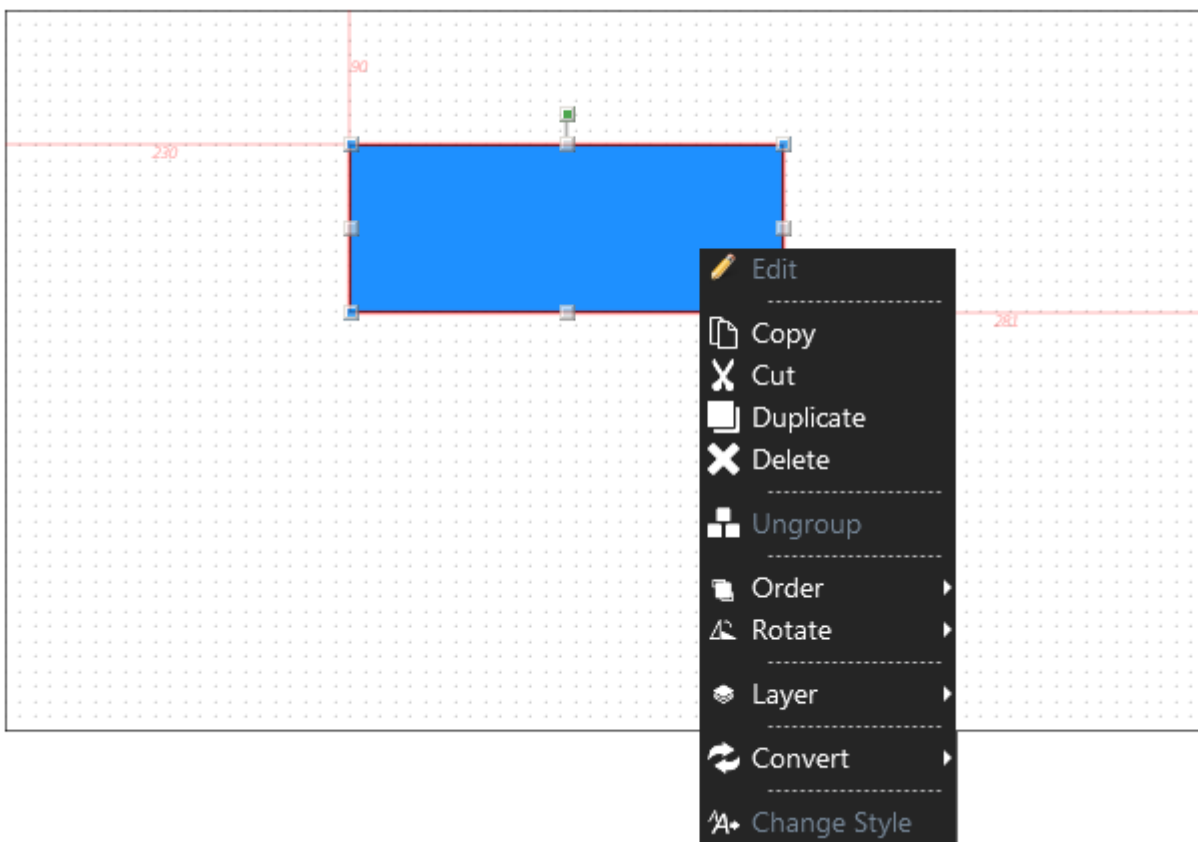
Trace the outline, in the desired position, on the preview page.



# CREW Manual

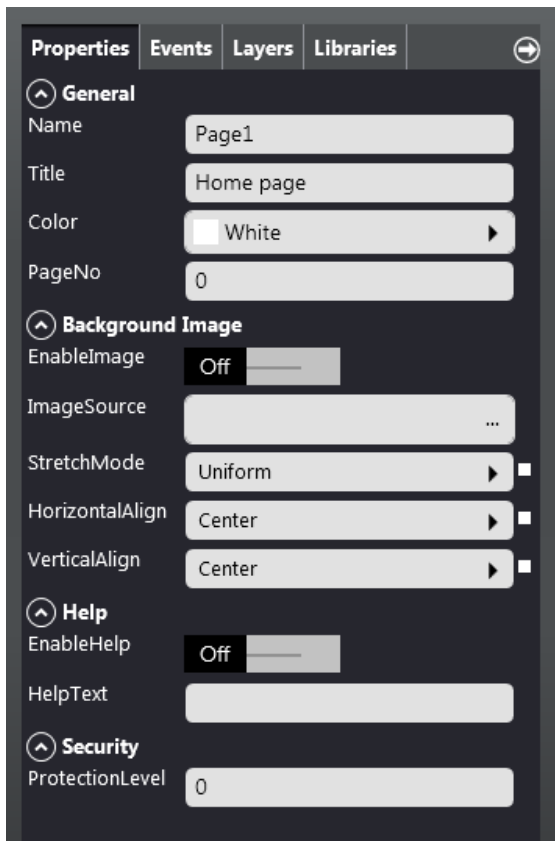
When it is entered, the element appears on the page and can be selected with a click. For each selected object, all of the user-settable items appear in the “Property Editor” and “Event Editor” (see the “[Property](#)” and “[Event](#)” sections).

Right click a selected object to access a menu with the following functions: “Edit”, “Copy”, “Cut”, “Duplicate”, “Delete”, “Order” (see “[Order Submenu](#)”), “Rotate” (see “[Rotate Submenu](#)”), “Layer” (see “[Layers](#)” and “Conversion”).



# CREW Manual

## Page properties



The screenshot shows a 'Properties' panel with several sections:

- General**
  - Name: Page1
  - Title: Home page
  - Color: White
  - PageNo: 0
- Background Image**
  - EnableImage: Off
  - ImageSource: [empty field with ... button]
  - StretchMode: Uniform
  - HorizontalAlign: Center
  - VerticalAlign: Center
- Help**
  - EnableHelp: Off
  - HelpText: [empty field]
- Security**
  - ProtectionLevel: 0

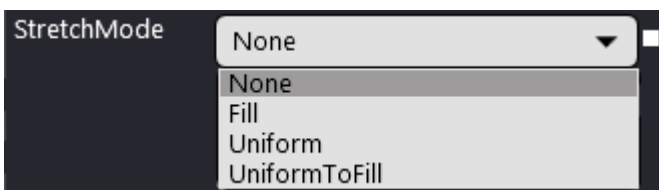
# CREW Manual

Properties	Description
Name	Page ID
Title	Page title
Color	Wallpaper colour of page; editable with RGB colours or colour palette
PageNo	Identifies the page number assigned
EnableImage	Determines whether the page should have a wallpaper image
ImageSource	Specifies the route from which the imported image is to be uploaded
StretchMode	Resizes the grouped elements maintaining their aspect
HorizontalAlign	This function allows to position the image horizontally (Centred, Right or Left)
VerticalAlign	This function allows to position the image vertically (Centred, Top or Bottom)
EnableHelp	Determines whether the help page should have a text message
HelpText	The "Help Text" property is a help message displayed on the help page
Level	Allows to define the levels of authentication required to control access to specific areas of the project

Some of the properties from the table in the image are listed below:

## StretchMode

The "StretchMode" property can be set as follows:

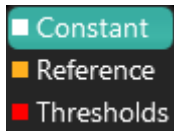




# CREW Manual

- None: the content maintains its original size.
- Fill: the content is resized to fill the target size but the aspect ratio is not maintained.
- Uniform: the content is resized to adjust to the target size while maintaining its original aspect ratio.
- UniformToFill: the content is resized to fill the target size while maintaining its original aspect ratio. If the proportions of the target rectangle are different from the source, the source content is cut to the target size.

It is possible to vary the type of associated value to the properties with the white square next to it, as follows:



- Constant: to associate a constant value.
- Reference: to associate a value to a variable.
- Thresholds: to attribute a value by linking it to the threshold management of the variables (see [Thresholds Management Feature](#)).

## ProtectionLevel

Within a project it is possible to define the levels of authentication to control access to specific areas. Up to ten access levels can be defined, with the lowest level (usually 1) being the one with the greatest degree of operational freedom.

If the user does not log in, he/she is treated by the system as a user of level 10 (lowest level of freedom) and can only access the features allowed for that level. If the non-logged in user tries to perform a task on a level lower than the tenth, the user will be asked to login again through an appropriate pop-up page predefined by the system.

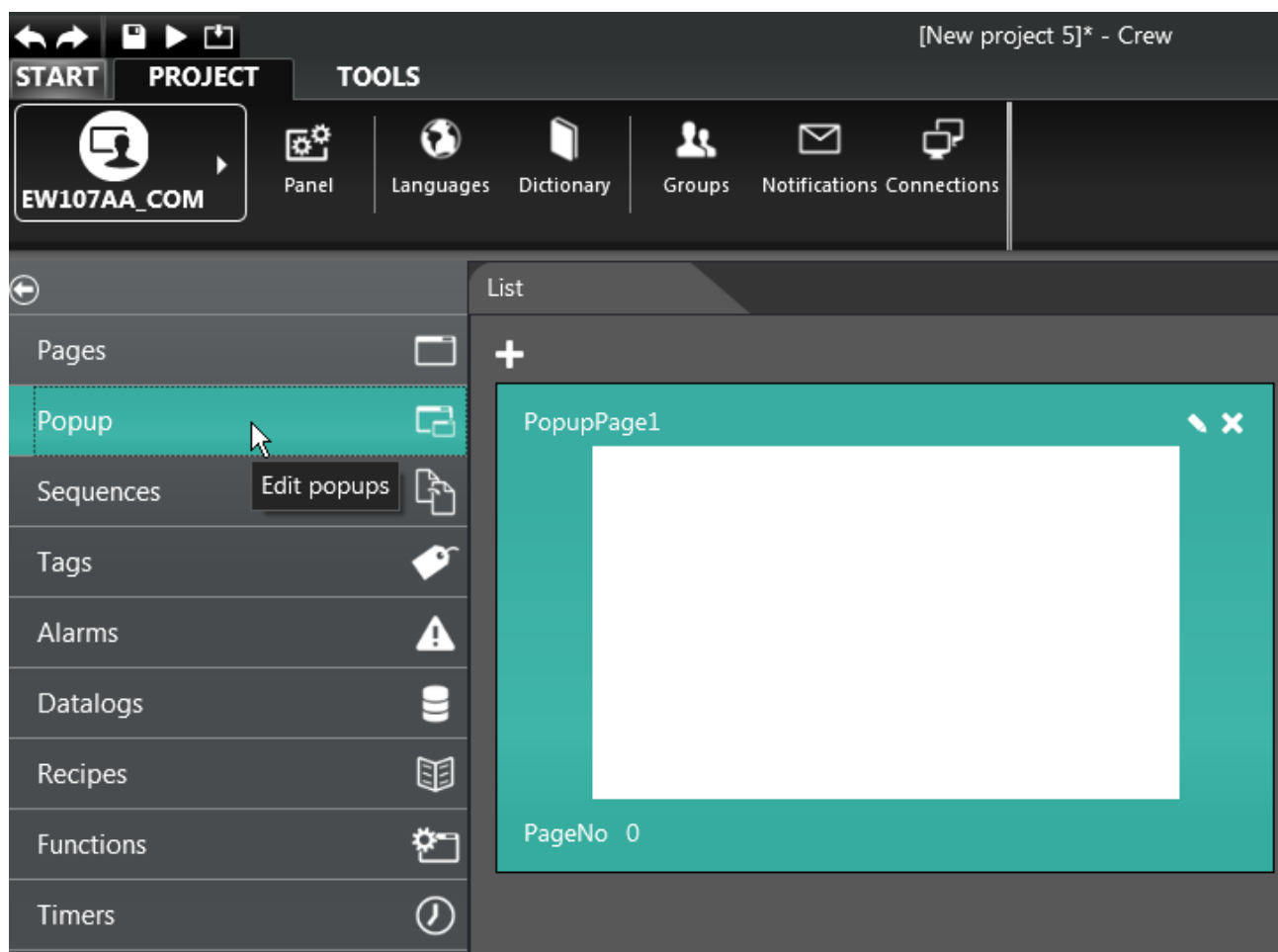
Crew defines the levels of initial users, or rather those who will be present at project startup. It is also possible to add or change users directly in Runtime. For this purpose it is possible to enter a default control for user management in the pages.

# CREW Manual

## Popup

The "PopUpPages" only appear after particular situations (command area, button with related function).

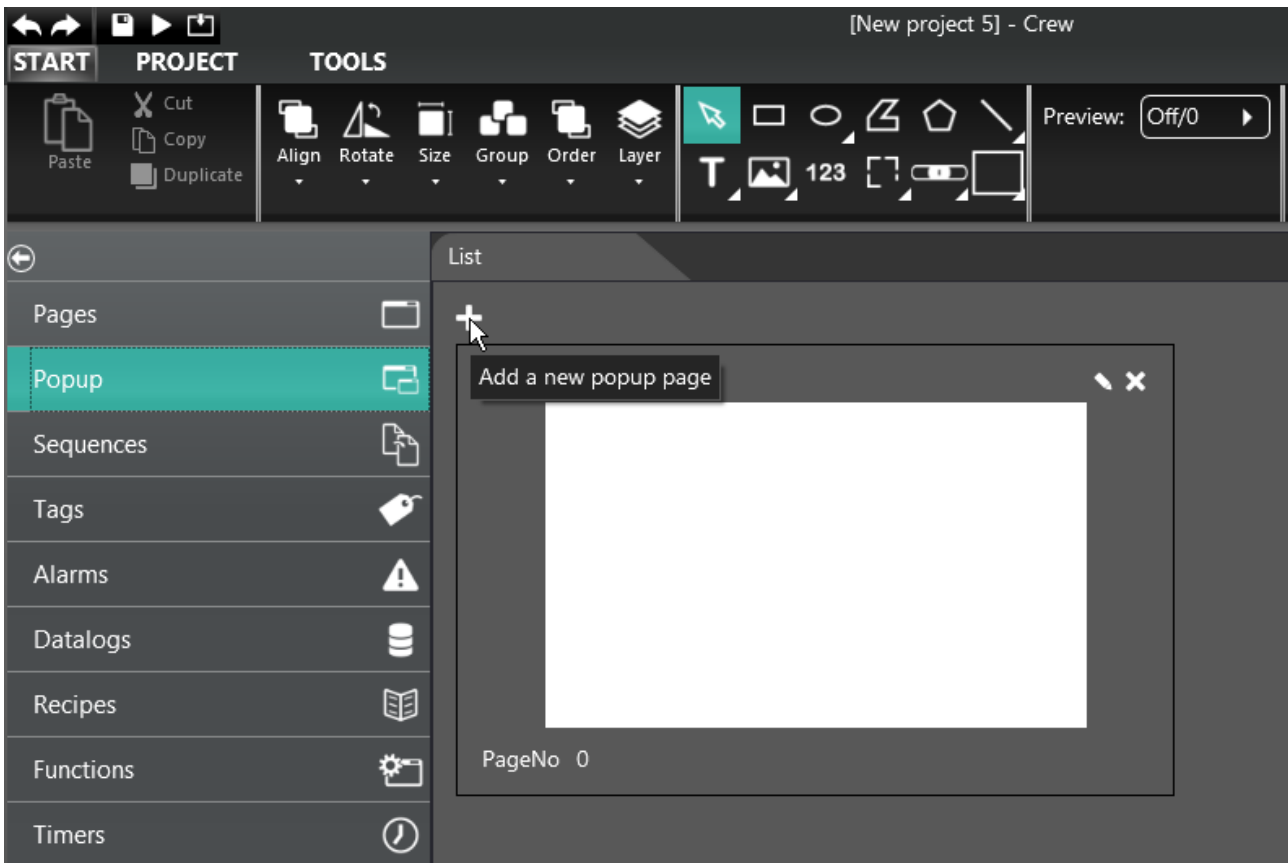
Click the "Popup" menu to open the list of the project's popup pages in the work area.



# CREW Manual

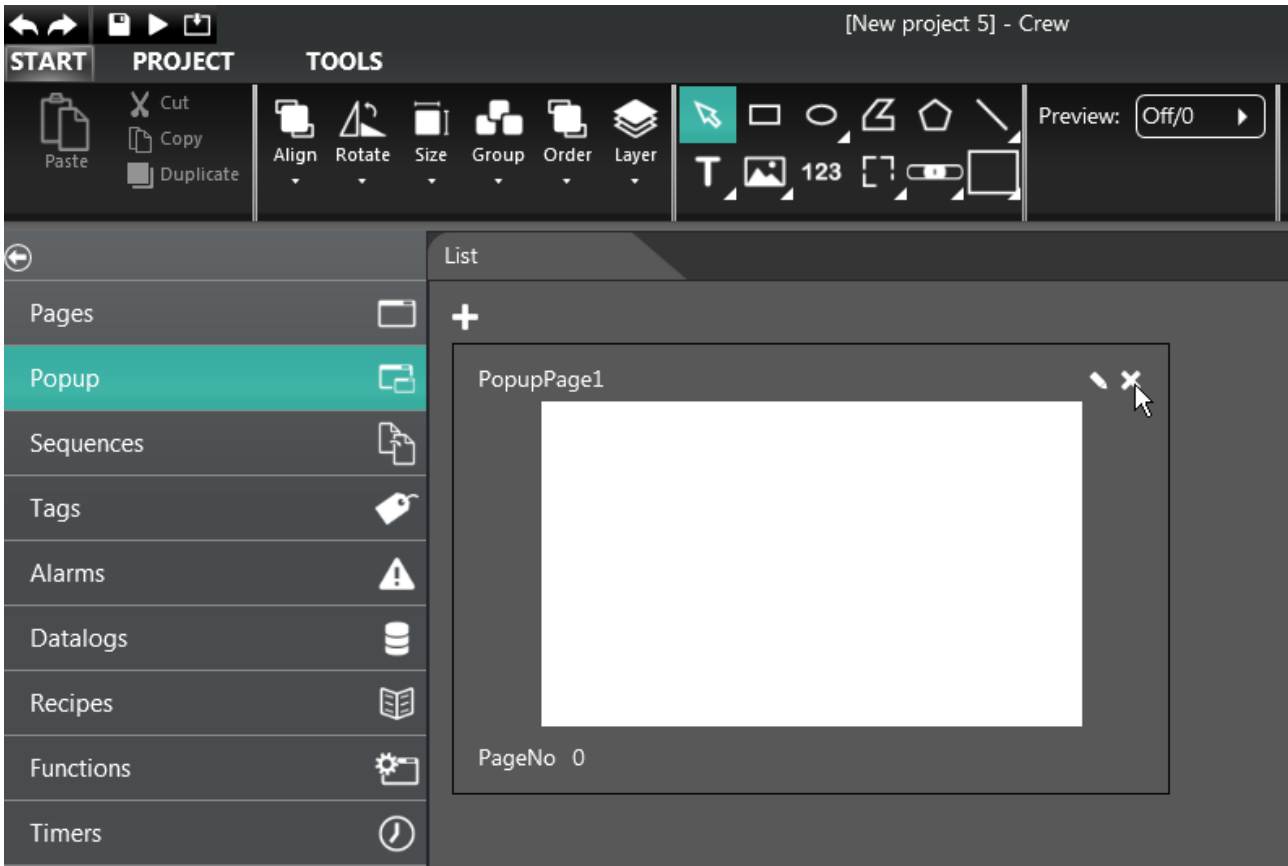
From this list it is possible to do the following operations.

Enter new pages.



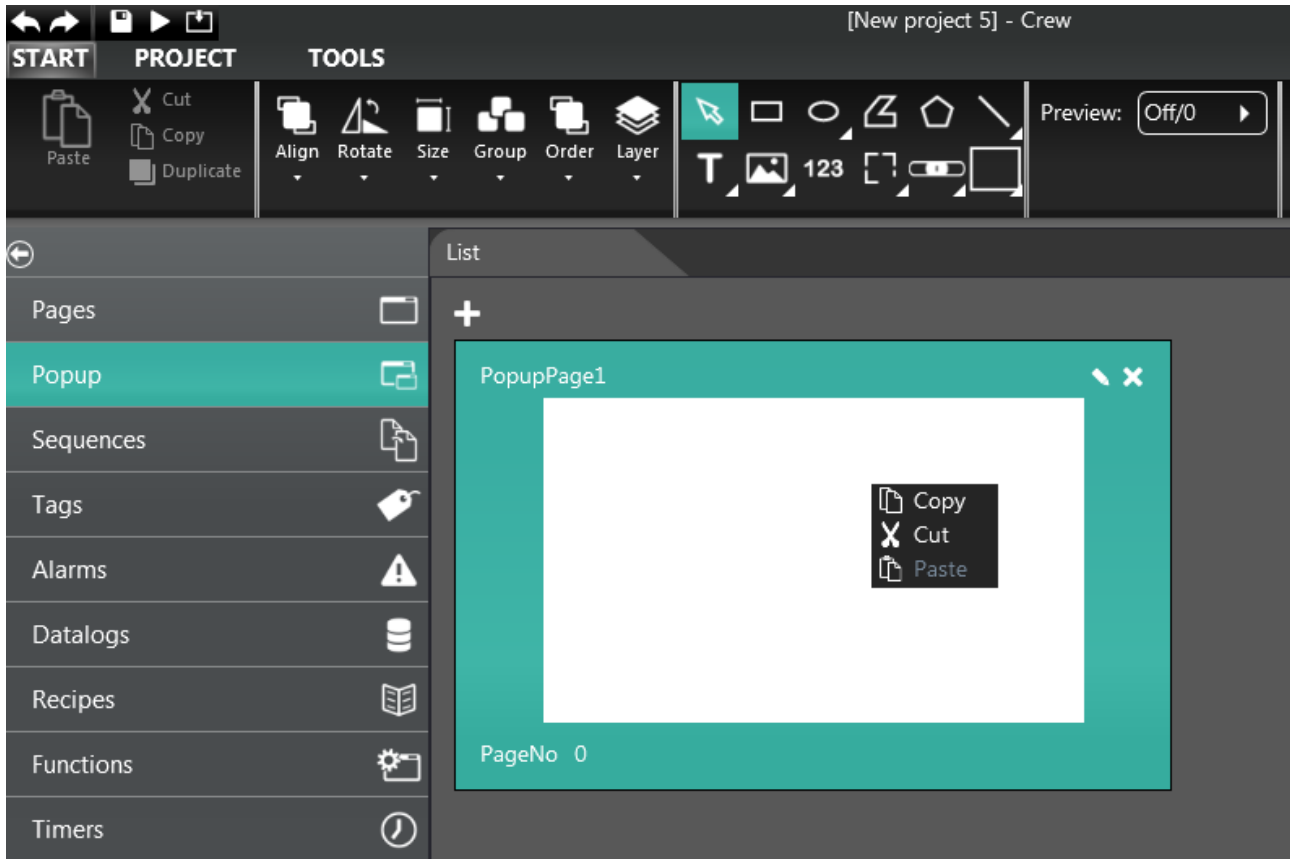
# CREW Manual

After selecting the pages, delete them.



# CREW Manual

Or, by right clicking the mouse, duplicate the existing pages (cut/copy/paste).

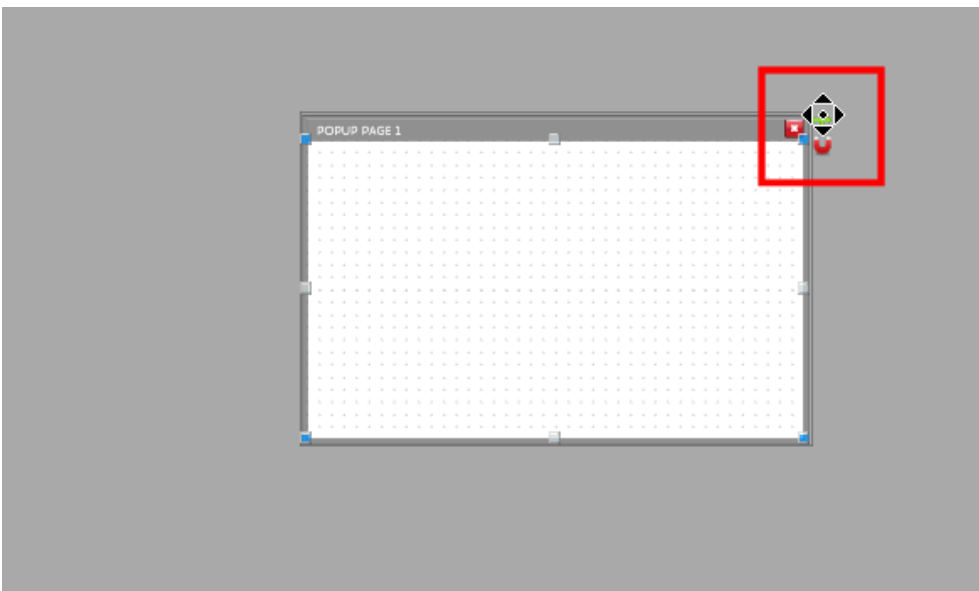
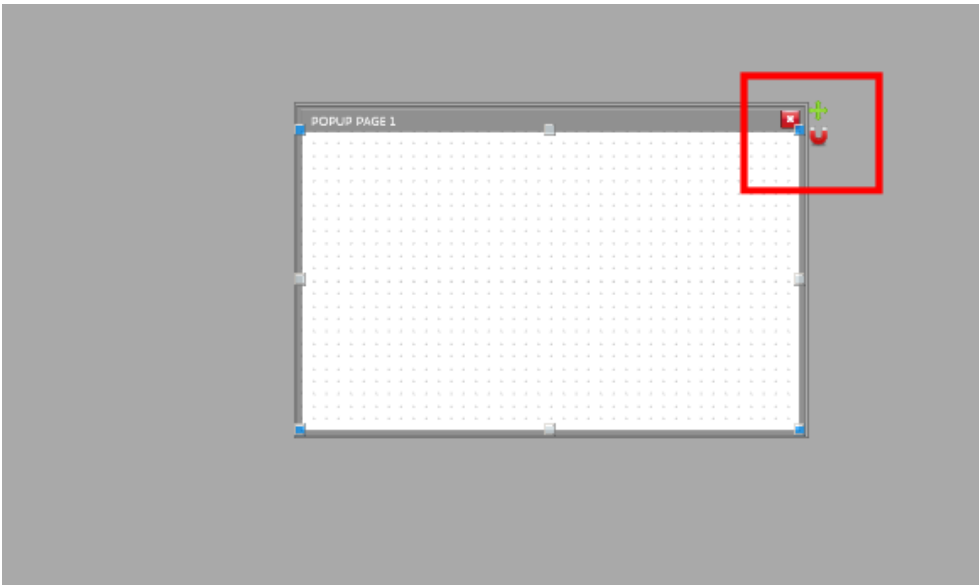


Once a page has been created, double-click it to edit it in the work area. The page editor is described in the "[Popup page management](#)" section. It is also possible to edit the properties of the Popup page (see "[Popup page Properties](#)" section) and link events to them (see "[Popup page Events](#)" section).

# CREW Manual

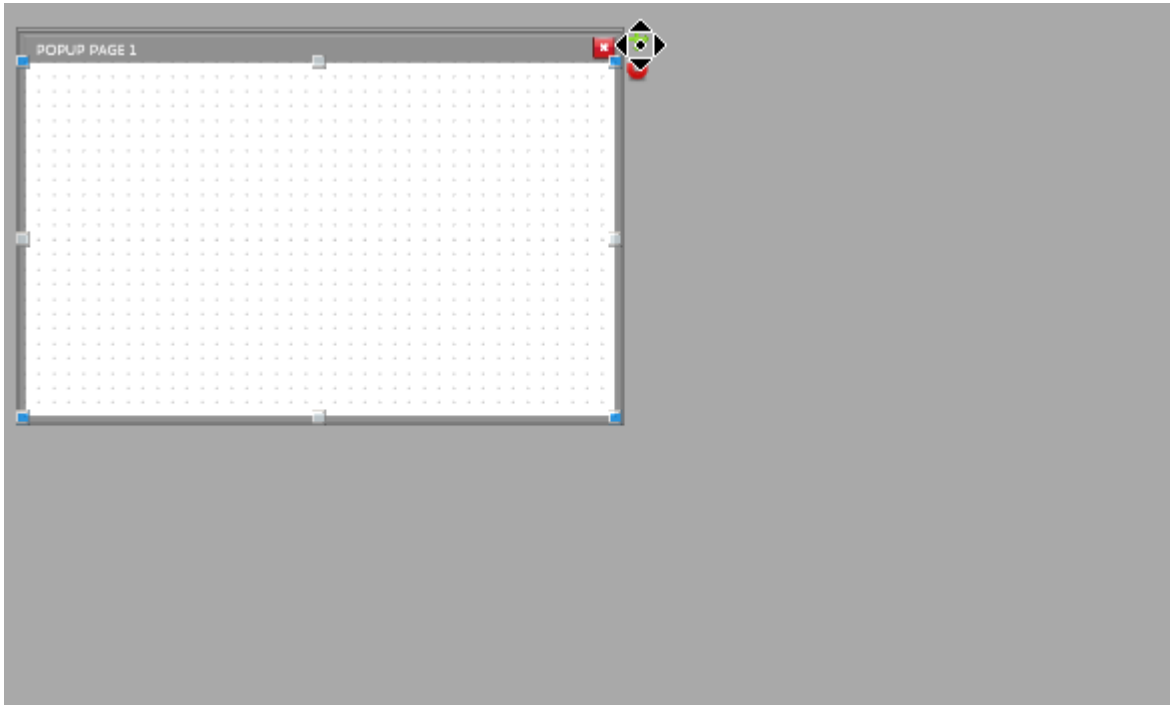
## Pop-up page management

Select the key at the top right of the popup page to enable movement so that it can be moved.



# CREW Manual

Drag the popup page to the position where you want it to appear in Runtime.



In the next chapter all the procedures for entering graphic objects, with their meanings and tools, will be illustrated.

In order for a graphic element, navigation or function button, command, view/edit data field, to be visible to the operator in Runtime, it needs to be placed on a page.

The "Zoom" commands are at the bottom of the screen.

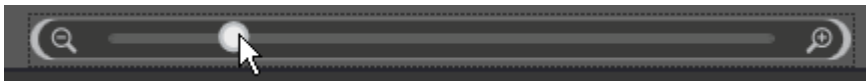


# CREW Manual

Click the drop down menu and choose from the available Zoom options to change the size of the displayed page.



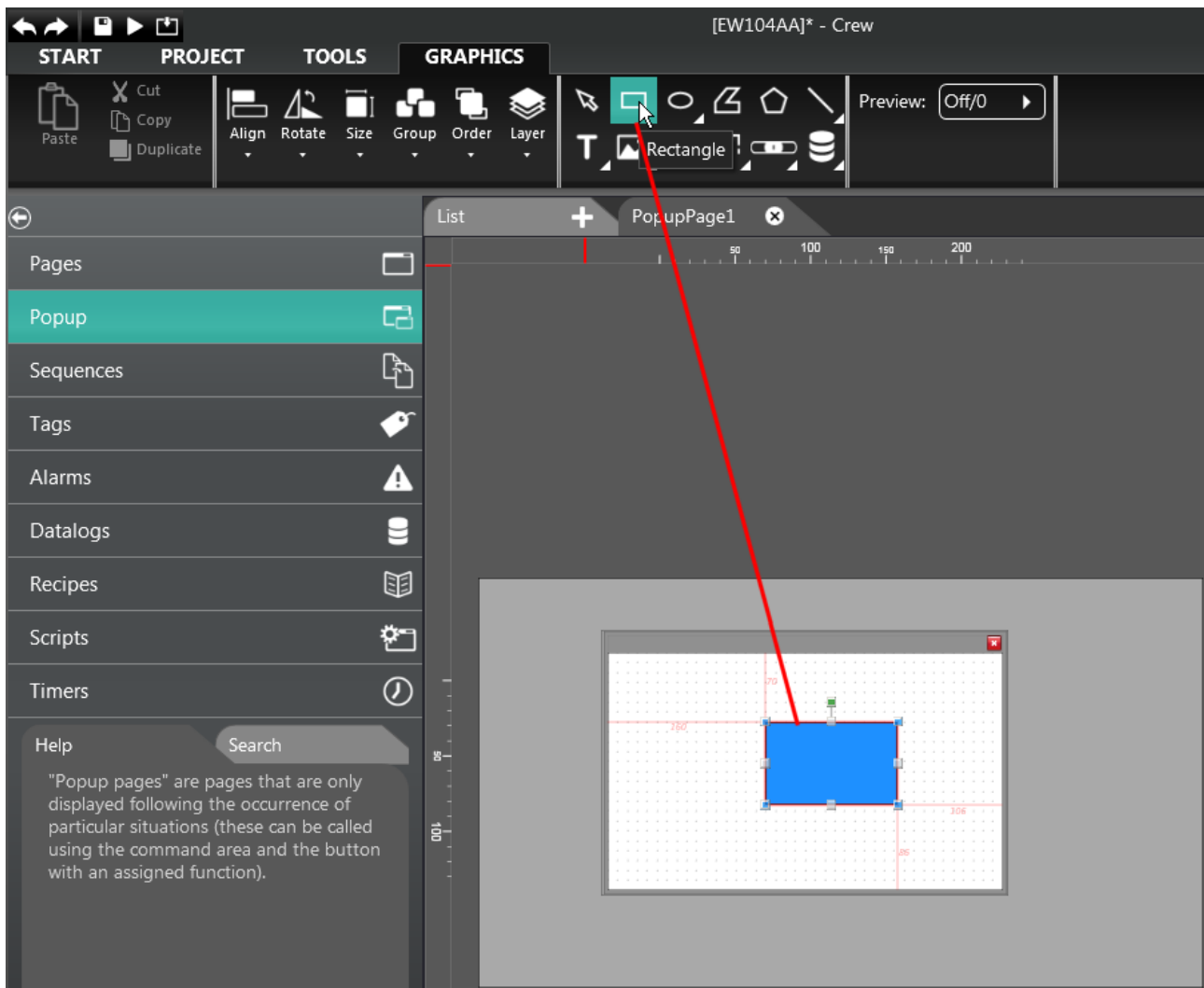
Or you can zoom with the zoom slide.





# CREW Manual

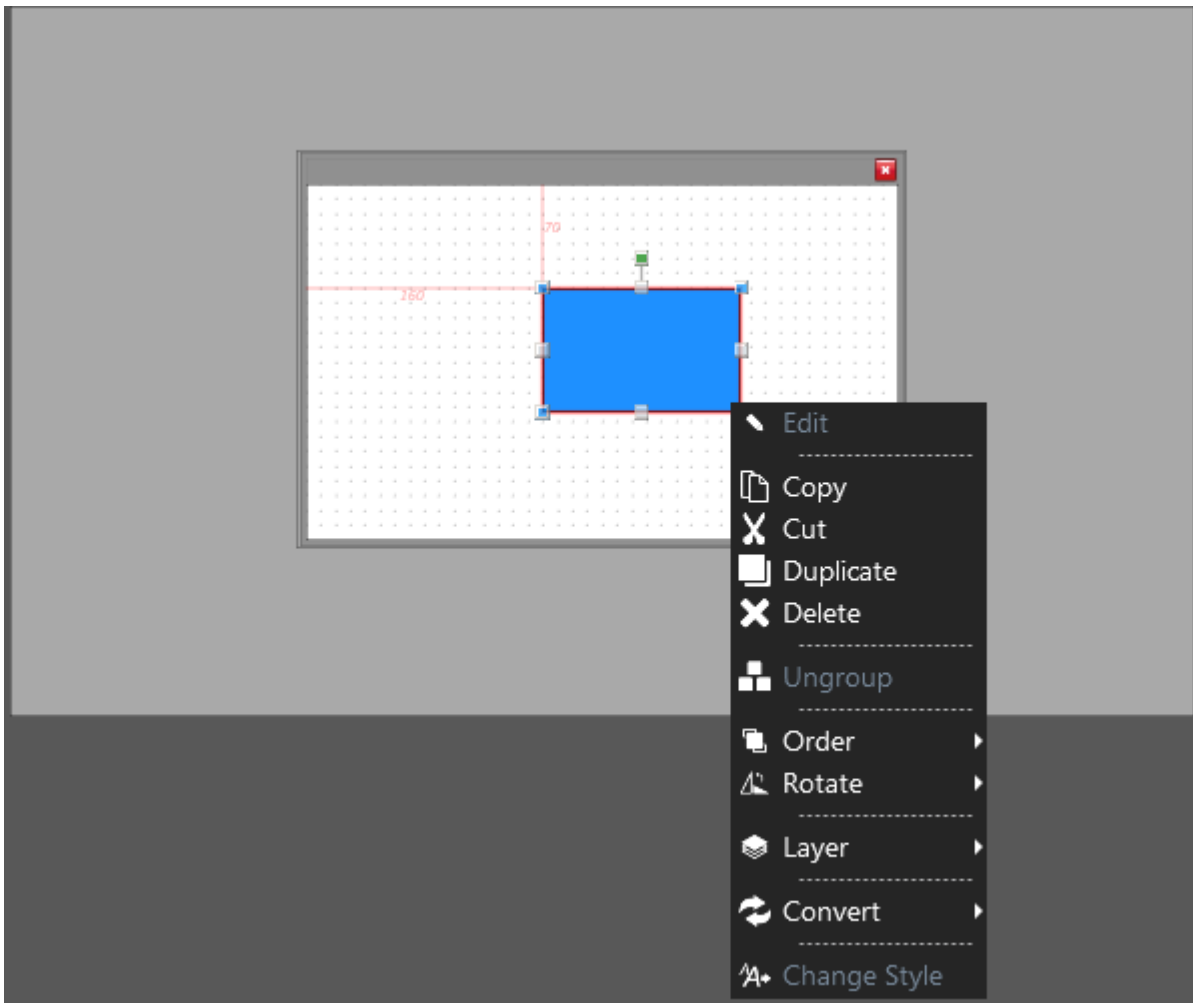
To enter an object on the page click its icon in the toolbar and trace the outline in the desired position on the preview page.



When it is entered, the element appears on the page and can be selected with a click. For each selected object, all of the user-settable items appear in the "Property Editor" and "Event Editor" (see the "[Popup Page Properties](#)" and "[Popup Page Events](#)" sections).

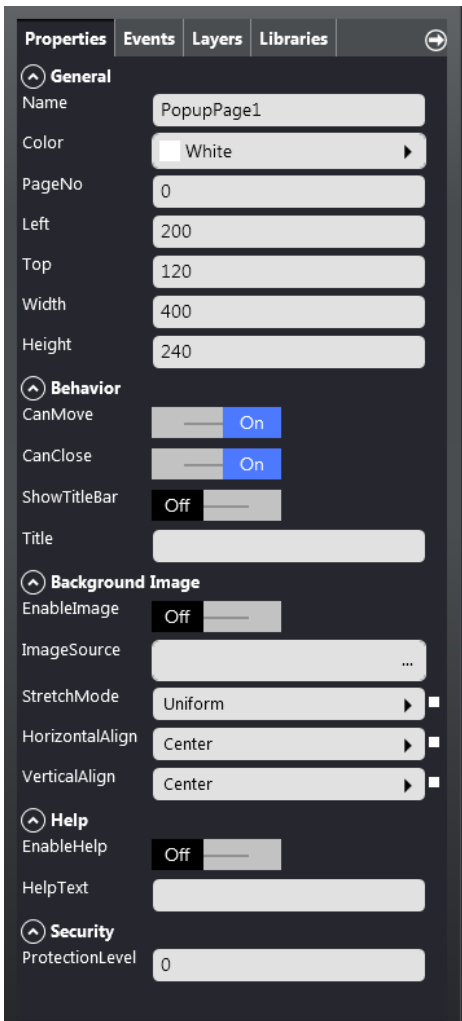
# CREW Manual

Right click a selected object to access a menu with the following functions: "Edit", "Copy", "Cut", "Duplicate", "Delete", "Order" (see "[Order Submenu](#)"), "Rotate" (see "[Rotate Submenu](#)"), "Layer" (see "[Layers](#)" and "Conversion").



# CREW Manual

## Pop-up page properties



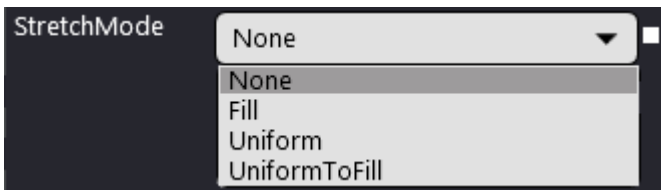
Properties	Description
<b>Name</b>	Page ID
<b>Color</b>	Wallpaper colour of page; editable with RGB colours or colour palette
<b>PageNo</b>	Identifies the page number assigned
<b>Left</b>	Horizontal coordination of position.
<b>Top</b>	Vertical coordination of position.
<b>Width</b>	Width.
<b>Height</b>	Height.
<b>CanMove</b>	Enabling "CanMove" allows to move the Pop-up page at Runtime
<b>CanClose</b>	Enabling "CanClose" allows to delete the "X" from the Title Bar
<b>TitleBar</b>	Enabling "TitleBar" allows to display or hide the Title Bar
<b>Title</b>	Page title
<b>EnableImage</b>	Determines whether the page should have a wallpaper image
<b>ImageSource</b>	Specifies the route from which the imported image is to be uploaded.
<b>StretchMode</b>	Resizes the grouped elements maintaining their aspect
<b>HorizontalAlign</b>	This function allows to position the image horizontally (Centred, Right or Left)
<b>VerticalAlign</b>	This function allows to position the image vertically (Centred, Top or Bottom)
<b>EnableHelp</b>	Determines whether the help page should have a text message.
<b>HelpText</b>	The "Help Text" property is a help message displayed on the help page
<b>Level</b>	Allows to define the levels of authentication required to control access to specific areas of the project

# CREW Manual

Some of the properties from the table in the image are listed below.

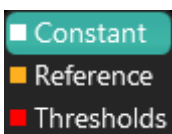
## StretchMode

The "StretchMode" property can be set as follows:



- None: the content maintains its original size.
- Fill: the content is resized to fill the target size but the aspect ratio is not maintained.
- Uniform: the content is resized to adjust to the target size while maintaining its original aspect ratio.
- UniformToFill: the content is resized to fill the target size while maintaining its original aspect ratio. If the proportions of the target rectangle are different from the source, the source content is cut to the target size.

It is possible to vary the type of associated value to the properties with the white square next to it, as follows:



- Constant: to associate a constant value.
- Reference: to associate a value to a variable.
- Thresholds: to attribute a value by linking it to the threshold management of the variables (see [Thresholds Management Feature](#)).

# CREW Manual

## ProtectionLevel

Within a project it is possible to define the levels of authentication to control access to specific areas. Up to ten access levels can be defined, with the lowest level (usually 1) being the one with the greatest degree of operational freedom.

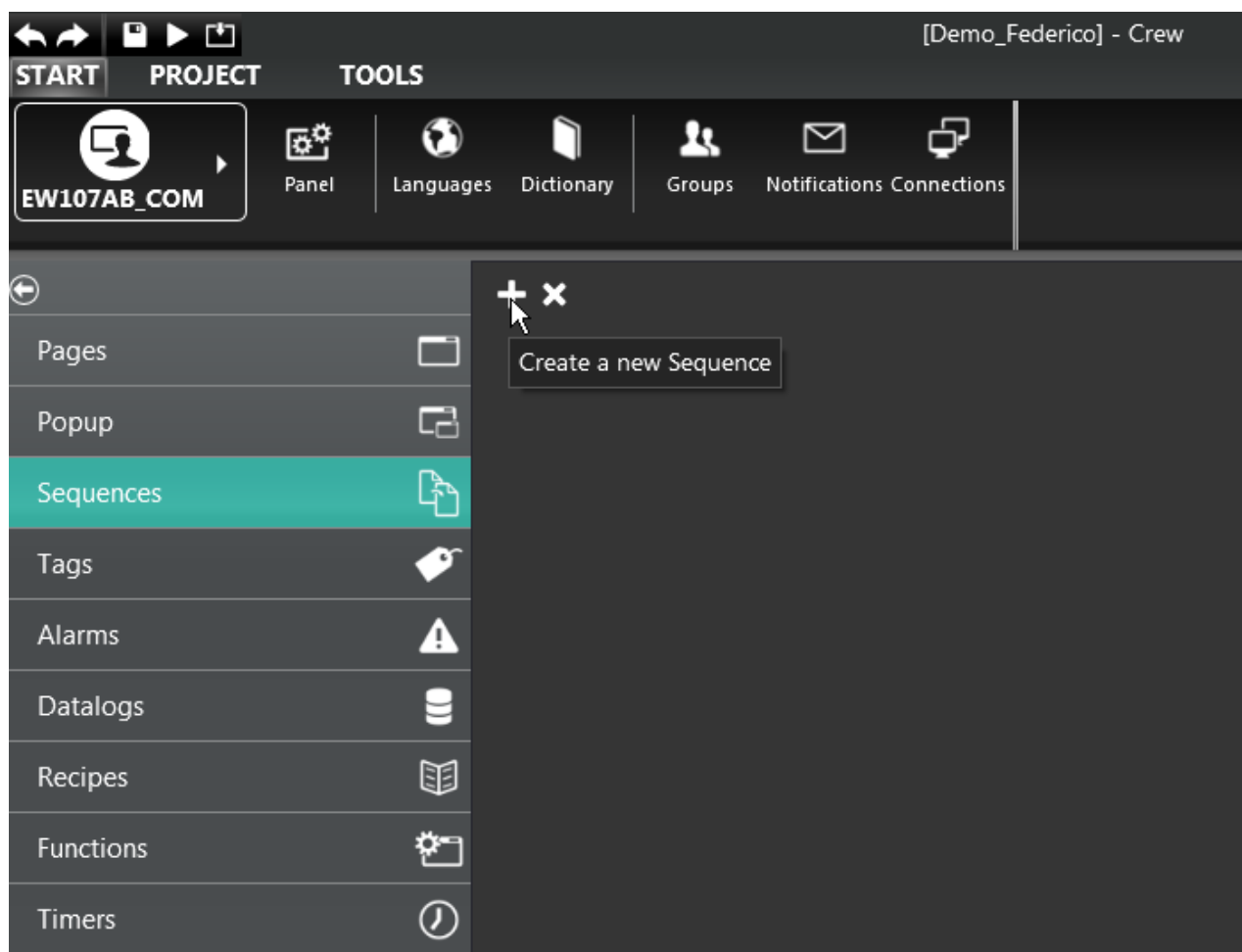
If the user does not log in, he/she is treated by the system as a user of level 10 (lowest level of freedom) and can only access the features allowed for that level. If the non-logged in user tries to perform a task on a level lower than the tenth, the user will be asked to login again through an appropriate pop-up page predefined by the system.

Crew defines the levels of initial users, or rather those who will be present at project startup. It is also possible to add or change users directly in Runtime. For this purpose it is possible to enter a default control for user management in the pages.

# CREW Manual

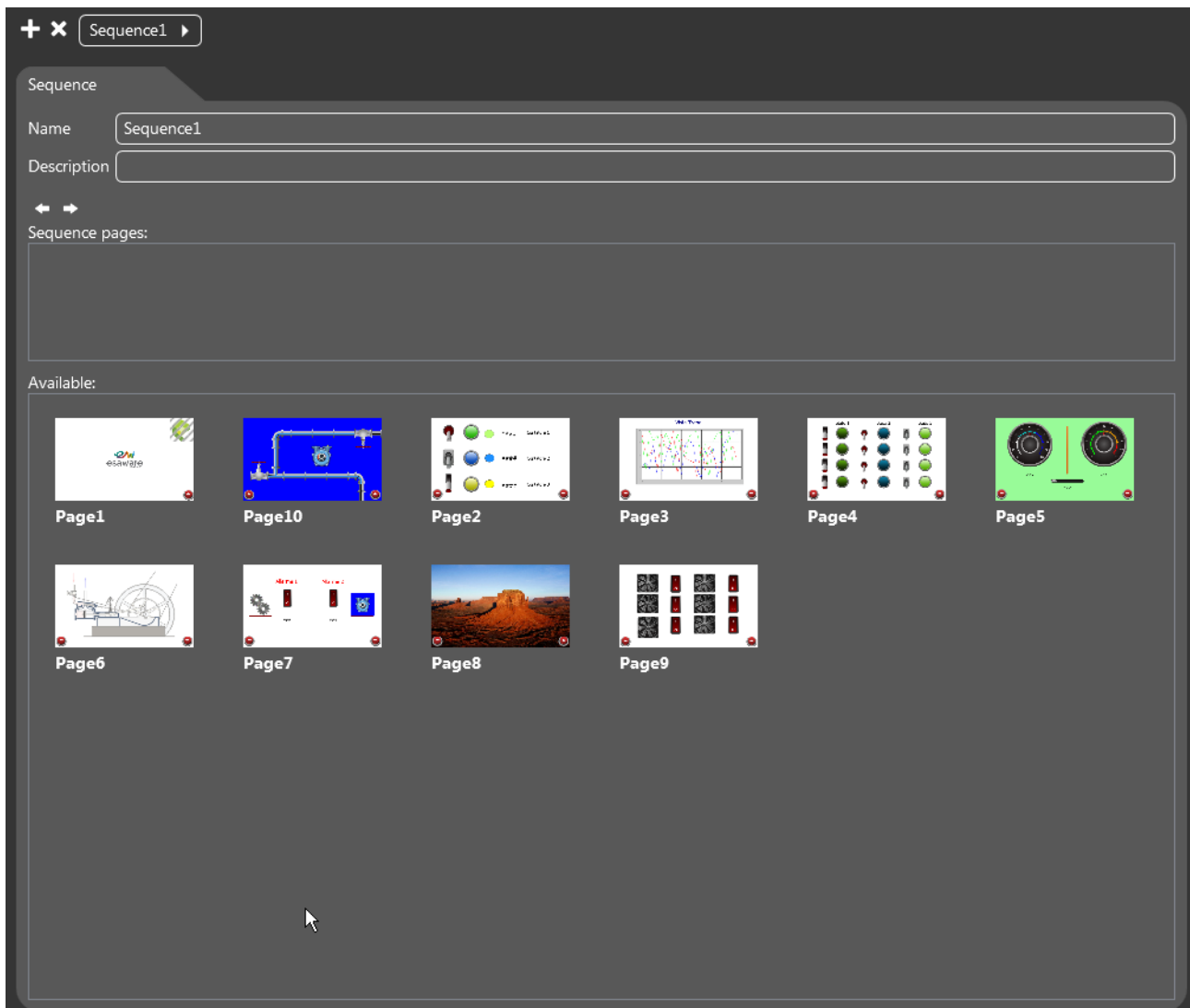
## Sequences

Click the "Sequences" menu to open the list of the project's sequences pages in the work area. From this list it is possible to add new sequences.



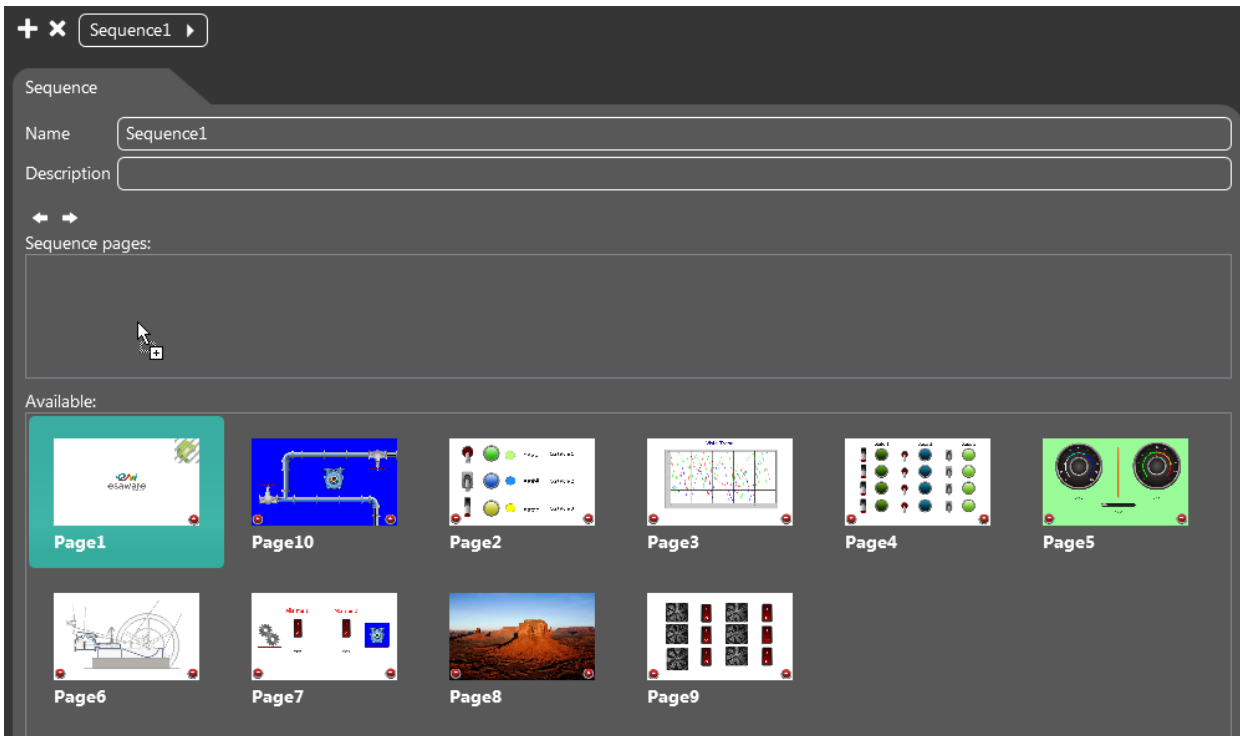
To create a sequence of pages, it is necessary to have previously added said pages to the project. From the created pages, the user can now select which ones need to form the sequence in Runtime.

# CREW Manual

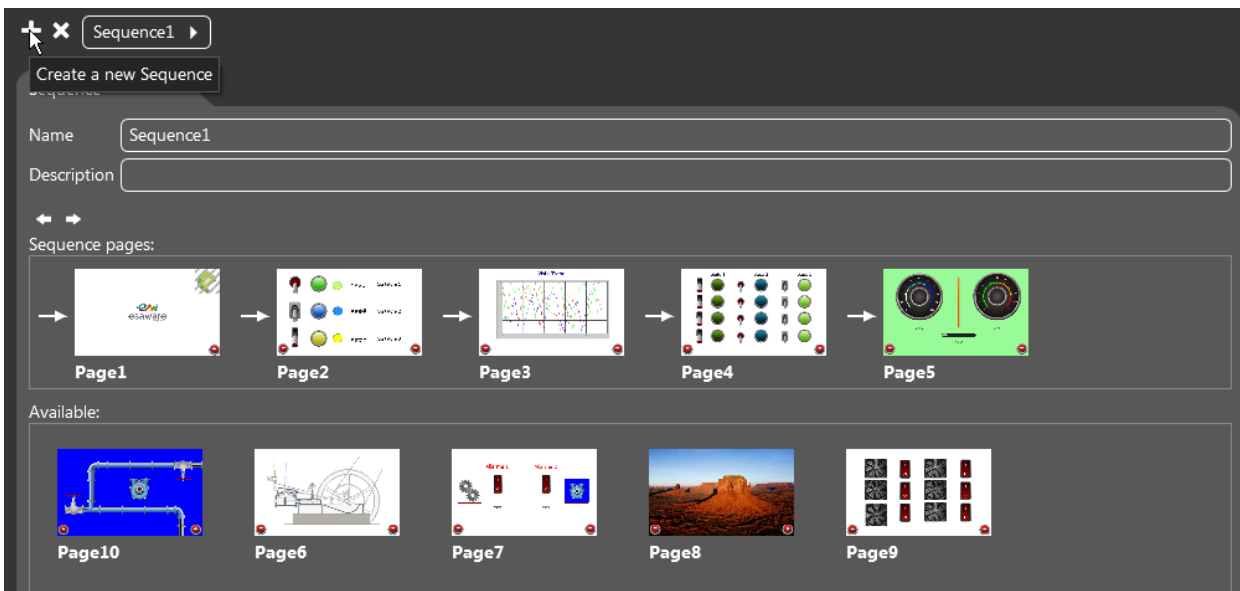


For example, to create a sequence with pages 1, 2, 3, 4 and 5, simply select these pages and use the mouse to drag them one at a time to the “Page sequence” area.

# CREW Manual



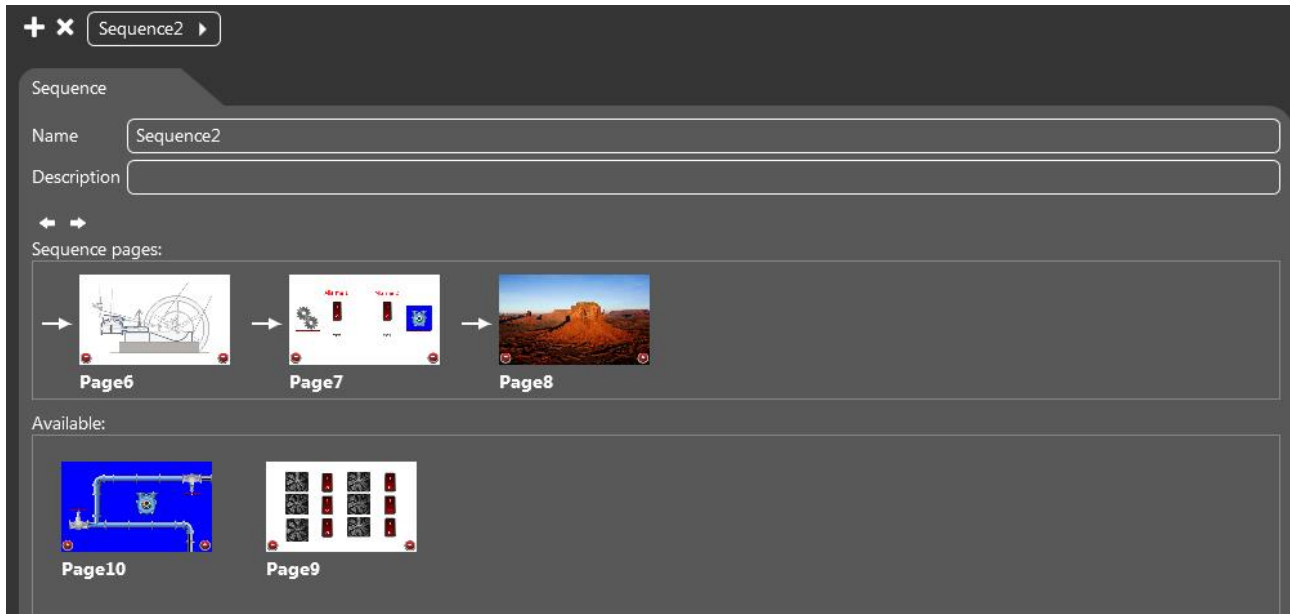
To create other sequences, click the appropriate key.





# CREW Manual

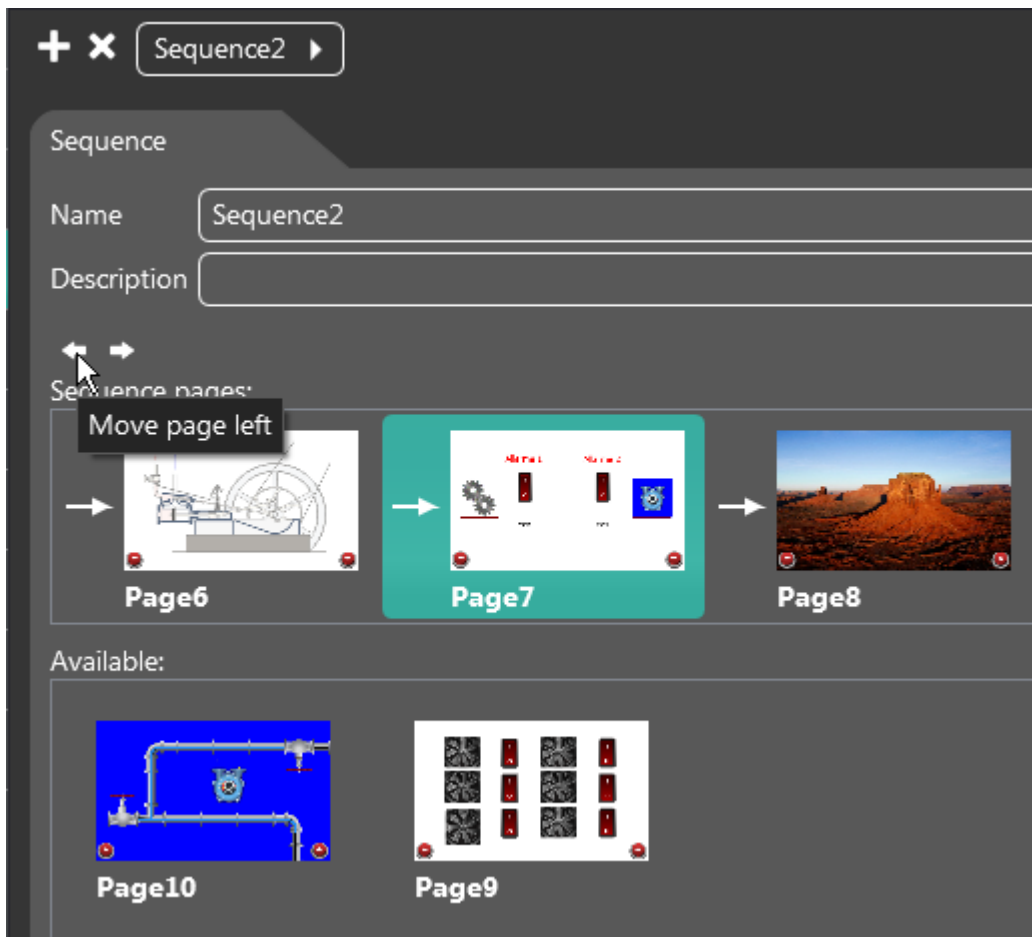
This time, in the example, pages 6, 7 and 8 are selected and dragged.



Accordingly, two different sequences are obtained.

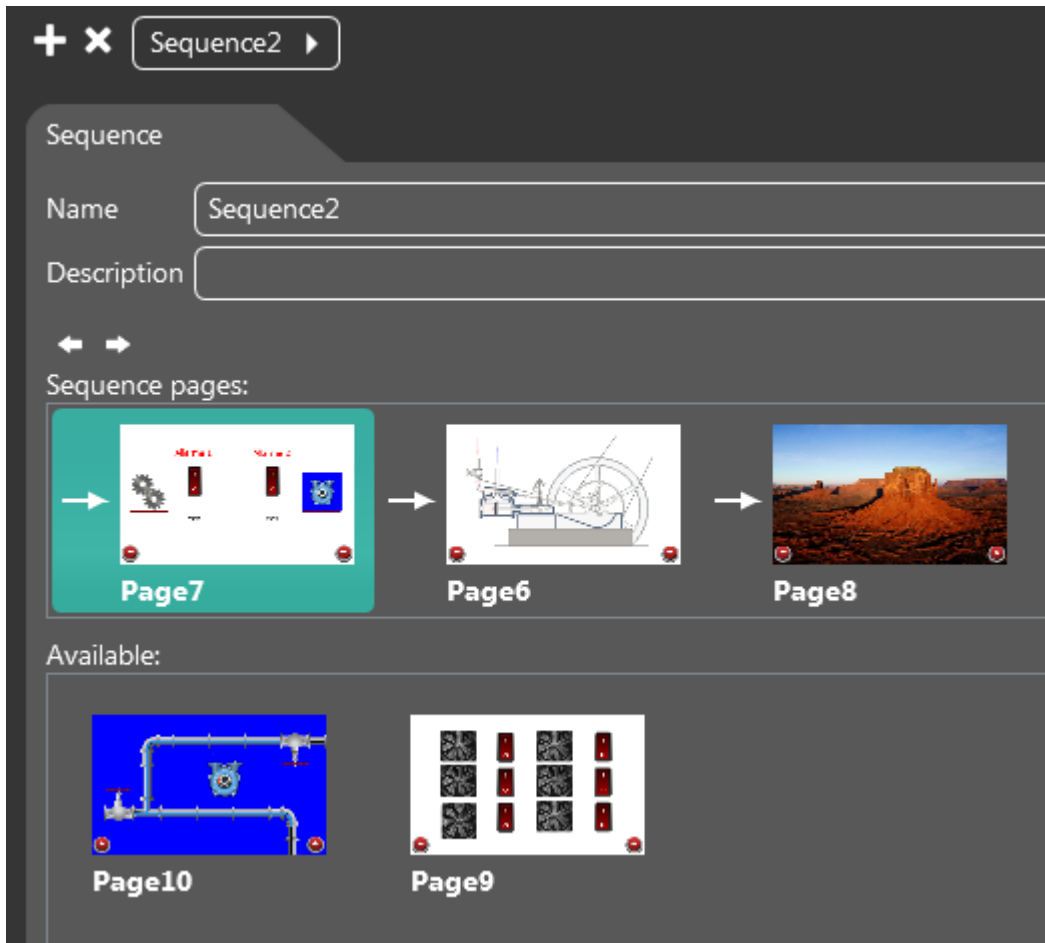
# CREW Manual

To move a page before or after in an existing sequence (for example, page 7 before page 6) simply use the mouse to select the page you wish to move ("Page7" in the example) and click on the left arrow.



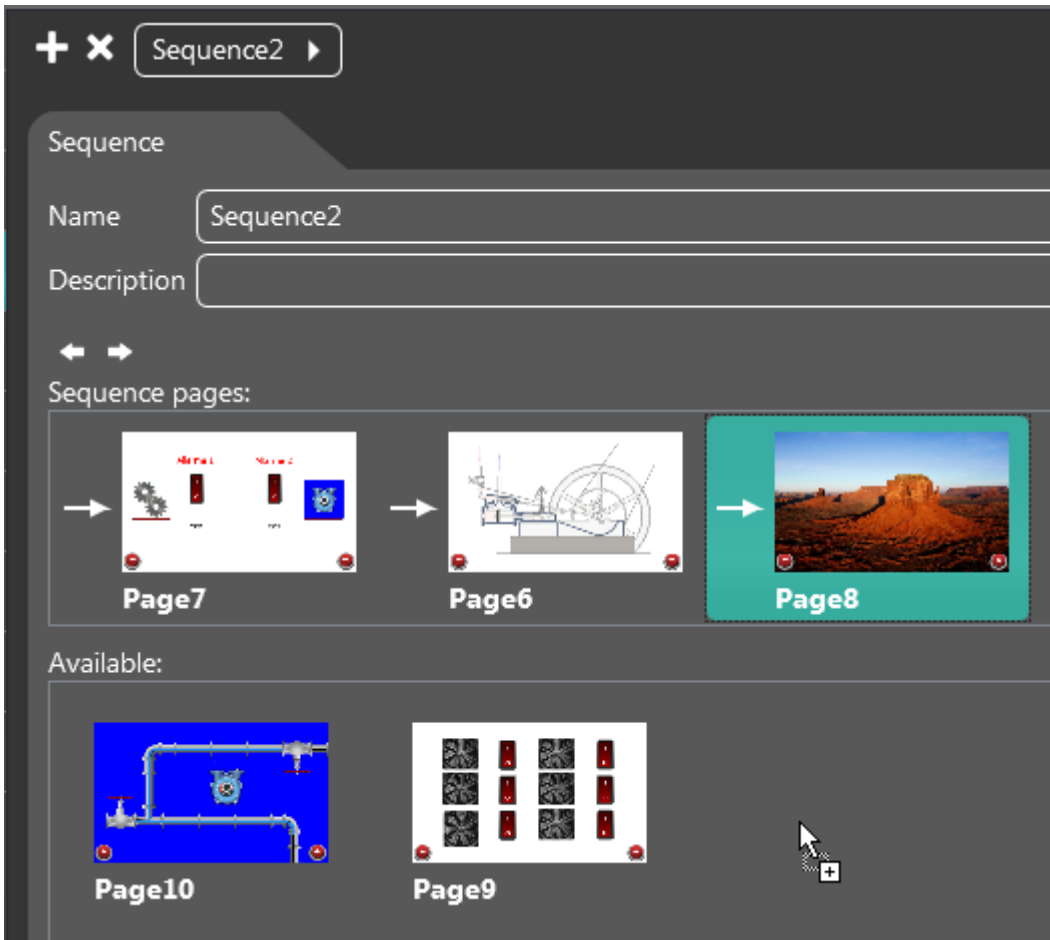
Now "Page7" will be before "Page6".

# CREW Manual

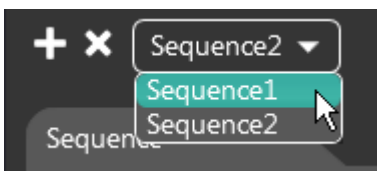


# CREW Manual

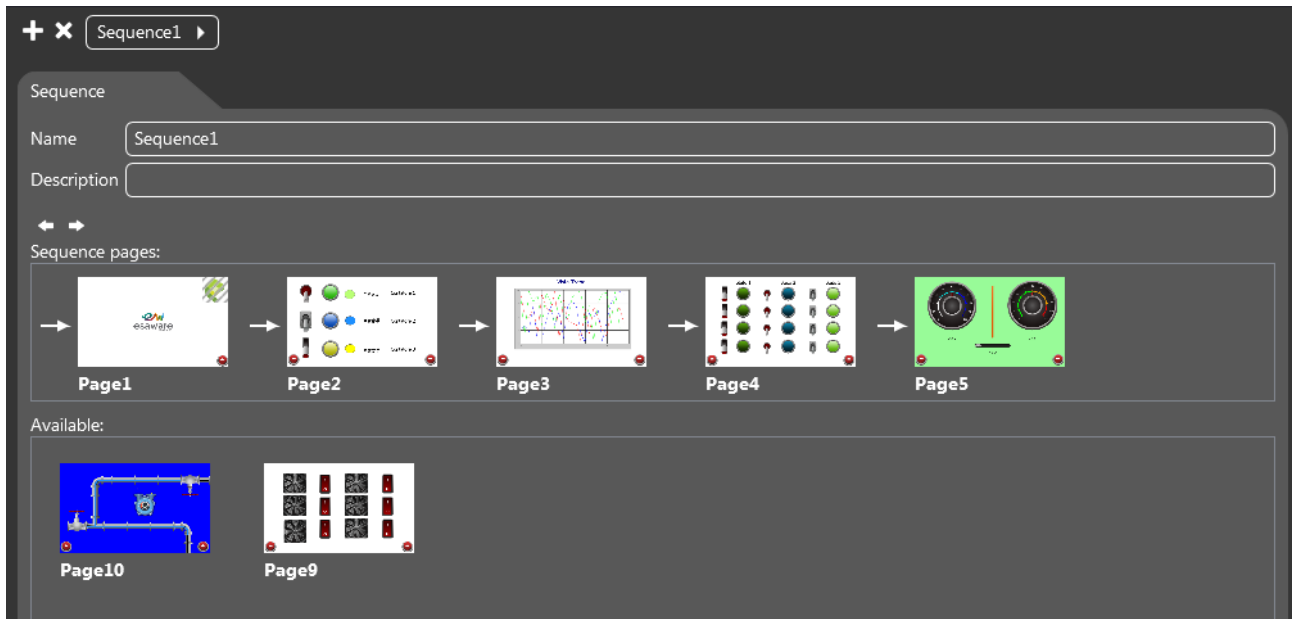
To delete a page from the sequence, simply select it and drag it back to the available pages.



To switch to the view of another created sequence (for example, from Sequence2 to Sequence 1), you need to select it from the menu shown in the image.

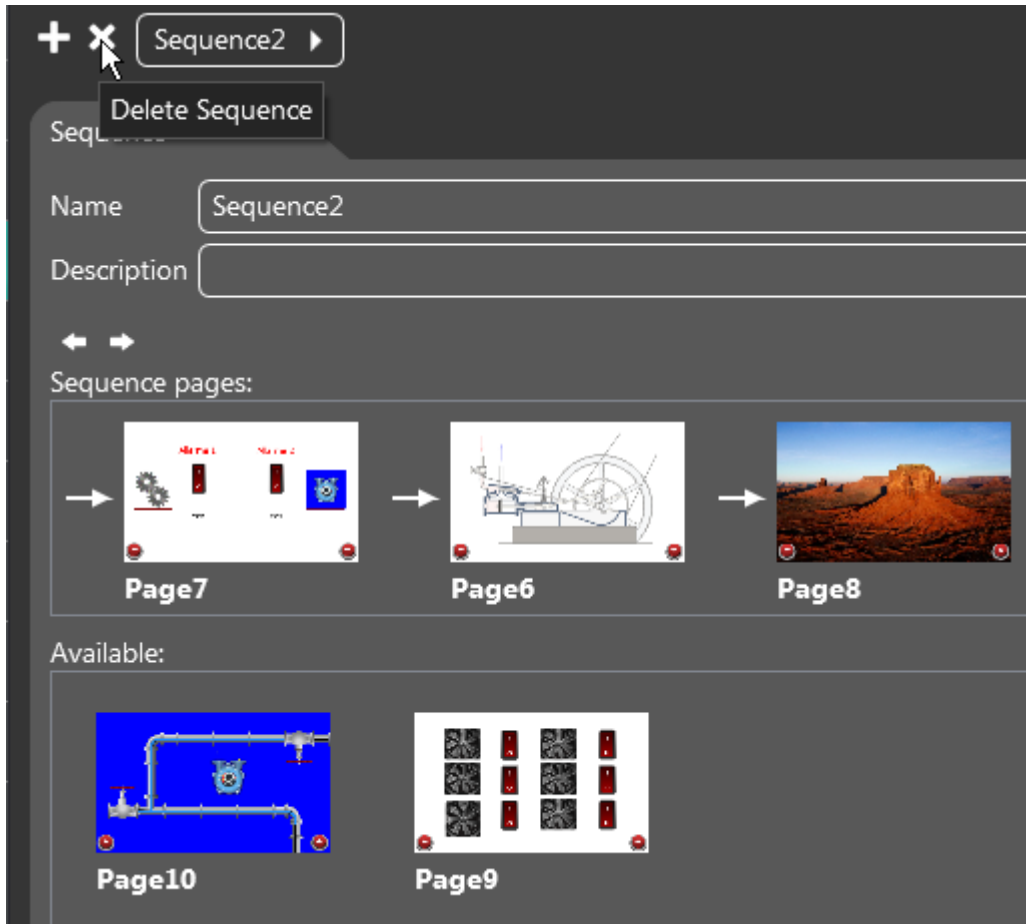


# CREW Manual



# CREW Manual

To delete a sequence simply select it from the available ones and click the “Delete sequence” (“X”) icon.



Click “Yes” to confirm deleting the sequence.

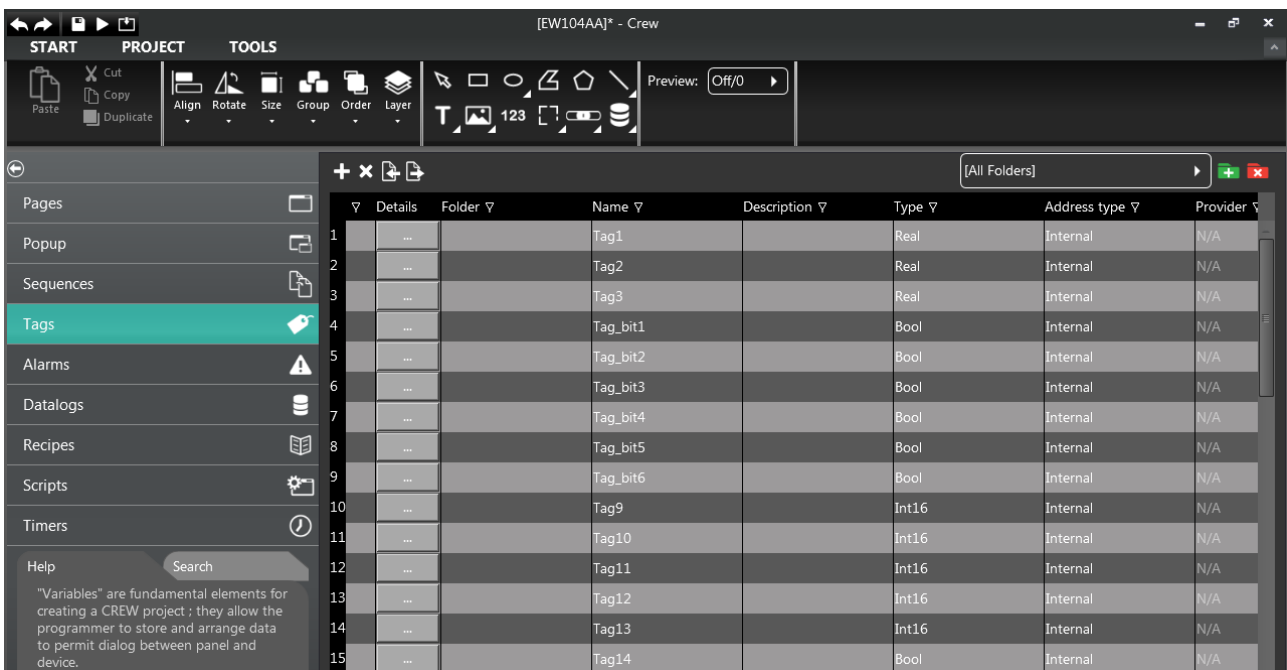
# CREW Manual

## Tags

Tags are key elements for creating a project with Crew, as they allow the programmer to collect the data to allow dialogue between panel and device.

The maximum number of tags that the user can create depends on the memory available on the device.

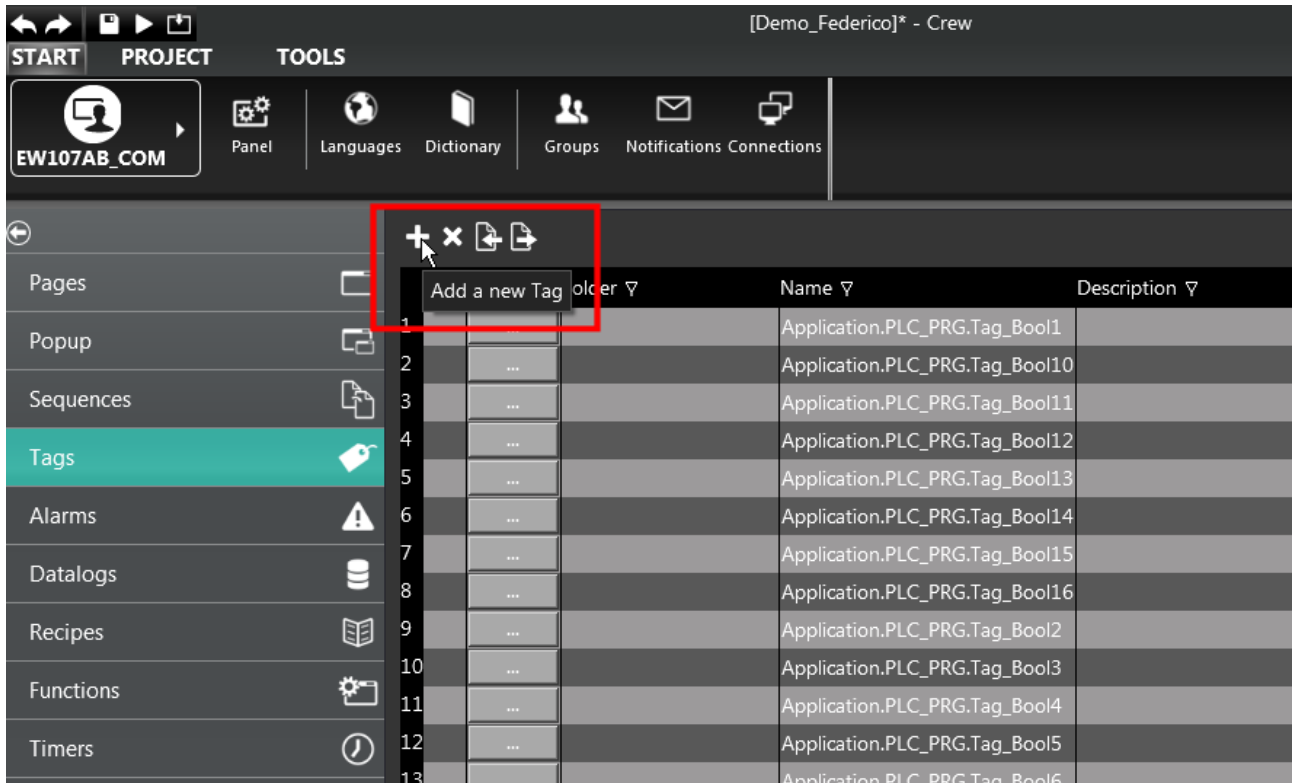
Click the "Tags" menu to open the project's list of variables in the work area.



From this list it is possible to do the following operations.

# CREW Manual

Enter new tags.



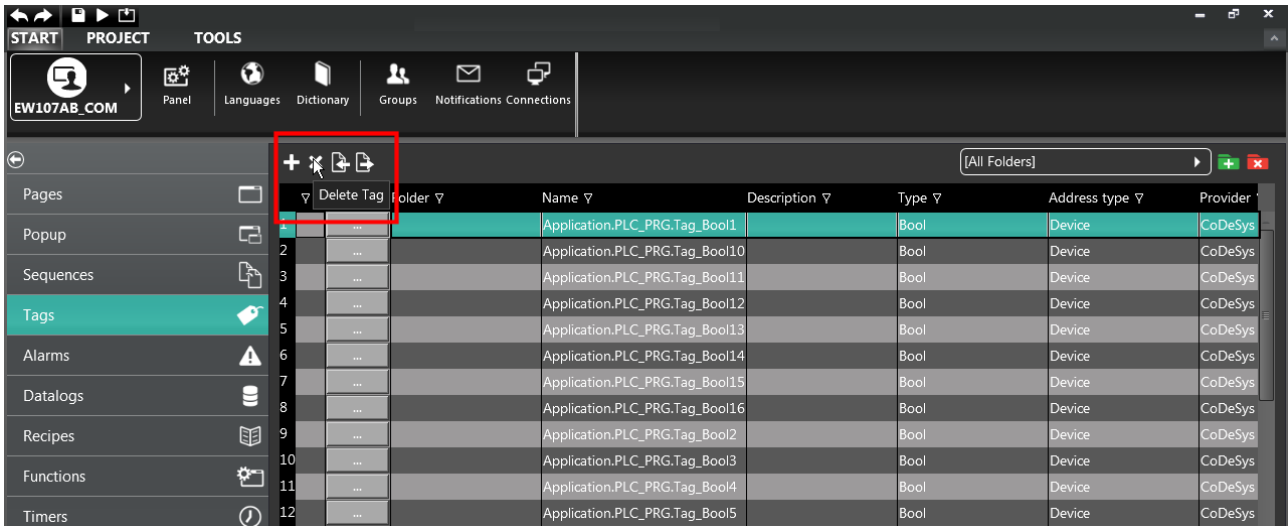
The screenshot shows the software interface with the 'Tags' section selected in the left sidebar. A table of existing tags is displayed, and a red box highlights the 'Add a new Tag' button in the top-left corner of the table.

	Add a new Tag	olcer ▾	Name ▾	Description ▾
1			Application.PLC_PRG.Tag_Bool1	
2	...		Application.PLC_PRG.Tag_Bool10	
3	...		Application.PLC_PRG.Tag_Bool11	
4	...		Application.PLC_PRG.Tag_Bool12	
5	...		Application.PLC_PRG.Tag_Bool13	
6	...		Application.PLC_PRG.Tag_Bool14	
7	...		Application.PLC_PRG.Tag_Bool15	
8	...		Application.PLC_PRG.Tag_Bool16	
9	...		Application.PLC_PRG.Tag_Bool2	
10	...		Application.PLC_PRG.Tag_Bool3	
11	...		Application.PLC_PRG.Tag_Bool4	
12	...		Application.PLC_PRG.Tag_Bool5	
13	...		Application.PLC_PRG.Tag_Bool6	

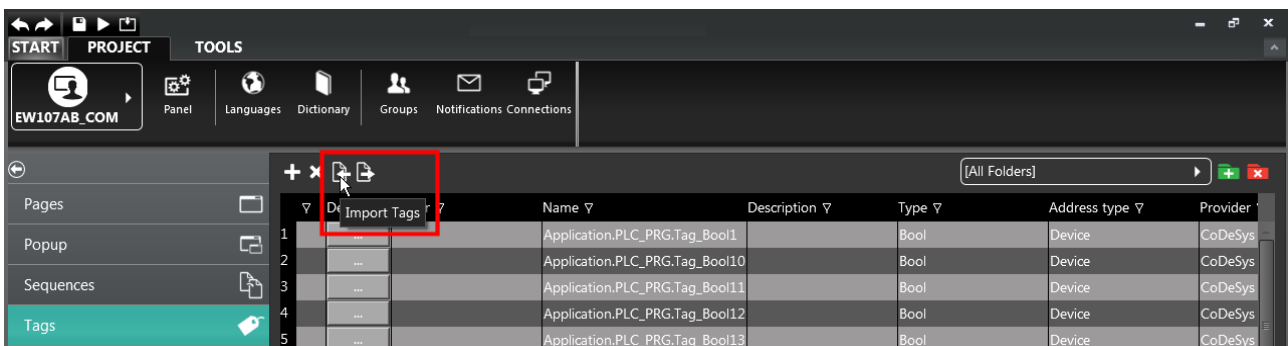


# CREW Manual

Delete the selected tag.

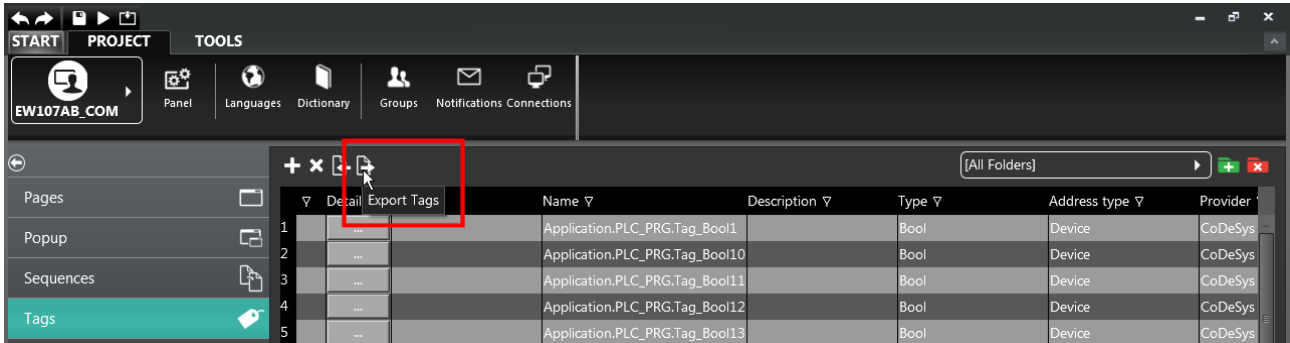


Import a set of previously created variables.

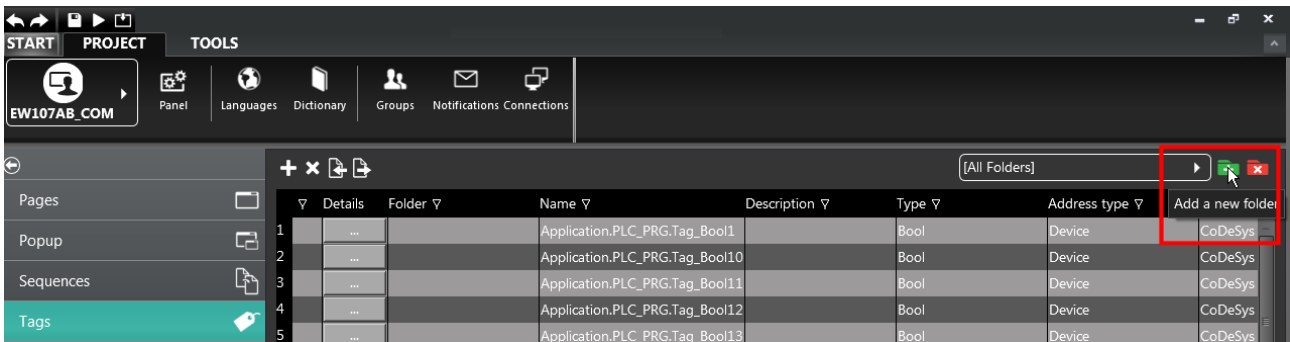


# CREW Manual

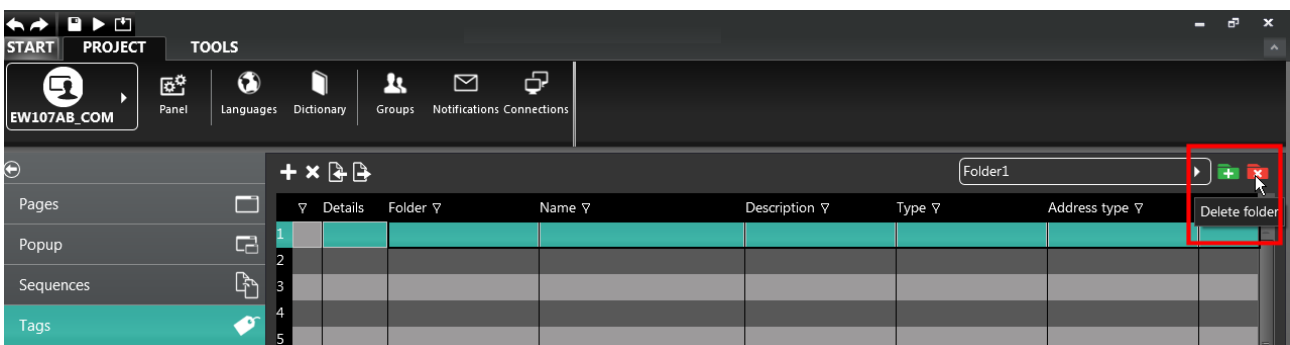
Export a set of previously created variables.



Create new folders to contain tags.



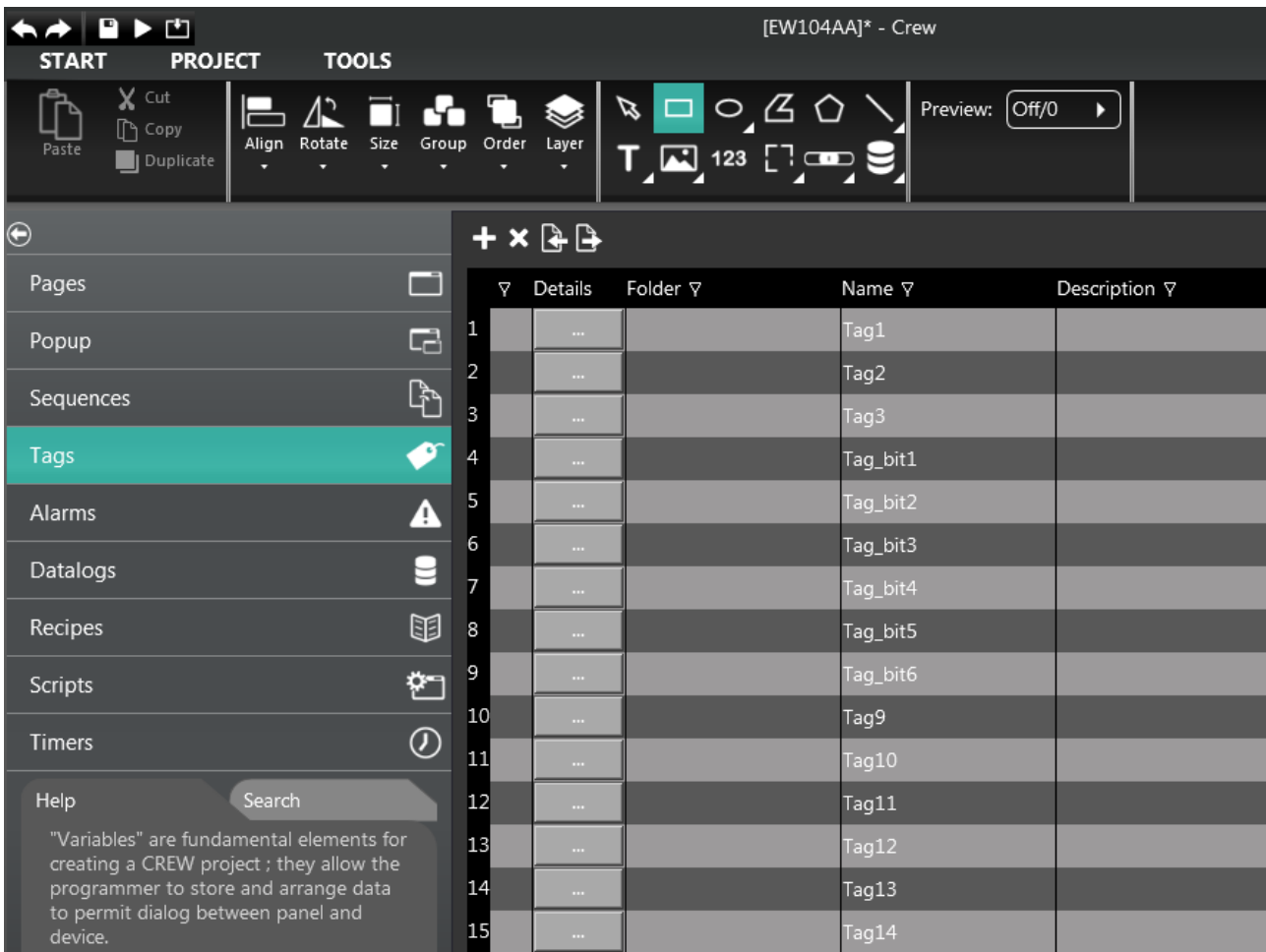
Delete previously created folders.



# CREW Manual

## Tags Grid

The "Tags Grid" is the main tag editing area. In fact, from here it is possible to edit the variables in the project and assign them certain properties (described below).



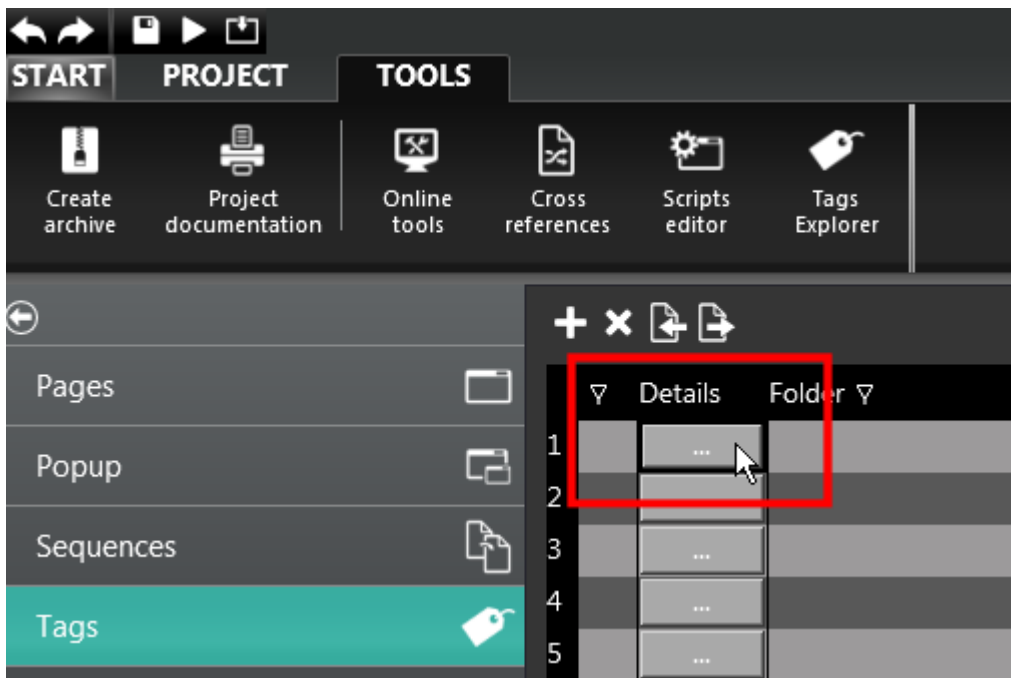
The Tags Grid consists of different columns to which display filters ([Tags Grid Filters](#)) can be applied.

Click the "Details" option of the Tags Grid to access "[Tags Editor](#)", the main tool used to define the features of the variables in the project.

# CREW Manual

## Tags Editor

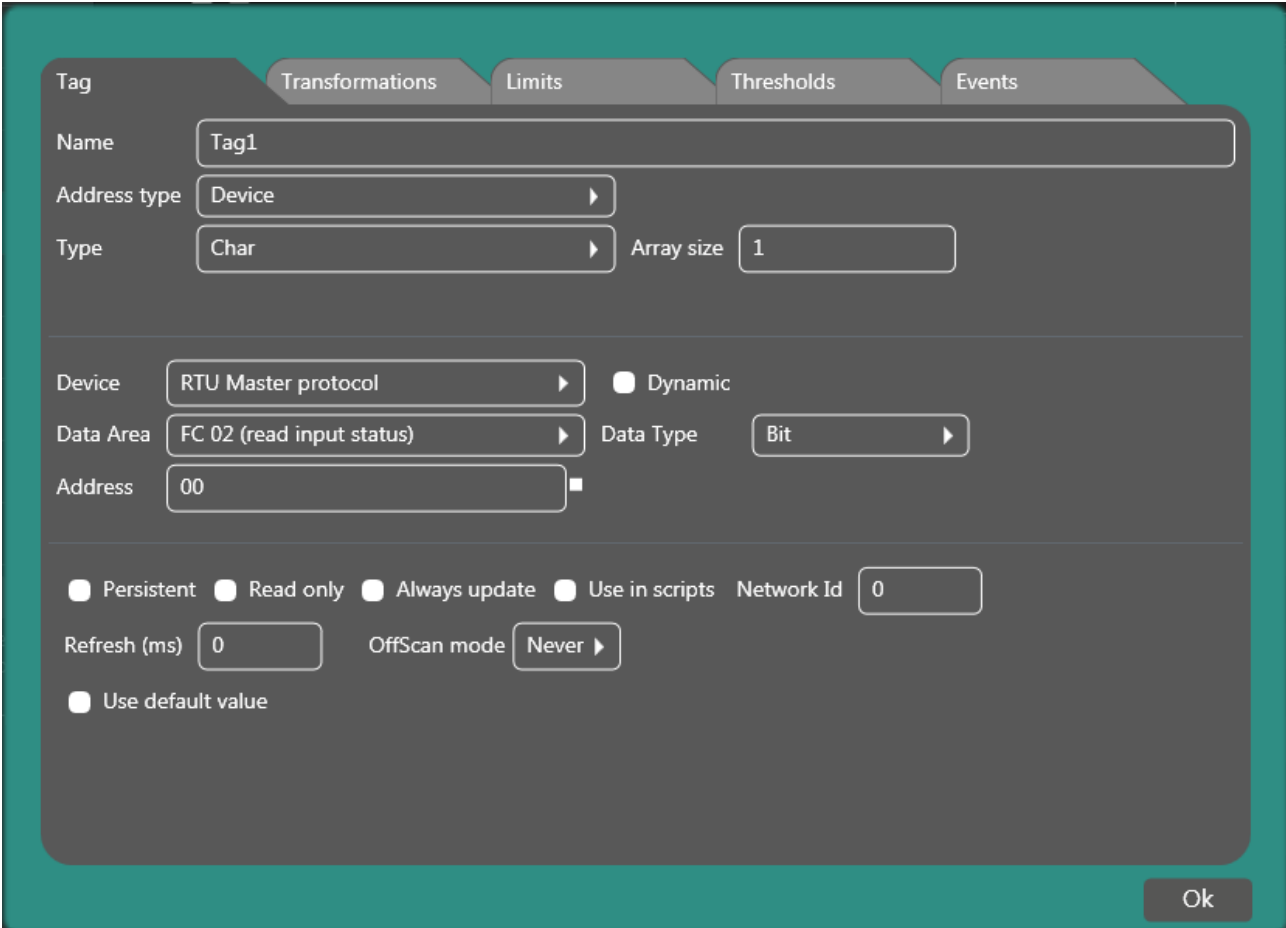
“Tags Editor” is the tool to edit tags and makes it possible to assign all of the associable properties to the variables belonging to the project. To access “Tags Editor”, click the relative key in the “Details” column.



# CREW Manual

## Tag

The first window of the Tags Editor is the "Tag" option.



The screenshot shows the 'Tag' configuration window with the following settings:

- Name: Tag1
- Address type: Device
- Type: Char
- Array size: 1
- Device: RTU Master protocol
- Dynamic:
- Data Area: FC 02 (read input status)
- Data Type: Bit
- Address: 00
- Persistent:
- Read only:
- Always update:
- Use in scripts:
- Network Id: 0
- Refresh (ms): 0
- OffScan mode: Never
- Use default value:

The "Tag" option includes, by default, the following editable areas.

- Name
- Address type
- Type
- Array size
- Device
- Data area

# CREW Manual

- Data type
- Address
- Read only
- Always update
- Use in scripts
- Network ID
- Update (ms)
- OffScan mode
- Use default value



Note: Based on the selected Address Type, the main mask automatically changes as follows:

[Device Tag](#)

[Internal Tag](#)

[System Tag](#)

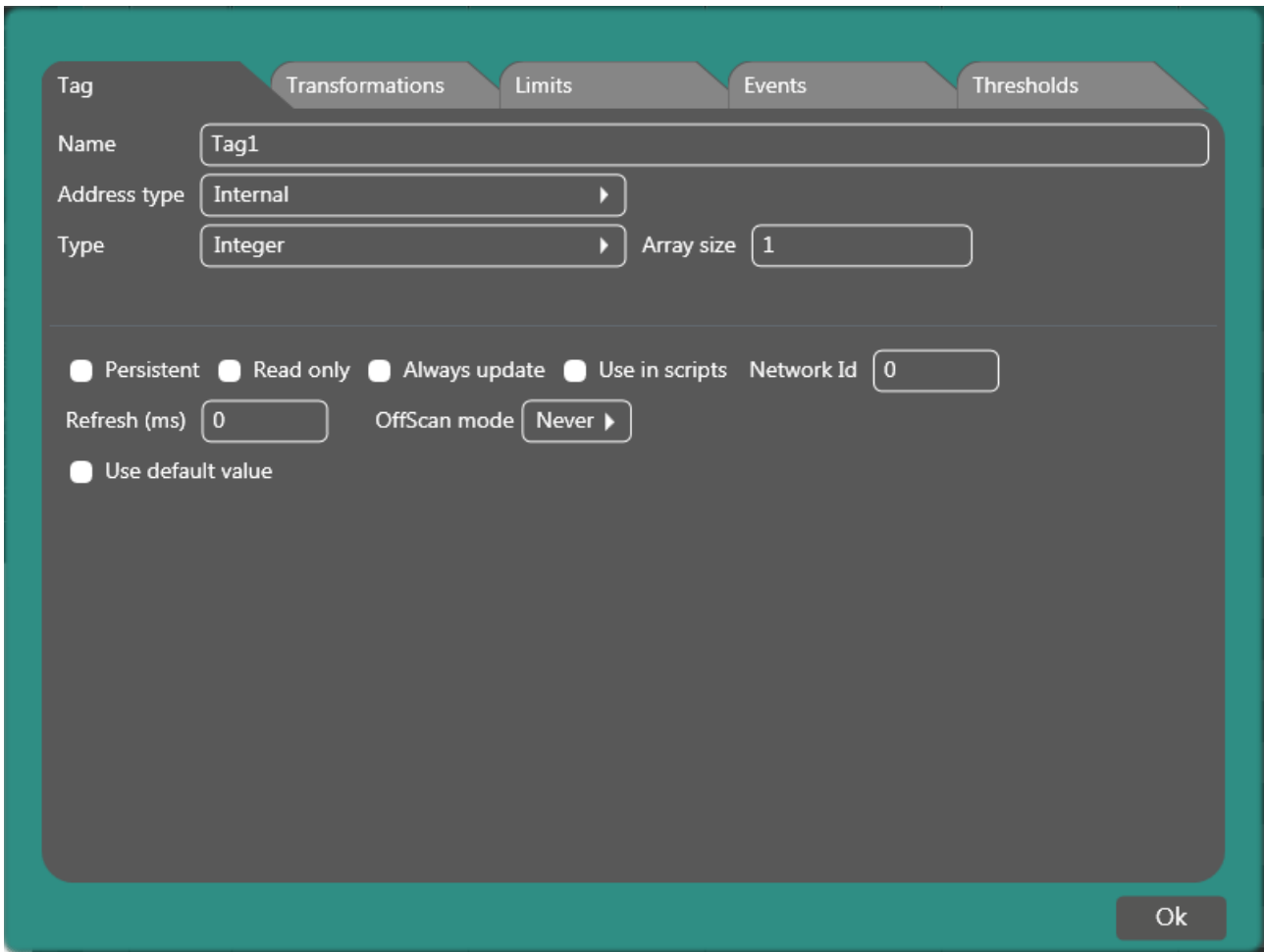
[Network Tag](#)

[Shared Device Tag](#)

# CREW Manual

## Internal Tag

"Internal" Address Type (Internal Tag)



Internal tags are used as storage for local data, the results of operations or scripts, and their value is not read by the PLC device. In this case it is possible to specify whether the value needs to be retentive by enabling the "**Persistent**" option that appears in the "Tags" window.



Note: The value of the retentive tag is stored when the terminal is turned off.

# CREW Manual

The other editing areas of the mask are as follows:

Name:

This is the name of the tag and must be unique, i.e. different tags cannot have the same name.

Address type:

It is necessary to specify the type of tag that you wish to edit. Based on the selected Address Type, the main mask automatically changes as follows:

[Device Tag](#)

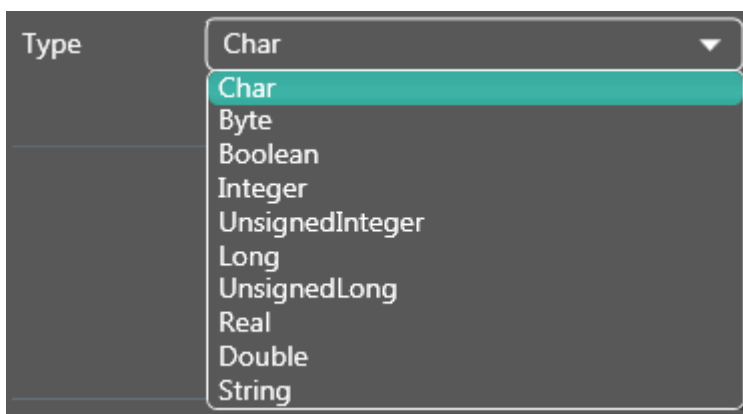
[System Tag](#)

[Network Tag](#)

[Shared Device Tag](#)

Type:

The “Type” mask is used to choose the type of datum that the tag is destined to contain.





# CREW Manual

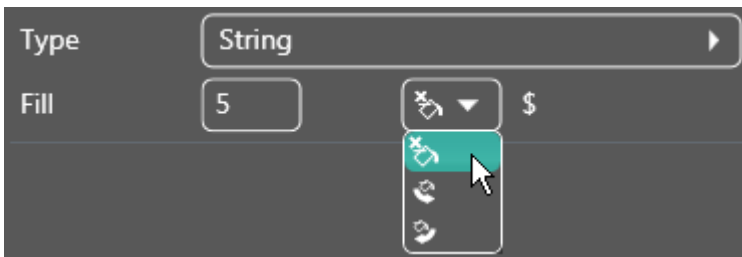
The table below illustrates the possible types of data.

Type	Description	Range
Char	8-bit signed Integer	-128 to 127
Byte	8-bit unsigned Integer	0 to 255
Boolean	Single bit	True (1) or False(0)
Integer	16-bit signed Integer	-32,768 to 32,767
Unsigned Integer	16-bit unsigned Integer	0 to 0xFFFF
Long	32-bit signed Integer	-2,147,483,648 to 2,147,483,647
Unsigned Long	32-bit unsigned Integer	0 to 0xFFFFFFFF
Real	IEEE Floating point 32-bit single precision	-3.402823E38 to -1.401298E-45 for negative values; 1.401298E-45 to 3.402823E38 for positive values
Double	IEEE Floating point 64-bit double precision	-1.7976931348623E308 a -4.9406564584124E-324 for negative values; 4.9406564584124E-324 a 1.7976931348623E308 for positive values
String	ASCII string	ASCII String (max length 0x7FFF characters)
Array of Char	Array of 8-bit signed Integers	1 a 1024
Array of Byte	Array of 8-bit unsigned Integers	1 a 1024
Array of Boolean	Array of Single bits	1 a 1024
Array of Integer (WORD)	Array of unsigned integers	1 a 1024
Array of Unsigned Integer (WORD)	Array of unsigned integers	1 a 1024
Array of Long (DWORD)	Array of 32-bit signed Integers	1 a 1024
Array of Unsigned Long (DWORD)	Array of 32-bit unsigned Integers	1 a 1024
Array of Real	Array of IEEE Floating points 32-bit single precision	1 a 1024
Array of Double	Array of IEEE Floating points 64-bit double precision	1 a 1024
Array of String	Array of ASCII Strings	1 a 1024

# CREW Manual

If the data is "String" type, its length and possible "Filling" can also be referred to.

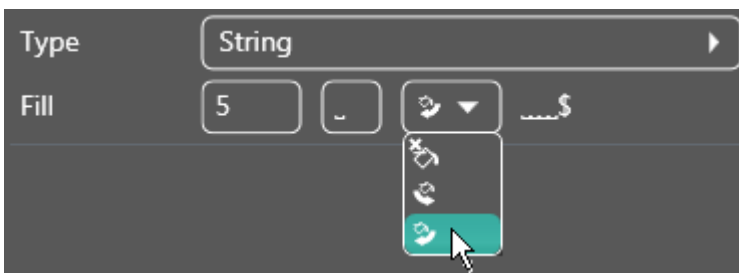
No filling.



Filling to the right of the string.



Filling to the left of the string.



Array Size:

The "Array" data type indicates the size of the data settable by Crew.

# CREW Manual

Read only:

The “Read only” option is used to view the value of a tag without being able to edit it. It is possible to set the tag in this mode when it is used in a data field.

Always update:

Click the "Always Update" option to enable updating even when a tag is not used in any field and when its value is not displayed on the page shown on the panel. This option is required if you wish to access the value of this tag via the scripts, for example.

In the case where the tag is part of Alarms, Trend or Recipes, this setting is ignored and the tag is monitored anyway.

Use in Scripts:

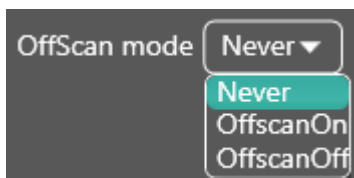
Activate this option to use the tag during execution of a Script in HTML5.

Update:

This determines the refresh rate (expressed in milliseconds) of the values of the relative tags. This function is useful when different degrees of changeability for field tags are required.

OffScan Mode:

The “OffScan” mode establishes whether the tag can be interrogated or not by the terminal. The option can be set as shown in the image.



- “Never”: the tag can never be in “OffScan” mode.
- "OffscanOn": the "OffScan" property of the Tag is enabled (the Tag is placed in "OffscanOn" and so the terminal makes no data request to the device)
- "OffscanOff": the "OffScan" property of the Tag is disabled (the Tag is placed in "OffscanOff" and so the terminal makes a data request to the device)

# CREW Manual

Use default value:

This represents the seed value of the tags that can be set at the start of the project.

The option can be set as follows:

Only in the terminal's data area.

Use default value  Use default value on device Default value

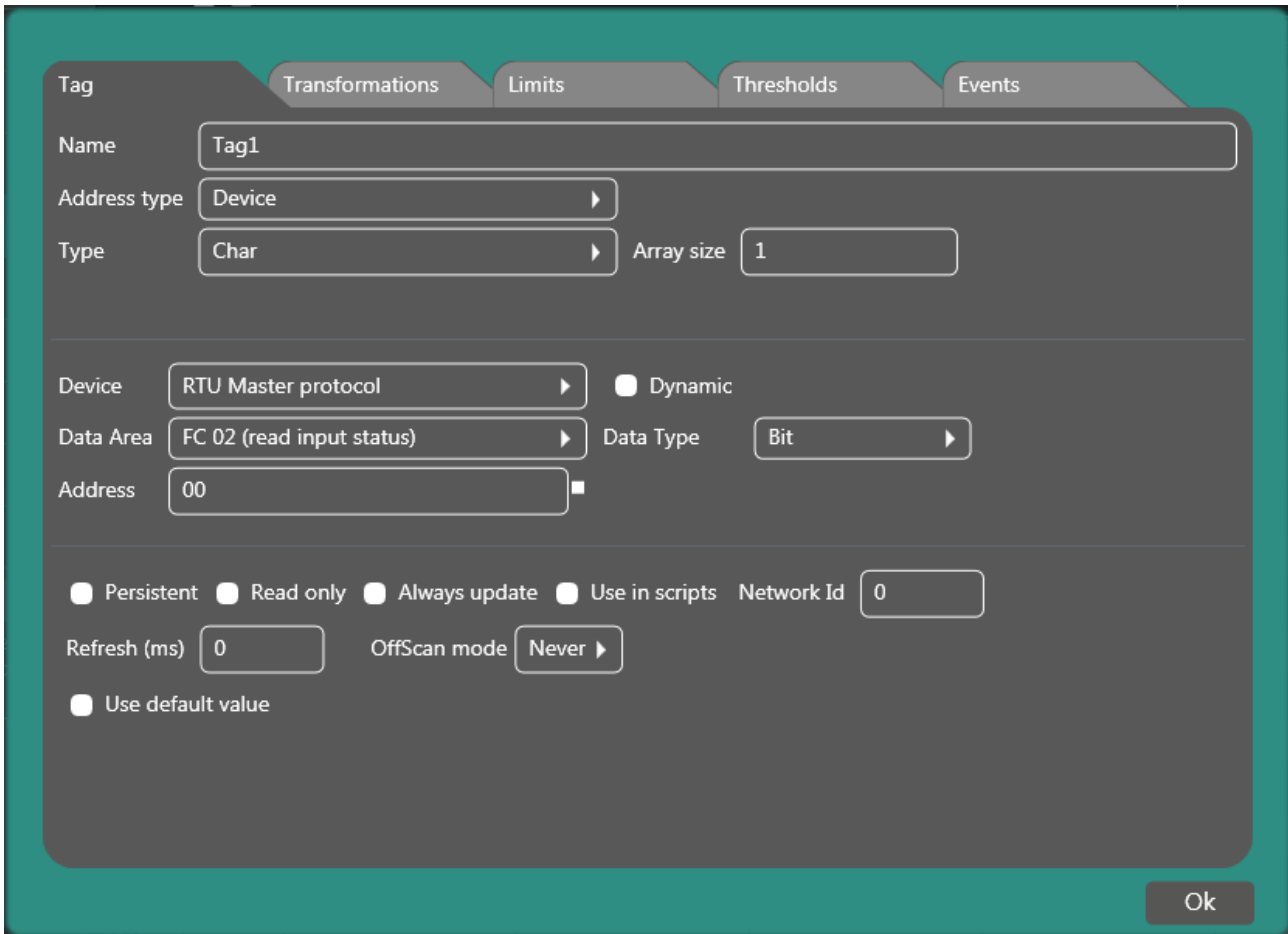
Both in the data area of the terminal and in the data area of the PLC device.

Use default value  Use default value on device Default value

# CREW Manual

## Device Tag

"Device" Address Type (Device Tag)



The screenshot shows the 'Tag' configuration window with the following settings:

- Tab:** Tag (selected), Transformations, Limits, Thresholds, Events
- Name:** Tag1
- Address type:** Device
- Type:** Char, **Array size:** 1
- Device:** RTU Master protocol,  Dynamic
- Data Area:** FC 02 (read input status), **Data Type:** Bit
- Address:** 00
- Persistent,  Read only,  Always update,  Use in scripts, **Network Id:** 0
- Refresh (ms):** 0, **OffScan mode:** Never
- Use default value
- Ok** button

The device tags are shared with related equipment and are the means of data exchange in both directions. It is possible to specify whether the value needs to be retentive by enabling the "**Persistent**" option that appears in the "Tags" window.



Note: The value of the retentive tag "Device" is stored when the terminal is turned off.

# CREW Manual

For each tag a seed value can be entered that is used at the start of the project.

Name:

This is the name of the tag and must be unique, i.e. different tags cannot have the same name.

Address type:

It is necessary to specify the type of tag that you wish to edit. Based on the selected Address Type, the main mask automatically changes as follows:

[Internal Tag](#)

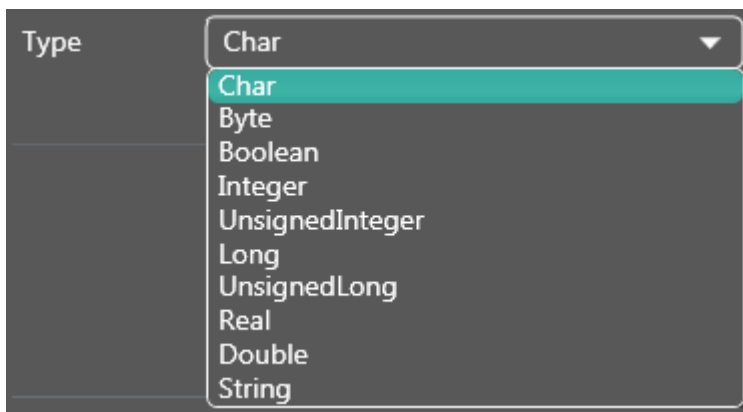
[System Tag](#)

[Network Tag](#)

[Shared Device Tag](#)

Type:

The “Type” mask is used to choose the type of datum that the tag is destined to contain.



# CREW Manual

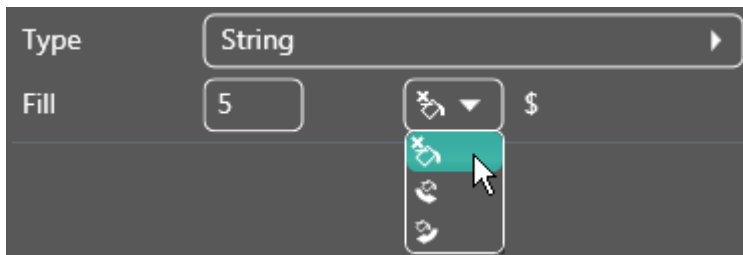
The table below illustrates the possible types of data.

Type	Description	Range
Char	8-bit signed Integer	-128 to 127
Byte	8-bit unsigned Integer	0 to 255
Boolean	Single bit	True (1) or False(0)
Integer	16-bit signed Integer	-32,768 to 32,767
Unsigned Integer	16-bit unsigned Integer	0 to 0xFFFF
Long	32-bit signed Integer	-2,147,483,648 to 2,147,483,647
Unsigned Long	32-bit unsigned Integer	0 to 0xFFFFFFFF
Real	IEEE Floating point 32-bit single precision	-3.402823E38 to -1.401298E-45 for negative values; 1.401298E-45 to 3.402823E38 for positive values
Double	IEEE Floating point 64-bit double precision	-1.7976931348623E308 a -4.9406564584124E-324 for negative values; 4.9406564584124E-324 a 1.7976931348623E308 for positive values
String	ASCII string	ASCII String (max length 0x7FFF characters)
Array of Char	Array of 8-bit signed Integers	1 a 1024
Array of Byte	Array of 8-bit unsigned Integers	1 a 1024
Array of Boolean	Array of Single bits	1 a 1024
Array of Integer (WORD)	Array of unsigned integers	1 a 1024
Array of Unsigned Integer (WORD)	Array of unsigned integers	1 a 1024
Array of Long (DWORD)	Array of 32-bit signed Integers	1 a 1024
Array of Unsigned Long (DWORD)	Array of 32-bit unsigned Integers	1 a 1024
Array of Real	Array of IEEE Floating points 32-bit single precision	1 a 1024
Array of Double	Array of IEEE Floating points 64-bit double precision	1 a 1024
Array of String	Array of ASCII Strings	1 a 1024

# CREW Manual

If the data is "String" type, its length and possible "Filling" can also be referred to.

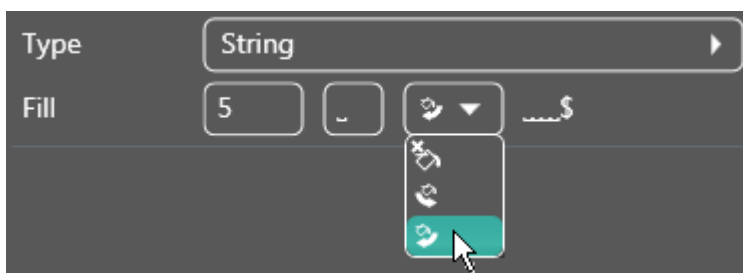
No filling.



Filling to the right of the string.



Filling to the left of the string.



Array Size:

The "Array" data type indicates the size of the data settable by Crew.



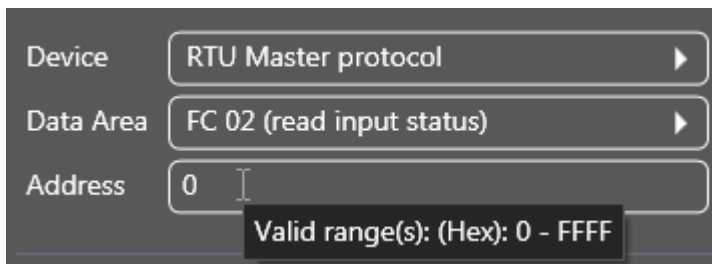
# CREW Manual

Device:

This makes it possible to enter the target device, the relative memory addresses and the class of update.

Date area, Data type, Address:

For the device tags, the target memory areas of the values need to be specified. To guide the user, Crew provides information on valid memory ranges, that it calculates automatically in accordance with the device entered in the project. Move the mouse to the “Address” editable field to view the valid ranges based on the type of data selected in the “Data Area” field.



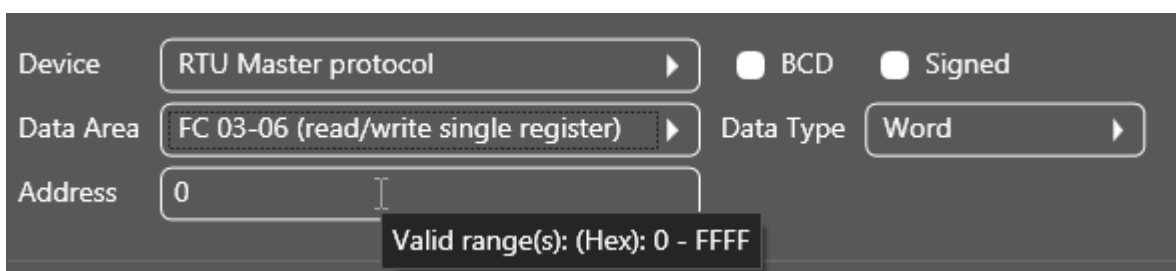
Device: RTU Master protocol

Data Area: FC 02 (read input status)

Address: 0

Valid range(s): (Hex): 0 - FFFF

It is also necessary to state which type of memory to reserve, from “Bit”, “Byte”, “Word”, “DWord” or “String”. If it is String type, it is possible to state whether the memory is “Signed” (for relative values) or “BCD”.



Device: RTU Master protocol

Data Area: FC 03-06 (read/write single register)

Address: 0

Data Type: Word

BCD  Signed

Valid range(s): (Hex): 0 - FFFF



Note: The Binary-coded decimal (BCD) is a class of binary encodings of decimal numbers. In this format, each digit of a number is represented by a binary code of four bits, the value of which is between 0 (0000) and 9 (1001). For example, the number 127 is represented in BCD as 0001, 0010, 0111.

# CREW Manual

Read only:

The “Read only” option is used to view the value of a tag without being able to edit it. It is possible to set the tag in this mode when it is used in a data field.

Always update:

Click the "Always Update" option to enable updating even when a tag is not used in any field and when its value is not displayed on the page shown on the panel. This option is required if you wish to access the value of this tag via the scripts, for example.

In the case where the tag is part of Alarms, Trend or Recipes, this setting is ignored and the tag is monitored anyway.

Use in Scripts:

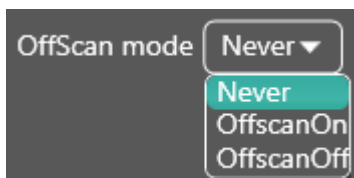
Activate this option to use the tag during execution of a Script in HTML5.

Update:

This determines the update frequency (expressed in milliseconds) of the values of the relative tags. This function is useful when different degrees of changeability for field tags are required.

OffScan Mode:

The “Offscan” mode establishes whether the tag can be interrogated or not by the terminal. The option can be set as shown in the image.



- “Never”: the tag can never be in “Offscan” mode.
- "OffscanOn": the "Offscan" property of the Tag is enabled (the Tag is placed in "OffscanOn" and so the terminal makes no data request to the device)
- "OffscanOff": the "Offscan" property of the Tag is disabled (the Tag is placed in "OffscanOff" and so the terminal makes a data request to the device)

# CREW Manual

Use default value:

Represents the seed value of the tags that can be set at the start of the project.

The option can be set as follows:

Only in the terminal's data area.

Use default value  Use default value on device Default value

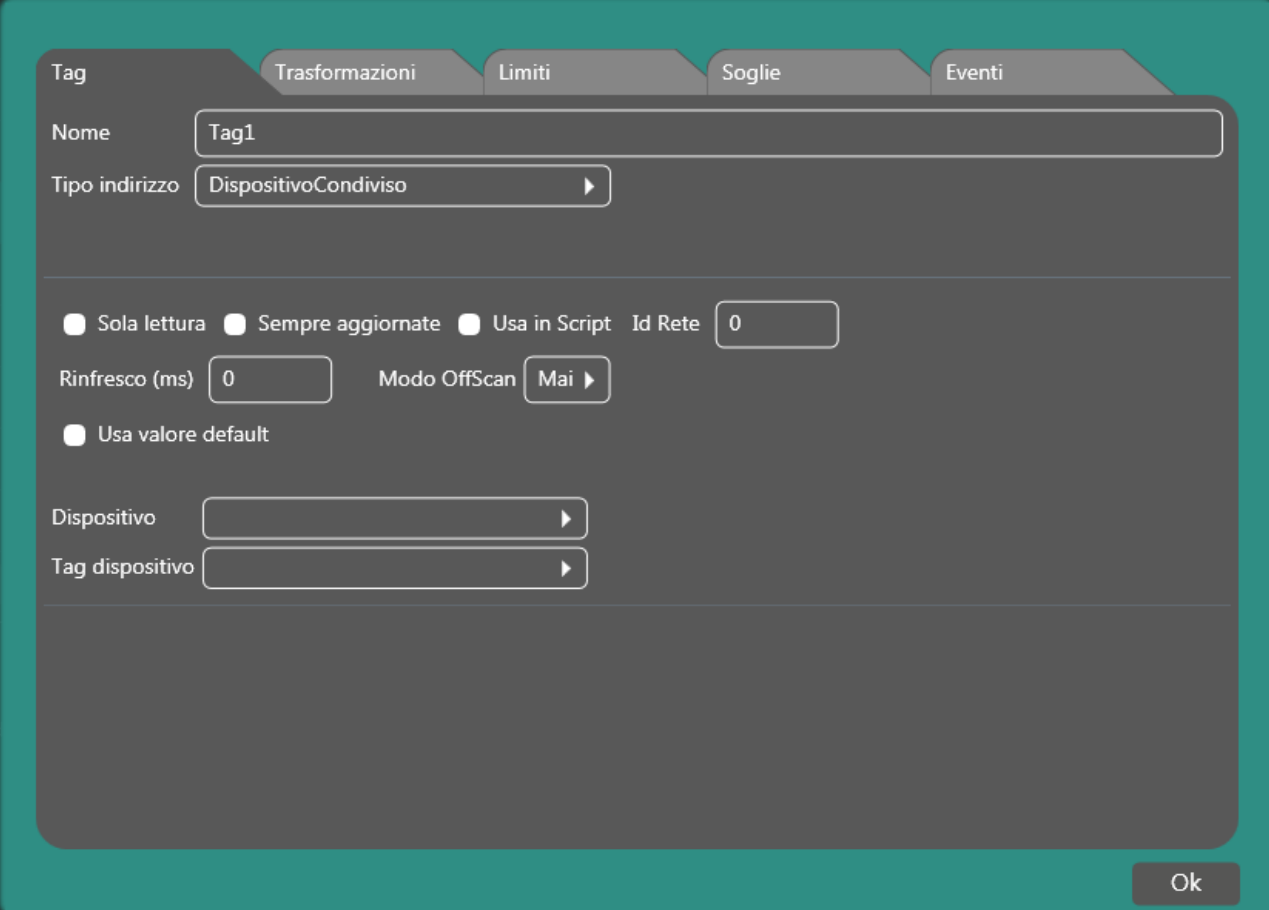
Both in the data area of the terminal and in the data area of the PLC device.

Use default value  Use default value on device Default value

# CREW Manual

## Shared Device Tag

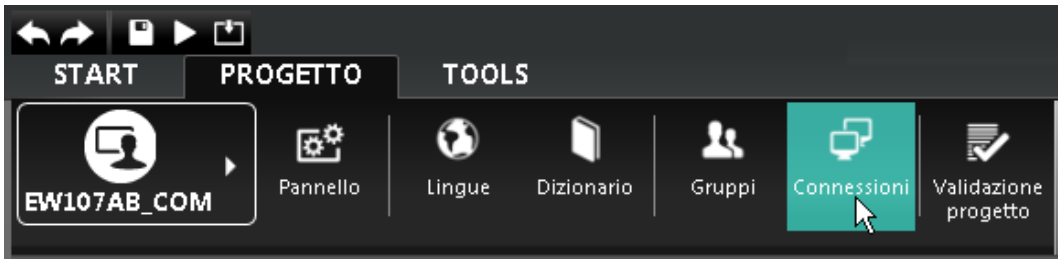
"SharedDevice" Address Type (Shared Device Tag)



This type of Tag is used when there is a system with multiple panels that read shared data areas in a single device (PLC). If it is necessary to change the Tag addresses, simply change the address configured in the tags/s of the shared device, without having to change the addresses on the panels. Below is an example describing the above.

# CREW Manual

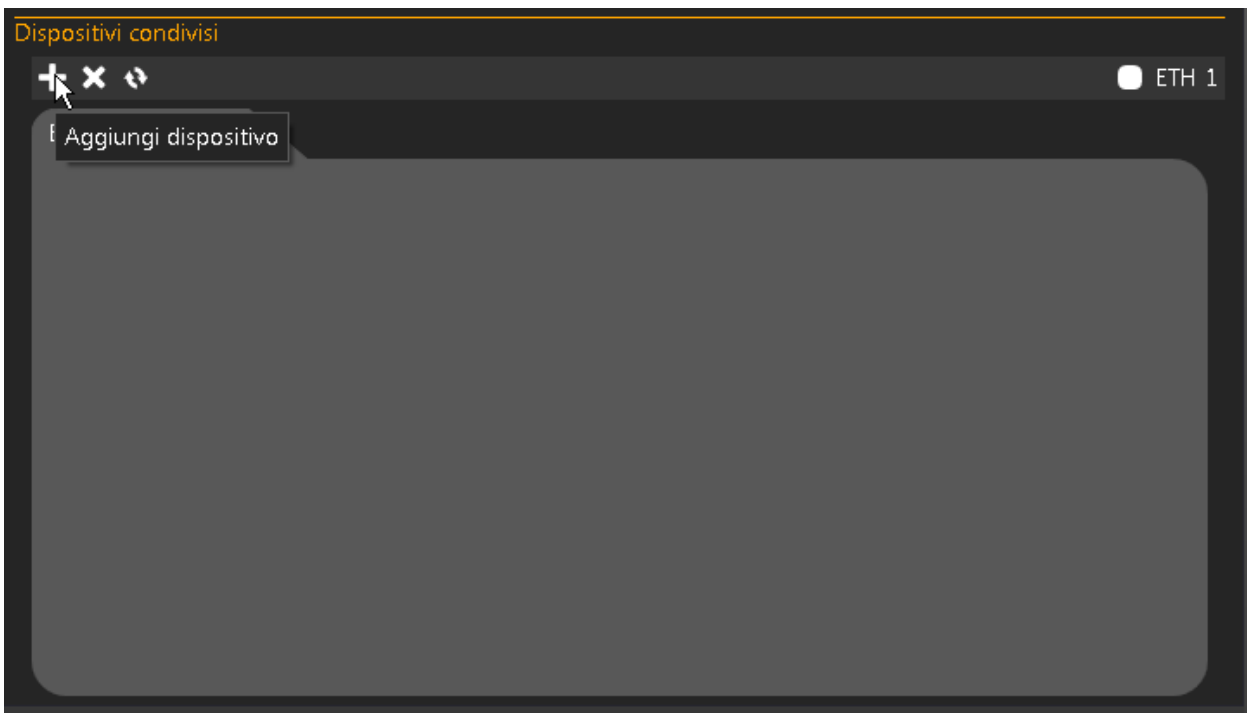
Select the "Connections" menu.



Then select "Shared Devices" from the bottom of the window that appears.

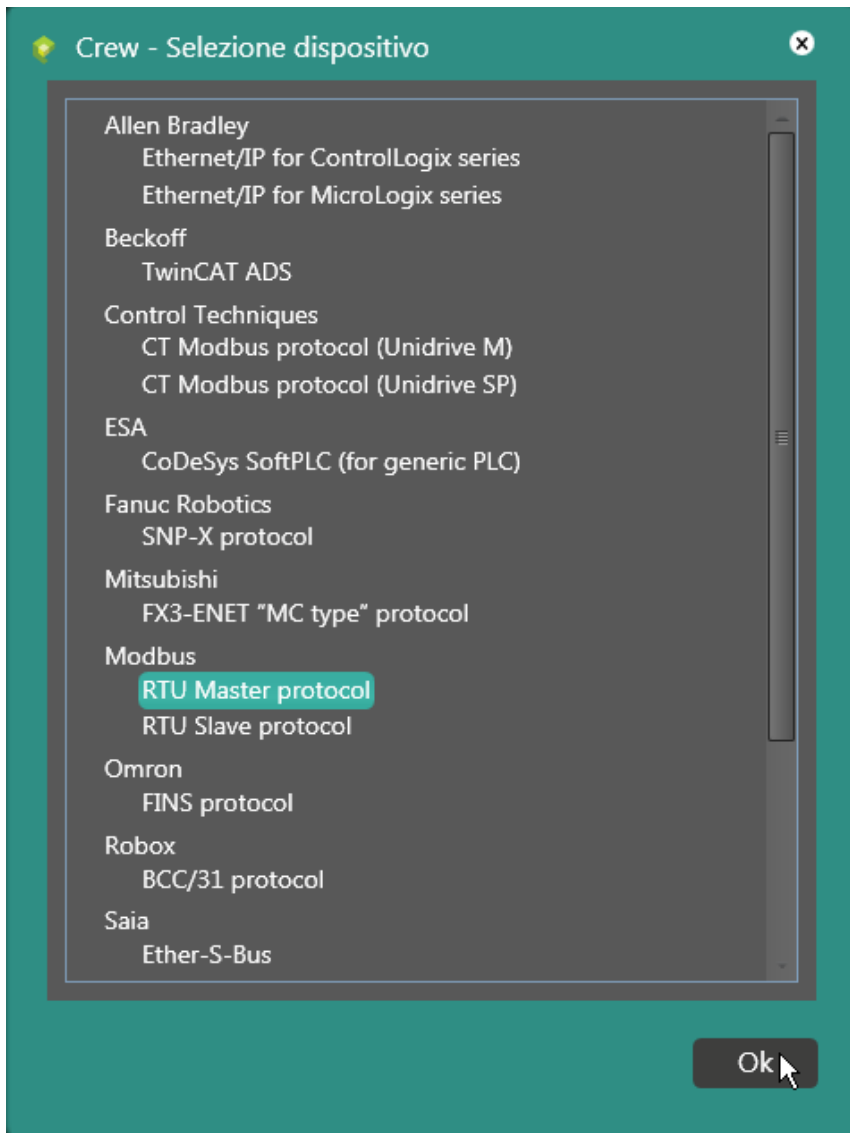


Click "Add device" to add a new device for sharing.



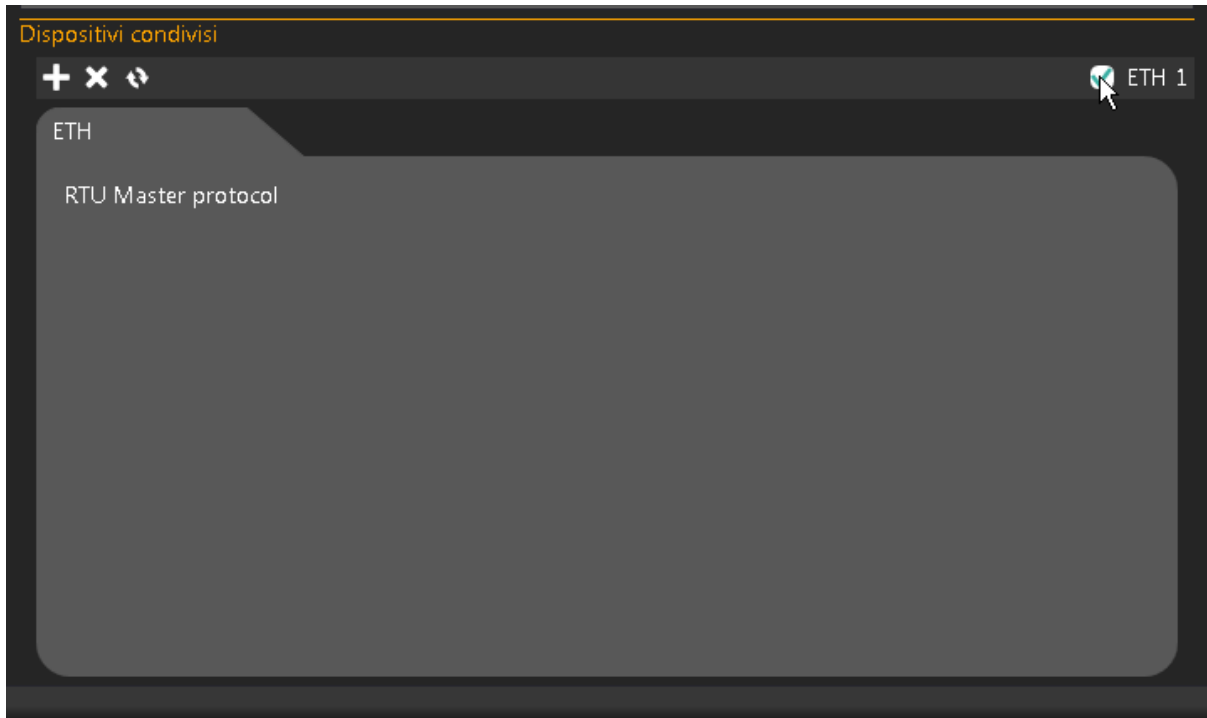
# CREW Manual

Choose the required device and confirm with “Ok”.



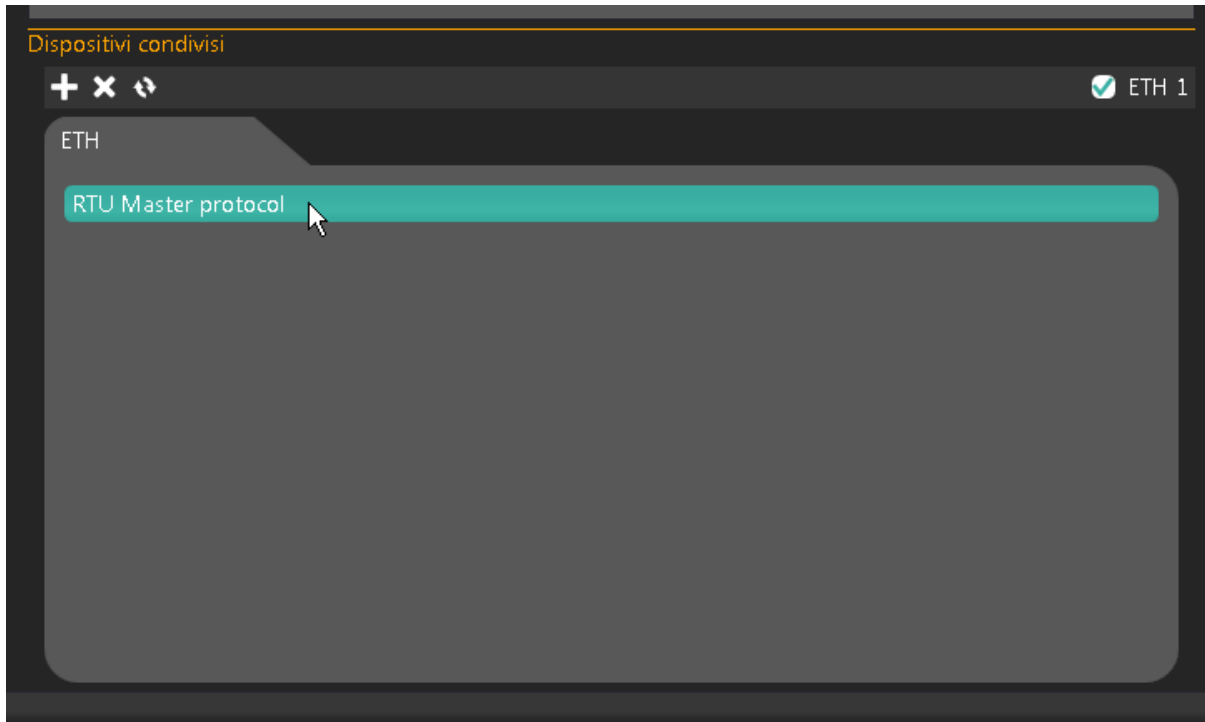
# CREW Manual

Enable the ETH 1 "Check box".



# CREW Manual

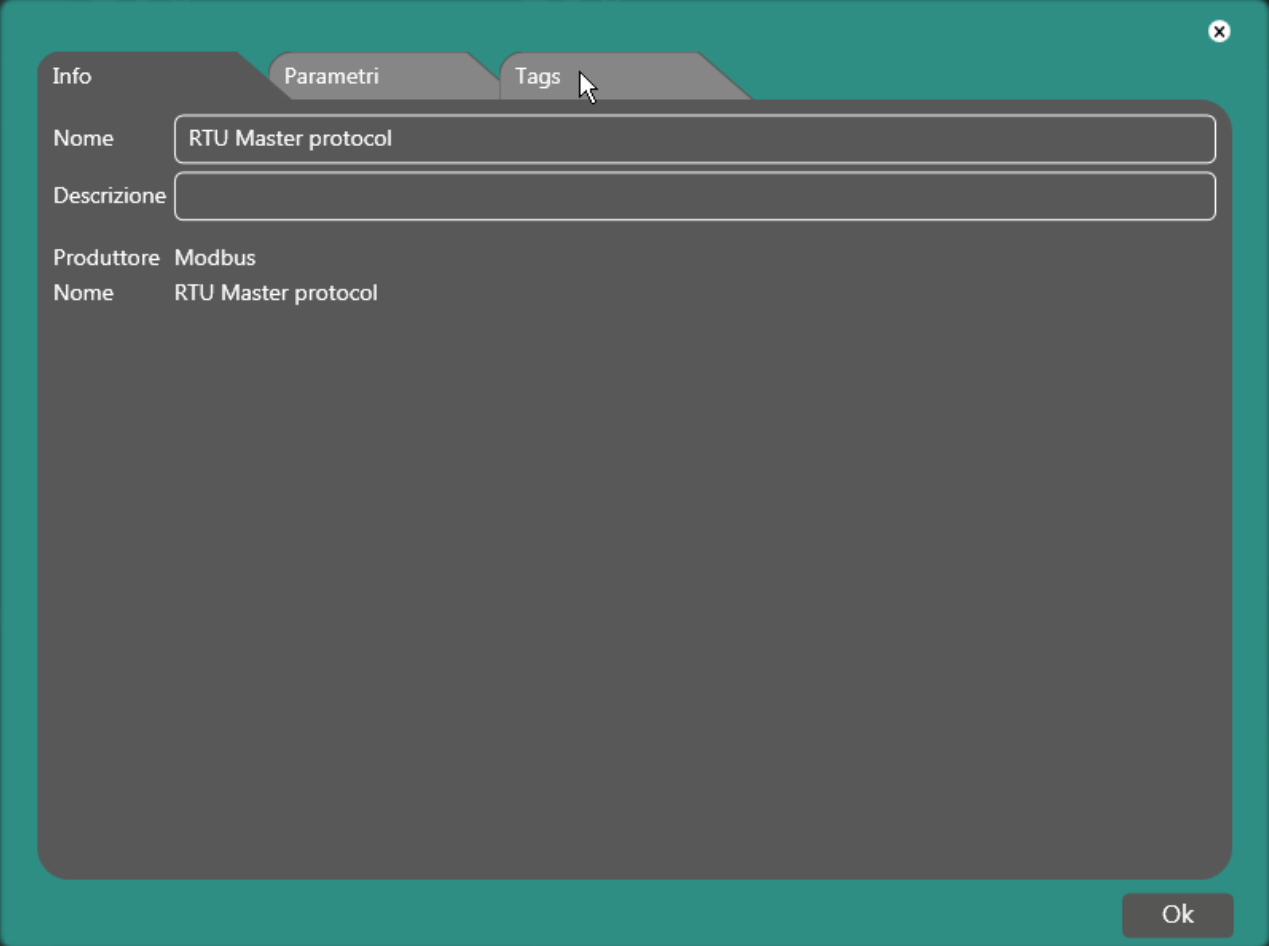
Double click to enter into the device.





# CREW Manual

Select the “Tags” option.



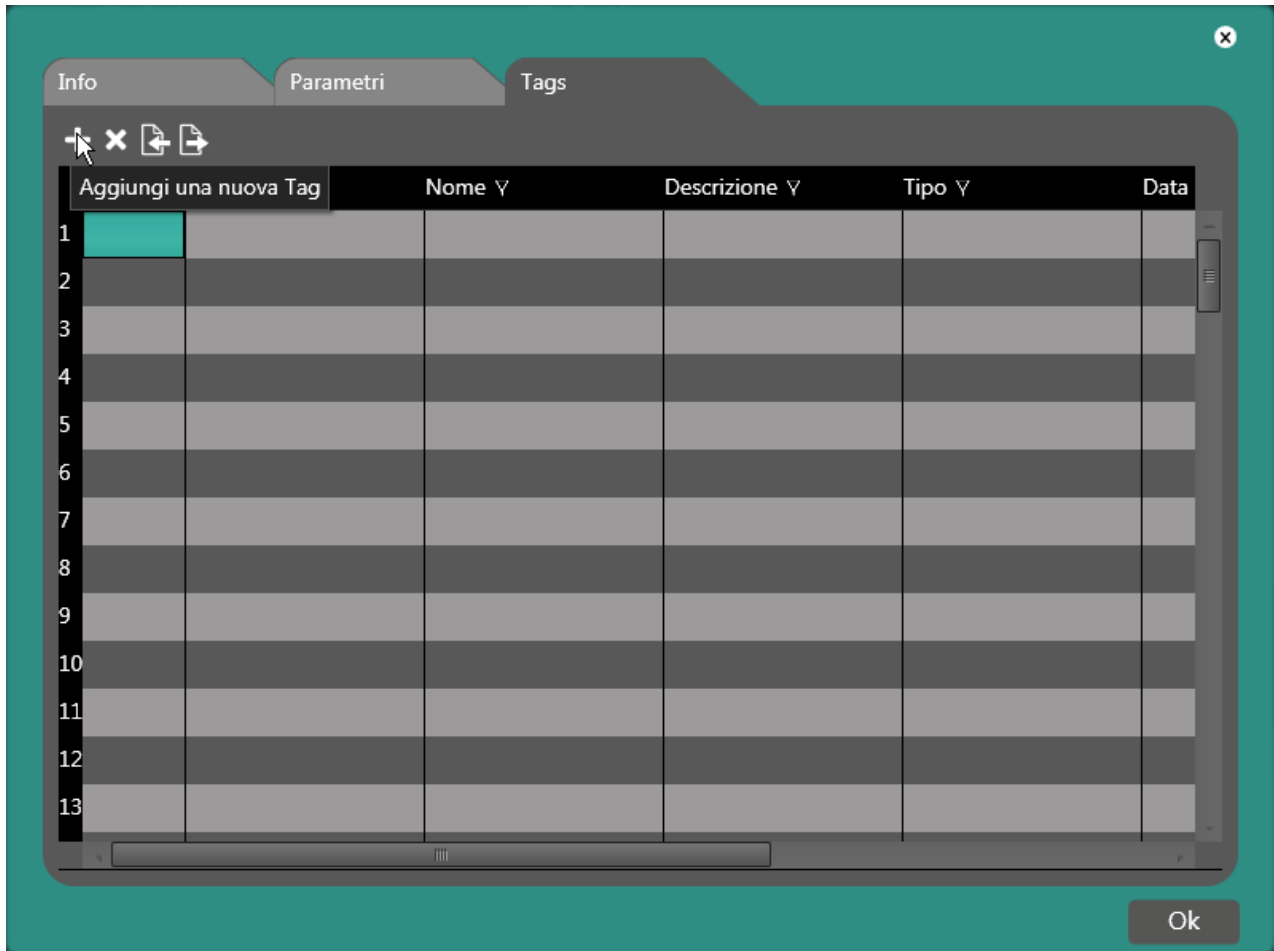
The screenshot shows a software window with three tabs: 'Info', 'Parametri', and 'Tags'. The 'Tags' tab is selected and active. The window contains the following fields and text:

- Nome:** RTU Master protocol
- Descrizione:** (empty text box)
- Produttore:** Modbus
- Nome:** RTU Master protocol

An 'Ok' button is located at the bottom right of the window.

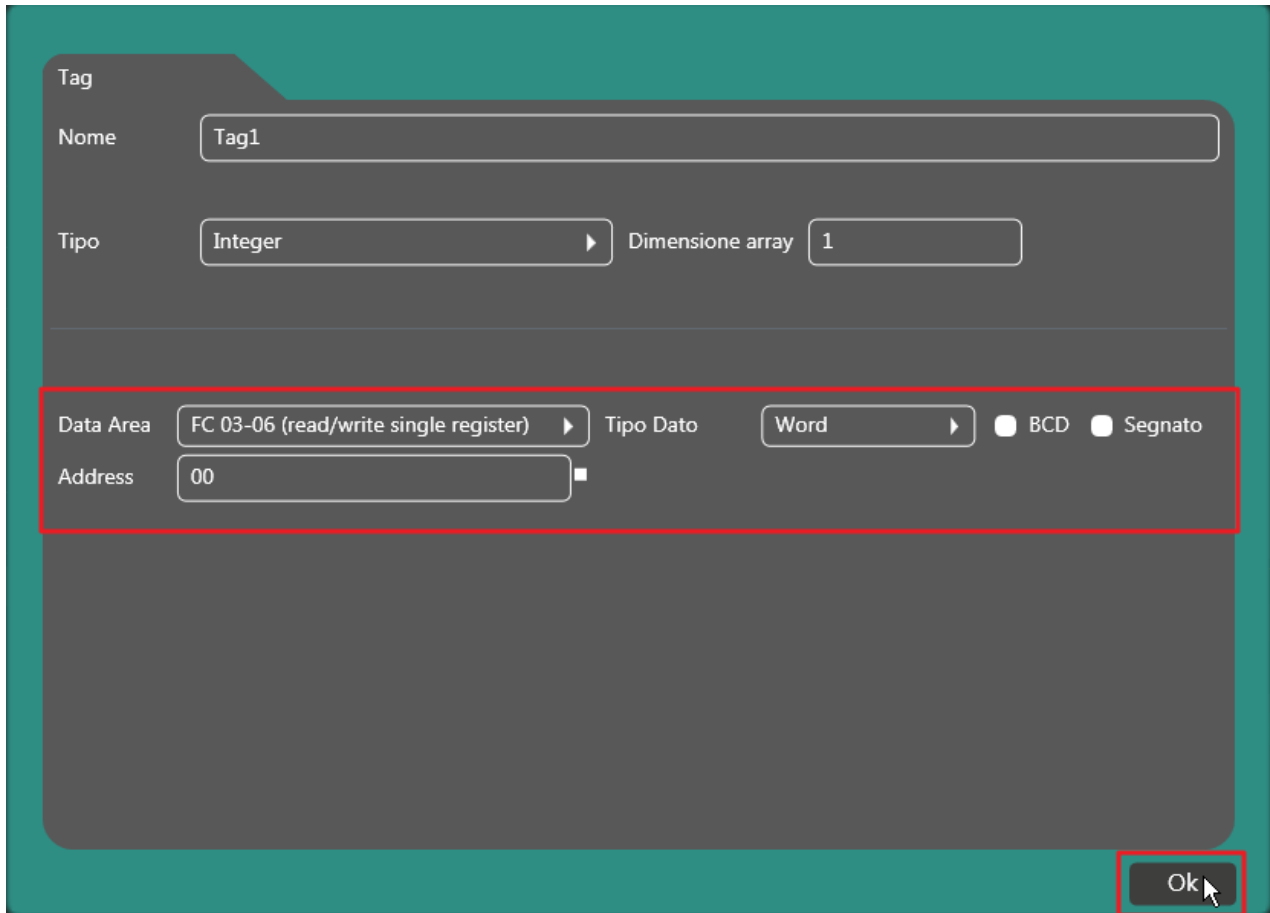
# CREW Manual

Click the “+” icon to add a new tag.



# CREW Manual

Set all of the necessary parameters and confirm with “Ok”.



Tag

Nome

Tipo  Dimensione array

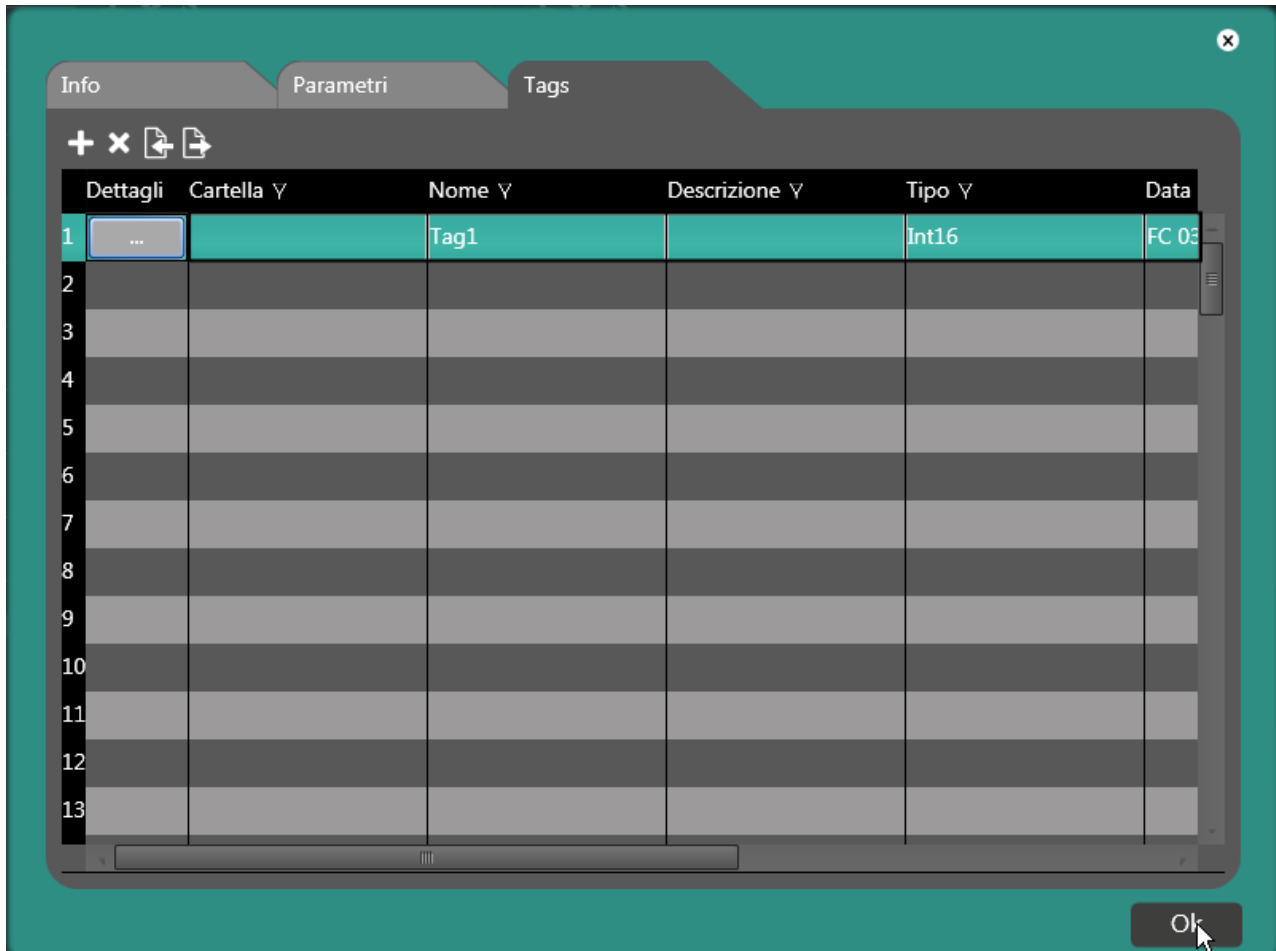
Data Area  Tipo Data   BCD  Segnato

Address

Ok

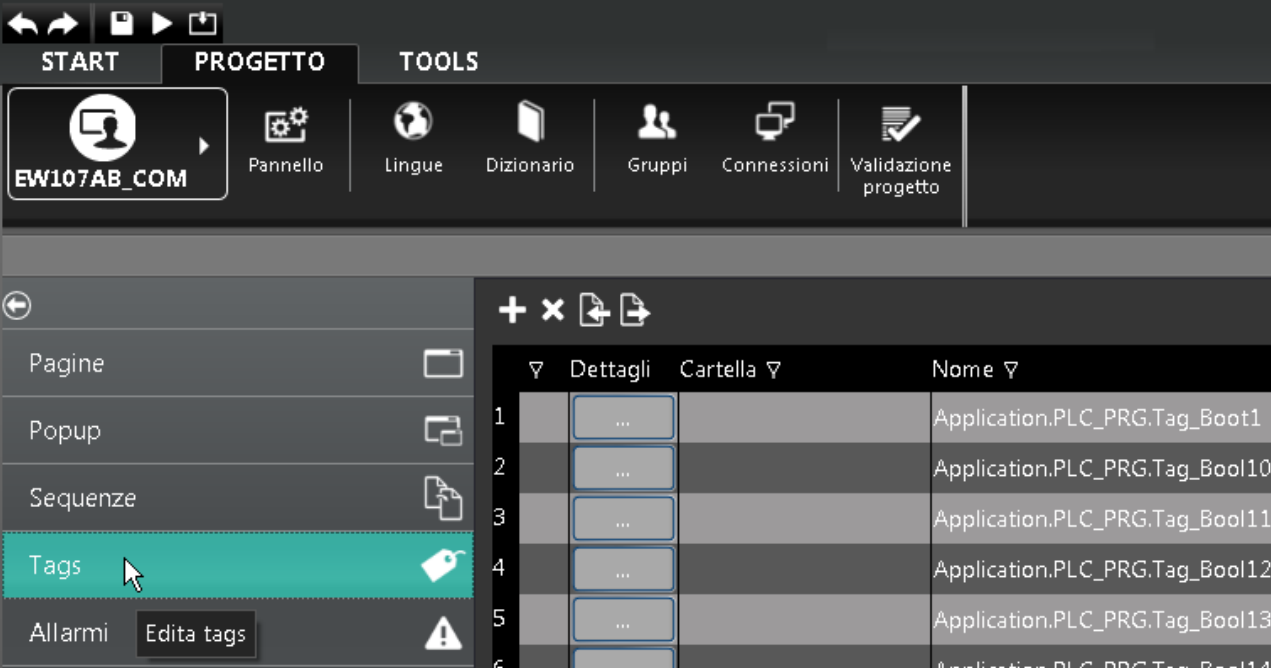
# CREW Manual

At this point it is necessary to create the reference tag in the PLC.



# CREW Manual

Confirm with “Ok” and go to the list of tags contained on the panel.



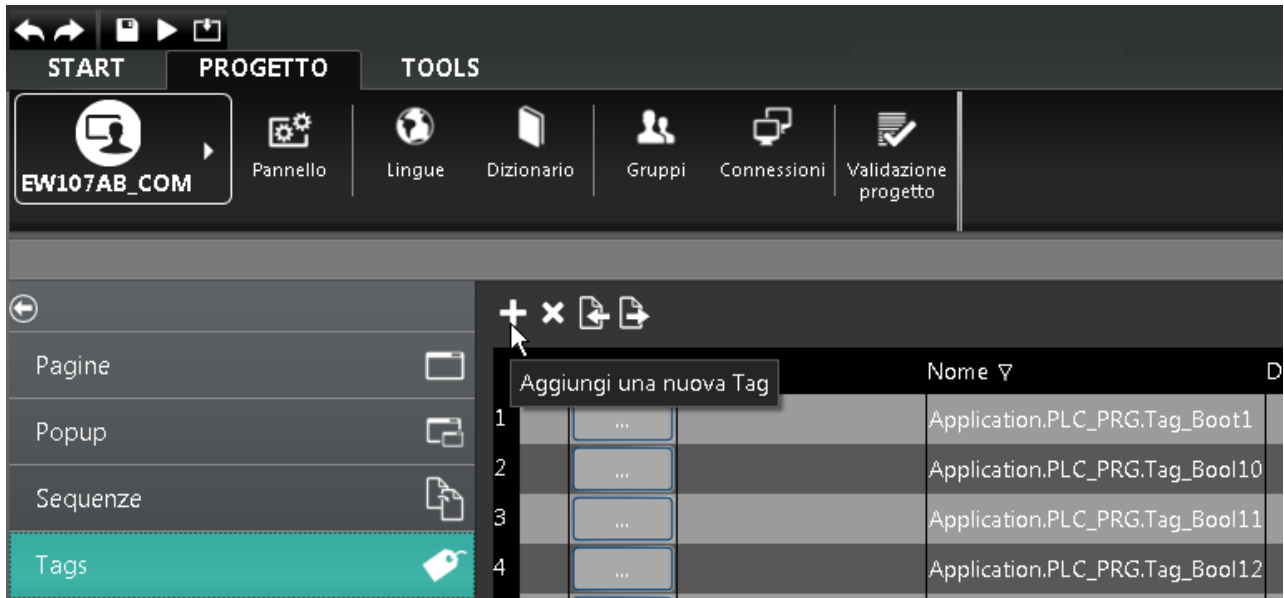
The screenshot shows the software interface with the following elements:

- Top Bar:** Navigation icons (back, forward, home, refresh) and tabs for **START**, **PROGETTO**, and **TOOLS**.
- Project Information:** A box on the left shows the project name **EW107AB\_COM**.
- Tools Panel:** A row of icons for **Pannello**, **Lingue**, **Dizionario**, **Gruppi**, **Connessioni**, and **Validazione progetto**.
- Left Sidebar:** A list of menu items: **Pagine**, **Popup**, **Sequenze**, **Tags** (highlighted with a mouse cursor), and **Allarmi** (with a sub-item **Edita tags**).
- Main Window:** A table with columns **Dettagli**, **Cartella**, and **Nome**. The table contains 6 rows of data, each with a row number (1-6) and a tag name.

	Dettagli	Cartella	Nome
1	...		Application.PLC_PRG.Tag_Boot1
2	...		Application.PLC_PRG.Tag_Boot10
3	...		Application.PLC_PRG.Tag_Boot11
4	...		Application.PLC_PRG.Tag_Boot12
5	...		Application.PLC_PRG.Tag_Boot13
6	...		Application.PLC_PRG.Tag_Boot14

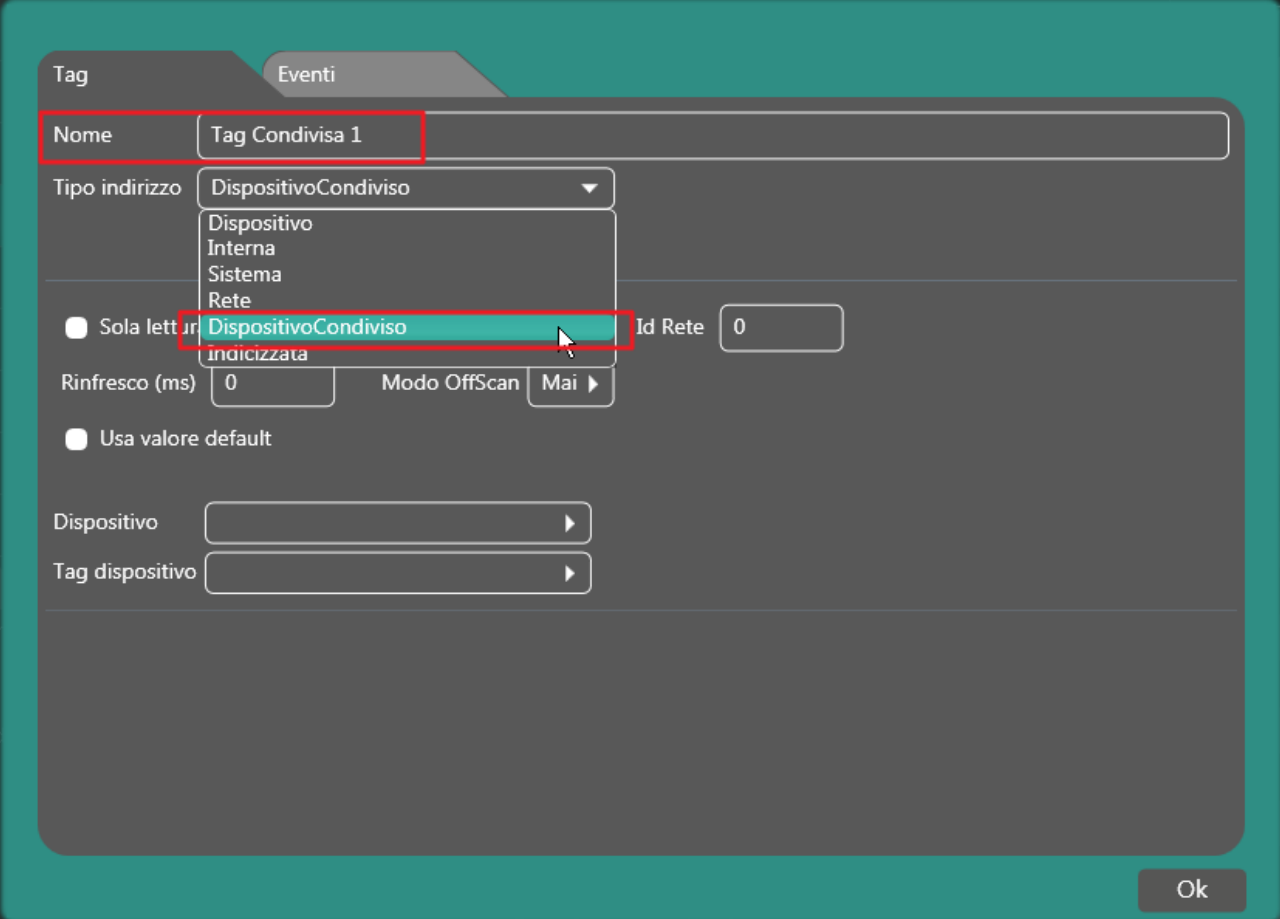
# CREW Manual

Add a new tag.



# CREW Manual

Call the new Tag "Shared Tag 1" and select "Shared Device" as the "Address Type" :



Tag

Eventi

Nome

Tipo indirizzo

- Dispositivo
- Interna
- Sistema
- Rete
- DispositivoCondiviso**
- Indicizzata

Sola lettura indicizzata

Rinfresco (ms)  Modo OffScan

Usa valore default

Dispositivo

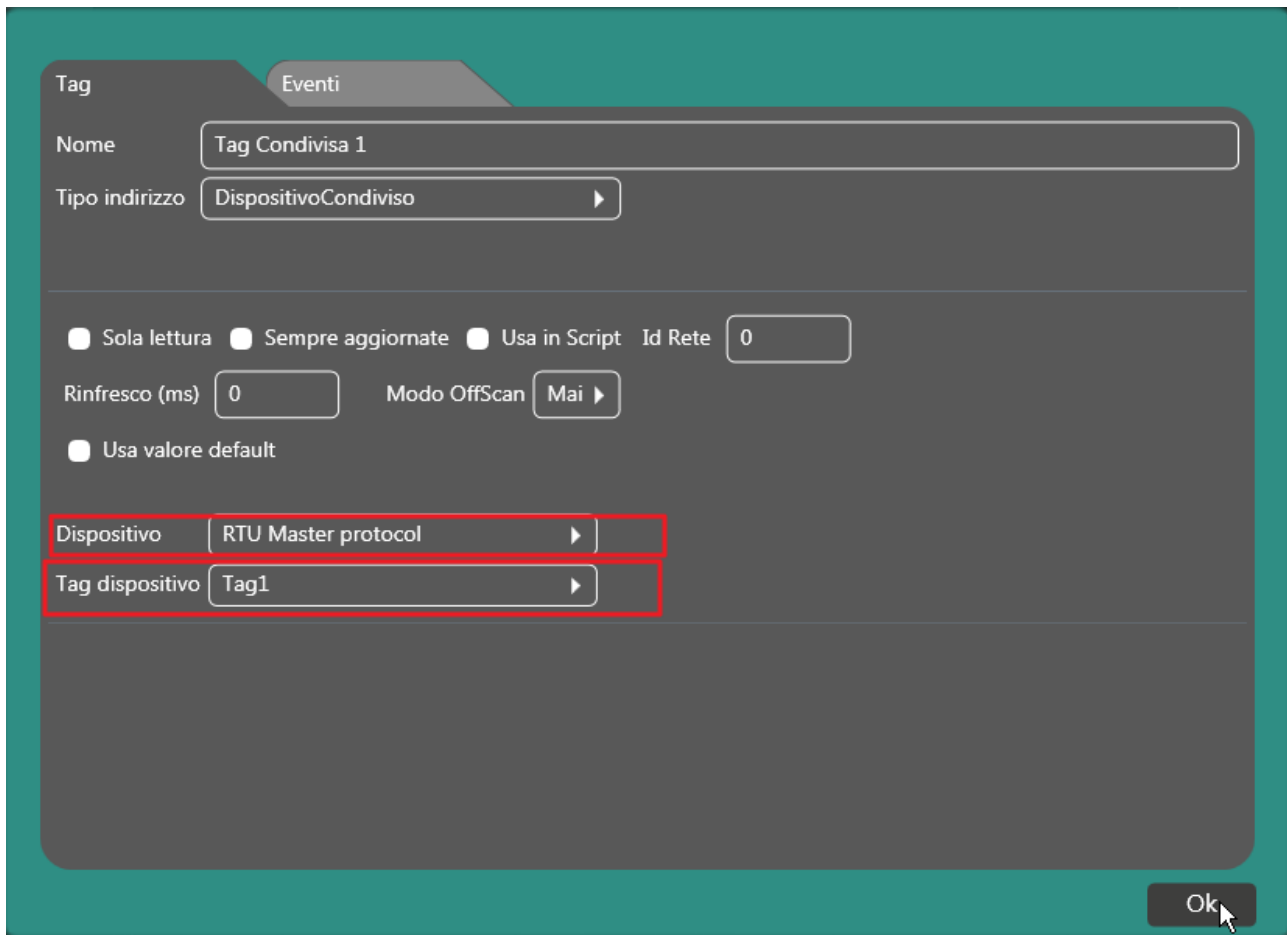
Tag dispositivo

Id Rete

Ok

# CREW Manual

Then select the device (PLC) that contains the reference tag that the panel/s needs to read and click "OK" to confirm.



The screenshot shows a configuration window for a tag. The window has two tabs: "Tag" (selected) and "Eventi". The "Tag" tab contains the following fields and options:

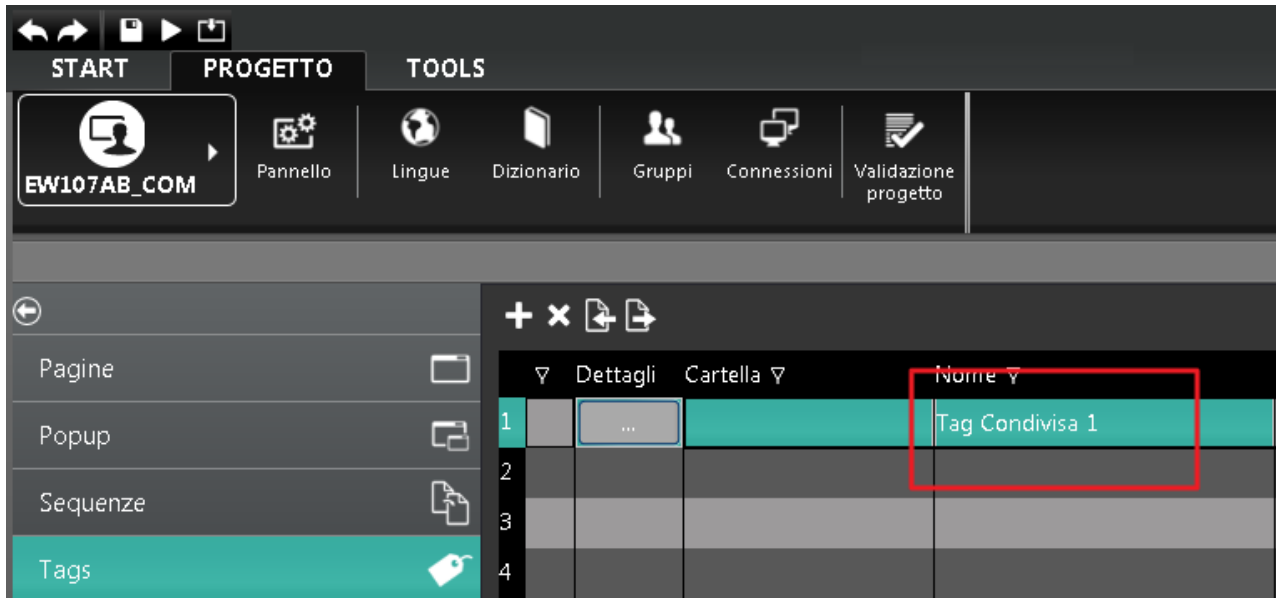
- Nome:** Tag Condivisa 1
- Tipo indirizzo:** DispositivoCondiviso
- Options:**  Sola lettura,  Sempre aggiornate,  Usa in Script
- Id Rete:** 0
- Rinfresco (ms):** 0
- Modo OffScan:** Mai
- Usa valore default
- Dispositivo:** RTU Master protocol
- Tag dispositivo:** Tag1

The "Dispositivo" and "Tag dispositivo" fields are highlighted with red boxes. An "Ok" button is located at the bottom right of the dialog.



# CREW Manual

Now "Shared Tag 1" has been entered correctly in the project panel.

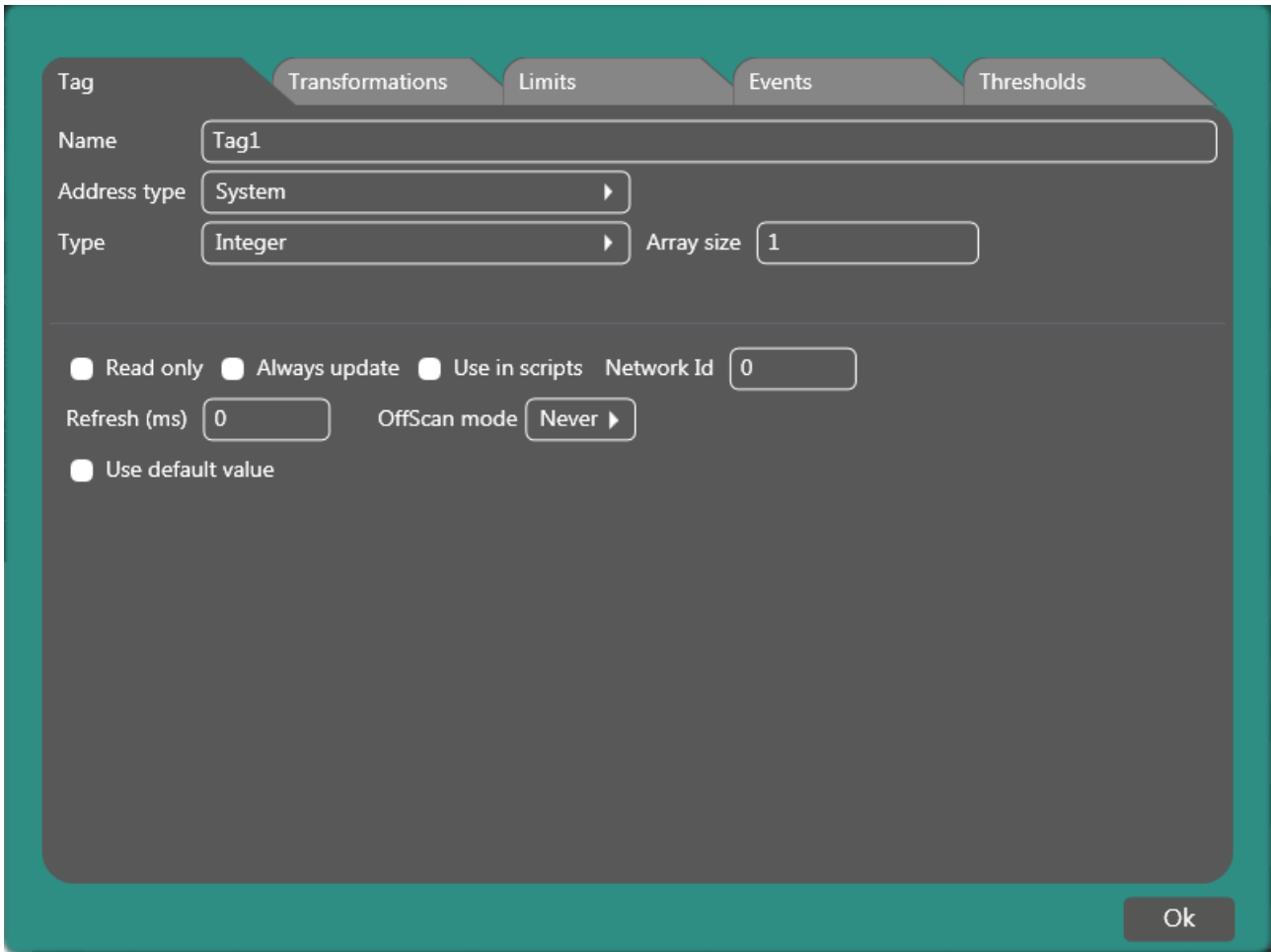


If "n" panels that want to read the shared tag (Tag1) on the PLC, you need to create "n" Tags of the same type shown above (Shared Tag 2, 3, 4... etc.).

# CREW Manual

## System Tag

"System" Address Type (System Tag)



The screenshot shows a configuration window for a tag. The window has a dark grey background with a teal border. At the top, there are five tabs: 'Tag' (selected), 'Transformations', 'Limits', 'Events', and 'Thresholds'. The 'Tag' tab contains the following fields and options:

- Name:** Tag1
- Address type:** System (dropdown menu)
- Type:** Integer (dropdown menu)
- Array size:** 1
- Read only:**
- Always update:**
- Use in scripts:**
- Network Id:** 0
- Refresh (ms):** 0
- OffScan mode:** Never (dropdown menu)
- Use default value:**

An 'Ok' button is located at the bottom right of the window.

This section is dedicated to system tags, namely tags that contain information about terminal operation and the project in execution. The system variables are inside the terminal and are accessible in Runtime as read-only (except for some tags that can be changed).

# CREW Manual

The system tags can be created, managed and used in the project in the same way as the other variables. The mask used to change the system variables is the same as the one for internal tags, except for the “Persistent” option, which does not apply in this case.

The name of the system variables starts by default with the prefix SYS\_, followed by a string identifying its function.

It is possible to select the system tag type from the drop down menu. The characteristics of each tag, listed in the table below, appear at the bottom of the mask.

## INFO (System and project information)

Name	Description	Access	Type
<b>SYS_MachineName</b>	Name of terminal, how it is configured in the project	R	string [64]
<b>SYS_ProjectName</b>	Name of project	R	string [64]
<b>SYS_PlatformType</b>	Type of machine on which runtime is running : 0 : HMI EW 1 : IPC EW	R	
<b>SYS_IPAddressStr</b>	Terminal's IP address	R	string [16]
<b>SYS_IPAddressDw</b>	Terminal's IP address	RW	u.long
<b>SYS_RunMode</b>	Current running mode of runtime : 0 : Server and client 1 modules have been uploaded : Only client modules (UI) have been uploaded. 2 : Only server modules have been uploaded; no window management 3 : Only server modules have been uploaded; a basic window is managed	RW	u.int
<b>SYS_ScreenHorDim</b>	Horizontal size of the screen (FullScreen page, in pixels)	R	u.int
<b>SYS_ScreenVertDim</b>	Vertical size of the screen (FullScreen page, in pixels)	R	u.int

# CREW Manual

## CONFIGURATION (Project configurations)

Name	Description	Access	Type
<b>SYS_TagsPath</b>	Path of the files in use by the persistent variables	R	string [256]
<b>SYS_AlarmsPath</b>	Path of the files in use by the alarm log	R	string [256]
<b>SYS_RecipesPath</b>	Path of the files in use by the recipes	R	string [256]
<b>SYS_DataLogsPath</b>	Path of the files in use by the samples	R	string [256]
<b>SYS_UsersPath</b>	Path of the files in use by the users	R	string [256]
<b>SYS_TagsNum</b>	Number of variables configured in the project	R	u.long
<b>SYS_AlarmsNum</b>	Number of alarms configured in the project	R	u.int
<b>SYS_RecipeStructsNum</b>	Number of recipe structures configured in the project	R	u.int
<b>SYS_DataLogsNum</b>	Number of Datalog buffers configured in the project	R	u.int
<b>SYS_UsersNum</b>	Number of users configured in the project or created at runtime	R	u.int
<b>SYS_PagesNum</b>	Number of pages configured in the project	R	u.int
<b>SYS_TimersNum</b>	Number of timers configured in the project	R	u.int
<b>SYS_LanguagesNum</b>	Number of languages configured	R	u.int
<b>SYS_LanguageName_&lt;LangId&gt;</b>	Language ID "<LangId>-esima" of the project. Esiste una variabile di questo tipo per ogni lingua di progetto esistente	R	string [32]

# CREW Manual

## ERRORS (Project errors during Runtime)

Name	Description	Access	Type
<b>SYS_LastErrorCode</b>	Code of last error displayed	R	u.long
<b>SYS_LastErrorText</b>	Description of last error displayed	R	string [256]
<b>SYS_LastErrorModule</b>	Code of runtime module linked to last error displayed	R	u.int

## STATUS - GENERAL - (General project status/system at Runtime)

Name	Description	Access	Type
<b>SYS_NumClients</b>	Number of remote clients currently connected by socket or HTTP	R	u.int
<b>SYS_Script</b>	Name of script currently running (empty if there are none)	R	string [32]
<b>SYS_CurrentPage</b>	Name of current FullScreen page	R	string [32]
<b>SYS_DateAndTime</b>	Date and time of system (t_time Windows format)	R / W	u.long

# CREW Manual

## STATUS - ALARMS - (Alarm status in Runtime)

Name	Description	Access	Type
<b>SYS_AlarmNum</b>	Number of alarms currently activated	R	u.int
<b>SYS_AlarmNotOff</b>	Number of alarms currently activated not yet reset	R	u.int
<b>SYS_AlarmNotAck</b>	Number of alarms (ISA) currently activated not yet identified	R	u.int
<b>SYS_HistoryNum</b>	Number of events in alarm log	R	u.long
<b>SYS_AlarmExist</b>	TRUE if at least one activated alarm exists	R	bool
<b>SYS_AlarmIsaExist</b>	TRUE if at least one activated ISA alarm exists	R	bool
<b>SYS_AlarmEventExist</b>	TRUE if at least one activated "simple" alarm exists	R	bool
<b>SYS_HistoryWarning</b>	TRUE if the alarm log has exceeded the set safety threshold	R	bool
<b>SYS_HistoryFull</b>	TRUE if the alarm log has reached its maximum capacity	R	bool
<b>SYS_AlarmFull</b>	TRUE if the activated alarm buffer has reached its maximum capacity	R	bool

## STATUS - RECIPES - (Status of recipes in Runtime)

Name	Description	Access	Type
<b>SYS_RecipesNum</b>	Total number of existing recipes, regardless of the type	R	u.int
<b>SYS_RecipeNum_&lt;StructId&gt;</b>	Number of type of recipes <StructId> currently existing There is a TAG for every structure configured in the project	R	u.int

# CREW Manual

STATUS - USERS & PASSWORD - (Status of users and passwords in Runtime)

Name	Description	Access	Type
<b>SYS_UserName</b>	Current user ID	R	string [32]
<b>SYS_UserLevelVisualize</b>	Protection level of current display	R	u.int
<b>SYS_UserLevelInteract</b>	Protection level of current interaction	R	u.int

STATUS - LANGUAGES - (Status of project languages in Runtime)

Name	Description	Access	Type
<b>SYS_LanguageName</b>	Name of current language	R	string [32]
<b>SYS_Languageld</b>	ID of current language	R	u.int

STATUS - TIMERS - (Status of project timers in Runtime)

Name	Description	Access	Type
<b>SYS_TimerProgress_&lt;TimerId&gt;</b>	The current value of the timer counter <TimerId>. There is a TAG for every timer configured in the project	R / W	u.long
<b>SYS_TimerLimit_&lt;TimerId&gt;</b>	The threshold value of the timer <TimerId>. There is a TAG for every timer configured in the project	R / W	u.long

# CREW Manual

COMMUNICATION - (Information on drivers and communication lines used in the project)

Name	Description	Access	Type
SYS_NumLinesTotal	Number of communication lines existing on the machine established by the project	R	u.int
SYS_NumLinesUsed	Number of communication lines actually used by the project	R	u.int
SYS_LineState_<LinIdx>	TRUE if the "<LinIdx>-esima" communication line is in use. There is a variable of this type for each communication line existing on the machine established by the project	R	bool
SYS_LineName_<LinIdx>	Descriptive name of the "<LinIdx>-esima" communication line. Corresponds to the name visible to the user of the configurator ("COM #", "ETH #", "Codesys"). There is a variable for each communication line existing on the machine established by the project	R	string [16]
SYS_NumDrivers	Number of communication drivers included in the project	R	u.int
SYS_NumDrivers_<LinIdx>	Number of drivers connected to the "<LinIdx>-esima" communication line. There is a variable for each communication line existing on the machine established by the project	R	u.int
SYS_DriverName_<DriverIdx>	<DriverIdx>-esimo driver ID There is a variable of this type for each driver included in the project	R	string [32]
SYS_DriverState_<DriverIdx>	Status (error) of <DriverIdx>-esimo driver. There is a variable of this type for each driver included in the project	R	string [32]
SYS_DriverVersion_<DriverIdx>	Version of <DriverIdx>-esimo driver. There is a variable of this type for each driver included in the project	R	string [16]
SYS_DriverAddress_<DriverIdx>	Terminal IP for <DriverIdx>-esimo driver. There is a variable of this type for each driver included in the project	R	string [16]



# CREW Manual



Note:

Type: type of variable

- int: 16-bit signed integers
- u.int: 16-bit unsigned integers
- long: 32-bit signed integers
- u.long: 32-bit unsigned integers
- bool: Boolean value
- string [x]: 16-bit string with x characters (all the strings are "unicode")



Note:

Access: type of access allowed

- R : read
- W : write

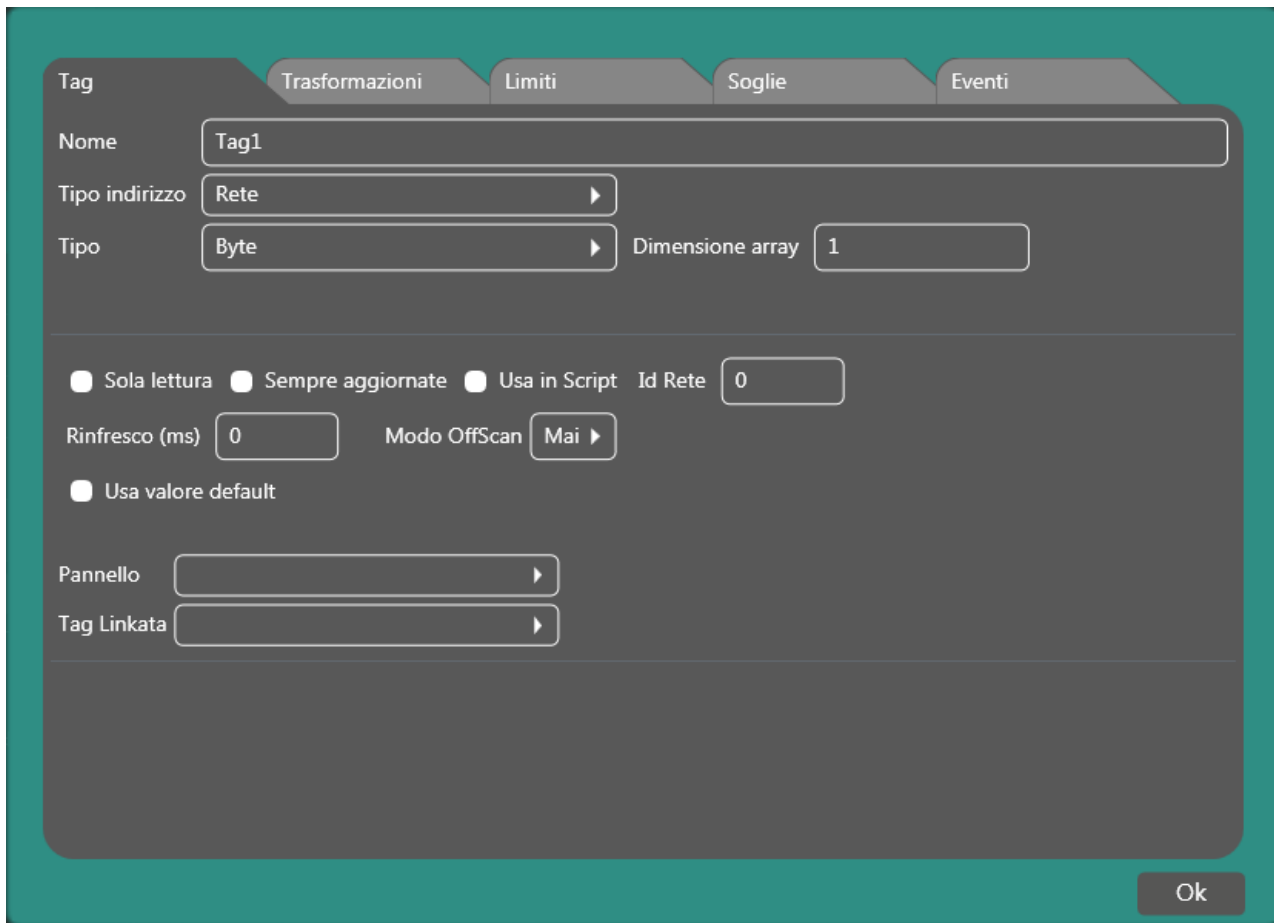


Note: Writing on read-only variables is not blocked, but the written data has no effect.

# CREW Manual

## Network Tag

"Network" Address Type (Network Tag) :



The screenshot shows a configuration window for a "Network Tag". The window has a title bar with tabs: "Tag", "Trasformazioni", "Limiti", "Soglie", and "Eventi". The "Tag" tab is active. The configuration fields are as follows:

- Nome:** Tag1
- Tipo indirizzo:** Rete
- Tipo:** Byte
- Dimensione array:** 1
- Sola lettura:**
- Sempre aggiornate:**
- Usa in Script:**
- Id Rete:** 0
- Rinfresco (ms):** 0
- Modo OffScan:** Mai
- Usa valore default:**
- Pannello:** (empty dropdown)
- Tag Linkata:** (empty dropdown)

An "Ok" button is located at the bottom right of the dialog.

Network tags can be used in a "Panel network" project, by all the terminals that make up the network.

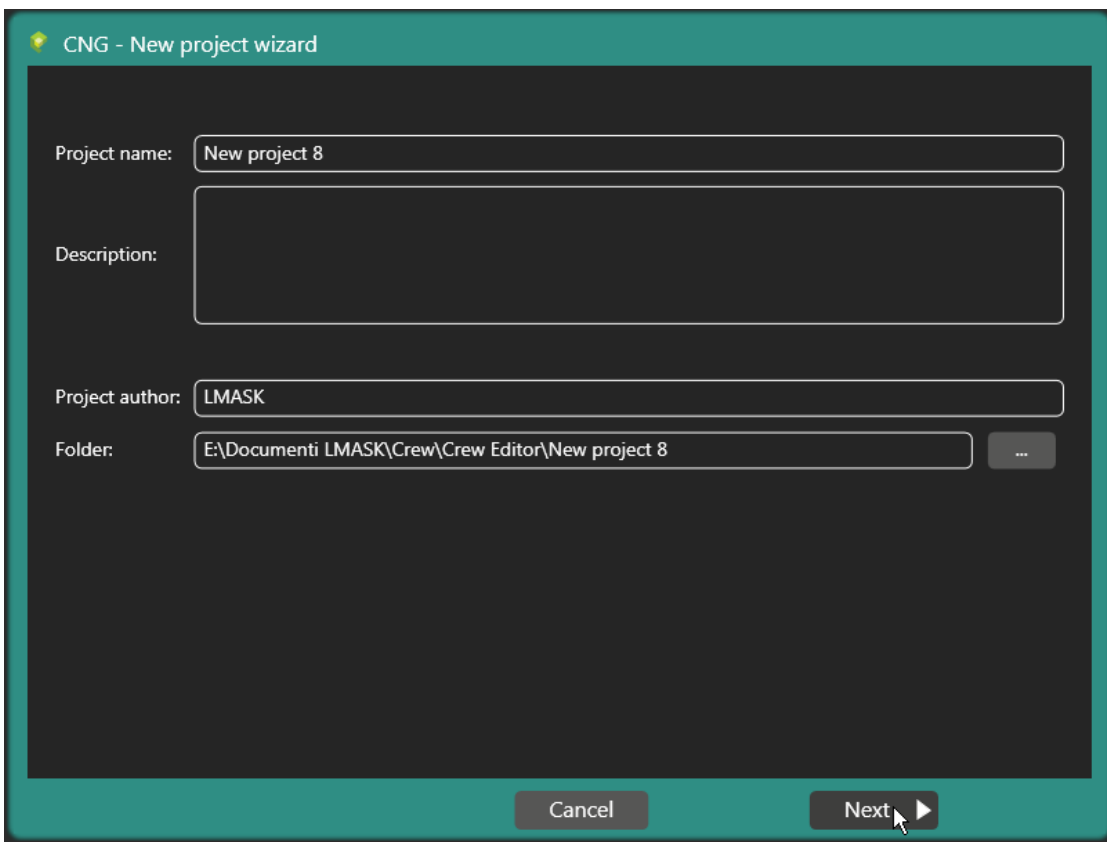
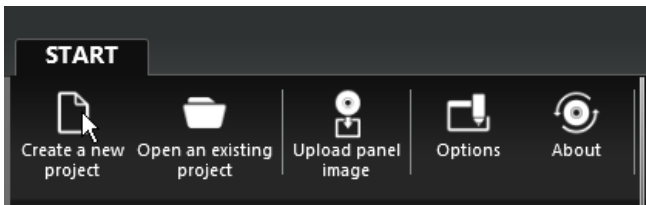
# CREW Manual

## Panel network

To create a panel network it is necessary to firstly create a project with at least 2 terminals and make the variables available on the network ([Network Tag](#)).

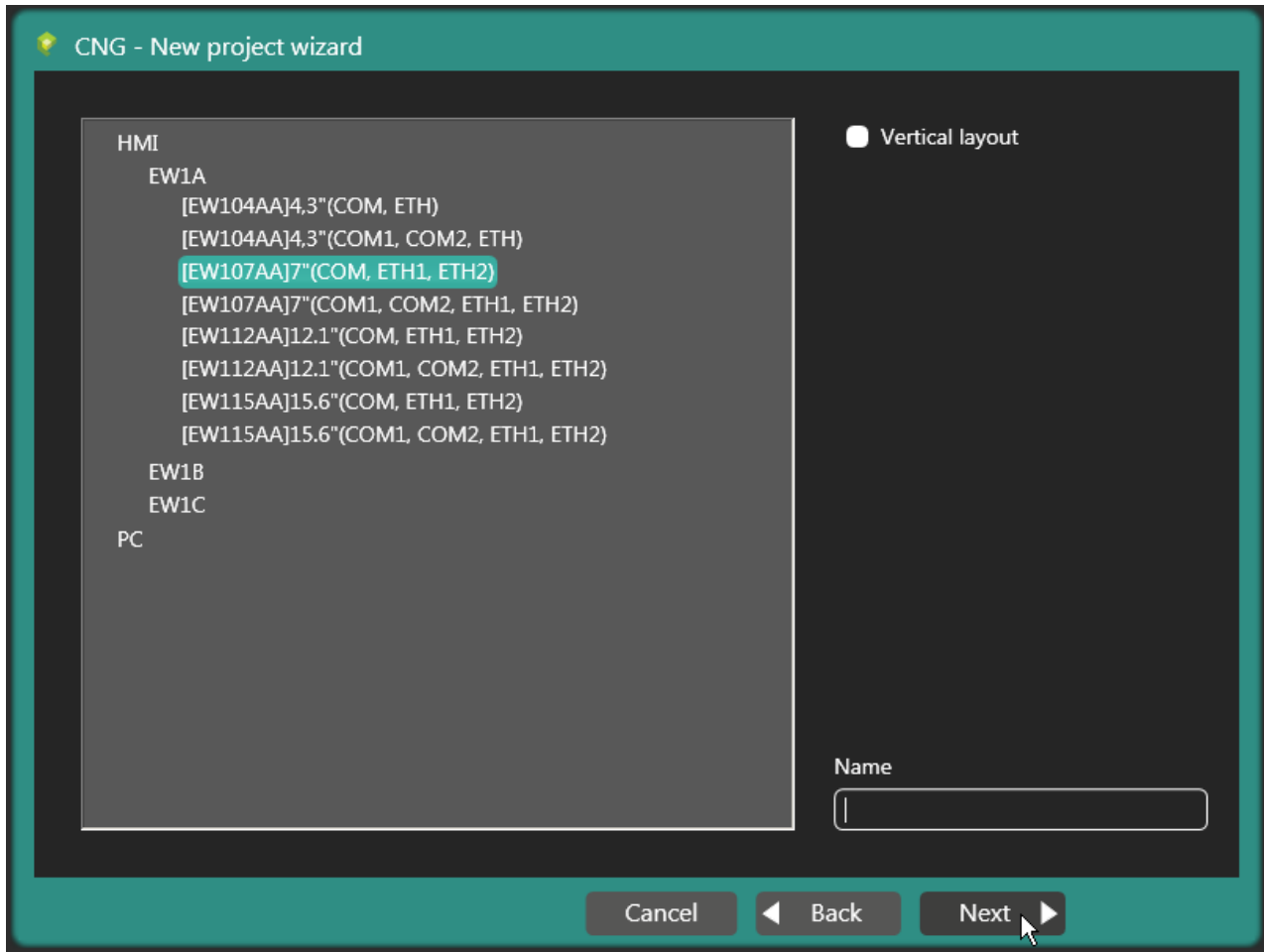
The procedure to create a panel network is described in detail below:

Choose "Create new project" and click "Next" :



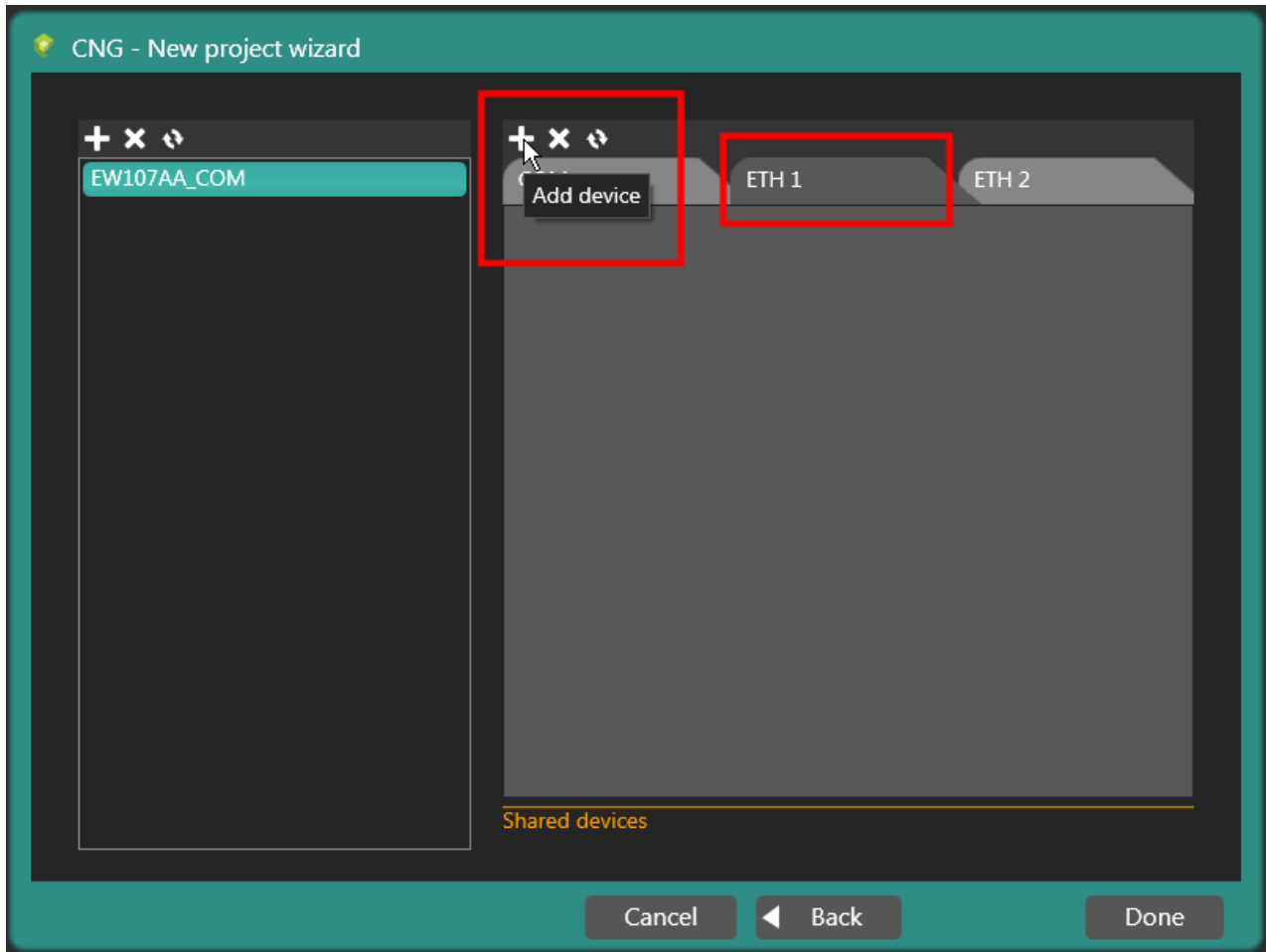
# CREW Manual

Select the first panel to be inserted in the project.



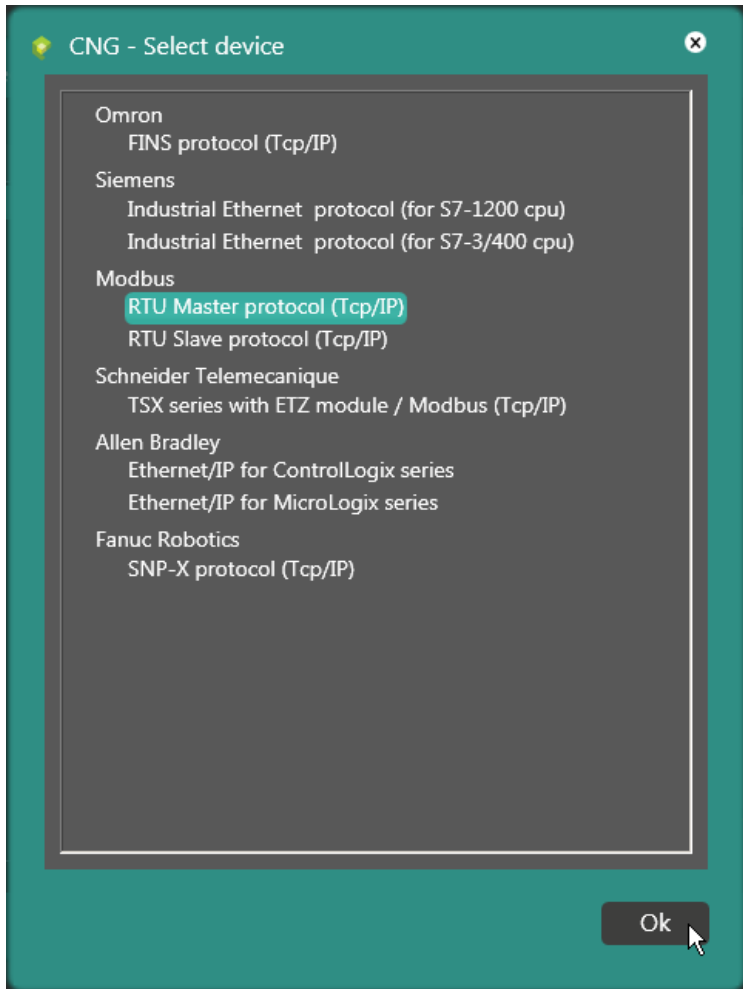
# CREW Manual

Select the Ethernet port for connection to the device (PLC), then click the appropriate key to select the device to be connected.



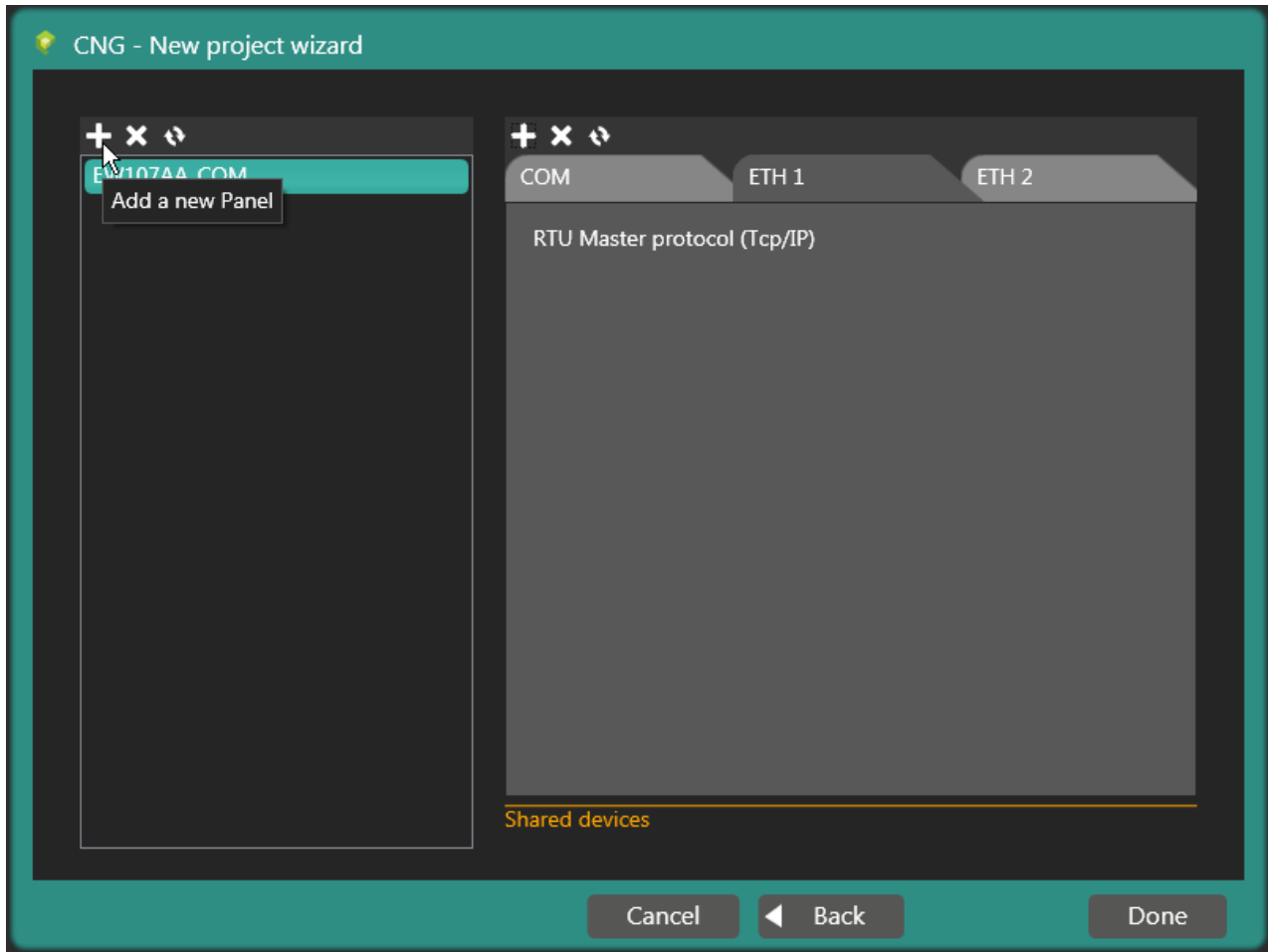
# CREW Manual

After selecting the device, click "OK" to confirm.



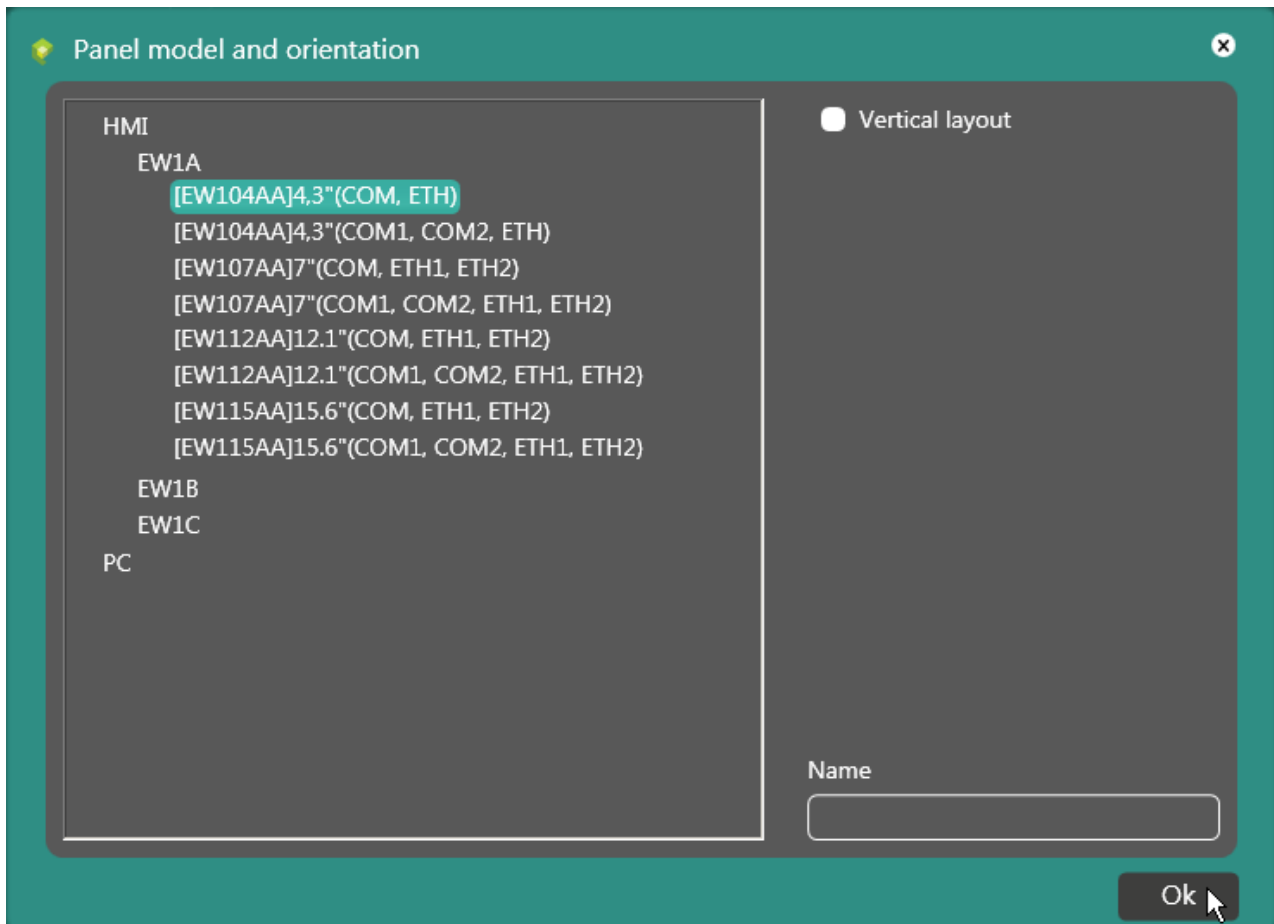
# CREW Manual

Click "Add new panel" on the window that appears.



# CREW Manual

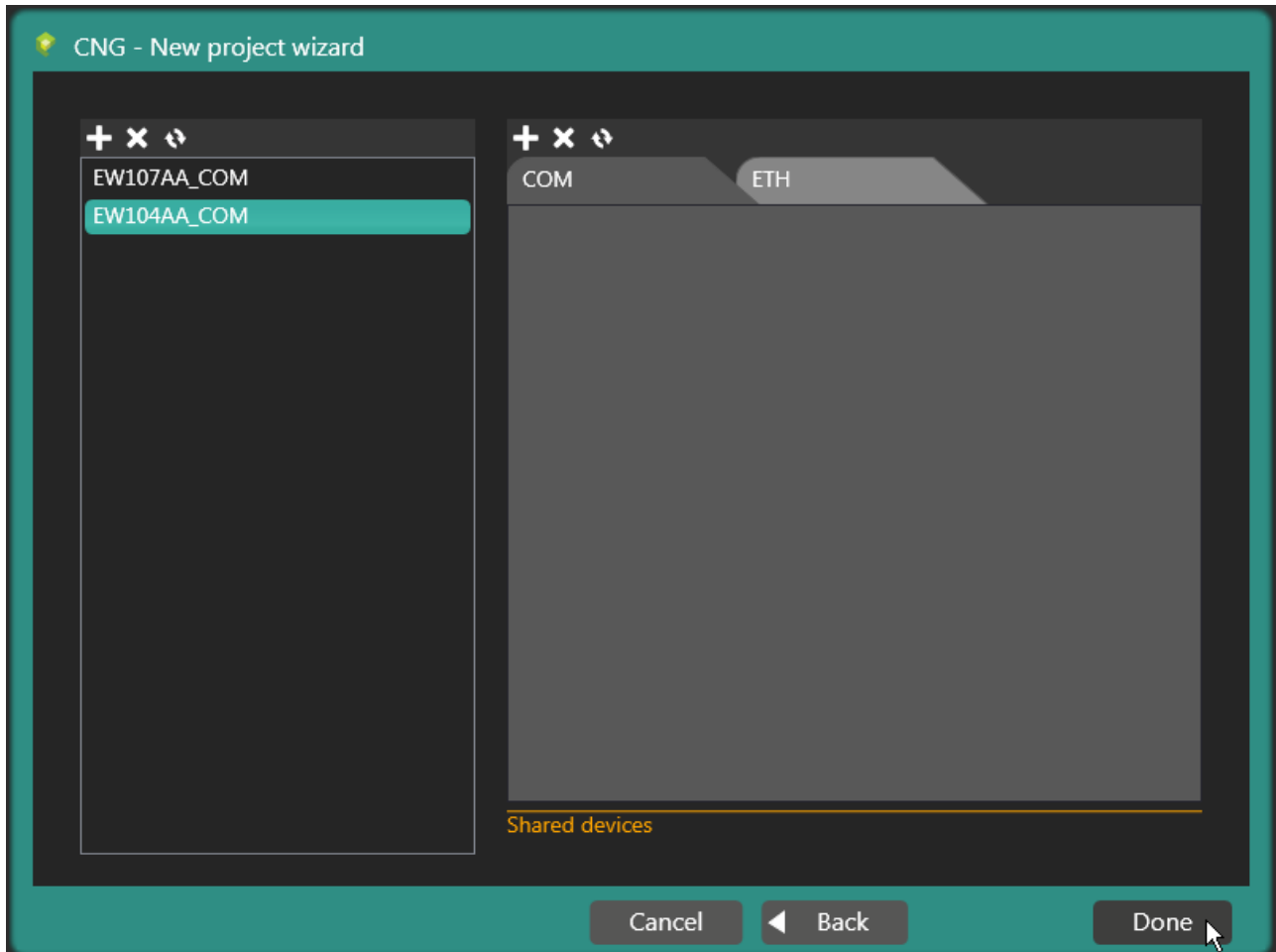
Select the second panel to be added.





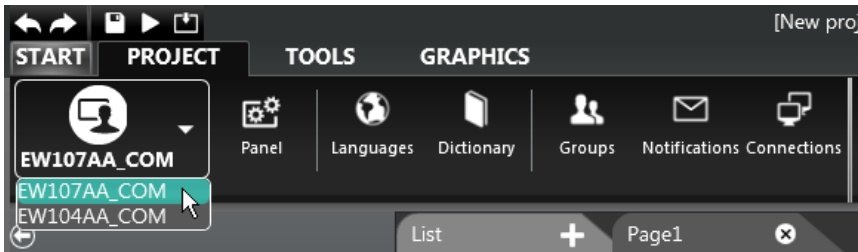
# CREW Manual

Click the “Done” key.

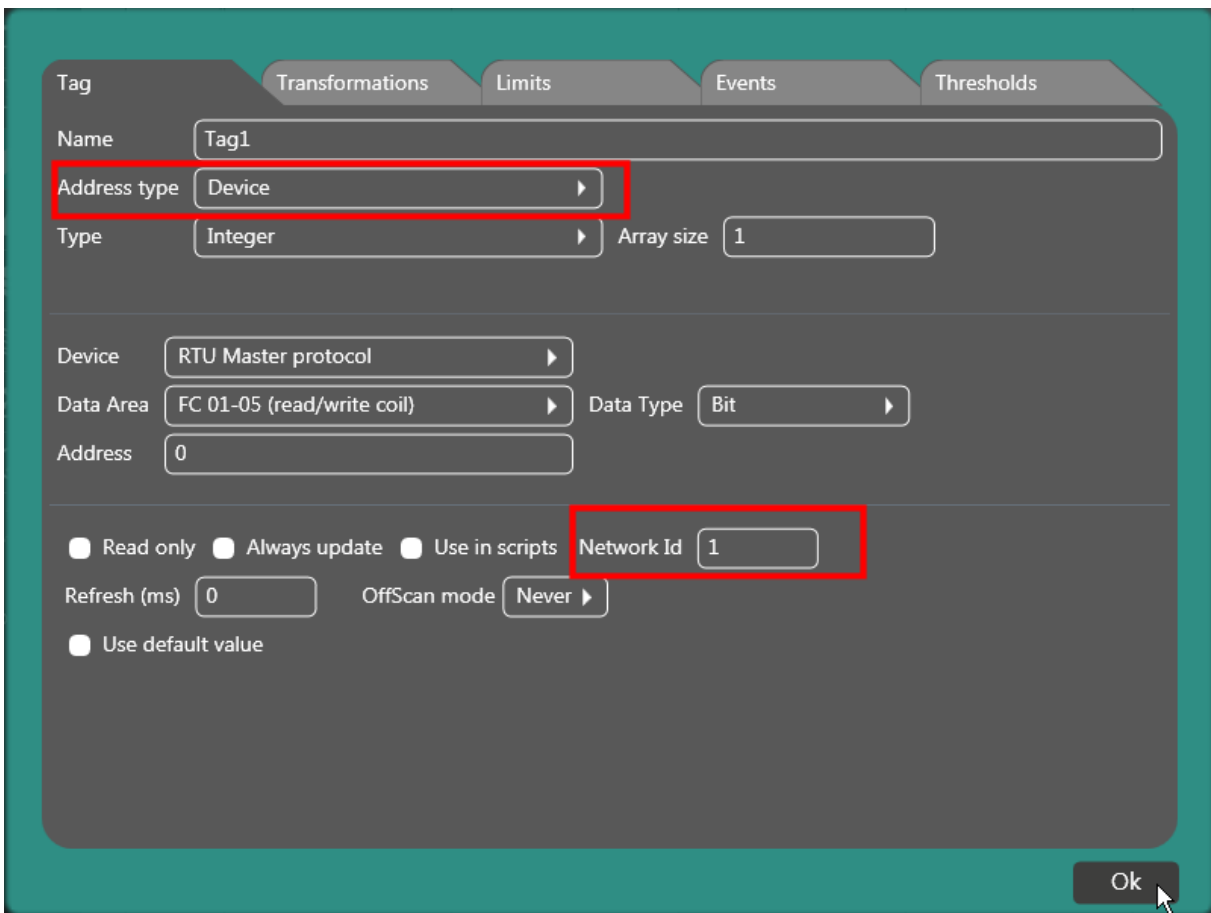


# CREW Manual

From the "Project" menu select the first panel (EW107).

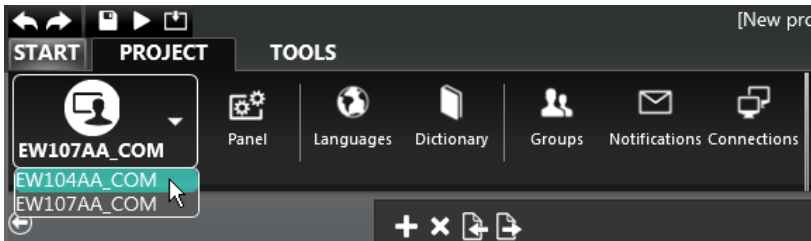


Then, after selecting the "Tags" menu, add a "Device" type of variable and set the value in the "Network ID" editable field (entering, for example, a value of "1").

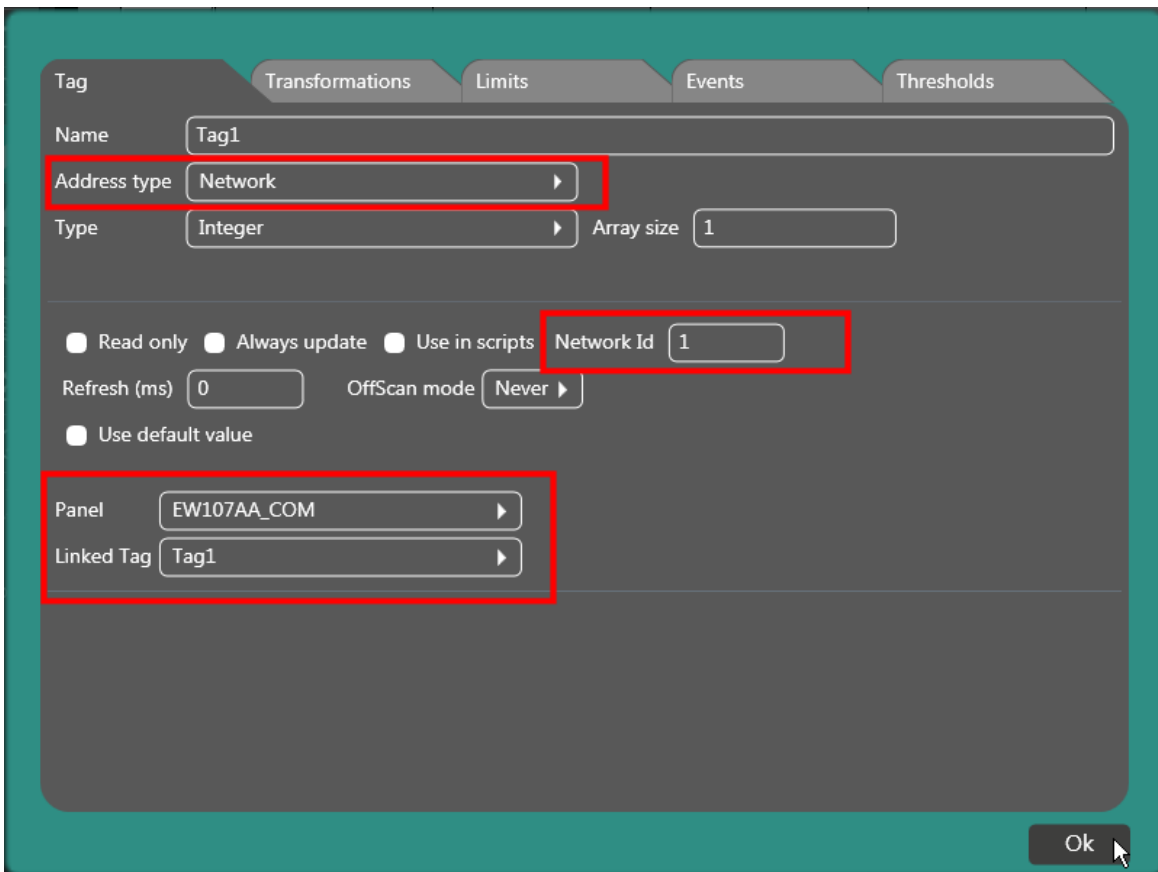


# CREW Manual

Now select the second panel (EW104) in the project.



Set the editable fields as shown in the image, to add a "Network" type of tag.



The "Panel network" is now complete and the EW104 panel can read the "Device" variable on the EW107 panel.

# CREW Manual

## Indexed Tags

The “Indexed” tags make it possible to view the value of each variable in a single field. The choice of tag whose value needs to be displayed by the “Indexed Variable” is managed by the “Index Variable”.

## Tags Grid Filters

The user can enable or disable viewing filters available on each column of the Tags Grid to make only certain types of values appear in the table.

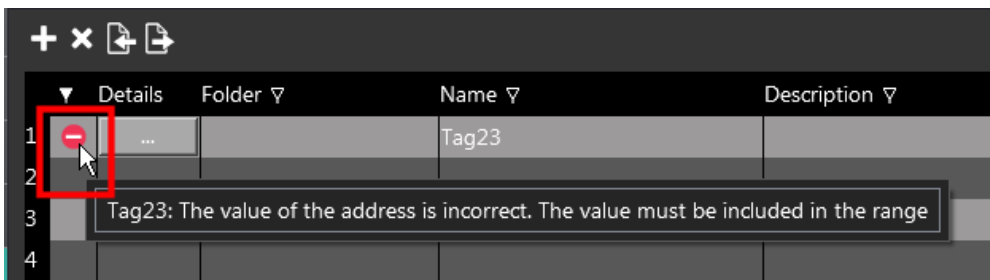
The columns to which display filters can be applied are the following:

- Validation
- Folder
- Name
- Description
- Type
- Address type
- Provider
- Data area
- Data type
- Address

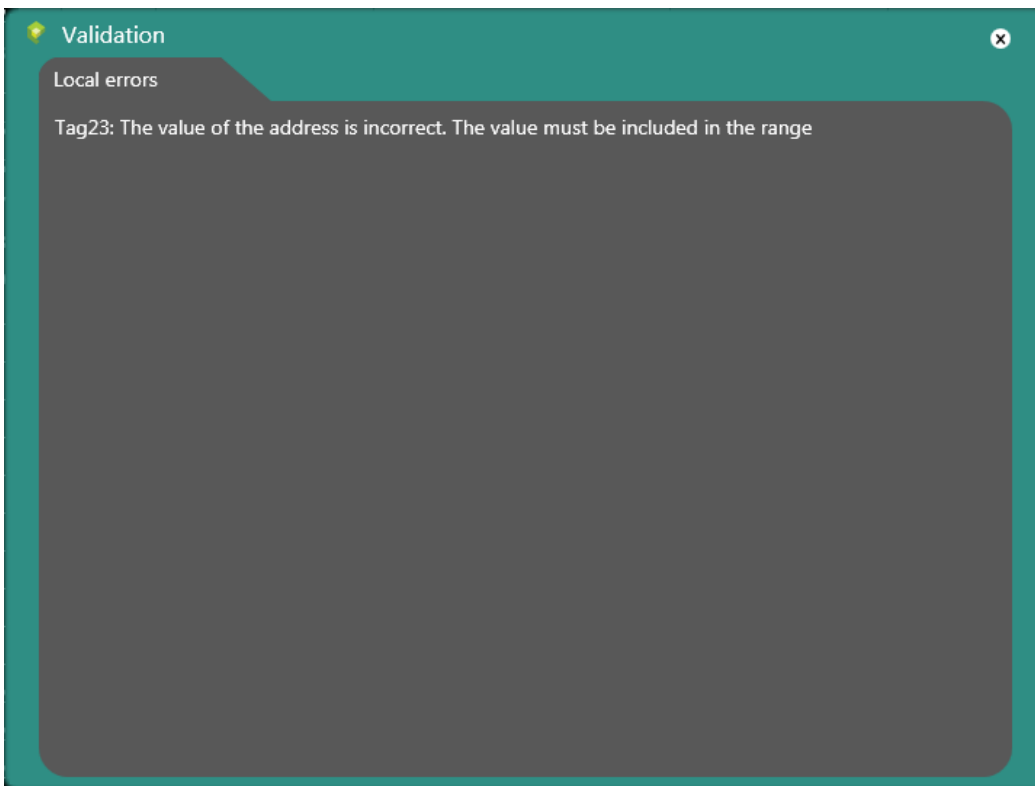
# CREW Manual

## Validation

The “Validation” column reports any association errors regarding the value of the variables. A red symbol next to the progressive number makes it possible to immediately see when there are any errors.

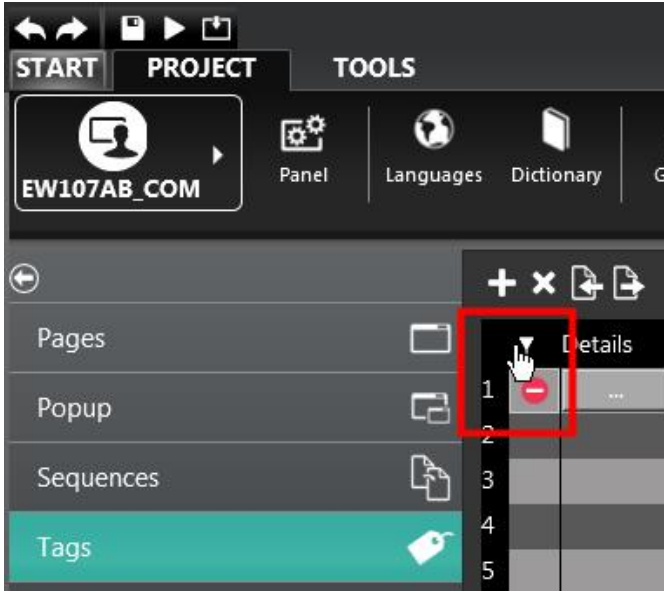


Click the error symbol to view the detailed error window.

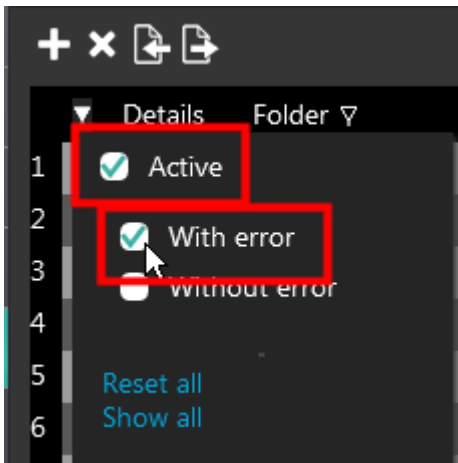


# CREW Manual

In the “Validation” column, enable or disable the error filter to view all of the tags with or without errors.

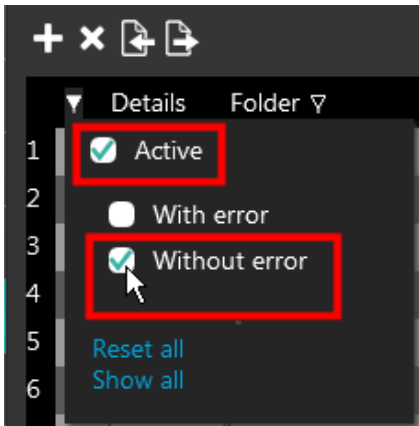


Enable the "With Errors" filter to display tags with errors.



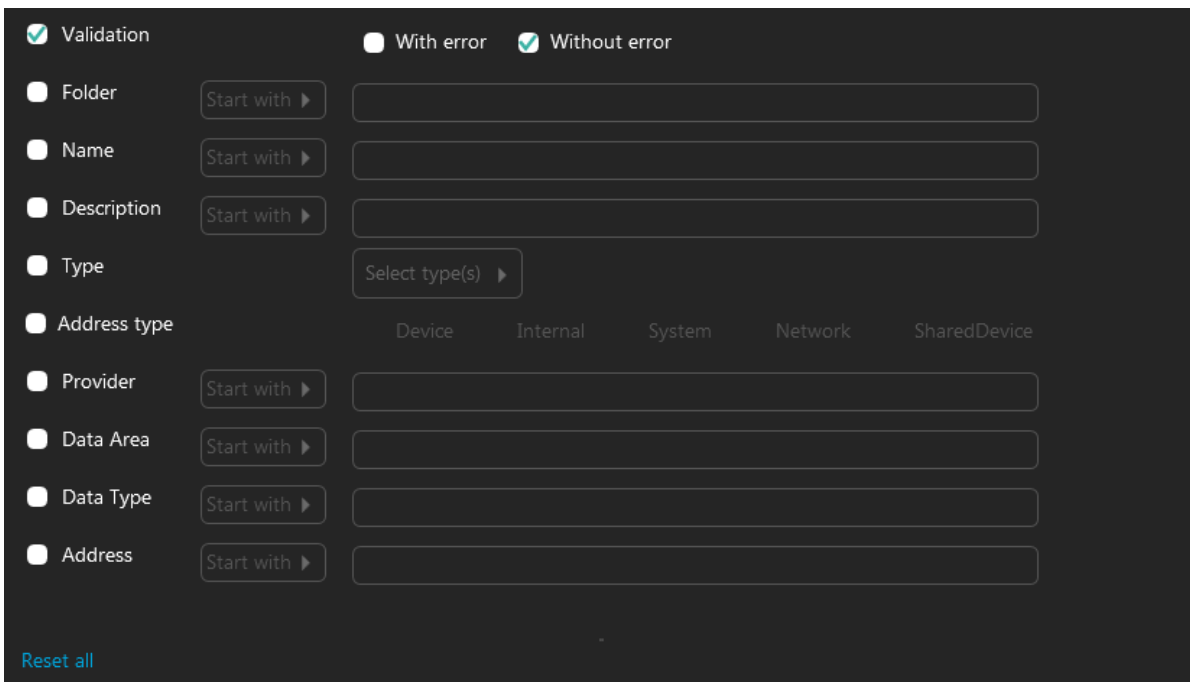
# CREW Manual

Enable the "Without Errors" filter to display tags without errors.



The "Reset all" option deletes all enabled filters.

The "Show all" option displays in a single window all options to which display filters can be applied.

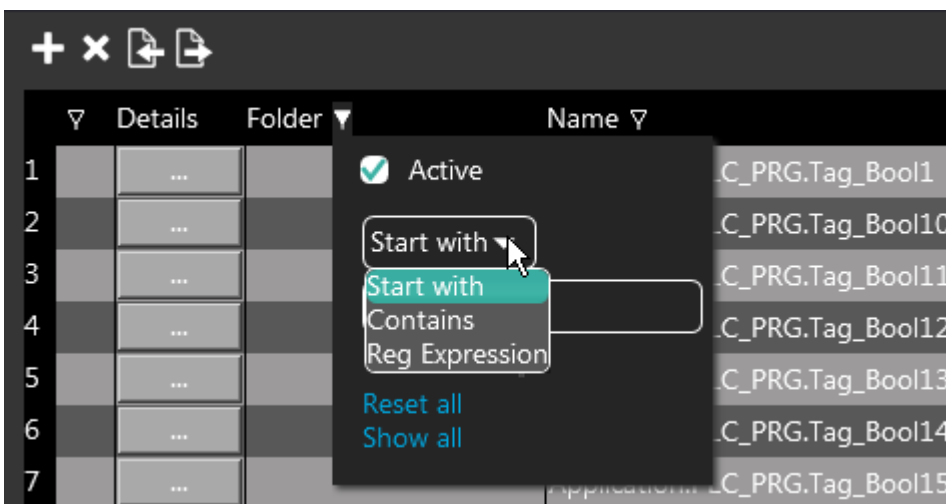


# CREW Manual

## Folder

In the "Folder" it is possible to view the folders contained in CREW according to one of the following viewing filters:

- **Beginning with:** the folders with names that start with the letter chosen by the user are displayed.
- **Containing:** the folders with names that contain the word chosen by the user are displayed.
- **Reg Expression**



The "Reset all" option deletes all enabled filters.



# CREW Manual

The "Show all" option displays in a single window all options to which display filters can be applied.

Validation
 

With error
  Without error

Folder

Start with ▶

Name

Start with ▶

Description

Start with ▶

Type

Select type(s) ▶

Address type

Device

Internal

System

Network

SharedDevice

Provider

Start with ▶

Data Area

Start with ▶

Data Type

Start with ▶

Address

Start with ▶

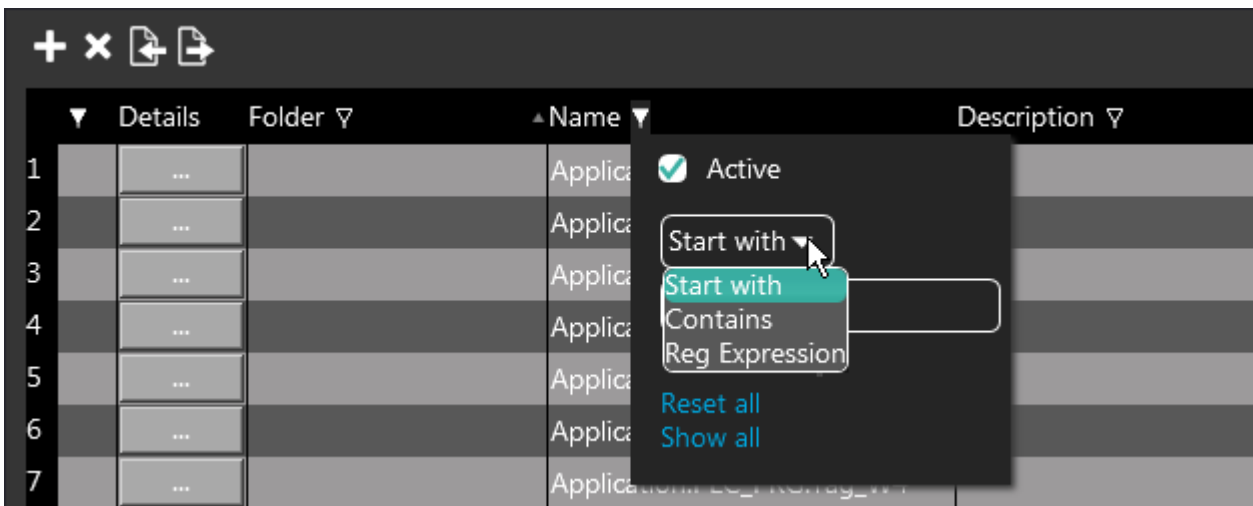
[Reset all](#)

# CREW Manual

## Name

In the "Name" column it is possible to apply one of the following viewing filters:

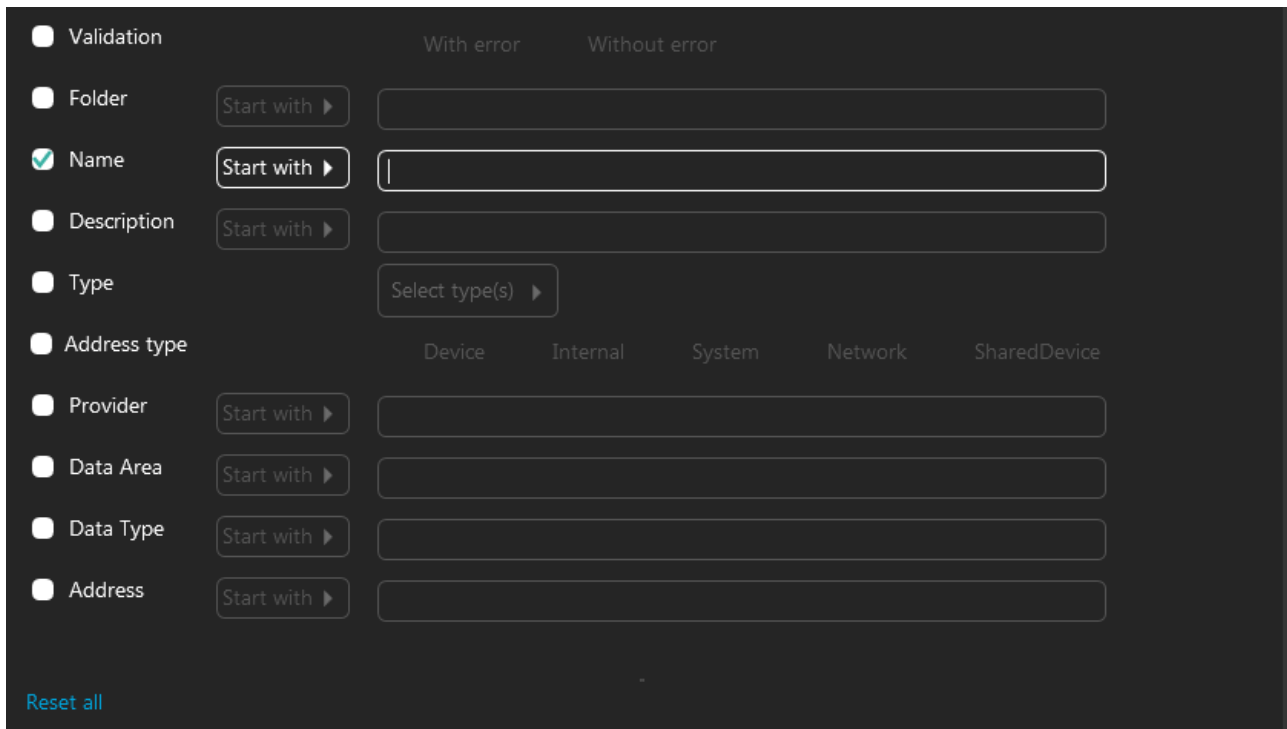
- **Beginning with:** the tags with names that start with the letter chosen by the user are displayed.
- **Containing:** the tags with names that contain the word chosen by the user are displayed.
- **Reg Expression**



The "Reset all" option deletes all enabled filters.

# CREW Manual

The "Show all" option displays in a single window all options to which display filters can be applied.



The screenshot shows a dark-themed filter configuration panel. On the left, a list of filter categories is shown with checkboxes: Validation, Folder, Name (checked), Description, Type, Address type, Provider, Data Area, Data Type, and Address. Each category has a 'Start with' button and a corresponding input field. The 'Name' filter is active, with its input field containing a vertical bar. The 'Address type' filter has radio buttons for Device, Internal, System, Network, and SharedDevice. At the bottom left, there is a 'Reset all' link.

Filter Category	Start with	Value
Validation		With error / Without error
Folder	Start with ▶	
Name	Start with ▶	
Description	Start with ▶	
Type	Select type(s) ▶	
Address type		Device / Internal / System / Network / SharedDevice
Provider	Start with ▶	
Data Area	Start with ▶	
Data Type	Start with ▶	
Address	Start with ▶	

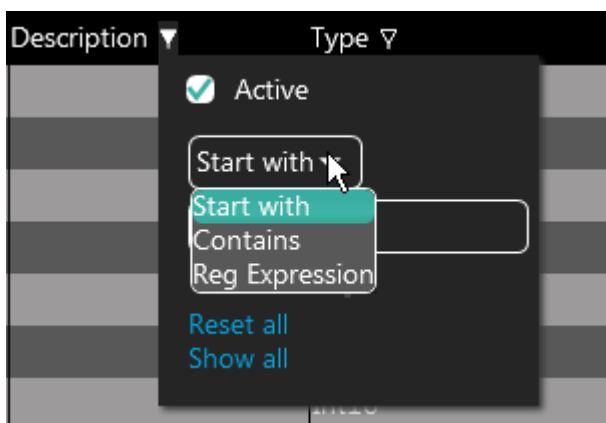
[Reset all](#)

# CREW Manual

## Description

In the "Description" column it is possible to apply one of the following viewing filters:

- Beginning with: the folders with names that start with the letter chosen by the user are displayed.
- Containing: the folders with names that contain the word chosen by the user are displayed.
- Reg Expression



The "Reset all" option deletes all enabled filters.

# CREW Manual

The "Show all" option displays in a single window all options to which display filters can be applied.

Validation

Folder

Name

Description

Type

Address type

Provider

Data Area

Data Type

Address

With error
Without error

Start with ▶

Start with ▶

Start with ▶

Select type(s) ▶

Device
Internal
System
Network
SharedDevice

Start with ▶

Start with ▶

Start with ▶

Start with ▶

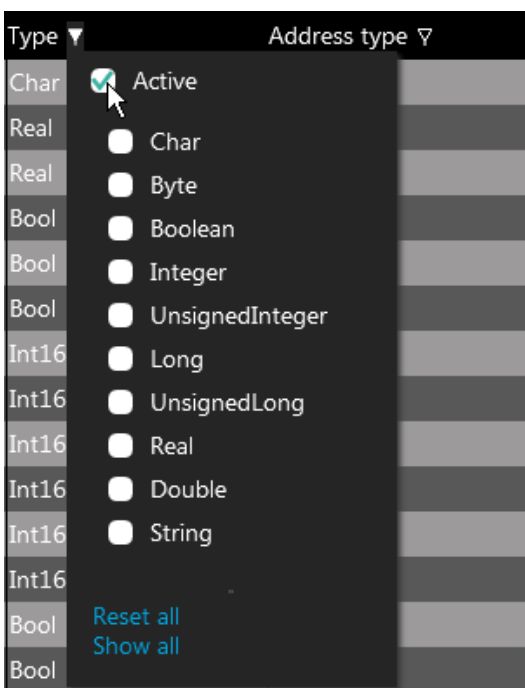
[Reset all](#)

# CREW Manual

## Type

In the "Type" column it is possible to apply the display filter based on the data type that the tag is destined to contain. The data types are listed in the "Type" table ([Data Type](#)).

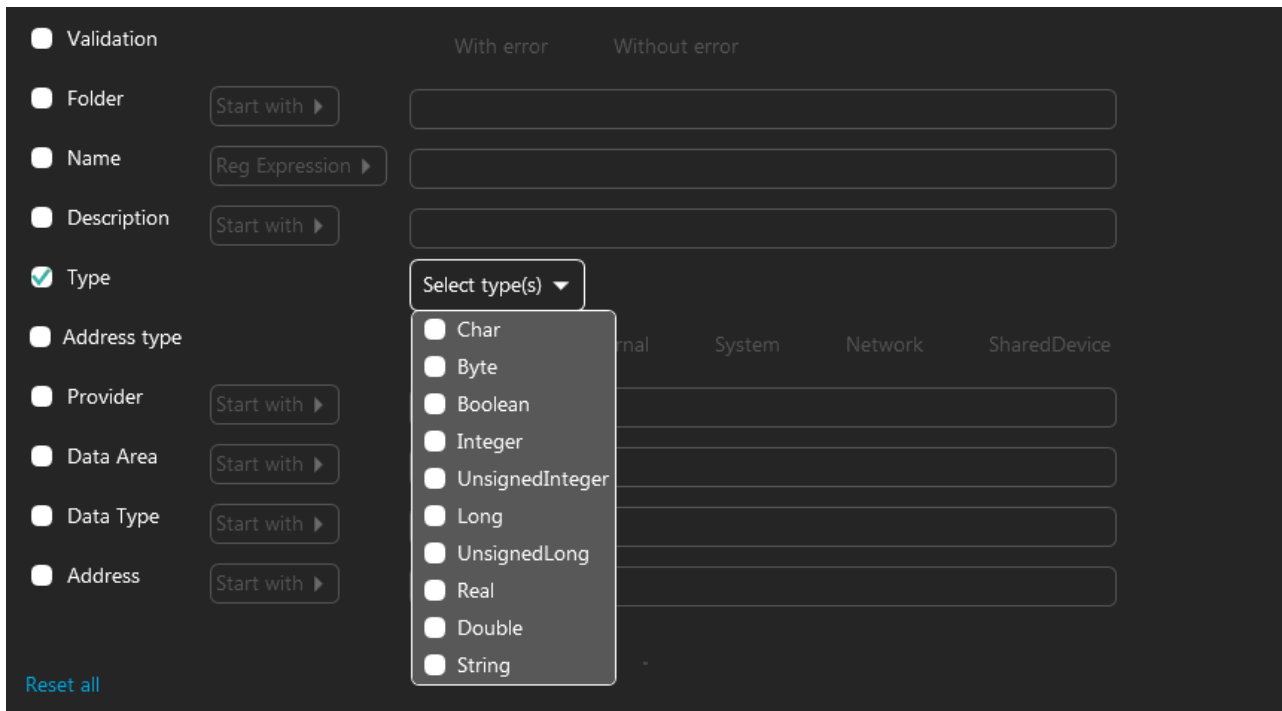
It is possible to choose from the following viewing filters:



The "Reset all" option deletes all enabled filters.

# CREW Manual

The "Show all" option displays in a single window all options to which display filters can be applied.

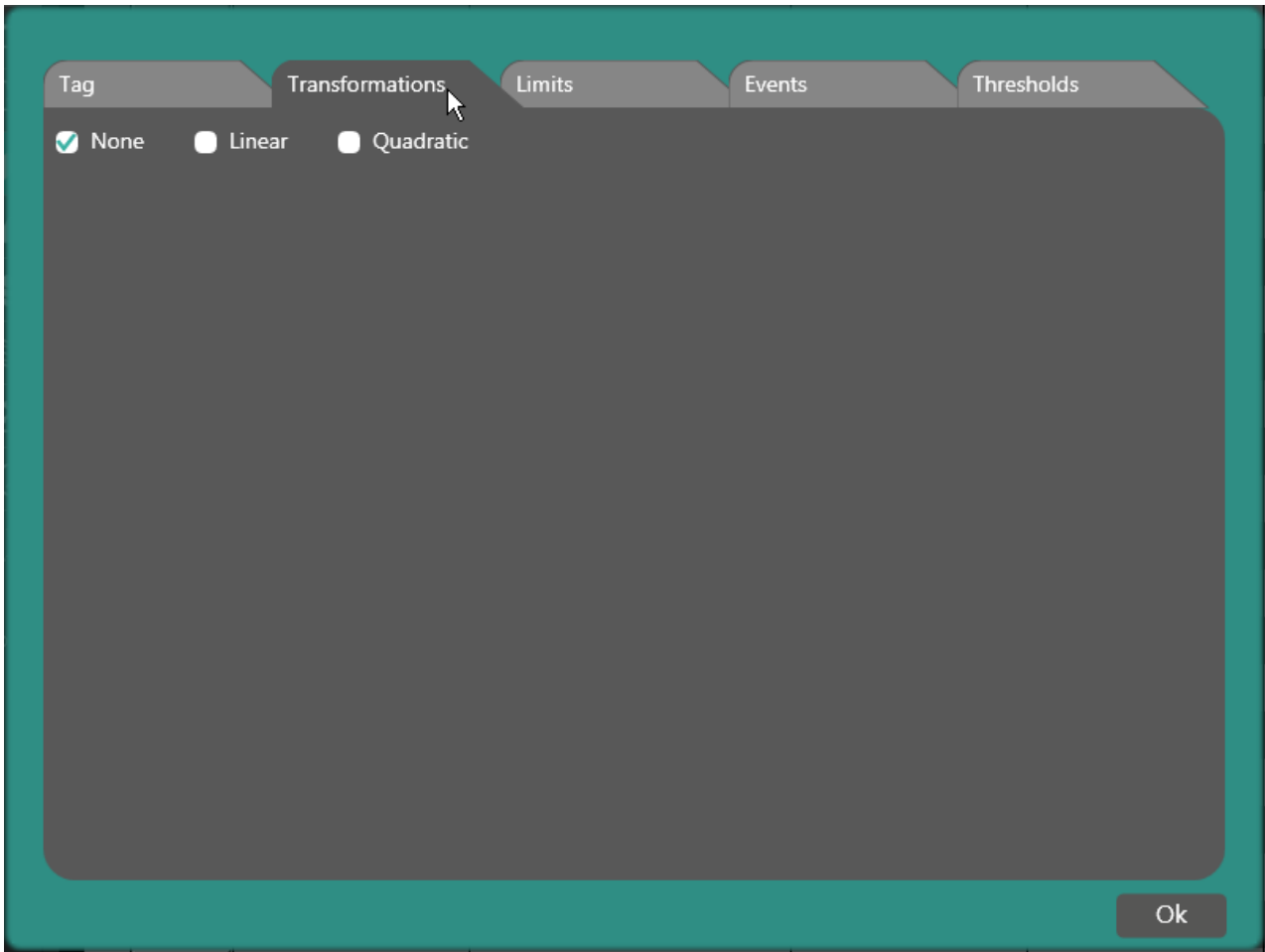


The screenshot shows a dark-themed configuration window for filters. On the left, a list of filter categories is shown with radio buttons: Validation, Folder, Name, Description, Type (checked), Address type, Provider, Data Area, Data Type, and Address. Each category has a 'Start with' button. To the right, there are input fields for each category. The 'Type' dropdown menu is open, showing a list of data types with radio buttons: Char, Byte, Boolean, Integer, UnsignedInteger, Long, UnsignedLong, Real, Double, and String. At the top right, there are radio buttons for 'With error' and 'Without error'. At the bottom left, there is a 'Reset all' link.

# CREW Manual

## Transformations

The second Tags Editor window is the "Transformations" option.



The value of the external numeric tag is always calculated by the system based on the rough value.

Often, in addition to standard conversions, a calculation must be made because the measurement units in which the rough value is expressed are different from those required for the value of the tag.

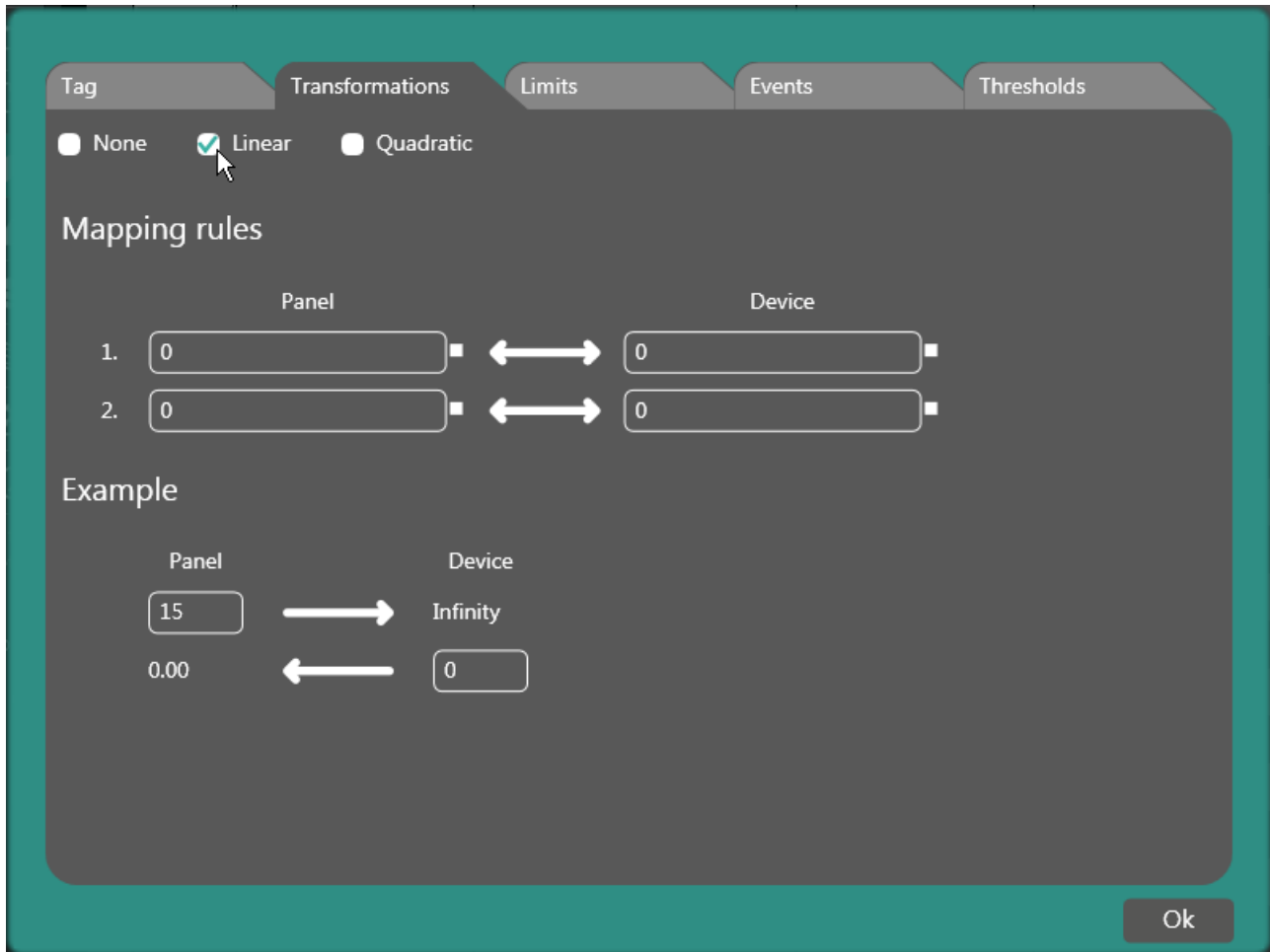
For example, the rough value is often expressed as an integer within the range of an analogue-digital converter, while the value of the tag is expressed in engineering units.



# CREW Manual

In this mask it is possible to choose the type of conversion required for the tag. The selectable conversions are: none, linear, quadratic.

## Linear transformation



Linear transformation entails definition of two pairs of values, each formed from the value of the tag and the corresponding rough value:

- Panel (P1, P2)
- Device (D1, D2)

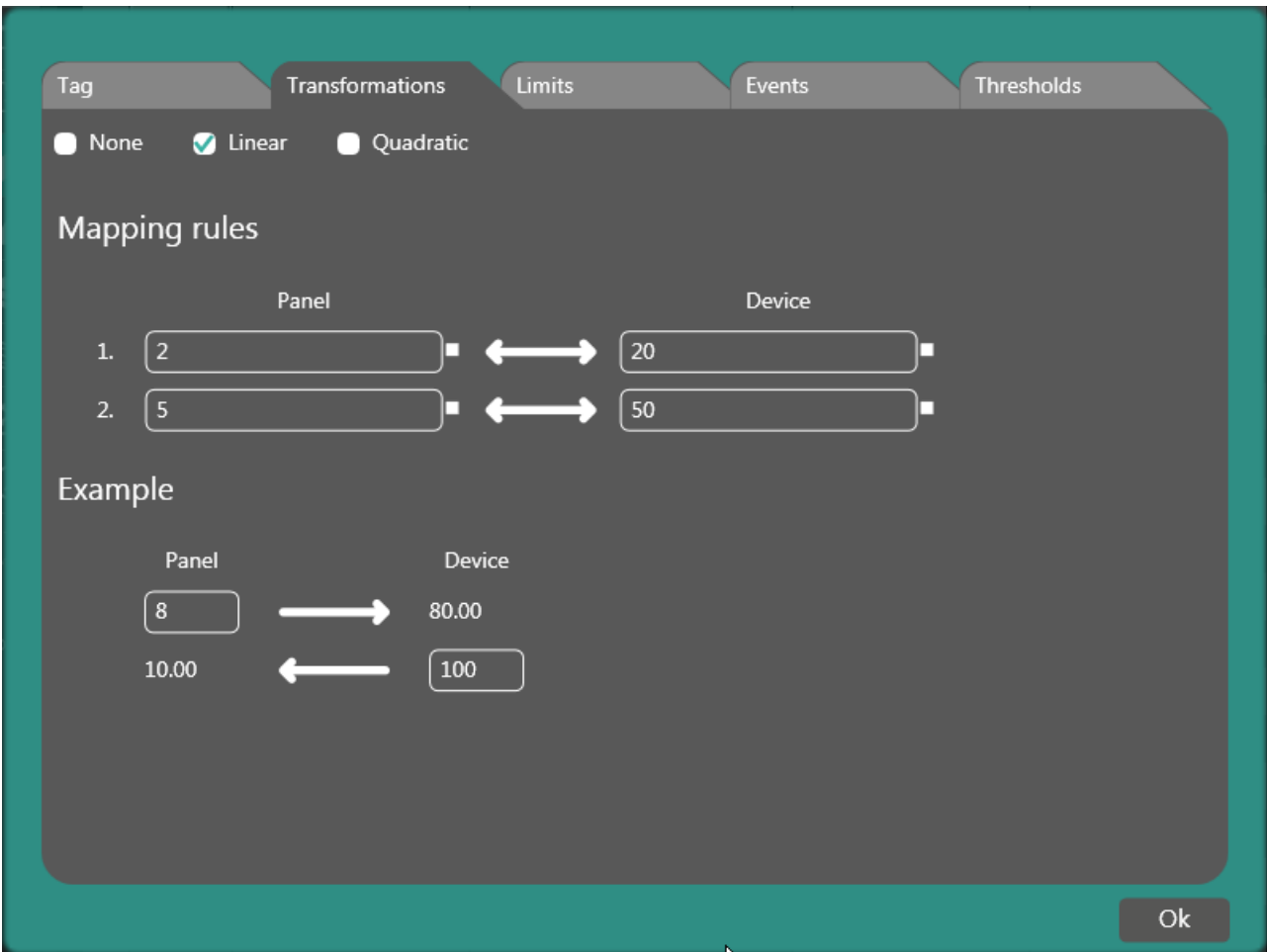
Where the values of the Panel are the rough values and the Device values are the corresponding "engineering" values.

# CREW Manual

The rough value of the Panel (X) and the corresponding value (Y) of the tags in the linear conversion are linked by the following equation:

$$\frac{Y - D1}{X - P1} = \frac{D2 - D1}{P2 - P1}$$

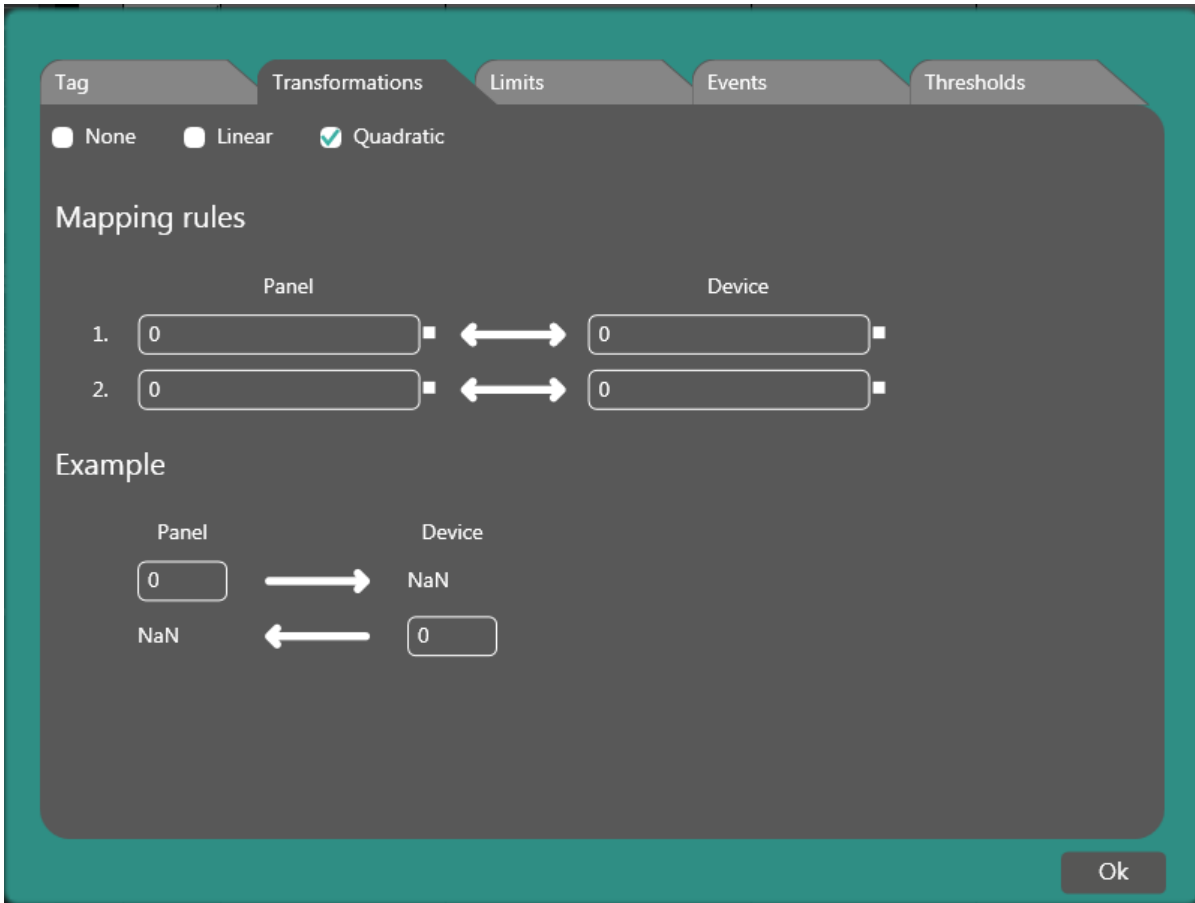
For example if you assign values 2 (P1) and 5 (P2) to the panel, and values 20 (D1) and 50 (D2) to the device (PLC), by setting a value of 8 on the panel, the device will have a value of 80. On the other hand, by setting a value of 100 on the device, we achieve a value of 10 on the panel, as shown below.



The screenshot shows the 'Transformations' configuration window for a tag. It has tabs for 'Tag', 'Transformations', 'Limits', 'Events', and 'Thresholds'. Under 'Transformations', the 'Linear' option is selected. The 'Mapping rules' section shows two points: 1. Panel: 2, Device: 20; 2. Panel: 5, Device: 50. The 'Example' section shows: Panel: 8, Device: 80.00; and Panel: 10.00, Device: 100. An 'Ok' button is at the bottom right.

# CREW Manual

## Quadratic transformation



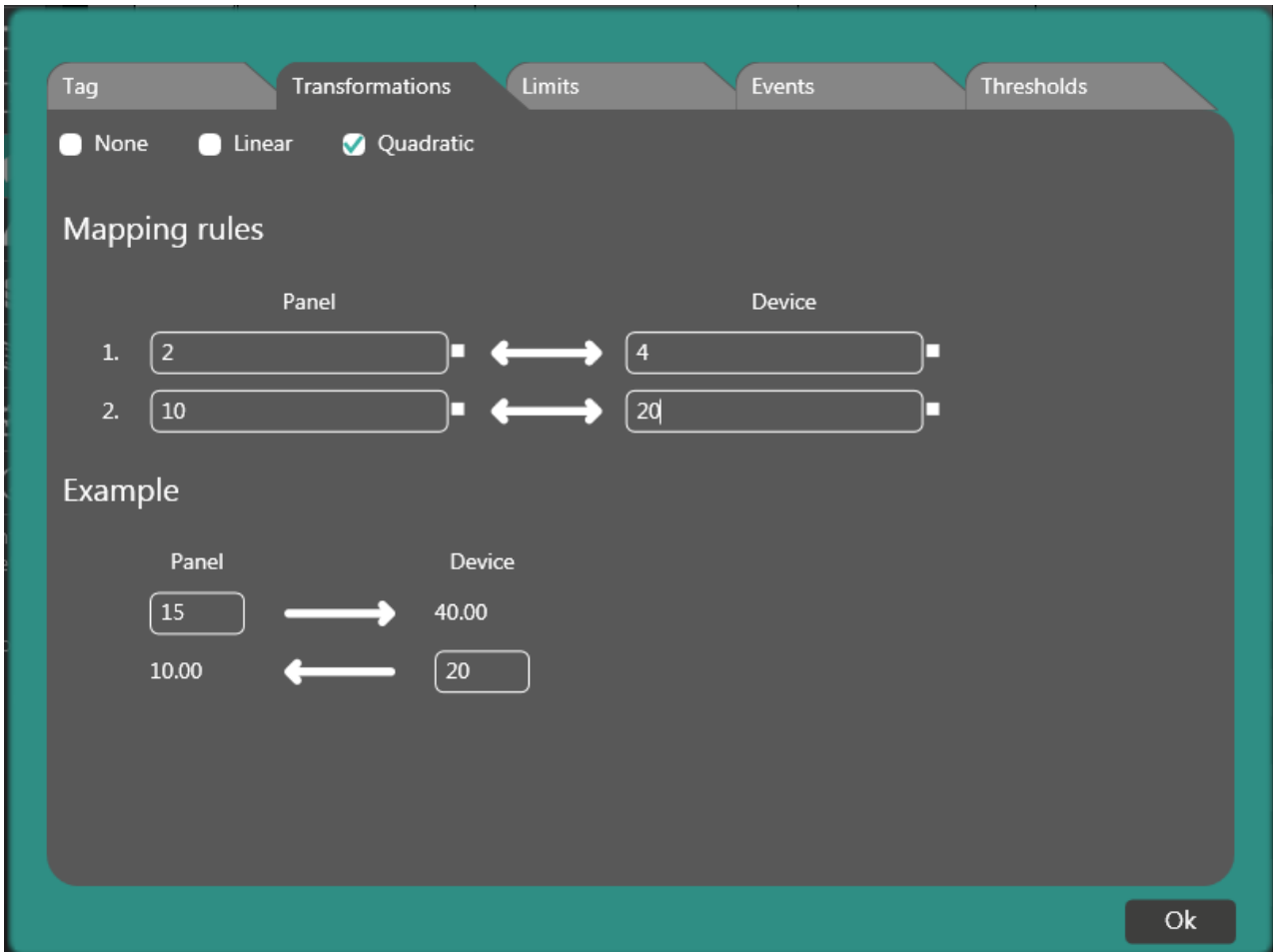
Quadratic conversion requires the same values, excluding Y1. Therefore, in quadratic transformation, the equation that links the rough value X and the Y value of the tag is:

$$\frac{Y^2}{X-P1} = \frac{D2^2}{P2-P1}$$

For example if you assign values 2 (P1) and 10 (P2) to the panel, and values 4 (D1) and 20 (D2) to the device (PLC), by setting a value of 15 on the panel, the

# CREW Manual

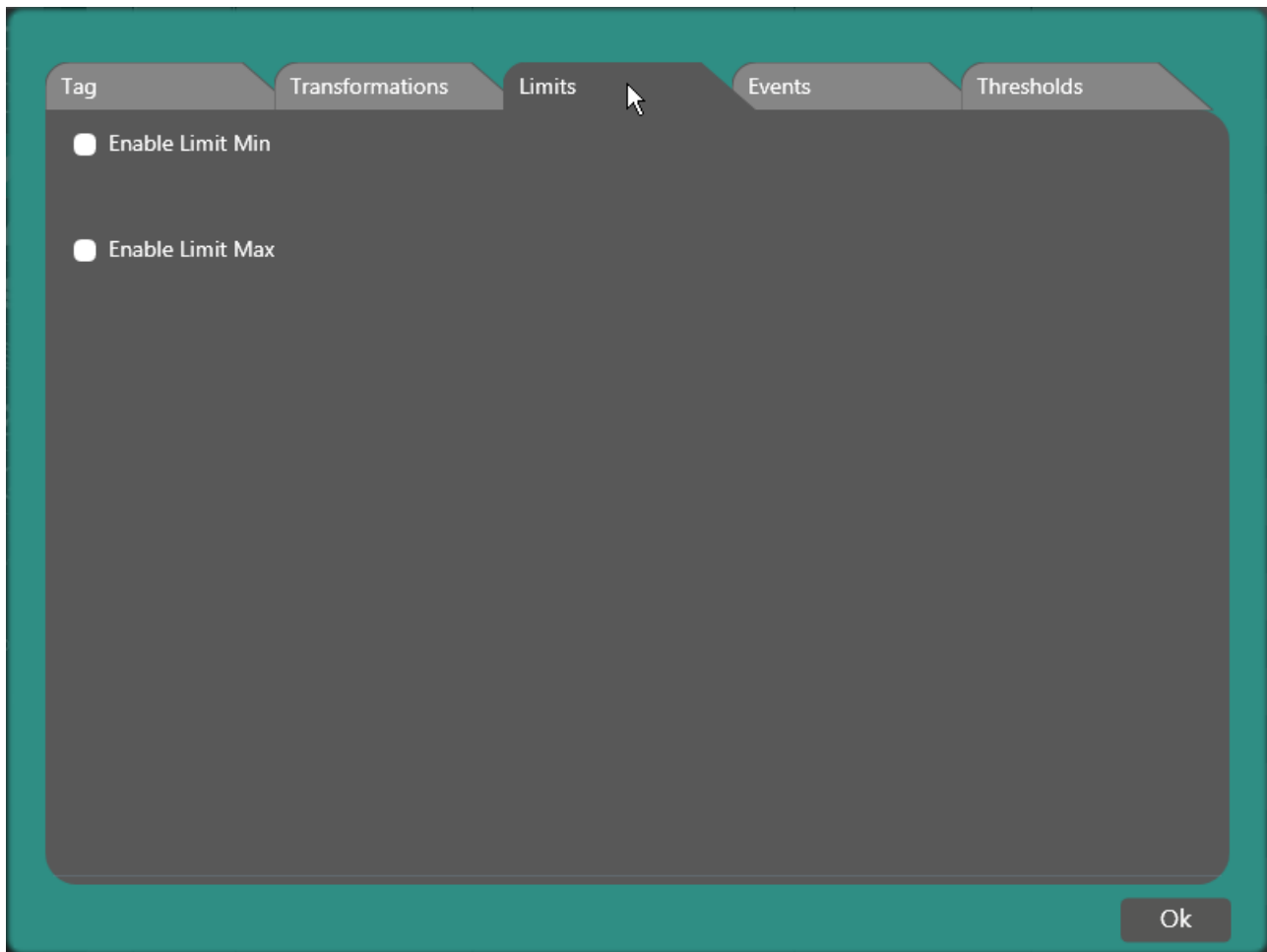
device will have a value of 40. On the other hand, by setting a value of 20 on the device, we achieve a value of 10 on the panel, as shown in the image below.



# CREW Manual

## Limits

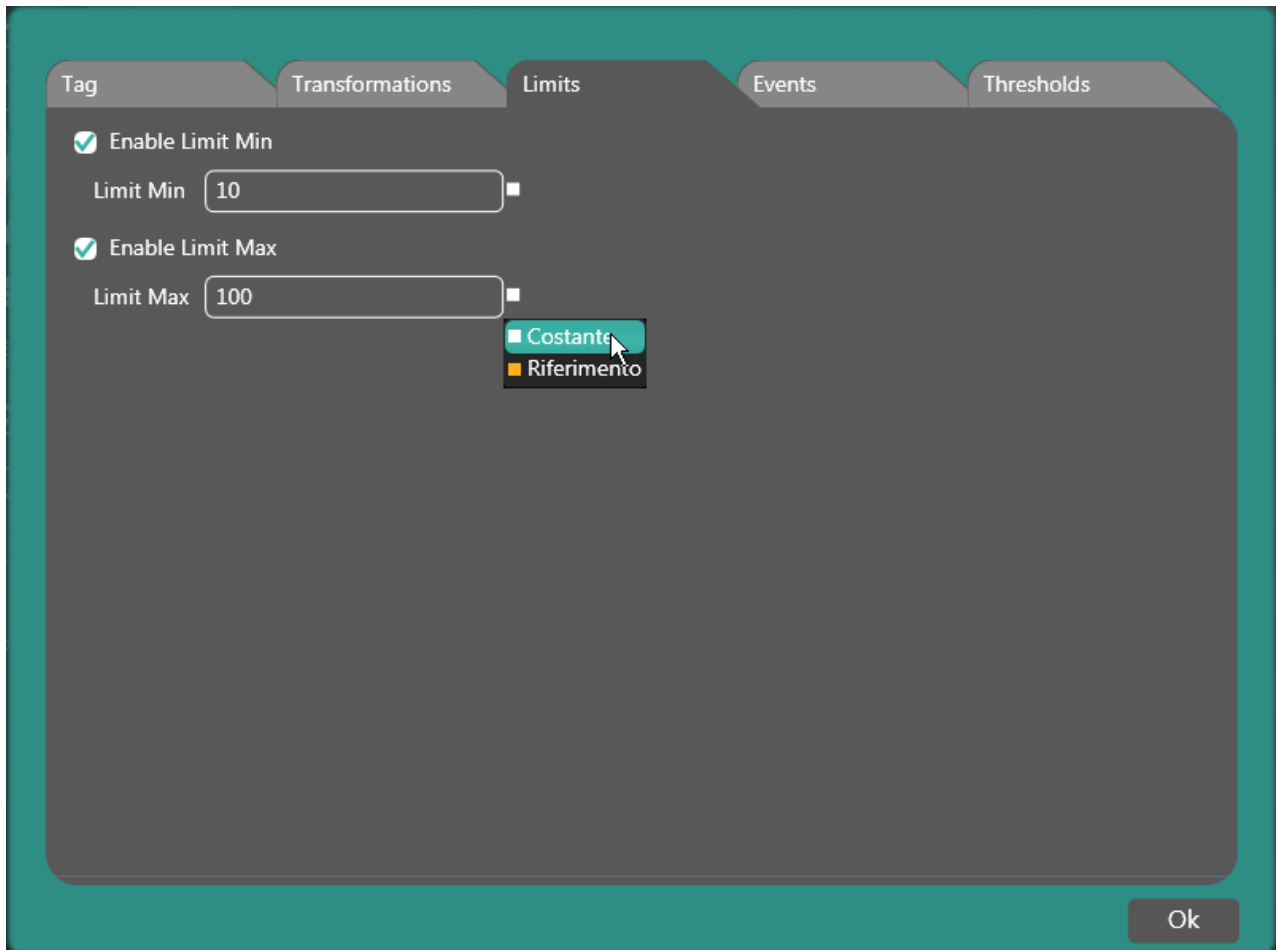
The third window of the Tags Editor is the "Limits" option.



The limits define the tag validity ranges. The limits assigned to the tag apply in the editing phase.

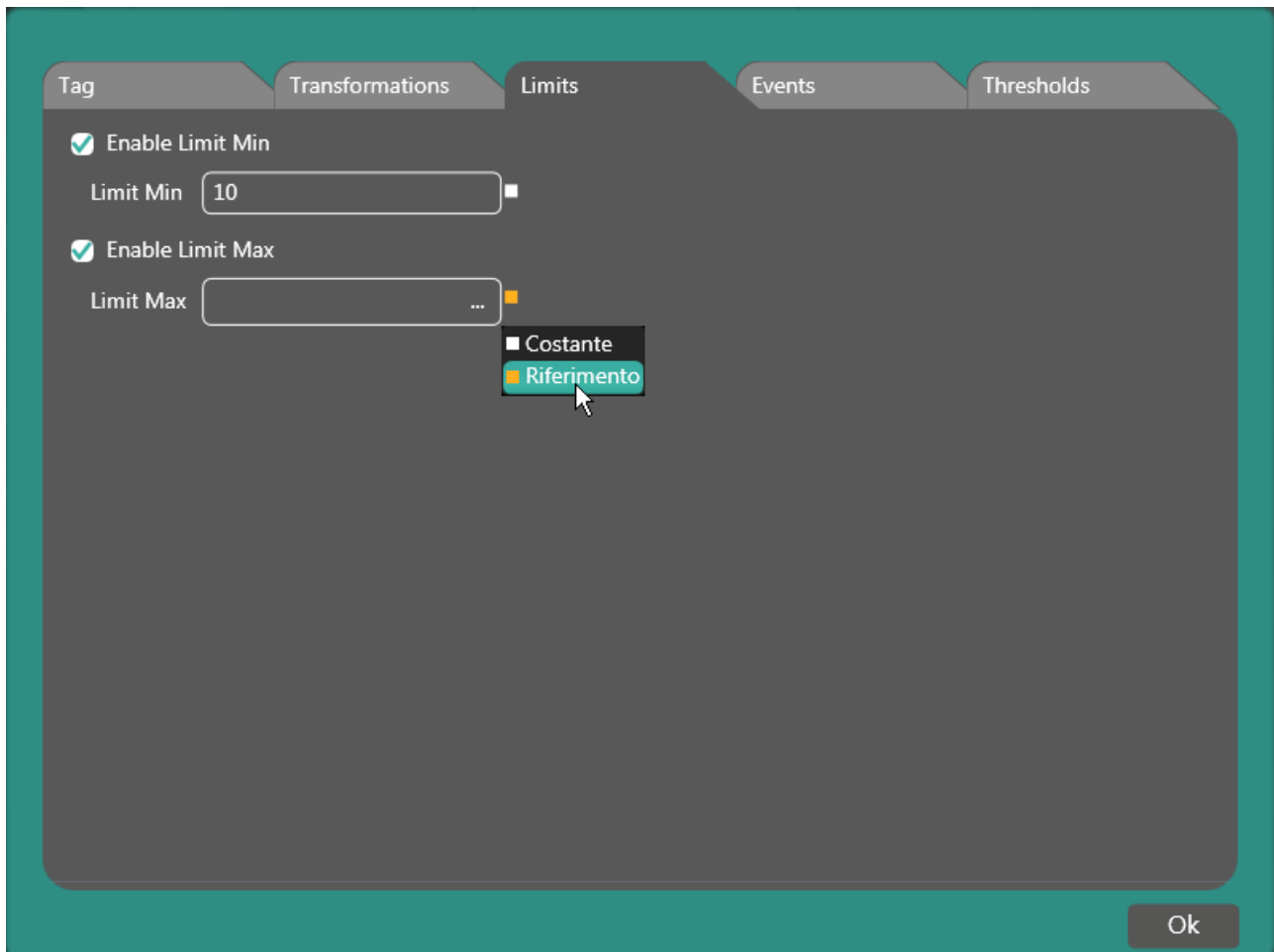
# CREW Manual

For example, if a maximum limit of 100 is set and the operator enters a higher value than that in the editing field, the field will automatically change it to 100 (namely the maximum limit). Once the relative box has been enabled, constant values can be assigned to the threshold.



# CREW Manual

Or click “Reference” to assign them dynamically by combining them with the tag ones.

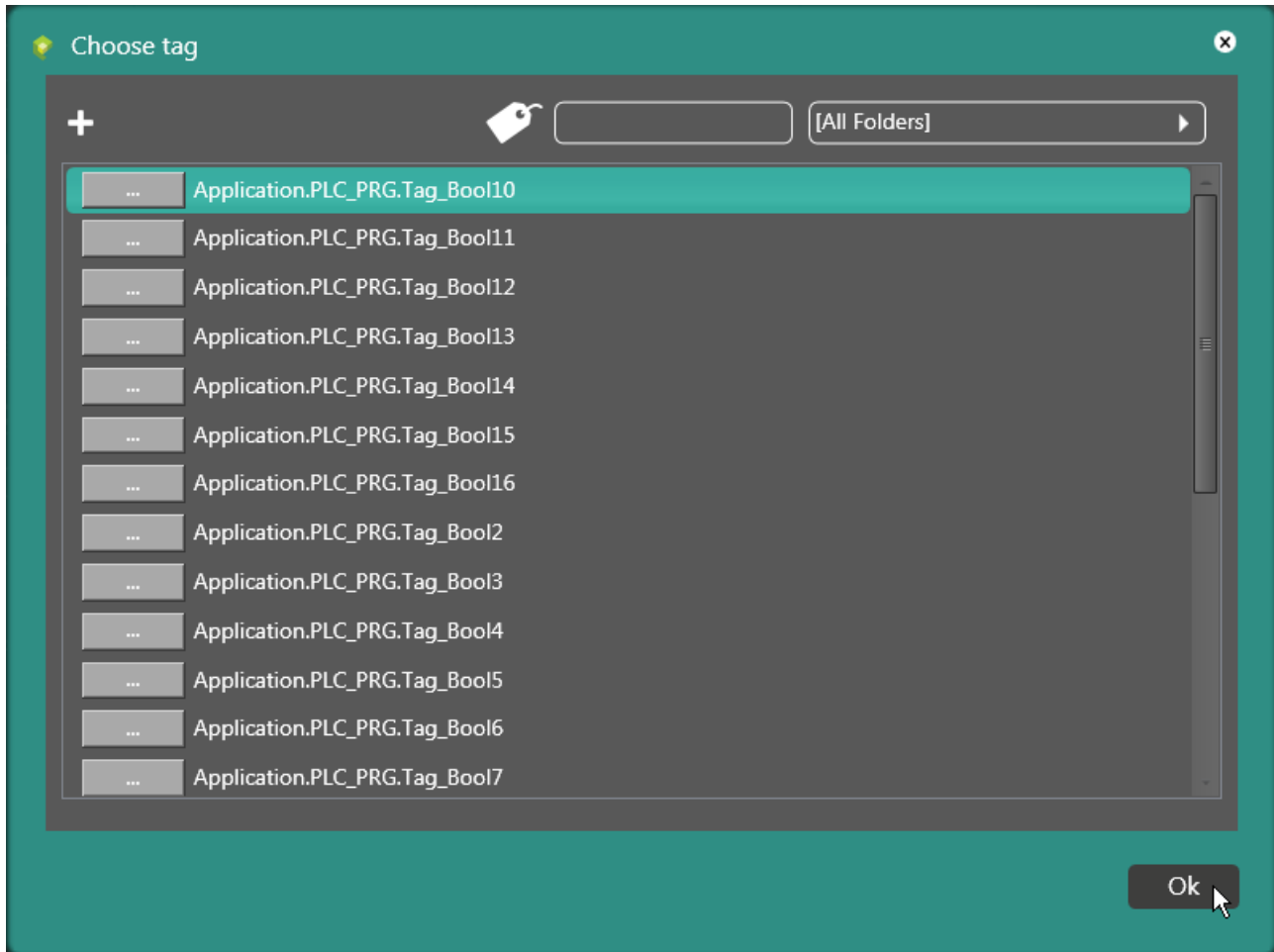


In this case it is necessary to choose the tag using the “Browse” key.



# CREW Manual

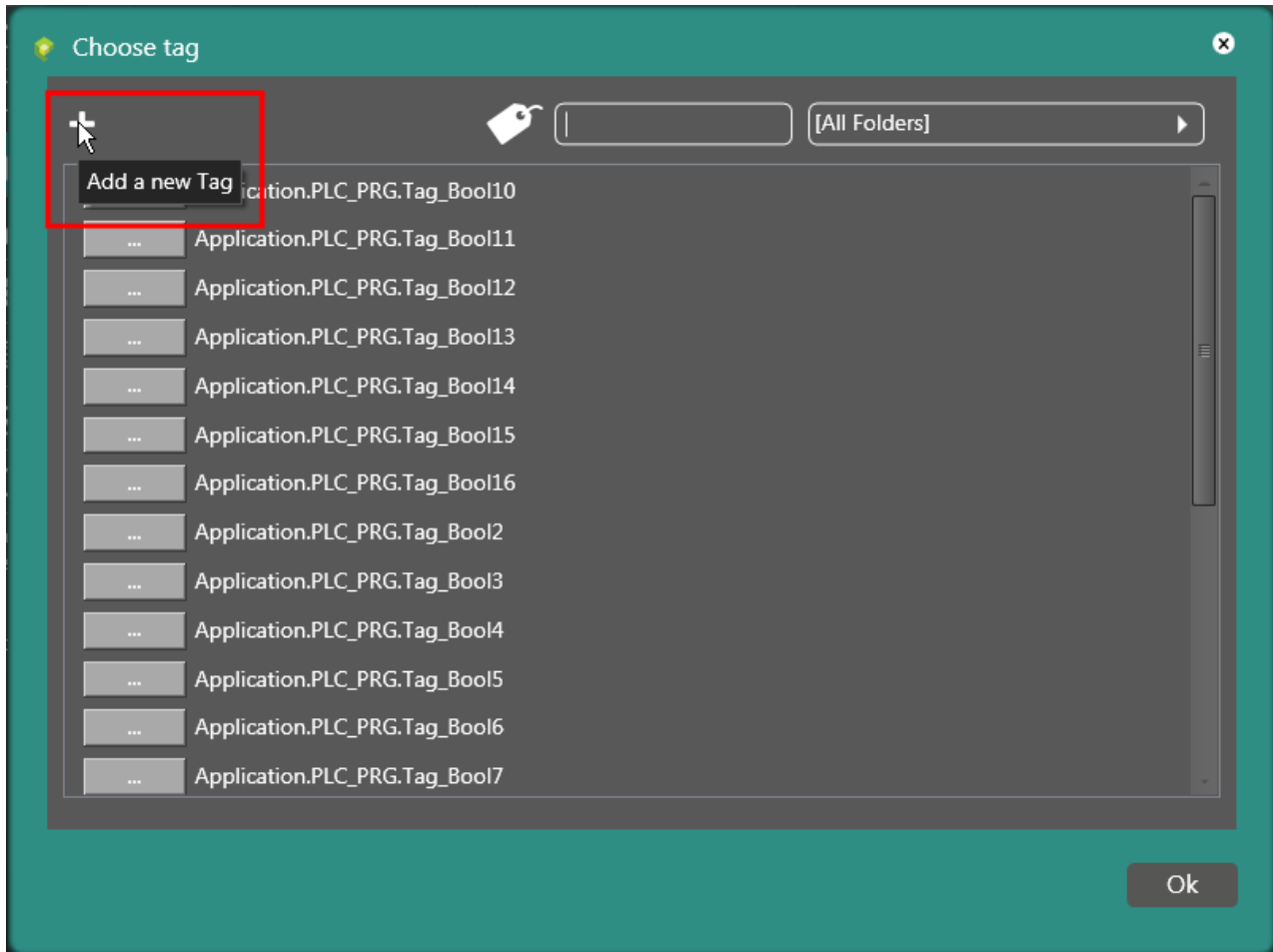
Select the required tag.





# CREW Manual

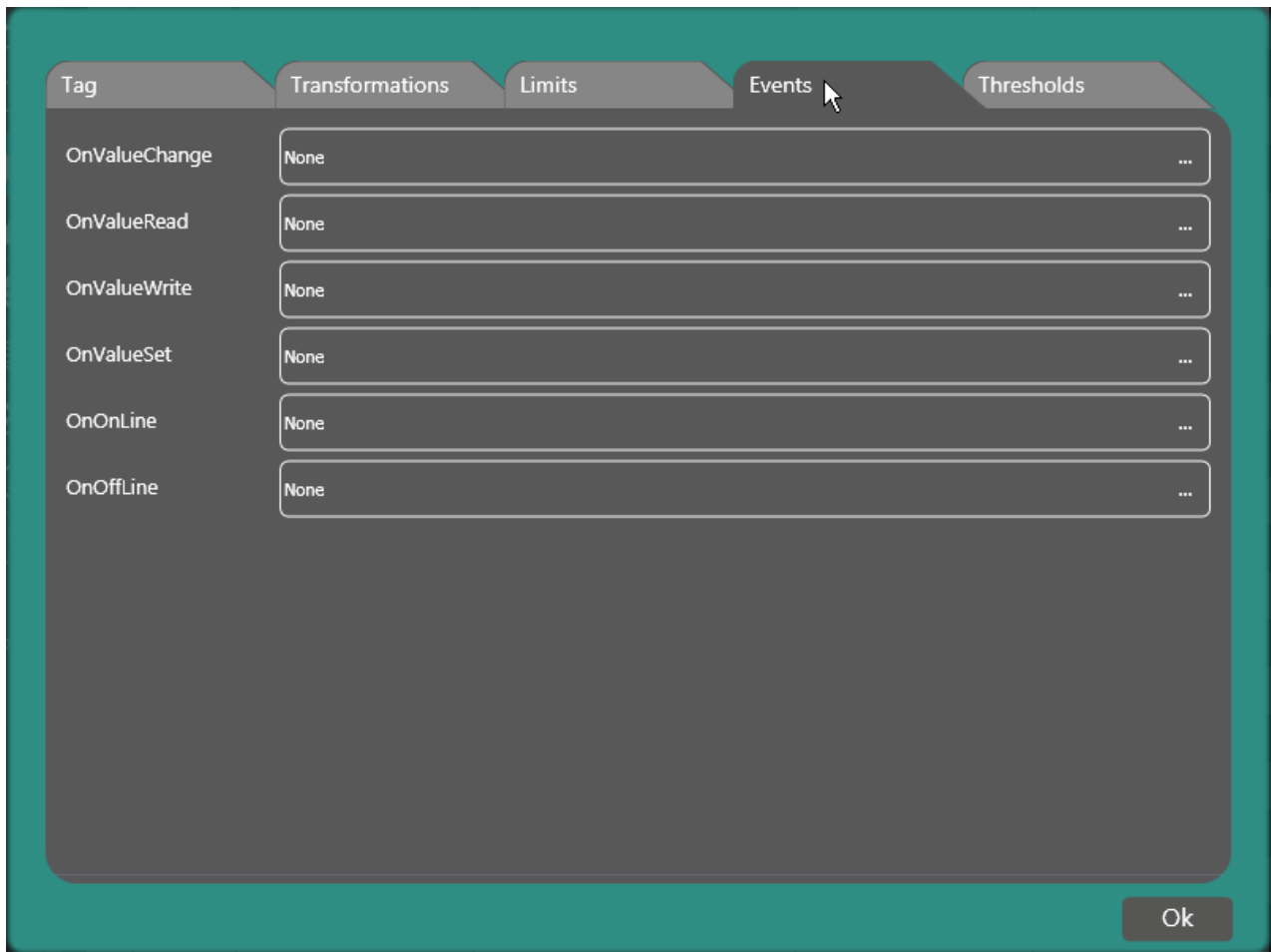
From this mask it is always possible to access tag creation-editing.



# CREW Manual

## Events - Tags -

The fourth window of the Tags Editor is the "Events" option.



An event (function or script) can be linked to each previously created Variable. Click the "Browse" key on the right.

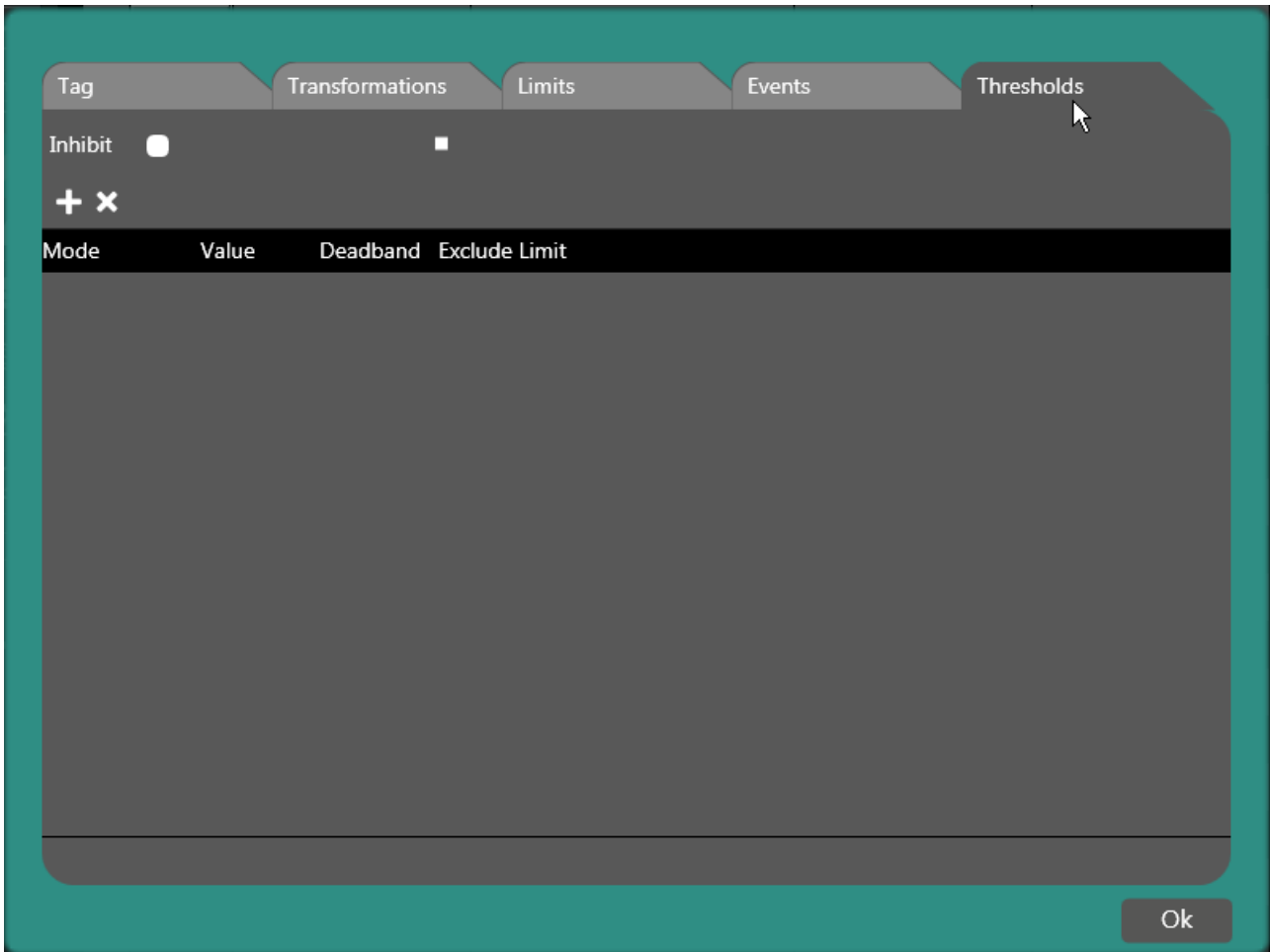


The event is enabled in the different conditions in which there is a Variable, which are described in the "Events of the variables" table.

# CREW Manual

## Thresholds

The fifth and last window of the Tags Editor is the "Thresholds" option.

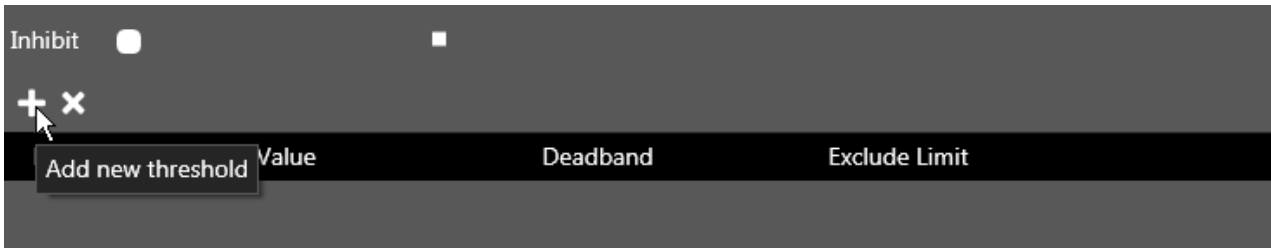


Values (called threshold values or simply thresholds) associated with the value of a Tag can be defined so as to generate events to coincide with reaching/exceeding these values. You can choose from the following options:

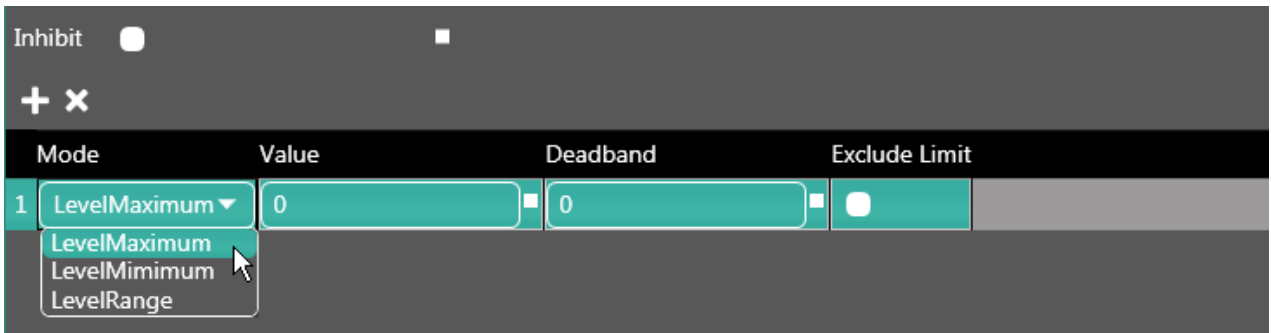
- Maximum Level (LevelMaximum)
- Minimum Level (LevelMinimum)
- Range Level (LevelRange)

# CREW Manual

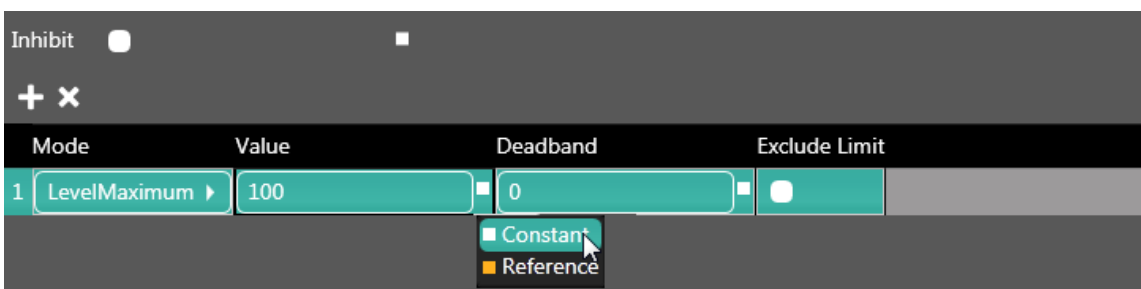
First of all, click the "+" icon to add a new threshold to "Tag001".



Select the "Maximum Level" option.

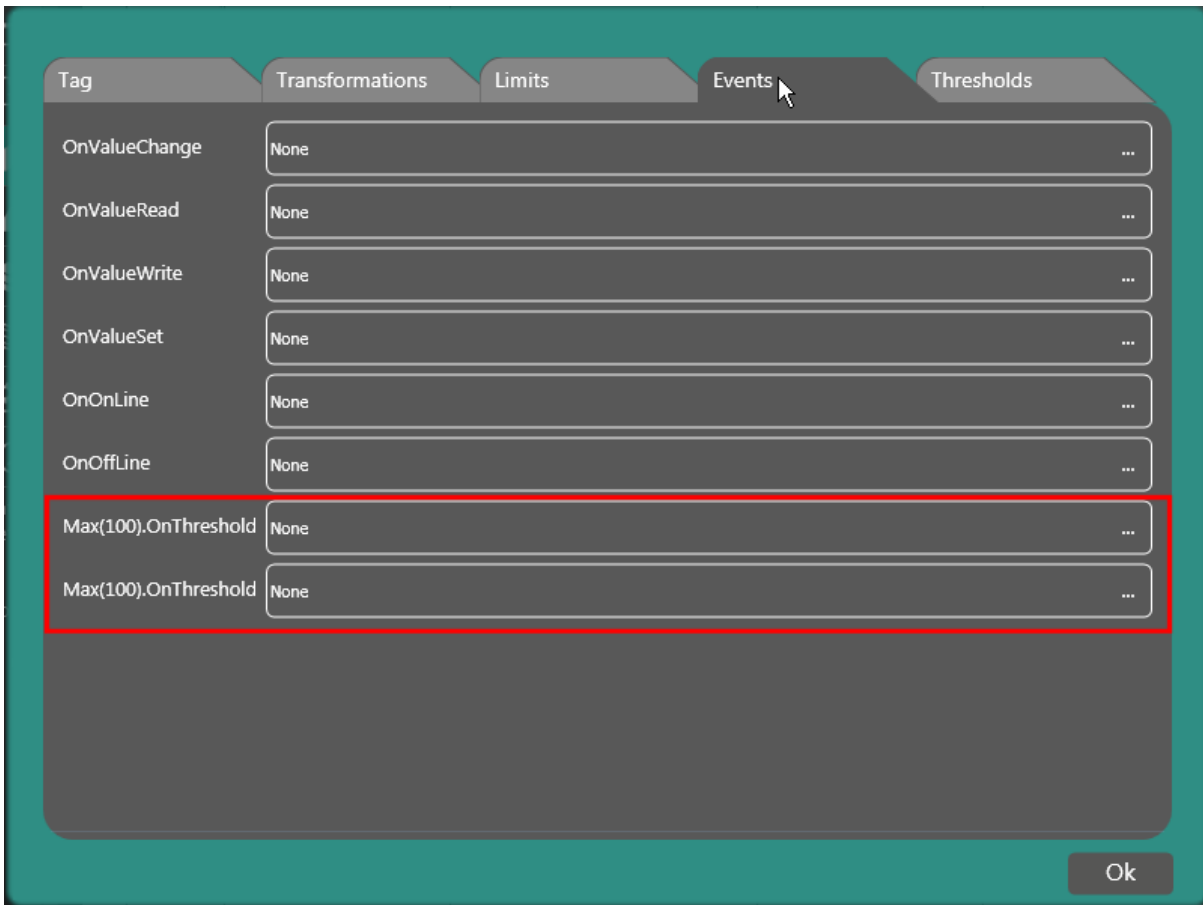


Attribute a "Constant" value of "100" to the "Maximum Level".



Assuming the "Deadband" and "Exclude Limit" fields are left unchanged, we notice that two new items have appeared in the Events that can be linked to the "Tag001" variable.

# CREW Manual

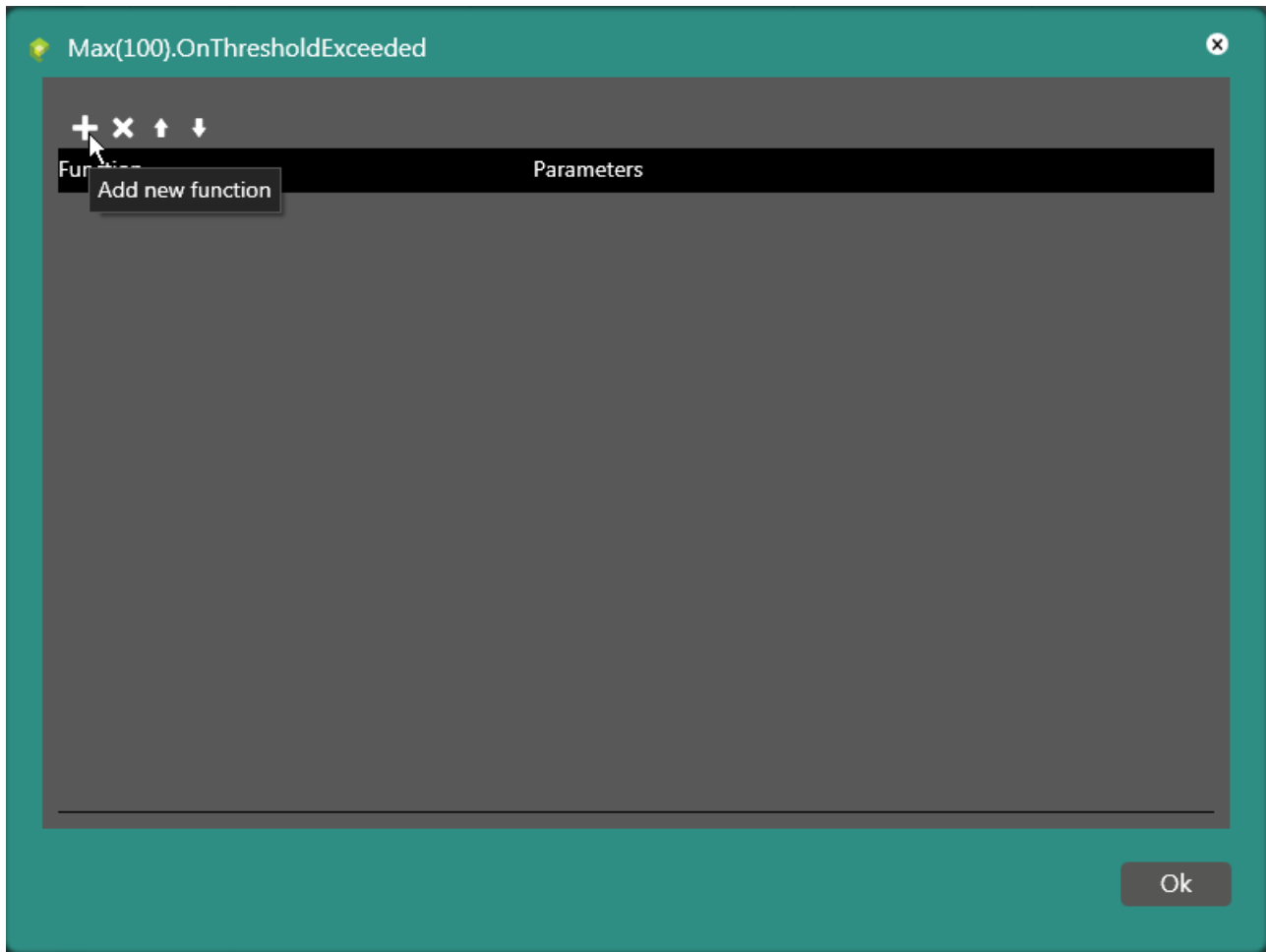


The first new option allows to associate an Event when the maximum level (100) is exceeded during the increase phase (rising values).



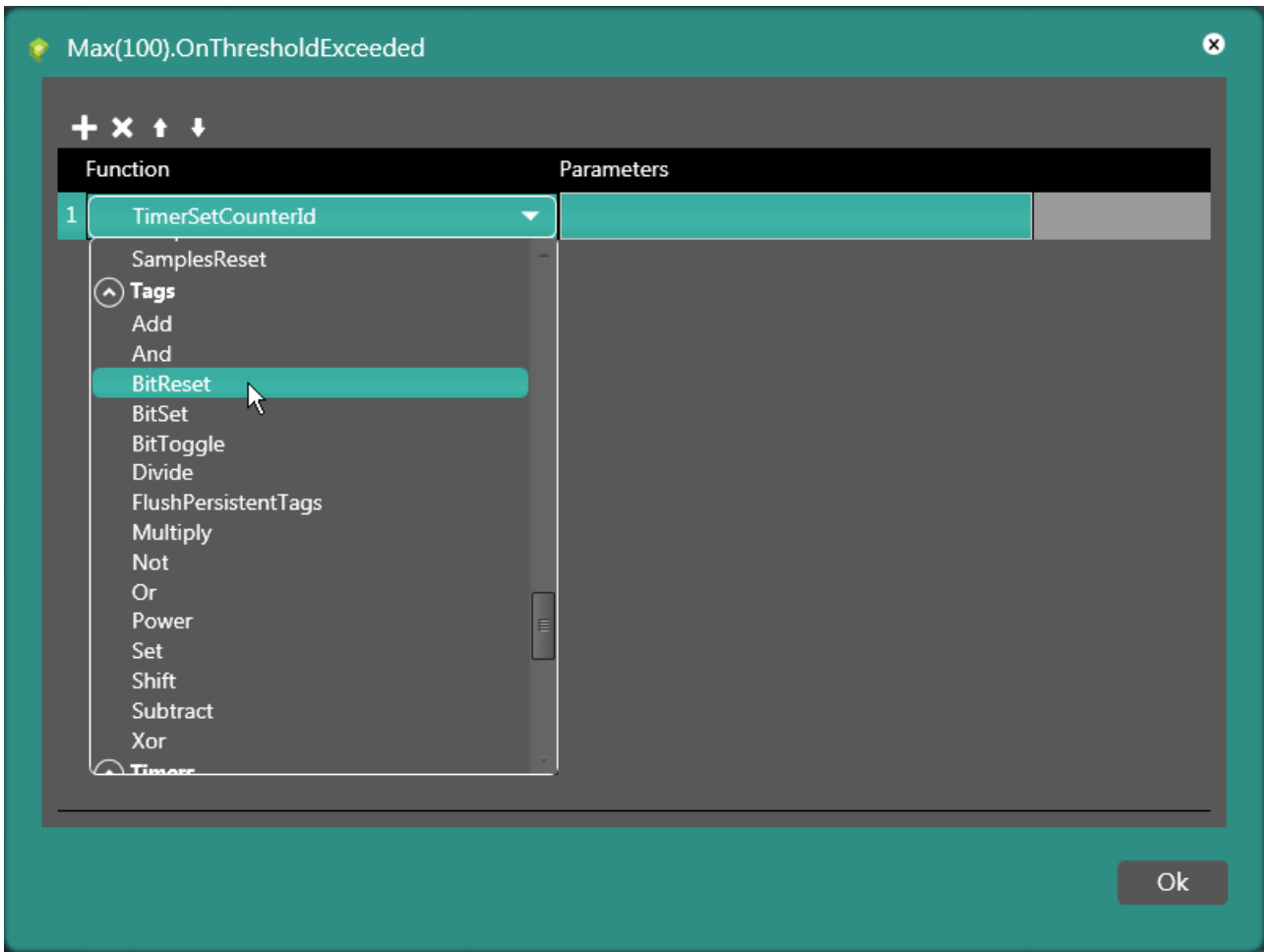
Associate one of the predefined Functions.

# CREW Manual



For example, choose the "BitReset" predefined function and confirm with "ok".

# CREW Manual

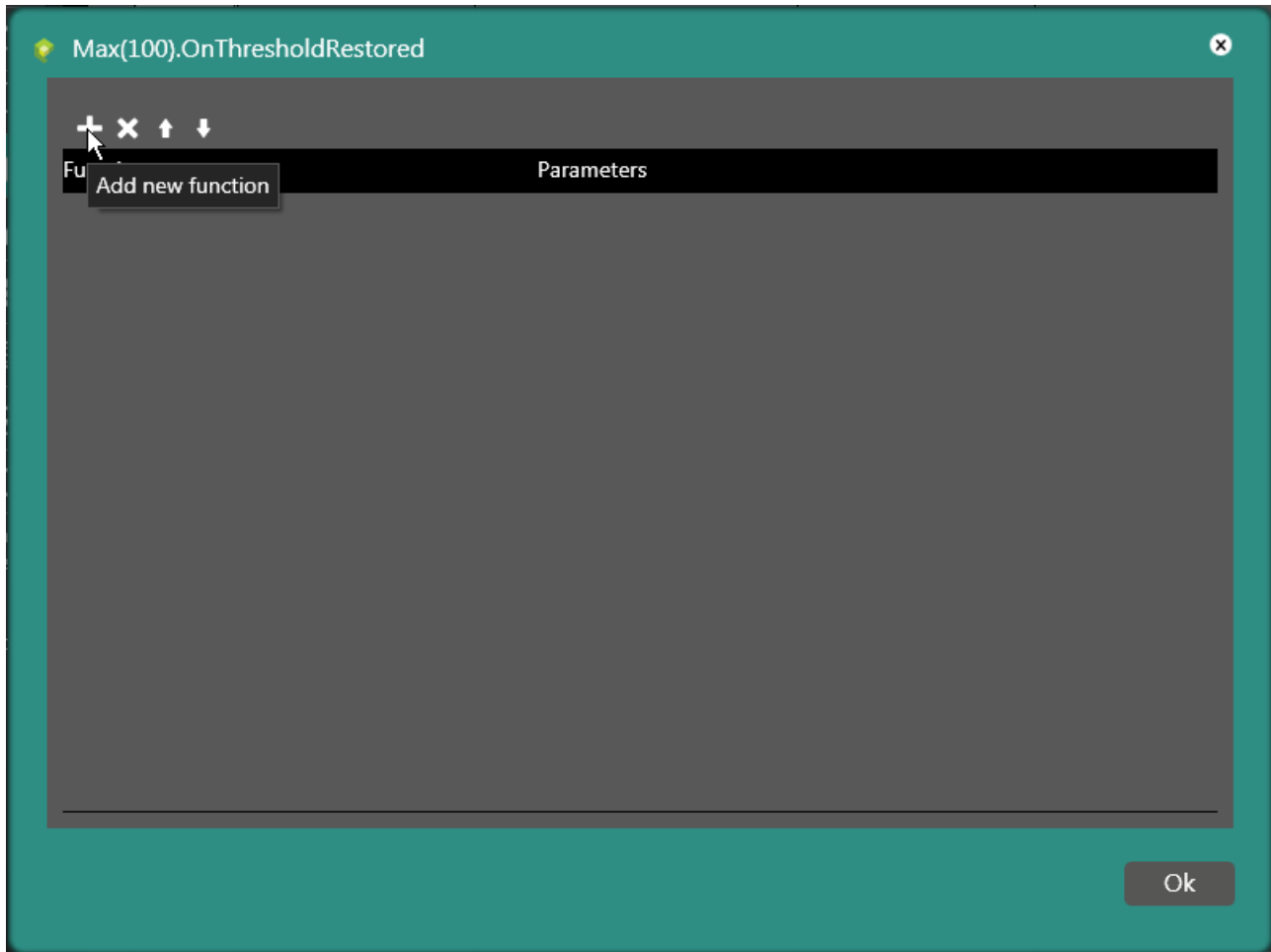


With the second new option, on the other hand, you can associate an Event when the maximum level (100) is exceeded during the decrease phase (decreasing values).



# CREW Manual

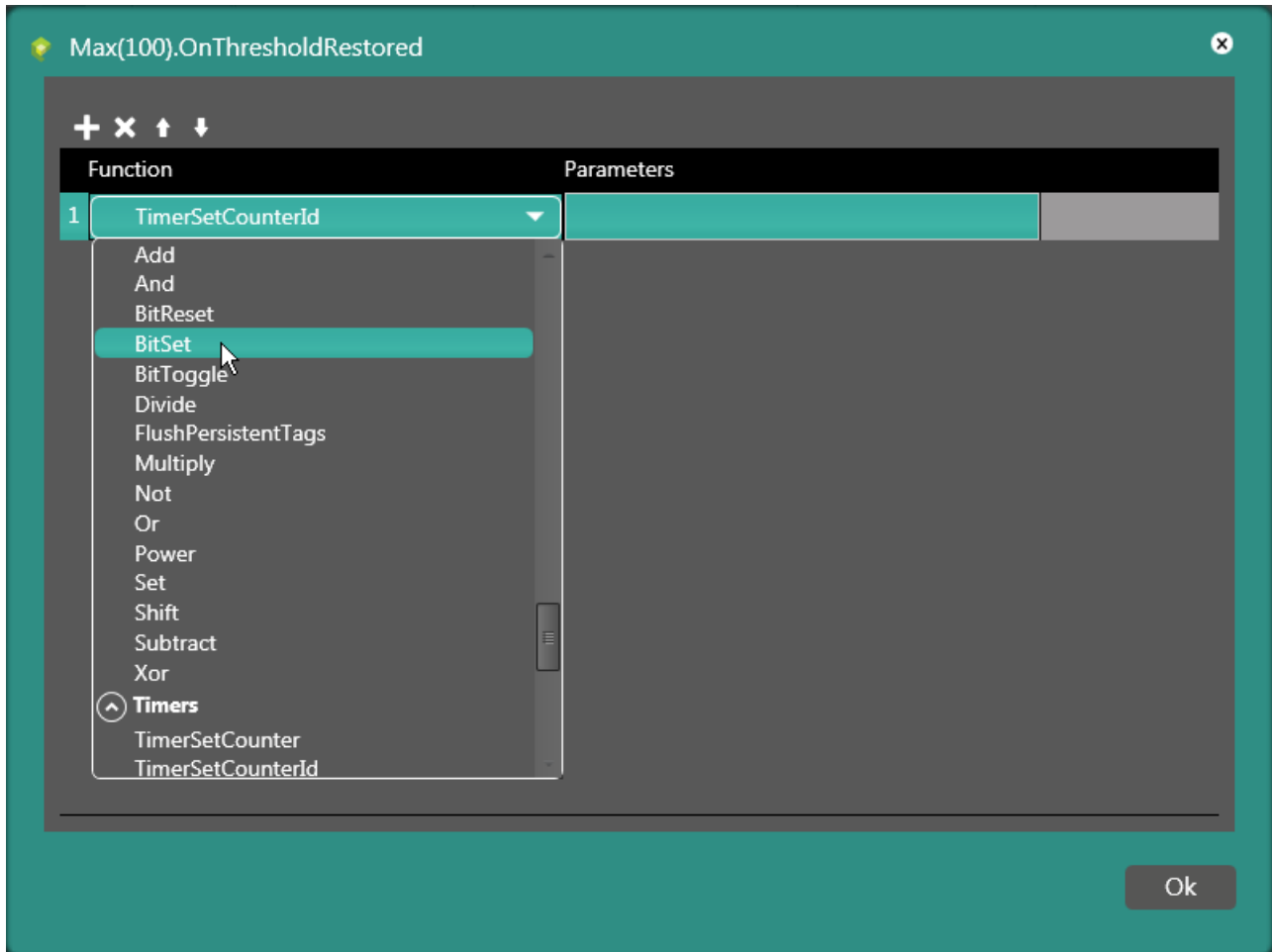
Associate one of the predefined Functions.





# CREW Manual

For example, choose the "BitSet" predefined function and confirm with "ok".

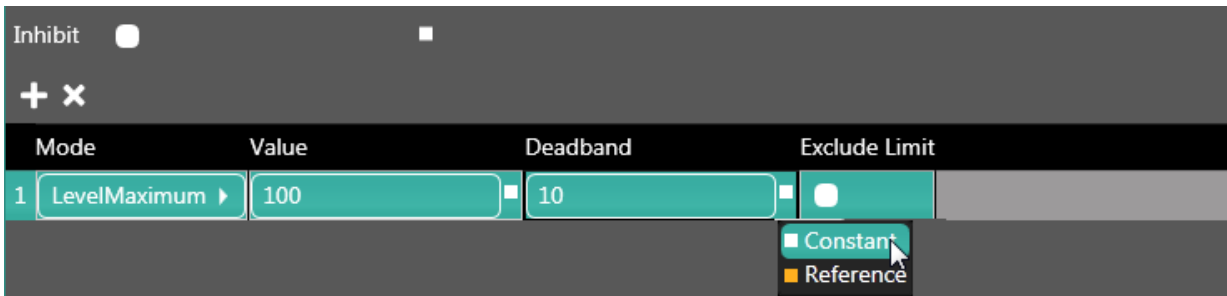


With the sequence described above, when the value of Tag001 exceeds "100" with increasing values (98, 99, 100, 101), the event associated with exceeding the threshold is triggered (OnThresholdExceeded), which the example identifies as the "BitReset" function. On the other hand, when the value exceeds "100" with decreasing values (103, 102, 101, 100, 99), the event associated with dropping below the threshold is triggered (OnThresholdRestored), which the example identifies as the "BitSet" function.

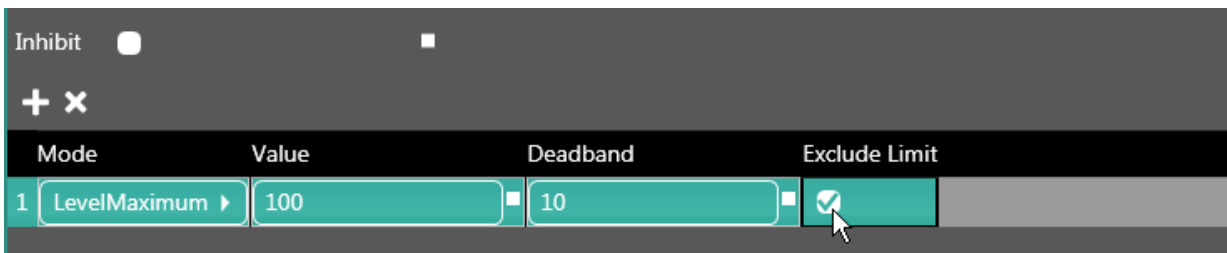
It is also possible to assign a "Deadband" value, during which time no event occurs.

# CREW Manual

For example, by assigning a “Deadband” value of “10” (“Constant”), for increasing values greater than 100 (if decimal values are not used), the associated event is not triggered until the value of “110” is reached.



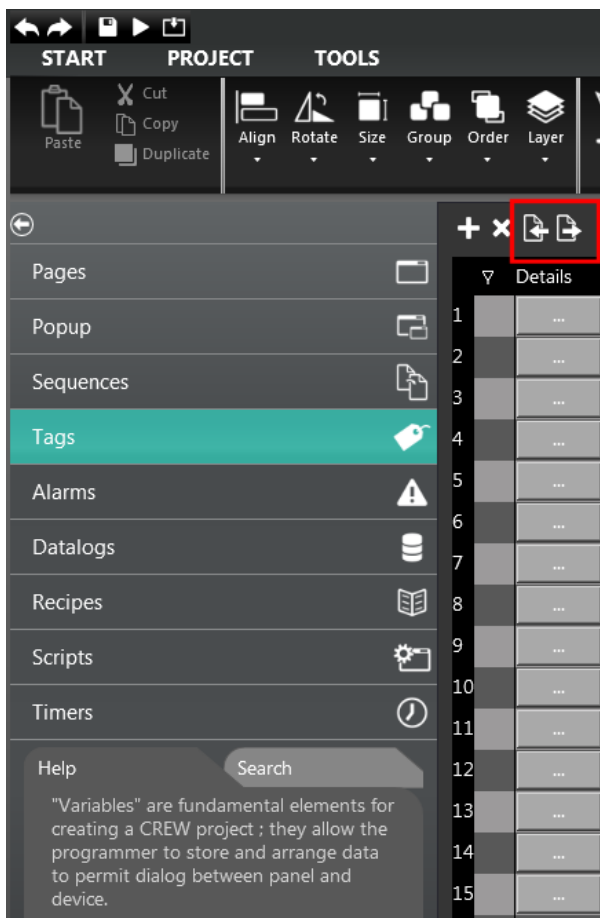
Lastly, the “Exclude Limit” option allows you to exclude the event from being triggered even at limit values (100 and 110 in the example).



# CREW Manual

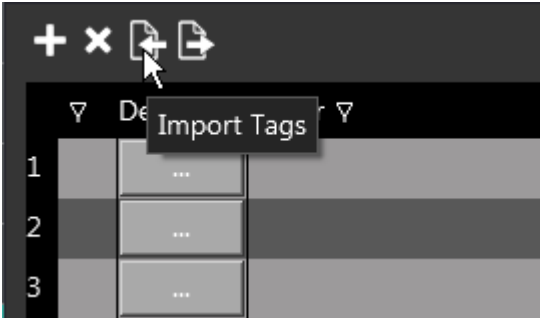
## Export - Import Tags

Click the relative icons (see image) to export and import previously created variables from/to the project.

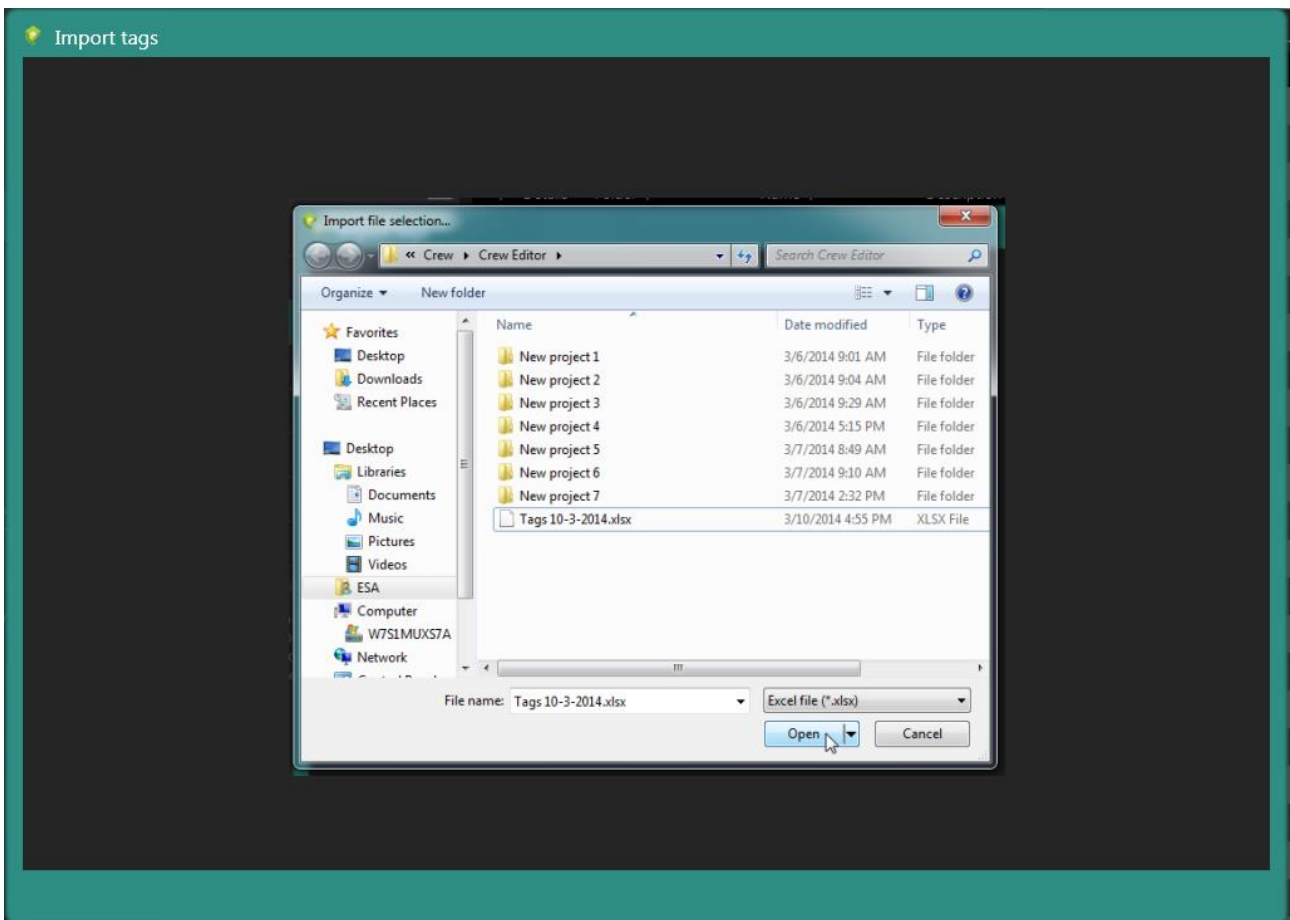


# CREW Manual

## Import Tags

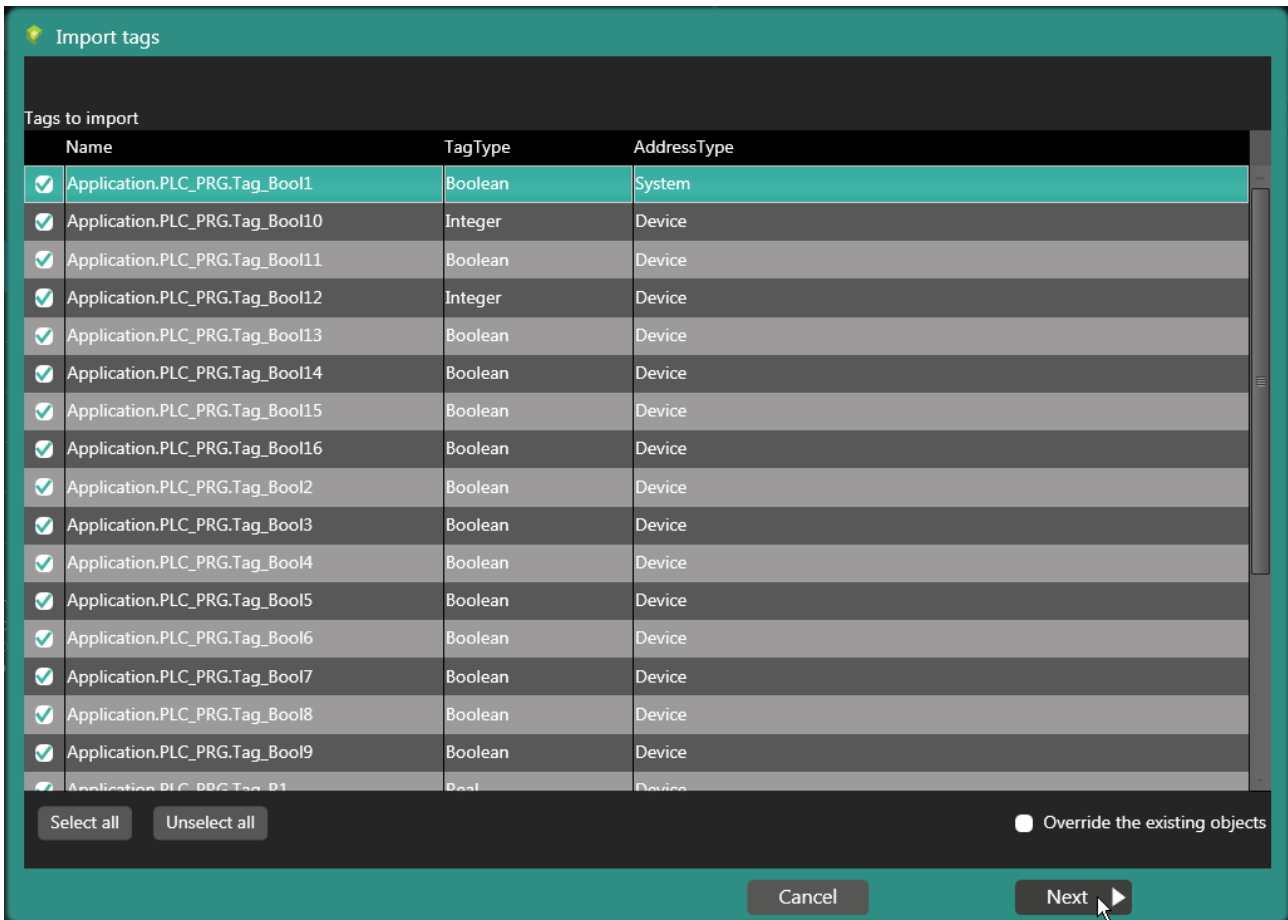


Click "Import Tags" and select the file with the list of tags to be imported into the project.



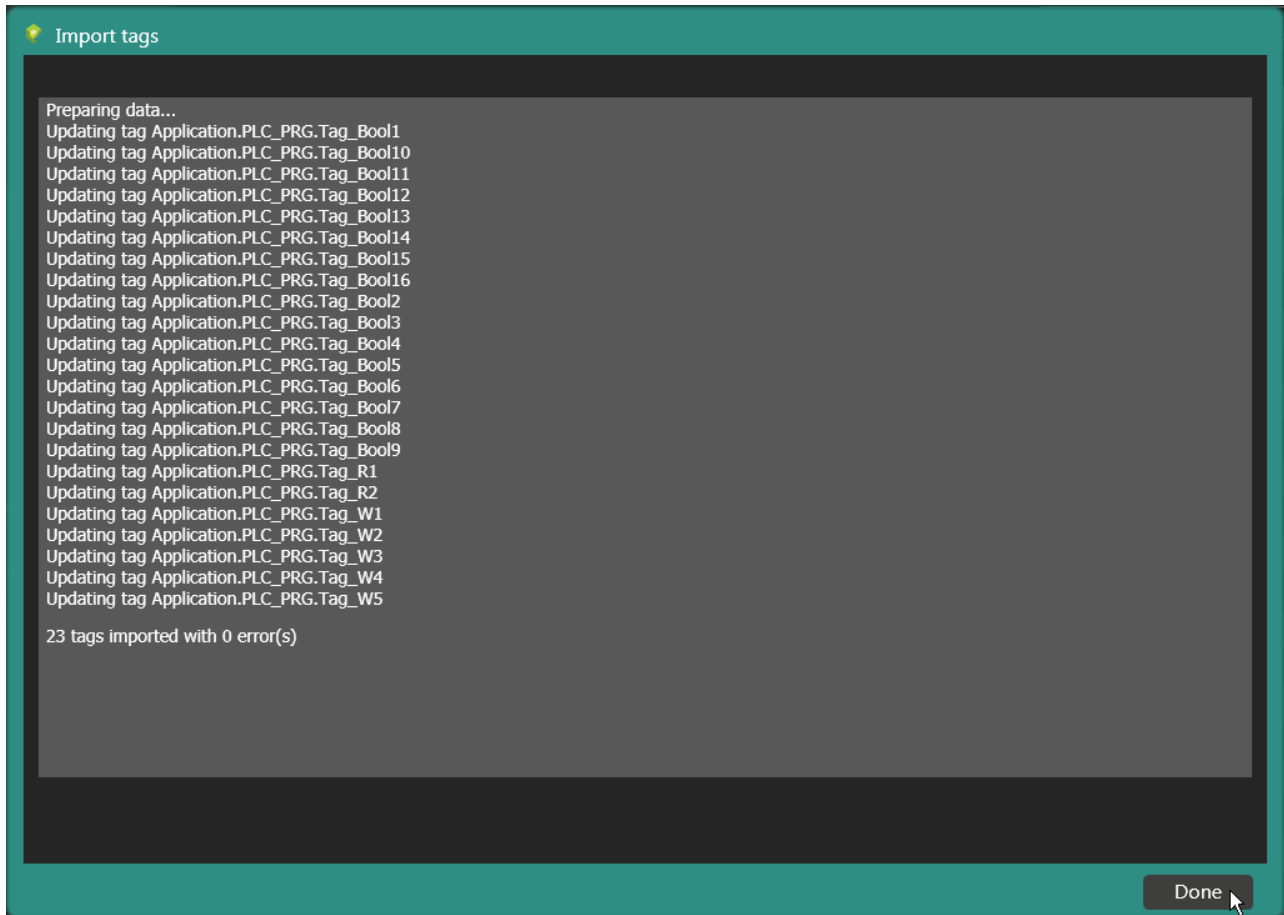
# CREW Manual

Once the file has been selected, decide whether to overwrite or not the objects existing in the project. Click “Next” to import the tags. It is possible to decide whether to import all or only those selected with the flag.



# CREW Manual

When you have clicked “Next”, the import procedure begins. At the end click “Done”.

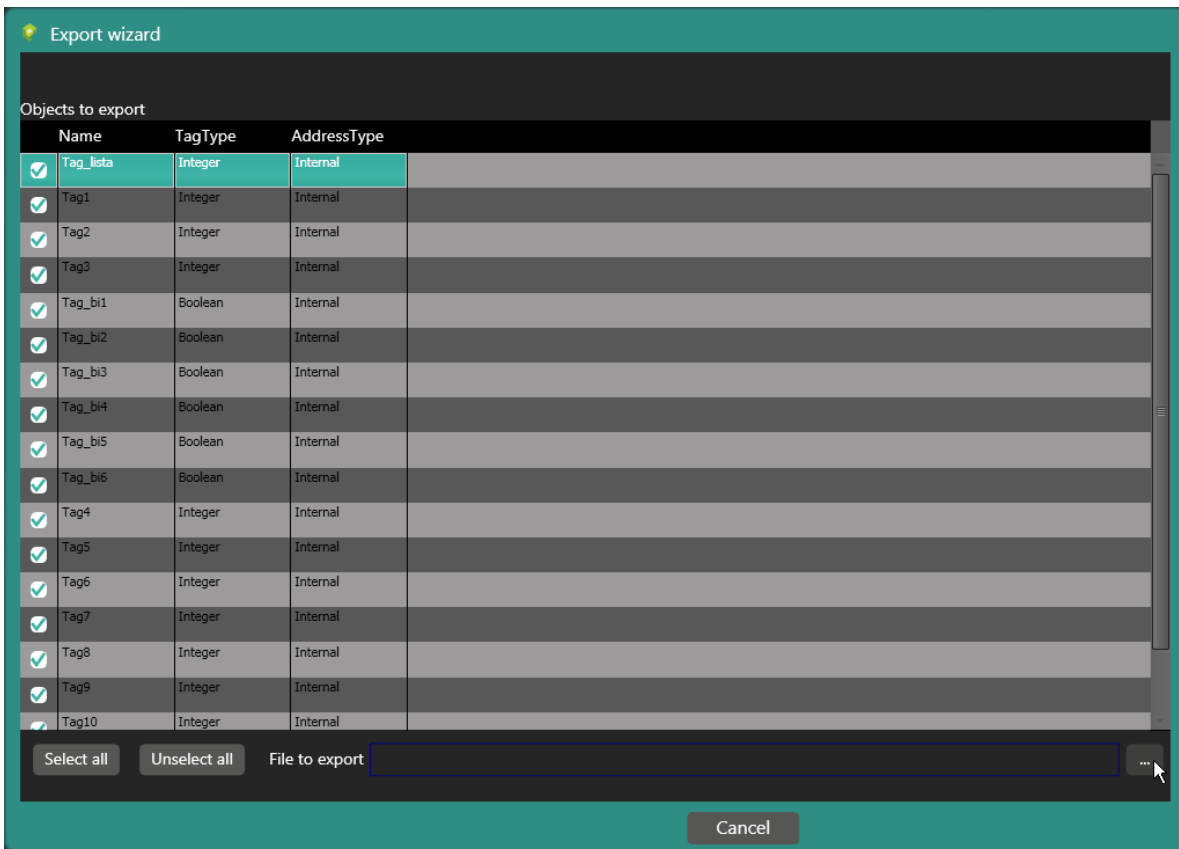


# CREW Manual

## Export Tags

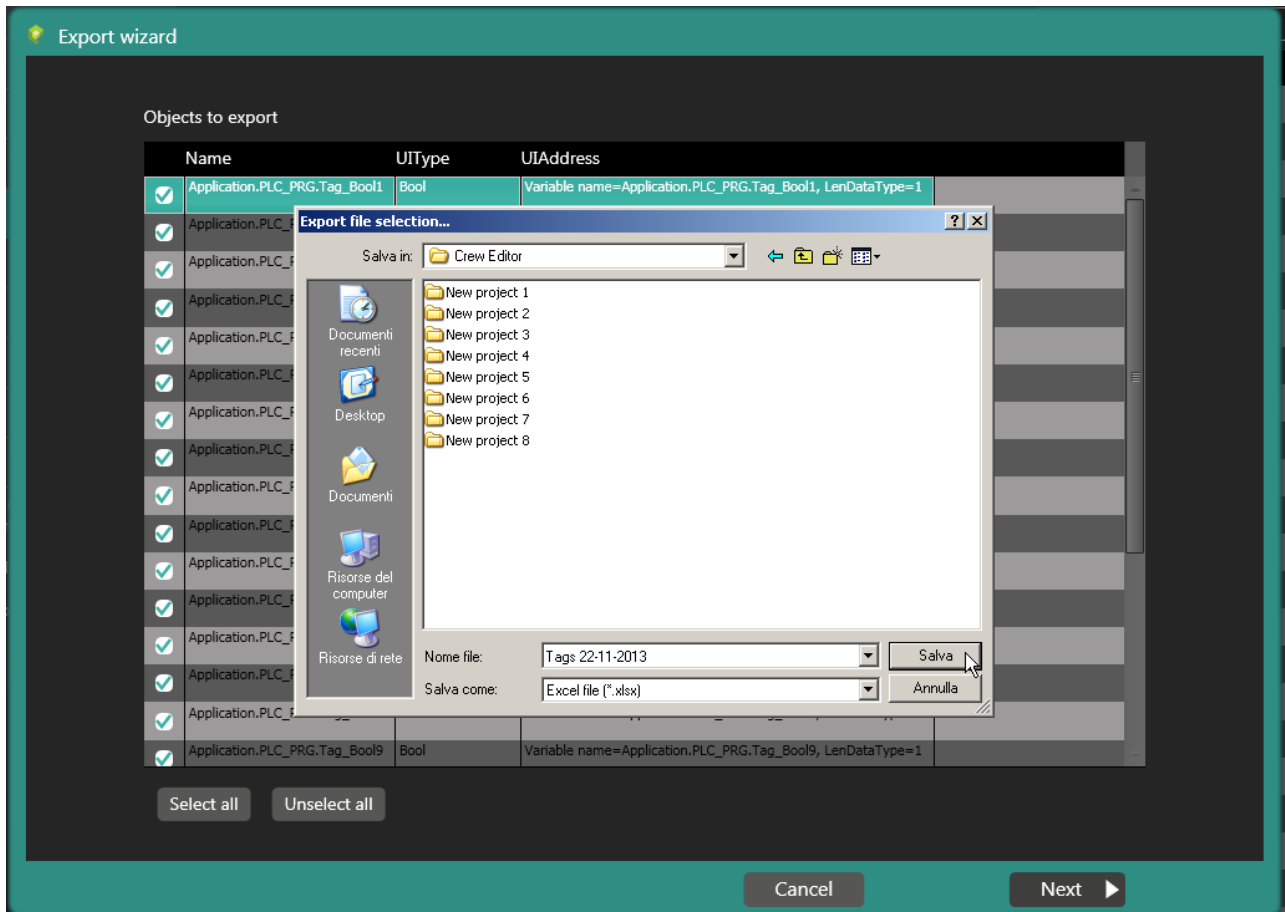


Click the “Export Tags” key to export the list of tags contained in the project to a file. It is possible to decide whether to export all of the tags or only those selected with the flag.



Click “Browse” and choose the target folder.

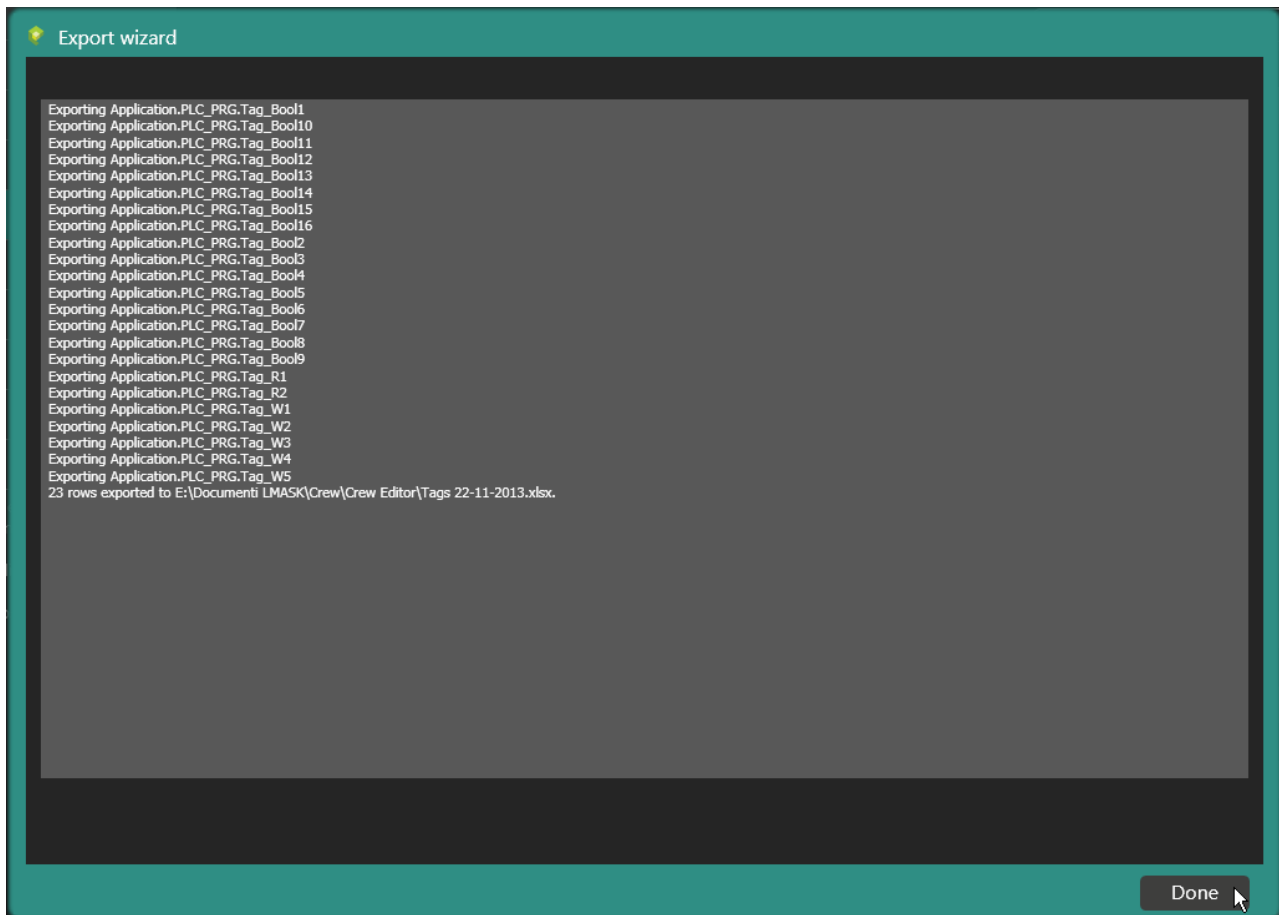
# CREW Manual





# CREW Manual

Click “Save” and “Next” to start the exporting procedure. At the end click “Done”.



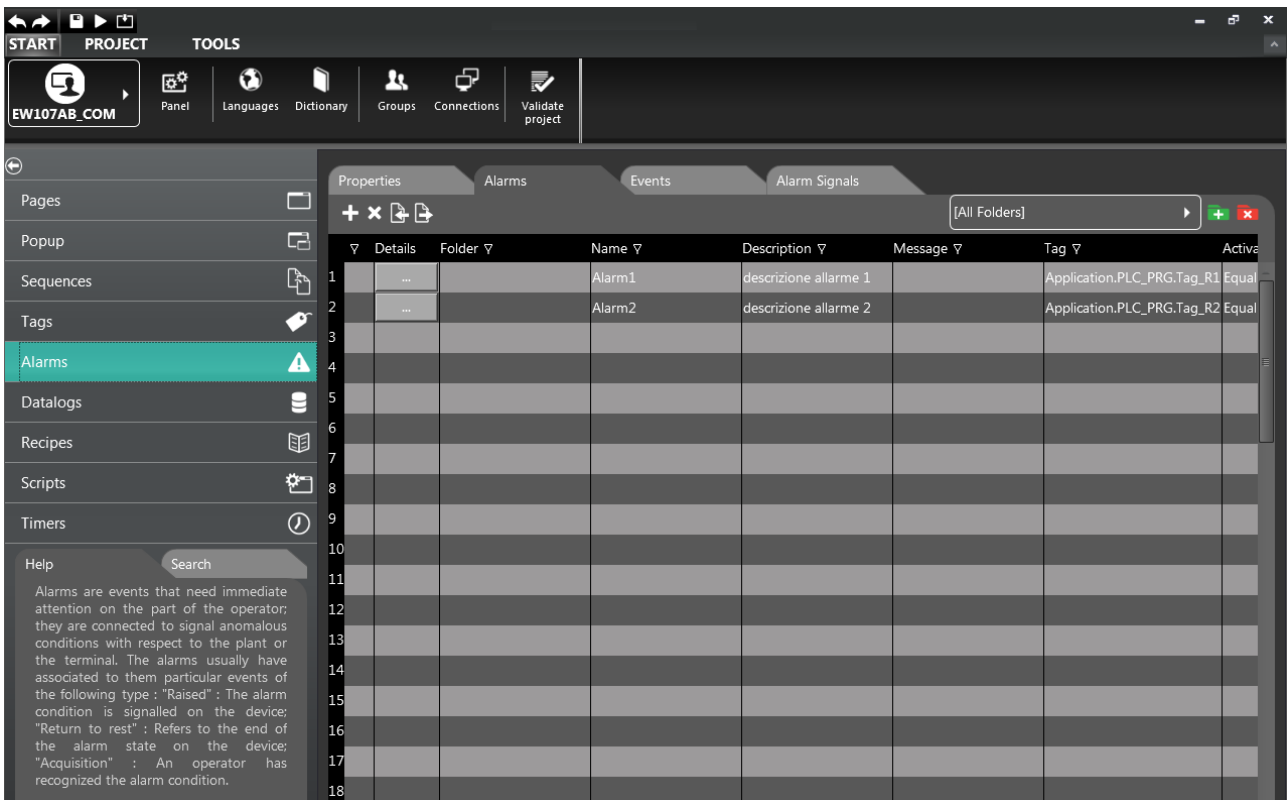
# CREW Manual

## Alarms

The “Alarms” are events that require the operator’s immediate attention. They occur when anomalous situations arise on the system or terminal. Generally the following events are associated to the alarms:

- “Access”: the alarm condition is activated on the device.
- “Reset”: linked to the end of the alarm condition on the device.
- “Acquisition”: an operator has taken over management of the alarm.

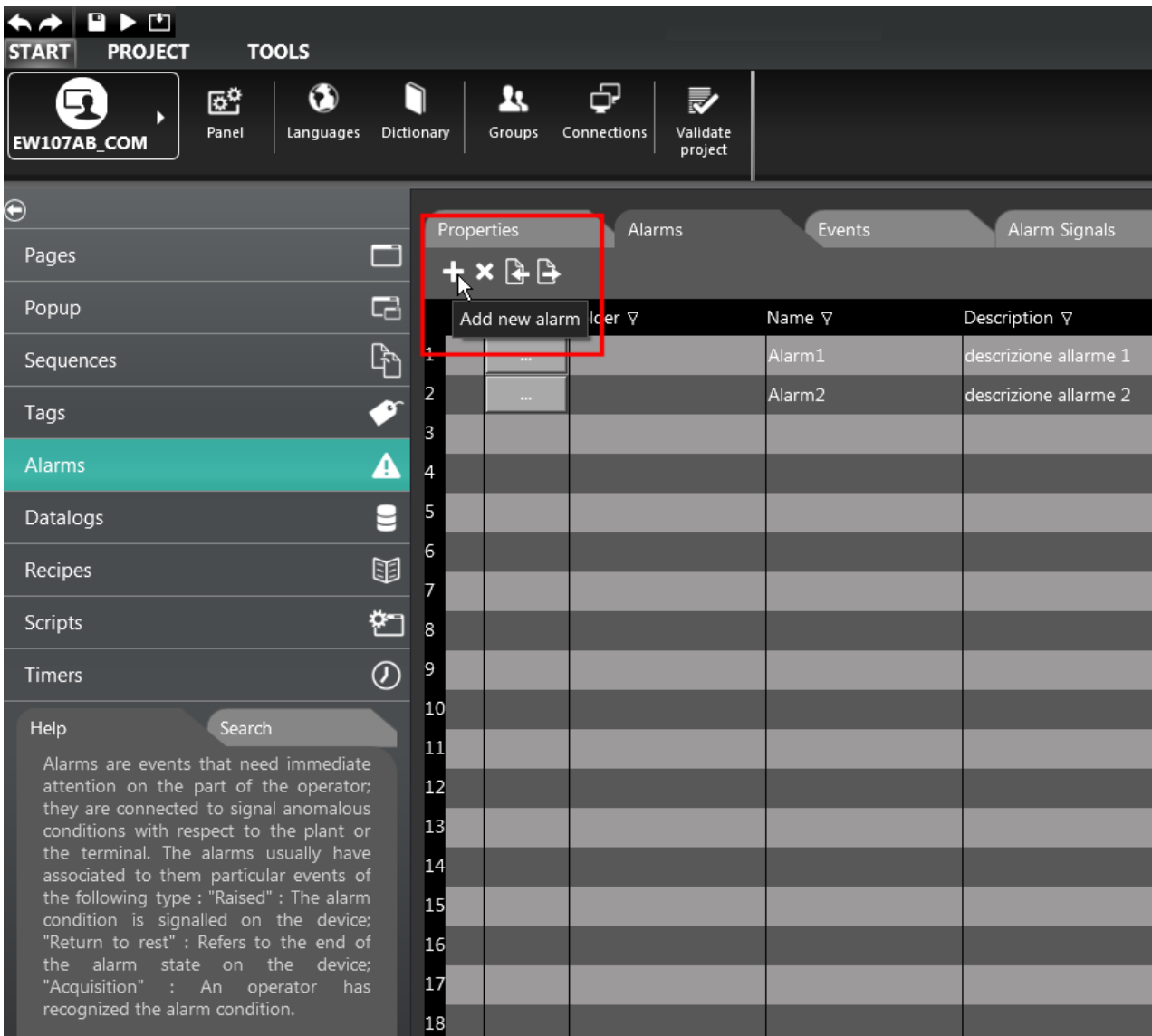
Click the "Alarms" menu, to make the list of alarms included in the project appear in the work area.



# CREW Manual

From this list it is possible to do the following operations.

Enter new alarms.



The screenshot shows the software interface with the following elements:

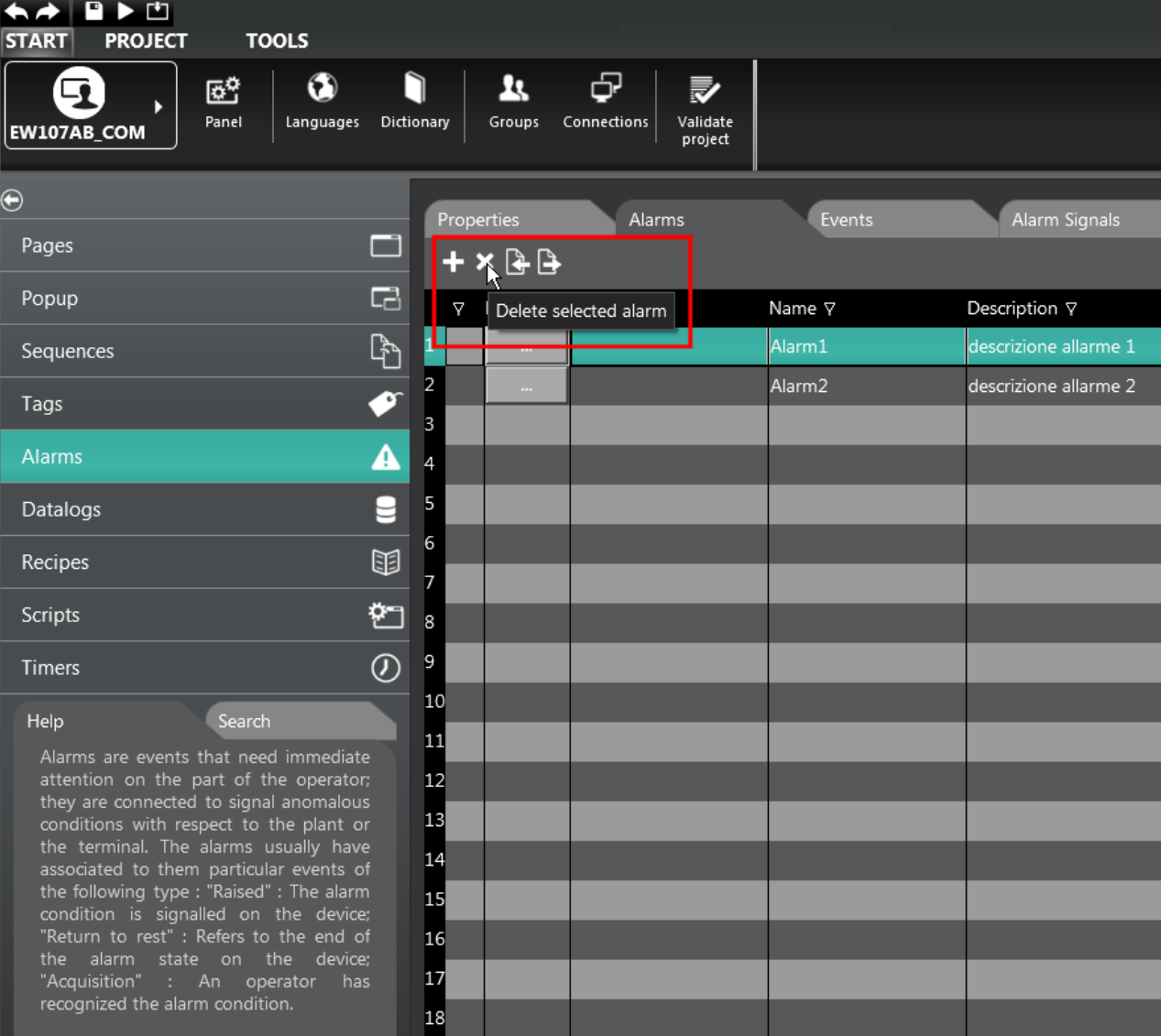
- Top Bar:** Navigation icons (back, forward, refresh, save) and tabs for **START**, **PROJECT**, and **TOOLS**.
- Tools Panel:** Includes icons for Panel, Languages, Dictionary, Groups, Connections, and Validate project.
- Left Sidebar:** A list of menu items: Pages, Popup, Sequences, Tags, **Alarms** (highlighted), Datalogs, Recipes, Scripts, and Timers. A 'Help' section is also visible at the bottom.
- Main Area:** A table for managing alarms. The 'Properties' tab is active, showing a table with columns for 'Index', 'Name', and 'Description'. Two existing alarms are listed: 'Alarm1' and 'Alarm2'. A red box highlights the '+ x' icon in the 'Properties' header, which is used to add new alarms.

Index	Name	Description
1	Alarm1	descrizione allarme 1
2	Alarm2	descrizione allarme 2
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		

**Help:** Alarms are events that need immediate attention on the part of the operator; they are connected to signal anomalous conditions with respect to the plant or the terminal. The alarms usually have associated to them particular events of the following type : "Raised" : The alarm condition is signalled on the device; "Return to rest" : Refers to the end of the alarm state on the device; "Acquisition" : An operator has recognized the alarm condition.

# CREW Manual

Delete selected alarm.



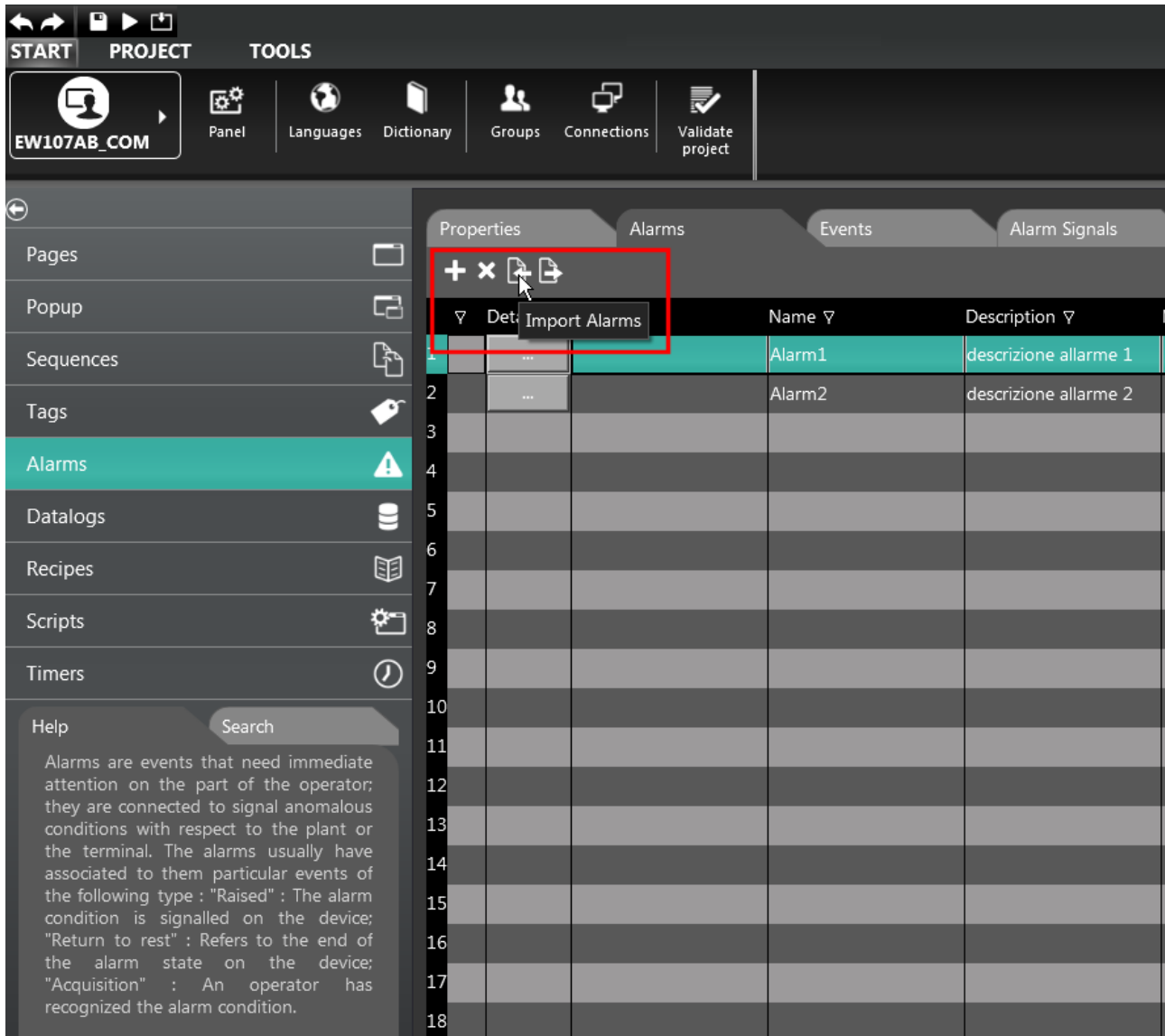
The screenshot shows the software interface with the 'Alarms' tab selected. The toolbar contains icons for adding (+), deleting (x), and moving (arrows) alarms. A red box highlights the delete icon, with a tooltip that reads 'Delete selected alarm'. Below the toolbar is a table with the following data:

		Name ▾	Description ▾
1	...	Alarm1	descrizione allarme 1
2	...	Alarm2	descrizione allarme 2
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			

On the left side, there is a sidebar with navigation options: Pages, Popup, Sequences, Tags, Alarms (selected), Datalogs, Recipes, Scripts, and Timers. Below the sidebar is a 'Help' section with a search bar and text explaining that alarms are events requiring immediate attention, connected to signal anomalous conditions. It lists types: 'Raised' (alarm condition signalled), 'Return to rest' (end of alarm state), and 'Acquisition' (operator recognized condition).

# CREW Manual

Import a set of previously created alarms.



The screenshot shows the software interface with the 'Alarms' section selected in the left sidebar. The main area displays a table with columns for 'Name' and 'Description'. A red box highlights the 'Import Alarms' button in the top toolbar of the table.

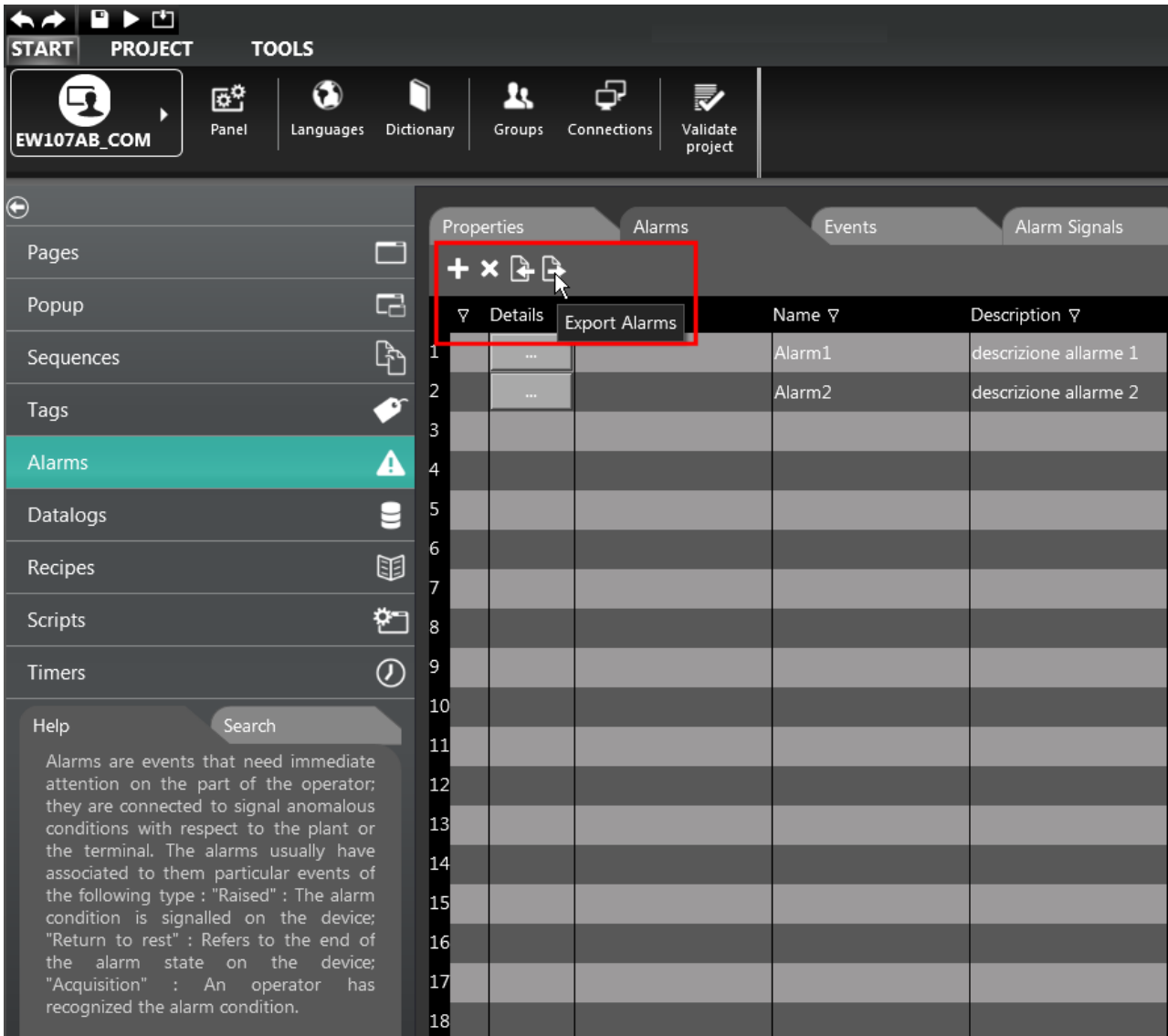
	Det.	Name ▾	Description ▾
1	...	Alarm1	descrizione allarme 1
2	...	Alarm2	descrizione allarme 2
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			

**Help** Search

Alarms are events that need immediate attention on the part of the operator; they are connected to signal anomalous conditions with respect to the plant or the terminal. The alarms usually have associated to them particular events of the following type : "Raised" : The alarm condition is signalled on the device; "Return to rest" : Refers to the end of the alarm state on the device; "Acquisition" : An operator has recognized the alarm condition.

# CREW Manual

Export a set of previously created alarms.



The screenshot shows the software interface with the 'Alarms' tab selected. A red box highlights the 'Export Alarms' button in the top toolbar. The main area displays a table of alarms.

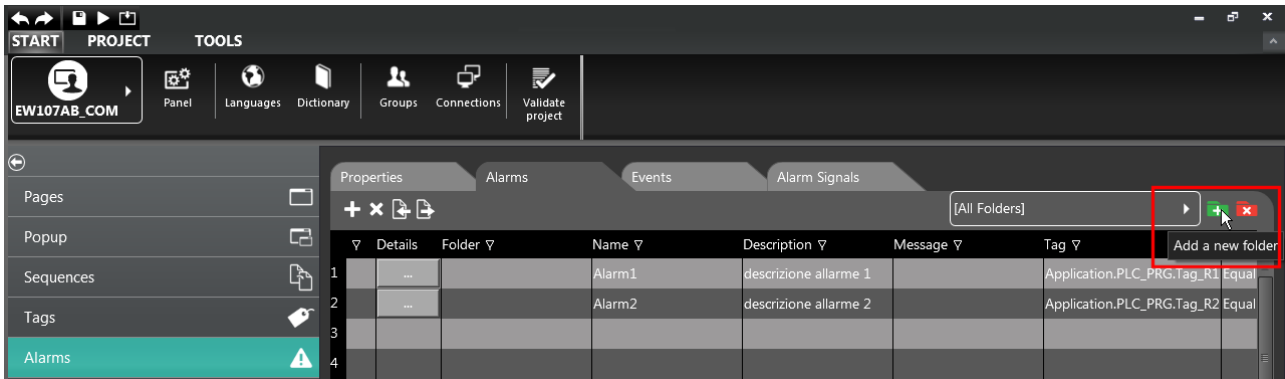
	Details	Name ▾	Description ▾
1	...	Alarm1	descrizione allarme 1
2	...	Alarm2	descrizione allarme 2
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			

**Help** Search

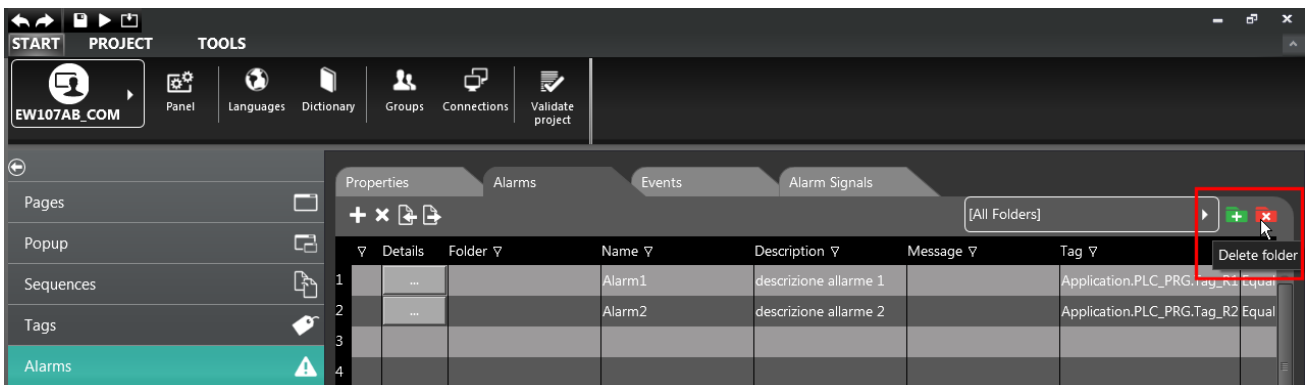
Alarms are events that need immediate attention on the part of the operator; they are connected to signal anomalous conditions with respect to the plant or the terminal. The alarms usually have associated to them particular events of the following type : "Raised" : The alarm condition is signalled on the device; "Return to rest" : Refers to the end of the alarm state on the device; "Acquisition" : An operator has recognized the alarm condition.

# CREW Manual

Create new folders to contain the alarms.



Delete previously created folders.

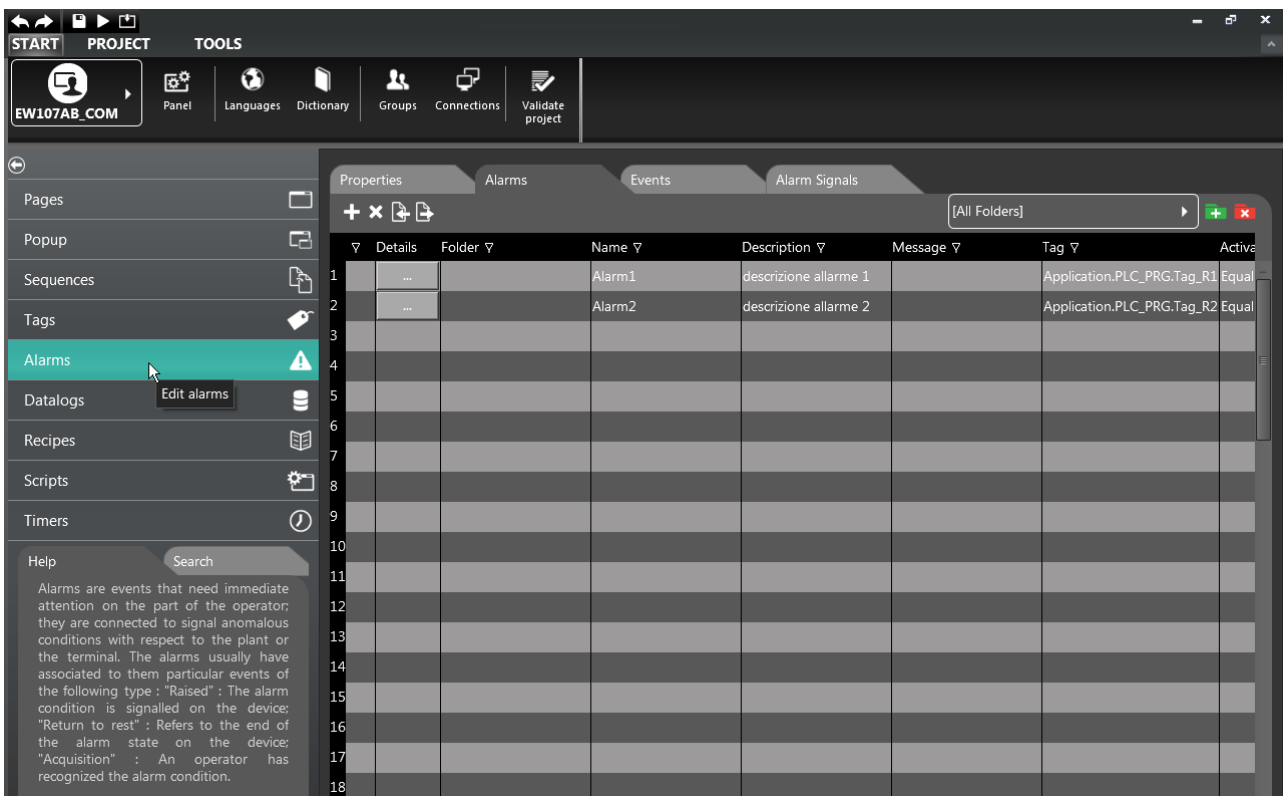


The main alarm editing area is the "[Alarms Grid](#)".

# CREW Manual

## Alarms Grid

The “Alarms Grid” is used to edit the Alarms contained in the project and all of their characteristics (explained below).



Click the "Details" option of the Alarms Grid to access "[Alarms Editor](#)", the main tool used to define the features of the alarms in the project.

The Alarms Grid consists of different columns to which display filters ([Alarms Grid Filters](#)) can be applied.



# CREW Manual

## Alarms Grid Filters

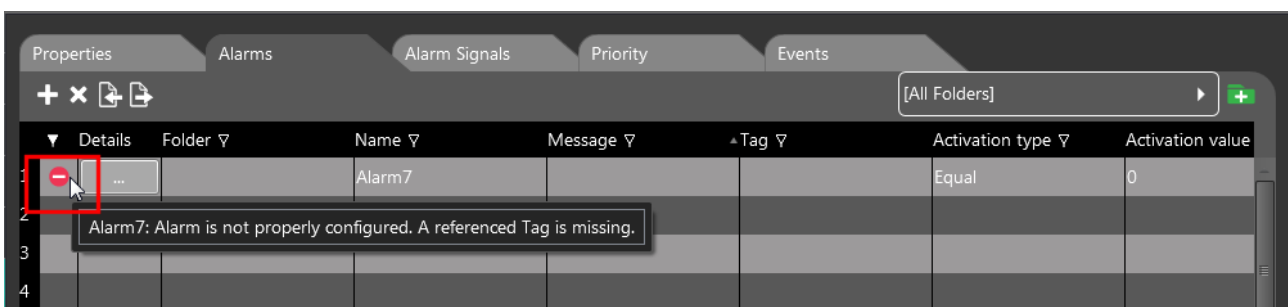
The user can enable or disable viewing filters available on each column of the Alarms Grid to make only certain types of values appear in the table.

The columns to which display filters can be applied are the following:

- Details
- Folder
- Name
- Message
- Tags
- Activation type
- Activation value
- Priority
- Recorded

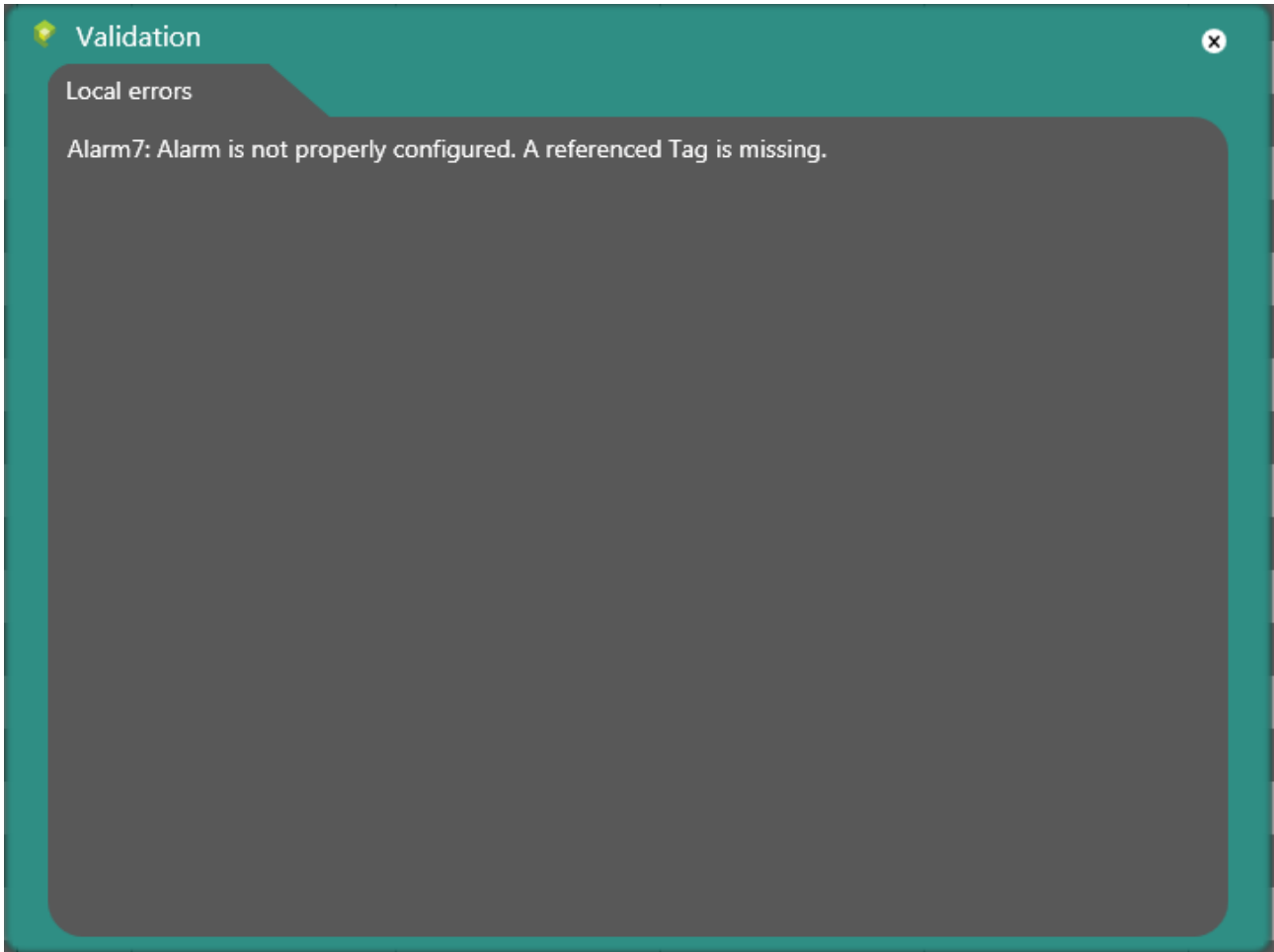
## Validation

The “Validation” column reports any association errors regarding the value of the alarms. A red symbol next to the progressive number makes it possible to immediately see when there are any errors.



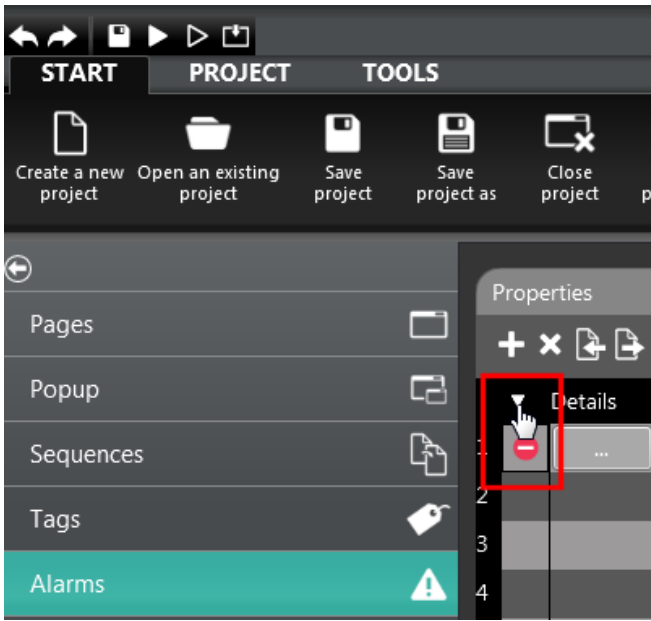
# CREW Manual

Click the error symbol to view the detailed error window.

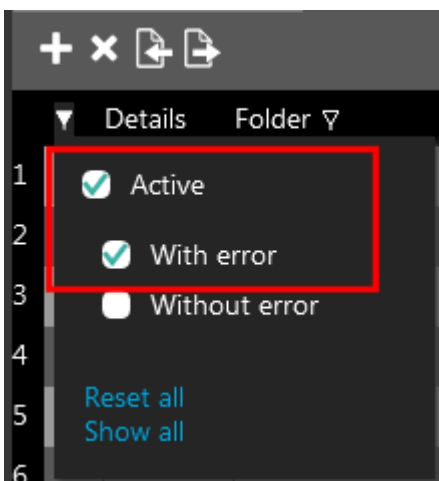


# CREW Manual

In the “Validation” column, enable or disable the error filter to view all of the alarms with or without errors.

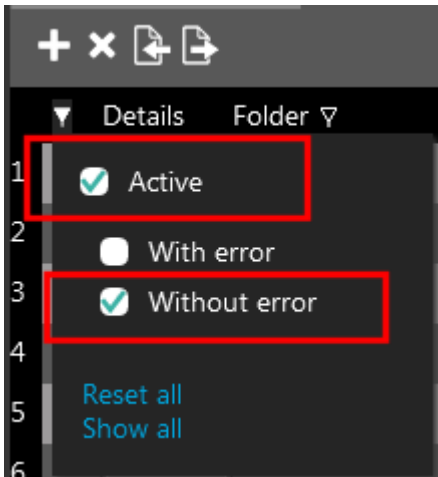


To display alarms with errors, enable the "With Errors" filter.



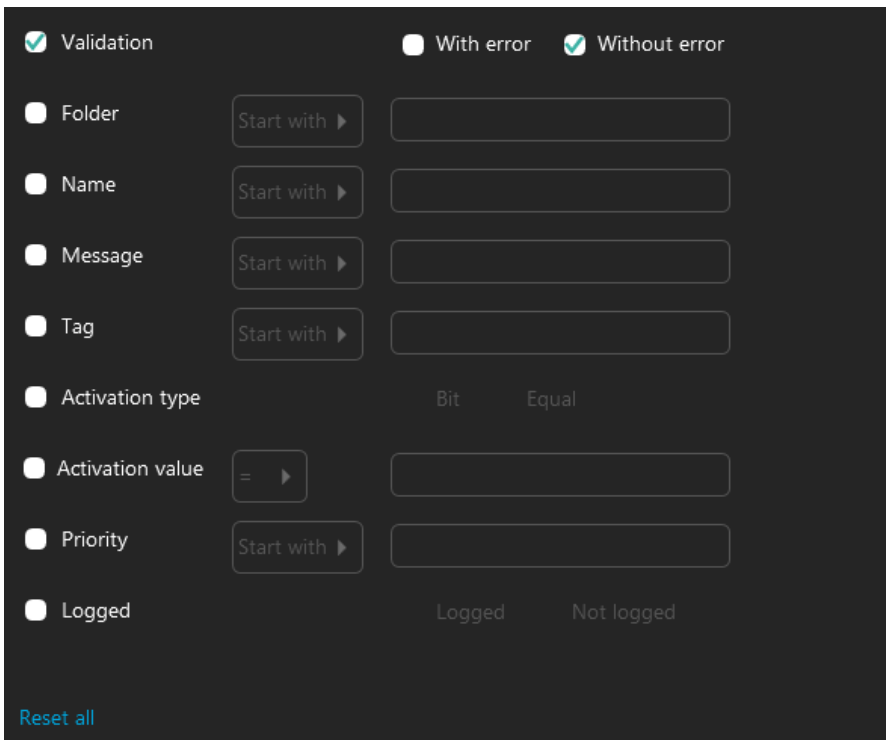
# CREW Manual

To display alarms without errors, enable the "Without Errors" filter.



The "Reset all" option deletes all enabled filters.

The "Show all" option displays in a single window all options to which display filters can be applied.

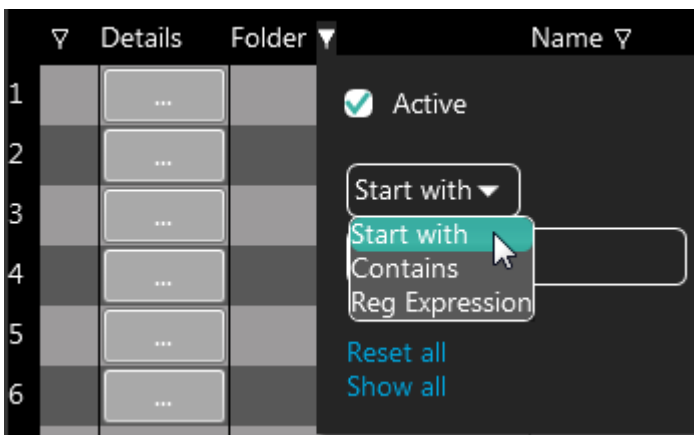


# CREW Manual

## Folder

In the "Folder" it is possible to view the folders contained in CREW according to one of the following viewing filters:

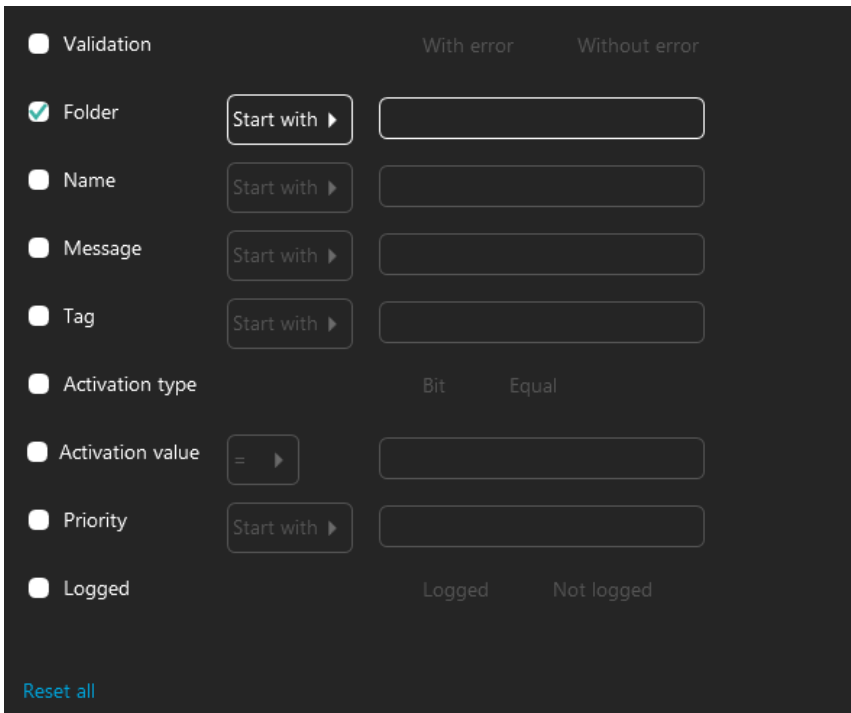
- **Beginning with:** the folders with names that start with the letter chosen by the user are displayed.
- **Containing:** the folders with names that contain the word chosen by the user are displayed.
- **Regular expression:** this displays the folders with names that match the chosen regular expression.



The "Reset all" option deletes all enabled filters.

The "Show all" option displays in a single window all options to which display filters can be applied.

# CREW Manual



The screenshot shows a dark-themed filter configuration panel. At the top, there are radio buttons for 'Validation' (unselected), 'With error', and 'Without error'. Below this, several filter categories are listed, each with a radio button and a corresponding input field:

- Folder:** Checked. Input field: 'Start with' followed by a text box.
- Name:** Unselected. Input field: 'Start with' followed by a text box.
- Message:** Unselected. Input field: 'Start with' followed by a text box.
- Tag:** Unselected. Input field: 'Start with' followed by a text box.
- Activation type:** Unselected. Radio buttons for 'Bit' and 'Equal'.
- Activation value:** Unselected. Input field: '=' followed by a text box.
- Priority:** Unselected. Input field: 'Start with' followed by a text box.
- Logged:** Unselected. Radio buttons for 'Logged' and 'Not logged'.

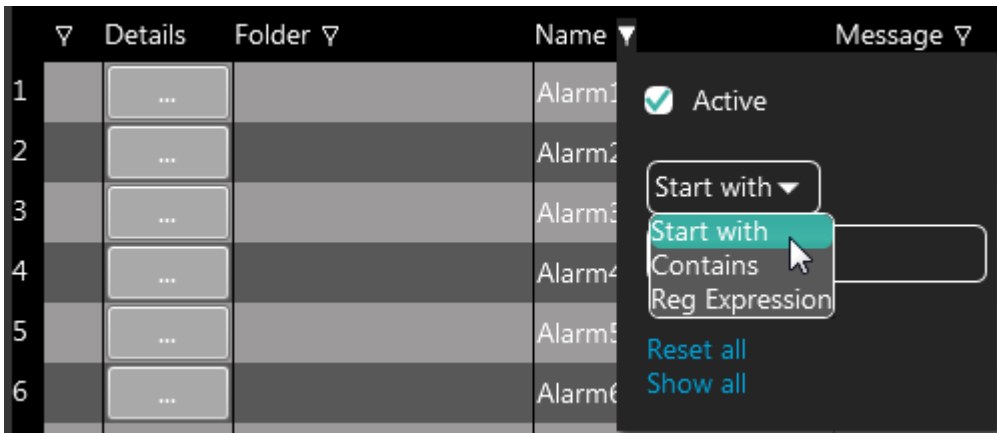
A 'Reset all' link is located at the bottom left of the panel.

## Name

In the "Name" column it is possible to apply one of the following viewing filters:

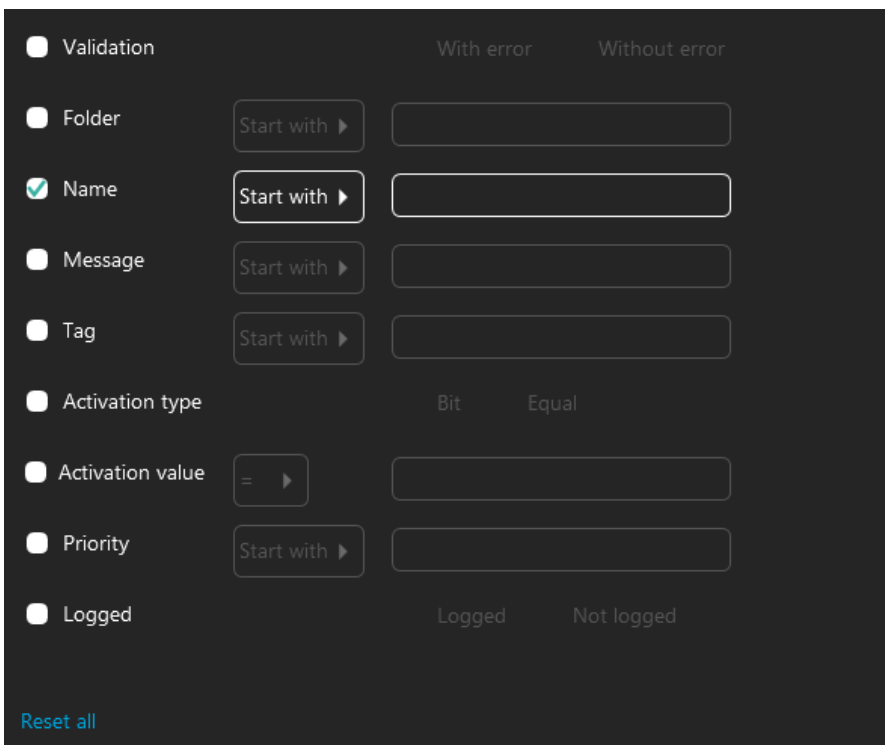
- **Beginning with:** the alarms with names that start with the letter chosen by the user are displayed.
- **Containing:** the alarms with names that contain the word chosen by the user are displayed.
- **Regular expression:** this displays the alarms with names that match the chosen regular expression.

# CREW Manual



The "Reset all" option deletes all enabled filters.

The "Show all" option displays in a single window all options to which display filters can be applied.



# CREW Manual

## Message

In the "Message" column it is possible to apply one of the following viewing filters:

- **Beginning with:** the alarms whose "Message" starts with the letter chosen by the user are displayed.
- **Containing:** the alarms whose "Message" contains the word chosen by the user are displayed.
- **Regular expression:** this displays the alarms with "Messages" with characters contained in the chosen regular expression.

	Details	Folder	Name	Message	Tag
1	...		Alarm1	Alarm 1	<input checked="" type="checkbox"/> Active
2	...		Alarm2	Alarm 2	
3	...		Alarm3	Alarm 3	
4	...		Alarm4	Alarm 4	
5	...		Alarm5	Alarm 5	
6	...		Alarm6	Alarm 6	

Start with ▾

Start with

Contains

Reg Expression

Reset all

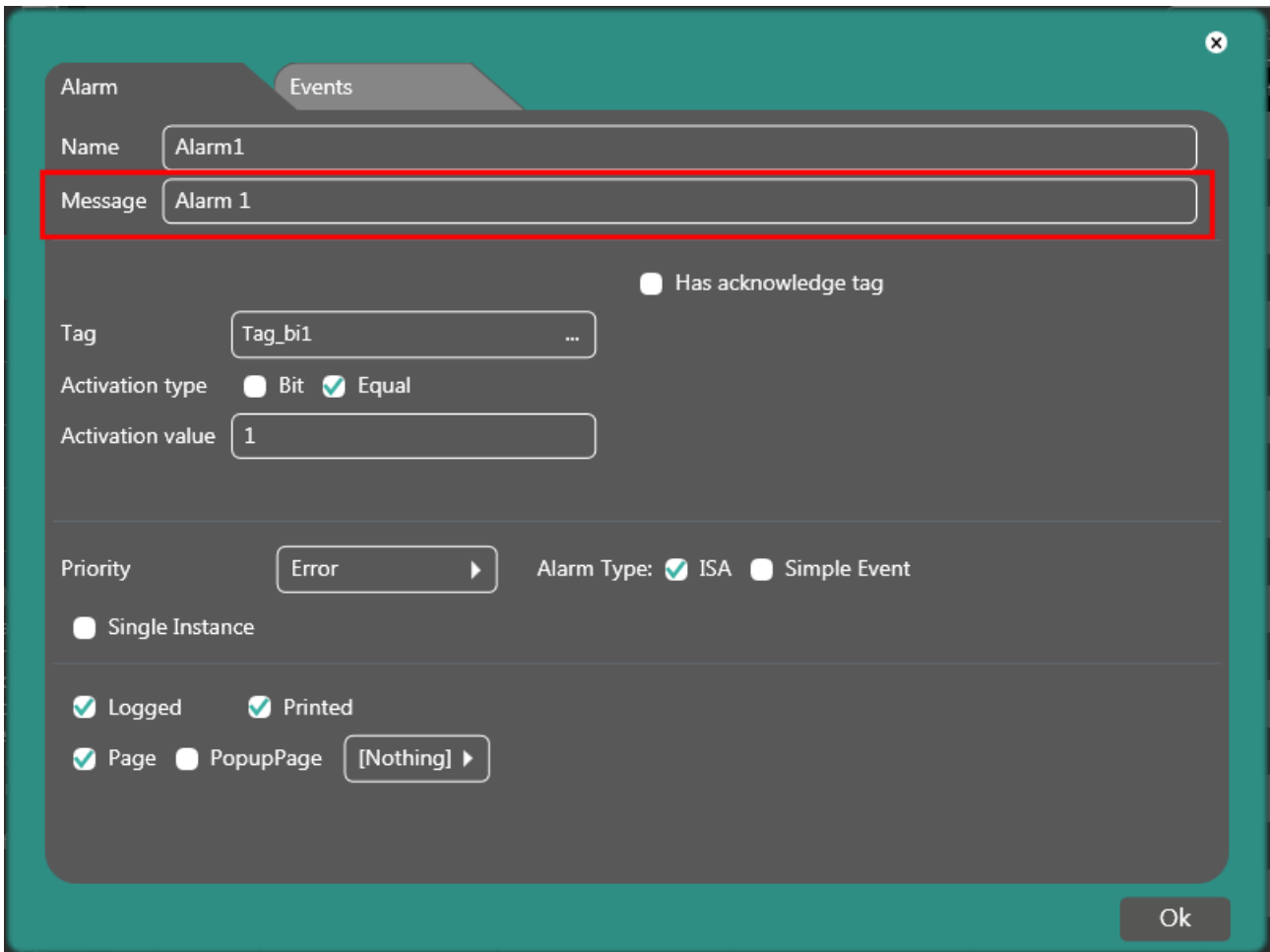
Show all



# CREW Manual



Note: The “Message” description of an alarm is found in the section marked in the image.

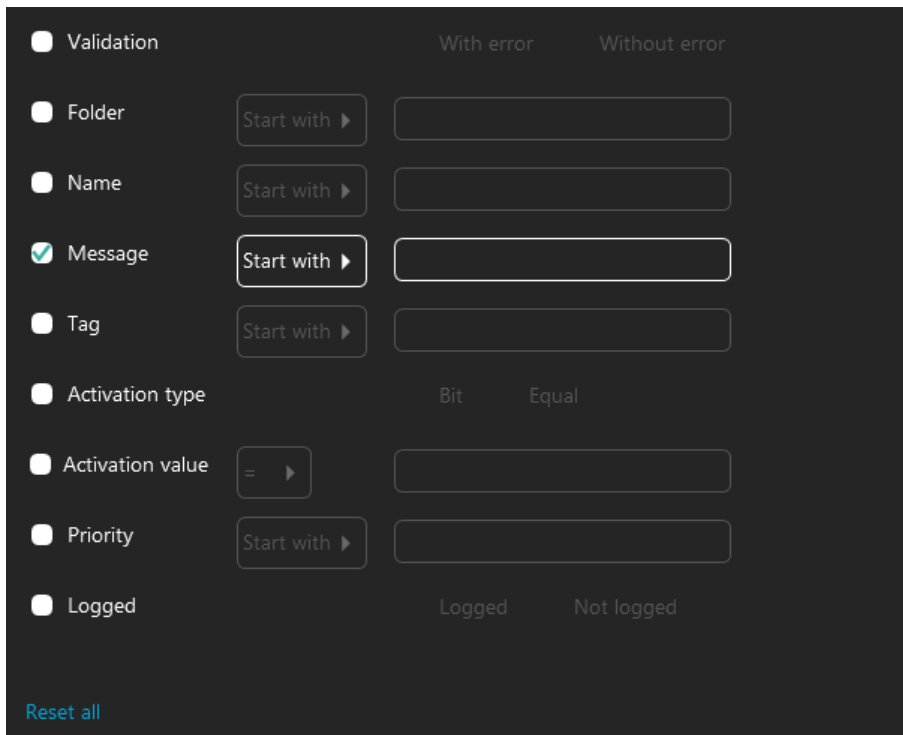


The screenshot shows a configuration window for an alarm. The window has two tabs: 'Alarm' and 'Events'. The 'Alarm' tab is active. The 'Name' field contains 'Alarm1'. The 'Message' field contains 'Alarm 1' and is highlighted with a red border. Below the 'Message' field, there is a checkbox for 'Has acknowledge tag' which is unchecked. The 'Tag' field contains 'Tag\_bi1'. The 'Activation type' has radio buttons for 'Bit' (unchecked) and 'Equal' (checked). The 'Activation value' field contains '1'. The 'Priority' dropdown is set to 'Error'. The 'Alarm Type' has radio buttons for 'ISA' (checked) and 'Simple Event' (unchecked). There is a checkbox for 'Single Instance' which is unchecked. At the bottom, there are checkboxes for 'Logged' (checked) and 'Printed' (checked). There are also checkboxes for 'Page' (checked) and 'PopupPage' (unchecked), followed by a dropdown menu set to '[Nothing]'. An 'Ok' button is located at the bottom right of the dialog.

# CREW Manual

The "Reset all" option deletes all enabled filters.

The "Show all" option displays in a single window all options to which display filters can be applied.



The screenshot shows a dark-themed filter configuration panel. It contains several filter categories, each with a radio button and a corresponding input field or options. The "Message" filter is currently selected and highlighted with a white border. At the bottom left, there is a "Reset all" link.

<input type="checkbox"/> Validation		With error	Without error
<input type="checkbox"/> Folder	Start with ▶	<input type="text"/>	
<input type="checkbox"/> Name	Start with ▶	<input type="text"/>	
<input checked="" type="checkbox"/> Message	Start with ▶	<input type="text"/>	
<input type="checkbox"/> Tag	Start with ▶	<input type="text"/>	
<input type="checkbox"/> Activation type		Bit	Equal
<input type="checkbox"/> Activation value	= ▶	<input type="text"/>	
<input type="checkbox"/> Priority	Start with ▶	<input type="text"/>	
<input type="checkbox"/> Logged		Logged	Not logged

[Reset all](#)

# CREW Manual

## Tag

In the "Tag" column it is possible to apply one of the following viewing filters based on the tag associated to the alarm:

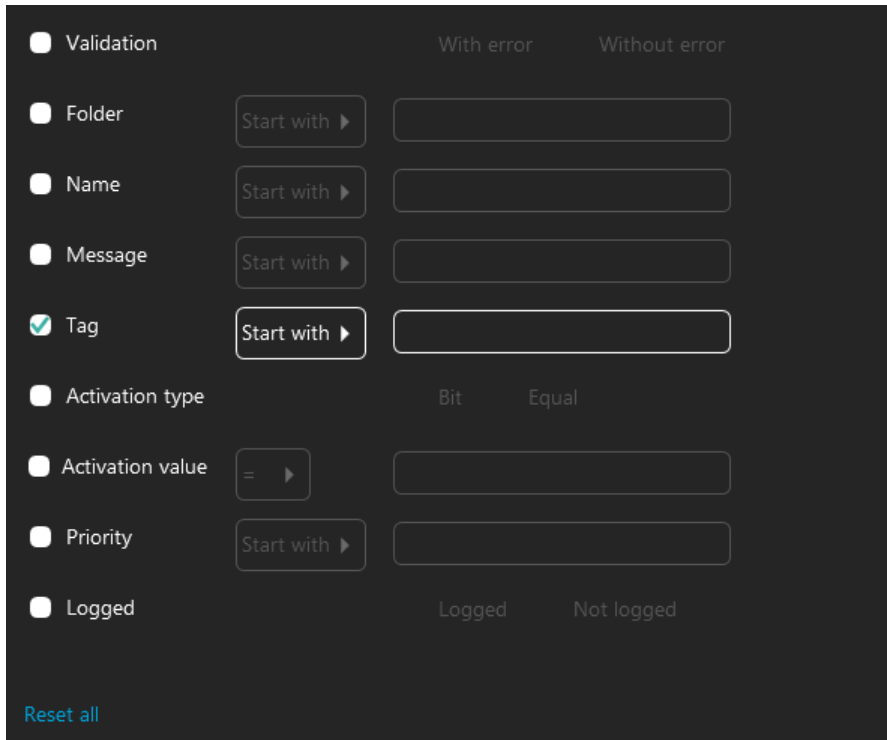
- **Beginning with:** the alarms whose associated variables start with the letter chosen by the user are displayed.
- **Containing:** the alarms whose associated variables contain the word chosen by the user are displayed.
- **Regular expression:** this displays the alarms whose associated variables include characters contained in the chosen regular expression.

Name ▾	Message ▾	Tag ▾
Alarm1	Alarm 1	Tag_ <input checked="" type="checkbox"/> Active
Alarm2	Alarm 2	Tag_ Start with ▾
Alarm3	Alarm 3	Tag_ Start with
Alarm4	Alarm 4	Tag_ Contains
Alarm5	Alarm 5	Tag_ Reg Expression
Alarm6	Alarm 6	Tag_ Reset all
		Tag_ Show all

The "Reset all" option deletes all enabled filters.

# CREW Manual

The "Show all" option displays in a single window all options to which display filters can be applied.



The screenshot shows a dark-themed filter configuration panel with the following elements:

- Validation: With error, Without error
- Folder: Start with ▶ [input field]
- Name: Start with ▶ [input field]
- Message: Start with ▶ [input field]
- Tag: Start with ▶ [input field]
- Activation type: Bit, Equal
- Activation value: = ▶ [input field]
- Priority: Start with ▶ [input field]
- Logged: Logged, Not logged

[Reset all](#)

# CREW Manual

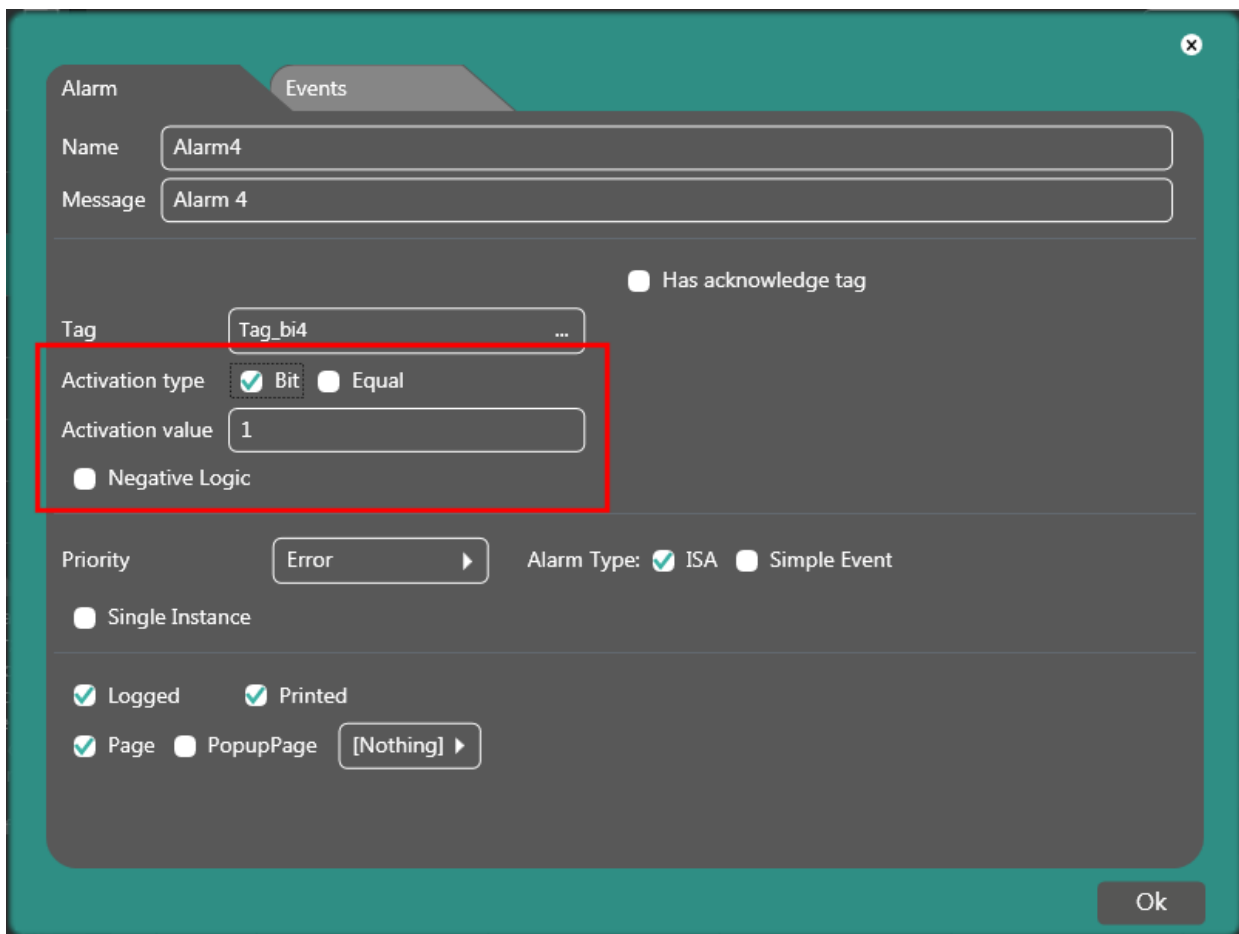
## Activation type

In the "Activation type" column it is possible to apply one of the following viewing filters based on the type of alarm activation:

- Bit: when the alarm is activated by a change to a particular bit of the reference variable associated to the alarm itself.
- Equal: when the alarm is activated by reaching the value set in the reference variable associated to the alarm itself.



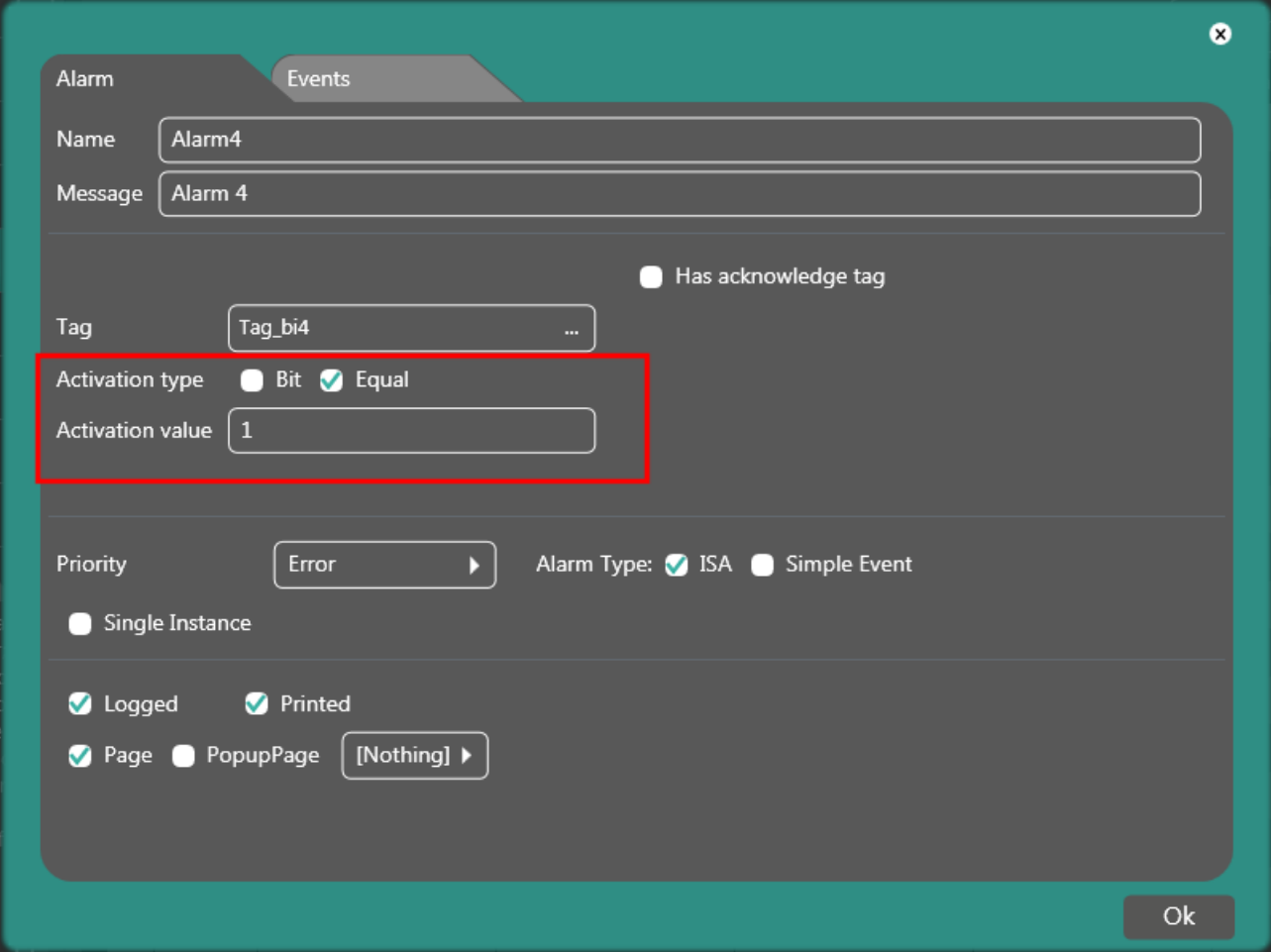
Note: The "Activation type" of an alarm is found in the section marked in the image.



The screenshot shows the 'Alarm' configuration dialog box with the following fields and options:

- Name: Alarm4
- Message: Alarm 4
- Tag: Tag\_bi4
- Activation type:  Bit,  Equal
- Activation value: 1
- Negative Logic:
- Has acknowledge tag:
- Priority: Error
- Alarm Type:  ISA,  Simple Event
- Single Instance:
- Logged:  Printed:
- Page:  PopupPage:  [Nothing]

# CREW Manual



Alarm    Events

Name

Message

Tag  ...

Has acknowledge tag

Activation type  Bit  Equal

Activation value

Priority  Alarm Type:  ISA  Simple Event

Single Instance

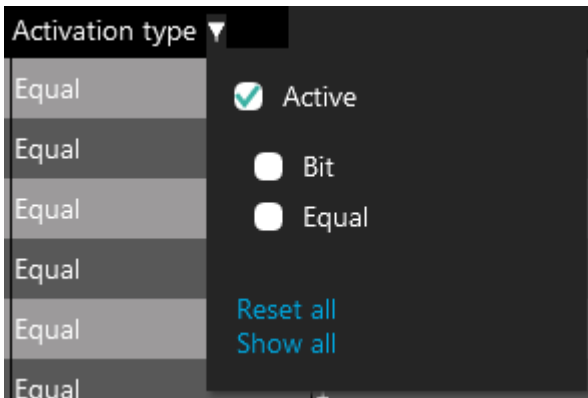
Logged  Printed

Page  PopupPage

Ok

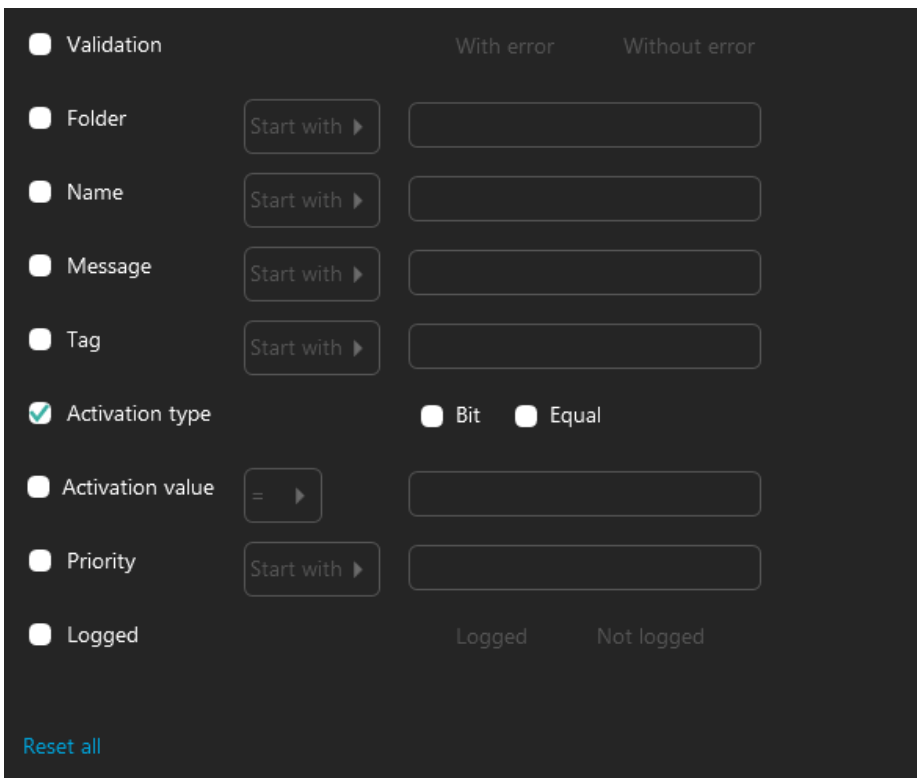
- Bit: the alarms whose variables include “Bit” activation are displayed.
- Equal: the alarms whose variables include “Equal” activation are displayed.

# CREW Manual



The "Reset all" option deletes all enabled filters.

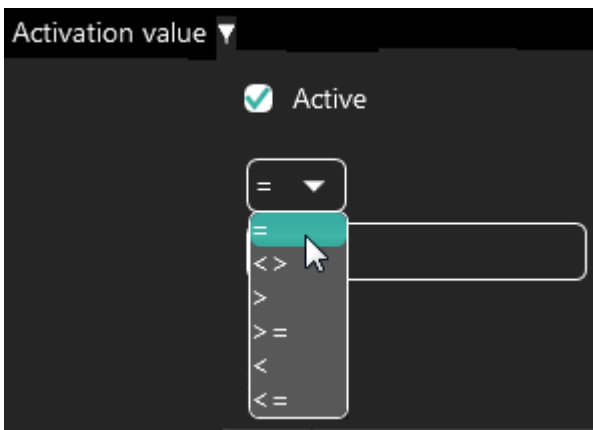
The "Show all" option displays in a single window all options to which display filters can be applied.



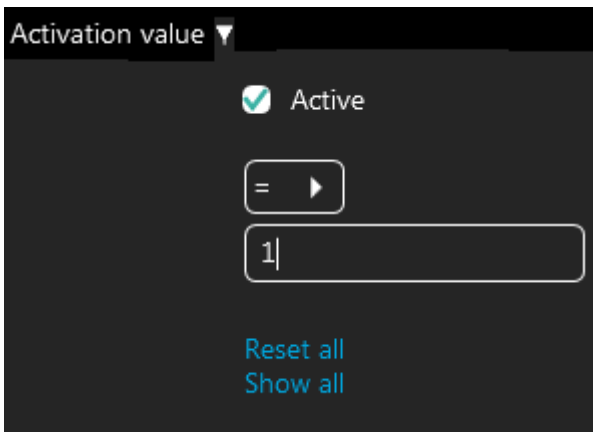
# CREW Manual

## Activation value

In the "Activation value" column it is possible to apply one of the following viewing filters based on the alarm activation value: "equal to" (=), "different from" (<>), "greater than" (>), "greater than or equal to" (>=), "less than" (<), "less than or equal to" (<=).

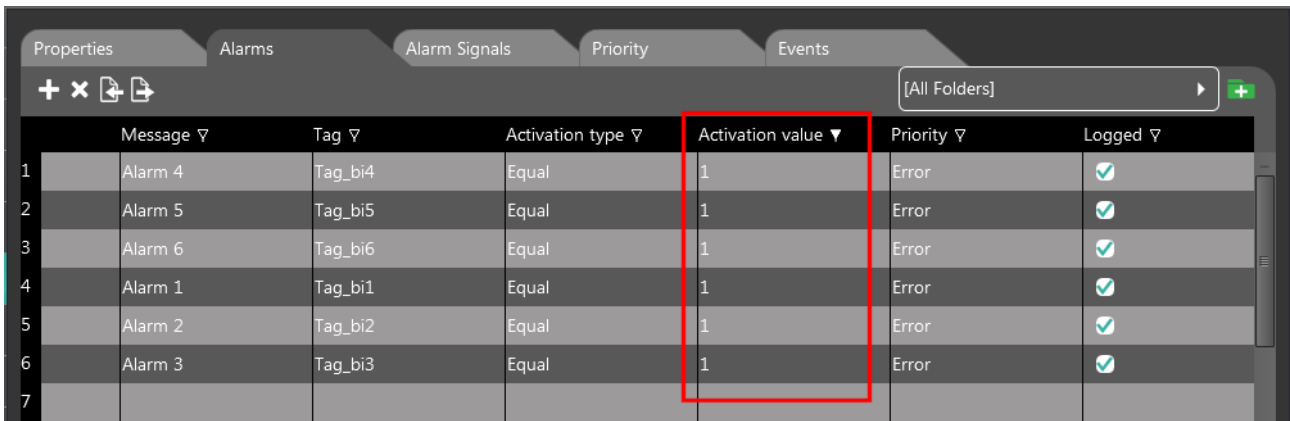


For example, by choosing "=" to "1" all of the alarms with an activation value equal to 1 are displayed.





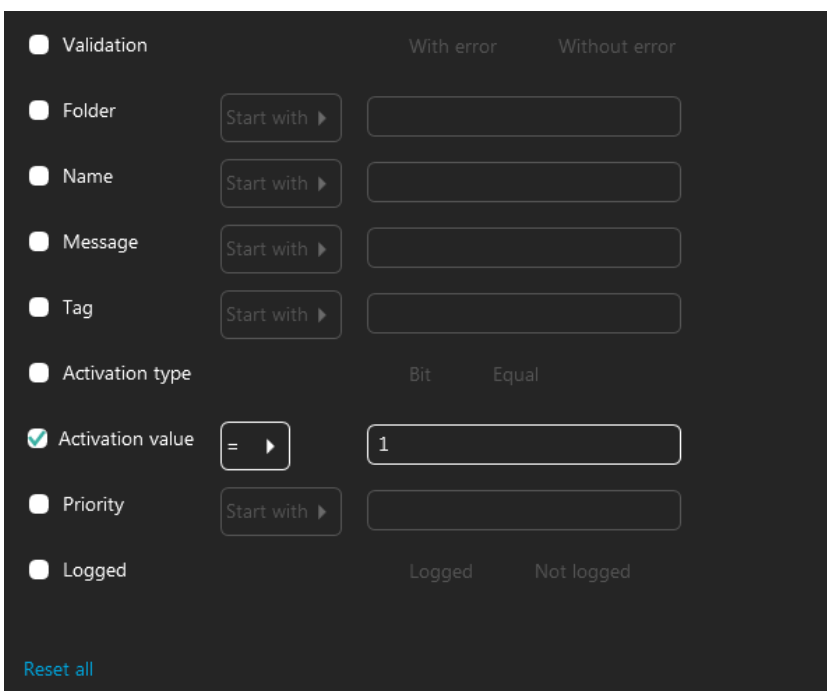
# CREW Manual



	Message ▾	Tag ▾	Activation type ▾	Activation value ▾	Priority ▾	Logged ▾
1	Alarm 4	Tag_bi4	Equal	1	Error	✓
2	Alarm 5	Tag_bi5	Equal	1	Error	✓
3	Alarm 6	Tag_bi6	Equal	1	Error	✓
4	Alarm 1	Tag_bi1	Equal	1	Error	✓
5	Alarm 2	Tag_bi2	Equal	1	Error	✓
6	Alarm 3	Tag_bi3	Equal	1	Error	✓
7						

The "Reset all" option deletes all enabled filters.

The "Show all" option displays in a single window all options to which display filters can be applied.

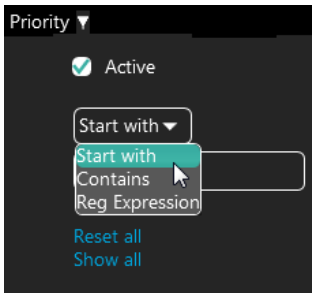


Validation      With error      Without error  
 Folder      Start with ▶        
 Name      Start with ▶        
 Message      Start with ▶        
 Tag      Start with ▶        
 Activation type      Bit      Equal  
 Activation value      = ▶        
 Priority      Start with ▶        
 Logged      Logged      Not logged  
[Reset all](#)

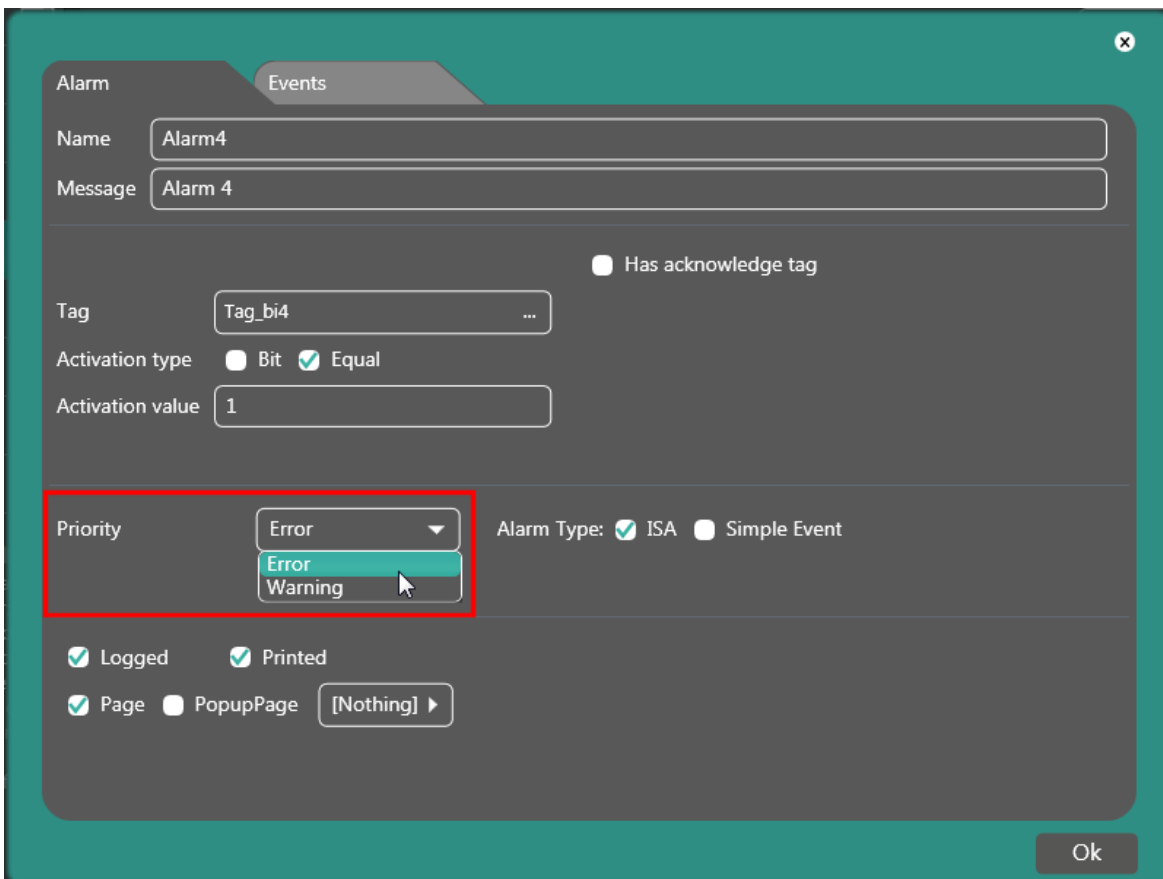
# CREW Manual

## Priority

In the "Priority" column it is possible to apply the display filter based on the set priority value.



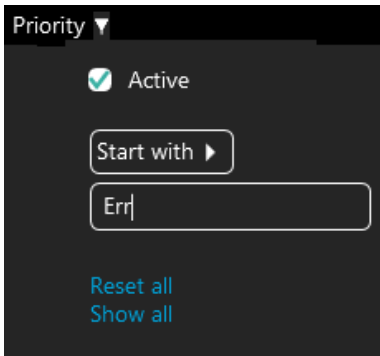
Note: The "Priority" value of an alarm is found in the section shown in the image.



# CREW Manual

An alarm can have two kinds of "Priorities": "Error" or "Warning".

If you choose "Beginning with", for example, and the letter "Err" is entered as reference, all of the alarms with priority value "100" (Error) will be displayed, as described in the "[Alarms Editor](#)" section (see "Priority").

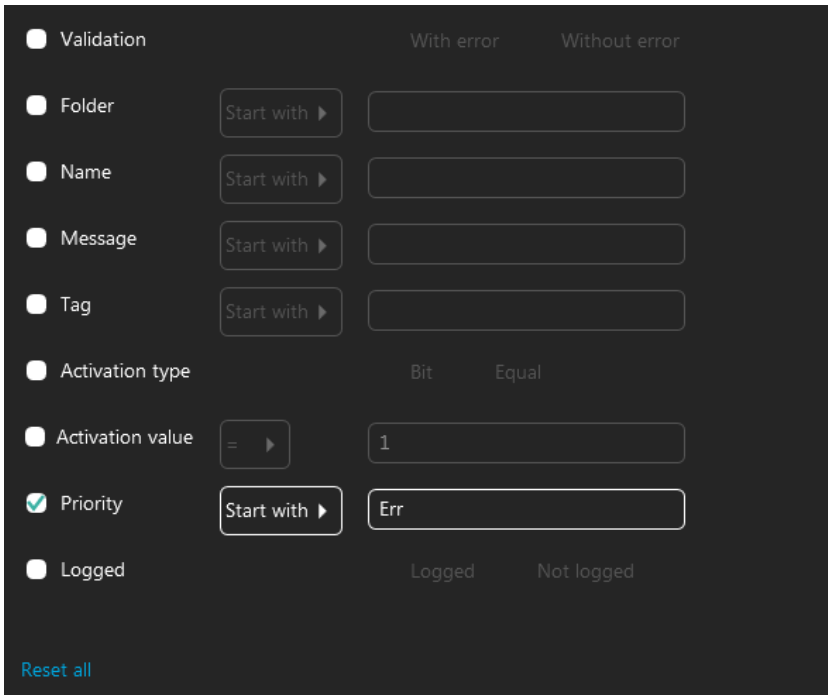


	Message ▾	Tag ▾	Activation type ▾	Activation value ▾	Priority ▾	Logged ▾
1	Alarm 4	Tag_bi4	Equal	1	Error	✓
2	Alarm 5	Tag_bi5	Equal	1	Error	✓
3	Alarm 6	Tag_bi6	Equal	1	Error	✓
4	Alarm 1	Tag_bi1	Equal	1	Error	✓
5	Alarm 2	Tag_bi2	Equal	1	Error	✓
6	Alarm 3	Tag_bi3	Equal	1	Error	✓
7						

The "Reset all" option deletes all enabled filters.

# CREW Manual

The "Show all" option displays in a single window all options to which display filters can be applied.



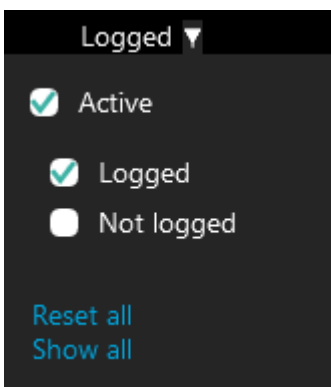
The screenshot shows a dark-themed filter configuration panel with the following options:

- Validation: With error, Without error
- Folder: Start with ▶, [text input]
- Name: Start with ▶, [text input]
- Message: Start with ▶, [text input]
- Tag: Start with ▶, [text input]
- Activation type: Bit, Equal
- Activation value: = ▶, [text input: 1]
- Priority: Start with ▶, [text input: Err]
- Logged: Logged, Not logged

Reset all

## Recorded

In the "Recorded" column it is possible to apply one of the two "Recorded" or "Not Recorded" viewing filters to the alarms.



The screenshot shows a dropdown menu for the "Recorded" filter with the following options:

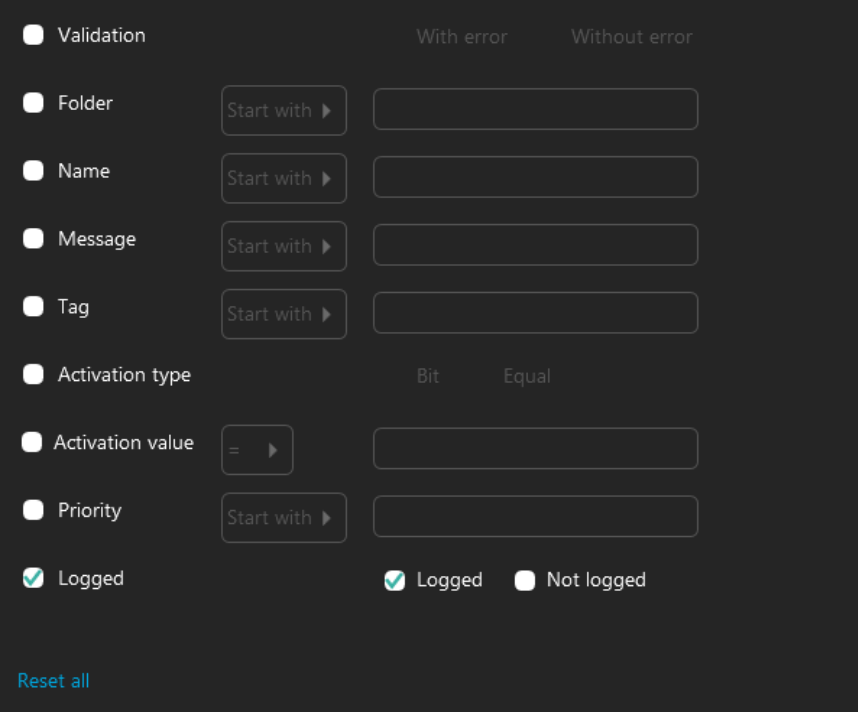
- Logged ▼
- Active
- Logged
- Not logged

Reset all  
Show all

# CREW Manual

The "Reset all" option deletes all enabled filters.

The "Show all" option displays in a single window all options to which display filters can be applied.



The screenshot shows a dark-themed filter configuration panel. It contains the following elements:

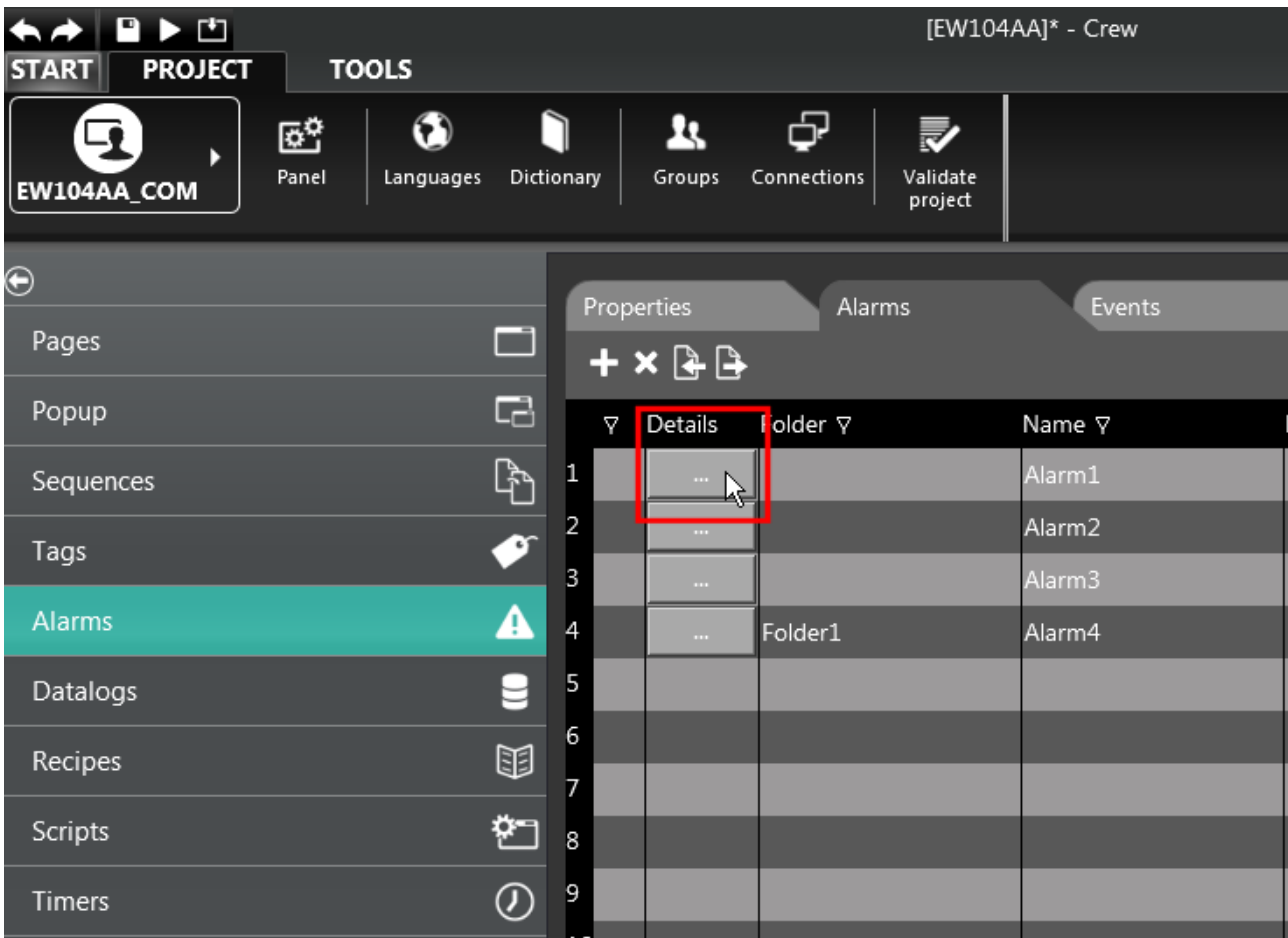
- Validation: With error / Without error
- Folder: Start with ▶ [input field]
- Name: Start with ▶ [input field]
- Message: Start with ▶ [input field]
- Tag: Start with ▶ [input field]
- Activation type: Bit / Equal
- Activation value: = ▶ [input field]
- Priority: Start with ▶ [input field]
- Logged:  Logged  Not logged

Reset all

# CREW Manual

## Alarms Editor

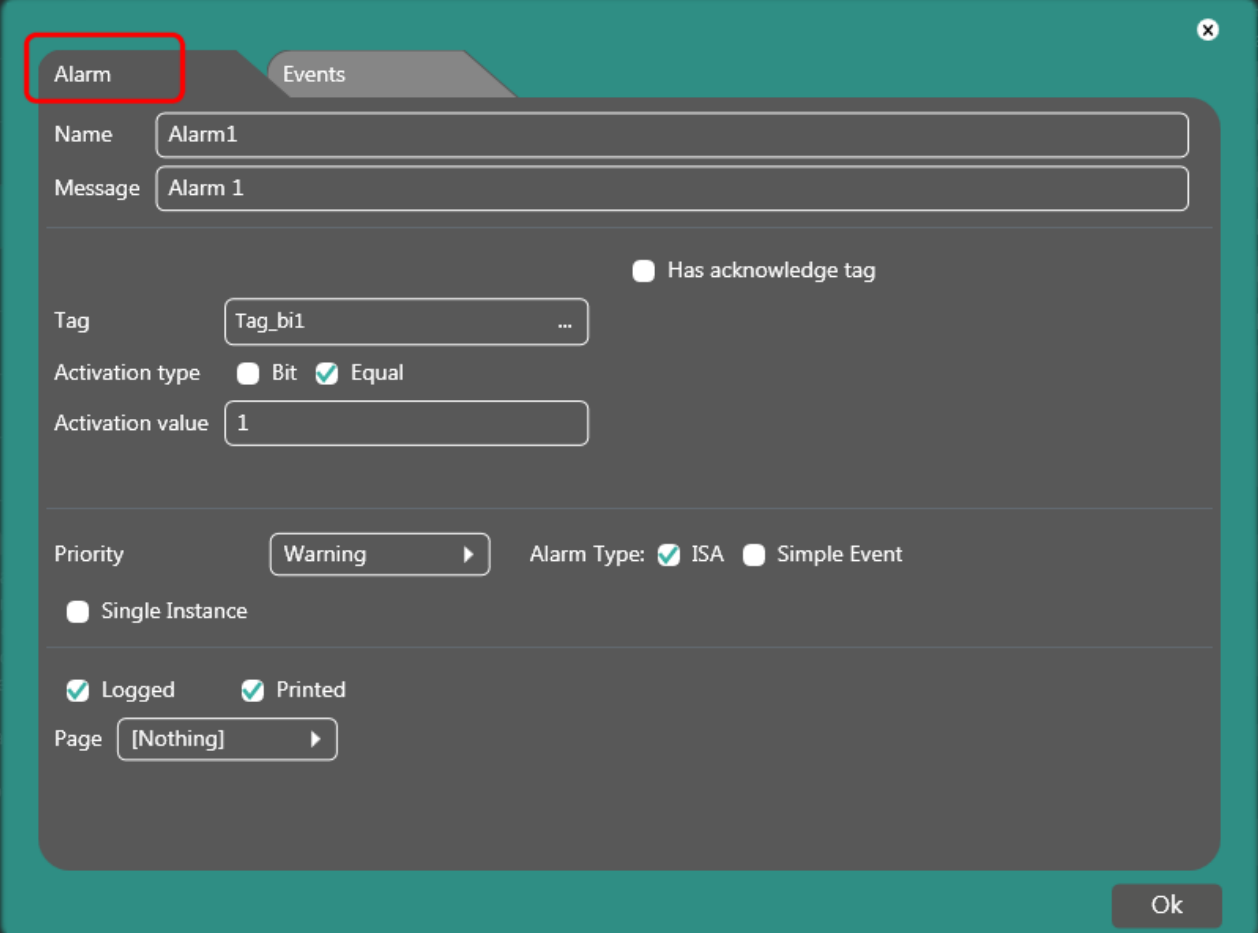
The Alarms Editor is the tool used to change the alarms and define the features of the alarms in the project. To access the “Alarms Editor”, click the relative key in the “Details” column.



# CREW Manual

## Alarm

The first window of the Alarms Editor is the "Alarm" option.



The screenshot shows the "Alarm" configuration window. The "Alarm" tab is highlighted with a red box. The window contains the following fields and options:

- Name: Alarm1
- Message: Alarm 1
- Has acknowledge tag:
- Tag: Tag\_bi1
- Activation type:  Bit  Equal
- Activation value: 1
- Priority: Warning
- Alarm Type:  ISA  Simple Event
- Single Instance:
- Logged:  Printed:
- Page: [Nothing]

An "Ok" button is located at the bottom right of the window.

The "Alarm" option includes, by default, the following editing areas :

- Name
- Message
- Tags
- Identification tag
- Activation type
- Activation value
- Priority

# CREW Manual

- Alarm type
- Single instance
- Logged / Printed
- Page

## Name:

This is the name of the alarm and it must be unique, i.e. there cannot be different alarms with the same name.

## Message:

This refers to the message that appears in the active alarms table. Therefore, the entered message (for example "HIGH TEMPERATURE")

Name	Alarm1
Message	HIGH TEMPERATURE



# CREW Manual

will appear when the alarm occurs.



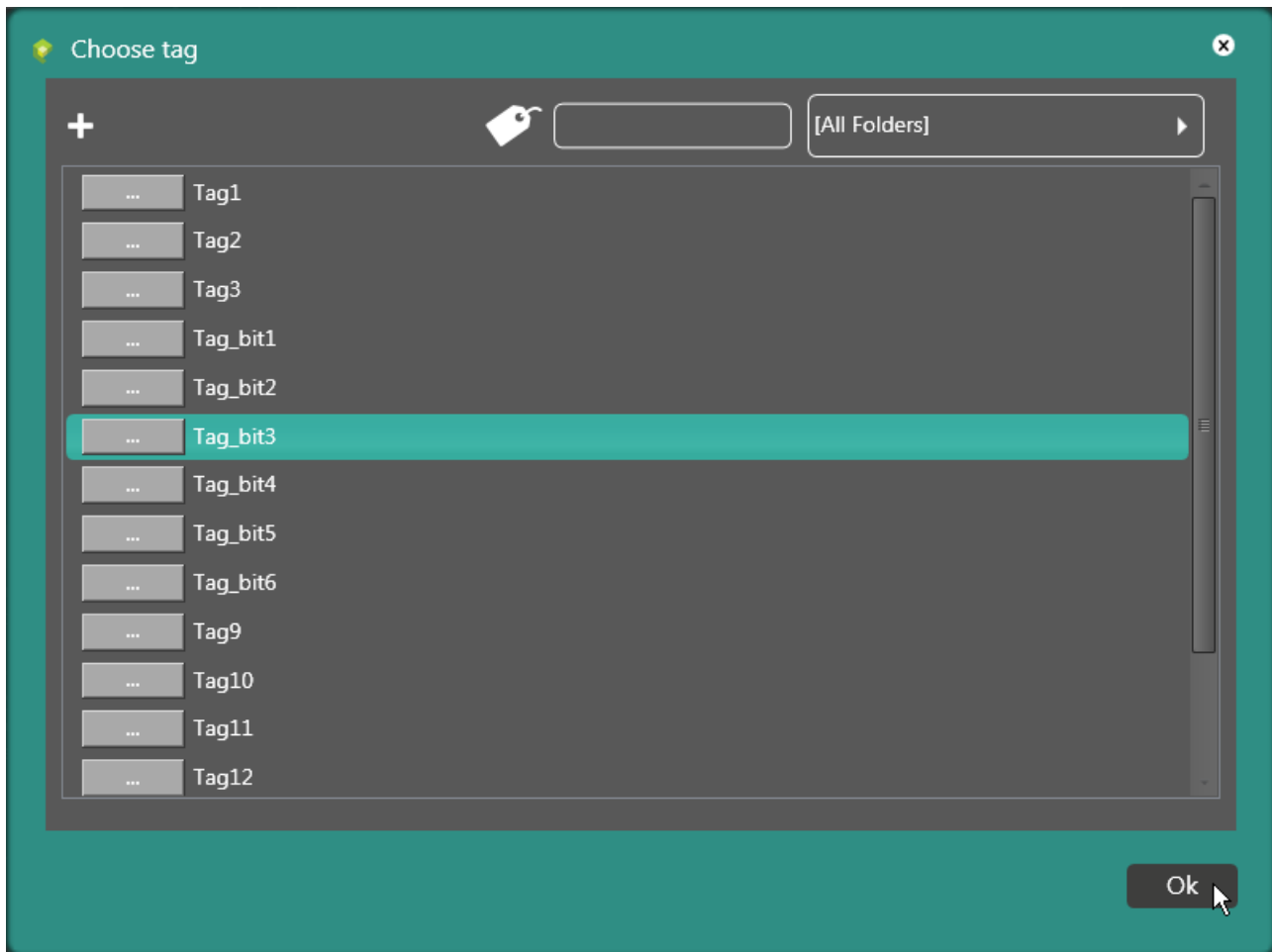
## Tag:

This is the variable associated to the alarm. Click the "Browse" icon to choose the tag to associate to the alarm from those contained in the project.



# CREW Manual

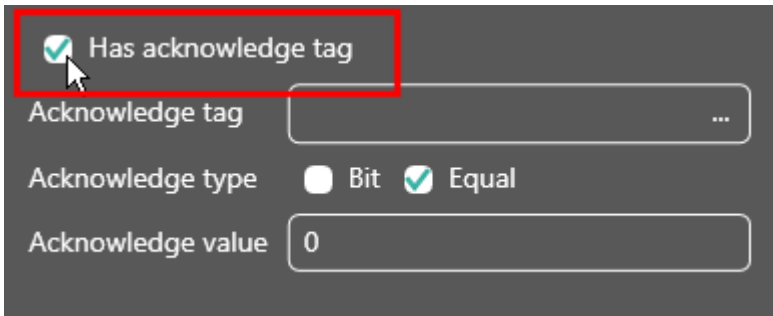
Select the required tag and confirm with “Ok”.



# CREW Manual

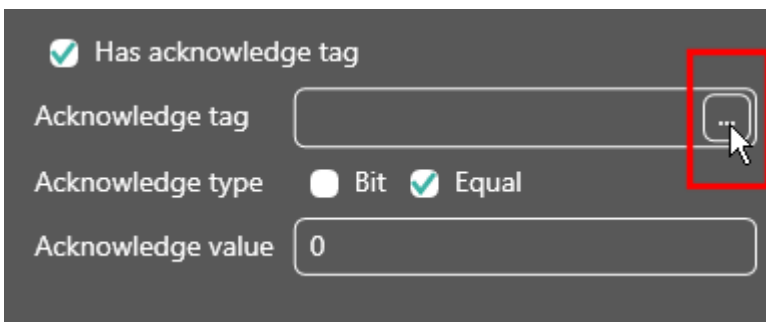
## Identification Tag:

The "Identification Tag" option allows you to use a variable to "Identify" the alarm.



A screenshot of a configuration panel with a dark background. At the top, there is a checkbox labeled "Has acknowledge tag" which is checked. This checkbox and its label are enclosed in a red rectangular box. Below this, there are three fields: "Acknowledge tag" with an empty text box and a three-dot menu icon to its right; "Acknowledge type" with two radio buttons, "Bit" (unchecked) and "Equal" (checked); and "Acknowledge value" with a text box containing the number "0".

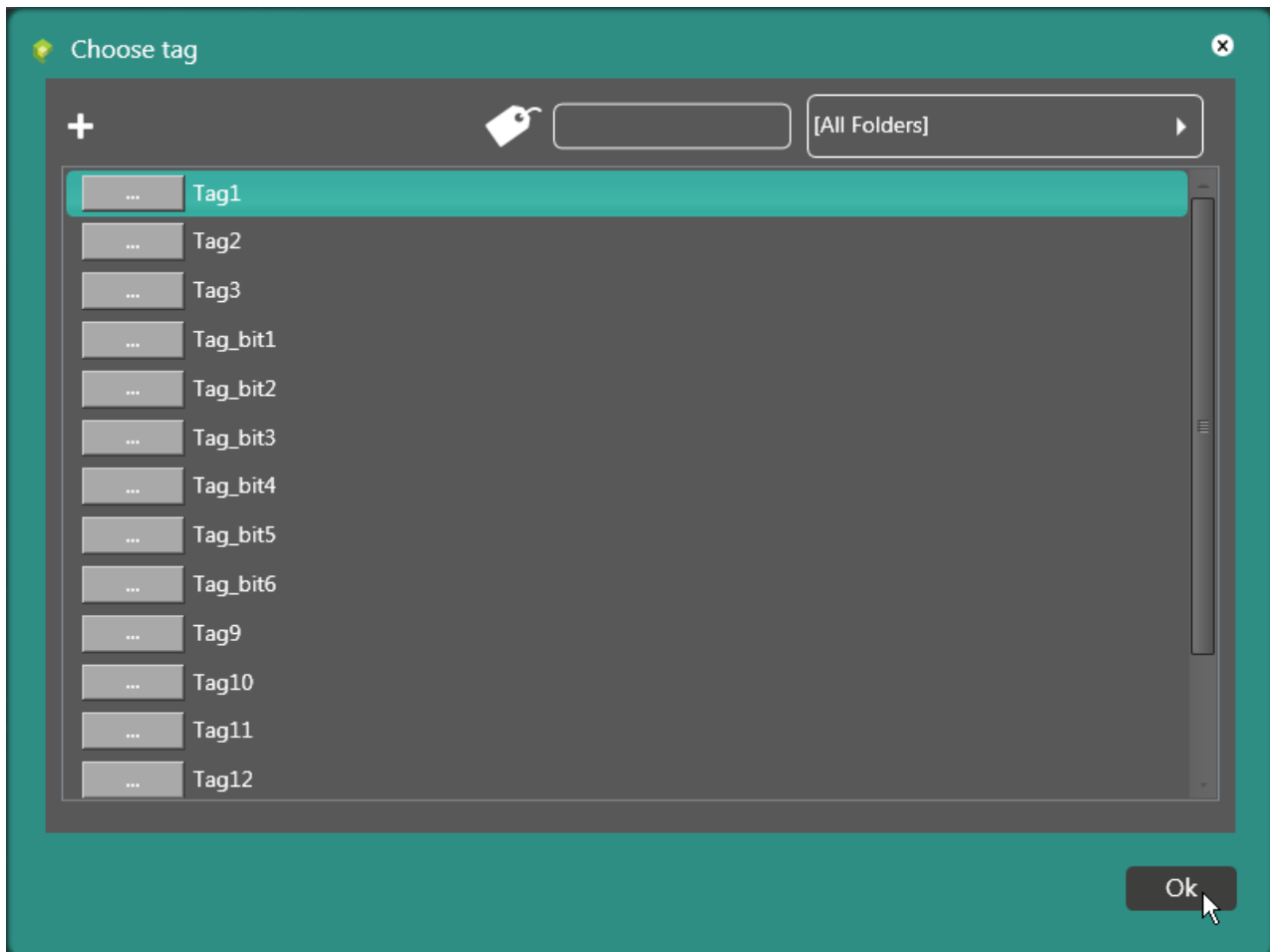
When the option is enabled, you need to choose which tag to use.



A screenshot of the same configuration panel as above. The "Has acknowledge tag" checkbox is checked. In this view, the three-dot menu icon next to the "Acknowledge tag" text box is highlighted with a red rectangular box. A mouse cursor is pointing at this icon.

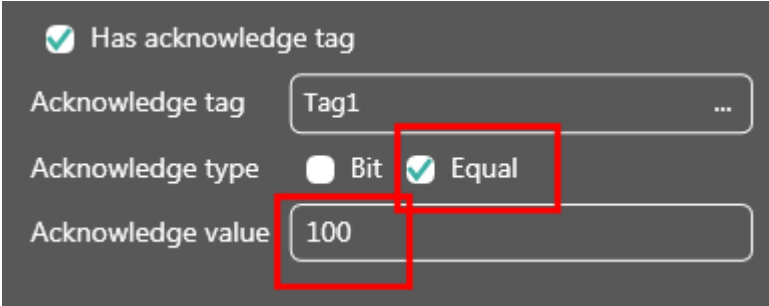
# CREW Manual

Select the Tag and confirm with “Ok”.



# CREW Manual

The "Type of Identification" option can be set as "Bit", in the case of an "Array", or "Equal to", in the case of a single tag where it is necessary to enter a value in the "Identification Value" field.



Has acknowledge tag

Acknowledge tag

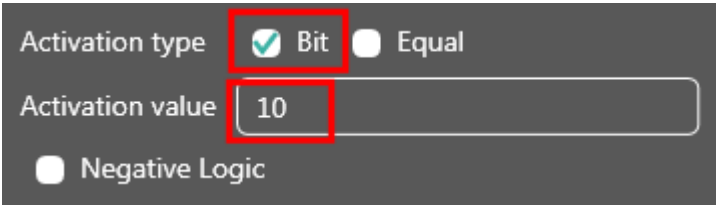
Acknowledge type  Bit  Equal

Acknowledge value

## Activation Type:

This represents the alarm's "Activation type" and can be set as "Bit", in the case of an "Array", or "Equal to", in the case of a single tag. In both cases it is necessary to enter a value in the "Activation Value" field.

"Bit"

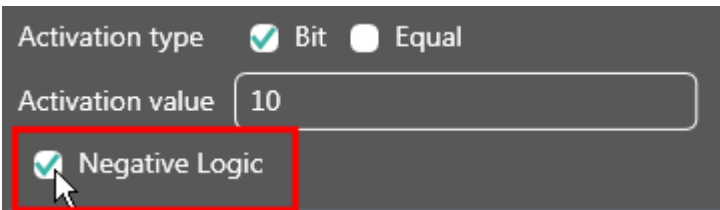


Activation type  Bit  Equal

Activation value

Negative Logic

For example, if you associate a value of "10" to "Bit", the alarm would be activated when bit number 10 takes on level "1". Whereas, if you also enable the "Negative Logic" option, the alarm would be activated when bit 10 switches from "1" to "0".



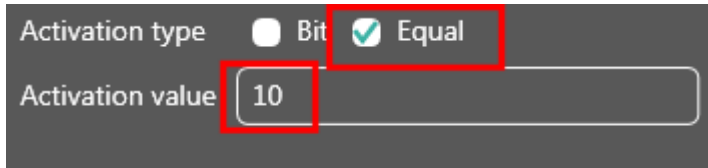
Activation type  Bit  Equal

Activation value

Negative Logic

# CREW Manual

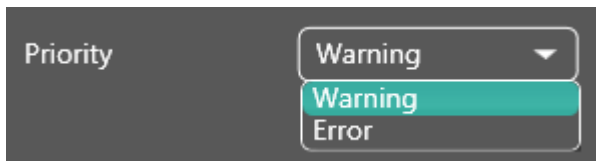
"Equal to":



The screenshot shows a configuration panel with two rows. The first row is labeled 'Activation type' and contains two radio button options: 'Bit' (unselected) and 'Equal' (selected). The second row is labeled 'Activation value' and contains a text input field with the number '10' entered. Red rectangular boxes highlight the 'Equal' radio button and the '10' in the input field.

Whereas, associating a value of "10" to "Equal to", the alarm is activated when the value of the Tag is 10.

Priority:



The screenshot shows a configuration panel with a label 'Priority' on the left and a dropdown menu on the right. The dropdown menu is open, showing three options: 'Warning' (highlighted in teal), 'Warning', and 'Error'.

The "Priority" mask makes it possible to manage the priorities that can be assigned to an alarm. Crew provides two levels of priority, with a value corresponding to each one: "Warning" (200), "Error" (100). An alarm with priority of "100" is more important than one with priority "200".

# CREW Manual

“Warning” (200):



# CREW Manual

“Error” (100):



Alarm type:



Crew has two types of alarms:

- ISA alarm
- Simple event

The ISA alarm requires the intervention of the operator (reset and acknowledge) and begins an ISA sequence:



# CREW Manual

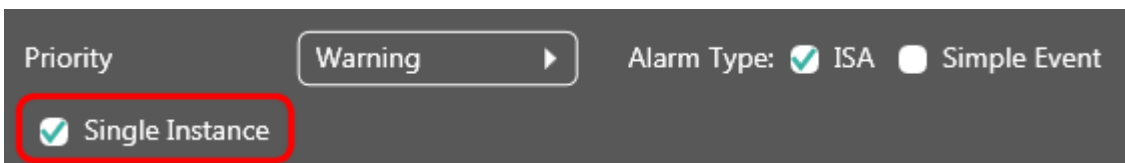
- “Access”: the alarm condition is activated on the device.
- “Acknowledge” (often also referred to as “ack”): an operator has taken over the management of the alarm.
- “Reset”: linked to the end of the alarm condition on the device.

The simple event is not an actual alarm, in fact it must be considered more of an informative message.

If the selected type of alarm is “ISA”, the “Single instance” option also appears in the editing mask.



Enable “Single instance” to make an alarm appear only once in the active alarms table, even with repeated occurrence of the same alarm, with failure to silence it.



# CREW Manual

If the “Single instance” option is not enabled, the alarm appears in the table of active alarms every time it is generated (in the example it is triggered three times in a row) and the reports only disappear when they have been silenced.

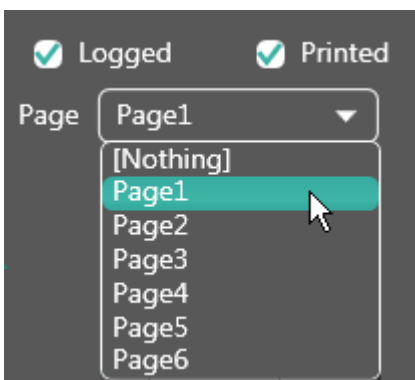


# CREW Manual

All alarms, both active and reset, are displayed in the alarm log.



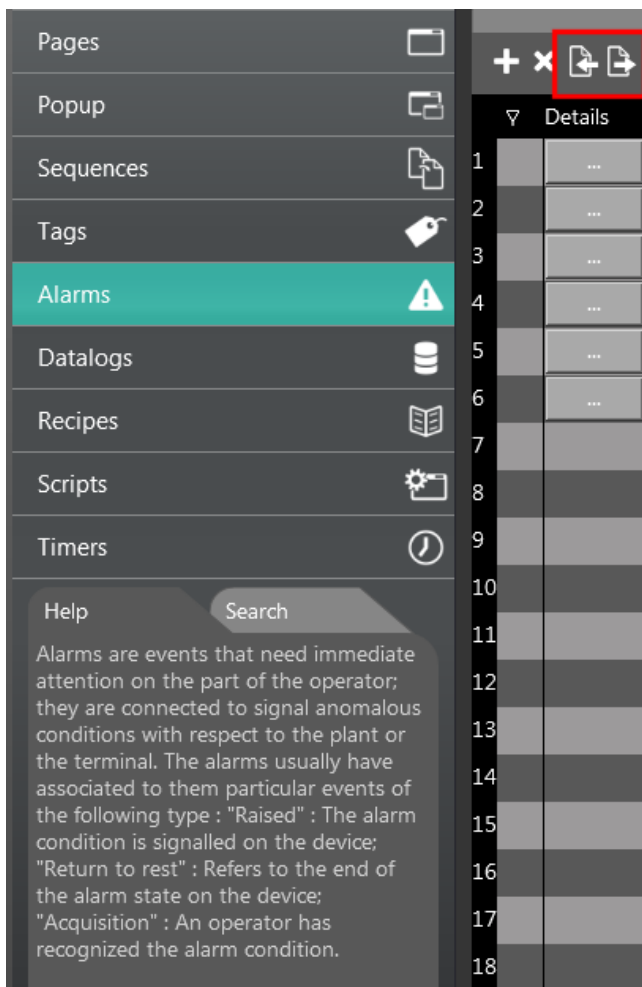
The “Recorded”, “Print” and “Page” options are respectively used to make the alarms appear/not appear in the log, and print/not print the single alarm, and establish what page to display.



# CREW Manual

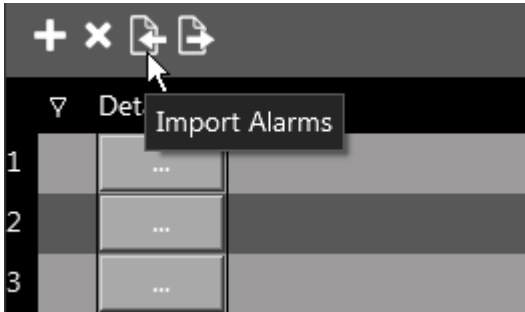
## Export - Import Alarms

Crew makes it possible to either export or import a series of previously created alarms from/to the project.

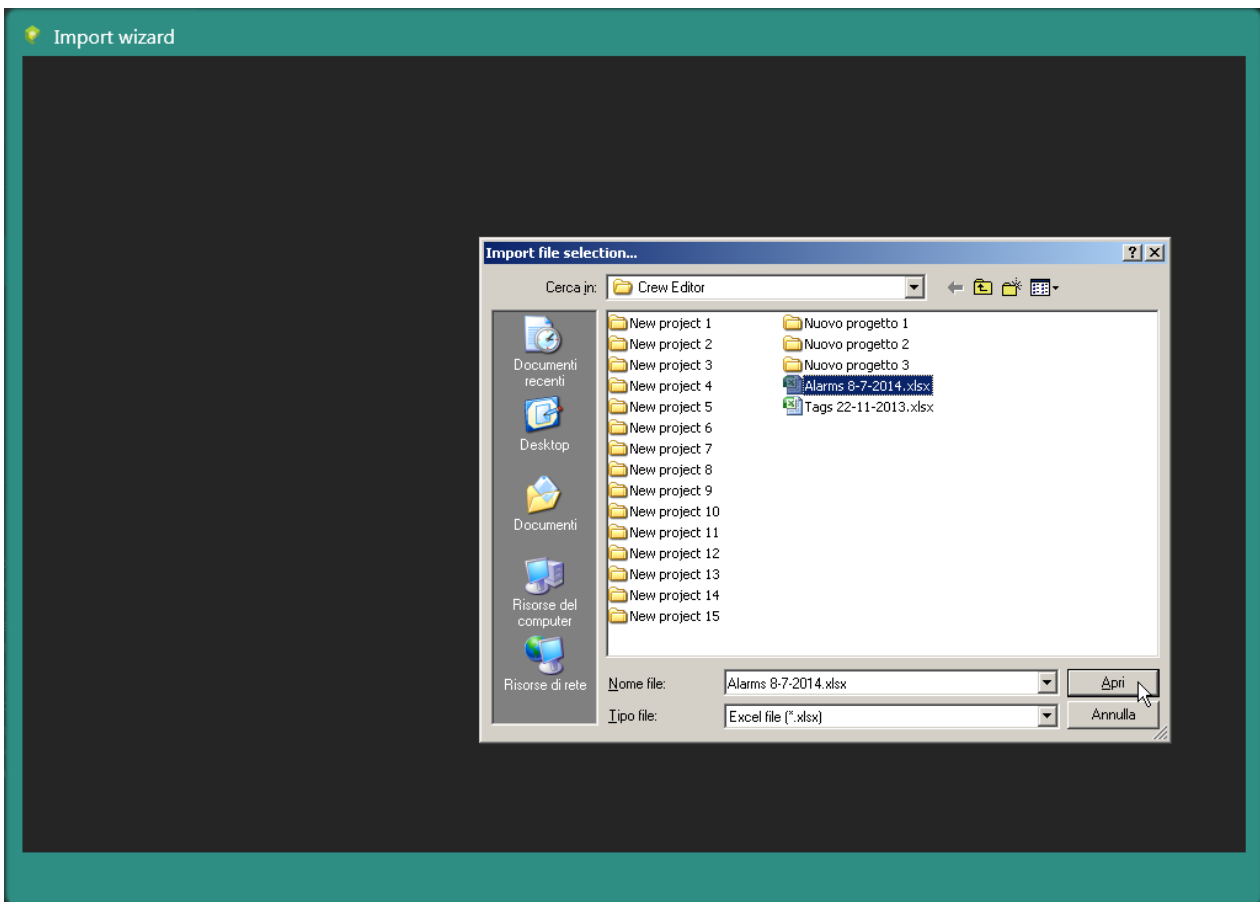


# CREW Manual

## Importing Alarms

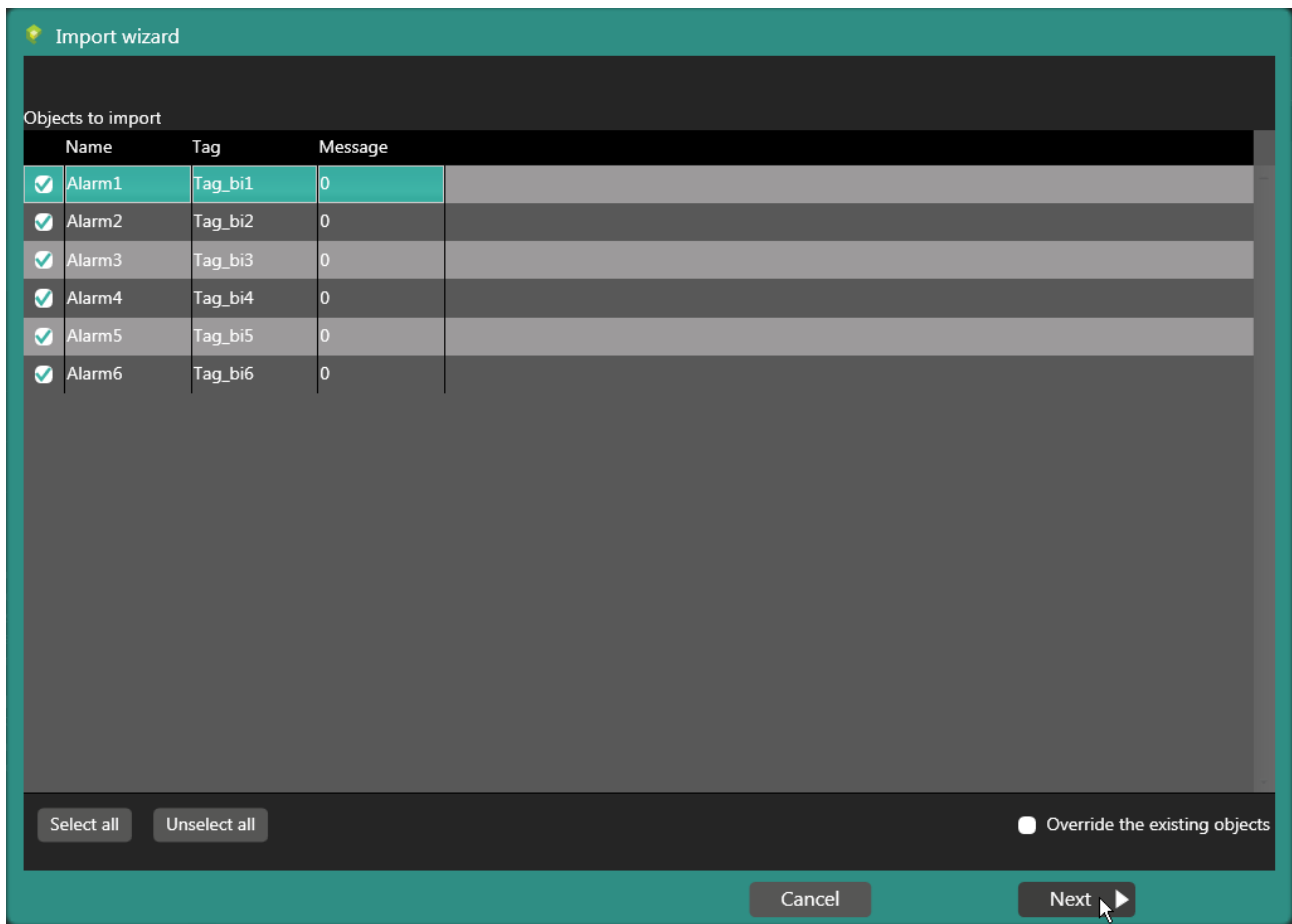


Click "Import Alarms" and select the file with the list of alarms to be imported into the project.



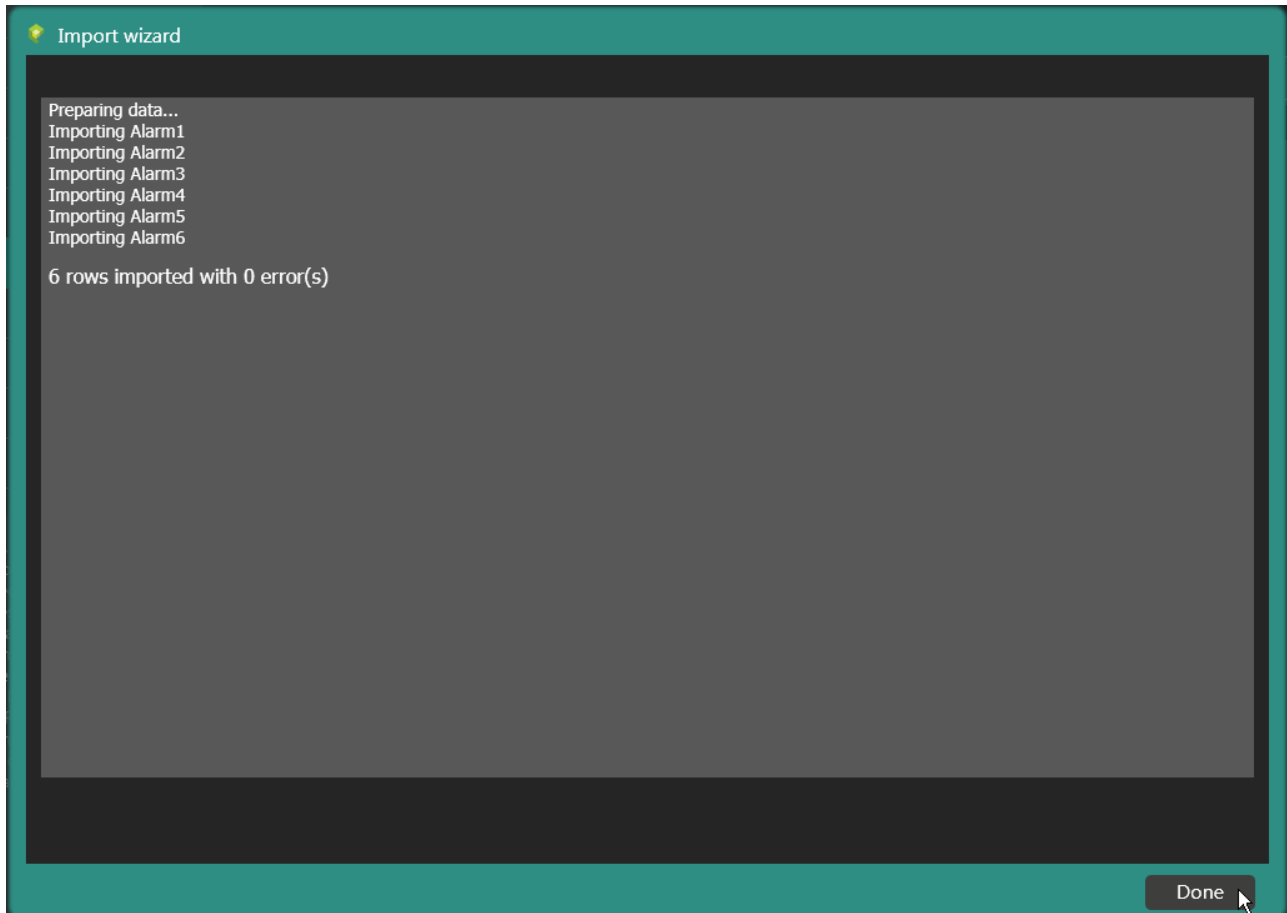
# CREW Manual

Decide whether to overwrite/not overwrite the existing objects in the project and click “Next” to import the alarms. It is possible to decide whether to import all or only those selected with the flag.



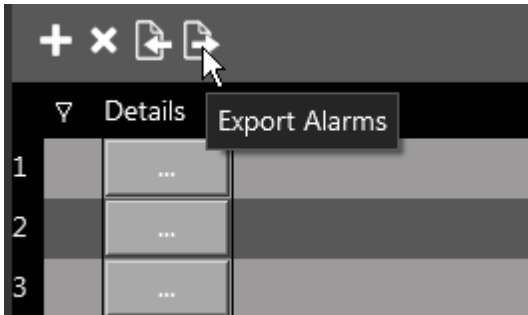
# CREW Manual

Click “Done” at the end of the importing procedure.

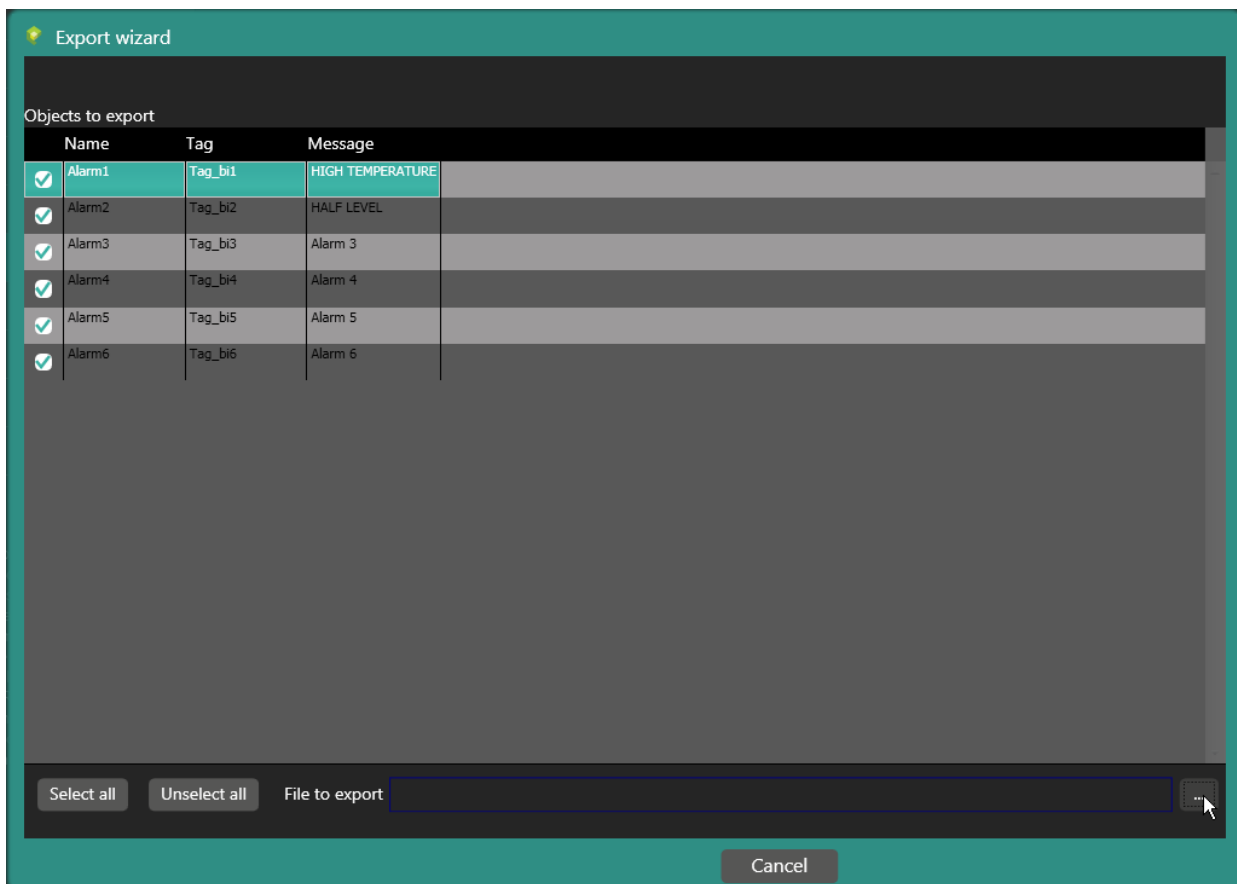


# CREW Manual

## Export Alarms



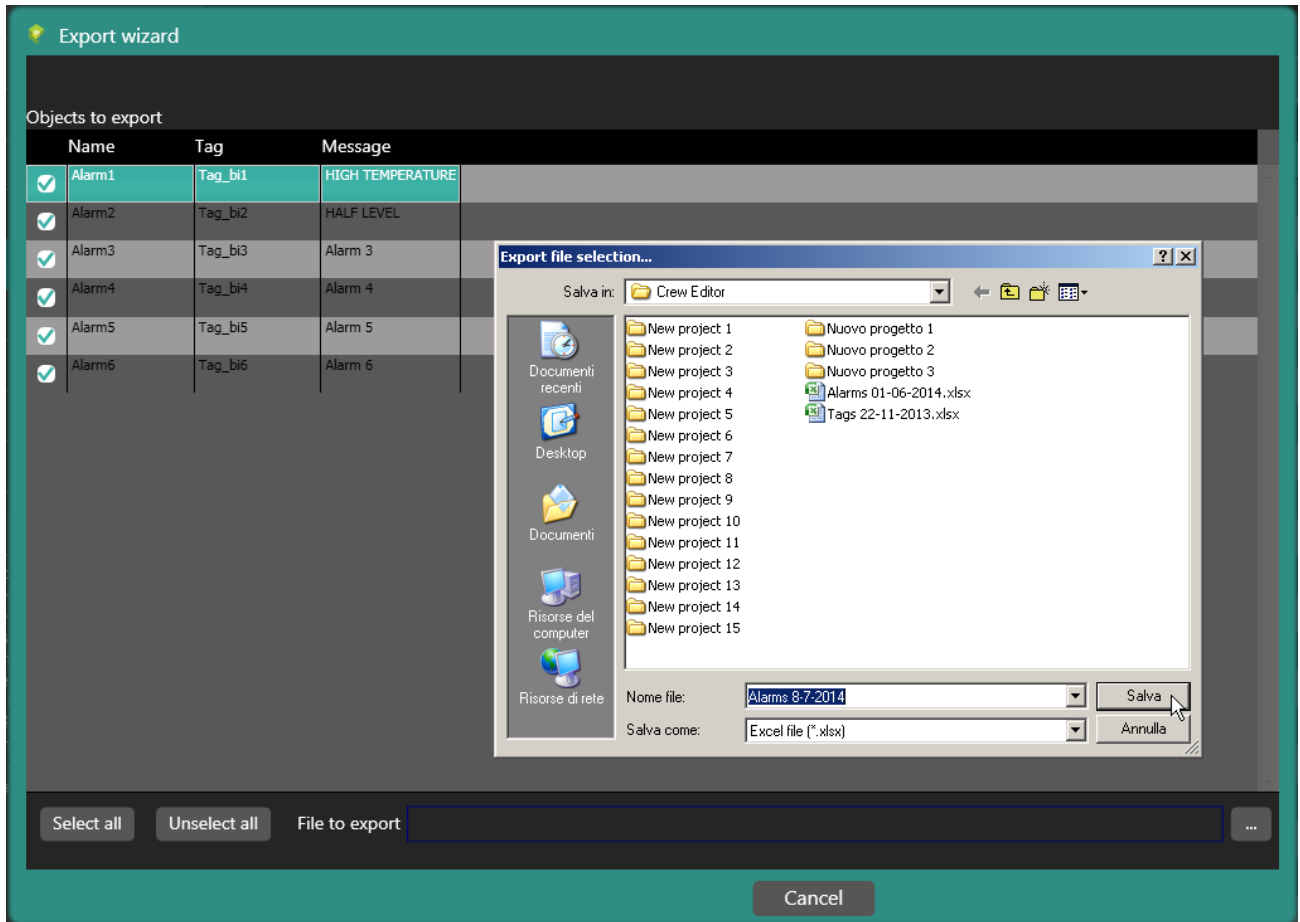
Click the “Export Alarms” key to export the list of alarms contained in the project to a file. It is possible to decide whether to export all or only those selected with the flag.





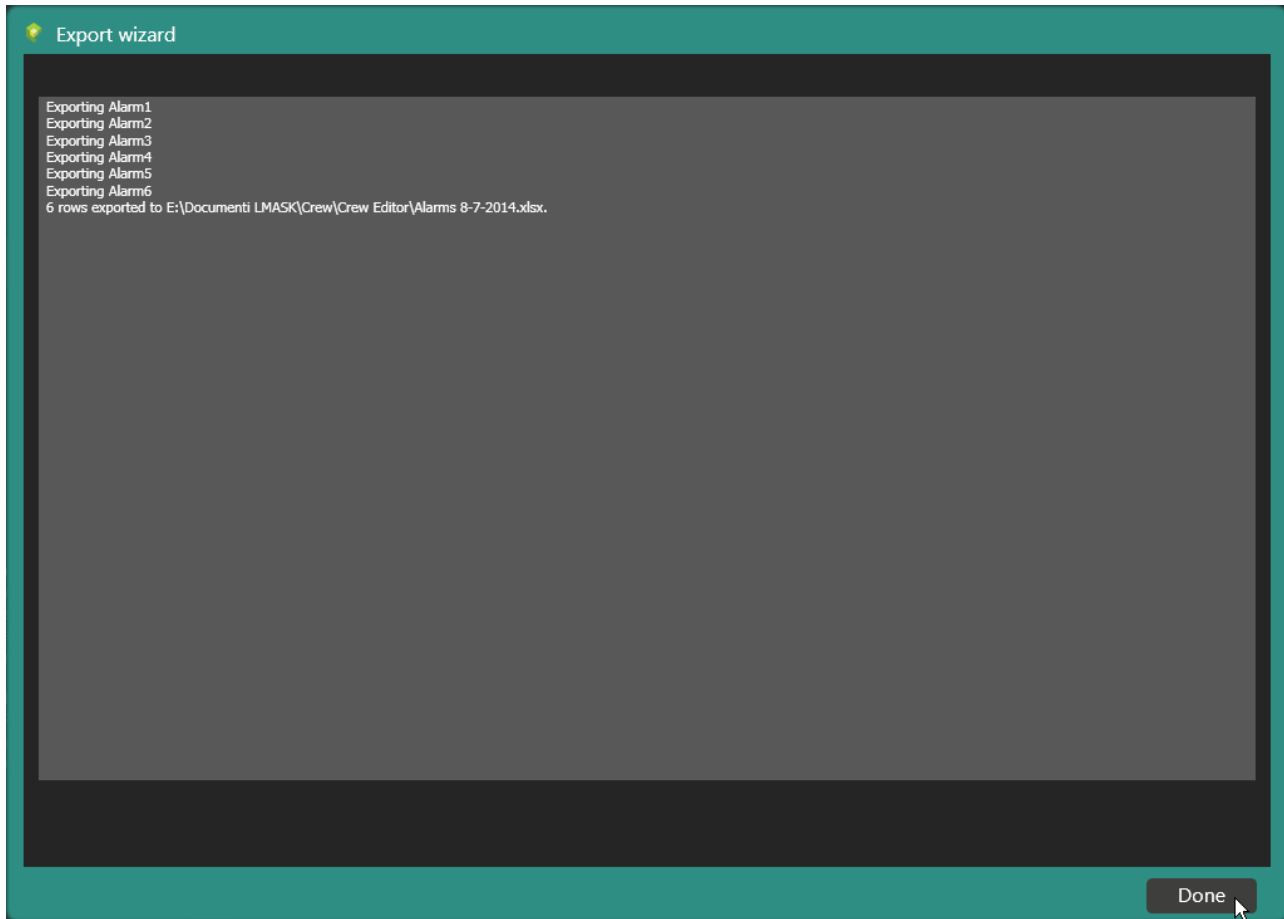
# CREW Manual

Click “Browse” to choose the target folder.




# CREW Manual

Click “Save” and “Next” to start the exporting procedure. At the end click “Done”.



# CREW Manual

## Properties - Alarms -



From the “Properties” mask it is possible to define the following parameters:

- The maximum number of recordings that can be made at the same time.
- The maximum capacity of the buffer of the active alarms log.
- The alarm level (expressed in percentage) of the capacity of the alarm log buffer.
- The alarm log recording operating mode (it is possible to choose whether to discard the old alarms or ignore the new).



- Assign a name to the log buffer file.
- Export the alarm log file.

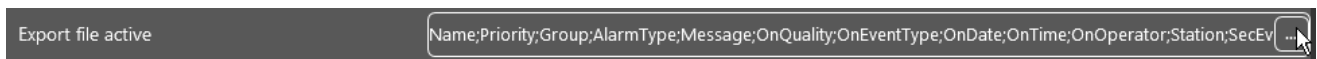


# CREW Manual

It is possible to define the structure of the file to be exported by selecting or unselecting the various options.

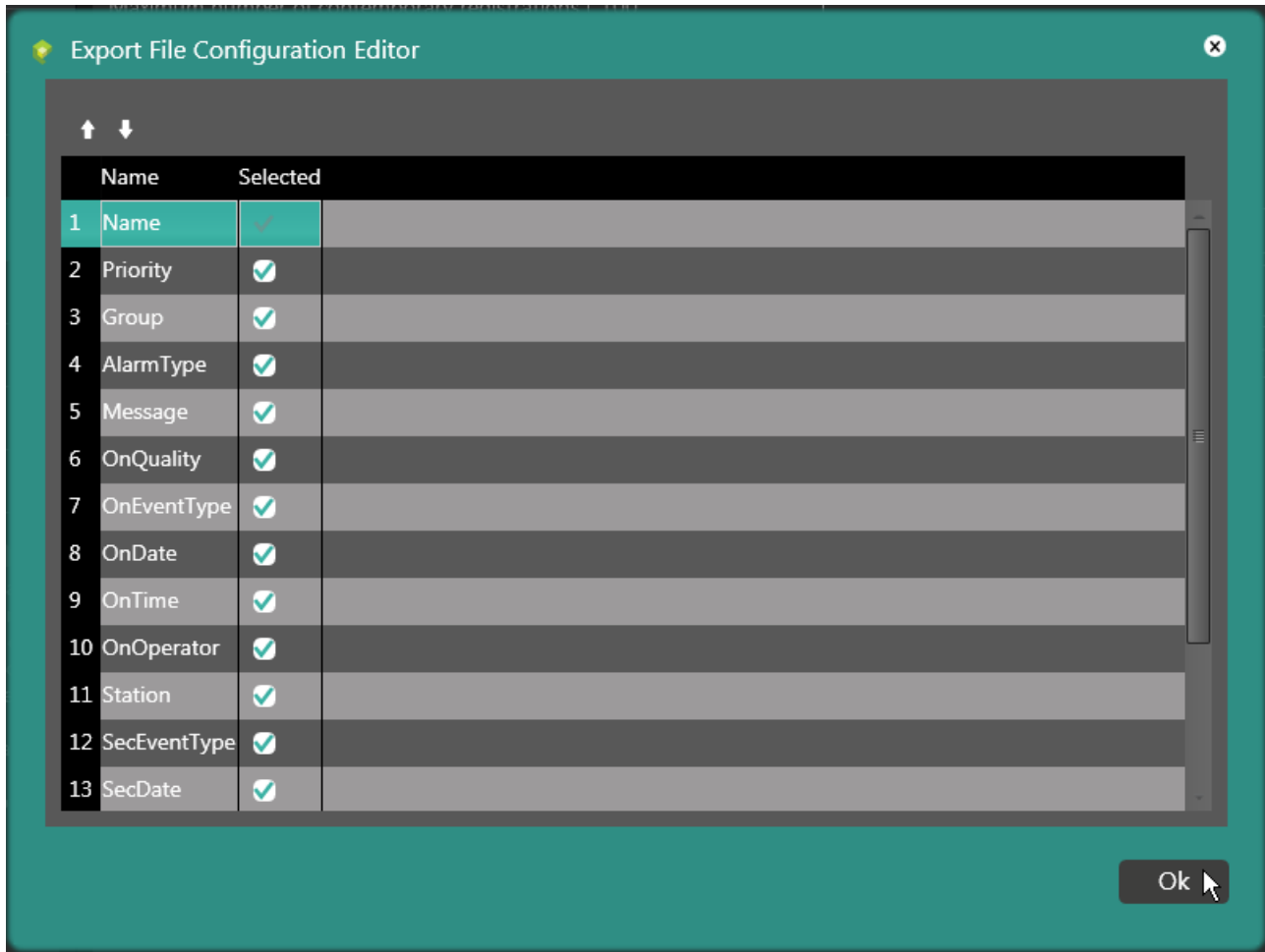


- Export the active alarms file.



# CREW Manual

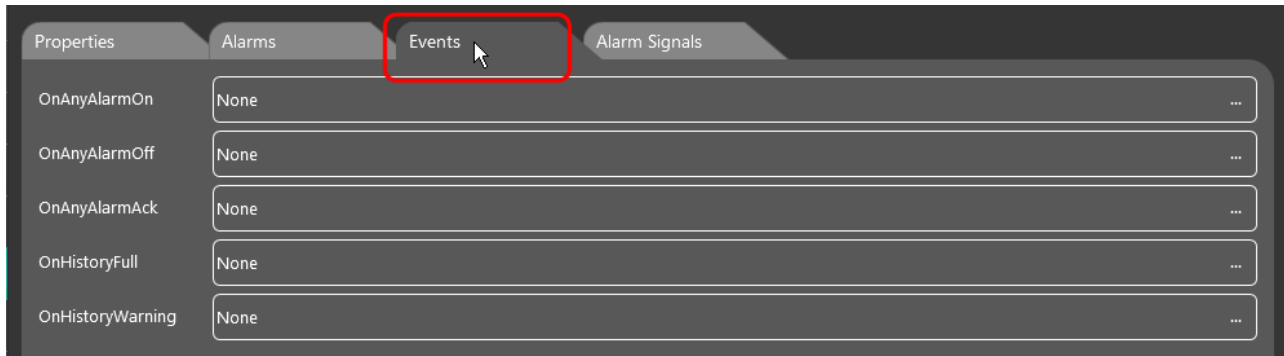
It is possible to define the structure of the file to be exported by selecting or unselecting the various options.



# CREW Manual

## Events - Alarms -

The third window of the Alarms Editor is the "Events" option.



An event (function or script) can be linked to each previously created Alarm. Click the "Browse" key on the right.

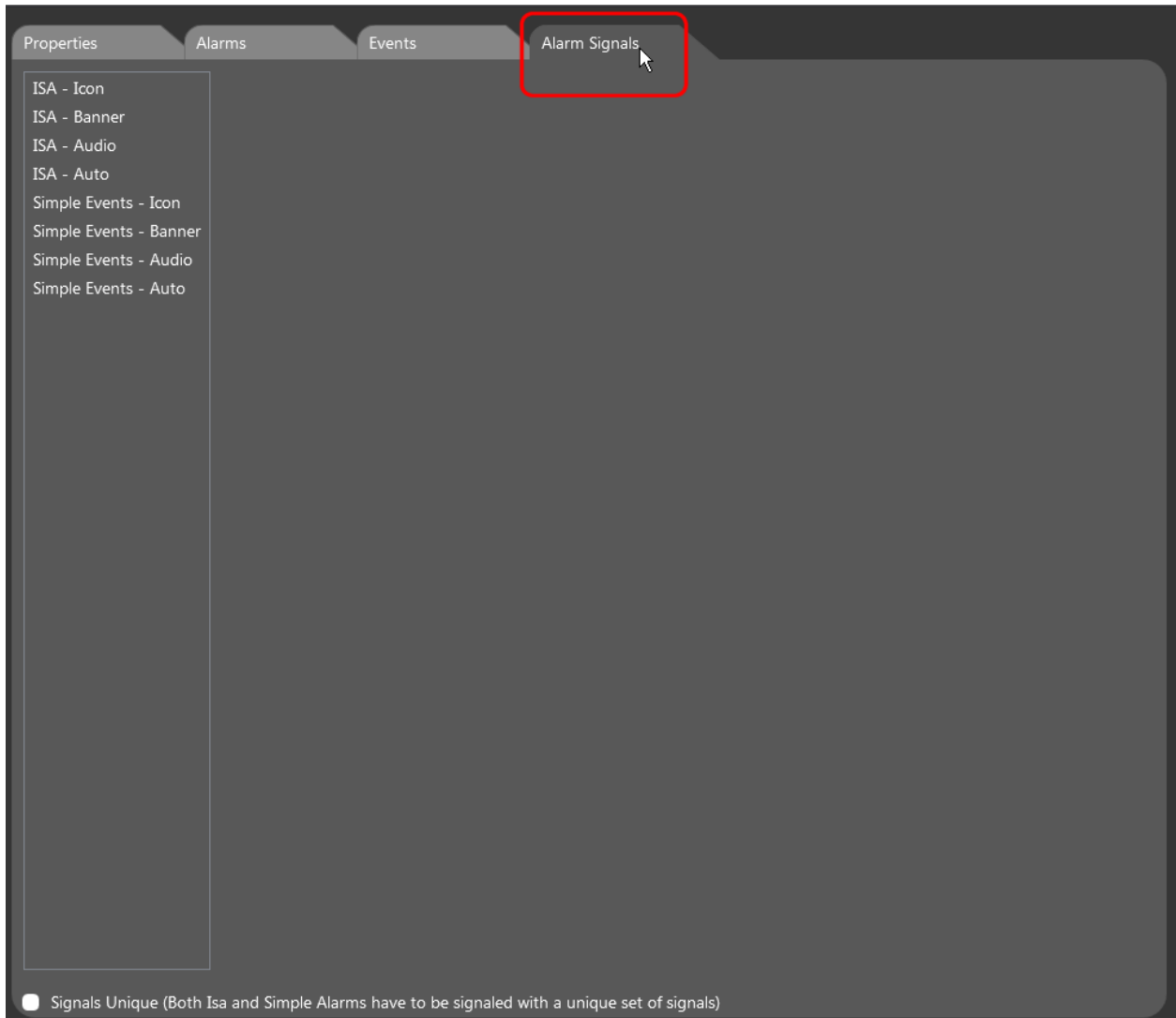


The event is enabled in the different conditions that the Alarm is in, which are described in the "Alarm Events" table.

# CREW Manual

## Alarm Signals

The fourth window of the Alarms Editor is the "Alarm Signals" option.



# CREW Manual

From here it is possible to assign and manage signals to two types of alarms (banners, audio signals, icons):

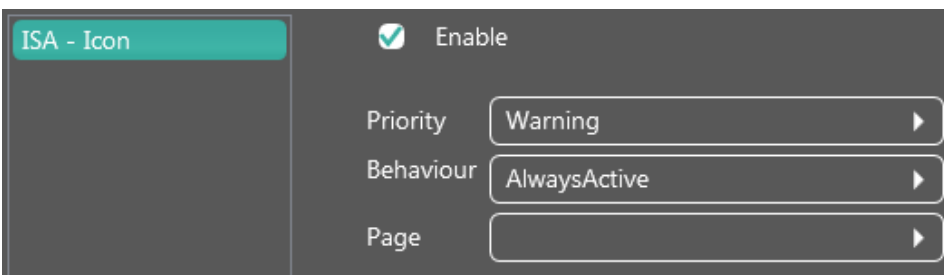
- ISA alarm
- Simple event

The ISA alarm requires the intervention of the operator (reset and acknowledge) and begins an ISA sequence:

- “Access”: the alarm condition is activated on the device.
- “Acknowledge” (often also referred to as “ack”): an operator has taken over the management of the alarm.
- “Reset”: linked to the end of the alarm condition on the device.

The simple event is not an actual alarm, in fact it must be considered more of an informative message.

## ISA alarm icon



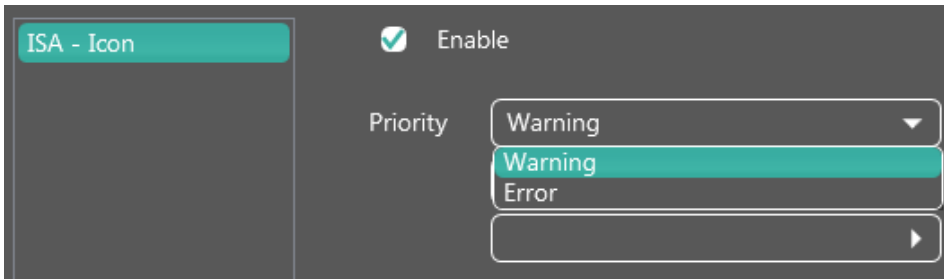
The screenshot shows a configuration window for an ISA alarm icon. The window has a dark grey background. On the left, there is a teal header bar with the text "ISA - Icon". To the right of the header bar, there is a checked checkbox labeled "Enable". Below this, there are three rows of configuration options, each with a label on the left and a dropdown menu on the right:

Priority	Warning
Behaviour	AlwaysActive
Page	



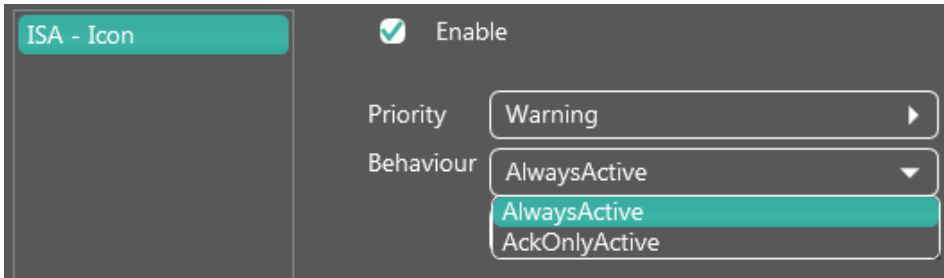
# CREW Manual

To assign the icon choose from the two levels of Priority: "Warning" or "Error".



By choosing "Warning" the icon is displayed when there is at least one alarm with "Warning" priority. By choosing "Error" the icon is displayed when there is at least one alarm with "Error" priority.

To assign the icon, choose from the two levels of Behaviour: "AlwaysActive" or "AckOnlyActive".

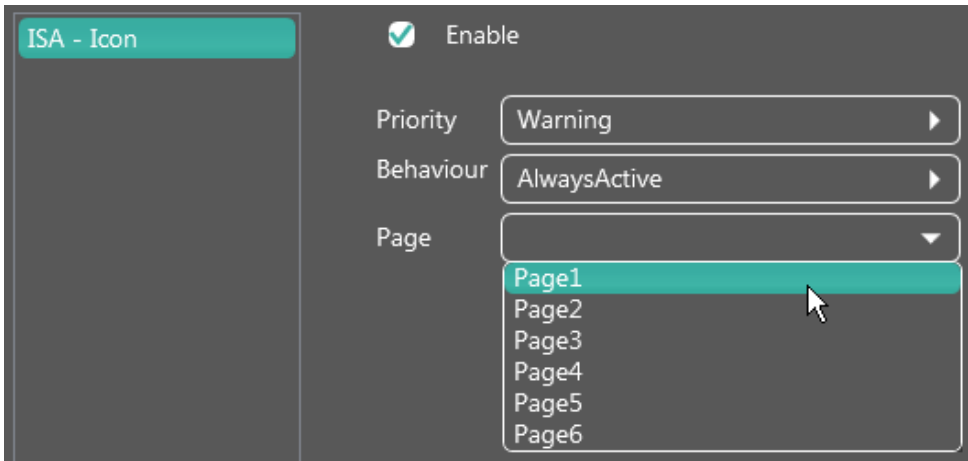


By choosing "AlwaysActive" the icon is displayed when there is an active alarm (regardless of whether it is acknowledged or not).

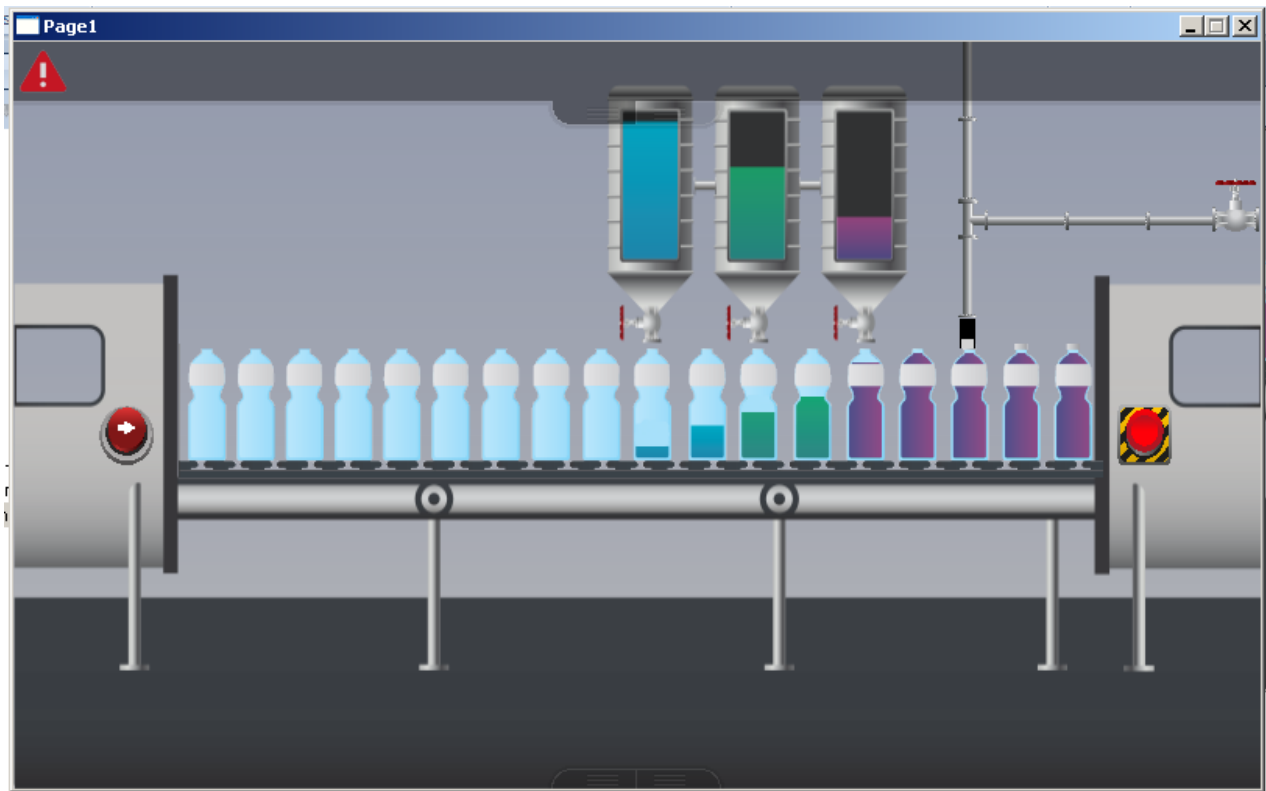
By choosing "AckOnlyActive" the icon is displayed when there is an active alarm acknowledged by the operator.

# CREW Manual

Plus, it is possible to establish what page to view after pressing the alarm icon.

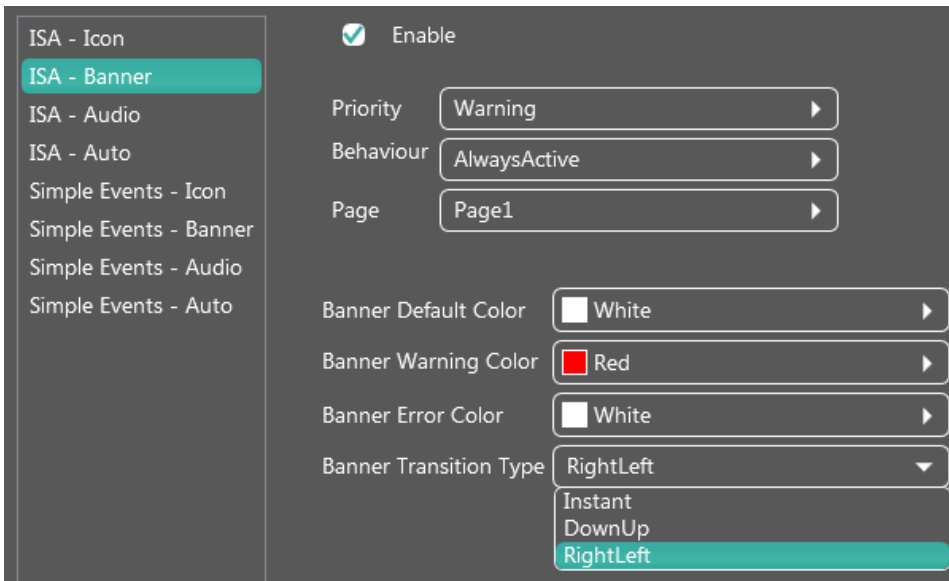


# CREW Manual

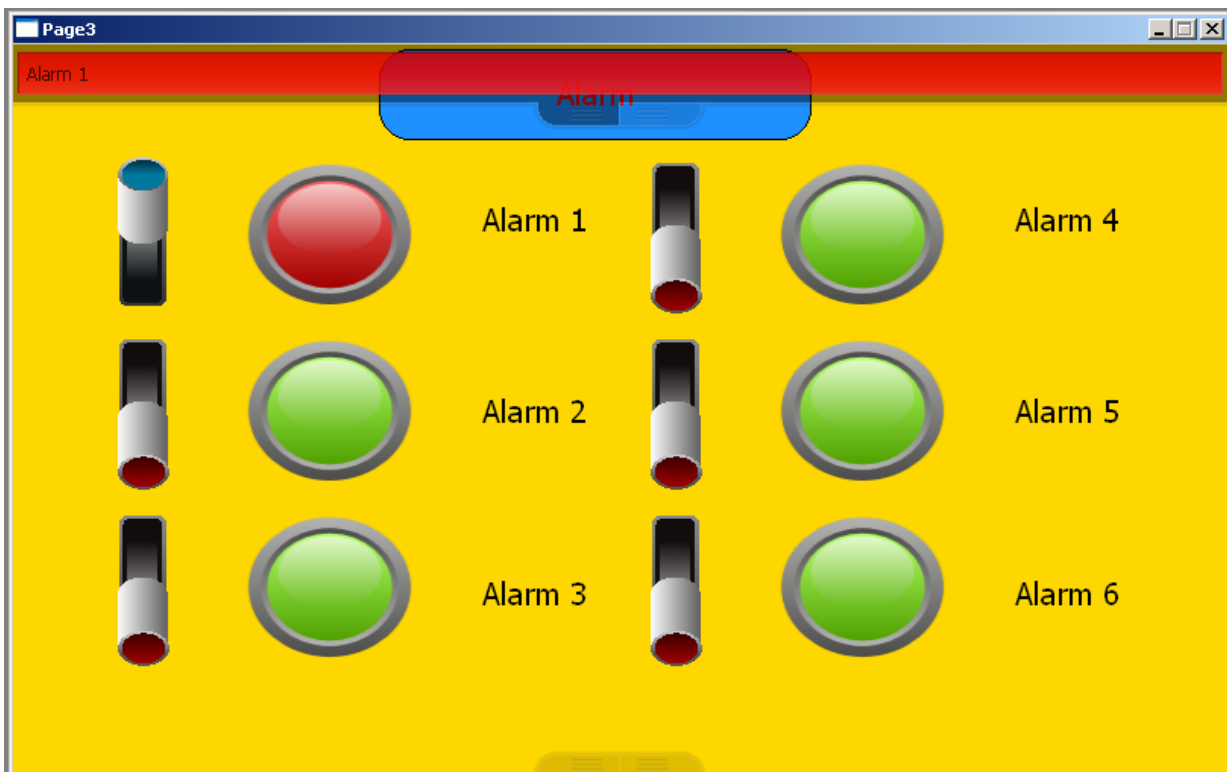


The above in reference to the ISA alarms icon, also applies to the “Banner”, with the only difference being that for the Banner it is possible to set the colours and the view (appearing instantaneously or scrolling from top to bottom or from right to left).

# CREW Manual



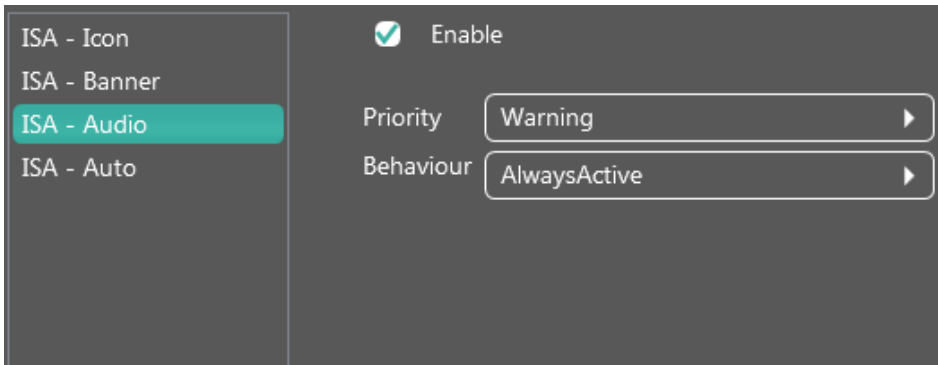
The image provides an example of Banner viewing.



# CREW Manual

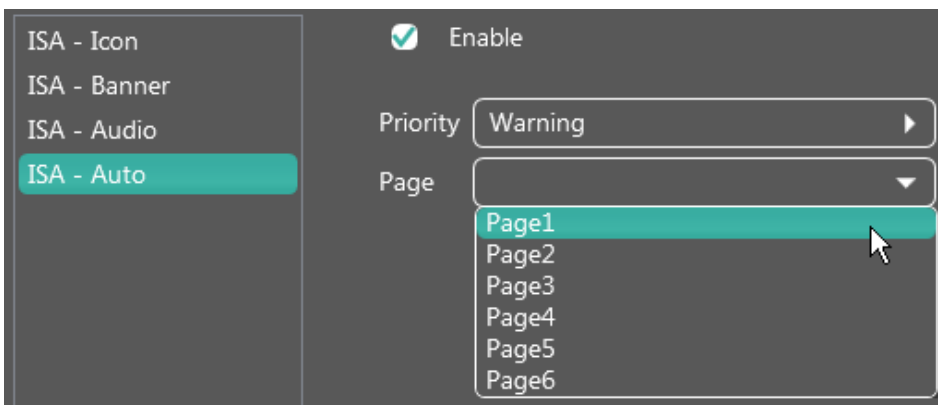
## ISA - Audio -

To issue an audio signal, associated to the alarm, from the terminal, select the levels of priority and behaviour as shown in the image.

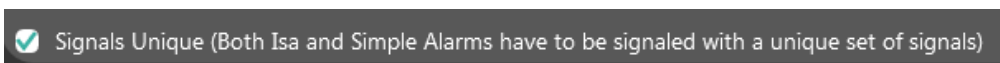


## ISA - Auto -

To automatically view a page when an alarm appears, choose from the available pages and select the corresponding level of priority.



Select the “Unique Signal” option (at the bottom of the page) to use a single icon for both ISA alarms and simple events.



# CREW Manual

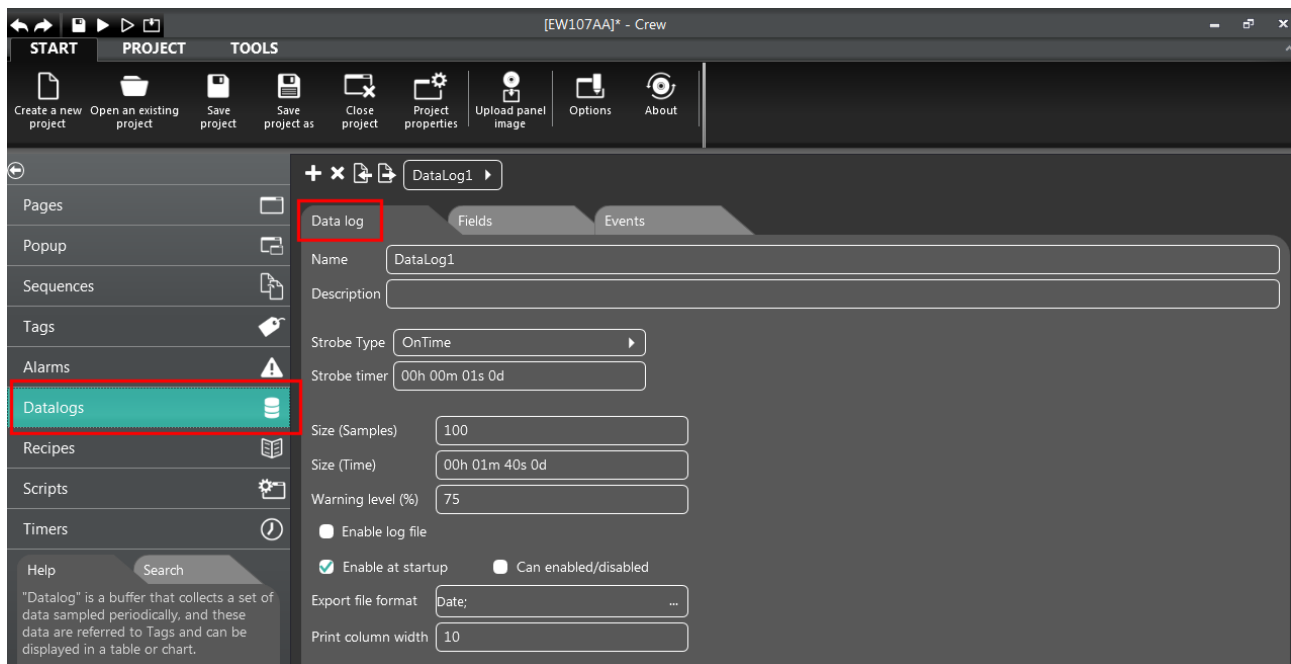
## Datalogs

The Runtime system provides support for the acquisition and collection of number values, and for their presentation in graphic form as “Trend curves” or data “Buffers” (DataLogs).

The “Trend” field displays the contents of the buffer (see sections: "[Trend](#)", "[Trend Property](#)", "[Trend XY](#)", "[Trend XY Property](#)", "[Touch Trend](#)", "[Touch Trend Property](#)").

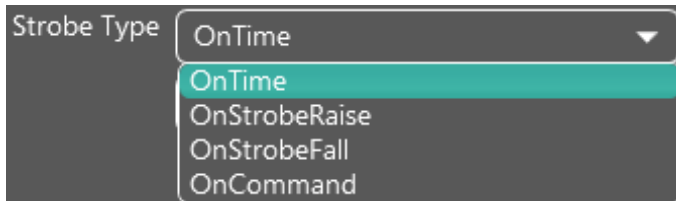
The “DataLog” is a buffer where a set of periodically sampled data is collected. This data refers to tags and can be displayed in a table or graph (see sections: "[Data Log](#)" and "[Data Log Property](#)").

## Data log



# CREW Manual

- Strobe: value sample taking mode where it is possible to choose between:



- Time-controlled: sampling is conducted at regular intervals.
- Raised strobe: sampling is carried out when the reference tag changes its value from FALSE to TRUE.
- Lowered strobe: sampling is carried out when the reference tag changes its value from TRUE to FALSE.
- Command-controlled: sampling is carried out on command from script or function (see section "[Functions relative to Datalogs - Samples -](#)").



NOTE: If sampling is “RaisedStrobe” or “LoweredStrobe” it is necessary to provide a “Boolean” tag.



NOTE: The system manages the buffer in FIFO mode (first in first out).

- Strobe Timer: time interval for each data acquisition.

- Size (Samples): number of samples.

- Size (Time): total sampling period. For example, 1 sample taking per second for 100 samples = 1 minute and 40 seconds of total sampling time.

- Attention level (%): warning value (expressed in percentage) above which the user needs to be warned that the buffer is filling. The percentage defines the sample taking level above which it is possible to associate an event. For example, a warning message through a popup page associated to the 75th sample of the planned 100.

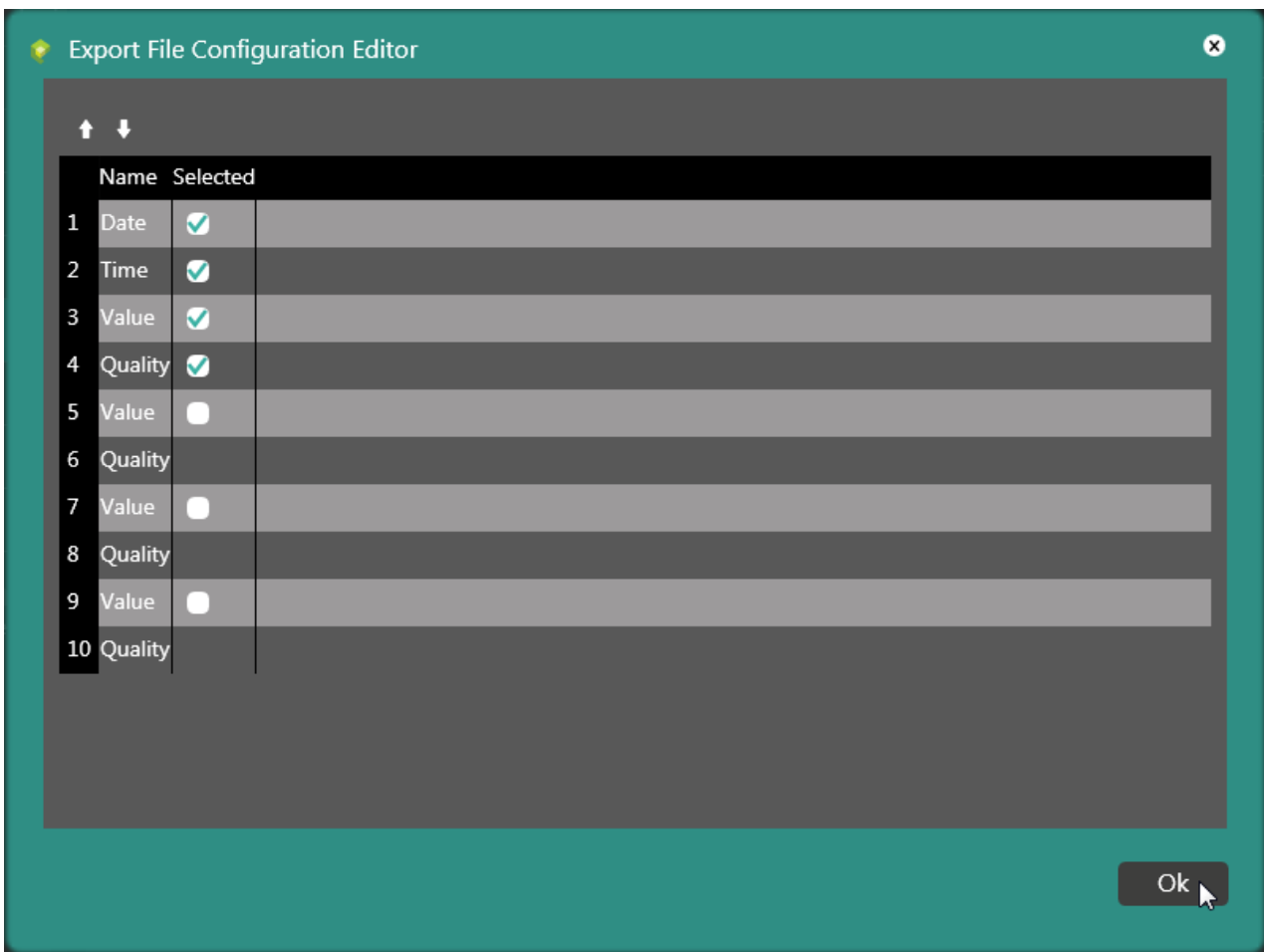
- Enable log file: this states whether the “Buffer” elements need to be saved on files so as to be kept after the terminal is switched off (otherwise they are saved on a volatile memory). In this way it is possible to perform automatic saving (every three minutes by default) of a file that saves data acquisition.

# CREW Manual



NOTE : The “log” file is not the data export file, but it is a file that can be consulted to avoid losing unsaved and unexported data (for information relative to the Data log data export file, see "[Data Log Configuration - EW Terminal side](#)" section).

- Enable at startup: sample taking starts when the project starts up.
- Enabled/Disabled: the possibility of managing sample taking start and end. For example, through a “Script” or a function (see "[Functions relative to Datalogs - Samples -](#)").
- Export format: to define the fields to be exported to the export file.





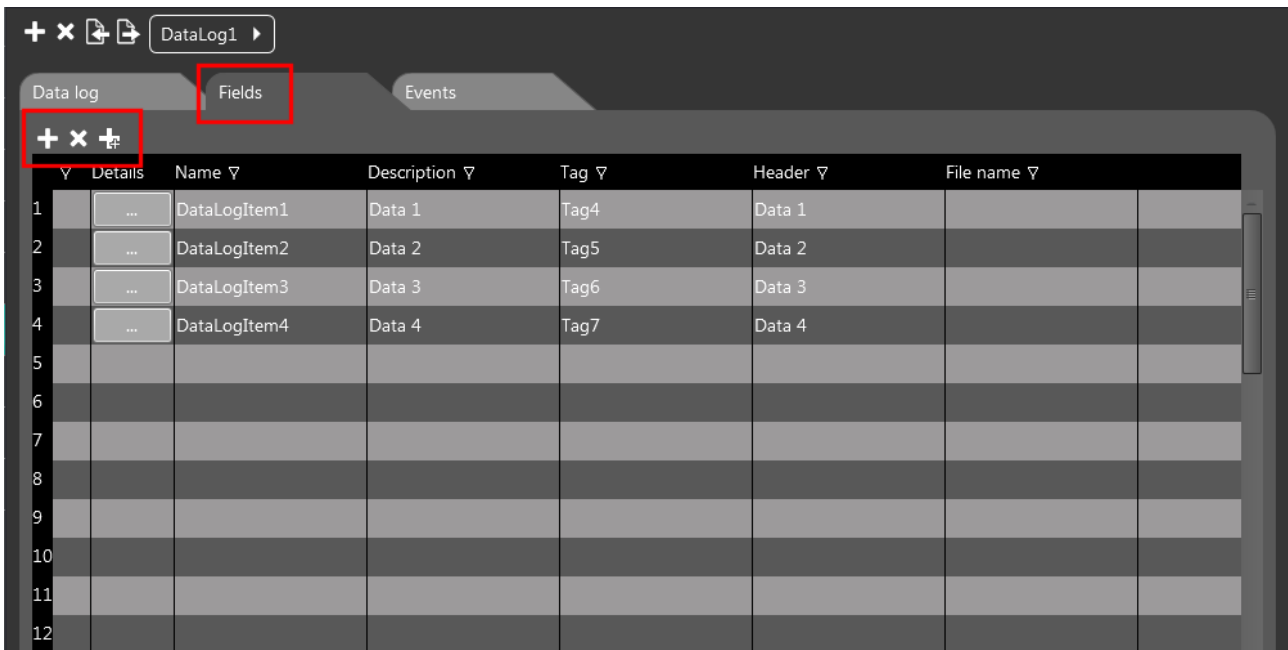
# CREW Manual

- Print column width: the possibility of defining the width of the columns to be set when printing the saved export file.

It is possible to see a Data Log configuration example in the "[Data Log Configuration - Crew Side](#)" section.

For a Data Log export file creation example, on the other hand, see "[Data Log Configuration - EW Terminal Side](#)" section.

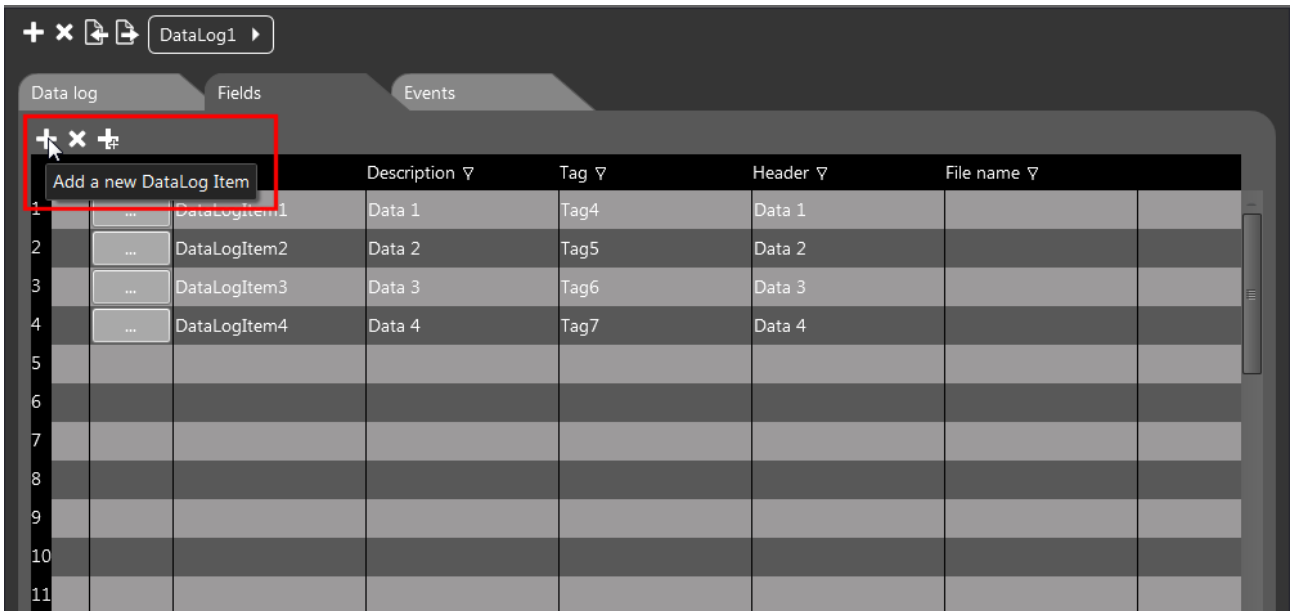
## Fields



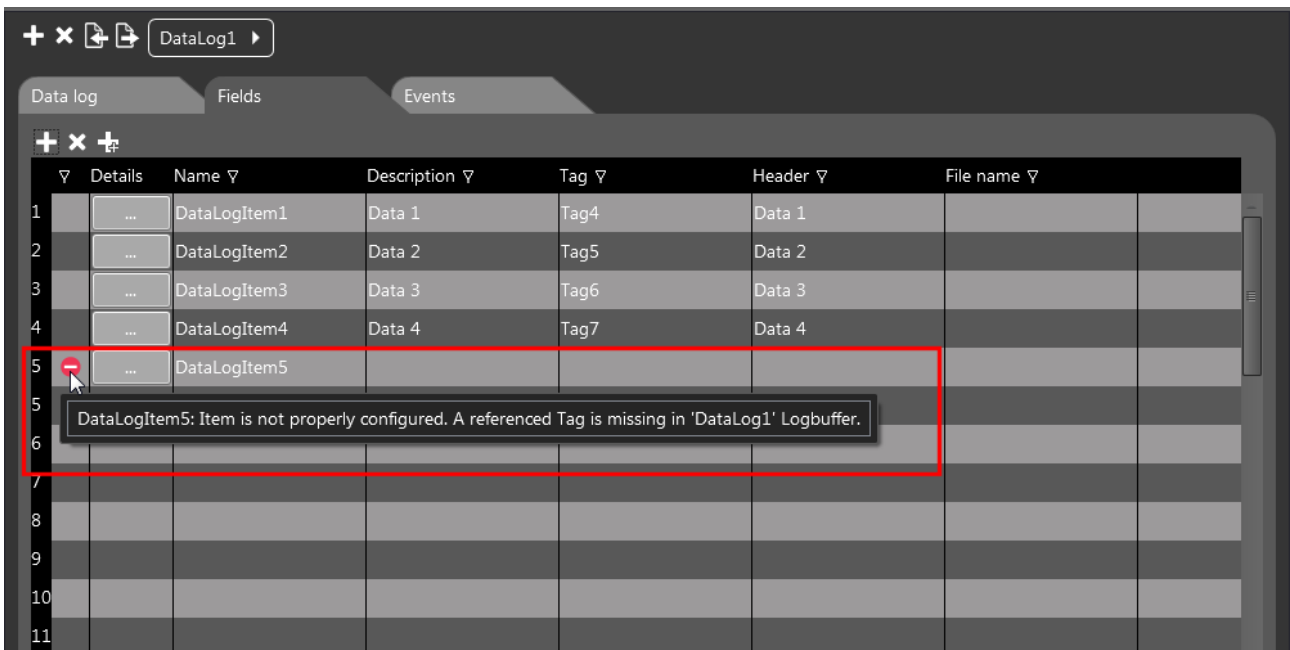
	Details	Name ▾	Description ▾	Tag ▾	Header ▾	File name ▾
1	...	DataLogItem1	Data 1	Tag4	Data 1	
2	...	DataLogItem2	Data 2	Tag5	Data 2	
3	...	DataLogItem3	Data 3	Tag6	Data 3	
4	...	DataLogItem4	Data 4	Tag7	Data 4	
5						
6						
7						
8						
9						
10						
11						
12						

# CREW Manual

In the “Fields” mask it is possible to associate new objects to the Data Log.

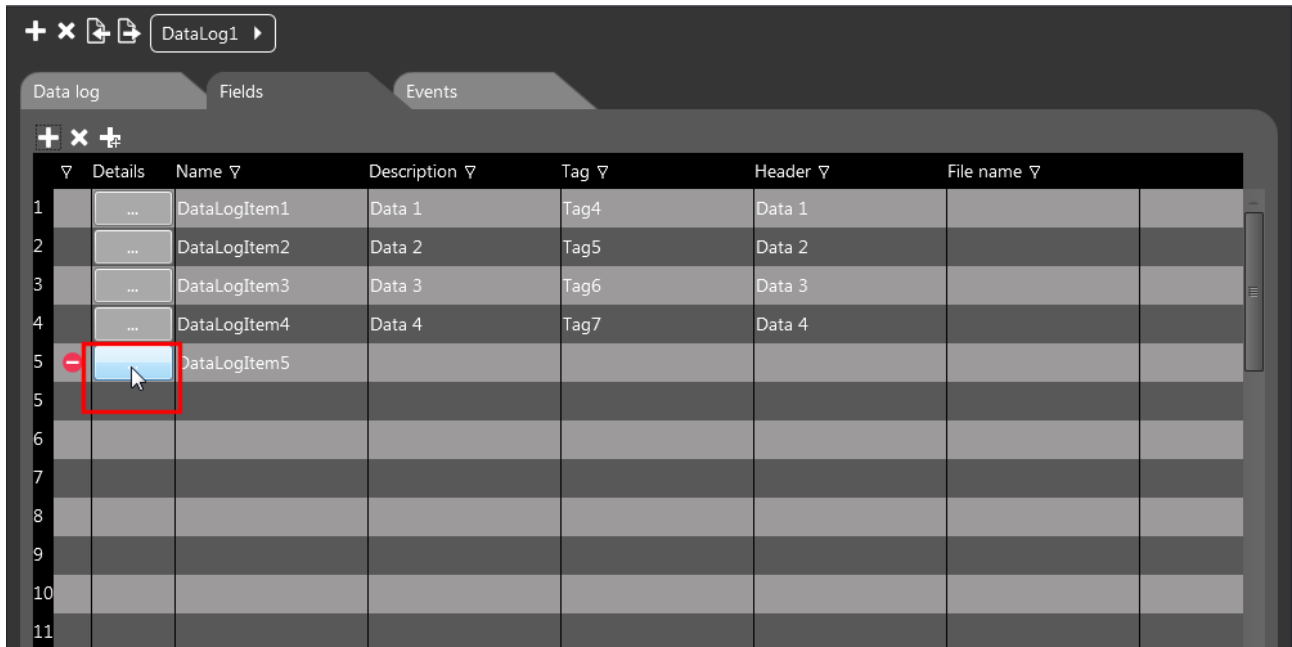


The object needs to be configured correctly choosing the reference tag.



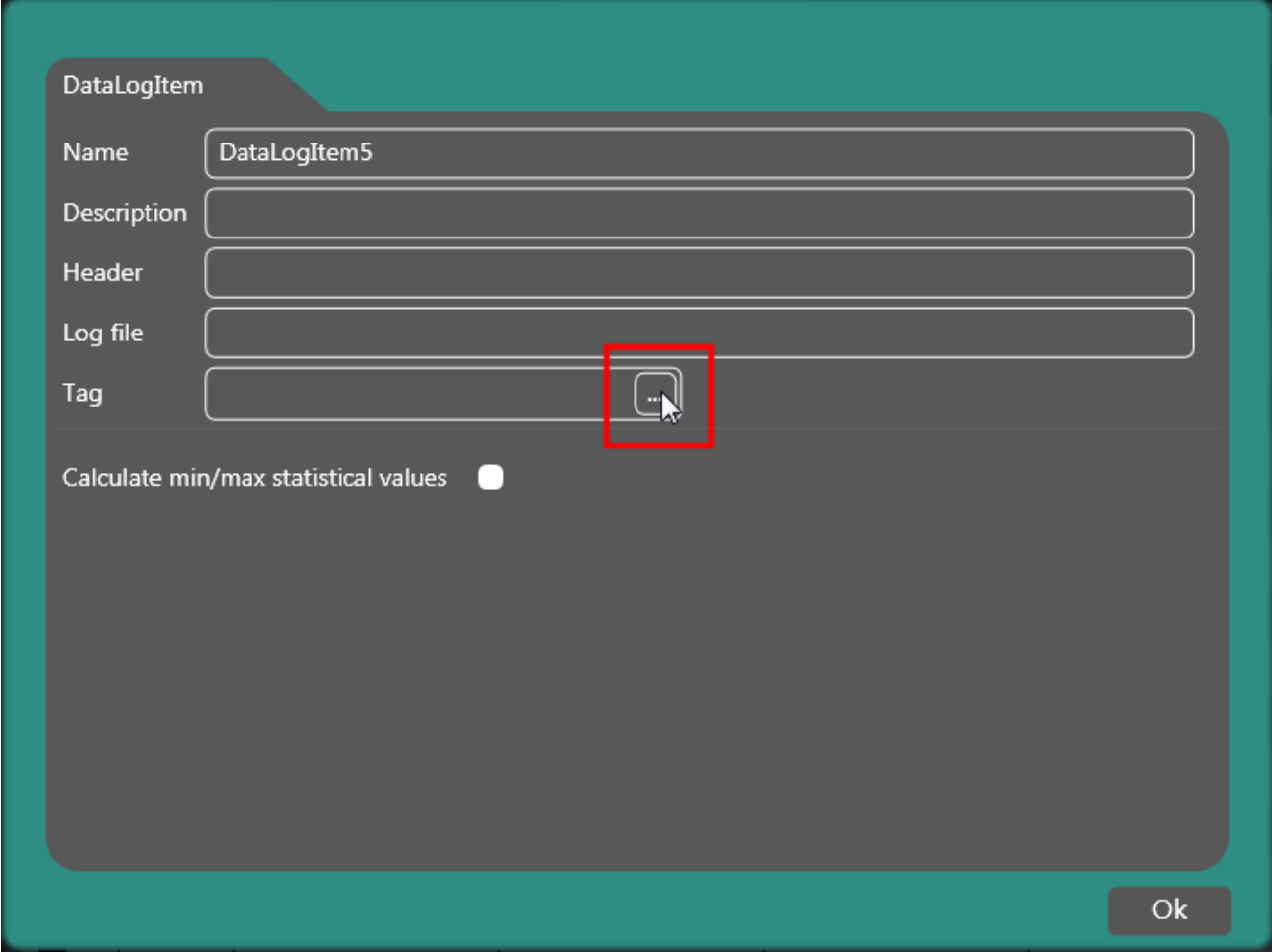
# CREW Manual

Click the “Details” key.



# CREW Manual

Click “Browse” in the “Tag” field.

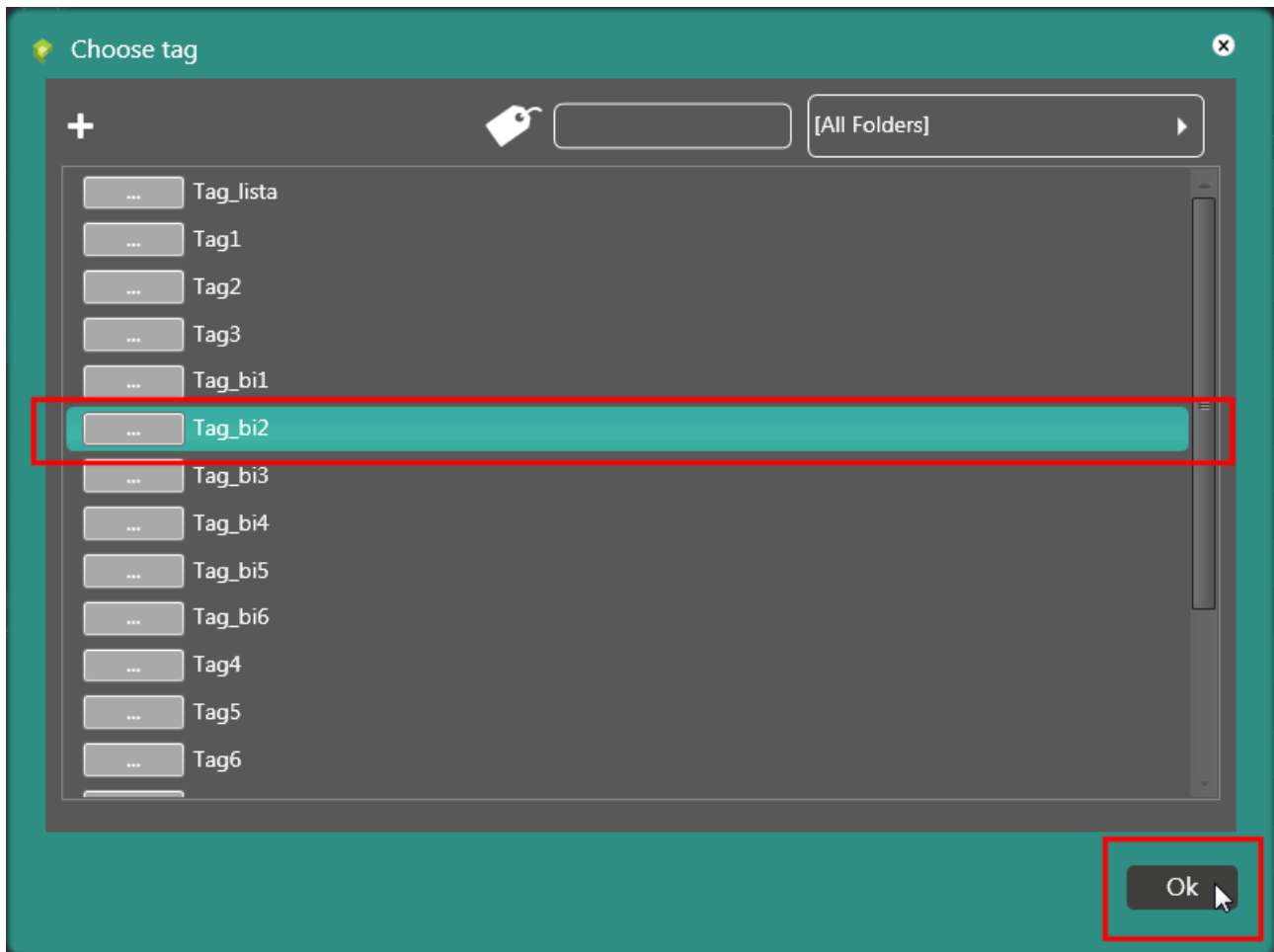


The screenshot shows a configuration window titled "DataLogItem" with the following fields and controls:

- Name: DataLogItem5
- Description: (empty)
- Header: (empty)
- Log file: (empty)
- Tag: (empty) with a "Browse" button (represented by a folder icon) highlighted by a red box.
- Calculate min/max statistical values:
- Ok button: (bottom right)

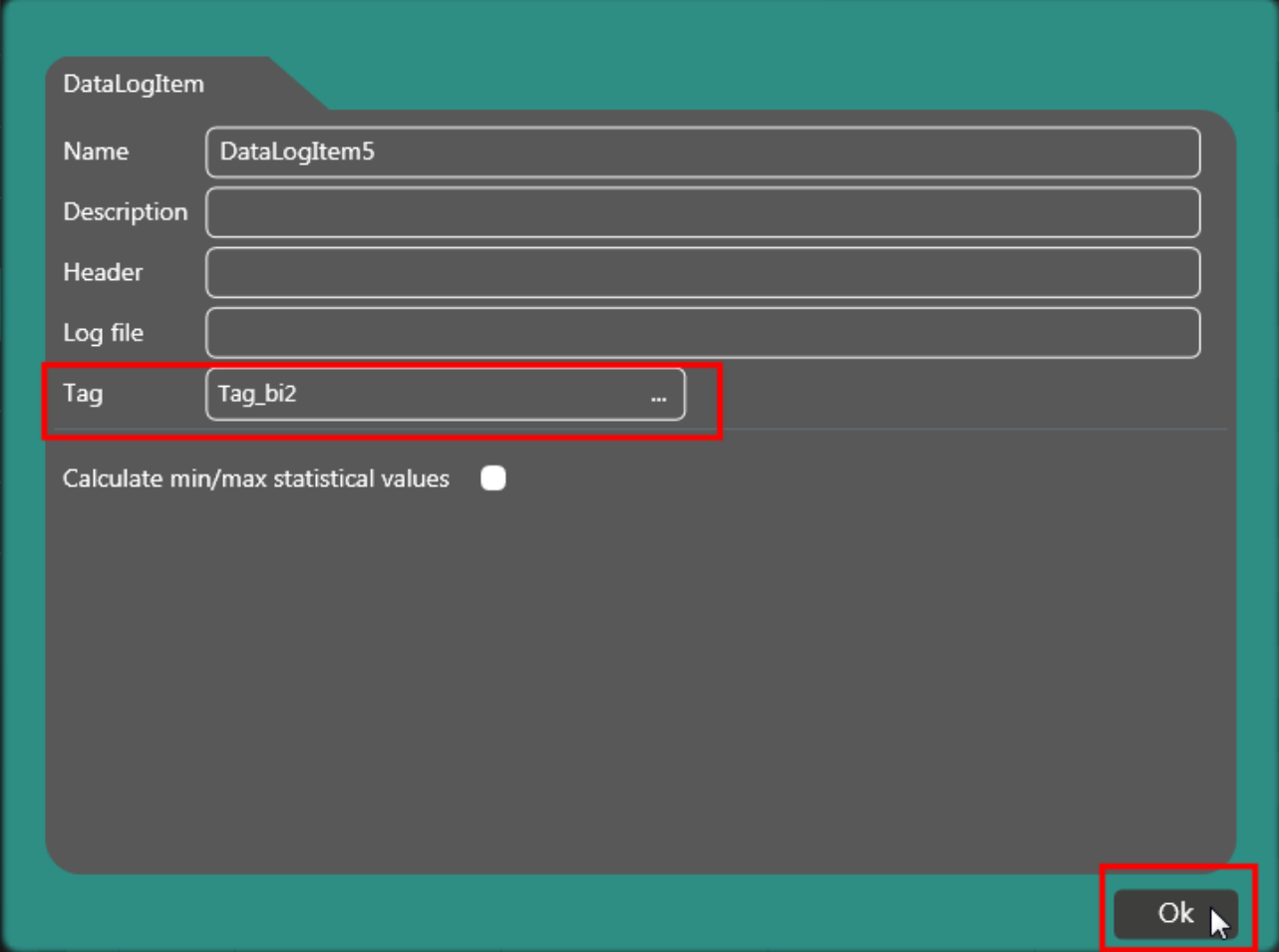
# CREW Manual

Select the reference tag and click “Ok”.



# CREW Manual

Confirm with "OK".



DataLogItem

Name DataLogItem5

Description

Header

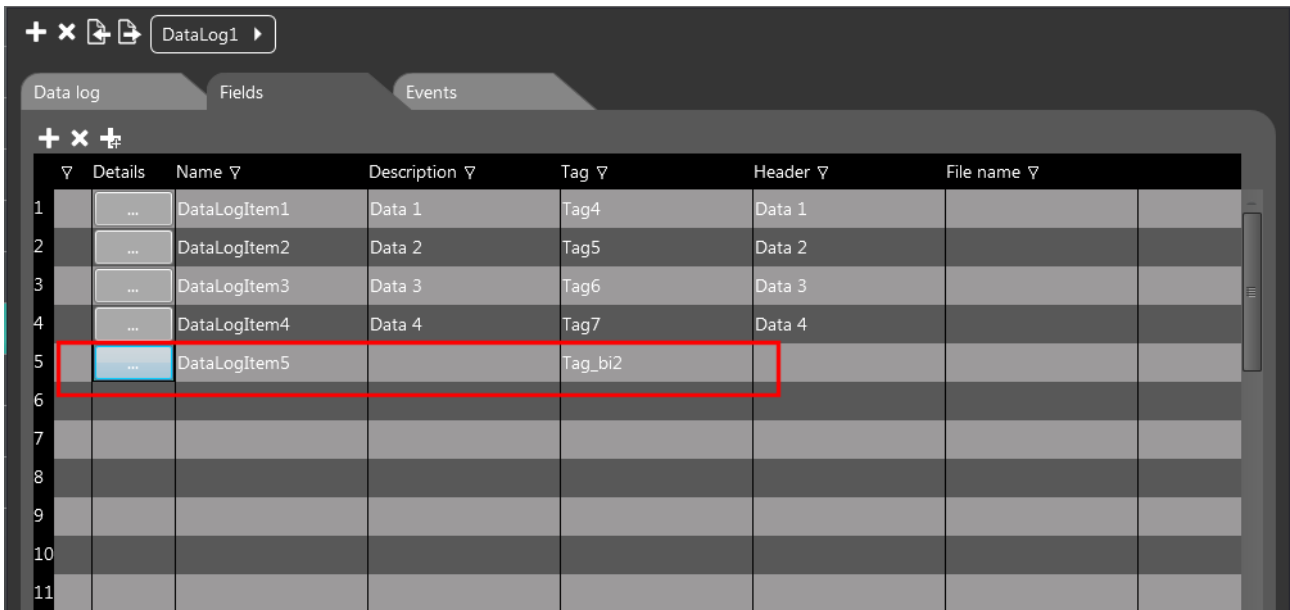
Log file

Tag Tag\_bi2 ...

Calculate min/max statistical values

Ok

# CREW Manual

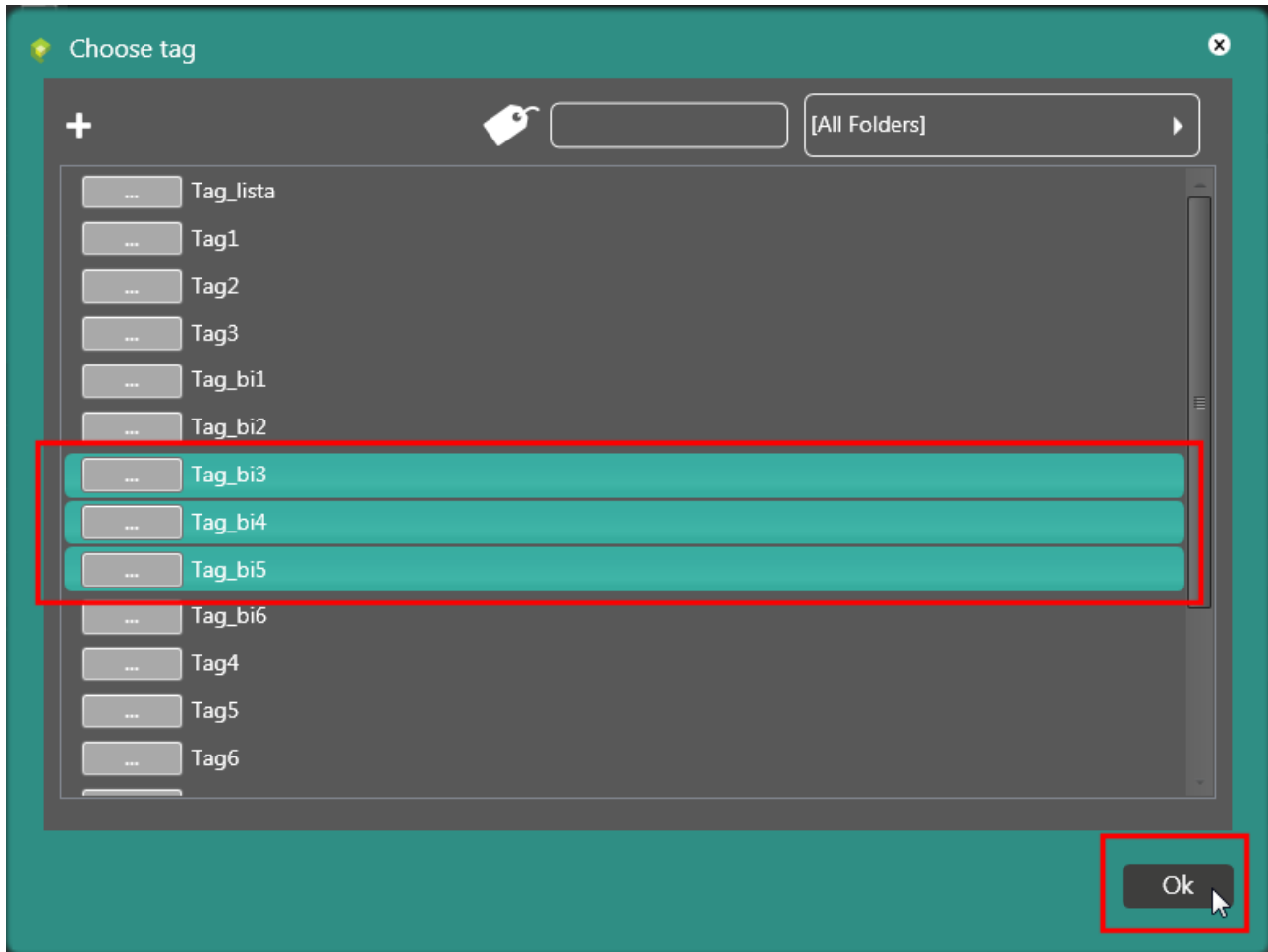


The multiple selection key is used to associate multiple objects to the Data Log at the same time.



# CREW Manual

Associate, for example, tags "Tag\_bi3", "Tag\_bi4" and "Tag\_bi5".





# CREW Manual

+ × ↶ ↷ DataLog1 ▶

Data log    Fields    Events

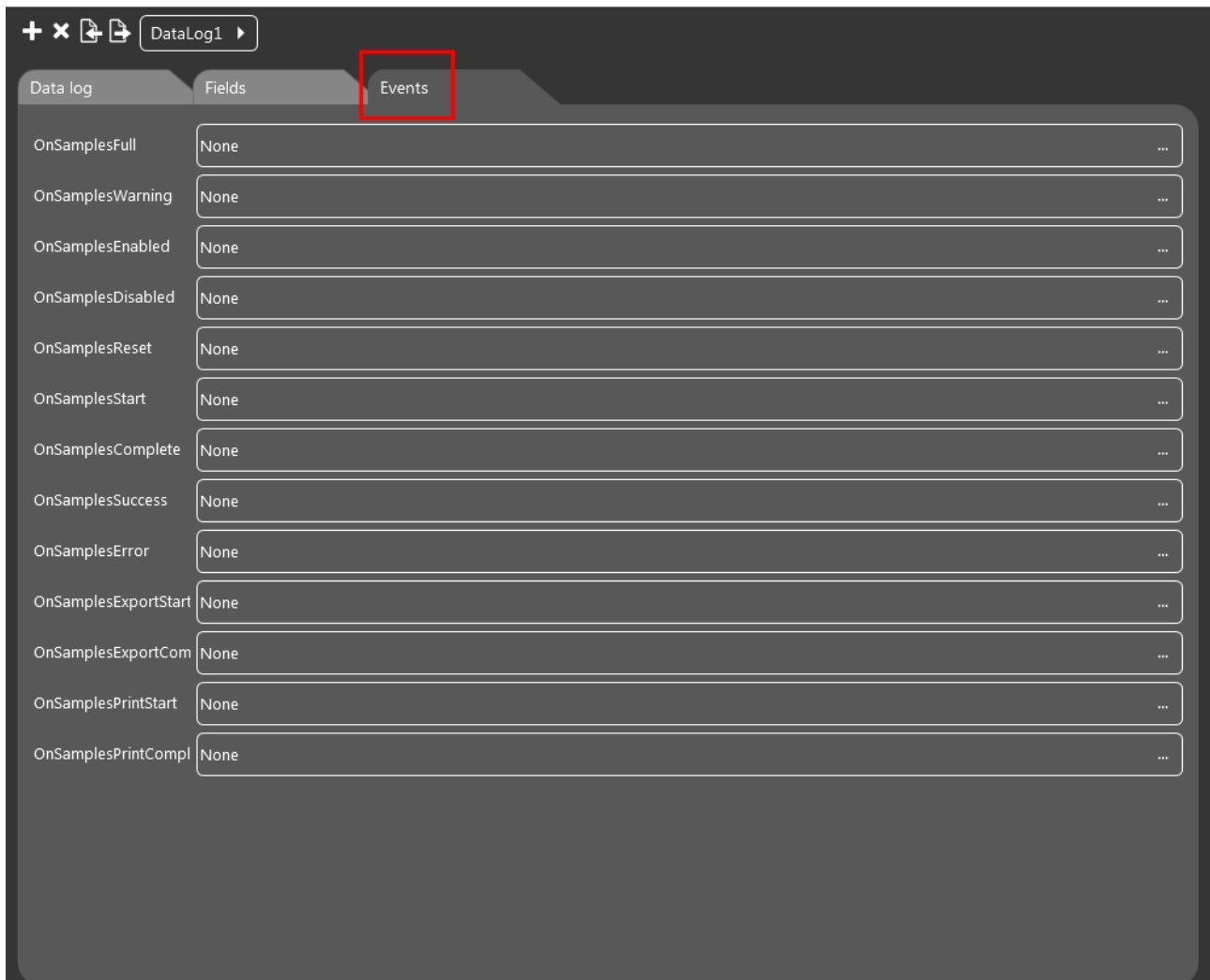
+ × ↶ ↷

	Details	Name ▾	Description ▾	Tag ▾	Header ▾	File name ▾
1	...	DataLogItem1	Data 1	Tag4	Data 1	
2	...	DataLogItem2	Data 2	Tag5	Data 2	
3	...	DataLogItem3	Data 3	Tag6	Data 3	
4	...	DataLogItem4	Data 4	Tag7	Data 4	
5	...	DataLogItem5		Tag_bi2		
6	...	Tag_bi3		Tag_bi3	Tag_bi3	
7	...	Tag_bi4		Tag_bi4	Tag_bi4	
8	...	Tag_bi5		Tag_bi5	Tag_bi5	
6						
7						
8						
9						
10						
11						

# CREW Manual

## Events

It is possible to associate a list of events to the element in question (see [“Events”](#) section).



# CREW Manual

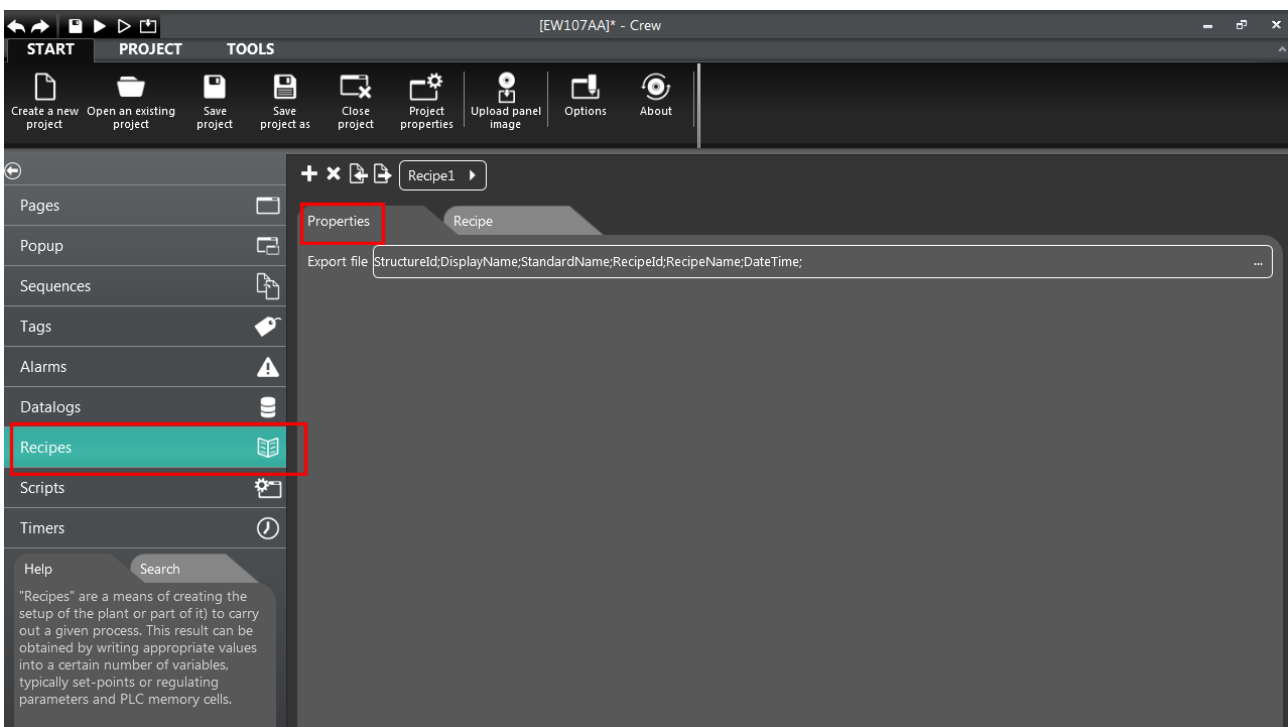
Event	Description
<b>OnSamplesFull</b>	Enabled when the samples buffer has reached full capacity
<b>OnSamplesWarning</b>	Enabled when the samples buffer has reached warning level
<b>OnSamplesEnabled</b>	Enabled when the samples buffer has been enabled
<b>OnSamplesDisabled</b>	Enabled when the samples buffer has been disabled
<b>OnSamplesReset</b>	Enabled when the samples buffer has been reset
<b>OnSamplesStart</b>	Enabled when new samples logging has been started
<b>OnSamplesComplete</b>	Enabled when new samples logging has been completed
<b>OnSamplesSuccess</b>	Enabled when new samples have been logged successfully
<b>OnSamplesError</b>	Enabled when new samples have been logged with errors
<b>OnSamplesExportStart</b>	Enabled when new samples export has been started
<b>OnSamplesExportComplete</b>	Enabled when new samples export has been completed
<b>OnSamplesPrintStart</b>	Enabled when new print of samples buffer has been started
<b>OnSamplesPrintComplete</b>	Enabled when new print of samples buffer has been completed

# CREW Manual

## Recipes

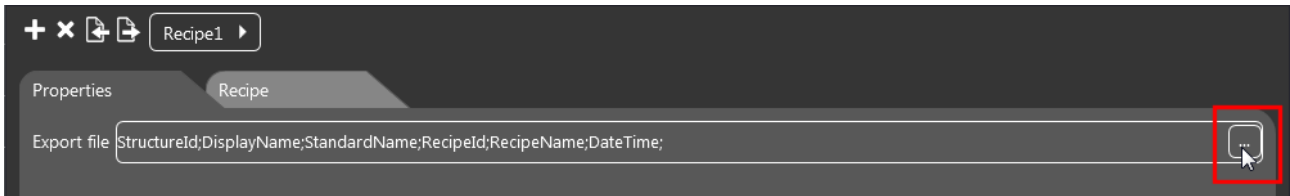
The recipes are used for the setup of the system (or parts of it) so that it performs a given process. In order to do so, you need to write the due values in a certain number of field tool tags, usually set-points or parameters of regulators and PLC memory cells.

From “Explore Project”, select “Recipes”.

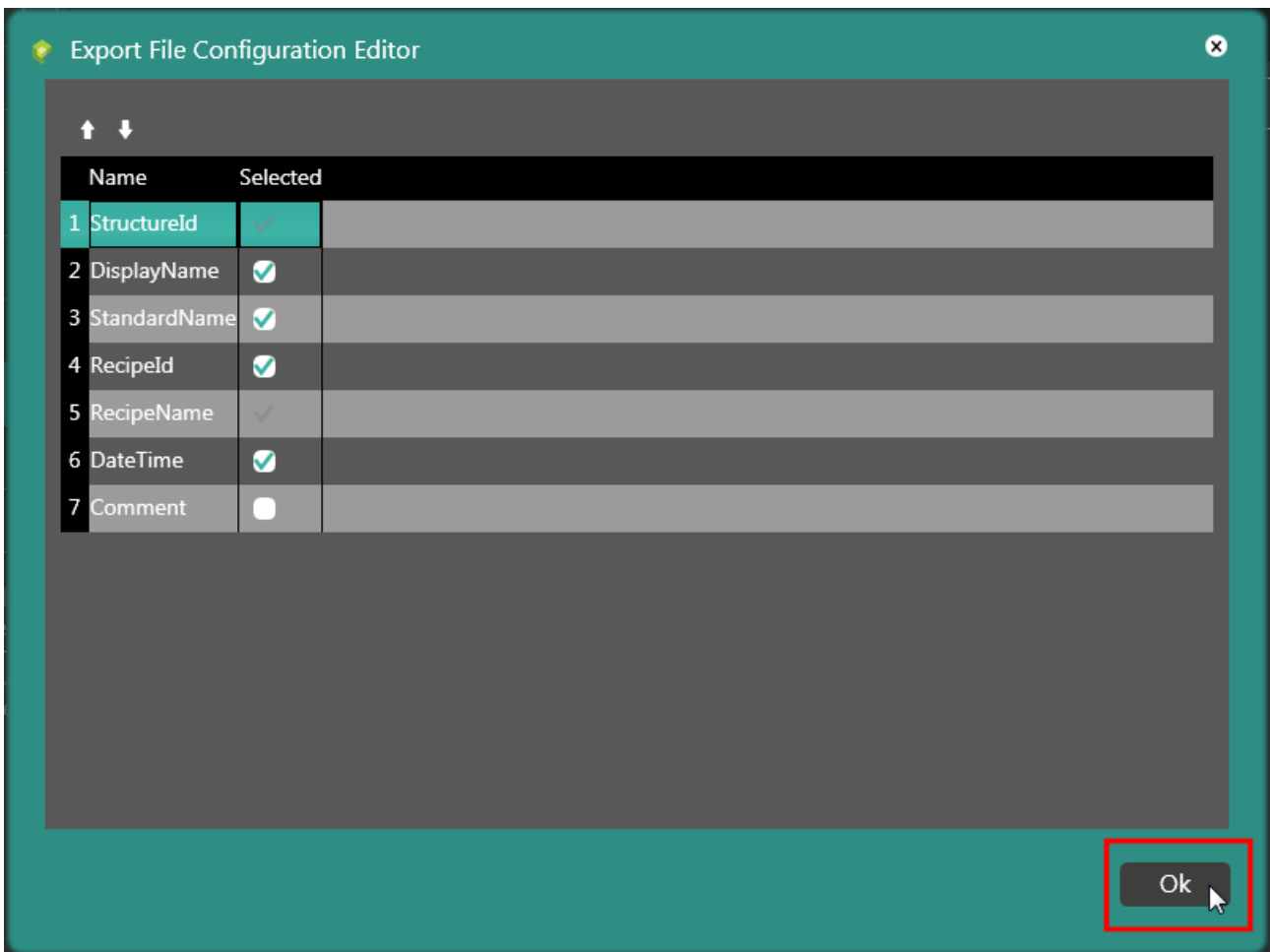


# CREW Manual

In the “Property” mask, click the “Browse” icon to select the options of the export file structure.



Choose the required options and confirm with “Ok”.



# CREW Manual

Click the “+” icon in the “Property” mask to create new recipes.



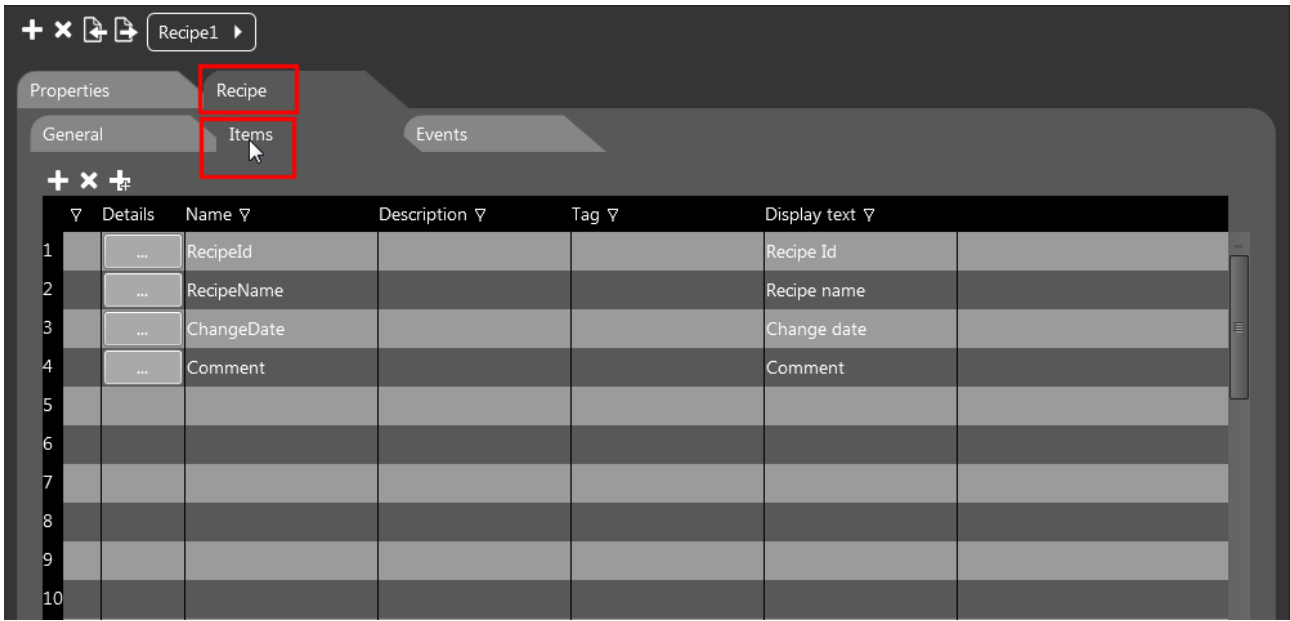
From the “General” mask it is possible to enter the name and description of the recipe.



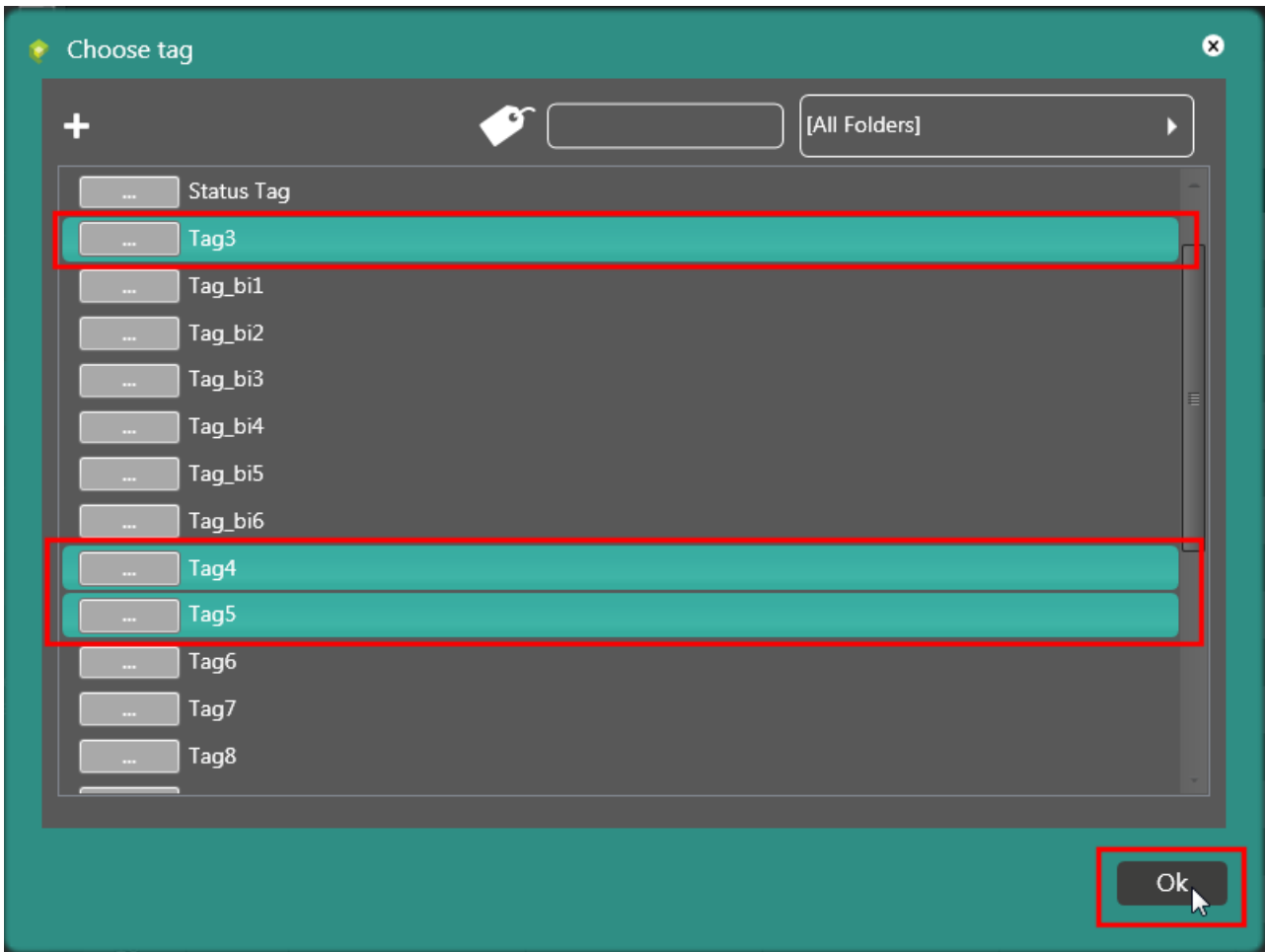
The “Command Area” and “Status Area” options offer the possibility of synchronising the recipe structure transfer through due reference tags (for more information refer to the "[SyncMode - Synchronised transfer](#)" section).

# CREW Manual

From the “Objects” mask it is possible to associate new objects to the recipe structure.

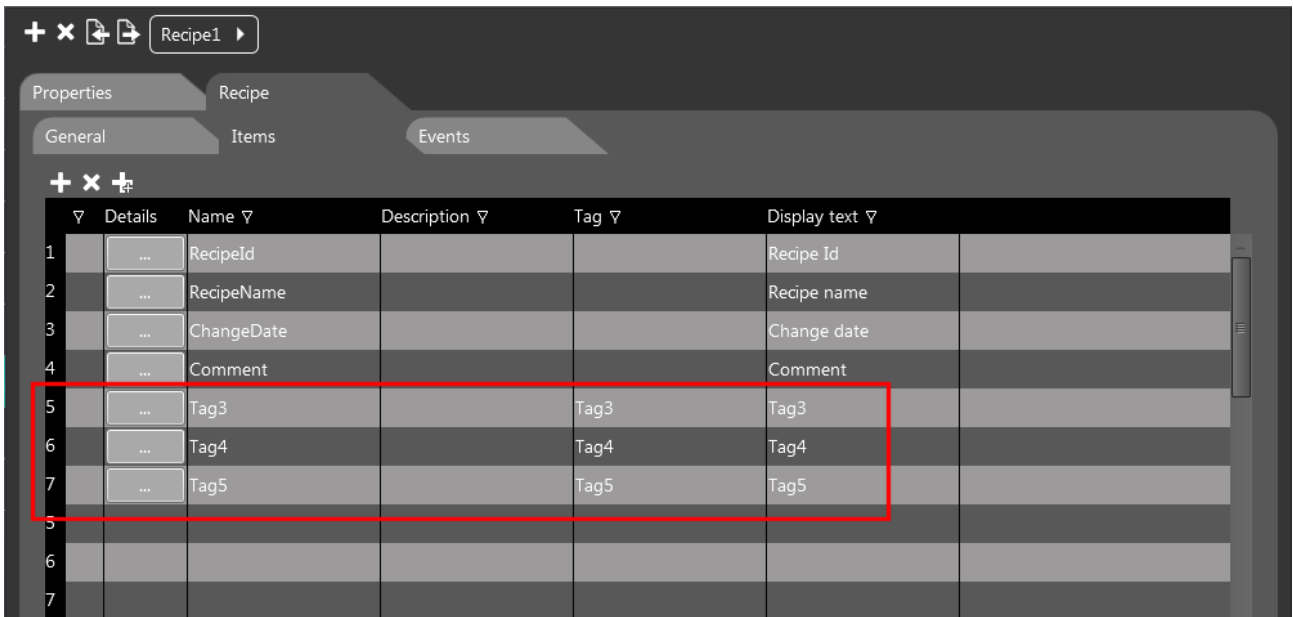


# CREW Manual





# CREW Manual

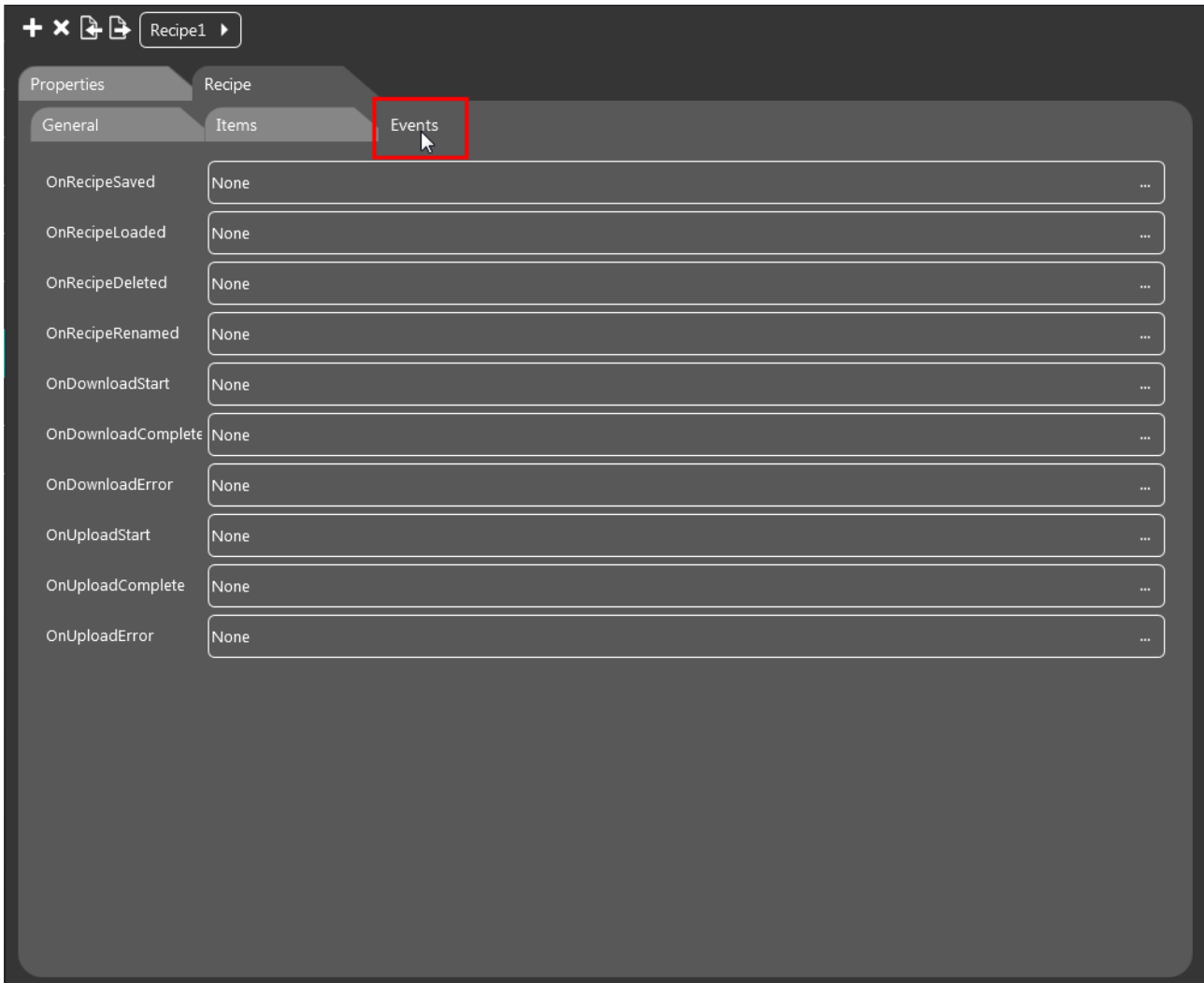


The screenshot shows the 'Recipe1' properties window with the 'Items' tab selected. The table below lists the items, with rows 5, 6, and 7 highlighted by a red box.

	Details	Name	Description	Tag	Display text
1	...	RecipeId			Recipe Id
2	...	RecipeName			Recipe name
3	...	ChangeDate			Change date
4	...	Comment			Comment
5	...	Tag3		Tag3	Tag3
6	...	Tag4		Tag4	Tag4
7	...	Tag5		Tag5	Tag5

# CREW Manual

From the “Events” mask it is possible to associate a list of events to the recipe (see “[Events](#)” section).



# CREW Manual

Event	Description
<b>OnRecipeSaved</b>	Enabled when a recipe (from the buffer) has been saved in the archives
<b>OnRecipeLoaded</b>	Enabled when a recipe (from the archive) has been uploaded to the buffer
<b>OnRecipeDeleted</b>	Enabled when a recipe has been deleted from the archive
<b>OnRecipeRenamed</b>	Enabled when a recipe in the archive has been renamed
<b>OnDownloadStart</b>	Enabled when transference to the device has been started (from either the buffer or the archive)
<b>OnDownloadComplete</b>	Enabled upon download completion from the terminal to the device
<b>OnDownloadError</b>	Enabled when errors appear while downloading from the terminal to the device
<b>OnUploadStart</b>	A transfer from the device is started (directly for storage or archive)
<b>OnUploadComplete</b>	Transfer from the device has been successfully completed
<b>OnUploadError</b>	Transfer from the device has ended with errors

# CREW Manual

## SyncMode - Synchronised transfer

Synchronised transfer ("SyncMode") is necessary if you wish to control the exchange of recipe data from the PLC program.

There are two "Status Area" and "Command area" associated Tag areas that are only used to manage the synchronised transfer of recipes from or to the external device.

### Status Area

The Status Area is composed of a single "word" and is written by the panel and read by the device:

WORD 0: status word – transfer status bits

# CREW Manual

Bits word status:

Bit	Description
0 (lsb)	1 = transfer in process
1	1 = download requested
2	-
3	1 = download completed
4	1 = upload requested
5	-
6	1 = upload completed
7	-
8	-
9	-
10	-
11	-
12	-
13	-
14	1 = download error
15 (msb)	1 = upload error

# CREW Manual

## Command Area

The Command Area is composed of a single “word” and is written by the device and read by the panel.

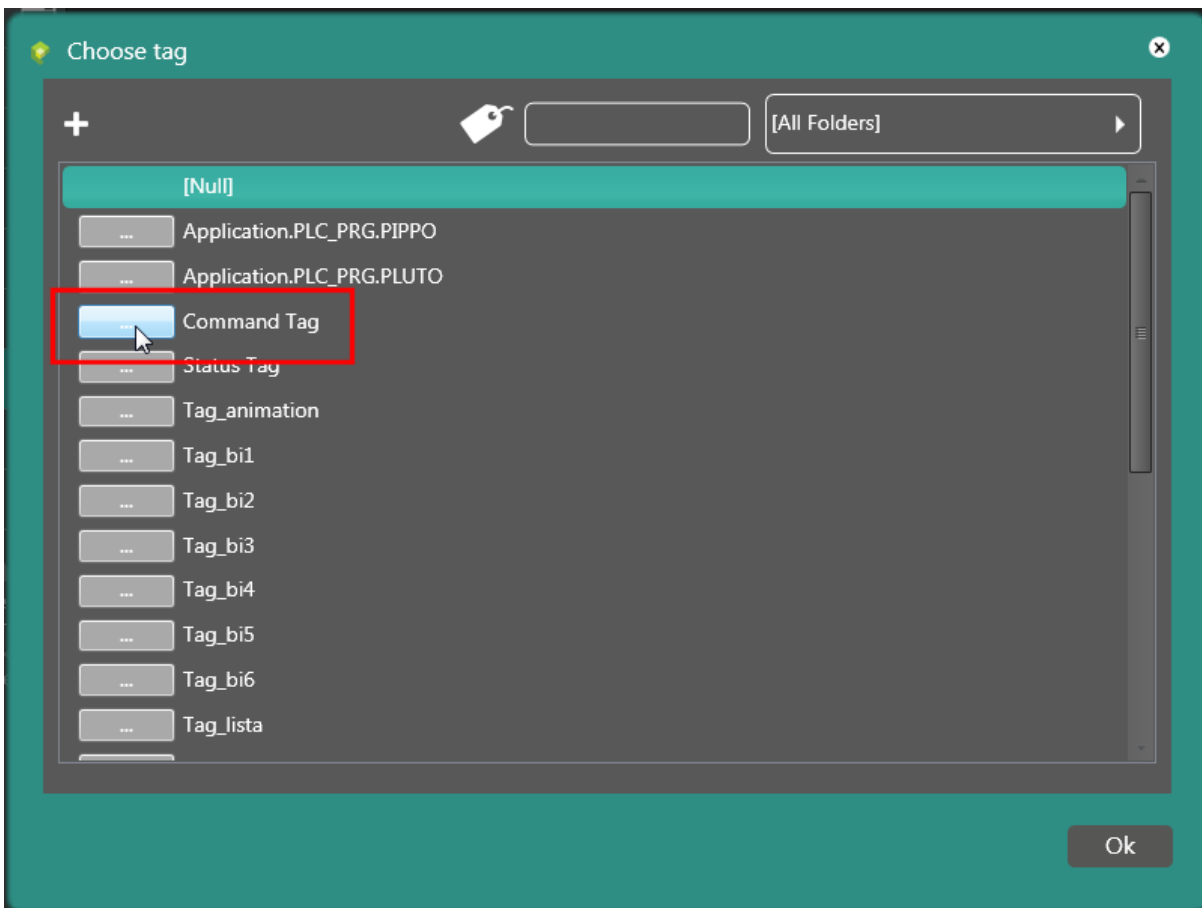
WORD 0: command word – transfer command bits

Bits word command:

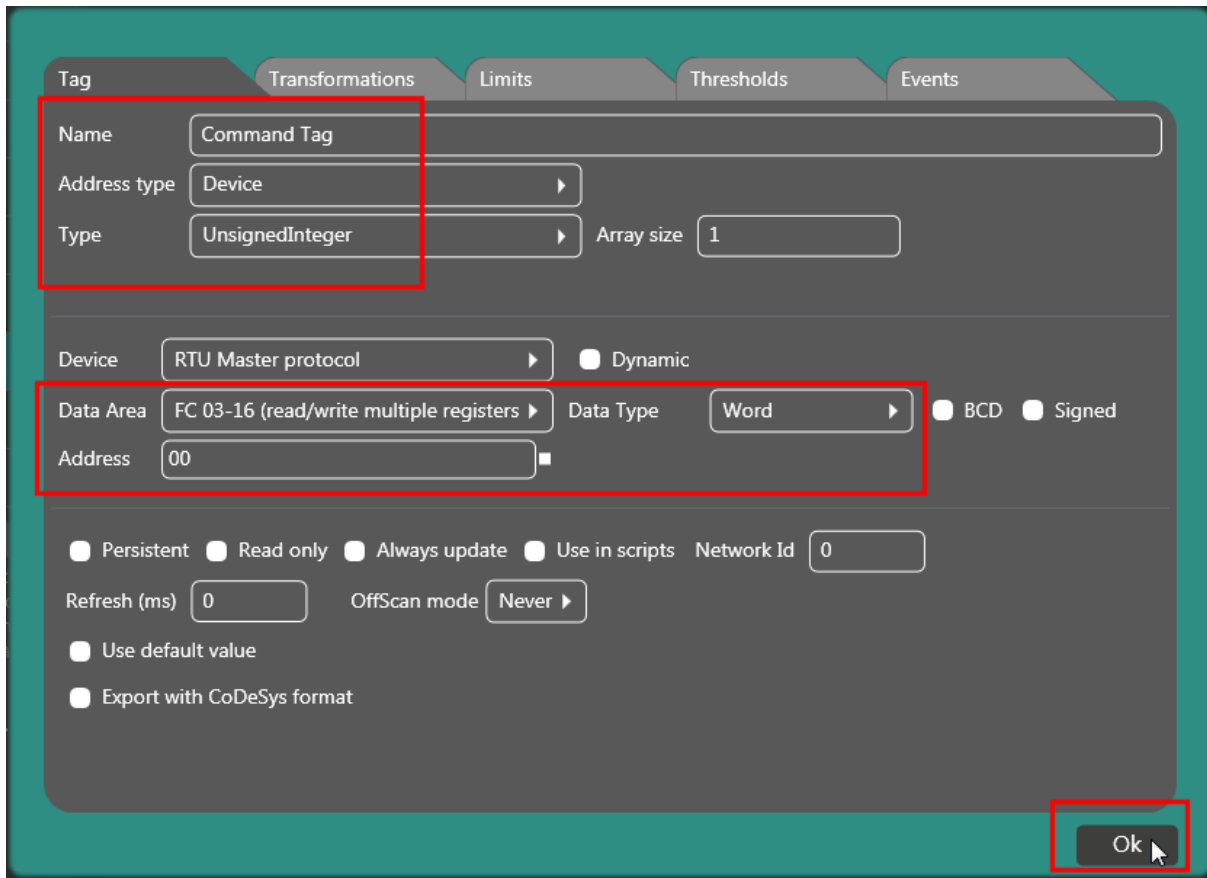
Bit	Description
0 (lsb)	1 = download confirmed
1	1 = upload confirmed
2	-
3	-
4	-
5	-
6	-
7	-
8	-
9	-
10	-
11	-
12	-
13	-
14	-
15 (msb)	-

# CREW Manual

The “Command Area” and “Status Area” options offer the possibility of synchronising the recipe structure transfer through due reference tags.



# CREW Manual



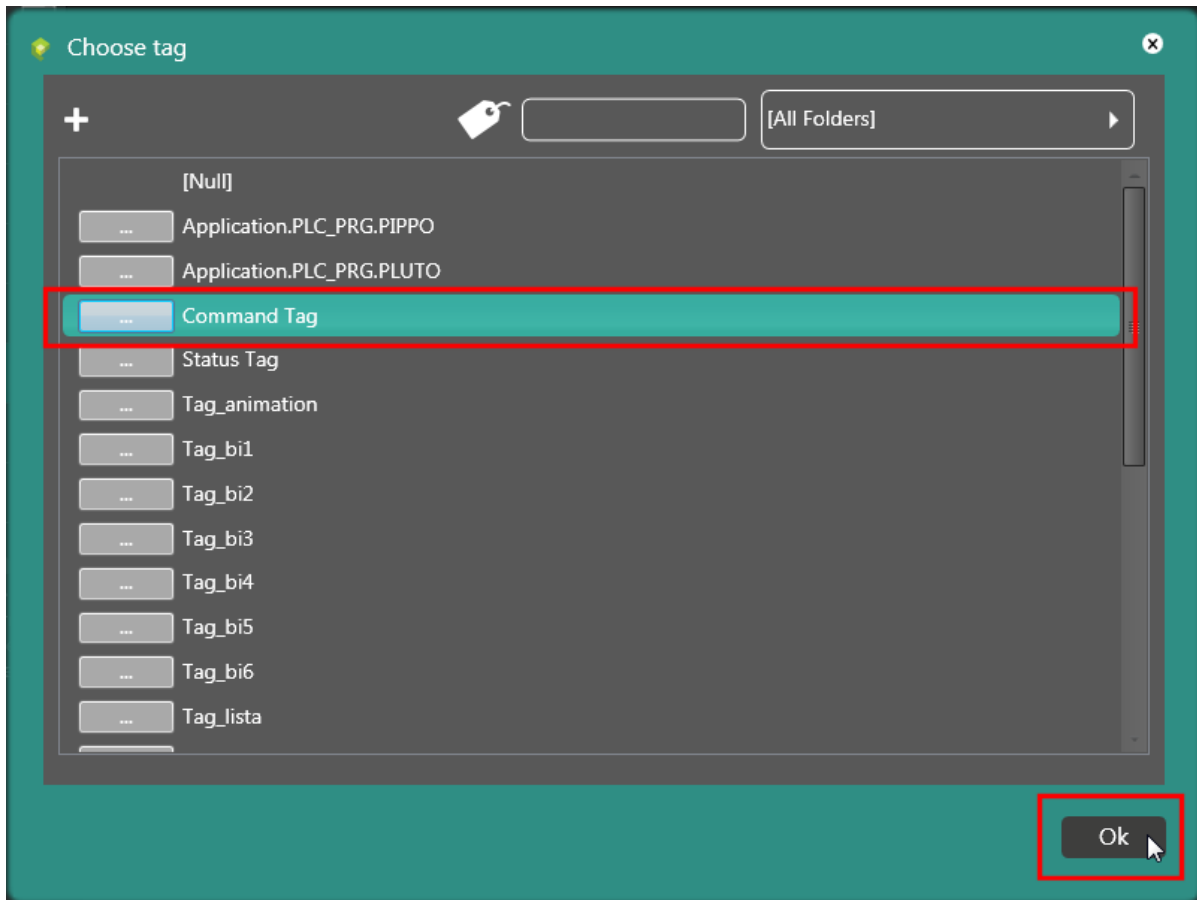
The screenshot displays the 'Tag' configuration window in the CREW software. The window has several tabs: 'Tag', 'Transformations', 'Limits', 'Thresholds', and 'Events'. The 'Tag' tab is active. The configuration is as follows:

- Name:** Command Tag
- Address type:** Device
- Type:** UnsignedInteger
- Array size:** 1
- Device:** RTU Master protocol
- Dynamic:**
- Data Area:** FC 03-16 (read/write multiple registers)
- Data Type:** Word
- BCD:**
- Signed:**
- Address:** 00
- Persistent:**
- Read only:**
- Always update:**
- Use in scripts:**
- Network Id:** 0
- Refresh (ms):** 0
- OffScan mode:** Never
- Use default value:**
- Export with CoDeSys format:**

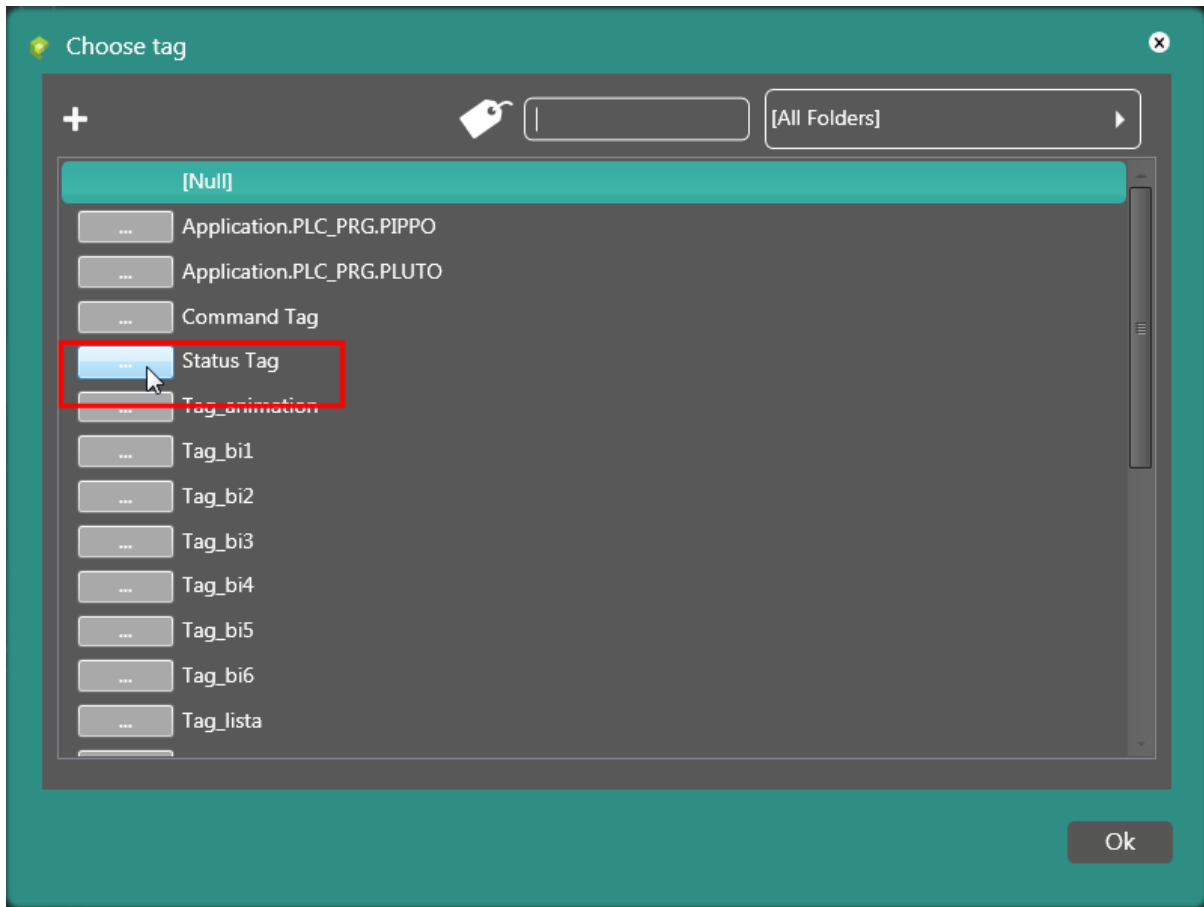
An 'Ok' button is located in the bottom right corner of the window.



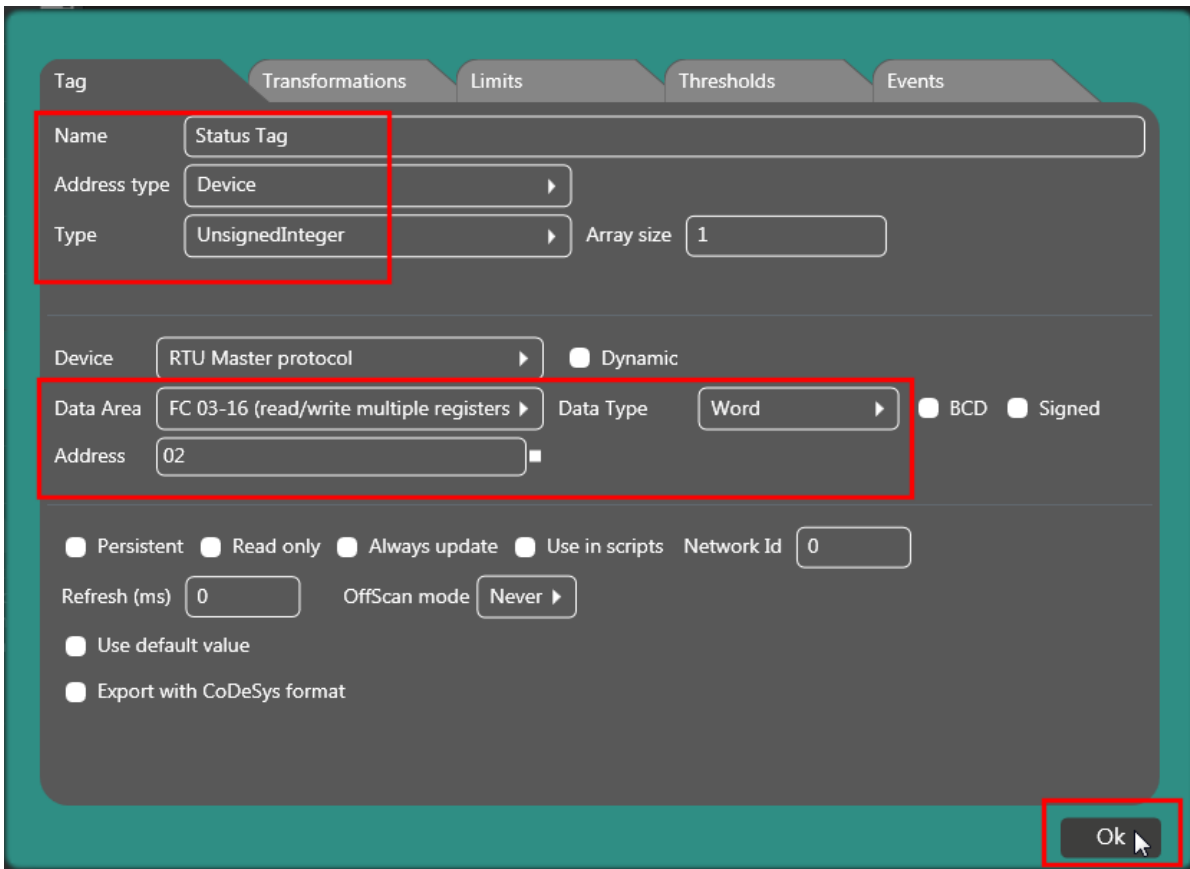
# CREW Manual



# CREW Manual



# CREW Manual

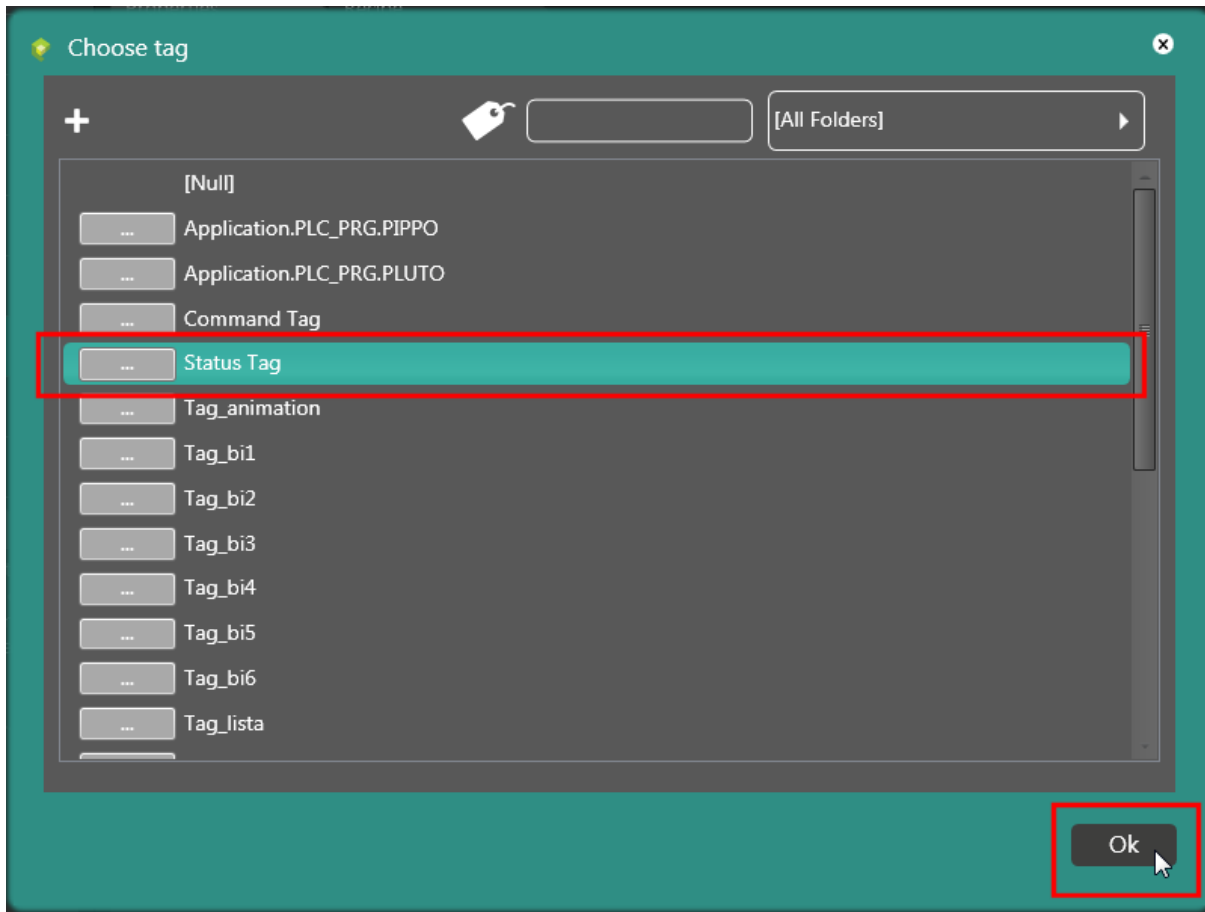


The screenshot displays the 'Tag' configuration window in the CREW software. The window has several tabs: 'Tag', 'Transformations', 'Limits', 'Thresholds', and 'Events'. The 'Tag' tab is active. The configuration fields are as follows:

- Name:** Status Tag
- Address type:** Device
- Type:** UnsignedInteger
- Array size:** 1
- Device:** RTU Master protocol
- Dynamic:**
- Data Area:** FC 03-16 (read/write multiple registers)
- Data Type:** Word
- BCD:**
- Signed:**
- Address:** 02
- Persistent:**
- Read only:**
- Always update:**
- Use in scripts:**
- Network Id:** 0
- Refresh (ms):** 0
- OffScan mode:** Never
- Use default value:**
- Export with CoDeSys format:**

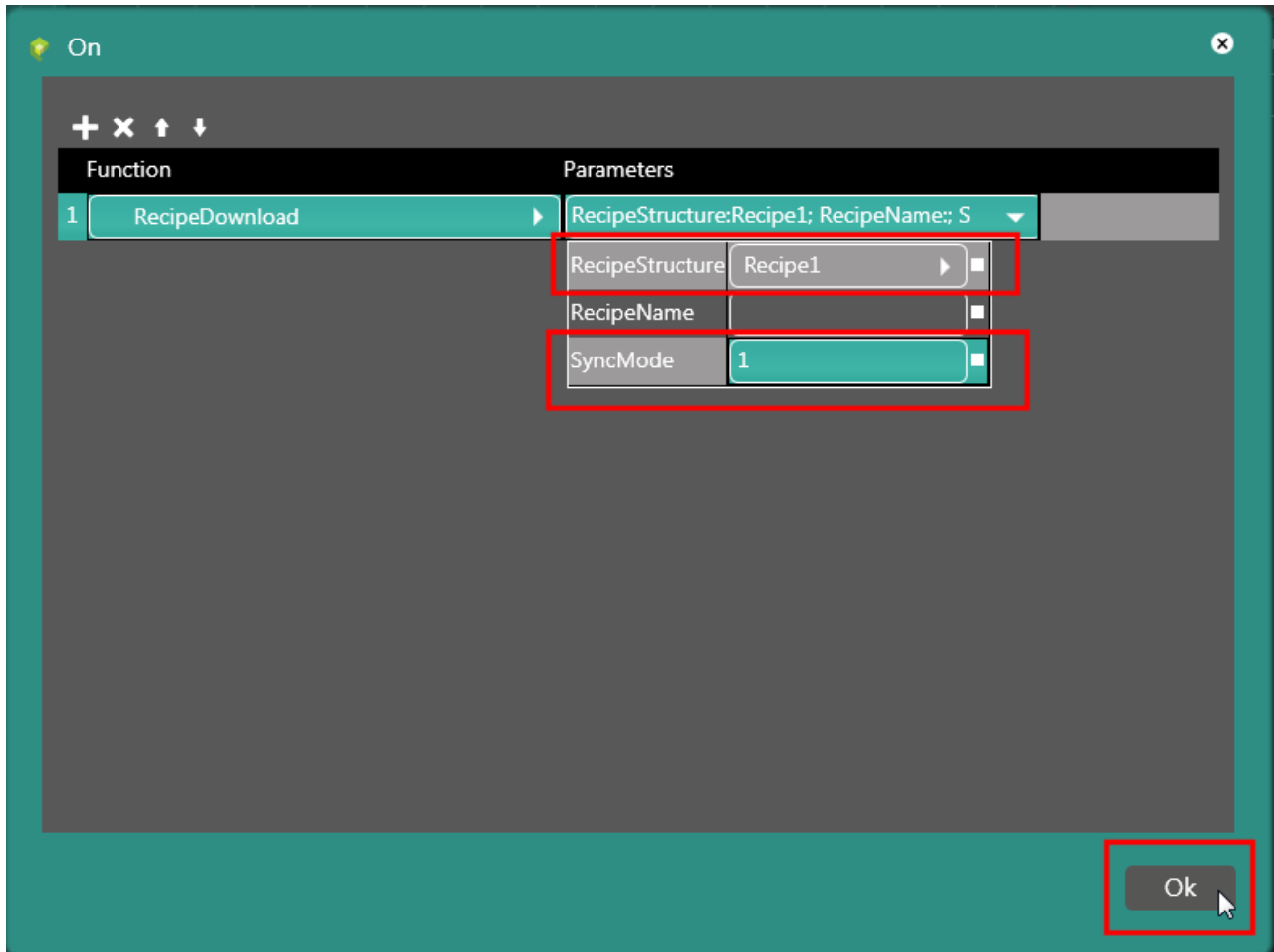
An 'Ok' button is located at the bottom right of the window.

# CREW Manual



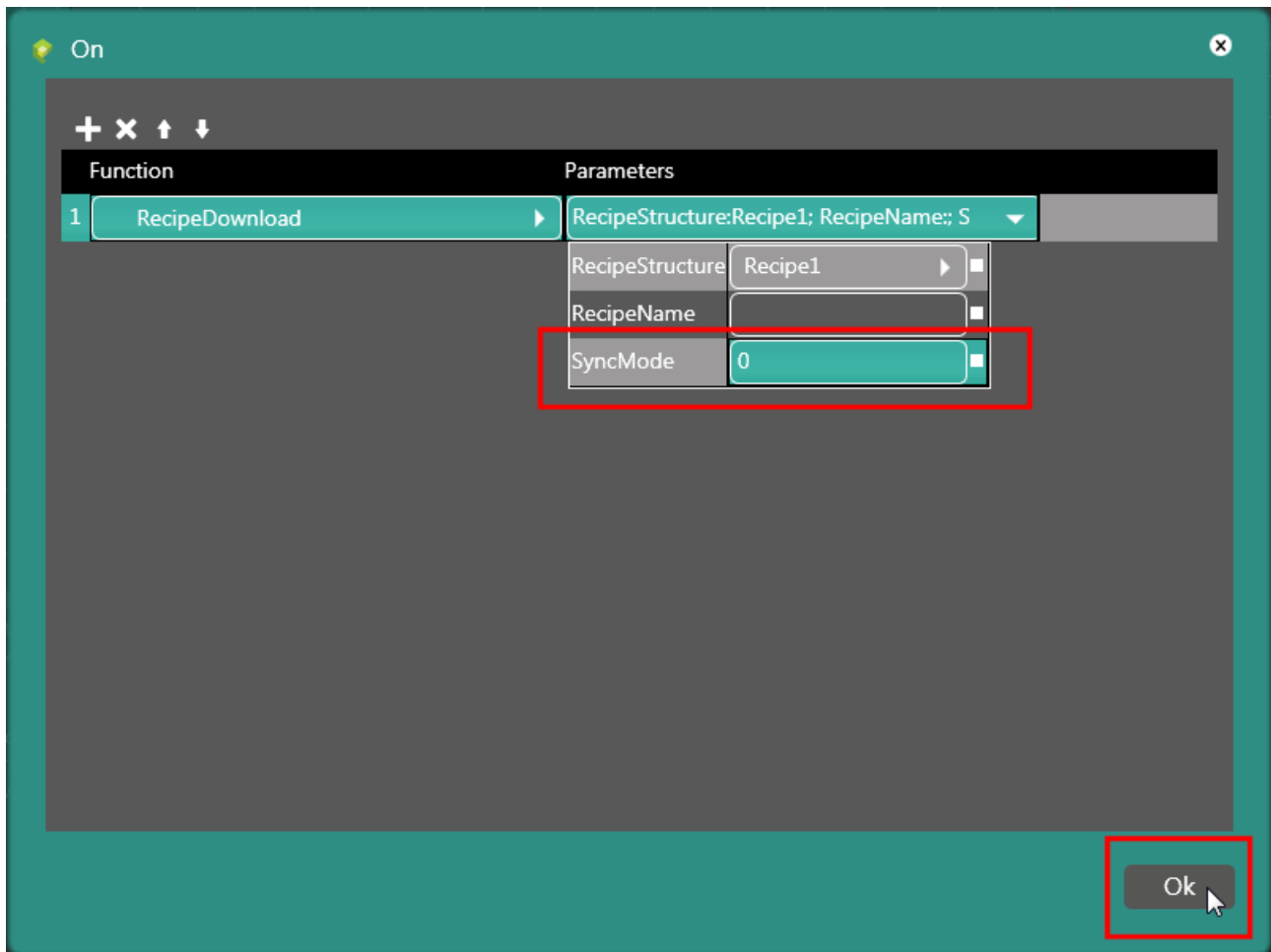
# CREW Manual

With the "SyncMode" parameter set to "1", it is possible to use recipe transfer synchronisation.



# CREW Manual

With the "SyncMode" parameter set to "0", recipe transfer is not controlled by any reference variable and is carried out automatically.



# CREW Manual

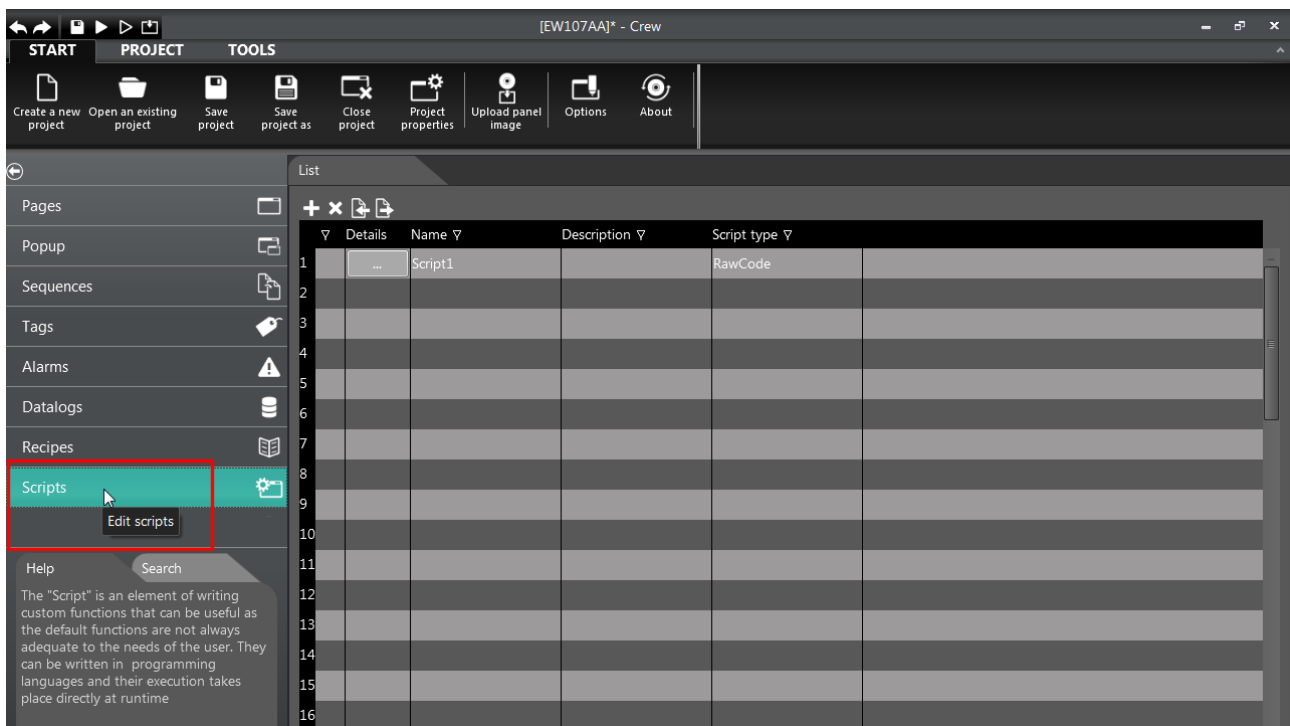
## Scripts

With Crew the programmer can enter actual programs or functions into his/her projects that make it possible to manage and change all of the components in Runtime (graphic objects, variables, recipes, etc.) of their application.

Thanks to this feature, the user can complete the set of predefined functions provided by Crew with those made by him/her according to his/her needs.

The user scripts can be referred to in the project when a button is pressed, when an event happens or even after being retrieved by other scripts.

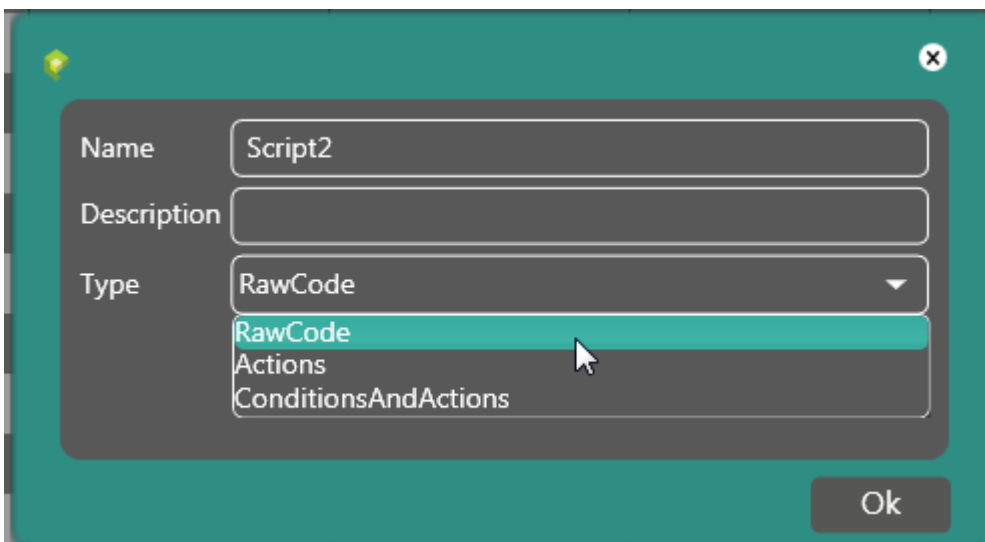
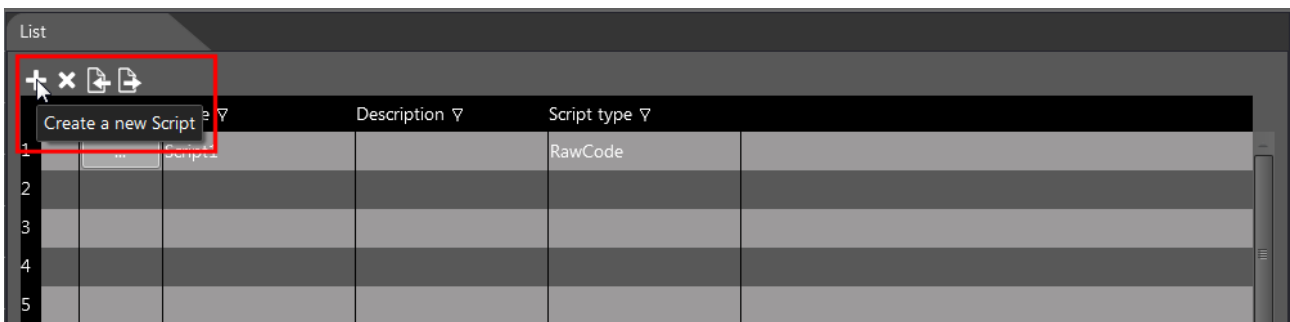
The “Scripts” can be entered in the project by “Explore Project”.



# CREW Manual

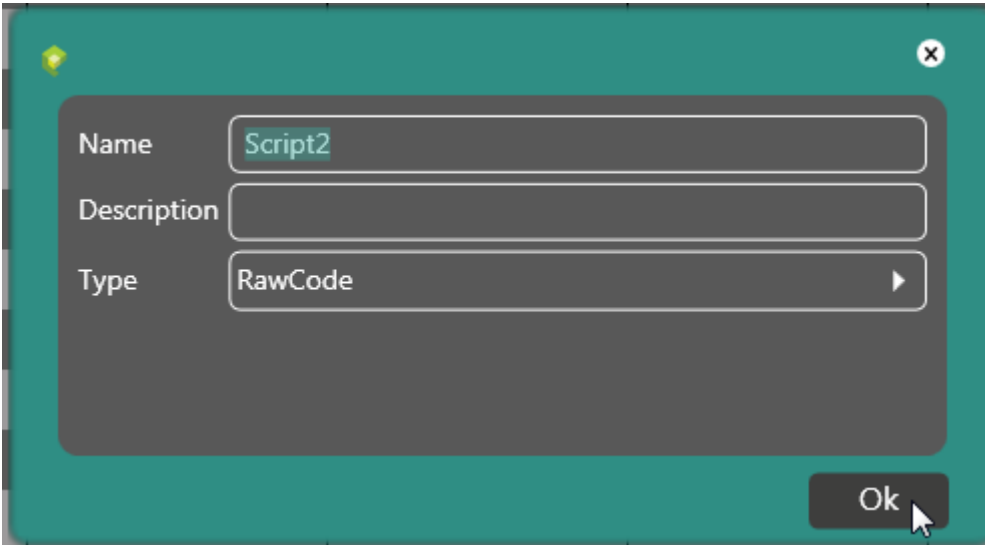
Their code can be prepared using simple programming/scripting languages such as VBScript. For programming techniques (variable declarations, operators, conditional structures and predefined functions) the user is required to refer to specialist manuals relative to the language that he/she intends to use.

Creating a new script:





# CREW Manual



Name

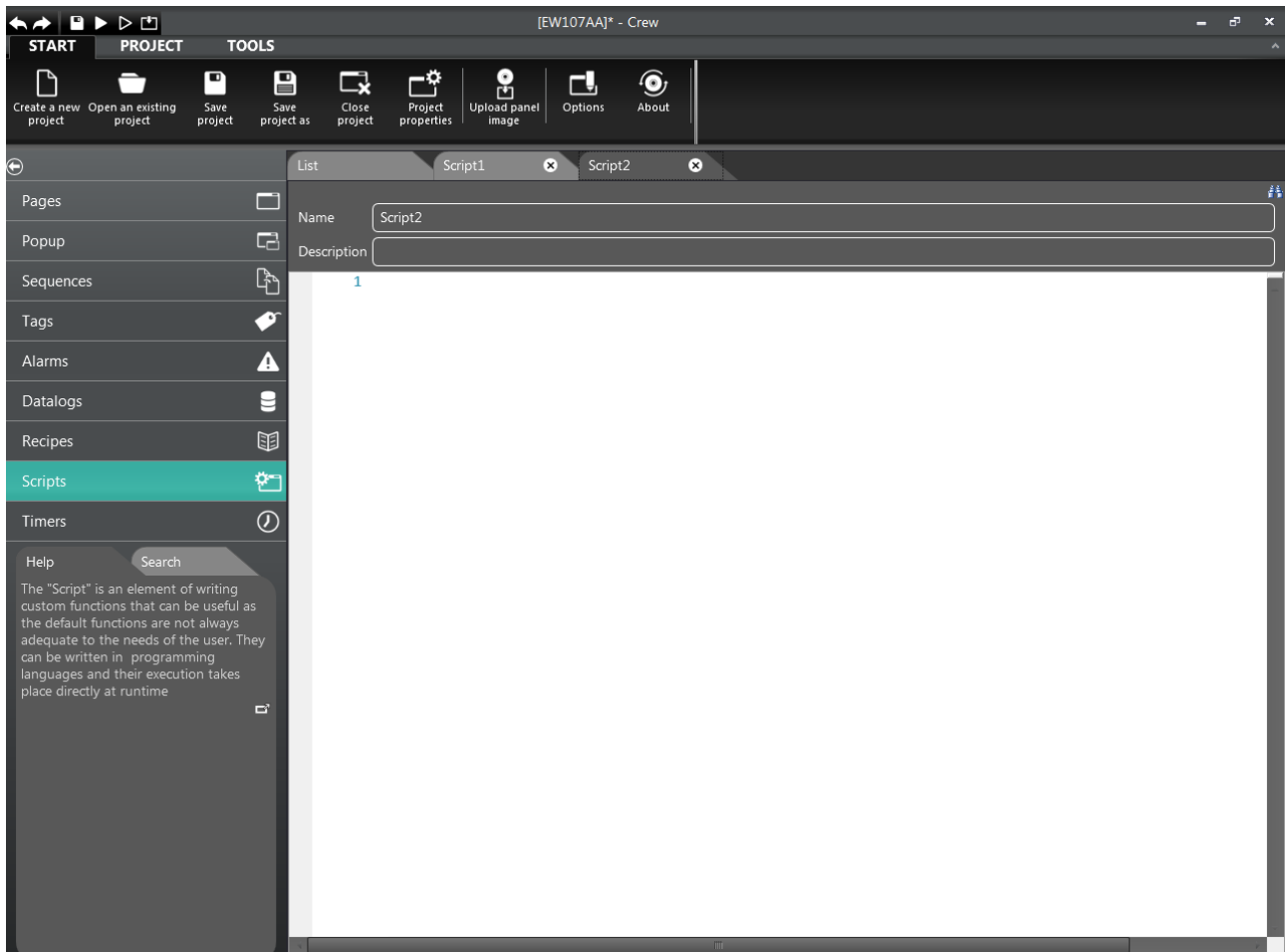
Description

Type

Ok

In Crew, once the script is added by “Explore Project”, it is possible to use the relative editor page to prepare the code. The editor runs a check, in real time, of the code’s syntax and immediately provides a warning on the screen if any imprecisions are found in the formula of the instructions.

# CREW Manual



Place the cursor of the mouse on it to view the complete description of the problem. Errors and the relative description are also listed at the time of project validation and compilation.

To facilitate code preparation, the editor shows the list of objects and the properties available for the added object (Intellisense mechanism). Said list appears every time you press the separation point between objects, or between an object and the method (or property) being retrieved.

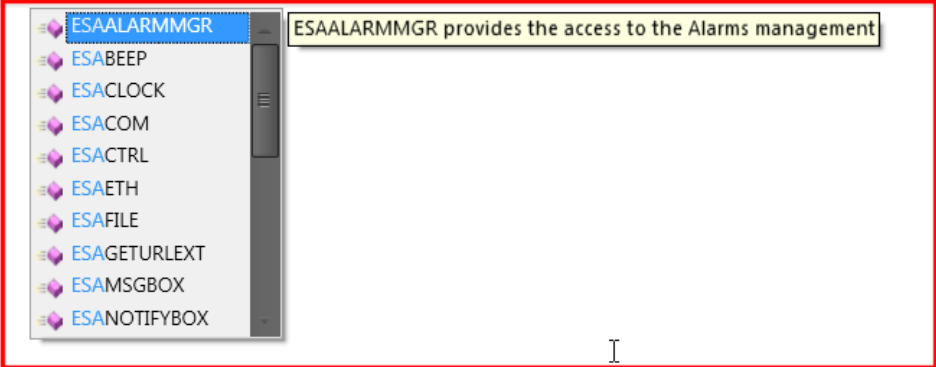
# CREW Manual

List Script2

Name Script2

Description

```
1 ESAHMI.ESAPAGE("Page").ESACNTRL("Label").ControlWidth=67.  
2 ESAHMI.ESA
```



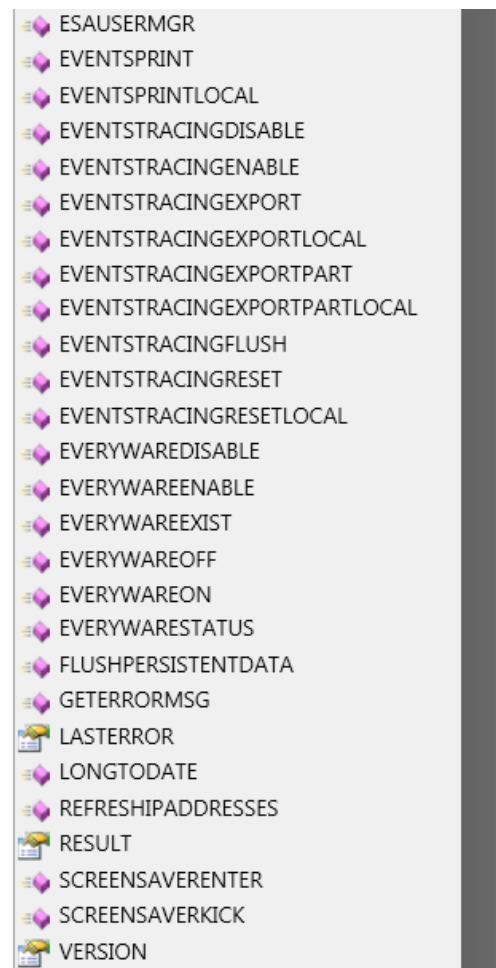
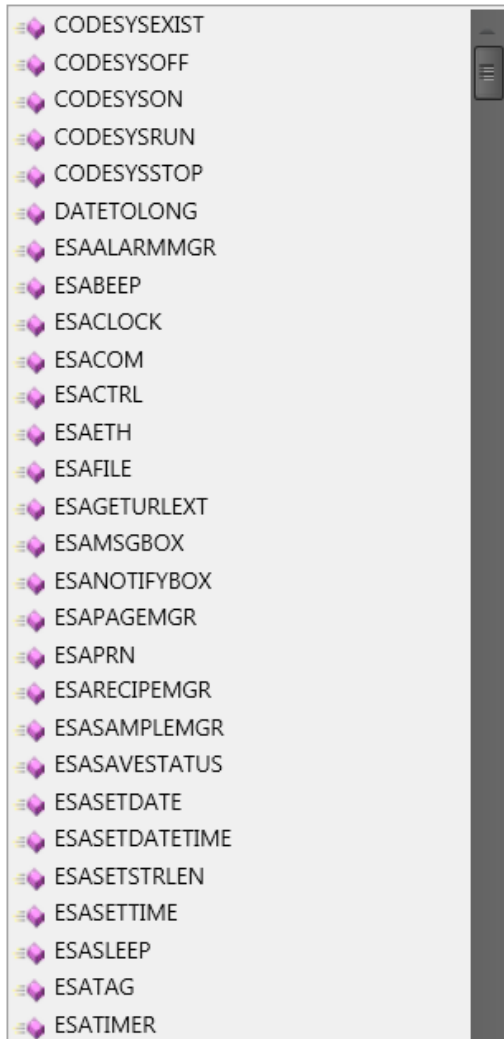
ESAALARMGR provides the access to the Alarms management

I

# CREW Manual

In code editing, the objects are, in fact, separated by their relative children or methods, by inserting a “.” (dot). Below is the tree hierarchy of the objects accessible from script.

## 1 | ESAHMI.



Therefore, to indicate an element of the page you will need to use an instruction such as:

**ESAHMI.ESAPAGE("Page").ESACNTRL("Label").ControlWidth=67.**

# CREW Manual

For objects that need to switch the name of the reference object (for example ESAPAGE, ESACNTRL, etc.), after opening the brackets simply press the “?” key on the keyboard to retrieve the list of objects available for insertion.

Various objects accessible from script, their properties and their functions, with practical examples of their use, will be described in the following paragraphs of this chapter.



Note: In the paragraphs below, some properties are identified as being read-only from script. For many of these properties, however, there is no physical protection, therefore the script has the possibility of overwriting the value. This operation is not recommended, however. The programmer is therefore required to ensure that the read-only properties (R) are not changed from the scripts.

# CREW Manual

## Key of Variable Types and premises on syntax

The paragraphs below discuss the properties and characteristic methods of the objects. The table here below provides a brief key of the abbreviations that are used.

Variable	Abbreviation
Whole	Int
String	Str
Boolean	Bool
Long	Long
Double	Dbl
RGB (color, returned by the RGB function)	RGB
Variant	Var
R	Read, read-only
RW	Read&Write, read and write

If a subroutine (a method that does not return any value) requires an input parameter, the step can be made using brackets or as below:

ESAHMI.ESAMSGBOX "Text"

ESAHMI.ESAMSGBOX("Text")

When a subroutine requires multiple input parameters, they must be written consecutively, separated by a comma (without brackets) as below:

ESAHMI.ESAPAGEMGR.ShowPageByNumber 32,0

# CREW Manual

If a function (a method that returns a value) requires one or more input parameters, the step can be made using brackets as below:

```
a=ESAHMI.ESATAG("Tag_Array").GetTagBitValue(1)
```

```
a=ESAHMI.ESAPAGEMGR.GetTAGBuffer ("RecipeType", "RecipeName")
```

## Notes

### Upper Case

The names of all the methods and properties are not case sensitive.

For example:

```
ESAHMI.ESAMSGBOX 123
```

is the same as:

```
ESAhmi.ESAMsgbox 123
```

### Subroutine

When not specified, the method does not return any value (subroutine).

### Functions that return boolean values

FALSE must be treated as "zero" (= 0).

TRUE must be treated as "any value other than zero" (<> 0).

### Parameters

Unless specified otherwise, the parameters are input-parameters.

The output-parameters are expressly stated.

### Boolean parameters

FALSE must be passed as "zero" (0).

TRUE must be passed as "one" (1).

# CREW Manual

## Optional parameters

Optional parameters are given in brackets (for example [suspensive]).

The default value of the suspensive parameter is defined in the Configurator at the time of design.

## “Local” Functions

There are pairs of functions defined as “Local” or “Global” (for example EventsPrintLocal and EventsPrint).

“Local” functions are from the client side, “Global” ones are from the server side.



# CREW Manual

## ESAHMI

ESAHMI is the main ESA object.

ESAHMI offers numerous features for the methods and provides access to ESA sub-objects.

## Sub-Objects

ESAALARMGR

ESACOM

ESACTRL

ESAETH

ESAFILE

ESAPAGEMGR

ESAPRN

ESARECIPEMGR

ESASAMPLEMGR

ESATAG

ESATIMER

ESAUSERMGR

## Shared properties

The following properties are common to all ESA objects:

<b>Name</b>	<b>Type</b>	<b>Read-Write</b>	<b>Description</b>
<b>Version</b>	integer	R	Object release version

# CREW Manual

Examples:

```
v = ESAHMI.ESAFILE.version
```

```
set obj = ESAHMI.ESAFILE
```

```
v = obj.version
```

## Methods - ESAHMI

**ESAMsgBox( message, [suspensive] )**

message (variant)                      body of message

suspensive (boolean-optional) TRUE = suspensive, FALSE = not suspensive

This shows the message box:



For example:

```
ESAHMI.ESAMSGBOX "Hello world!"
```

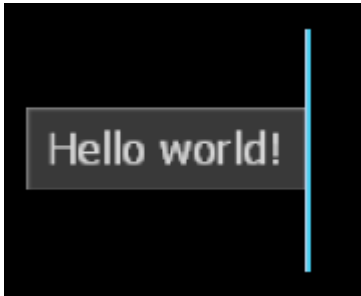
```
ESAHMI.ESAMSGBOX 1234.56
```

**ESANotifyBox( message )**

message (variant)                      body of message

# CREW Manual

This shows a scrolling notice box



## ESASleep( msec )

msec (integer) milliseconds

This suspends execution of the script for a specific interval.

## ESABeep( frequency, duration )

frequency (integer) sound frequency (Hertz)

duration (integer) sound duration (milliseconds)

This generates simple tones in the speakers

## ESASaveStatus()

This empties the Windows log.

## ESASetDate( Day, Month, Year )

Day (integer) Day

Month (integer) Month

Year (integer) Year

This sets the system date (valid range: 1970-2105).

# CREW Manual

## ESASetTime( Hour, Minutes, Seconds )

Hour (integer) Hour  
Minutes (integer) Minutes  
Seconds (integer) Seconds

This sets the time on the system.

## ESASetDateTime( date )

Date (DATE) Date & Time

This set both date and time on the system.

Example:

' set date value

v = datevalue("28/12/2013") or

v = dateserial(2013,12,28) or

v = CDATE("28/12/2013")

' set time value

v = timevalue("21:34:56") or

v = timeserial(21,34,56) or

v = CDATE("21:34:56")

' set date and time value

v = CDATE("04/10/2013 17:30:11")

' set system date and time

# CREW Manual

## ESAHMI.ESASetDateTime v

' display date and time

d = day(v)

m = month(v)

y = year(v)

h = hour(v)

p = minute(v)

s = second(v)

ESAHMI.ESAMsgbox d & "-" & m & "-" & y & " " & h & ":" & p & ":" & s

## FlushPersistentData()

Save all persistent data on disc.

## ESASetStrLen( String, Character, Len, Right )

String (string) original string (it can be empty)

Character (string) filling character (the first character is used)

Len (integer) final length of the string.

If Len is greater than the length of the original string, the final string

is filled using Characters, otherwise the original string is cut off at the first Len characters

Right (boolean): TRUE = the fill characters are placed at the end of the original string

FALSE = the fill characters are placed at the beginning of the original string

This builds a string from another string. The new string is filled or cut off.

RETURN (string): the new string

# CREW Manual

Example:

```
s = ESAHMI.ESASetStrLen( "hello", "#", 10, 1 )
```

## ESAClock()

RETURN (integer): the number of milliseconds since the device was started up.

## DateToLong( date )

Date (DATE) Date & Time

This converts a VBS-DATE into a LONG date

RETURN (integer): the date in LONG format  
(number of seconds elapsed since 01-01-1970 00:00:00)

Example:

```
t = CDATE("04/10/2013 17:30:11")  
v = ESAHMI.DateToLong(t)
```

## LongToDate( Date )

Date (integer) Date & Time

This converts a LONG date into a VBS-DATE

RETURN (integer): the date in VBS-DATE format

Example:

```
t = ESAHMI.LongToDate(value)  
d = day(t)  
m = month(t)  
y = year(t)
```

## GetErrorMsg( error )

error (integer) System code error

RETURN (string): the message corresponding to the specific error code

# CREW Manual

Example:

```
error = ESAHMI.LastError  
  
str = ESAHMI.GetErrorMsg(error)  
ESAHMI.ESAMsgbox str
```

ESAGetUriExt( Url, Dest, Proxy, ProxyUsername, ProxyPassword, ServerPort, ServerUsername, ServerPassword )

Url (input parameter, string): the address of the file to be translated

Dest (input parameter, string): complete file destination path

Proxy (input parameter, string): IP address of proxy:port (this can be an empty string)

ProxyUsername (input parameter, string): the username used for the authentication process to the proxy (it may be empty if the proxy is not used)

ProxyPassword (input parameter, string): the password used for the authentication process to the proxy (it may be empty if the proxy is not used)

ServerPort (input parameter, integer): server port (default=80)

ServerUsername (input parameter, string): the username used for the authentication process to the server (it may be empty if the server is not used)

ServerPassword (input parameter, string): the password used for the authentication process to the server (it may be empty if the server is not used)

This obtains a file from Url.

Example:

```
ESAHMI.ESAGetUriExt "http://198.168.100.1/image.jpg", "\picture1.jpg",  
"proxy:8080", "PROXYUSER", "PROXYPASSWORD",  
5001, "admin", "admin"
```

# CREW Manual

## ScreenSaverEnter()

This activates the screen-saver (the screen-saver must be enabled in the project).

## ScreenSaverKick()

This resets the screen-saver timeout.

## RefreshIpAddresses()

This updates the system variables with the list of the active IP address.

## EventsTracingEnable()

This enables FDA detection.

## EventsTracingDisable ()

This enables FDA detection.

## EventsTracingFlush()

Save all of the FDA detection data to disc.

## EventsTracingExport( Pathname )

Pathname (string) Full file path name.

Export all of the FDA findings data to text file.

RETURN (integer): the number of exported archives.

## EventsTracingExportLocal( Pathname, [suspensive] )

Pathname (string) Full file path name.

If the Pathname is an empty string (""), a dialog box appears so that the user can select a path.

suspensive (boolean-optional) TRUE = suspensive dialog box

FALSE = non-suspensive dialog box

Export all of the FDA findings data to text file.

RETURN (integer): the number of exported archives.



# CREW Manual

## EventsTracingExportPart( Pathname, From, To )

Pathname (string) Full file path name.

From (DATE) Start date/time

To (DATE) End date/time

Export all of the FDA findings data between dates/times to text file.

RETURN (integer): the number of exported archives.

Example:

```
t1 = CDATE("04/10/2013 08:00:00")
t2 = CDATE("04/10/2013 20:00:00")
n = ESAHMI.EventsTracingExportPart( "\data.txt", t1, t2 )
```

## EventsTracingExportPartLocal( Pathname, From, To, [suspensive] )

Pathname (string) Full file path name.

If the Pathname is an empty string (""), a dialog box appears so that the user can select a path.

From (DATE) Start date/time

To (DATE) End date/time

suspensive (boolean-optional) TRUE = suspensive dialog box

FALSE = non-suspensive dialog box

Export all of the FDA findings data between dates/times to text file.

RETURN (integer): the number of exported archives.

Example:

```
t1 = CDATE("04/10/2013 08:00:00")
t2 = CDATE("04/10/2013 20:00:00")
n = ESAHMI.EventsTracingExportPartLocal( "", t1, t2 )
```

# CREW Manual

## EventsTracingReset( Pathname )

Pathname (string) Full file path name

Export all of the FDA findings data to text file and clean the internal buffer.

## EventsTracingResetLocal( Pathname, [suspensive] )

Pathname (string) Full file path name

If the Pathname is an empty string (""), a dialog box appears so that the user can select a path.

suspensive (boolean-optional) TRUE = suspensive dialog box

FALSE = non-suspensive dialog box

Export all of the FDA findings data to text file and clean the internal buffer.

## EventsPrint()

## EventsPrintLocal()

This prints FDA detection data.

RETURN (integer): The number of printed events.

## EverywareOn()

## EverywareOff()

This switches the EveryWare process on/off.

## EverywareEnable()

## EverywareDisable()

This enables/disables the EveryWare process.

## EverywareStatus()

This achieves the EveryWare status.

RETURN (integer): 0 = process not on (!0 = process active)

1 = process enabled

2 = process disabled

... = process enabled, connection error

# CREW Manual

## EverywareExist()

This achieves the EveryWare status.

RETURN (integer): TRUE = active process

FALSE = inactive process

## CoDeSysOn()

## CoDeSysOff()

This switches the CODESYS process on/off.

## CoDeSysRun()

## CoDeSysStop()

This starts/Interrupts the CODESYS application.

## CoDeSysExist()

This achieves the CODESYS status.

RETURN (integer): TRUE = active process

FALSE = inactive process

## Properties

Name	Type	Read-Write	Description
<b>Result</b>	integer	R	Result of the last ESA call (0 = no error)
<b>LastError</b>	integer	R	System error of the last ESA call (0 = no error)

# CREW Manual

## *Examples:*

```
Sub test1
  on error resume next
  ESAHMI.ESAFILE.Open "filename","r"
  res = ESAHMI.Result
  if res <> 0 then
    syserr = ESAHMI.LastError
    strerr = ESAHMI.GetErrorMsg(syserr)
    ESAHMI.ESMsgbox "result=" & HEX(res) & " - " & syserr & " : " & strerr
  end if
  ...
```

```
Sub test2
  on error resume next
  ESAHMI.ESAFILE.Open "filename","r"
  if err.number <> 0 then
    syserr = ESAHMI.LastError
    strerr = ESAHMI.GetErrorMsg(syserr)
    ESAHMI.ESMsgbox "result=" & HEX(err.number) & " - " & syserr & " : " & strerr
  end if
  ...
```

# CREW Manual

## ESAALARMMGR

ESAALARMMGR provides access to alarm management.

### Methods - ESAALARMMGR

If the `UserName` is an empty string "", the current username is used.

If the `StationName` is an empty string "", the name of the local station is used.

#### **AlarmOn( AlarmName, UserName, StationName )**

`AlarmName` (string) Name of the alarm to be reported.

`UserName` (string) The name of the user reporting the alarm.

`StationName` (string) The name of the station reporting the alarm.

This reports the specified alarm.

**RETURN** (integer): the instance of the identified alarm (used by `AckAlarm`)

#### **ClearAlarm( AlarmName, UserName, StationName )**

`AlarmName` (string) Name of the alarm to be deleted

`UserName` (string) The name of the user performing the action.

`StationName` (string) The name of the station that the action is executed from.

This deletes an alarm.

#### **AckAlarm( Instance, UserName, StationName )**

`Instance` (integer) Instance of the requested alarm (returned from `AlarmOn`)

`UserName` (string) The name of the user requesting the action.

`StationName` (string) The name of the station requesting the action.

This recognises a single alarm instance.

# CREW Manual

## **AckInstances( AlarmName, UserName, StationName )**

AlarmName (string) Name of the alarm to be recognised.

UserName (string) The name of the user requesting the action.

StationName (string) The name of the station requesting the action.

This recognises all of the instances of the specified alarm.

## **AckGroup( GroupName, UserName, StationName )**

GroupName (string) Name of the alarm group.

UserName (string) The name of the user requesting the action.

StationName (string) The name of the station requesting the action.

This recognises all of the alarms of a specified group.

## **AckGlobal( UserName, StationName )**

UserName (string) The name of the user requesting the action.

StationName (string) The name of the station requesting the action.

This recognises all activated alarms.

## **IsAlarmOn( AlarmName )**

AlarmName (string) Name of the alarm to be controlled

RETURN (boolean): TRUE if the specified alarm is ON

## **AlarmsExport( PathName )**

Pathname (string) Full file path name.

This exports the description of all of the alarms that are active at that time to a file.

RETURN (integer): the number of exported archives.

# CREW Manual

## AlarmsExportLocal( PathName, [suspensive] )

Pathname (string) Full file path name.

If the Pathname is an empty string (""), a dialog box appears so that the user can select a path.

suspensive (boolean-optional) TRUE = suspensive dialog box

FALSE = no non-suspensive dialog box

This exports the description of all of the alarms that are active at that time to a file.

RETURN (integer): the number of exported archives.

## HistoryExport( PathName )

Pathname (string) Full file path name.

This exports the description of all of the alarms recorded in the chronology to a file.

RETURN (integer): the number of exported archives.

## HistoryExportLocal( PathName, [suspensive] )

Pathname (string) Full file path name.

If the Pathname is an empty string (""), a dialog box appears so that the user can select a path.

suspensive (boolean-optional) TRUE = suspensive dialog box

FALSE = no non-suspensive dialog box

This exports the description of all of the alarms recorded in the chronology to a file.

RETURN (integer): the number of exported archives.

## HistoryDelete()

This deletes the alarm history.

# CREW Manual

## HistoryFlush()

This saves the data of the alarm history on the disc.

## AlarmsPrint()

## AlarmsPrintLocal()

This prints the currently active alarms.

RETURN (integer): The number of printed alarms.

## HistoryPrint()

## HistoryPrintLocal()

This prints the alarms recorded in the history.

RETURN (integer): The number of printed alarms.



# CREW Manual

## ESACOM( [index] )

index (integer-optional) port index (1 .. 4 - default=1)

ESACOM provides access to COM serial ports.

Up to four COM ports are available at the same time.

**There is no relation between port index and the COM port number.**

The following calls are equivalent:

ESAHMI.ESACOM.Open 2,9600,8,0,0

ESAHMI.ESACOM().Open 2,9600,8,0,0

ESAHMI.ESACOM(1).Open 2,9600,8,0,0

## Methods - ESACOM( [index] )

**Open( Port, Baud, DataBits, Parity, StopBits, [Rts], [Cts] )**

Port (integer) port number (1=COM1, ...)

Baud (integer) baud rate

DataBits (integer) 4 .. 8

Parity (integer) 0=NONE, 1=ODD, 2=EVEN, 3=MARK, 4=SPACE

StopBits (integer) 0=1, 1=1.5, 2=2

Rts (integer -optional) 0=DISABLE (default), 1=ENABLE, 2=HANDSHAKE, 3=TOGGLE

Cts (integer -optional) 0 .. 1 (default=0)

Open a serial port for reading/writing.

Up to four ports can be open at the same time.

**Close()**

This closes an open serial port.

**IsOpen()**

# CREW Manual

This checks whether the port is open.

RETURN (boolean): TRUE if the port is open, otherwise FALSE

## IsData()

Check whether there is any available data.

RETURN (integer): 0 = no byte in queue

>0 = number of bytes in queue

## WriteByte( byte )

byte (integer) byte value to be written

This writes a single byte on the serial port.

## WriteStr( text )

text (string) The text to be written

This writes a text string on the serial port.

## ReadByte()

This reads a single byte on the serial port.

RETURN (integer): the read byte.

## Clear()

This deletes/resets the serial port.

# CREW Manual

## Escape( code )

- code (integer)
- 1 Simulates XOFF received
  - 2 Simulates XON received
  - 3 Sets RTS high
  - 4 Sets RTS low
  - 5 Sets DTR high
  - 6 Sets DTR low
  - 7 Resets the device if possible
  - 8 Sets the device's interruption line
  - 9 Deletes the device's interruption line

This runs the extended function on the serial port.

## SetRTS()

This sets the RTS signal (request-to-send).

## ClrRTS()

This deletes the RTS signal (request-to-send).

## GetCTS()

This controls the CTS signal (clear-to-send).

Return (boolean): TRUE = HI signal, FALSE = LOW signal

## GetDSR()

This controls the DSR signal (data-set-ready).

Return (boolean): TRUE = HI signal, FALSE = LOW signal

# CREW Manual

## GetRing()

This controls the loop closing signal of the communication line.

Return (boolean): TRUE = HI signal, FALSE = LOW signal

## GetRLSD()

This controls the RLSD signal (receive-line-signal-detect).

Return (boolean): TRUE = HI signal, FALSE = LOW signal

## Examples

```
set ser = ESAHMI.ESACOM
ser.Open 1, 19200, 8, 0, 0
ser.WriteByte 97          ` a
ser.WriteByte &h61       ` a
ser.WriteByte ASC("a")   ` a
ser.Close
```

```
set ser1 = ESAHMI.ESACOM(1)
set ser2 = ESAHMI.ESACOM(2)
ser1.Open 3, 9600, 8, 0, 0
ser2.Open 4, 9600, 8, 0, 0
ser1.WriteString "com1"
ser2.WriteString "com2"
ser1.Close
ser2.Close
```

```
s = ""
n = ser.IsData
for i=1 to n
  a = ser.ReadByte
  s = s & chr(a)
next
```

# CREW Manual

## ESACTRL

ESACTRL provides access to the visual objects displayed on an open page.

### ESACTRL ( page, controlname )

page (string/ integer) page name/id

controlname (string) control name

*Example 1 (slower):*

```
v1 = ESAHMI.ESACTRL("page1", "txt1").Left  
v2 = ESAHMI.ESACTRL("page1", "txt1").Top  
ESAHMI.ESACTRL("page1", "txt1").Left = v1 + 20  
ESAHMI.ESACTRL("page1", "txt1").Top = v2 + 10
```

*Example 2 (faster, recommended):*

```
set ctrl = ESAHMI.ESACTRL("page1", "txt1")  
v1 = ctrl.Left  
v2 = ctrl.Top  
ctrl.Left = v1 + 20  
ctrl.Top = v2 + 10
```

# CREW Manual

## Methods - ESACTRL

### SetRangeColor ( index, normalcolor, gradientcolor )

index            (integer)    range index (1..32)  
normalcolor    (integer)    RGB color  
gradientcolor   (integer)    RGB color

This sets a range of colour in BAR Control.

### GetRangeColor ( index, normalflag )

index            (integer)    range index (1..32)  
normalflag     (boolean)   TRUE=normalcolor, FALSE=gradientcolor

This obtains a BAR Control range of colour.

RETURN (integer): RGB range color (BAR Control).

### SetRangeValue ( index, value )

index    (integer) range index (1..32)  
value    (double) floating-point value

This sets a range of colour in BAR Control.

### GetRangeValue ( index )

index    (integer)    range index (1..32)

This obtains a BAR Control range of colour.

RETURN (double): range value (BAR Control).

# CREW Manual

## **SetMoveState ( index, left, top, angle, time )**

index (integer) step index (1..16)  
left (double) X coordinate  
top (double) Y coordinate  
angle (double) angle of rotation (degrees)  
time (double) step time (msec)

This sets a movement step.

## **GetMoveStateLeft ( index )**

## **GetMoveStateTop ( index )**

## **GetMoveStateAngle ( index )**

## **GetMoveStateTime ( index )**

index (integer) step index (1..16)

This obtains the data of a movement step.

RETURN (double/integer): movement step data

## **SetImage ( index, imagename )**

index (integer) image index (1...)  
imagename (string) full image path name

This sets an image in Image Control.

## **GetImage ( index )**

index (integer) image index (1...)

This obtains an image from Image Control.

RETURN (string): full image path name

# CREW Manual

## SetText ( index, text )

index (integer) text index (1..)

text (string) text

This sets a text in Text Control.

## GetText ( index )

index (integer) text index (1..)

This obtains a text from Text Control.

RETURN (string): text string

## SetPointCoord ( index, normalcolor, gradientcolor )

index (integer) range index (1..16)

x (double) X coordinate

y (double) Y coordinate

This sets the coordinates of an Interlines Control.

## SetTrendTraceMinX ( index, value )

## SetTrendTraceMaxX ( index, value )

## SetTrendTraceMinY ( index, value )

## SetTrendTraceMaxY ( index, value )

index (integer) trace index (1..32)

value (double) floating-point value

This sets the value range of a Trend Control.



# CREW Manual

GetTrendMinX ( index )

GetTrendMaxX ( index )

GetTrendMinY ( index )

GetTrendMaxY ( index )

index (integer) trace index (1..32)

This obtains the value range of a Trend Control.

RETURN (double): range value (BAR Control).

TraceImportLocal( Pathname, [suspensive] )

Pathname (string) full name of file path.

If the Pathname is an empty string (“”), a dialog box appears so that the user can select a path.

suspensive (boolean-optional) TRUE = suspensive dialog box

FALSE = no non-suspensive dialog box

This imports detection data from a text file.

# CREW Manual

## Methods Table - ESACTRL

	SHAPE	COMPLEX	LINE	RECTANGLE	ELLIPSE	PATH	IMAGE	TEXT	PIPE	BUTTON	INTERLINE	BAR	SLIDER	METER	TREND	GRID
SetImage							*									
GetImage							*									
SetText								*								
GetText								*								
SetPointCoord											*					
SetRangeColor												*				
GetRangeColor												*				
SetRangeValue												*				
GetRangeValue												*				
SetMoveState	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
GetMoveStateLeft	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
GetMoveStateTop	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
GetMoveStateAngle	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
GetMoveStateTime	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

# CREW Manual

## Property Table - ESACTRL

Name	Type	Read-Write
AlphaLevel	byte	RW
AnimationTime	integer	RW
BarContinuousColors	boolean	RW
Brightness	double	RW
ClipMode	integer	RW
ControlType	integer	R
CornerDownLeft	double	RW
CornerDownRight	double	RW
CornerUpLeft	double	RW
CornerUpRight	double	RW
Disabled	boolean	RW
FillAreaBlinking	boolean	R
FillColorBlink	integer	RW
FillColorBlink Grad	integer	RW
FillColorGrad	integer	RW
FillColorNormal	integer	RW
Flashing	boolean	R
GridBorderColor	integer	RW
GridBorderColorGrad	integer	RW
GridCellAlternateColor	integer	RW
GridCellHeaderColor	integer	RW
GridCellNormalColor	integer	RW
GridCellSelectColor	integer	RW
GridEmptyAlpha	byte	RW
GridEmptyColor	integer	RW
GridTextHeaderColor	integer	RW
GridTextNormalColor	integer	RW
Height	double	RW
HorizontalAlignment	integer	RW
Hue	Double	RW
Left	double	RW
LineBlinking	boolean	R
LineColorBlink	integer	RW
LineColorBlinkGrad	integer	RW
LineColorGrad	integer	RW
LineColorNormal	integer	RW
LineSize	integer	RW
MoveOn	boolean	RW
RotationAngle	double	RW
RotationCenterX	double	RW
RotationCenterY	double	RW
Saturation	double	RW
ScaleValueMaximum	double	RW

ScaleValueMinimum	double	RW
ShowLayerDisabled	Boolean	RW
ShowLayerInvalid	Boolean	RW
ShowLayerOffline	Boolean	RW
ShowLayerProtected	Boolean	RW
StartAngle	double	RW
StretchMode	integer	RW
SweepAngle	double	RW
TextContentLength	integer	R
Top	double	RW
TrendAreaInColor	integer	RW
TrendAreaInColorGrad	integer	RW
TrendAreaOutColor	integer	RW
TrendAreaOutColorGrad	integer	RW
TrendBorderColor	integer	RW
TrendBorderColorGrad	integer	RW
TrendGridMode	boolean	RW
TrendHasArea	boolean	RW
TrendHasLines	boolean	RW
TrendHasMarkers	boolean	RW
TrendIsDigitalLine	boolean	RW
TrendIsHistogram	boolean	RW
TrendLabelsColor	integer	RW
TrendTimeMode	integer	RW
TrendTimeSpan	integer	RW
TrendTouchLeft	double	RW
TrendTouchRight	double	RW
TrendTouchTop	double	RW
TrendTouchBottom	double	RW
Value	variant	RW
ValueType	integer	R
VerticalAlignment	integer	RW
Visible	boolean	RW
VoidBlinkColor	integer	RW
VoidBlinkColorGrad	integer	RW
VoidBlinking	boolean	R
VoidColor	integer	RW
VoidColorGrad	integer	RW
Width	double	RW
X1	double	RW
X2	double	RW
Y1	double	RW
Y2	double	RW

# CREW Manual

	SHAPE	COMPLEX	LINE	RECTANGLE	ELLIPSE	PATH	IMAGE	TEXT	PIPE	BUTTON	INTERLINE	BAR	SLIDER	METER	TREND	GRID
ControlType	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Visible	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Flashing	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Disabled	*						*	*	*	*	*	*	*	*	*	*
ValueType	*						*	*	*	*	*	*	*	*	*	*
Value	*						*	*	*	*	*	*	*	*	*	*
Left	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Top	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Width	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Height	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
RotationAngle	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
RotationCenterX	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
RotationCenterY	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
MoveOn	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
AlphaLevel			*	*	*	*	*	*	*	*	*	*	*	*	*	*
FillColorNormal				*	*	*	*	*	*	*	*	*	*	*	*	*
FillColorGrad				*	*	*	*	*	*	*	*	*	*	*	*	*
FillAreaBlinking				*	*	*	*	*	*	*	*	*	*	*	*	*
FillColorBlink				*	*	*	*	*	*	*	*	*	*	*	*	*
FillColorBlinkGrad				*	*	*	*	*	*	*	*	*	*	*	*	*
LineSize			*	*	*	*	*	*	*	*	*	*	*	*	*	*
LineColorNormal			*	*	*	*	*	*	*	*	*	*	*	*	*	*
LineColorGrad			*	*	*	*	*	*	*	*	*	*	*	*	*	*
LineBlinking			*	*	*	*	*	*	*	*	*	*	*	*	*	*
LineColorBlink			*	*	*	*	*	*	*	*	*	*	*	*	*	*
LineColorBlinkGrad			*	*	*	*	*	*	*	*	*	*	*	*	*	*
HorizontalAlignment							*	*								
VerticalAlignment							*	*								
AnimationTime	*						*	*								
X1			*													
X2			*													
Y1			*													
Y2			*													
CornerUpLeft				*												
CornerUpRight				*												
CornerDownLeft				*												
CornerDownRight				*												
SweepAngle					*											
StartAngle					*											
ClipMode				*												

	SHAPE	COMPLEX	LINE	RECTANGLE	ELLIPSE	PATH	IMAGE	TEXT	PIPE	BUTTON	INTERLINE	BAR	SLIDER	METER	TREND	GRID
StretchMode							*									
TextContentLength								*								
VoidColor									*			*				
VoidColorGrad									*			*				
VoidBlinking								*				*				
VoidBlinkColor								*				*				
VoidBlinkColorGrad								*				*				
BarContinuousColors												*				
TrendBorderColor															*	
TrendBorderColorGrad															*	
TrendAreaInColor															*	
TrendAreaInColorGrad															*	
TrendAreaOutColor															*	
TrendAreaOutColorGrad															*	
TrendLabelsColor															*	
TrendGridMode															*	
TrendTimeMode															*	
TrendTimeSpan															*	
TrendHasMarkers															*	
TrendHasLines															*	
TrendHasArea															*	
TrendIsDigitalLine															*	
TrendIsHistogram															*	
GridBorderColor															*	
GridBorderColorGrad															*	
GridEmptyColor															*	
GridEmptyAlpha															*	
GridTextNormalColor															*	
GridTextHeaderColor															*	
GridCellNormalColor															*	
GridCellAlternateColor															*	
GridCellSelectColor															*	
GridCellHeaderColor															*	
ScaleValueMinimum												*	*	*		
ScaleValueMaximum												*	*	*		
Brightness	*															
Hue	*															
Saturation	*															
ShowLayerDisabled	*						*	*		*	*	*	*	*	*	
ShowLayerOffline	*						*	*	*	*	*	*	*	*	*	
ShowLayerInvalid	*						*	*	*	*	*	*	*	*	*	
ShowLayerProtected	*						*	*	*	*	*	*	*	*	*	*

# CREW Manual

## ESAETH

ESAETH provides access to the Ethernet ports.

Sub-Objects:

TCPCLIENT

TCPSERVER

UDP

# CREW Manual

## ESAETH.TCPCLIENT

TCPCLIENT provides client management in the TCP protocol.

### Methods

#### Open( IPAddress, Port )

IPAddress (string) Server IP Address (“xxx.xxx.xxx.xxx”)

Port (integer) Server Port number

This opens a TCP Client socket and connects it to a server specified by address and port.

RETURN (integer): TCP Client socket identifier.

#### Close( SocketId )

SocketId (integer) TCP Client socket identifier

This closes the TCP Client socket.

#### IsData( SocketId )

SocketId (integer) TCP Client socket identifier

Check whether there is any available data.

RETURN (boolean): 0 = no byte in queue  
<>0 = number of bytes in queue

#### GetAddress( SocketId, IPAddress )

SocketId (integer) TCP Client socket identifier

IPAddress (string - output) Client IP Address

This obtains the port and address of the specified Client.

RETURN (integer): Client Port.

# CREW Manual

## **GetServerAddress( SocketId, IPAddress, serverPort )**

SocketId (integer) TCP Client socket identifier

IPAddress (integer - output) Server IP Address

This obtains the port and address of the server that the specified Client is connected to.

RETURN (integer): Server Port.

## **WriteString( SocketId, String )**

SocketId (integer) TCP Client socket identifier

String (string) String to send

This sends a string of characters to the TCP server.

RETURN (integer): The number of written characters.

## **WriteBuffer( SocketId, Buffer, NumBytes )**

SocketId (integer) TCP Client socket identifier

Buffer (array of bytes) Binary data to be sent

NumBytes (integer) Number of bytes to be sent

This sends binary data to the TCP server.

RETURN (integer): The number of written bytes.

## **ReadString( SocketId, String, NumChars )**

SocketId (integer) TCP Client socket identifier

String (string - output) Received string

NumChars (integer) Number of characters to be read (0 = read all available)

This receives a string of characters from a TCP Client socket.

RETURN (integer): The number of read characters.

# CREW Manual

## ReadBuffer( SocketId, Buffer, NumBytes )

SocketId (integer) TCP Client socket identifier

Buffer (array of bytes - output) Received data

NumBytes (integer) Number of bytes to read (0 = read all available)

This receives binary data from a TCP Client socket.

RETURN (integer): The number of read bytes.

### Example 1

Send and receive a data string:

```
serverIPAddress = "192.168.1.20"
```

```
serverIPPort = 20000
```

```
TCPClientID = ESAHMI.ESAETH.TCPCClient.OPEN( serverIPAddress, serverIPPort )
```

```
result = ESAHMI.ESAETH.TCPCClient.ISDATA(TCPCClientID)
```

```
if result <> 0 then
```

```
result =
```

```
ESAHMI.ESAETH.TCPCClient.READSTRING(TCPCClientID,stringReceived,10)
```

```
ESAHMI.ESAMsgbox "Received " + CStr(result) + " character(s) from " +
```

```
serverIPAddress + ":" + CStr(serverIPPort) + " " +
```

```
stringReceived
```

```
end if
```

```
result = ESAHMI.ESAETH.TCPCClient.WRITESTRING(TCPCClientID,"example")
```

```
ESAHMI.ESAETH.TCPCClient.CLOSE TCPCClientID
```



# CREW Manual

## Example 2

Send and receive a byte matrix:

```
serverIPAddress = "192.168.1.20"
```

```
serverIPPort = 20000
```

```
TCPCClientID = ESAHMI.ESAETH.TCPCClient.OPEN( serverIPAddress, serverIPPort )
```

```
result = ESAHMI.ESAETH.TCPCClient.ISDATA(TCPCClientID)
```

```
if result <> 0 then
```

```
Dim dataReceived
```

```
bytesRead =
```

```
ESAHMI.ESAETH.TCPCClient.READBUFFER(TCPCClientID,dataReceived,10)
```

```
For i = LBound(dataReceived) To UBound(dataReceived)
```

```
ESAHMI.ESAMsgBox "Bytes read (" & CStr(bytesRead) & ") from " &  
CStr(serverIPAddress) & ":" & CStr(serverIPPort) & vbCrLf & dataReceived(i)
```

```
Next
```

```
end if
```

```
'create an array of 13 bytes
```

```
Dim dataToSend
```

```
ReDim dataToSend(12)
```

```
'initialize array of bytes
```

```
For kk = LBound(dataToSend) To UBound(dataToSend)
```

```
dataToSend(kk) = CByte(kk)
```

```
Next
```

```
bytesWritten =
```

```
ESAHMI.ESAETH.TCPCClient.WRITEBUFFER(TCPCClientID,dataToSend,10)
```

```
ESAHMI.ESAMsgbox "Written " + CStr(result) + " byte(s)"
```

```
ESAHMI.ESAETH.TCPCClient.CLOSE TCPCClientID
```

# CREW Manual

## ESAETH.TCPSERVER

TCPSERVER provides Server management in the TCP protocol.

### Methods:

#### Open( IPAddress, Port )

IPAddress (string) Server IP Address (“xxx.xxx.xxx.xxx”)

Port (integer) Server port number

This opens a TCP Server socket.

RETURN (integer): TCP Server socket identifier.

#### Close( SocketId )

SocketId (integer) TCP Server socket identifier

This closes a TCP Server socket.

#### CloseClient( SocketId )

SocketId (integer) TCP Server socket identifier

IPAddress (string) Client IP address (“xxx.xxx.xxx.xxx”)

Port (integer) Client Port Number

This interrupts the connection between the Specified Client and the Server.

#### IsData( SocketId, IPAddress, Port )

SocketId (integer) TCP Server socket identifier

IPAddress (string - output) Client IP address (“xxx.xxx.xxx.xxx”)

Port (integer - output) Client Port Number

Check whether there is any available data.

RETURN (boolean): 0 = no byte in queue

<>0 = number of bytes in queue

# CREW Manual

## **IsDataClient( SocketId, IPAddress, Port )**

SocketId (integer) TCP Server socket identifier

IPAddress (string) Client IP address (“xxx.xxx.xxx.xxx”)

Port (integer) Client Port Number

This checks whether there is any available data from the specified client

RETURN (boolean): 0 = no byte in queue

<>0 = number of bytes in queue

## **GetAddress( SocketId, IPAddress )**

SocketId (integer) TCP Server socket identifier

IPAddress (integer - output) Server IP Address

This obtains the port and address of the Server.

RETURN (integer): Server Port.

## **GetClientAddress( SocketId, ClientID, IPAddress )**

SocketId (integer) TCP Server socket identifier

ClientId (integer) Index used to number the client connected to the server.

This identifies the Nth connected client.

Range: 1 to NumberOfClientConnected

IPAddress (integer - output) Server IP Address

This obtains the Port number and IP address of the Nth client connected to the server.

RETURN (integer): Client Port.

## **NumberOfClientsConnected( SocketId, [Message] )**

SocketId (integer) TCP Server socket identifier

Message (array of bytes - optional) Binary message to send to the client

# CREW Manual

This obtains the number of clients connected to the server.

RETURN (integer): The number of clients connected to the server.

## **WriteString( SocketId, String, IPAddress, Port )**

SocketId (integer) TCP Server socket identifier

String (string) String to send

IPAddress (string) Client IP address (“xxx.xxx.xxx.xxx”)

Port (integer) Client Port Number

This sends a string of characters to the specified address and to the client port.

RETURN (integer): The number of written characters.

## **WriteBuffer( SocketId, Buffer, NumBytes, IPAddress, Port )**

SocketId (integer) TCP Server socket identifier

Buffer (array of bytes) Binary data to be sent

NumBytes (integer) Number of bytes to be sent

IPAddress (string) Client IP address (“xxx.xxx.xxx.xxx”)

Port (integer) Client Port Number

This sends binary data to the specified address and to the client port.

RETURN (integer): The number of written bytes.

# CREW Manual

## **ReadString( SocketId, IPAddress, Port, String, NumChars )**

SocketId (integer) TCP Server socket identifier

IPAddress (string - output) Client IP address (“xxx.xxx.xxx.xxx”)

Port (integer - output) Client Port Number

String (string - output) Received string

NumChars (integer) Number of characters to be read (0 = read all available)

This receives a string of characters from a TCP Server socket.

RETURN (integer): The number of read characters.

## **ReadStringClient( SocketId, IPAddress, Port, String, NumChars )**

SocketId (integer) TCP Server socket identifier

IPAddress (string) Client IP address (“xxx.xxx.xxx.xxx”)

Port (integer) Client Port Number

String (string) Received string

NumChars (integer) Number of characters to be read (0 = read all available)

This receives a string of characters from a specified address and client port.

RETURN (integer): The number of read characters.

## **ReadBuffer( SocketId, IPAddress, Port, Buffer, NumBytes )**

SocketId (integer) TCP Server socket identifier

IPAddress (string - output) Client IP address (“xxx.xxx.xxx.xxx”)

Port (integer - output) Client Port Number

Buffer (array of bytes - output) Received data

NumBytes (integer) Number of bytes to read (0 = read all available)

This receives binary data from a TCP Server socket.

RETURN (integer): The number of read bytes.

## **ReadBufferClient( SocketId, IPAddress, Port, Buffer, NumBytes )**

ReadBufferClient( SocketId, IPAddress, Port, Buffer, NumBytes )

# CREW Manual

SocketId (integer) TCP Server socket identifier

IPAddress (string) Client IP address ("xxx.xxx.xxx.xxx")

Port (integer) Client Port Number

Buffer (array of bytes) Received data

NumBytes (integer) Number of bytes to read (0 = read all available)

This receives binary data from a specified address and client port.

RETURN (integer): The number of read bytes.

## Example 1

Send and receive a data string:

```
serverIPAddress = "192.168.1.20"
```

```
serverIPPort = 20000
```

```
socketIDOpened = ESAHMI.ESAETH.TCPServer.OPEN( serverIPAddress,  
serverIPPort )
```

```
result = ESAHMI.ESAETH.TCPServer.ISDATA( socketIDOpened,  
senderClientIPAddress, senderClientPort )
```

```
if result <> 0 then
```

```
result = ESAHMI.ESAETH.TCPServer.READSTRING( socketIDOpened,  
senderClientIPAddress, senderClientPort,
```

```
stringReceived, 10 )
```

```
ESAHMI.ESAMsgbox "Received " + CStr(result) + " character(s) from " +  
senderClientIPAddress + ":" + CStr(senderClientPort) + " " + stringReceived
```

```
End if
```

```
numOfClientConnected =
```

```
ESAHMI.ESAETH.TCPServer.NUMBEROFCLIENTSCONNECTED(  
socketIDOpened )
```

```
if numOfClientConnected > 0 then portClient =
```

```
ESAHMI.ESAETH.TCPServer.GETCLIENTADDRESS( socketIDOpened,
```

# CREW Manual

```

numOfClientConnected, ipAddressClient )
result = ESAHMI.ESAETH.TCPServer.WRITESTRING( socketIDOpened,
"example", ipAddressClient, portClient )
ESAHMI.ESAMsgbox "Written " + CStr(result) + " character(s) to " +
ipAddressClient + ":" + CStr(portClient)
else
ESAHMI.ESAMsgbox "No client connected"
end if
ESAHMI.ESAETH.TCPServer.CLOSE socketIDOpened

```

## Example 2

Send and receive a byte matrix:

```

serverIPAddress = "192.168.1.20"
serverIPPort = 20000
socketIDOpened = ESAHMI.ESAETH.TCPServer.OPEN( serverIPAddress,
serverIPPort )
result = ESAHMI.ESAETH.TCPServer.ISDATA( socketIDOpened,
senderClientIPAddress, senderClientPort )
if result <> 0 then
Dim dataReceived
bytesRead = ESAHMI.ESAETH.TCPServer.READBUFFER( socketIDOpened,
senderClientIPAddress, senderClientPort, dataReceived, 10 )
For i = LBound(dataReceived) To UBound(dataReceived)
ESAHMI.ESAMsgBox "Bytes read (" & CStr(bytesRead) & ") from " &
CStr(senderClientIPAddress) & ":" & CStr(senderClientPort) & vbCrLf &
dataReceived(i)
Next
End If
'prepare array of bytes
Dim message

```

# CREW Manual

ReDim message(0)

'send the byte 0 (zero) to the clients to get the number of the active  
'connections

message(0) = CByte(0)

numOfClientConnected =

ESAHMI.ESAETH.TCPServer.NUMBEROFCLIENTSCONNECTED(  
socketIDOpened, message )

if numOfClientConnected > 0 then

portClient = ESAHMI.ESAETH.TCPServer.GETCLIENTADDRESS(  
socketIDOpened, numOfClientConnected, ipAddressClient )

'create an array of 11 bytes

Dim dataToSend

ReDim dataToSend(10)

' initialize array of bytes

For kk = LBound(dataToSend) To UBound(dataToSend)

dataToSend(kk) = CByte(kk)

Next

bytesWritten = ESAHMI.ESAETH.TCPServer.WRITEBUFFER( socketIDOpened,  
dataToSend, 10, ipAddressClient,

portClient )

ESAHMI.ESAMsgbox "Written " + CStr(result) + " byte(s) to " + ipAddressClient  
+ ":" + CStr(portClient)

end if

ESAHMI.ESAETH.TCPServer.CLOSE socketIDOpened



# CREW Manual

## ESAETH.UDP

UDP provides management of the UDP protocol.

### Methods:

#### Open( IPAddress, Port )

IPAddress (string) IP Address (“xxx.xxx.xxx.xxx”)

Port (integer) Port number

This opens a UDP socket.

RETURN (integer): UDP socket identifier.

#### Close( SocketId )

SocketId (integer) UDP socket identifier

This closes a UDP socket.

#### IsData( SocketId )

SocketId (integer) UDP socket identifier

Check whether there is any available data.

RETURN (boolean): 0 = no byte in queue

<>0 = number of bytes in queue

# CREW Manual

## **GetAddress( SocketId, IPAddress )**

SocketId (integer) UDP socket identifier

IPAddress (integer - output) IP Address

This obtains the port and address of a specified UDP socket.

RETURN (integer): The Port number.

## **WriteString( SocketId, String, IPAddress, Port )**

SocketId (integer) UDP socket identifier

String (string) String to send

IPAddress (string) IP Address (“xxx.xxx.xxx.xxx”)

Port (integer) Port number

This sends a string of characters to a specified address and port.

RETURN (integer): The number of written characters.

## **WriteBuffer( SocketId, Buffer, NumBytes, IPAddress, Port )**

SocketId (integer) UDP socket identifier

Buffer (array of bytes) Binary data to be sent

NumBytes (integer) Number of bytes to be sent

IPAddress (string) IP Address (“xxx.xxx.xxx.xxx”)

Port (integer) Port number

This sends binary data to a specified address and port.

RETURN (integer): The number of written bytes.

# CREW Manual

## ReadString( SocketId, IPAddress, Port, String, NumChars )

SocketId (integer) UDP socket identifier

IPAddress (string - output) IP Address ("xxx.xxx.xxx.xxx")

Port (integer - output) Port number

String (string - output) Received string

NumChars (integer) Number of characters to be read (0 = read all available)

Receive a characters string from an UDP socket.

RETURN (integer): The number of read characters.

## ReadBuffer( SocketId, IPAddress, Port, Buffer, NumBytes )

SocketId (integer) UDP socket identifier

IPAddress (string - output) IP Address ("xxx.xxx.xxx.xxx")

Port (integer - output) Port number

Buffer (array of bytes - output) Received data

NumBytes (integer) Number of bytes to read (0 = read all available)

This receives binary data from a UDP socket.

RETURN (integer): The number of read bytes.

### Example 1

Send and receive a data string:

```
socketIDOpened = ESAHMI.ESAETH.UDP.OPEN("192.168.1.20", 10000)
```

```
result = ESAHMI.ESAETH.UDP.ISDATA(socketIDOpened)
```

```
if result <> 0 then
```

```
result = ESAHMI.ESAETH.UDP.READSTRING( socketIDOpened,
```

```
IPSenderAddress,
```

```
IPSenderPort,
```

# CREW Manual

```

stringReceived,10 )
ESAHMI.ESAMsgbox "Received " + CStr(result) + " character(s) from " +
CStr(IPSenderAddress) + ":" + CStr(IPSenderPort) + " " +
stringReceived
end if
result = ESAHMI.ESAETH.UDP.WRITESTRING( socketIDOpened,
"example",
"192.168.1.255",
20000 )
ESAHMI.ESAMsgbox "Written " + CStr(result) + " character(s)"
ESAHMI.ESAETH.UDP.CLOSE socketIDOpened

```

## Example 2

Send and receive a byte matrix:

```

socketIDOpened = ESAHMI.ESAETH.UDP.OPEN("192.168.1.20", 10000)
result = ESAHMI.ESAETH.UDP.ISDATA(socketIDOpened)
if result <> 0 then
Dim dataReceived
bytesRead = ESAHMI.ESAETH.UDP.READBUFFER( socketIDOpened,
IPSenderAddress, IPSenderPort, dataReceived,
10 )
For i = LBound(dataReceived) To UBound(dataReceived)
ESAHMI.ESAMsgBox "Bytes read (" & CStr(bytesRead) & ") from " &
CStr(SenderIP) & ":" & CStr(SenderPort) & vbCrLf &
dataReceived(i)
Next
end if
'create an array of 6 bytes
Dim dataToSend

```

# CREW Manual

ReDim dataToSend(5)

'initialize array of bytes

For kk = LBound(dataToSend) To UBound(dataToSend)

dataToSend(kk) = CByte(kk)

Next

bytesWritten = ESAHMI.ESAETH.UDP.WRITEBUFFER( socketIDOpened,  
dataToSend, 10, "192.168.1.255", 20000 )

ESAHMI.ESAMsgbox "Written " + CStr(bytesWritten) + " byte(s)"

ESAHMI.ESAETH.UDP.CLOSE socketIDOpened

# CREW Manual

## ESAFILE

ESAFILE provides access to the file system. This makes it possible to create and eliminate files and folders. This also makes it possible to read and write on binary and text files, both ASCII and Unicode.

# CREW Manual

## Methods - ESAFILE

### Methods:

#### Copy( from\_pathname, to\_pathname )

From\_pathname (string) Complete file arrival path

To\_pathname (string) Complete file destination path

Copy an existing file to a new file.

#### Delete( pathname )

pathname (string) Full file path name

This eliminates the specified file.

#### Rename( old\_pathname, new\_pathname )

old\_pathname (string) Full path name of old file

new\_pathname (string) Full path name of new file

Renames an existing file.

#### Exists( pathname )

pathname (string) Full file path name

RETURN (boolean): TRUE = the file exists

FALSE = the file does NOT exist

# CREW Manual

## IsDirectory( pathname )

pathname (string) Full file path name

Note: The specified pathname must exist.

RETURN (boolean): TRUE = the pathname specifies a directory  
FALSE = the pathname does NOT specify a directory

## GetFileLen( pathname )

pathname (string) Full file path name

RETURN (integer): file size (byte)

## SetFileLen( pathname )

pathname (string) Full file path name

This changes the size of an existing file or creates a new file.

## AvailableSpace( pathname )

pathname (string) Full directory path name

RETURN (integer): the amount of free space on the storage unit  
that the specified directory is on (byte)



# CREW Manual

## MD( pathname )

pathname (string) Full directory path name

This creates a directory of the file system

## RD( pathname )

pathname (string) Full directory path name

This eliminates an existing empty directory

## FindFirst(pathname)

pathname (string) Full path name, that can include special characters,  
for example an asterisk (\*) or a question mark (?)

This searches in a directory file or subdirectory with a name that matches a given name

RETURN (string): the name of the found files or subdirectories.

An empty string if no other files have been found.

### Example:

```
v = ESAHMI.ESAFILE.FindFirst("D:\test\*.**")
```

```
v = ESAHMI.ESAFILE.FindFirst("D:\test\*.txt")
```

```
v = ESAHMI.ESAFILE.FindFirst("D:\test\esa.txt")
```

# CREW Manual

## FindNext()

This continues a search from a previous call in FindFirst

RETURN (string): the name of the found files or subdirectories.

An empty string if no other files have been found.

### Example:

```
v = ESAHMI.ESAFILE.FindFirst("D:\test\*.*.")
```

```
Do While v <> ""
```

```
    ESAHMI.ESAMsgbox v
```

```
    v = ESAHMI.ESAFILE.FindNext
```

```
Loop
```

# CREW Manual

## Stream Methods - ESAFILE

### Open( pathname, mode )

pathname (string) Full file path name

mode (string) Opening mode:

"r" Open to read.

fails. If there is no such file or it cannot be found, the call

"w" This opens an empty file for writing.

If the file exists, its contents are destroyed.

"a" This opens to write at the end of the file (add);  
if the file does not exist, it creates it.

"r+" This opens for both reading and writing.

The file must exist.

writing. "w+" This opens an empty file for both reading and

If the file exists, its contents are destroyed.

"a+" This opens for reading and for adding;

The operation of adding includes removing the

EOF marker

before writing new data on the file and the EOF

marker is restored

after writing is completed;

if the file does not exist, it creates it.

Unicode file (example:

"r+u")

This opens of file for reading/writing.

Up to 32 files can be open at the same time.

# CREW Manual

Example:

```
file = "D:\note.txt"  
ESAHMI.ESAFILE.Open file,"r+"  
b = ESAHMI.ESAFILE.ReadByte(file)  
ESAHMI.ESAMsgbox chr(b)  
ESAHMI.ESAFILE.WriteByte file,asc("a")  
ESAHMI.ESAFILE.Close file
```

## Close( pathname )

Pathname (string) Full file path name

This closes a file.

The file must have been previously opened with the Open method.

## Rewind( pathname )

Pathname (string) Full file path name

This repositions the pointer of the file at the beginning of the file.

The file must have been previously opened with the Open method.

## Commit( pathname )

Pathname (string) Full file path name

This clears a stream: this writes the contents of the associated buffer on the file.

The file must have been previously opened with the Open method.

Also see FileFlush properties.

## IsEOF( pathname )

# CREW Manual

IsEOF( pathname )

Pathname (string) Full file path name

This verifies the end of the file on a row.

This checks whether a reading operation tried to read past the end of the file.

**The file must have been previously opened with the Open method.**

RETURN (boolean): TRUE = the current position is the end of the file

Example:

```
f1 = "C:\file.txt"  
ESAHMI.ESAFILE.Open f1,"r"  
flag = true  
Do While flag  
v = ESAHMI.ESAFILE.ReadLine(f1,100)  
ESAHMI.ESAMsgbox v  
if ESAHMI.ESAFILE.IsEOF(f1) then  
flag = false  
end if  
Loop  
ESAHMI.ESAFILE.Close f1
```

# CREW Manual

## GetLen( pathname )

Pathname (string) Full file path name

This obtains the current length of an open file.

The stream is cleaned prior to calculation.

**The file must have been previously opened with the Open method.**

RETURN (integer): the current length of the file

## SetUnicode( pathname )

Pathname (string) Full file path name

This writes the header of the FFFEh unicode.

**The file must have been previously opened with the Open method.**

Example:

```
ESAHMI.ESAFILE.Open file,"w+u"
```

```
ESAHMI.ESAFILE.SetUnicode file
```

```
ESAHMI.ESAFILE.WriteByte file,asc("a")
```

# CREW Manual

## SkipUnicode( pathname )

Pathname (string) Full file path name

This skips the header of the FFFEh unicode.

The file must have been previously opened with the Open method.

Example:

```
ESAHMI.ESAFILE.Open file,"r+u"
```

```
ESAHMI.ESAFILE.SkipUnicode file
```

```
b = ESAHMI.ESAFILE.ReadByte(file)
```

## WriteByte( pathname, byte )

Pathname (string) Full file path name

Byte (integer) Byte to be written [0..255]

This writes a byte on a file.

The stream is cleaned according to the FileFlush properties.

The file must have been previously opened with the Open method.

# CREW Manual

Example:

```
ESAHMI.ESAFILE.WriteByte file,asc("a")
```

```
ESAHMI.ESAFILE.WriteByte file,10
```

**ReadByte( pathname )**

Pathname (string) Full file path name

This reads a byte from a file.

The stream is cleaned prior to reading.

**The file must have been previously opened with the Open method.**

RETURN (integer): the read byte

Example:

```
b = ESAHMI.ESAFILE.ReadByte(file)
```

**WriteStr( pathname, str )**

Pathname (string) Full file path name

str (string) String to be written



# CREW Manual

Writes a string on a file.

The stream is cleaned according to the FileFlush properties.

**The file must have been previously opened with the Open method.**

Example:

```
ESAHMI.ESAFILE.WriteStr file,"abcd"
```

ReadStr( pathname, count )

Pathname (string) Full file path name

count (integer) Maximum number of characters to be read

This reads a string from a text file. The number of actually read characters could be less than those counted in case of error or if the end of the file is reached before counting finishes.

The stream is cleaned prior to reading.

**The file must have been previously opened with the Open method.**

RETURN (string): the read string

# CREW Manual

Example:

```
s = ESAHMI.ESAFILE.ReadStr(file,10)
```

```
WriteStrIdx( pathname, offset, str )
```

Pathname (string) Full file path name

Offset (integer) File offset [0...]

str (string) String to be written

This writes a string on a file to a specified offset.

The stream is cleaned according to the FileFlush properties.

**The file must have been previously opened with the Open method.**

Example:

```
ESAHMI.ESAFILE.WriteStrIdx file,100,"abcd"
```

```
ReadStrIdx( pathname, offset, count )
```

Pathname (string) Full name of file path

Offset (integer) File offset [0...]

count (integer) Maximum number of characters to read

# CREW Manual

This reads a string from a file to a specified offset. The number of actually read characters could be less than those counted in case of error or if the end of the file is reached before counting finishes.

The stream is cleaned prior to reading.

**The file must have been previously opened with the Open method.**

Example:

```
s = ESAHMI.ESAFILE.ReadStrIdx(file,100,10)
```

ReadLine( pathname, count )

Pathname (string) Full name of file path

count (integer) Maximum number of characters to read

This reads a row from a text file. This reads the characters from the current position in the stream position at the first character of the new row, up to the end of the stream, or until the number of read characters is equal to the count, whatever comes first. The number of actually read characters may be less than the counted number. The character of the new row, if read, is not included in the string.

The stream is cleaned prior to reading.

**The file must have been previously opened with the Open method.**

# CREW Manual

Example:

flag = true

Do While flag

v = ESAHMI.ESAFILE.ReadLine(file,100)

ESAHMI.ESAMsgbox v

if ESAHMI.ESAFILE.IsEOF(file) then

ESAHMI.ESAMsgbox "End of file!"

flag = false

end if

Loop

## Properties - ESAFILE

Name	Type	Read-Write	Description
<b>FileCount</b>	integer	R	Number of currently opened files
<b>FileFlush</b>	boolean	RW	If TRUE, the file writing is direct. If FALSE, the file writing is buffered (default).

### Note:

The FileFlush property is set on FALSE by default, so as to prolong the life of the panel's memory card, minimising the number of written operations.

In any case, even of the FileFlush is FALSE, the file stream is cleared when one of the following methods is called: Commit(), GetLen(), ReadByte(), ReadStr(), ReadLine(), ReadStrIdx(), WriteStrIdx(), Rewind(), Close().

# CREW Manual

## ESAPAGEMGR

ESAPAGEMGR provides access to the pages of the project and to some UI properties.

# CREW Manual

## Methods - ESAPAGEMGR

### ShowPage( Page )

Page (string/ integer) page name/id

This shows the specified page.

Examples:

ESAHMI.ESAPAGEMGR.ShowPage "pagestart"

ESAHMI.ESAPAGEMGR.ShowPage 12

### SetPageColor( Page, Color )

Page (string/ integer) page name/id

Color (integer) RGB color

This changes the wallpaper colour of the specified page.

### GetPageColor( Page )

Page (string/ integer) page name/id

RETURN (integer): The wallpaper colour of the specified page

### GetPageWidth( Page )

Page (string/ integer) page name/id

RETURN (integer): The wallpaper colour of the specified page

# CREW Manual

## GetPageHeight( Page )

Page (string/ integer) page name/id

RETURN (integer): The height of the specified page (pixel)

## ShowPageNext ( )

ShowPageNextFull( )

ShowPageNextPopup( )

This shows the next page.

## ShowPagePrevious ( )

ShowPagePreviousFull ( )

ShowPagePreviousPopup ( )

This shows the previous page.

## ShowPageLast( )

This shows the entire screen of the page opened prior to the current one.

It maintains a collection of 32 old pages. It only works with full screen pages.

## ClosePopUp( Page )

Page (string/ integer) page name/id

This closes a popup page.

# CREW Manual

## ClosePopUpTop( )

This closes the top popup page.

## ClosePopUpAll( )

This closes all popup pages.

## ShowHelpPage( Page )

## ShowHelpFullscreen( )

## ShowHelpPopup( )

Page (string/ integer) page name/id

This shows the help page.

## CloseHelpPage( Page )

Page (string/ integer) page name/id

This closes the help page.

## CloseHelpPages( )

This closes all the help pages.

## LanguageSet( Language )

## LanguageNext( )

## LanguagePrevious()

Language (integer) language identifier [1...]

This changes the current language.



# CREW Manual

LanguageGet( )

RETURN (integer): This is the identifier of the current language [1...]

DisableInteraction( ShowSignal )

ShowSignal (boolean) if TRUE, the “Interaction Disabled” image is shown



This shows every user interaction (touch-screen, mouse, keyboard, etc.)

EnableInteraction()

This enables user interaction (touch-screen, mouse, keyboard, etc.)

IsPageOpen( Page )

Page (string/ integer) page name/id

RETURN (boolean): TRUE = the page is open

FALSE = the page is NOT open

# CREW Manual

## **GetPageName( PageID )**

PageID (integer) Page identifier

RETURN (string): This name of the specified page

## **GetPageld( PageName )**

PageName (string) Page name

RETURN (integer): The identifier of the specified page

## **GetFullScreenName()**

RETURN (string): The name of the full screen page

## **GetFullScreenId( )**

RETURN (integer): The identifier of the current full screen page

## **GetNumPopups( )**

RETURN (integer): The number of popup pages

## **GetPopupName(index)**

Index (integer) page index [0...]

RETURN (string): This name of the specified popup page

## **GetPopupId( index)**

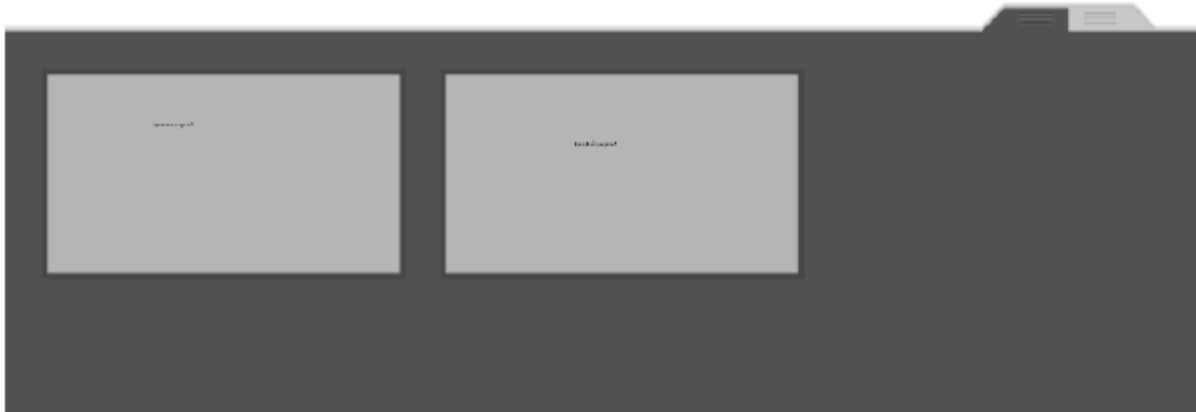
Index (integer) page index [0...]

RETURN (integer): The identifier of the specified popup page

# CREW Manual

## ShowRoadMap()

This shows the Roadmap page



## ShowPopupMap()

This shows the map of the popup pages

## ShowSequenceRoll()

This shows the Pages Sequence Roll



# CREW Manual

## ShowDateTimeBox( [suspensive] )

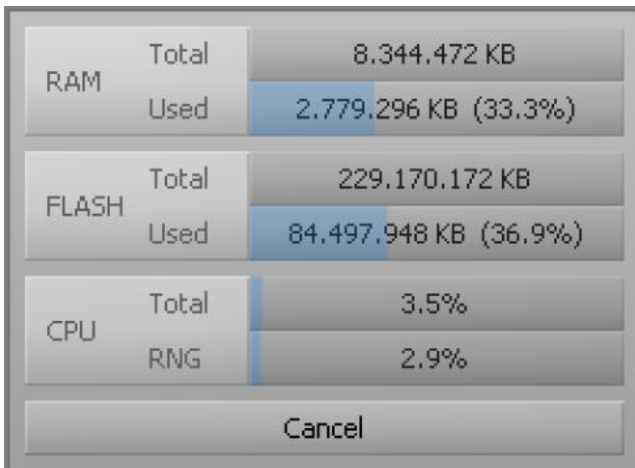
suspensive (boolean-optional) TRUE = suspensive, FALSE = non-suspensive

This shows the Date/Time box



## ShowResourceMonitorBox()

This shows a box containing some system data



# CREW Manual

## ShowCalculatorBox()

This shows the Calculator box



# CREW Manual

## ESAPRN

ESAPRN provides access to the printer.

# CREW Manual

## Methods - ESAPRN

### Start( UserFlag )

UserFlag (boolean) TRUE = this shows the Print box, FALSE = no box

This starts a printing session.

RETURN (boolean): TRUE = Ok, FALSE = Delete or Error

### End()

This ends a printing session.

### Abort()

This aborts a printing session.

### NewPage()

This runs a form-feed.

### WriteLN( Text )

Text (string) text to be written

Write a text and a CR-LF.

### WriteRC( Row, Column, Text )

Row (integer) row (1...)

Column (integer) column (1...)

Text (string) text to be written

This writes a text on a specified row-column.

The column is calculated correctly only with monospaced font.

# CREW Manual

## WriteXY( x, y, Text )

x (integer) x-coord. (0...)  
 y (integer) y-coord. (0...)  
 Text (string) text to be written

This writes a text at one of the specified coordinates.

## PrintImage( PathName, x, y, [width], [height] )

PathName (string) Image file to be printed (full path)  
 x (integer) x-coord. (0...)  
 y (integer) y-coord. (0...)  
 width (integer-optional) image width  
 height (integer-optional) image height

### Notes:

- If PathName specified only with a file name, the file is uploaded by the project images default folder.
- BMP, JPG, GIF and PNG images are supported.
- If Width or Height are not specified, the real dimensions are used.

This prints an image at the specified coordinates.

## SetFont( Name, Size, [Bold], [Italic], [Underline], [Charset] )

Name (string) font name  
 Size (integer) font size in the device unit  
 Bold (boolean-optional) TRUE = Bold (default=FALSE)  
 Italic (boolean-optional) TRUE = Italic (default=FALSE)  
 Underline (boolean-optional) TRUE = Underlined (default=FALSE)  
 Charset (integer-optional) 0 = ANSI (default), 1 = DEFAULT, 2 = SYMBOL

Set the print font.



# CREW Manual

## Properties - ESAPRN

Name	Type	Read-Write	Description
<b>FontSize</b>	integer	RW	Font size in device unit (Courier New)
<b>PageWidth</b>	integer	R	Page width in pixels
<b>PageHeight</b>	integer	R	Page height in pixels
<b>PageRows</b>	Integer	R	Rows per page
<b>PageColumns</b>	integer	R	Page columns
<b>MarginHor</b>	integer	RW	Horizontal margin in pixels
<b>MarginVert</b>	integer	RW	Vertical margin in pixels

### Example

```

set prn = ESAHMI.ESAPRN
go = prn.start(1)
if go <> 0 then
    prn.setfont "Courier New",20,1
    prn.writeln "Font Size = " & prn.FontSize
    prn.writeln "Page Width = " & prn.PageWidth
    prn.writeln "Page Height = " & prn.PageHeight
    prn.writeln "Rows = " & prn.PageRows
    prn.writeln "Columns = " & prn.PageColumns
    prn.writeln "Hor. Margin = " & prn.MarginHor
    prn.writeln "Vert. Margin = " & prn.MarginVert
    prn.newpage
    prn.writexy 100,100,"100,100"
    prn.writerc 5,10,"005,010"
    prn.printimage "logo.png", 200, 200
    prn.printimage "\hard disk\temp\logo.bmp", 300, 300, 100, 200
    prn.end
end if

```



# CREW Manual

## ESARECIPEMGR

ESARECIPEMGR provides access to the recipes of the project.

# CREW Manual

## Methods - ESARECIPEMGR

### **LoadRecipe( StructureName, RecipeName )**

StructureName (string) structure name

RecipeName (string) recipe name

This transfers a recipe from the Archive to the Buffer Tags.

### **SaveRecipe( StructureName )**

StructureName (string) structure name

This transfers a recipe from the Buffer Tags to the Archive.

Use the Recipe-Name-buffer-tag.

### **SaveRecipeAs( StructureName, RecipeName )**

StructureName (string) structure name

RecipeName (string) recipe name

This transfers a recipe from the Buffer Tags to the Archive with the specified name.

### **DeleteRecipe( StructureName, RecipeName )**

StructureName (string) structure name

RecipeName (string) recipe name

This deletes the specified recipe from the Archive.

The recipe is eliminated from the Archive; the corresponding record can be reused.

Call PackArchive() to restore the space of the archive.

# CREW Manual

## DeleteAllRecipes( StructureName )

StructureName (string) structure name

This deletes all of the recipes of the specified structure from the Archive. The Archive recipes are deleted; the corresponding record can be reused. A following call to PackArchive() is recommended.

## RenameRecipe( StructureName, OldRecipeName, NewRecipeName )

StructureName (string) structure name

RecipeName (string) recipe name

This renames an existing recipe in the Archive.

## PackArchive( StructureName )

StructureName (string) structure name

This compacts the Archive of the specified structure. The empty records are physically removed.

## ClearTagBuffer( StructureName )

StructureName (string) structure name

This deletes all of the Buffer Tags of the specified structure.

# CREW Manual

## RecipeDownload( StructureName, RecipeName, Synch )

StructureName (string) structure name  
RecipeName (string) recipe name  
Synch (boolean) TRUE = synchronised download

This transfers a recipe from the Archive to the Device Tags.

## RecipeUpload( StructureName, RecipeName, Synch )

StructureName (string) structure name  
RecipeName (string) recipe name  
Synch (boolean) TRUE = synchronised upload

This transfers a recipe from the Device Tags to the Archive.

## RecipeBufferDownload( StructureName, Synch )

StructureName (string) structure name  
Synch (boolean) TRUE = synchronised download

This transfers a recipe from the Buffer Tags to the Device Tags.

# CREW Manual

## RecipeBufferUpload( StructureName, Synch )

StructureName (string) structure name

Synch (boolean) TRUE = synchronised upload

This transfers a recipe from the Device Tags to the Buffer Tags.

## GetRecipeCount( StructureName )

StructureName (string) structure name

RETURN (integer): The number of valid recipes in the archive.

## GetRecipeRecords( StructureName )

StructureName (string) structure name

RETURN (integer): The number of recipe-records in the archive, including the empty records.

# CREW Manual

Example:

```
set rm = ESAHMI.ESARECIPEMGR
snam = "Structure1"
num1 = rm.GetRecipeCount(snam)
num2 = rm.GetRecipeRecords(snam)
s = num1 & " recipes, " & num2 & " records" & CHR(13) & CHR(13)
for i=1 to num2
v = rm.GetRecipeName(snam,i)
s = s & i & ": " & v & CHR(13)
next
ESAHMI.ESAMsgbox s
```

## GetRecipeName( StructureName, Recipeld )

StructureName (string) structure name

Recipeld (integer) recipe id (record id) [1...]

RETURN (string): The name of recipes with the specified numerical indicator.

## GetTagName( StructureName, FieldName, Device )

StructureName (string) structure name

FieldName (string) filed name ("RecipeName" and "Comment" are also allowed)

Device (boolean) TRUE = device-tag, FALSE = buffer-tag

RETURN (string): The name of the Tag associated to the specified data field.

# CREW Manual

## **RecipeExists( StructureName, RecipeName )**

StructureName (string) structure name

RecipeName (string) recipe name

RETURN (boolean): TRUE = recipe name found in the archive

FALSE = recipe name NOT found in the archive

## **RecipeExport( PathName, StructureName )**

### **RecipeExportLocal( PathName, StructureName )**

PathName (string) Full file export path name.

If the Pathname is an empty string (“”), a dialog box appears allowing the user to select a path (only Local version).

StructureName (string) structure name

To export the recipes of the specified structure to a CSV ESA-format file.

RETURN (integer): The number of exported recipes.

## **RecipeExportAll( PathName )**

PathName (string) Full file export path name.

This exports all of the recipes of all of the structures to a CSV ESA-format file.

RETURN (integer): The number of exported recipes.



# CREW Manual

## RecipeExportAllLocal( PathName, [suspensive] )

PathName (string) Full file export path name.

If the Pathname is an empty string (""), a dialog box appears allowing the user to select a path.

suspensive (boolean-optional) TRUE = suspensive dialog box

FALSE = non-suspensive dialog box

This exports all of the recipes of all of the structures to a CSV ESA-format file.

RETURN (integer): The number of exported recipes.

## RecipeImport( PathName, StructureName, [suspensive] )

PathName (string) Full file import path name.

If the Pathname is an empty string (""), a dialog box appears

allowing the user to select a path.

StructureName (string) structure name

suspensive (boolean-optional) TRUE = suspensive dialog box

FALSE = non-suspensive dialog box

This imports the recipes of the specified structure from a file.

Supported file formats:

- ESA-format CSV, (Unicode)

- Standard CSV, semicolon separated fields (Unicode and ANSI)

- Standard TXT, TAB separated fields (Unicode and ANSI)

The ImportedNew and ImportedOld properties are set.

RETURN (integer): The number of exported recipes.

# CREW Manual

## RecipeImportAll( PathName, [suspensive] )

PathName (string) Full file import path name

If the Pathname is an empty string (""), a dialog box appears allowing the user to select a path.

suspensive (boolean-optional) TRUE = suspensive dialog box  
FALSE = non-suspensive dialog box

This imports all of the recipes of all of the structures from a CSV ESA-format file. The ImportedNew and ImportedOld properties are set.

RETURN (integer): The number of imported recipes.

Example:

```
a = ESAHMI.ESARECIPEMGR.RecipeImportAll("c:\import.csv")
b = ESAHMI.ESARECIPEMGR.ImportedNew
c = ESAHMI.ESARECIPEMGR.ImportedOld
s = "Imported:" & a & " New:" & b & " Replaced:" & c
ESAHMI.ESAMsgbox s
```

## RecipeCompare( StructureName, RecipeName1, RecipeName2 )

StructureName (string) structure name

RecipeName1 (string) recipe name 1

RecipeName2 (string) recipe name 2

This compares the field values of two recipes with the same structure.

# CREW Manual

RETURN (boolean): TRUE = identical recipes  
FALSE = different recipes

## RecipeLoadBox( StructureName, [suspensive] )

StructureName (string) structure name  
suspensive (boolean-optional) TRUE = suspensive,  
FALSE = non-suspensive

This shows the Recipe Load box to load a recipe.

## RecipeSaveBox( StructureName, [suspensive] )

StructureName (string) structure name  
suspensive (boolean-optional) TRUE = suspensive,  
FALSE = non-suspensive

This shows the Recipe Save box to save a recipe.

## RecipeSaveAsBox( StructureName, [suspensive] )

StructureName (string) structure name  
suspensive (boolean-optional) TRUE = suspensive, FALSE = non-suspensive

This shows the Recipe Save As box to save a recipe with a name.

# CREW Manual

## RecipeDeleteBox( StructureName, [suspensive] )

StructureName (string) structure name  
 suspensive (boolean-optional) TRUE = suspensive,  
 FALSE = non-suspensive

This shows the Recipe Delete box to delete a recipe.

## RecipeRenameBox( StructureName, [suspensive] )

StructureName (string) structure name  
 suspensive (boolean-optional) TRUE = suspensive,  
 FALSE = non-suspensive

This shows the Recipe Rename box to rename a recipe.

## RecipeDownloadBox( StructureName, Synch, [suspensive] )

StructureName (string) structure name  
 Synch (boolean) TRUE = synchronized download  
 suspensive (boolean-optional) TRUE = suspensive,  
 FALSE = non-suspensive

This shows the Recipe Download box to download a recipe.

## RecipePrint( StructureName ) RecipePrintLocal( StructureName )

StructureName (string) structure name

This prints the recipes of the specified structure.

RETURN (integer): The number of printed recipes.

# CREW Manual

RecipePrintAll()  
RecipePrintAllLocal()

This prints all of the recipes of all of the structures.

RETURN (integer): The number of printed recipes.

## Properties - ESARECIPEMGR

Name	Type	Read-Write	Description
<b>Busy</b>	integer	R	0 = no transfer in progress 1 = transfer in progress
<b>ImportedNew</b>	Integer	R	the number of new imported recipes after a <i>RecipeImport()</i> call
<b>ImportedOld</b>	Integer	R	the number of replaced recipes after a <i>RecipeImport()</i> call

# CREW Manual

## ESASAMPLEMGR

ESASAMPLEMGR provides access to project samples.

# CREW Manual

## Methods - ESASAMPLEMGR

### Enable( SampleName )

SampleName (string) Sample name  
This enables the sample taking activity.

### Disable( SampleName )

SampleName (string) Sample name  
This disables the sample taking activity.

### ResetSamples( SampleName )

SampleName (string) Sample name  
This deletes all of the samples from the sampling buffer.

### AcquireSample( SampleName )

SampleName (string) Sample name  
This requests a “one-shot” acquisition of a new sample from the source.

# CREW Manual

## ExportSamples( PathName, SampleName )

Pathname (string) Full name of file path

SampleName (string) Sample name

This exports all of the sampling buffer samples to a file.

## ExportSamplesLocal( PathName, SampleName, [suspensive] )

Pathname (string) Full name of file path

If the Pathname is an empty string (""), a dialog box appears allowing the user to select a path.

SampleName (string) Sample name

suspensive (boolean-optional) TRUE = suspensive dialog box

FALSE = non-suspensive dialog box

This exports all of the sampling buffer samples to a file.

## ImportSamplesLocal( Pathname, PageName, ControlName, [suspensive] )

Pathname (string) Full name of file path.

dialog box appears

allowing the user to select a path.

PageName (string) Page name

ControlName (string) Control Name.

suspensive (boolean-optional) TRUE = suspensive dialog box

FALSE = non-suspensive dialog box

This imports tracking data from a text file.



# CREW Manual

**ExportInProgress( SampleName )**

**ExportInProgressLocal( SampleName )**

SampleName (string) Sample name

This establishes whether an export operation is in progress on the specified sampling buffer.

RETURN (boolean): TRUE if an export operation is in progress.

**WaitForExport( SampleName )**

**WaitForExportLocal( SampleName )**

SampleName (string) Sample name

Wait for an export operation to finish.

**TerminateExport( SampleName )**

**TerminateExportLocal( SampleName )**

SampleName (string) Sample name

Terminate an export operation.

**FlushPersistentData( SampleName )**

SampleName (string) Sample name

Save the collected persistent samples on disc.

# CREW Manual

SamplesPrint( SampleName )

SamplesPrintLocal( SampleName )

SampleName (string) Sample name

Print the sample data.



# CREW Manual

## ESATAG

ESATAG provides access to the project tags. Plus, it allows you the access the external device directly.

# CREW Manual

## Methods - ESATAG

### GetTagId( TagName )

TagName (string) Tag name

RETURN (integer): This is the identifier of the Tag

### GetTagName( TagId )

TagId (integer) Tag identifier

RETURN (string): The Tag name

### GetTagValueType( TagName )

TagName (string) Tag name

RETURN (integer): The type of Tag value:

16	1- byte signed integer	(I1)
2	2-byte signed integer	(I2)
3	4-byte signed integer	(I4)
17	1-byte unsigned integer	(UI1)
18	2-byte unsigned integer	(UI2)
19	4-byte unsigned integer	(UI4)
4	4-byte floating point	(R4)
5	8-byte floating point	(R8)
11	boolean	(BOOL)
8	string	(BSTR)

Note:

The types of returned Tag Arrays are defined by adding 8192 to the above values.

(example of a string matrix: 8200 = 8 + 8192)

# CREW Manual

## GetTagStrLength( TagName )

TagName (string) Tag name

RETURN (integer): The length of a Tag string

## GetTagArraySize( TagName )

TagName (string) Tag name

RETURN (integer): The number of elements of a Tag matrix

## GetDeviceId( DeviceName )

DeviceName (string) Device name

RETURN (integer): This is the identifier of the Device

## GetDeviceName( DeviceId )

DeviceId (integer) Device identifier

RETURN (string): The name of the Device

# CREW Manual

## GetCurrentValue( TagName )

TagName (string) Tag name

This reads the current value contained in the tag (it does not access the value in the device)

RETURN (variant): The Tag value

## ReadValue( TagName )

TagName (string) Tag name

This reads the value of a tag from the device

RETURN (variant): The Tag value

### Example:

```
v = ESAHMI.ESATAG.ReadValue("Tag")
```

## WriteValue( TagName, Value )

TagName (string) Tag name

Value (variant) New Tag value

This writes a new tag value on the device

### Example:

```
ESAHMI.ESATAG.WriteValue "Tag",123
```

# CREW Manual

## ReadElement( TagName, Index )

TagName (string) Tag name

Index (integer) Index of the requested element [0...]

This reads the value of a single element of a tag matrix from the device

RETURN (variant): The value of the element

## WriteElement( TagName, Index, Value )

TagName (string) Tag name

Index (integer) Index of the requested element [0...]

Value (variant) New tag value

This writes the value of a single element of a tag matrix on the device

### Example:

```
a = Array(10,20,30)
```

```
ESAHMI.ESATAG.WriteValue "TagArray",a
```

```
r = ESAHMI.ESATAG.ReadElement( "TagArray",2 )
```

```
r = r + 1000
```

```
ESAHMI.ESATAG.WriteElement "TagArray",2,r
```

```
q = ESAHMI.ESATAG.ReadValue( "TagArray" )
```

```
b = q(1)
```

# CREW Manual

## ReadBit( TagName, Index )

TagName (string) Tag name

Index (integer) Index of the requested bit [0...]

This reads the value of a numerical tag bit (or matrix) from the device

RETURN (boolean): The bit value

## WriteBit( TagName, Index, Value )

TagName (string) Tag name

Index (integer) Index of the requested element [0...]

Value (boolean) New tag value

This writes the value of a single numerical tag bit (or matrix) on the device

## ReadItem( DeviceId, Areald, ValueType, StringLen, ArraySize, IsBCD, AF1, [AF2], [AF3], [AF4], [AF5], [AF6], [AF7], [AF8] )

DeviceId (integer) Device identifier

Areald (integer) Area identifier

ValueType (integer) Value type (see GetTagValueType)

StringLen (integer) String length (valid for String tags)

ArraySize (integer) Number of elements in the matrix (valid for matrix-tag)

IsBCD (boolean) TRUE = BCD coding

AF1 (variant) Address Field 1

AF2 - AF8 (variant-optional) Address Fields 2 . . 8

This reads a value directly from the device

RETURN (variant): The device value



# CREW Manual

**WriteItem( Value, DeviceId, AreaId, ValueType, StringLen, ArraySize, IsBCD, AF1, AF2, AF3, AF4, AF5, AF6, AF7, AF8 )**

Value (variant) Value to be written  
 DeviceId (integer) Device identifier  
 AreaId (integer) Area identifier  
 ValueType (integer) Value type (see GetTagValueType)  
 StringLen (integer) String length (valid for String tags)  
 ArraySize (integer) Number of elements in the matrix (valid for matrix-tag)  
 IsBCD (boolean) TRUE = BCD coding  
 AF1 (variant) Address Field 1  
 AF2 - AF8 (variant-optional) Address Fields 2 . . 8

This writes a value directly on the device

**SetTagOffscan( TagName, offScan )**

TagName (string) Tag name  
 offScan (boolean) TRUE = set off-scan, FALSE = reset off-scan  
 This sets the off-scan status of the Tag

**SetDeviceOffscan( DeviceName, offScan )**

DeviceName (string) Device name  
 offScan (boolean) TRUE = set off-scan, FALSE = reset off-scan

This sets the off-scan status of the Device

# CREW Manual

## IsOffline( TagName )

TagName (string) Tag name

This checks if the specified Tag is Off Line.

RETURN (boolean): TRUE = Tag Off-Line, FALSE = Tag On-Line

# CREW Manual

## ESATIMER

ESATIMER provides access to the project Timers.

# CREW Manual

## Methods - ESATIMER

### Start( timername )

timername (string) timer to be started  
This starts the specified timer.

### Stop( timername )

timername (string) timer to be stopped  
This stops the specified timer.

### Suspend( timername )

timername (string) timer to be suspended  
This suspends the specified timer.

### SetTimerValue( timername, value )

timername (string) timer to be set  
value (integer) limit value to be set  
This sets the limit of the specified timer.

### GetTimerValue( timername )

timername (string) timer name  
RETURN (integer): The limit of the specified timer.

# CREW Manual

## SetProgress( timername, value )

timername (string) timer to be set  
value (integer) progress value to be set

This sets the progress value of the specified timer.

## GetProgress( timername )

timername (string) timer name  
RETURN (integer): The progress value of the specified timer.

## IsStarted( timername )

timername (string) timer name  
RETURN (boolean): TRUE if the timer has started.

## IsSuspended( timername )

timername (string) timer name  
RETURN (boolean): TRUE if the timer is suspended.

# CREW Manual

## ESAUUSERMGR

ESAUUSERMGR manages the project Users.

# CREW Manual

## Methods - ESAUSERMGR

**Add( username, groupname, mode, password, [language], [email], [phone], [validity] )**

username	(string)	username to be added
groupname	(string)	group name
mode graphical)	(integer)	password mode (0 = alphanumerical, 1 =
password	(string)	password
language change)	(integer-optional)	language identifier [1...] (0 = no language
email	(string-optional)	email address
phone	(string-optional)	telephone number
validity	(integer-optional)	days of password validity (0 = no limit)

Add a new user.

**Remove( username )**

username (string) username to be removed

Delete an existing user.

**Login( username, password )**

username (string) username for login

password (string) password

This is to login an existing user.

# CREW Manual

## Logout()

This is to logout the current user.

## ChangePassword( username, mode, password )

username (string) username  
mode (integer) password mode (0 = alphanumerical, 1 = graphical)  
password (string) password

This changes the password of an existing user.

## ChangePasswordValidity( username, validity )

username (string) username  
validity (integer) days of password validity (0 = no limit)

This changes the validity of the password of an existing user.

## ChangeGroup( username, groupname )

username (string) username  
groupname (string) name of new group

This changes the group of an existing user.



# CREW Manual

## ChangeLanguage( username, language )

username (string) username  
language (integer-optional) language identifier [1...] (0 = no language change)

This changes the language of an existing user.

## ChangeEmail( username, email )

username (string) username  
email (string) email address

This changes the email address of an existing user.

## ChangeTelNumber( username, phone )

username (string) username  
phone (string) telephone number

This changes the telephone number of an existing user.

## UsersFlush()

This clears the user log file on the disc.

# CREW Manual

## GetCurrentUserName()

RETURN (string): The Name of the currently logged user.

## GetCurrentVisibility()

RETURN (integer): The Visibility Level of the currently logged user.

## GetCurrentInteractivity()

RETURN (integer): The Interactivity Level of the currently logged user.

## GetCurrentGroup()

RETURN (string): The name of the Group associated to the currently logged user.

## GetUserVisibility( username )

username (string) username

RETURN (integer): The Visibility Level of the specified user.

## GetUserInteractivity( username )

username (string) username

RETURN (integer): The Interactivity Level of the specified user.

# CREW Manual

## GetUserGroup( username )

username (string) username

RETURN (string): The name of the Group associated to the specified user.

## GetUserLanguage( username )

username (string) username

RETURN (integer): The language associated to the specified user [1...] (0 = no language change)

## GetUserEmail( username )

username (string) username

RETURN (string): The email address associated to the specified user.

## GetUserTelNumber( username )

username (string) username

RETURN (string): The telephone number associated to the specified user.

## GetUserPasswordValidity( username )

username (string) username

RETURN (integer): Days of password validity (0 = no limit).

# CREW Manual

## LogExport( pathname )

Pathname (string) Full name of file path.

This exports the Users Log to the specified file.

RETURN (integer): The number of exported registers.

## LogExportLocal( pathname, [suspensive] )

Pathname (string) Full name of file path.

If Pathname is an empty string (""), a box appears so that the user can select a path.

suspensive (boolean-optional) TRUE = suspensive dialog box

FALSE = non-suspensive dialog box

This exports the Users Log to the specified file.

RETURN (integer): The number of exported registers.

## LoginBox( [suspensive] )

suspensive (boolean-optional) TRUE = suspensive, FALSE = non-suspensive

This displays the User Login box for user login.

## LoginPasswordBox( username, [suspensive] )

username (string) user name

suspensive (boolean-optional) TRUE = suspensive, FALSE = non-suspensive

This displays the Password Login box for the login of the specified user.

# CREW Manual

## AddBox( [suspensive] )

suspensive (boolean-optional) TRUE = suspensive, FALSE = non-suspensive  
This displays the User Add box to add a user to the project.

## RemoveBox( [suspensive] )

suspensive (boolean-optional) TRUE = suspensive, FALSE = non-suspensive  
This displays the User Remove box to remove a user from the project.

## ChangeInfoBox( [suspensive] )

suspensive (boolean-optional) TRUE = suspensive, FALSE = non-suspensive  
This displays the User Change Info box to change the user settings (password, ...)

## UserLock( username )

username (string) username

This locks the specified user.

## UserUnlock( username )

username (string) username

This unlocks the specified user.

# CREW Manual

UsersPrint()  
UsersPrintLocal()

This prints the Users Log.

RETURN (integer): The number of printed events.

UserJoinList( username, mailinglist, type )

username (string) username  
mailinglist (string) mailing list  
type (integer) type of receiver (0=normal, 1=copy, 2=hidden)

This adds the user to a mailing list.

UserLeaveList( username, mailinglist )

username (string) username  
mailinglist (string) mailing list

This removes the user from a mailing list.

UserResetLists( username )

username (string) username  
This removes the user from all of the mailing lists.

# CREW Manual

## SendMailSingle( email, subject, message )

email (string) email address  
 subject (string) email subject  
 message (string) email message

This sends an email.

## SendMailList( mailinglist, replyto, attachments, subject, message )

mailinglist (string) name of mailing list  
 attachments (string) separator TAB (CHR(9)) name attachment path  
 subject (string) email subject  
 message (string) email message

This an email to a mailing list.

## SendMailBox( [suspensive] )

suspensive (boolean-optional) TRUE = suspensive, FALSE = non-suspensive

This shows the “Send E-mail” box.



# CREW Manual

## SendSmsSingle( phone, message )

phone (string) phone number  
message (string) sms message

Sends an s.m.s.

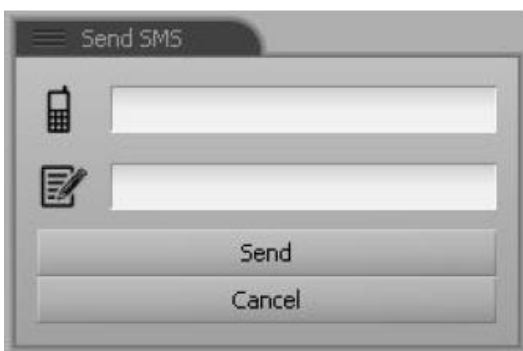
## SendSmsList( phonenumberlist, message )

phonenumberlist (string) phone numbers list name  
message (string) sms message

Sends an s.m.s. to a list of numbers.

## SendSmsBox( [suspensive] )

suspensive (boolean-optional) TRUE = suspensive, FALSE = non-suspensive  
Shows a “Send SMS” box.





# CREW Manual

## ImportNetworkUsers()

This imports a user list from the specified network in the project.

## Predefined functions

This section explains the predefined functions contained in Crew. They are useful during project development, as they can generally be associated to the events of the various objects (see “Events” section), after having been selected from the relative drop down menu. For certain types of function it is also necessary to specify the variables or the objects you need to work on and the values that they need to work at.

A typical example of the use of Crew's predefined functions is when they are associated with buttons and switches, changes of values in value fields or in the opening and closing pages and pop-ups.

In the following sections, the various predefined functions of Crew, divided by subject, will be shown :

[Alarm Functions](#)

[Hardware Functions](#)

[Message Functions](#)

[Page Functions](#)

[Project Functions](#)

[Recipe Functions](#)

[Datalogs - Samples - Functions](#)

[System Functions](#)

[Variables - Tags - Functions](#)

[Timers Functions](#)

[Users Functions](#)

# CREW Manual

## Alarm Functions

Function	Description
<b>AlarmAck</b>	Activated when the alarm is silenced
<b>AlarmAckGlobal</b>	Acknowledges all alarms (if type AlarmISA or OnlyAck); requires as input the name of the operator and the station from which the request originates (parameter valid in case of network).
<b>AlarmAckGroup</b>	Acknowledges all alarm instances (if type AlarmISA or OnlyAck) of the specified unit; requires as input the name of the unit, the operator and the station from which the request originates (parameter valid in case of network).
<b>AlarmAckInstances</b>	Acknowledges all record of the specified alarm; requires as input the name of the alarm, the operator and the station from which the request originates (parameter valid in case of network).
<b>AlarmExport</b>	Exports all active alarms at RUNTIME to file. The name of the destination file (CSV format) must be specified.
<b>AlarmExportBox</b>	Export all the active alarms, a dialog box will ask for the exported file name
<b>Alarm ExportLocal</b>	Exports all active alarms at RUNTIME to file. The name of the destination file (CSV format) must be specified. Export is performed on the machine where the user interface is active.
<b>Alarm Off</b>	Controls the termination of the alarm
<b>Alarm On</b>	Controls the activation of the alarm
<b>HistoryExport</b>	Exports all records of the alarm log at RUNTIME to file.
<b>HistoryExportLocal</b>	Exports all records of the alarm log at RUNTIME to file. Export is performed on the machine where the user interface is active.
<b>HistoryFlush</b>	Allows to force writing on file of the alarm log
<b>HistoryReset</b>	Allows to reset all the contents of the alarm log

# CREW Manual

## Hardware functions

Function	Description
<b>LightDown</b>	Allows to decrease the brightness of the terminal display
<b>LightSet</b>	Allows to set the brightness of the terminal display
<b>LightUp</b>	Allows to increase the brightness of the terminal display

## Message Functions

Function	Description
<b>SendEmail</b>	Send an email to a specific address
<b>SendEmailBox</b>	Show a dialog box that allows to enter and send an email message
<b>SendEmailList</b>	Send an email to a whole preconfigured mailing list.

# CREW Manual

## Page Functions

Function	Description
<b>ClosePopupAll</b>	Closes all Pop-up pages currently open at RUNTIME
<b>ClosePopupId</b>	Closes specific Pop-up page; the number of the page to be closed must be specified
<b>ClosePopupName</b>	Closes specific Pop-up page; the name of the page to be closed must be specified
<b>ClosePopupTop</b>	Closes the top Pop-up page
<b>ShowPageId</b>	Switches to display the specified page; the number of the page to be displayed must be specified
<b>ShowPageLast</b>	Switches to display the last page of the project displayed
<b>ShowPageName</b>	Switches to display the specified page; the name of the page to be displayed must be specified
<b>ShowPageNext</b>	Displays the next page (according to the enabled sequence)
<b>ShowPageNextFull</b>	Displays the next Full Screen page (according to the page ID)
<b>ShowPageNextPopup</b>	Displays the next pop-up page (order of the ID number of the page is followed)
<b>ShowPagePrevious</b>	Displays the previous page (according to the enabled sequence)
<b>ShowPagePreviousFull</b>	Displays the previous Full Screen page (according to the page ID)
<b>ShowPagePreviousPopup</b>	Displays the previous pop-up page (order of the ID number of the page is followed)
<b>ShowPagePopupsMap</b>	Displays a window with a preview of all pop-up pages
<b>ShowRoadMap</b>	Displays a window with a preview of all Full Screen pages
<b>ShowSequenceRoll</b>	Displays up front a sequence of images (Carousel) from which the page to display can be chosen

# CREW Manual

## Project Functions

Function	Description
<b>DisableInteraction</b>	Disables the user's interactions on the terminal (for example, touch screen, mouse or keyboard functions are disabled)
<b>EnableInteraction</b>	Enables the user's interactions on the terminal (the user can interact on the terminal through touch screen, mouse or keyboard)
<b>ExitRuntime</b>	At RUNTIME, completely exits the project and returns to the operating system of the panel
<b>FdaTracingDisable</b>	Disable FDA logging of runtime events
<b>FdaTracingEnable</b>	Enable FDA logging of runtime events
<b>FdaTracingExport</b>	Export the log of the FDA runtime events
<b>FdaTracingExportBox</b>	Export the log of the FDA runtime events; a dialog box will ask for the exported file name
<b>FdaTracingExportLocal</b>	Export the log of the FDA runtime events on the local machine; if no FileName is provided a request dialog box is visualized
<b>FdaTracingPrint</b>	Print the log of the FDA runtime events
<b>FdaTracingPrintLocal</b>	Print the log of the FDA runtime events, a dialog box allows the selection of the target printer
<b>FdaTracingReset</b>	Export the log of the FDA runtime events and reset the logging content and state
<b>FdaTracingResetBox</b>	Export the log of the FDA runtime events and reset the logging content and state, a dialog box will be prompted to request the name of the export file
<b>FdaTracingResetLocal</b>	Export the log of the FDA runtime events and reset the logging content and state, if no FileName is provided, a request dialog box is visualized
<b>FlushFdaTracing</b>	Save in hard disk the FDA runtime events log
<b>FlushPersistentData</b>	Saves the required buffers on the hard disk (saving is applied to all existing persisting elements)

<b>NextLanguage</b>	Changes the language of the project currently in use to the next one specified in CREW's list; all elements to be translated will be displayed in the new language
<b>PreviousLanguage</b>	Changes the language of the project currently in use to the next one specified in CREW's list; all elements to be translated will be displayed in the new language
<b>RefreshIpAddresses</b>	Cause a check of the IP addresses active on the terminal and the update of the related system variables
<b>RunApplication</b>	Allows to launch an application installed on the device
<b>ScreenSaverEnter</b>	Immediately enter the screen saver condition, without waiting for the inactivity time expiration
<b>ScreenSaverKick</b>	Reset the screen saver timer; leave screen saver condition if currently active
<b>SetDate</b>	Sets the "Date" parameter on the terminal
<b>SetDateTime</b>	Sets the "Date" and "Time" parameters on the terminal
<b>SetLanguage</b>	Sets the project language on the terminal
<b>SetTime</b>	Sets the "Time" parameter on the terminal
<b>ShowCalculatorBox</b>	Show a dialog box representing a scientific calculator
<b>ShowDateTimeBox</b>	Displays a window with Date and Time
<b>ShowHelpPage</b>	Displays a Help page associated with a "Full Screen" page
<b>ShowHelpPagePopup</b>	Displays a Help page associated with a Pop-up page
<b>ShowMessageBox</b>	During RUNTIME pops up a window containing a message; the window can have 3 different types of wallpaper, grey, light blue and red depending on the meaning: plain text, information, error
<b>ShowResourceMonitorBox</b>	Show a dialog box for the visualization of the state of the system resources (RAM, Flash and CPU time)
<b>ShutdownPC</b>	Closes RUNTIME and shuts down the machine (IPC)
<b>TerminateApplication</b>	Terminate an external application currently in execution

# CREW Manual

## Recipe Functions

Function	Description
<b>RecipeClearBuffer</b>	Deletes all tag buffer contents of the data structure
<b>RecipeDelete</b>	Allows to delete a recipe; recipe type and name are required
<b>RecipeDeleteBox</b>	Allows to delete a recipe; recipe type is required; a window is displayed to allow selection of the recipe to be deleted.
<b>RecipeDeleteId</b>	Allows to delete a recipe, by its ID; recipe type and ID are required
<b>RecipeDownload</b>	Allows to download a recipe to the device; recipe type and name are required (the buffer is not influenced during this operation)
<b>RecipeDownloadBox</b>	Allows to download a recipe to the device (the buffer is not influenced during this operation); a dialog box is displayed for selection of the recipe to be downloaded
<b>RecipeDownloadBuffer</b>	Allows to download a recipe from the buffer to the device (PLC)
<b>RecipeDownloadId</b>	Allows to download a recipe, by its ID, to the device (the buffer is not influenced during this operation)
<b>RecipeExport</b>	Allows to export to a file (CSV) a recipe to the terminal; in CREW, recipe type to which this command refers must be specified
<b>RecipeExportAll</b>	Allows to export to a file (CSV) all present recipes, regardless of the type
<b>RecipeExportAllBox</b>	Export, in the given file, all the recipes of all the existing structures; a dialog box will ask for the exported file name
<b>RecipeExportAllLocal</b>	Allows to export to a file (CSV) all present recipes, regardless of the type; export is performed on the machine on which the user interface is enabled
<b>RecipeExportBox</b>	Export, in the given file, all the recipes of a given structure a dialog box will ask for the exported file name
<b>RecipeExportLocal</b>	Allows to export to a file (CSV) a recipe to the terminal; in CREW, recipe type to which this command refers must be specified; export is performed on the machine on which the user interface is enabled
<b>RecipeImport</b>	Imports the recipes to a file (CSV) on the terminal
<b>RecipeImportAll</b>	Import all the recipes contained in the given file, into the needed archives of the existing structures
<b>RecipeImportAllBox</b>	Import all the recipes contained in the given file, into the needed archives of the existing structures, in this case the box allows the selection
<b>RecipeImportBox</b>	Import the recipes contained in the given file, into the archive of the given structure, in this case the box allows the selection

<b>RecipeLoad</b>	Uploads a specific type of recipe; in CREW, recipe type to which this command refers must be specified
<b>RecipeLoadBox</b>	Uploads a recipe from the archive to the buffer; a window is displayed to allow selection of the recipe to be uploaded
<b>RecipeLoadId</b>	Uploads a recipe, by its ID, from the archive to the buffer
<b>RecipePack</b>	Compresses recipes contained in one archive; the operation may result in changing the ID of the existing recipes
<b>RecipePrint</b>	Print all the recipes of a given structure; the print is automatically directed to the default printer
<b>RecipePrintAll</b>	Print all the recipes of all the existing structures; the print is automatically directed to the default printer
<b>RecipePrintAllLocal</b>	Print all the recipes of all the existing structures; a dialog box allows the selection of the target printer
<b>RecipePrintLocal</b>	Print all the recipes of a given structure; a dialog box allows the selection of the target printer
<b>RecipeRename</b>	Changes the name of a recipe in the archive, by its old name
<b>RecipeRenameBox</b>	Changes the name of a recipe in the archive; a window is displayed to allow selection of the recipe to be renamed and editing of its new name
<b>RecipeRenameld</b>	Changes the name of a recipe in the archive, by its ID
<b>RecipeSave</b>	Saves a recipe from the buffer to the archive; the function performs an immediate backup of the recipe currently in the buffer and overwrites the file without prior notice; if the valid name of a recipe is in the buffer, then the function can be executed, otherwise the backup fails with an error
<b>RecipeSaveAs</b>	Saves a recipe from the buffer to the archive; the name of the recipe is initially assigned to the dedicated buffer, therefore backup is performed; the old recipes are overwritten without prior notice
<b>RecipeSaveAsBox</b>	Saves a recipe from the buffer to the archive; a window is displayed to allow a name for the recipe to be entered; the name of the recipe is initially assigned to the dedicated buffer, therefore backup is performed; in case of overwriting, a window is displayed requesting confirmation
<b>RecipeSaveBox</b>	Saves a recipe from the buffer to the archive; if the valid name of a recipe is in the buffer, then the function can be executed, otherwise the function acts as "Recipe SaveAs"; in case there is already a recipe with the same name, a window is displayed requesting confirmation, otherwise saving is performed immediately
<b>RecipeUpload</b>	Uploads a recipe from the device to the archive (the buffer is not influenced during the operation); the device -PLC- variables must include the name of the recipe
<b>RecipeUploadBuffer</b>	Uploads the recipe indicated by the input parameters to the buffer from the device. The ID of the recipe type must be supplied

# CREW Manual

## Datalog - Sample Functions

Function	Description
<b>SamplesAcquire</b>	Reads a sample of the Data Log; requests input of the Log-Buffer ID on which the command acts upon
<b>SamplesDisable</b>	Disables acknowledgement of the Log Data indicated; requests input of the Log-Buffer ID on which the command acts upon
<b>SamplesEnable</b>	Enables acknowledgement of the Log Data indicated; requests input of the Log-Buffer ID on which the command acts upon
<b>SamplesExport</b>	Exports to file the Log Data indicated; requests ID of LogBuffer relative to the name of the destination file
<b>SamplesExportBox</b>	Export all the samples of a given buffer; a dialog box will ask for the exported file name
<b>SamplesExportLocal</b>	Exports to file the Log Data indicated; requests ID of LogBuffer relative to the name of the destination file; export is performed on the machine on which the user interface is enabled
<b>SamplesFlush</b>	Allows to force writing on file of the Log Data values in the buffer
<b>SamplesPrint</b>	Print all the samples of a given buffer; the print is automatically directed to the default printer
<b>SamplesPrintLocal</b>	Print the samples of a given buffer; a dialog box allows the selection of the target printer
<b>SamplesReset</b>	Resets the buffer of the Log Data indicated; requests input of the Log-Buffer ID on which the command acts upon



# CREW Manual

## System Functions

Function	Description
CoDeSysOff	Unload CoDeSys runtime
CoDeSysOn	Load CoDeSys runtime
CoDeSysRun	Start CoDeSys applications
CoDeSysStop	Stop CoDeSys applications
EverywareDisable	Disable Everyware services
EverywareEnable	Enable Everyware services
EverywareOff	Unload Everyware services manager
EverywareOn	Load Everyware services manager

# CREW Manual

## Variables - Tag Functions

Function	Description
<b>Add</b>	Allows to increase a given variable by a value; the variable to which the command is to be applied and the increased value must be specified; the ID and the Tag Value are required
<b>And</b>	With this operation, the AND logic operation is executed on the binary values representation; the variable on which to perform the operation and the value with which to perform the AND operation must be specified. The result of the operation replaces the original value of the variable; Tag's ID and BitMask are required
<b>BitReset</b>	Allows reset of the value of a bit (only for internal type Tags, including Arrays); the ID of the variable to which you want to apply reset and the bit position to be reset must be specified
<b>BitSet</b>	Forces the value of a bit of a variable to a specified value (only for the internal type Tags, Arrays included); the ID of the variable to which you want to apply reset and the bit position to be reset must be specified
<b>BitToggle</b>	Inverts the value of a bit of a variable (only for the internal type Tags, Arrays included); the ID of the variable to which you want to apply inversion and the bit position to be inverted must be specified
<b>Divide</b>	Allows to divide a given variable by a value; the ID of the variable to which you want to apply the command and the division value (numeric) are required

<b>FlushPersistentTags</b>	Allows to force writing of the current values of the persisting internal type Tag
<b>Multiply</b>	Allows to multiply a given variable by a value; the ID of the variable to which you want to apply the command and the multiplication value (numeric) are required
<b>Not</b>	With this operation, the NOT logic operation is executed on the binary values representation; Tag's ID and BitMask are required The result of the operation replaces the original value of the variable
<b>Or</b>	With this operation, the OR logic operation is executed on the binary values representation; Tag's ID and BitMask are required The result of the operation replaces the original value of the variable
<b>Power</b>	Sets the value of the variable by elevating it to a given power factor; Tag's ID and power factor (numeric) are required
<b>Set</b>	Forces the value of a variable to a specified value; Tag's ID and value are required
<b>Shift</b>	Displaces the Tag value by a given number of bits; Tag's ID and number of bits are required
<b>Subtract</b>	Allows to decrease a given variable by a value; Tag's ID and decrease value are required
<b>Xor</b>	With this operation, the XOR logic operation is executed on the binary values representation; Tag's ID and Value are required; The result of the operation replaces the original value of the variable

# CREW Manual

## Timers Functions

Function	Description
<b>TimerSetCounter</b>	Sets the counter value of a timer, by its name; may have effect on the client or server, depending on the execution environment of the timer; name and value of the Timer are required
<b>TimerSetCounterId</b>	Sets the counter value of a timer, by its ID; may have effect on the client or server, depending on the execution environment of the timer; name and value of the Timer are required
<b>TimerStart</b>	Starts the selected Timer count; name of timer to which the command refers is to be specified
<b>TimerStartId</b>	Sets the count of a Timer, by its ID; may have effect on the client or server, depending on the execution environment of the timer; Timer ID is required
<b>TimerStop</b>	Stops the selected Timer count; name of timer to which the command refers is to be specified
<b>TimerStopId</b>	Stops the count of a Timer, by its ID; may have effect on the client or server, depending on the execution environment of the timer; Timer ID is required
<b>TimerSuspend</b>	Temporarily suspends the selected Timer count; name of timer to which the command refers is to be specified
<b>TimerSuspendId</b>	Temporarily suspends the Timer count, by its ID; may have effect on the client or server, depending on the execution environment of the timer; Timer ID is required

# CREW Manual

## Users Functions

Function	Description
UserAdd	Adds a new user to the user list, the parameters required are: User name, User group index, Password Mode (0 = alphanumeric password, 1 = graphic password), Password
UserAddBox	Displays a dialogue window for input of UserName, UserGroup and Password
UserChangeInfo	Changes the information (user level and password) of an existing user
UserChangeInfoBox	Displays a dialogue window for selection of a User Name and input of User Level and Password
UserLock	Lock the specified user (can be processed only if the current user is of suitable level)
UserLogin	Allows to recall the operation of user login, User name and Password are required
UserLoginBox	Displays a dialogue window for login of a User Name and Password
UserLoginPassword	Displays a dialogue window for entering Password
UserLogout	Allows to recall the logout operation
UserRemove	Removes a user from the user list, by his name
UserRemoveBox	Displays a dialogue window for input of UserName
UserResetPredefined	Resets all changes made during definition of Users and Passwords, the original default settings are reloaded from the project. A user with a 1/1 qualifying level must be connected in order to perform the function; the current user will be logged off at the end of the operations
UsersExport	Export the log of the users actions
UsersExportBox	Export the log of the users actions, a dialog box will ask for the exported file name
UsersExportLocal	Exports the user actions log to the local machine (run by the "Client")
UsersFlush	Saves to the hard disk the log's user actions (login, logout, password change, etc.)
UsersPrint	Print the log of the users actions, the print is automatically directed to the default printer
UsersPrintLocal	Print the log of the users actions, a dialog box allows the selection of the target printer
UserUnlock	Unlock the specified user (can be processed only if the current user is of suitable level)

## CREW Manual

# Tutorial

The purpose of this section is to provide practical examples of how Crew is used, the settings and basic operations of CODESYS and Everyware.

Crew Tutorial Video:

<https://www.youtube.com/playlist?list=PLAE8hBeg92GKqVqCnVbvdhQvbatO9s4mH>

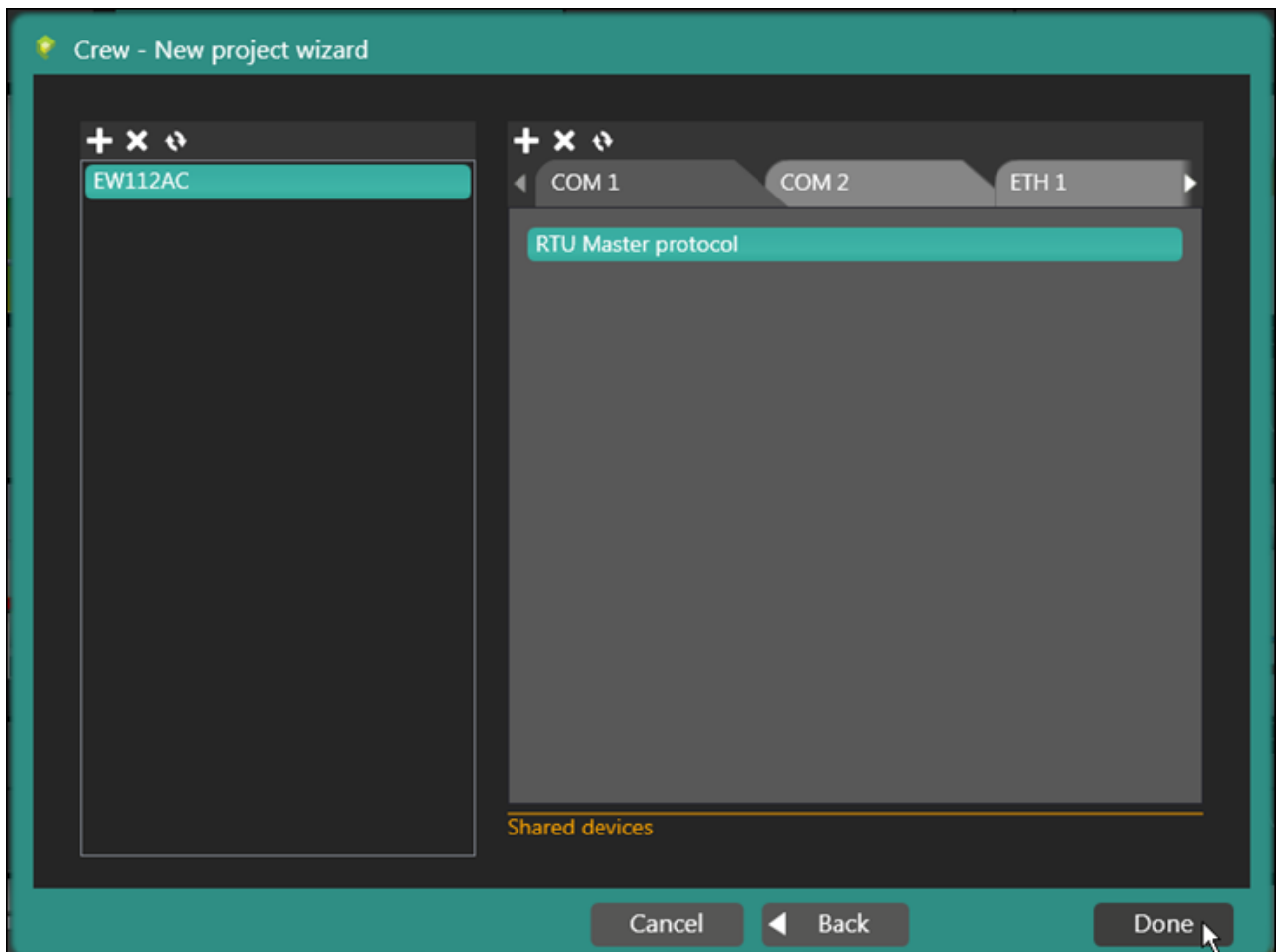
# CREW Manual

## Example Crew+CODESYS project

Esaware Crew - Basic settings -

Modbus RTU Master protocol configuration

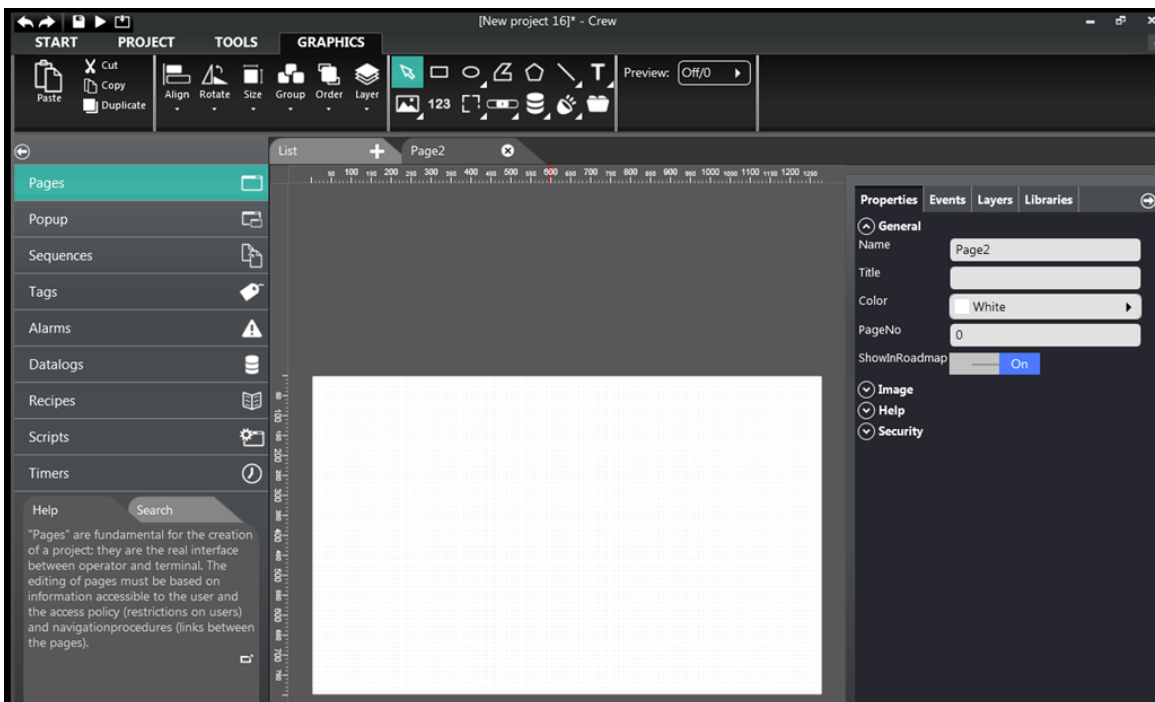
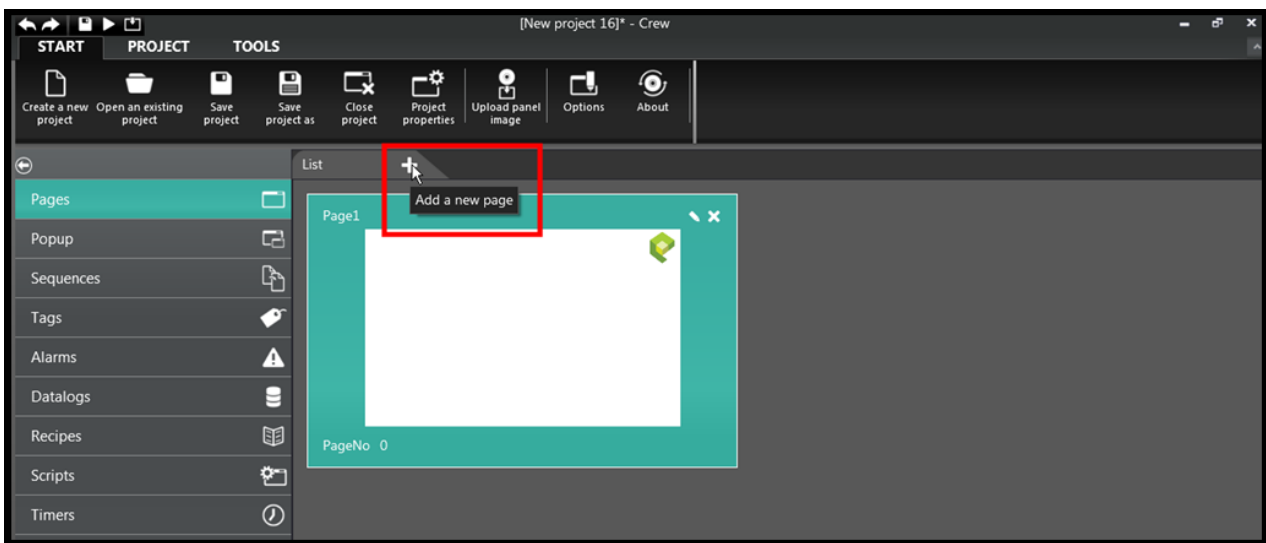
Below is an explanation of how to create a new project with Crew and with an Esaware line product (for example EW112AC), connecting it to a “Modbus RTU Master”.



# CREW Manual

## Adding a project page

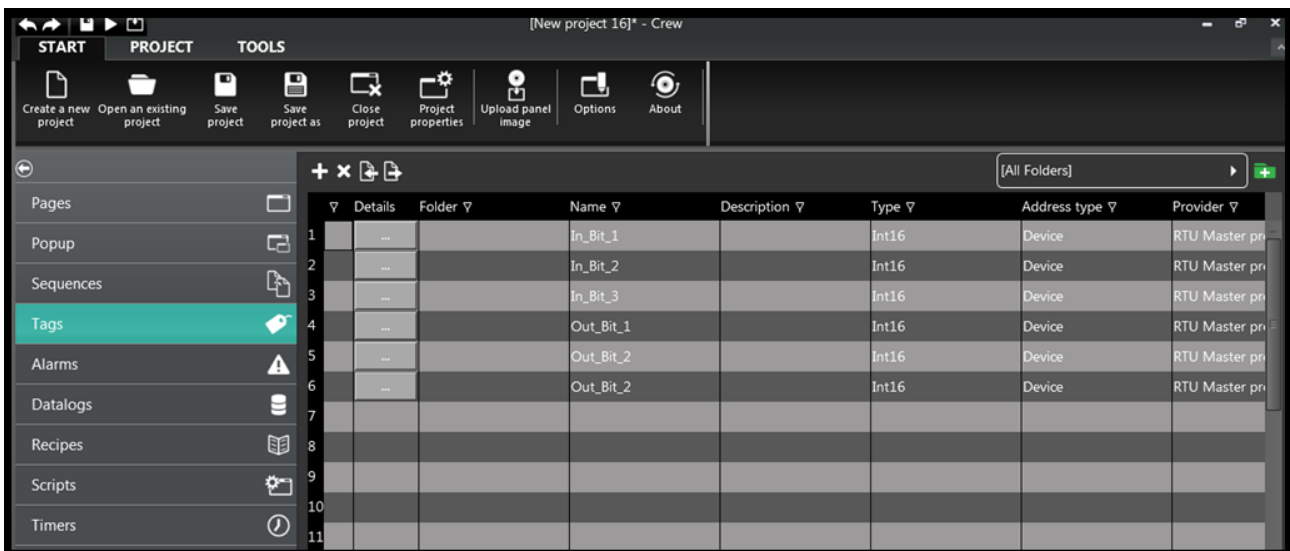
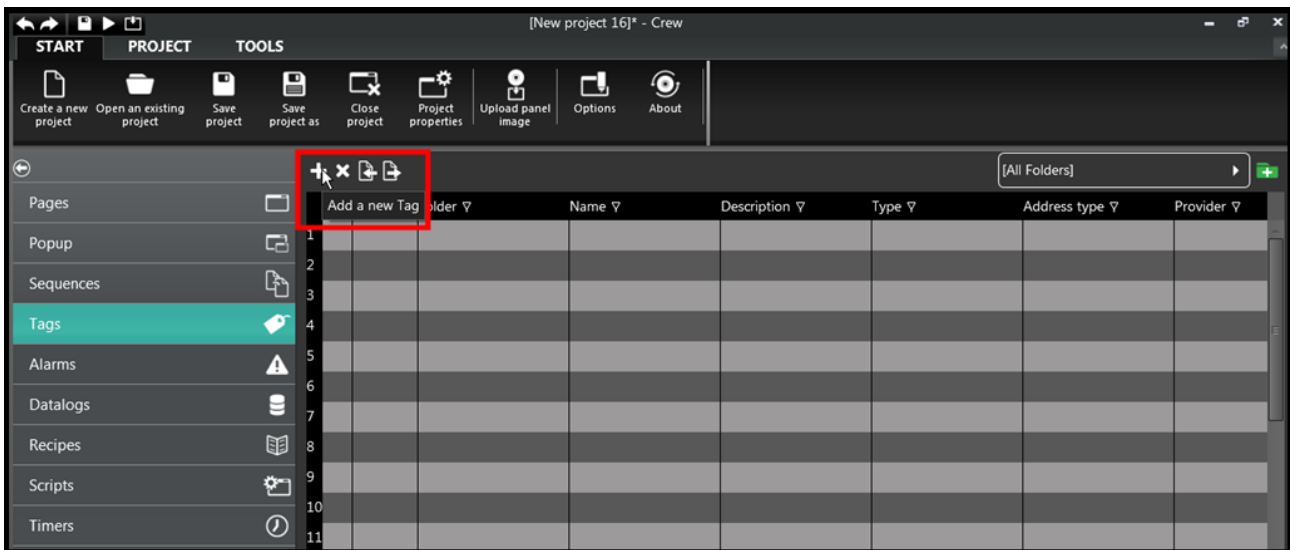
With Crew you can add and manage a new product page in a very simple manner.



# CREW Manual

## Creating a variable or Tag

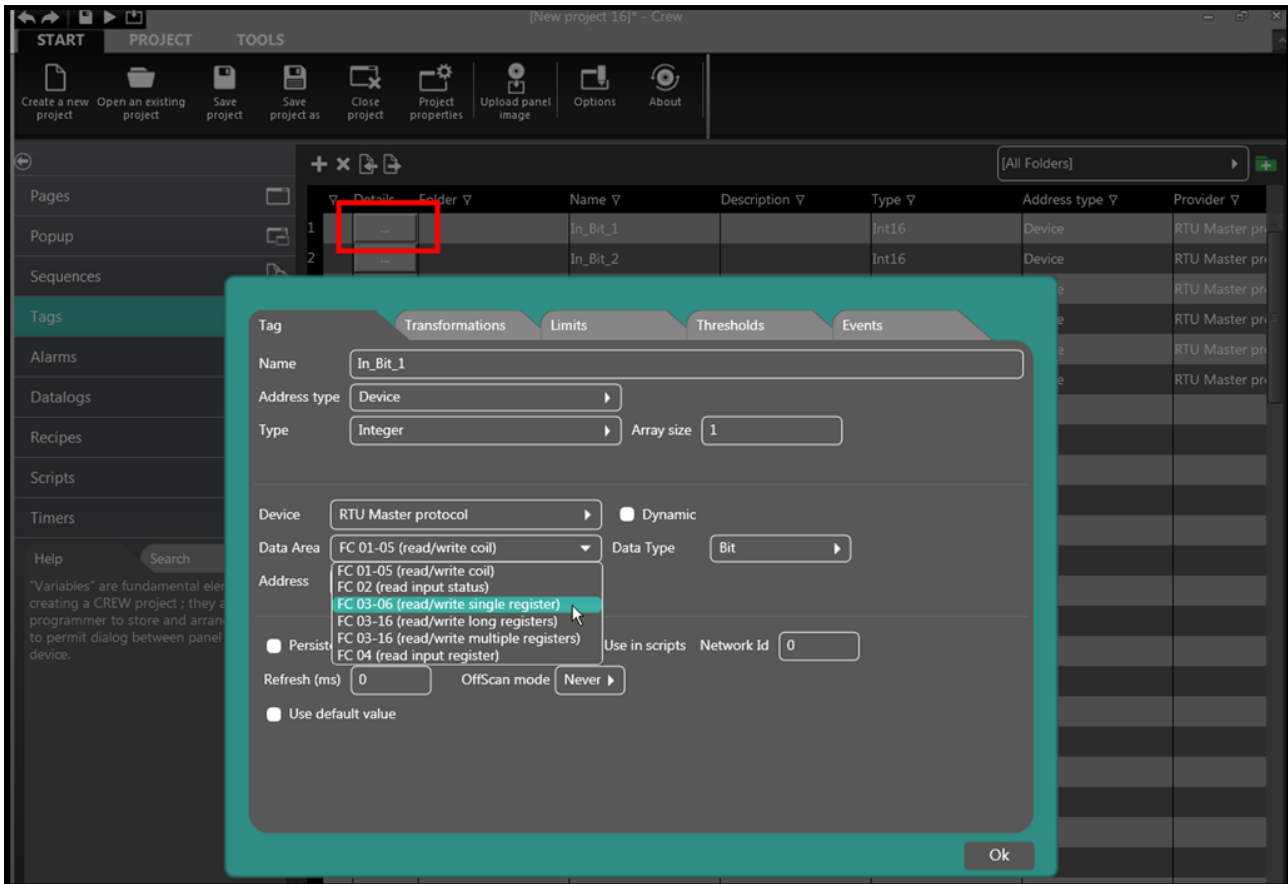
A series of variables (Tags) can be created and managed simply and intuitively. Use the “Tags” function to create one or more variables.





# CREW Manual

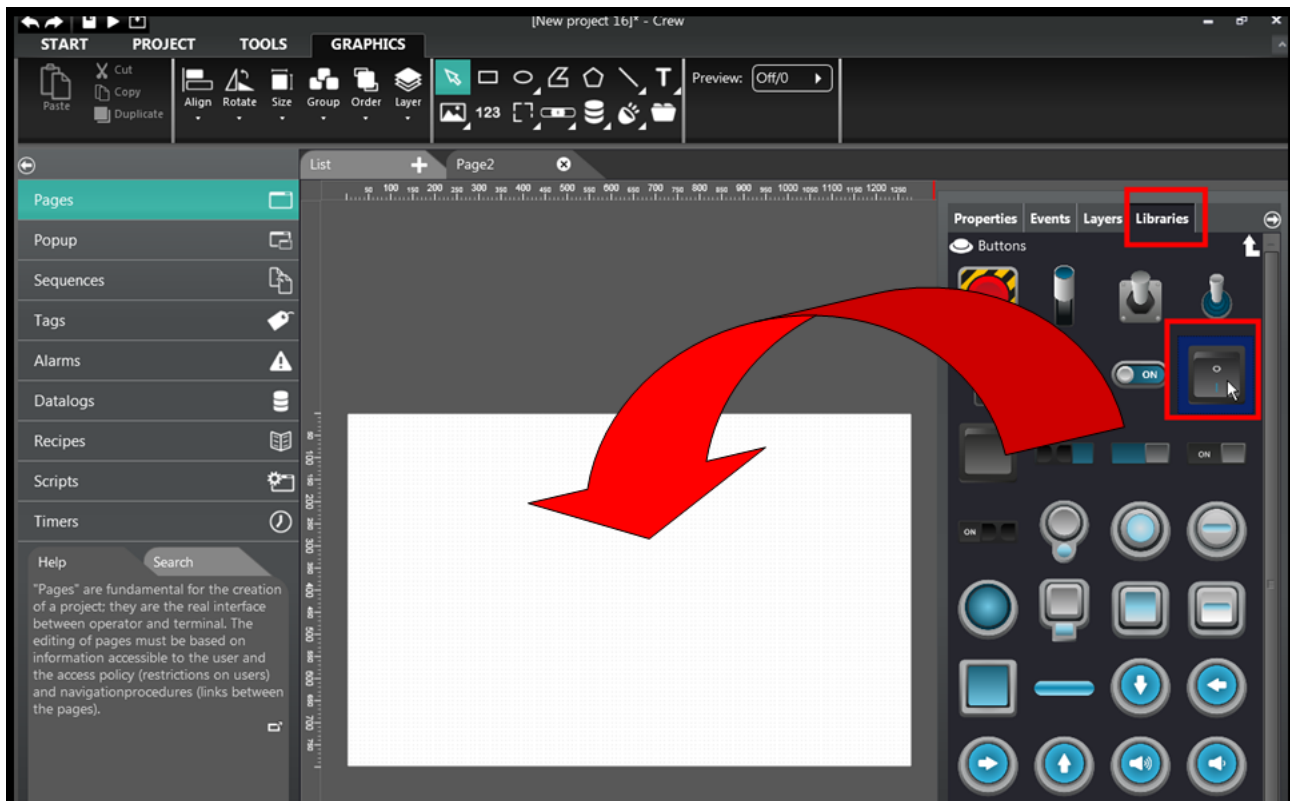
Various protocols can be used. For example, choose a “RTU Master” and select the required Modbus function for each tag.



# CREW Manual

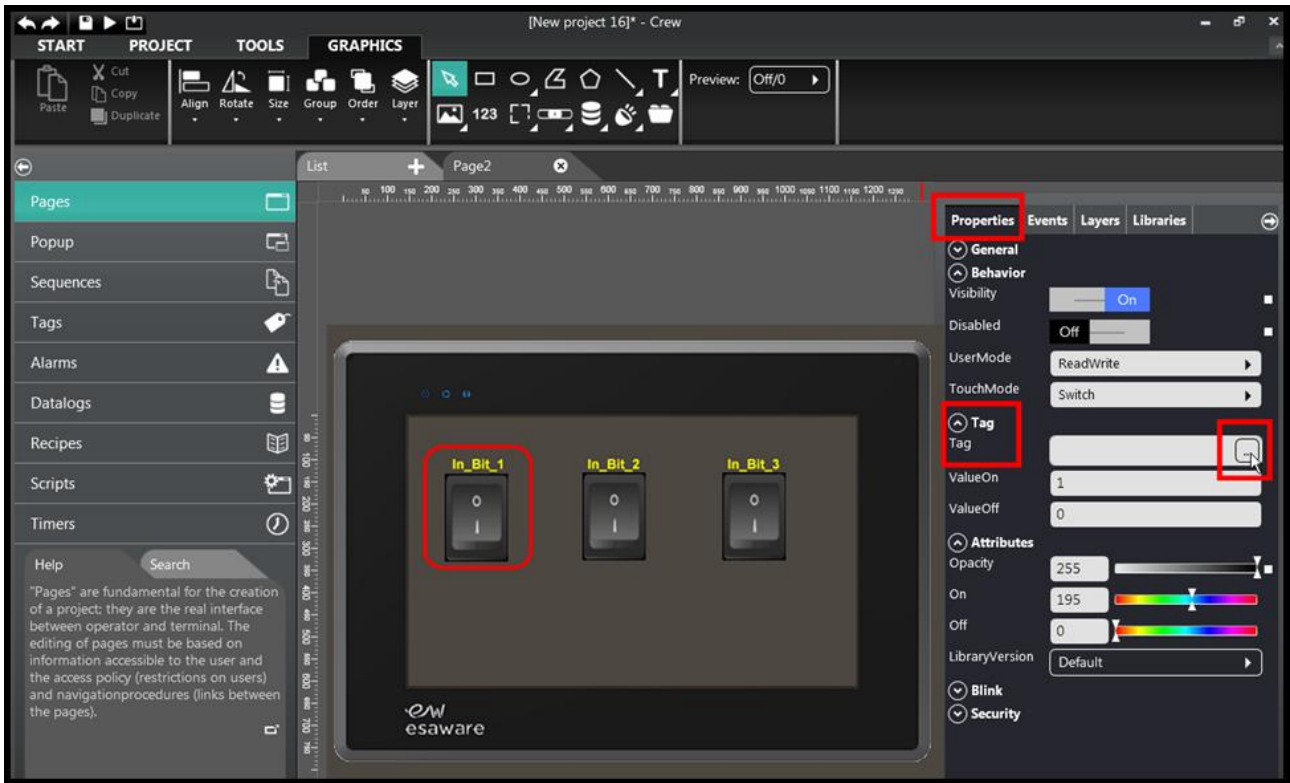
## Creating a Switch Button and associating it to a tag

Every object in the library can be easily placed (drag & drop) in a project page.

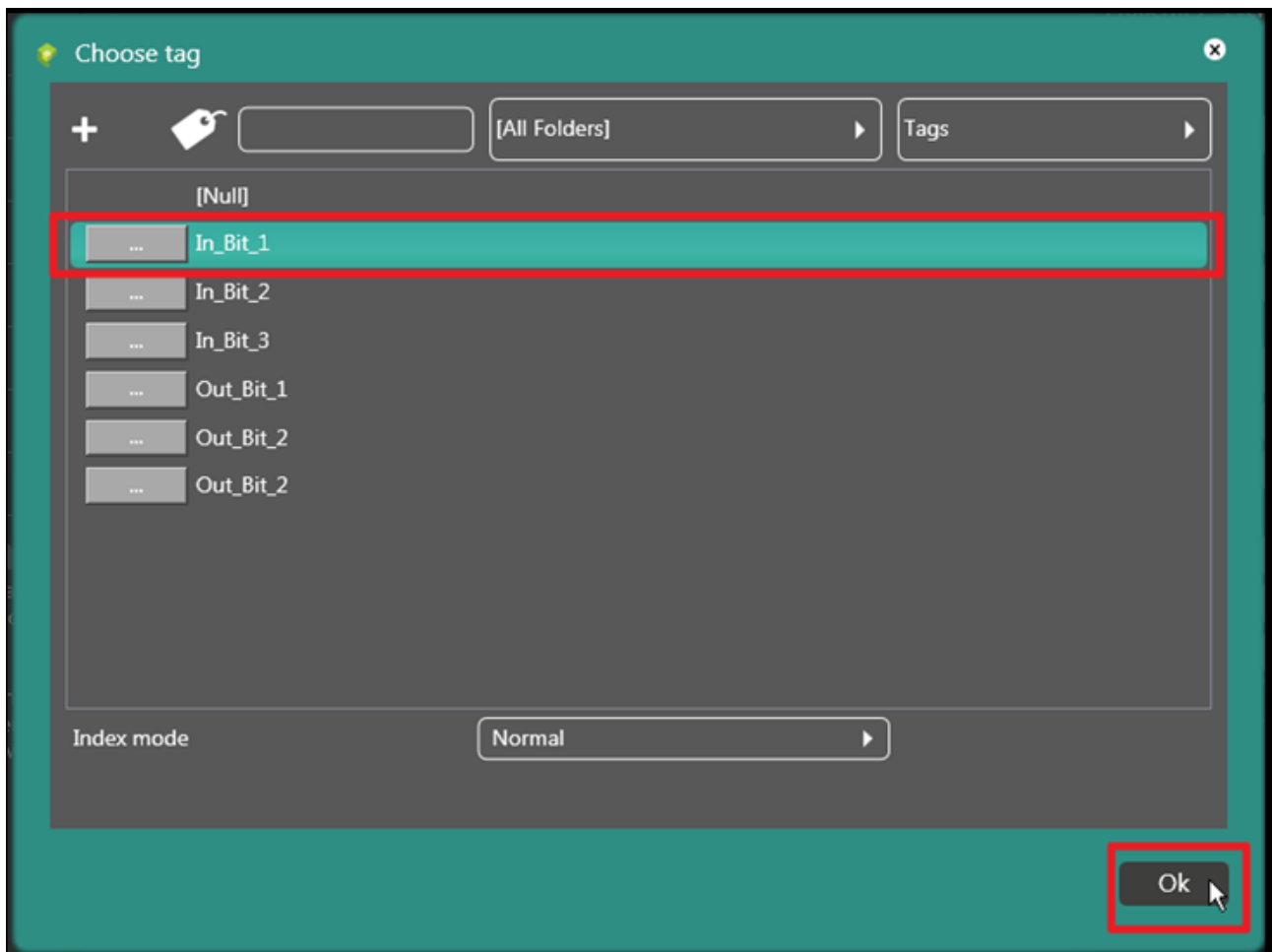


# CREW Manual

After selecting the Switch Button, click “Property”, then “Tag” and lastly “Browse” to choose which variable to associate.



# CREW Manual

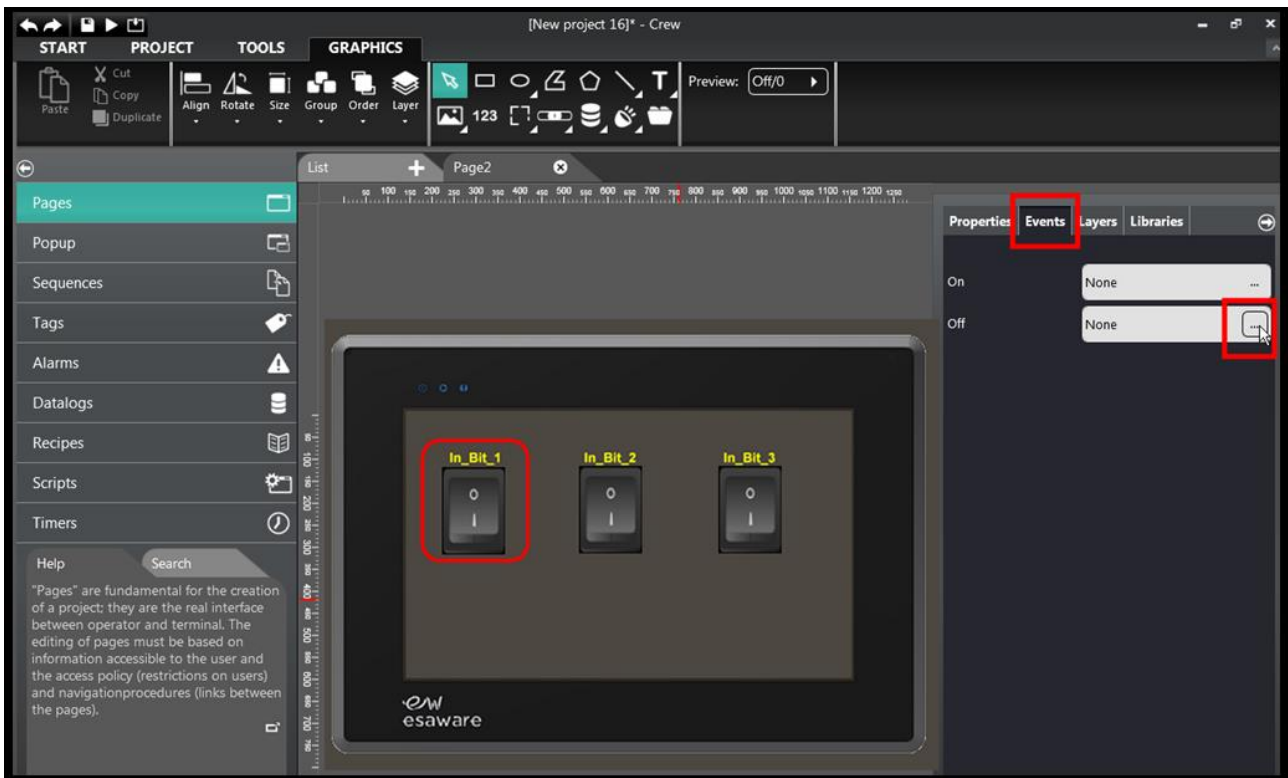


# CREW Manual

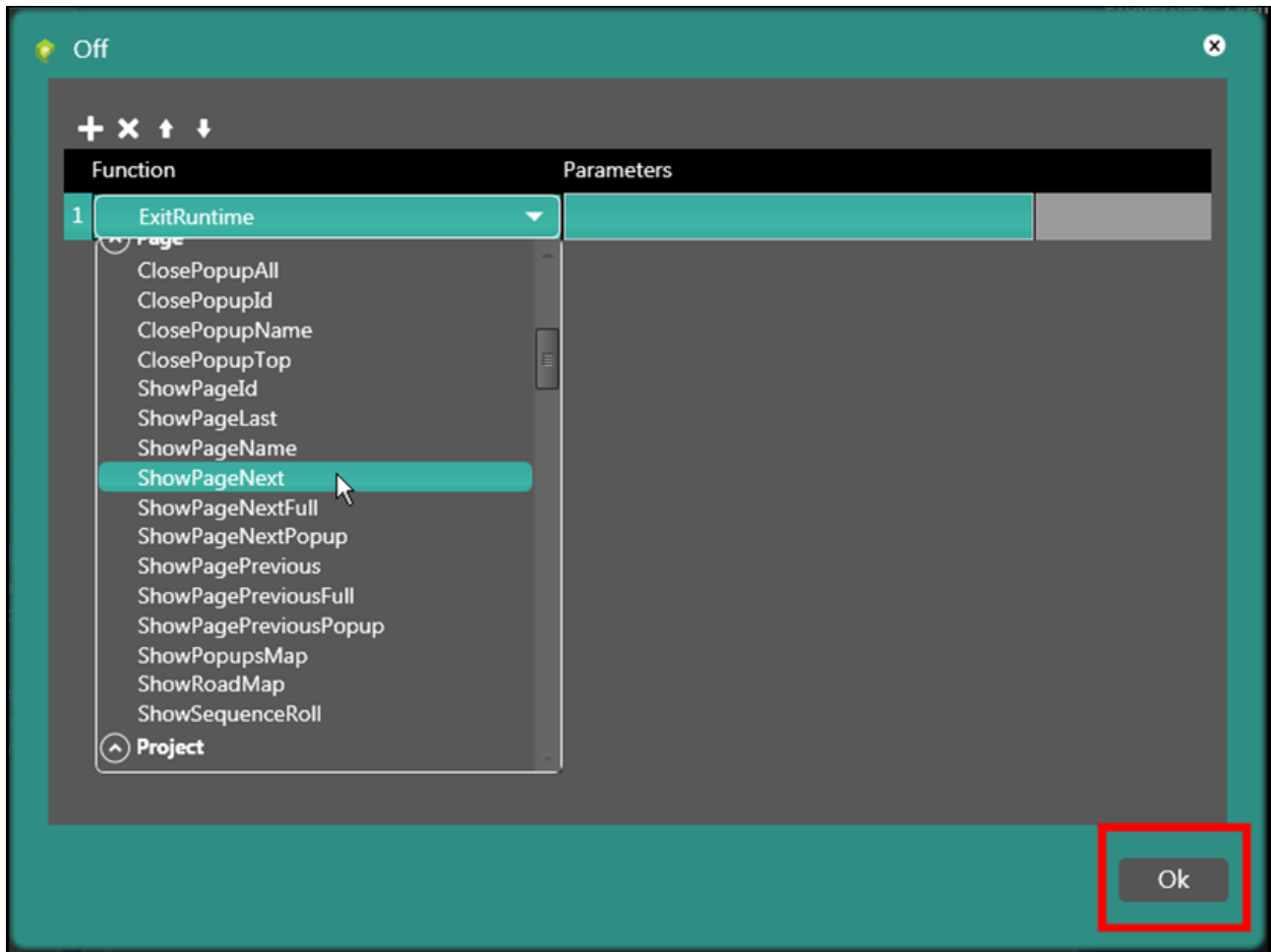
## Object events

Every object on a page can have Scripts and Functions associated to it.

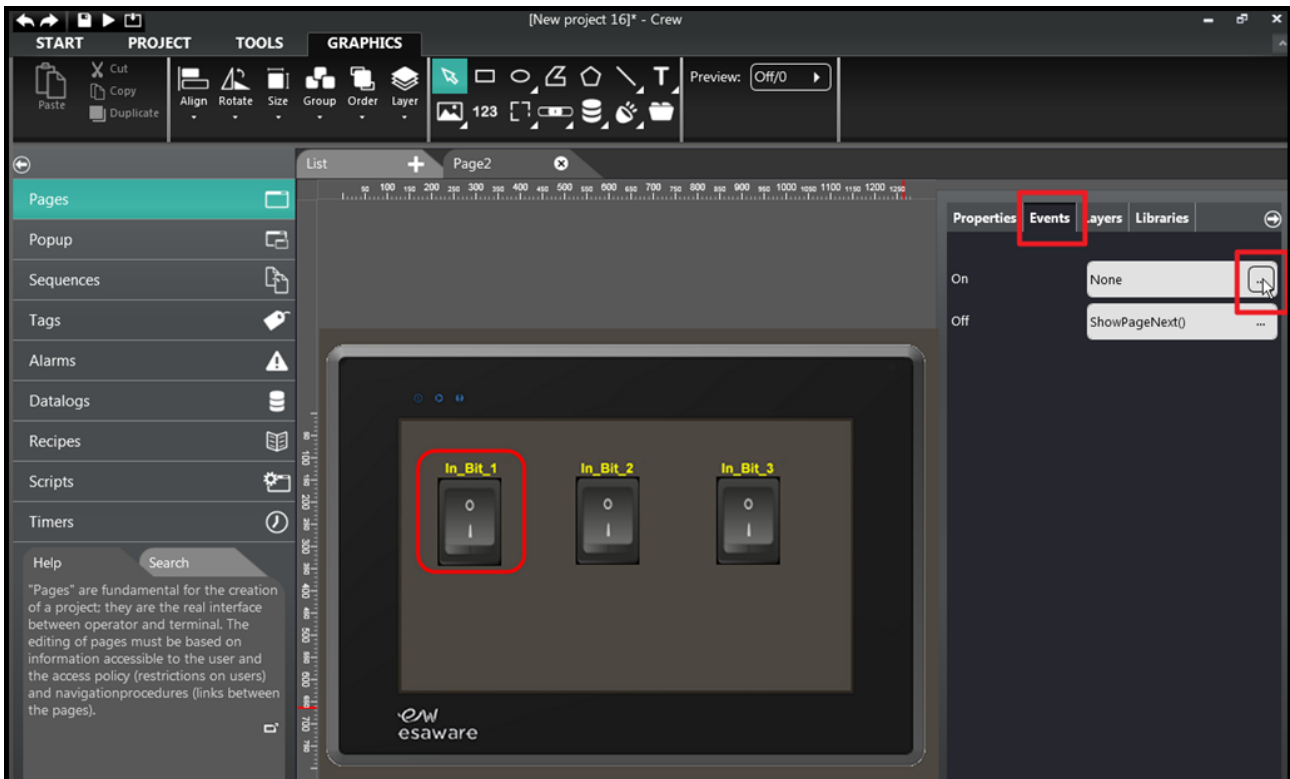
Below are examples of associating functions “ShowPageNext” and “BitSet”) to a Switch Button.



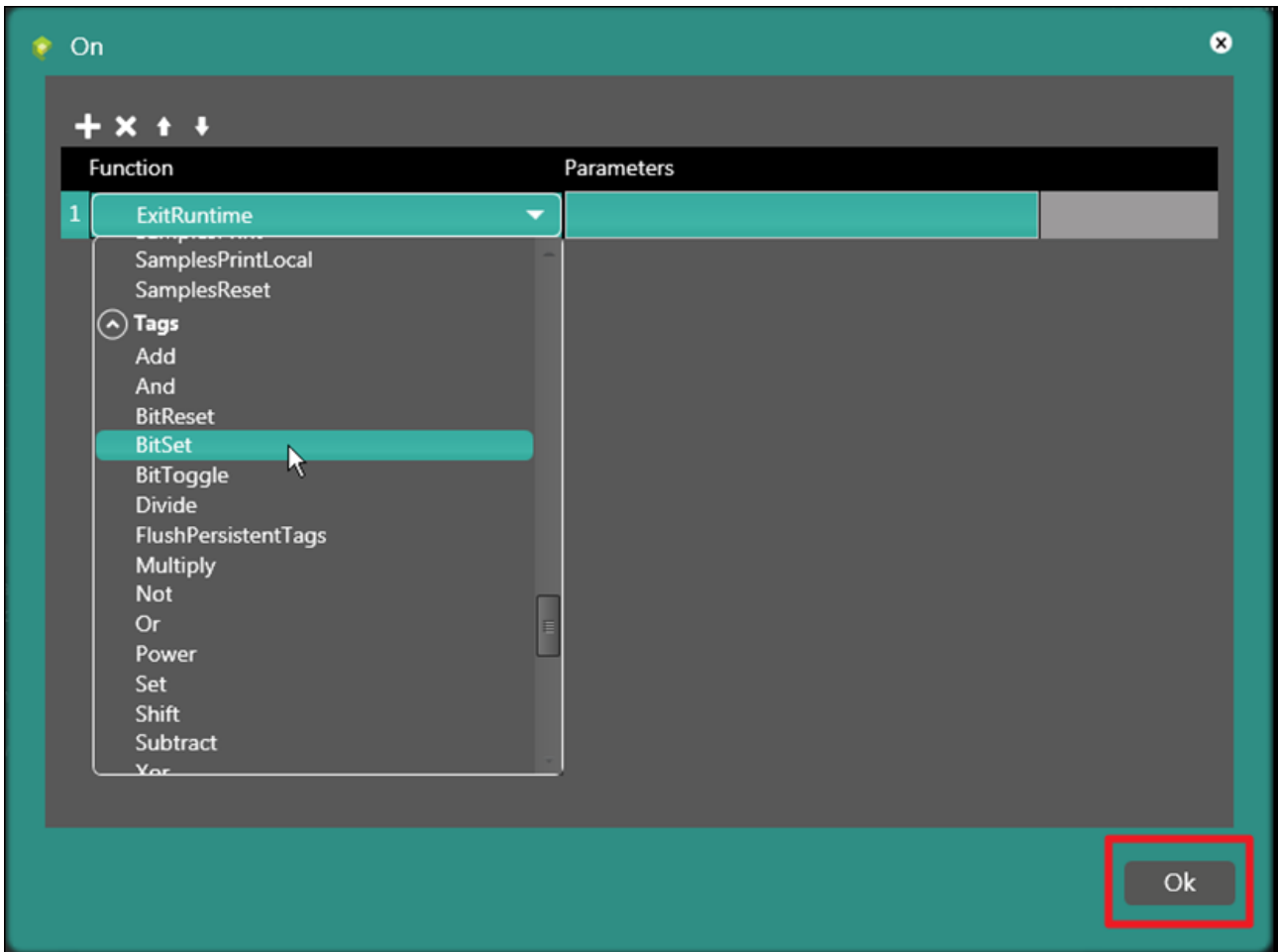
# CREW Manual



# CREW Manual



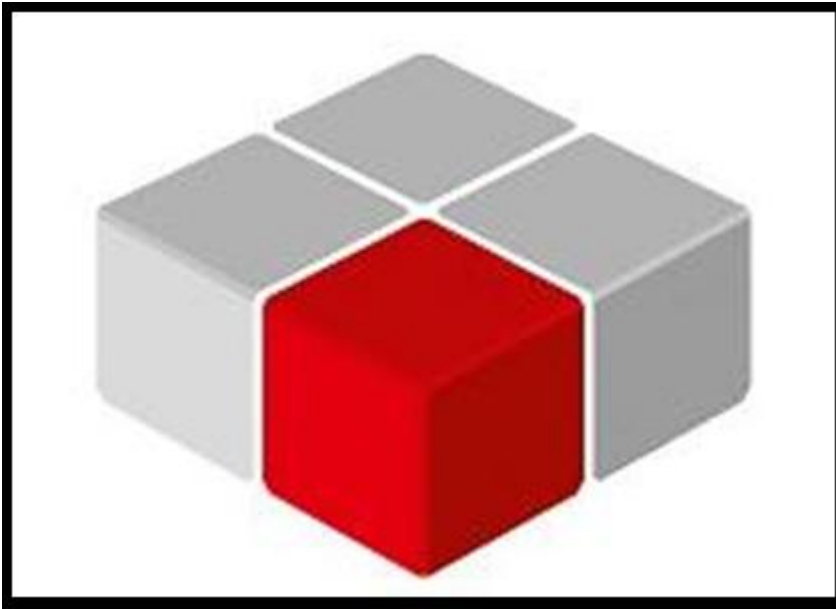
# CREW Manual





# CREW Manual

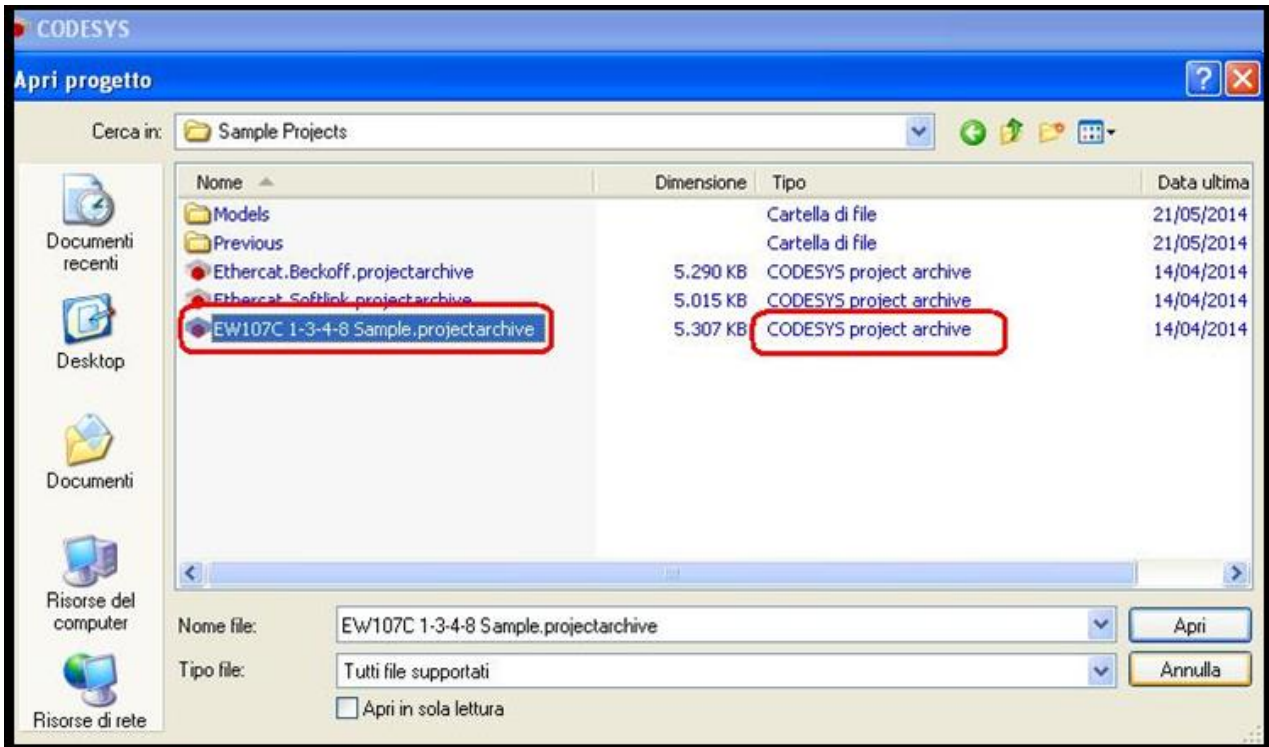
## CODESYS settings



Installing CODESYS 3.x

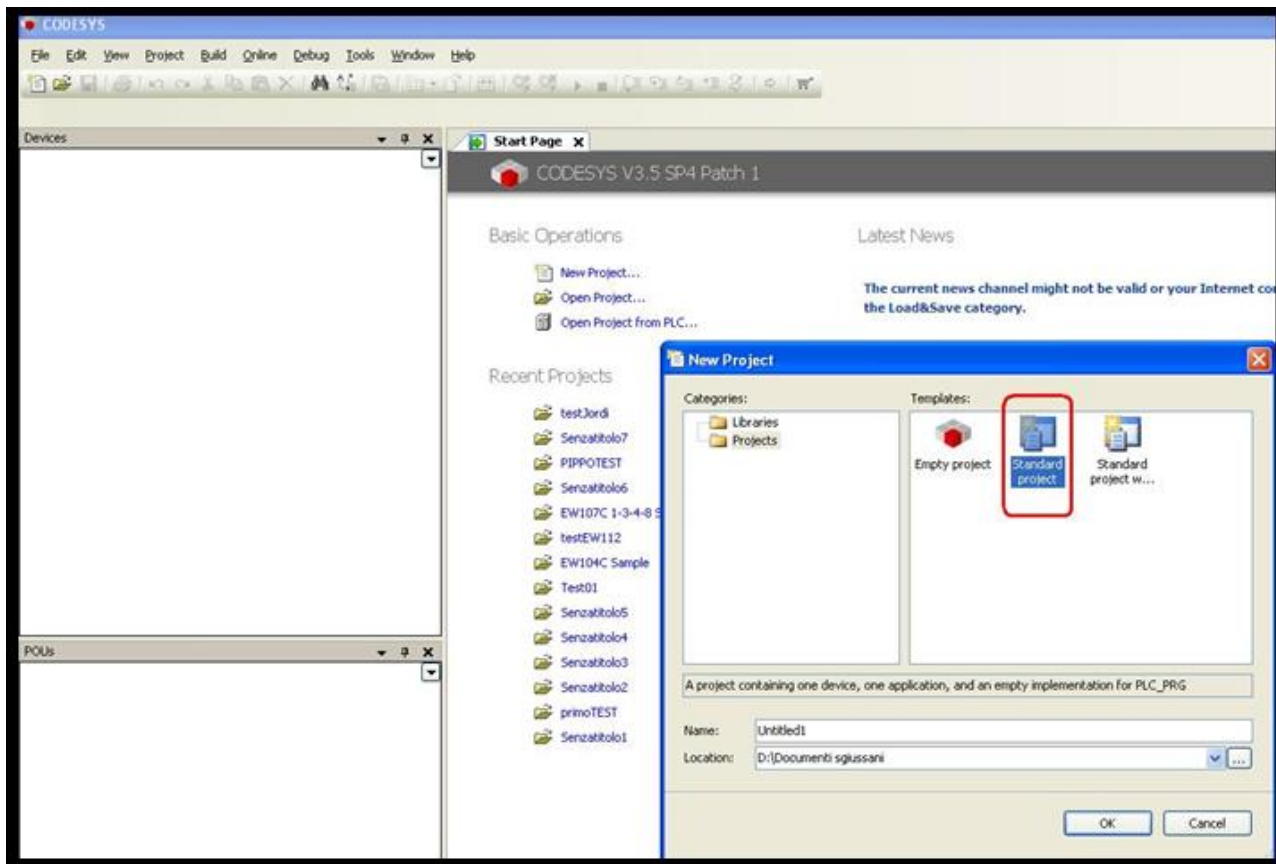
# CREW Manual

The first time it is necessary to open the Project Archive provided by ESA Automation, so as to install (if required) some CODESYS libraries.



# CREW Manual

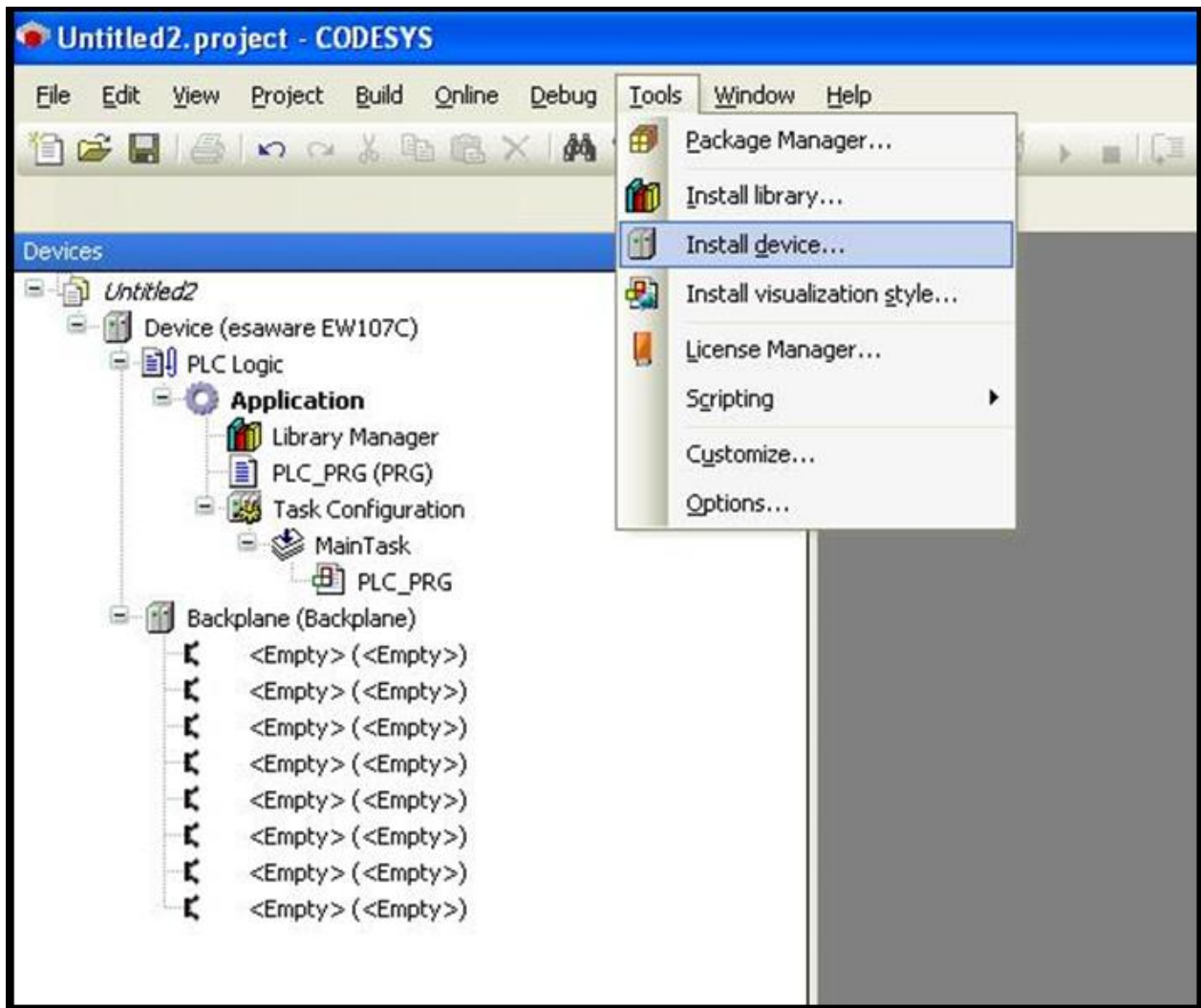
## Opening the Standard Project



# CREW Manual

## Installing ESA-LIBRARIES (XML-file)

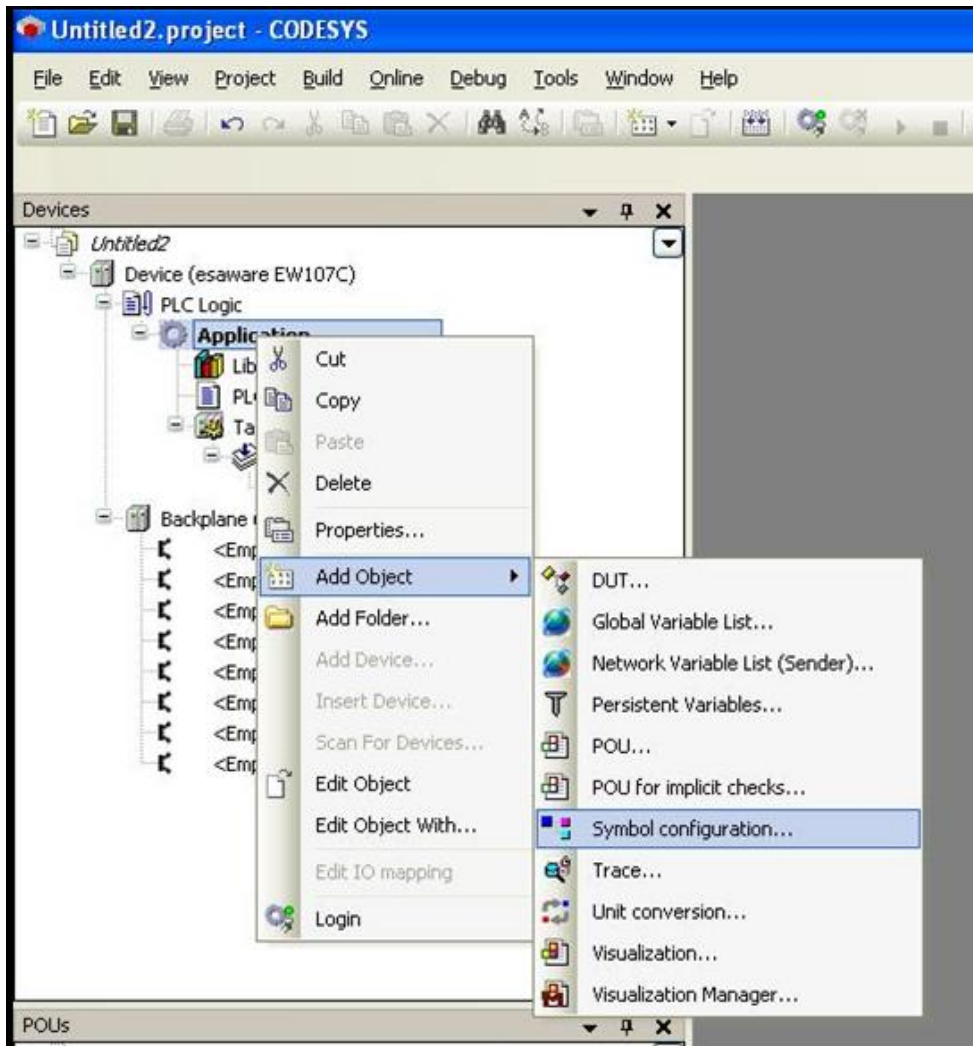
From “Tools” choose “Install device” to configure all of the ESA libraries (XML-file), step by step.



# CREW Manual

## Symbol Configuration

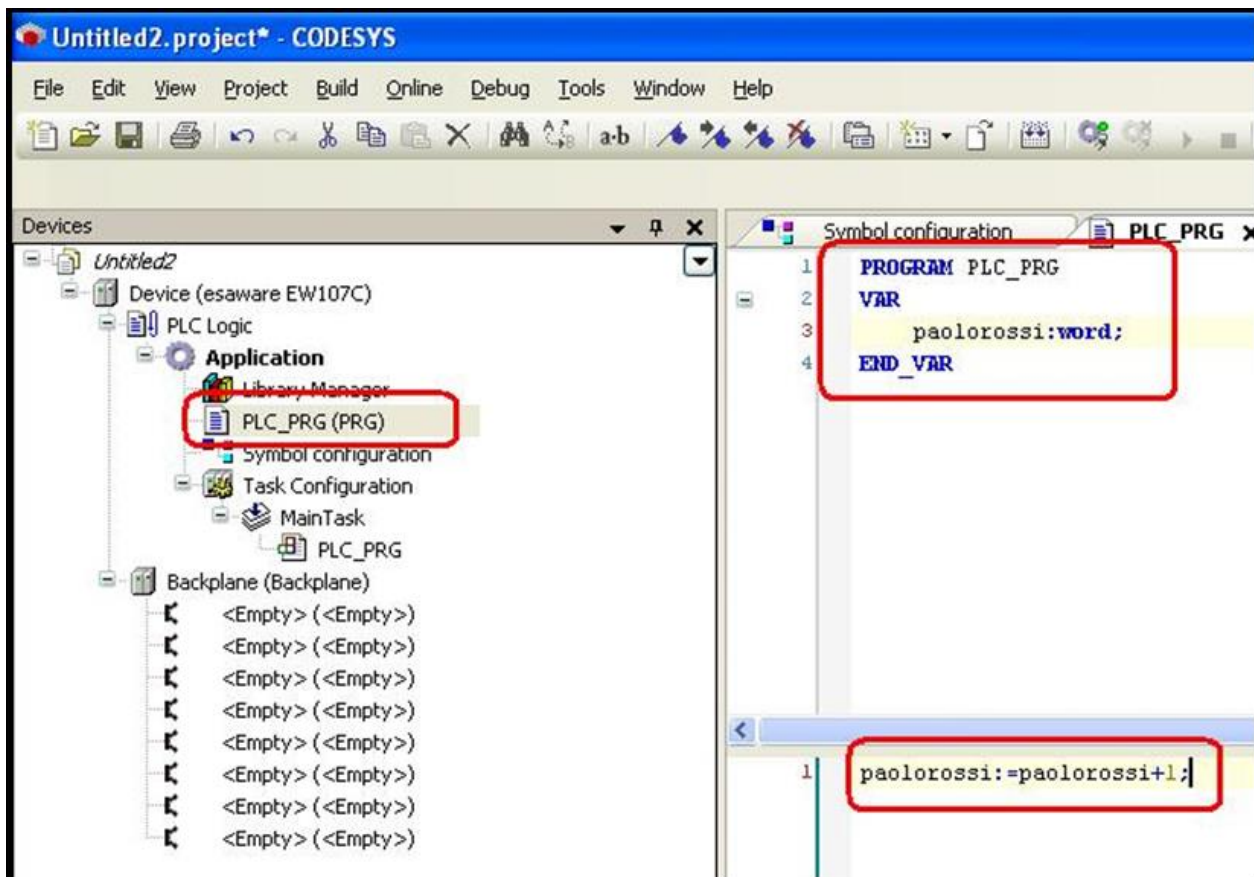
From “Application” go to “Add Object” and choose “Symbol Configuration”.



# CREW Manual

## Creating Tag + PRG program

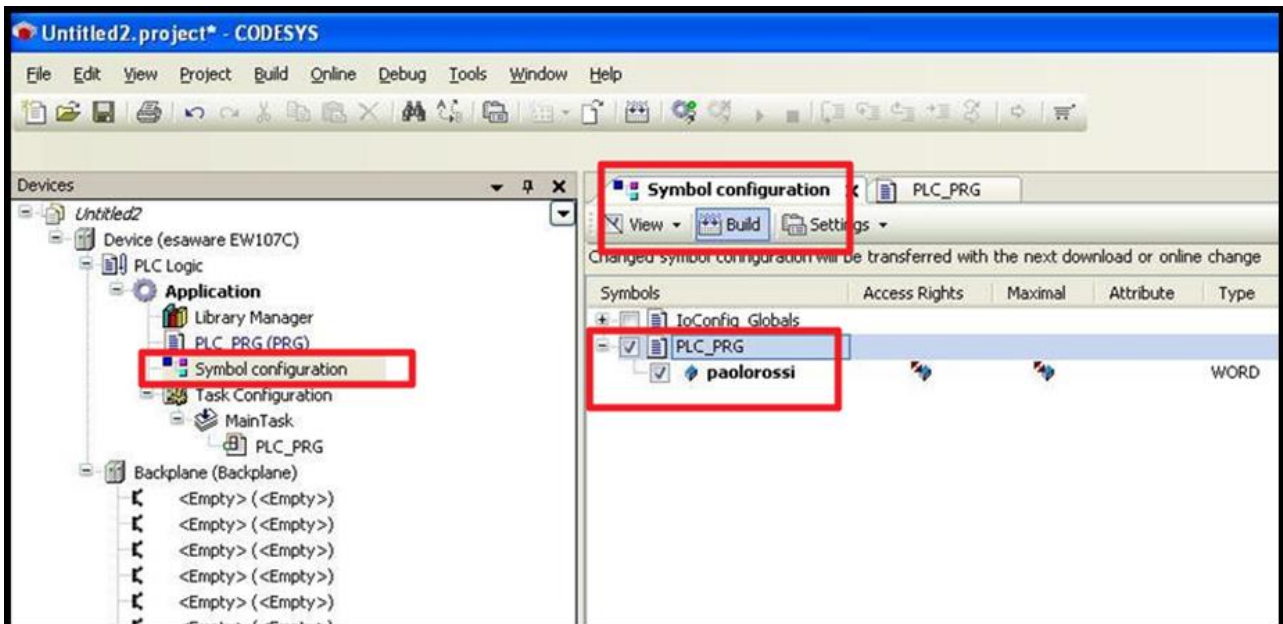
From "PLC\_PRG" define the project tags.



# CREW Manual

## Build up SYMBOL

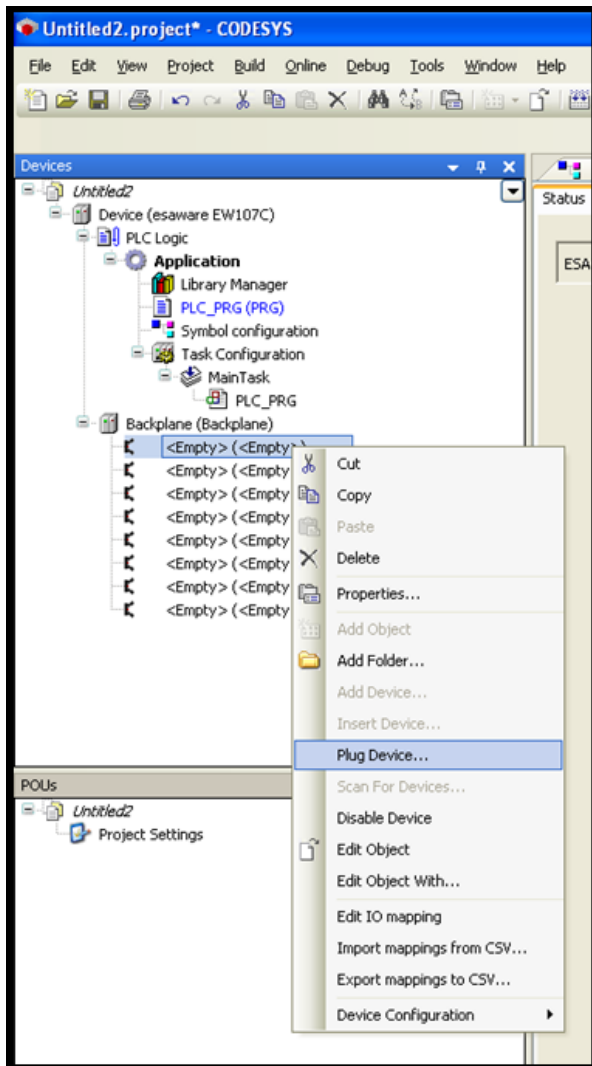
From “Symbol configuration”, make it.



# CREW Manual

## Add ESA I/O (Plug Device)

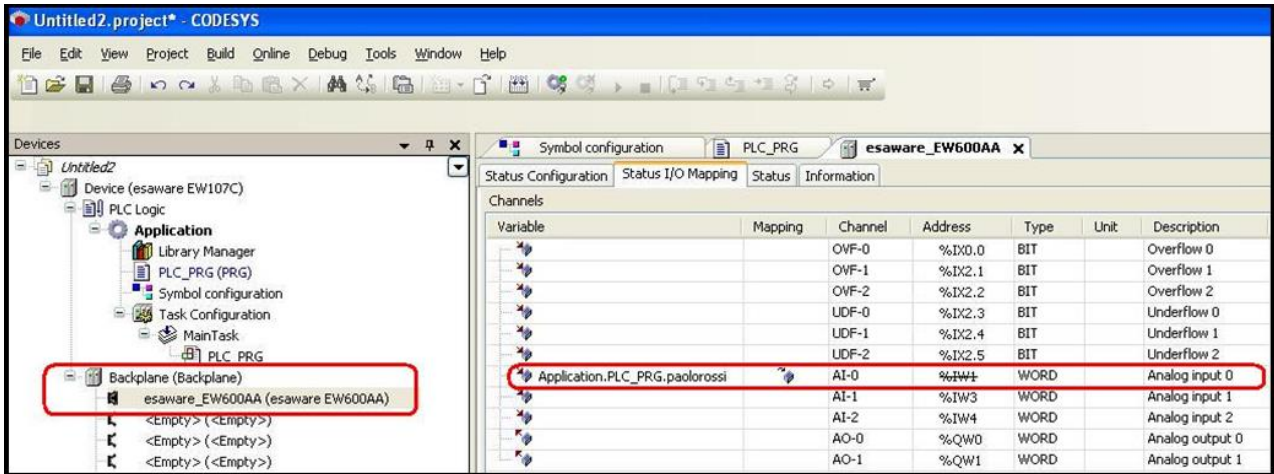
If there are Esaware I/O (EW600), add “ESA I/O (Plug Device)”.





# CREW Manual

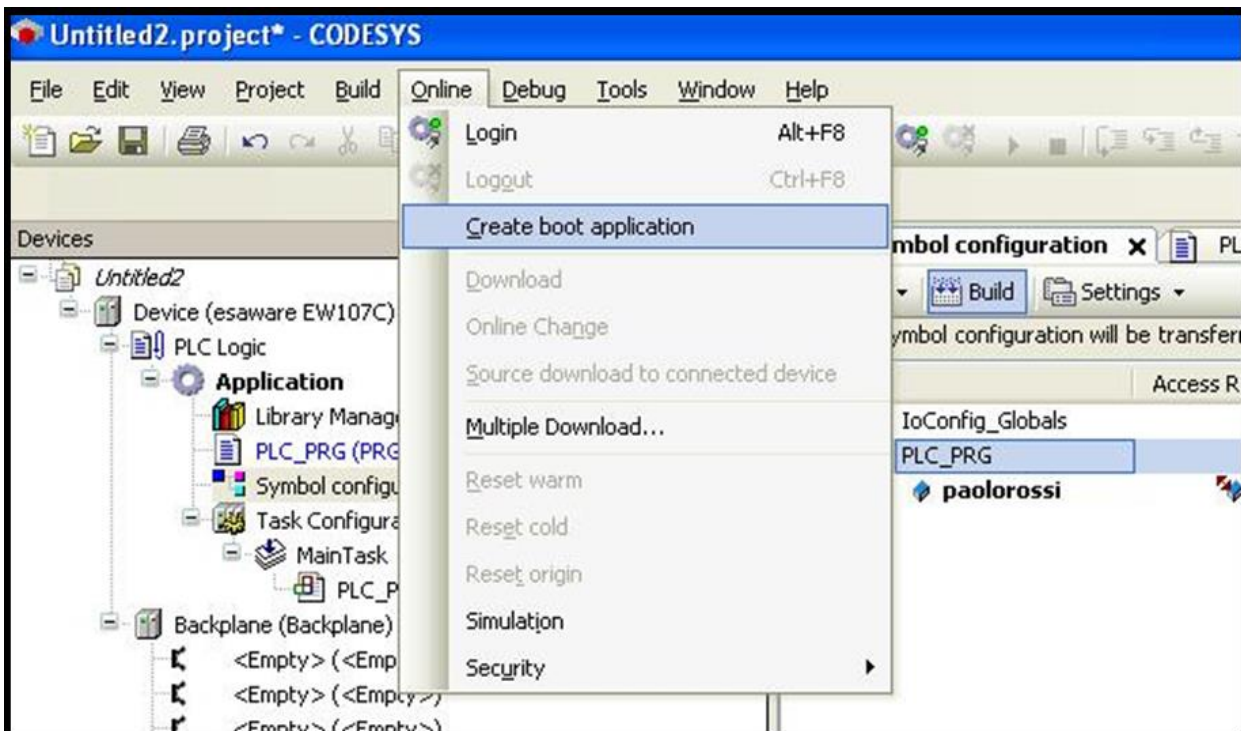
Associate the program tags to the I/O signals



Make the project (F11)

# CREW Manual

Create the application boot file



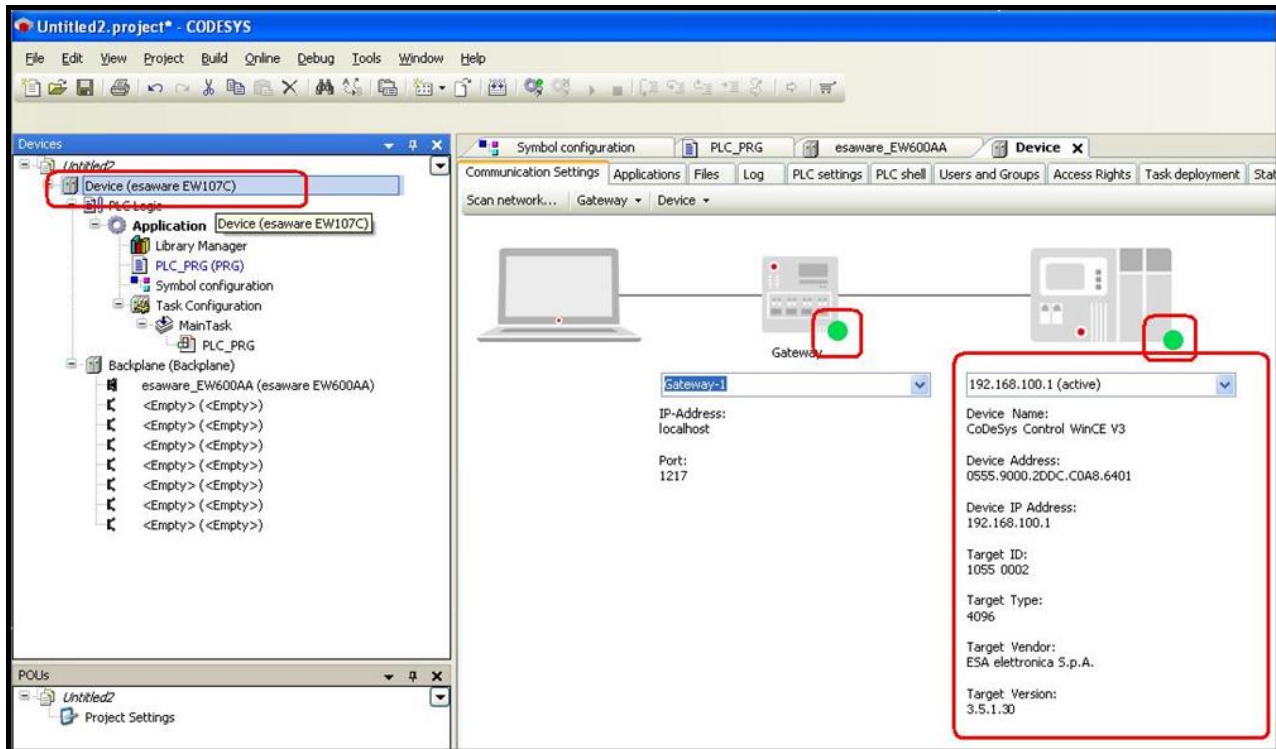
Save project as...



Note: You now have three CODESYS files (.PROJECT/.APP/.XML) available, ready for import to the Crew project (Soft PLC) and downloaded on the panel. It is now already possible to run the test for correct communication between CODESYS and the EWxxx Soft PLC Esaware panel.

# CREW Manual

To do so, it is necessary to perform the operation illustrated in the image.



ONLINE >> LOGIN to download the application on the EWxxx SOFT PLC panel.



Note: The addition of “ESA I/O (Plug Device)” I/O devices and the association of program tags to the I/O signals must be carried out when using an ESA EW1xxACxxx panel.



Note: The correct communication test and downloading the application on the EWxxx SOFT PLC ESA panel are carried out for test purposes. It is possible to leave them out and proceed to the Crew section.

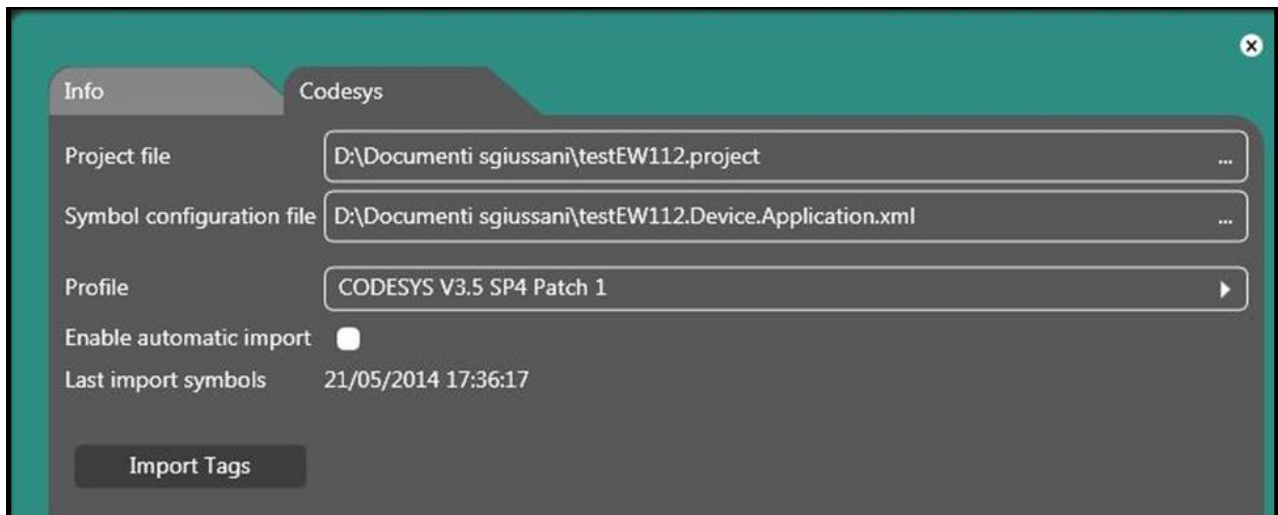
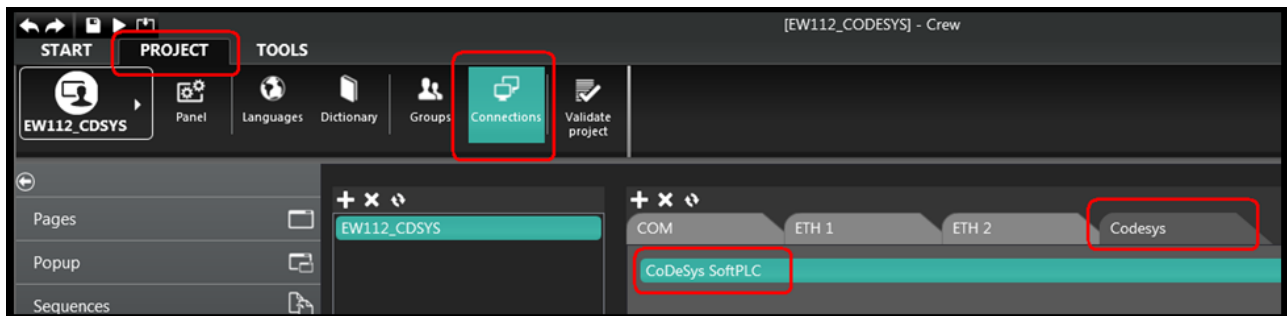
# CREW Manual

## Crew Section



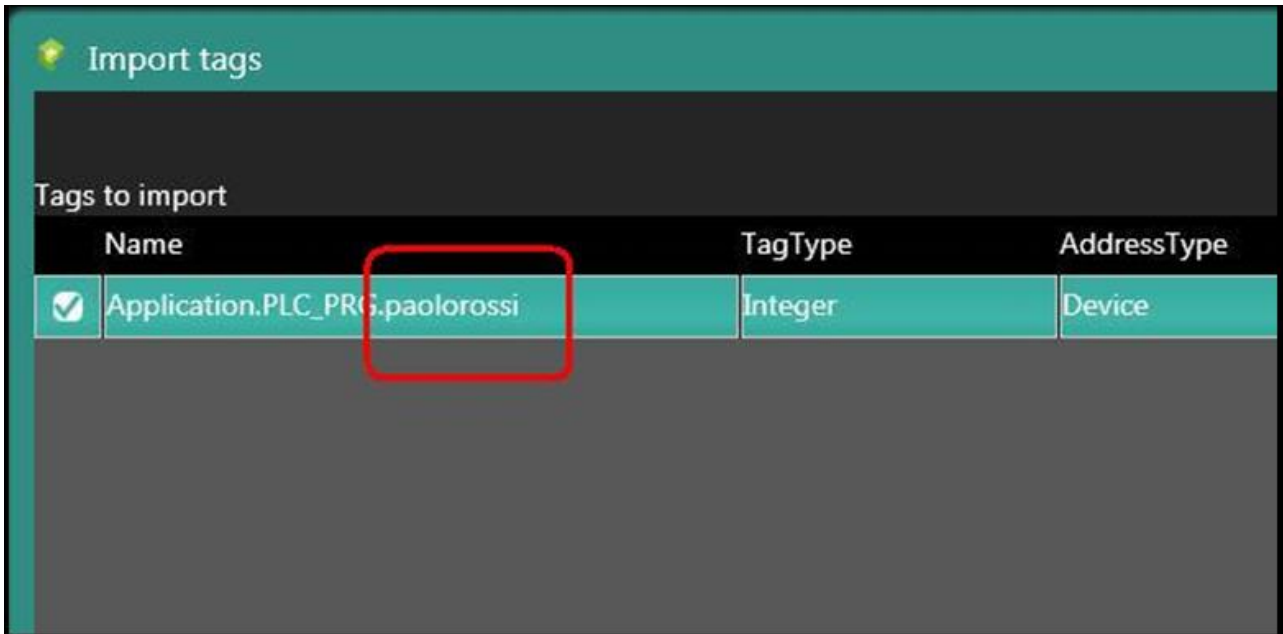
Note: To create a Crew project refer to "Esaware Crew - Basic settings -". The following phases refer to the CODESYS parts in a Crew project.

After installing the Crew software and creating a project with an EW1B panel (SOFT PLC), or with an EW1C (SOFT PLC + ESA I/O EW600), it is possible to import the files (.PROJECT/.XML) from CODESYS to Crew.



# CREW Manual

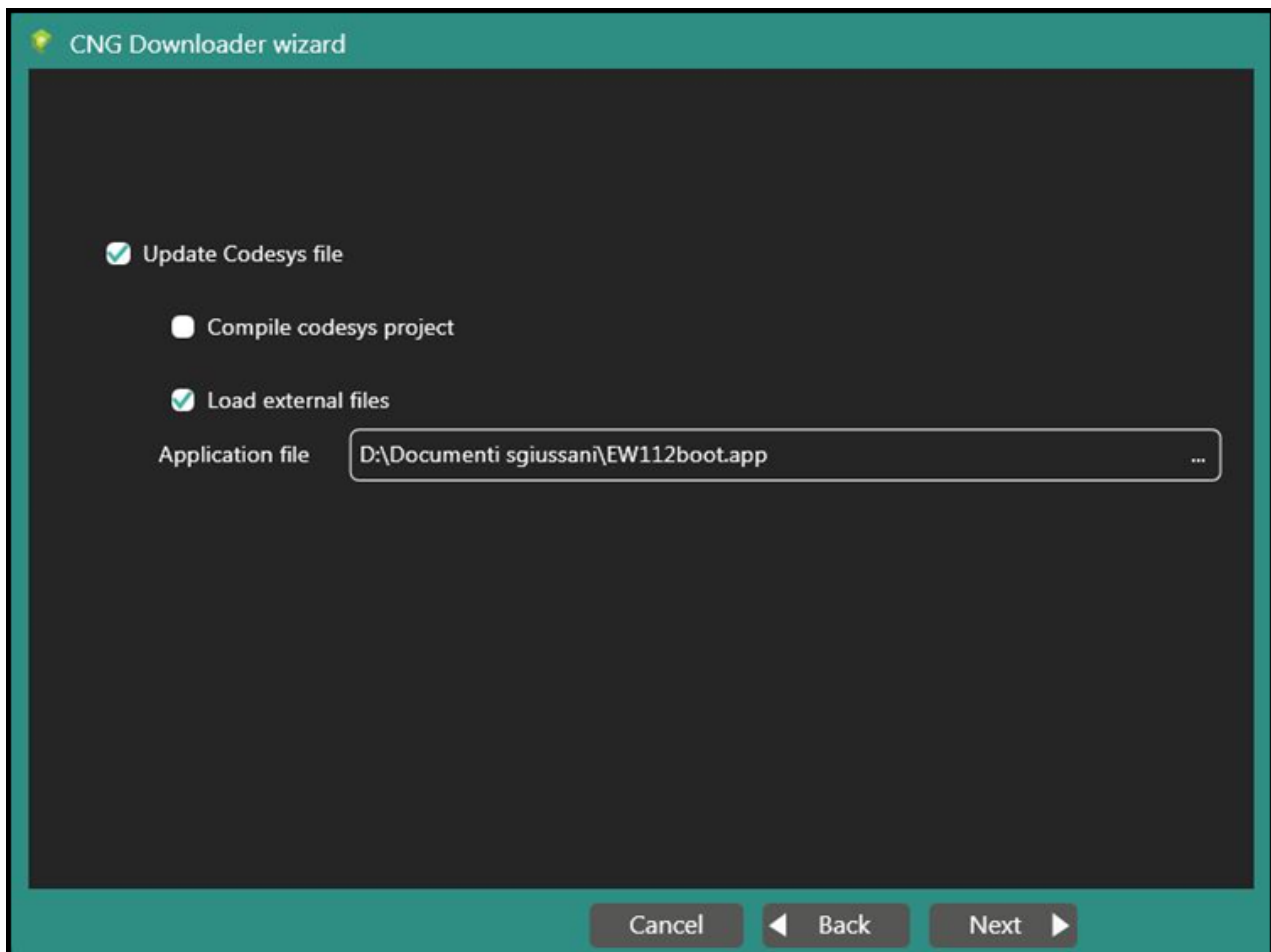
IMPORT TAGS is used to view the tags created in the CODESYS project.



It is now possible to download the Crew project to the EW panel. After the download box it is possible to view the CODESYS options:

- 1) Close the CODESYS software and allow Crew to make the CODESYS project.
- 2) Download the CODESYS “boot application” (“.APP” file).

# CREW Manual



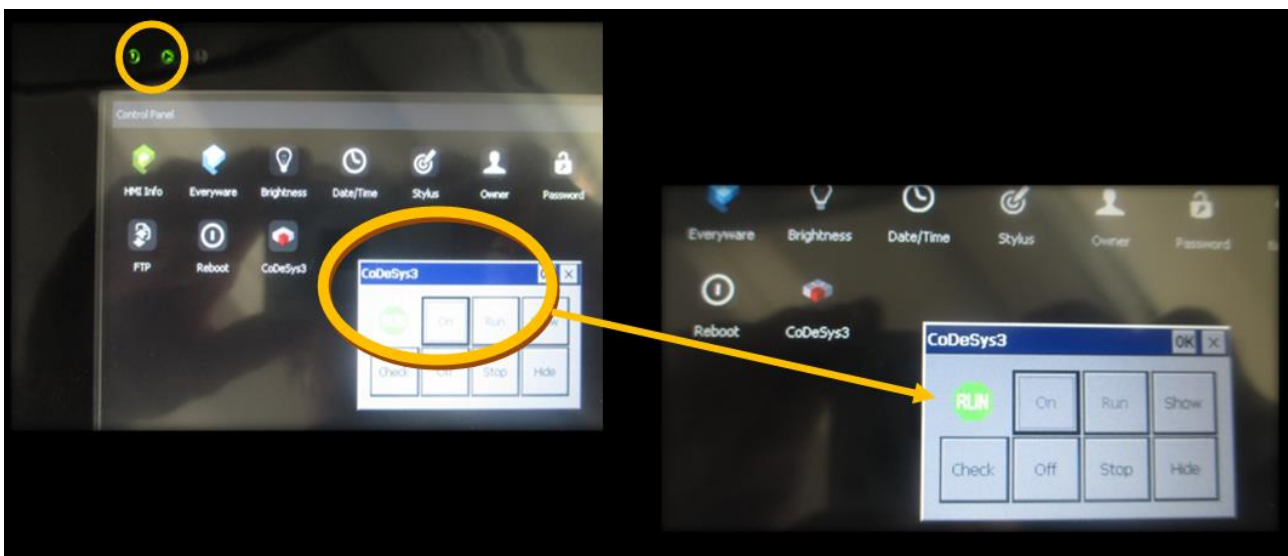
The Crew + CODESYS SOFT PLC projects now work together on the EW panel.

# CREW Manual

## EW panel

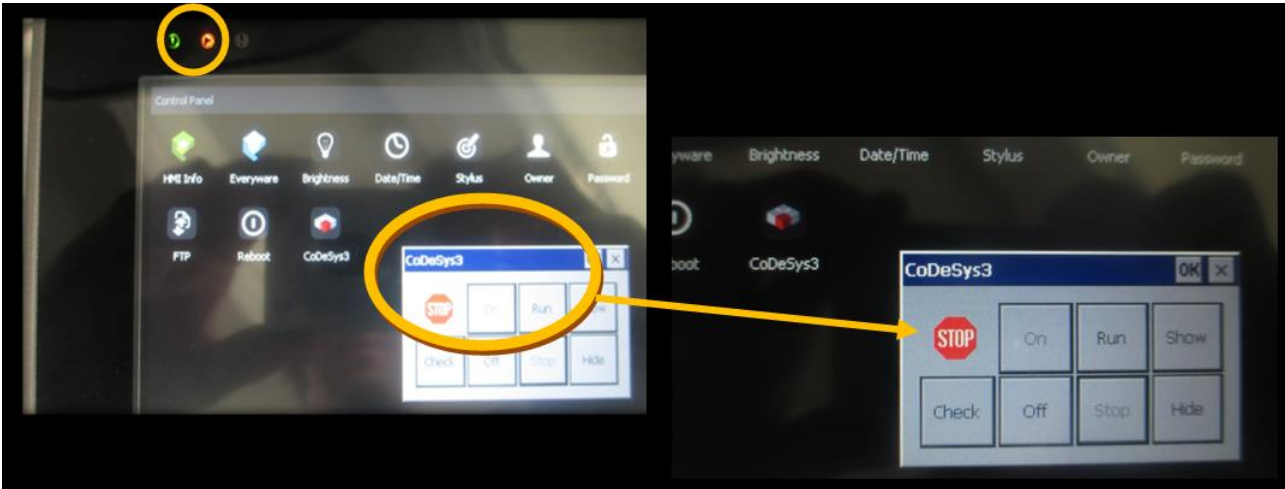
On the EW panel it is possible to set various SOFT PLC operating modes.  
Below are the main settings.

### RUN mode

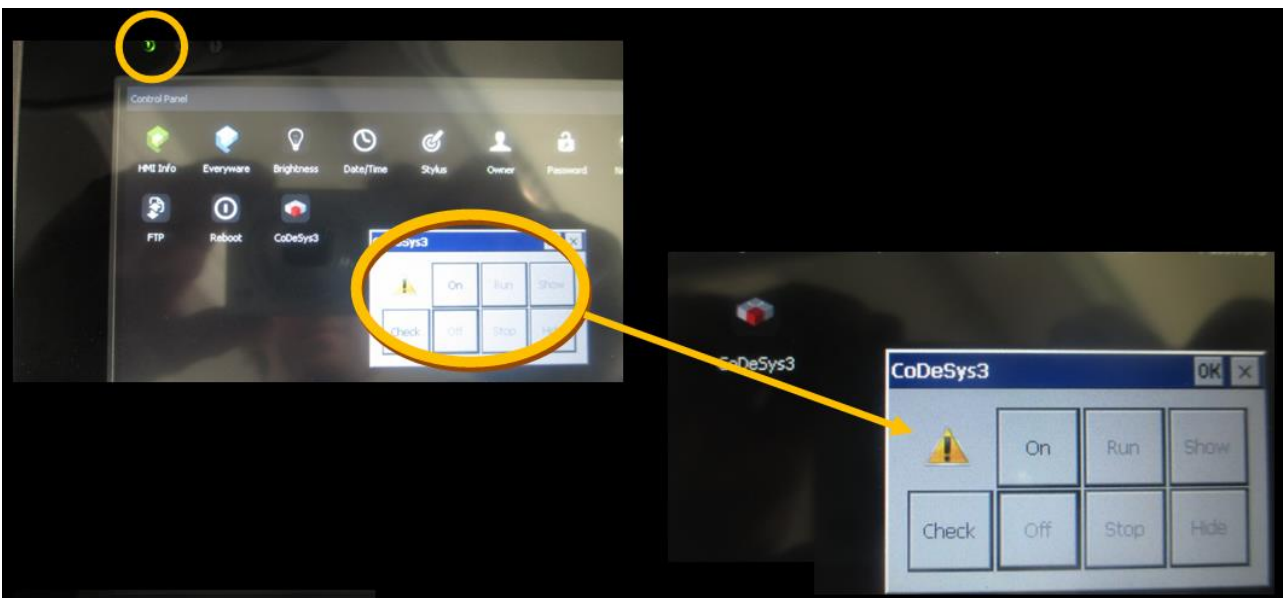


# CREW Manual

## STOP mode



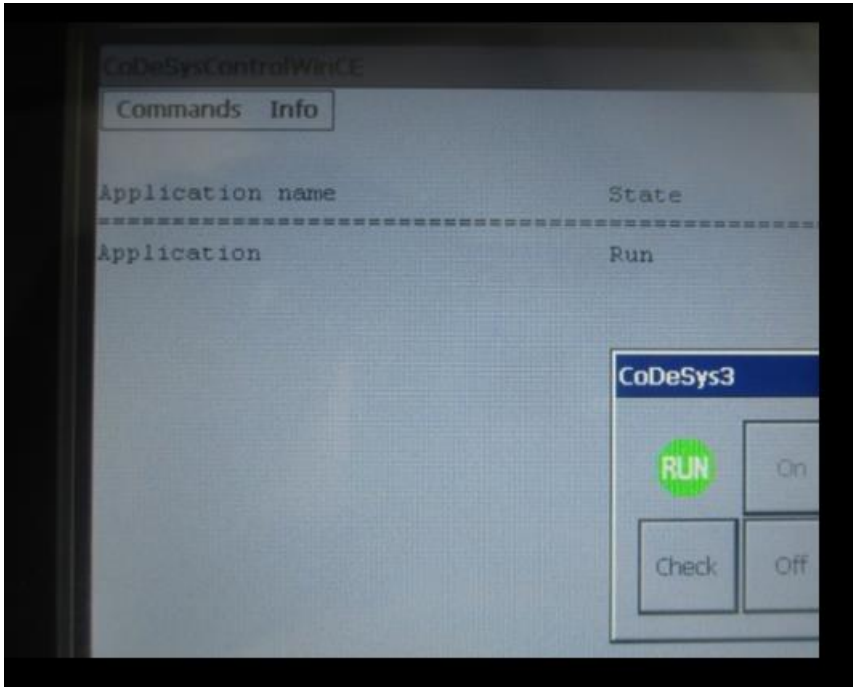
## OFF mode





# CREW Manual

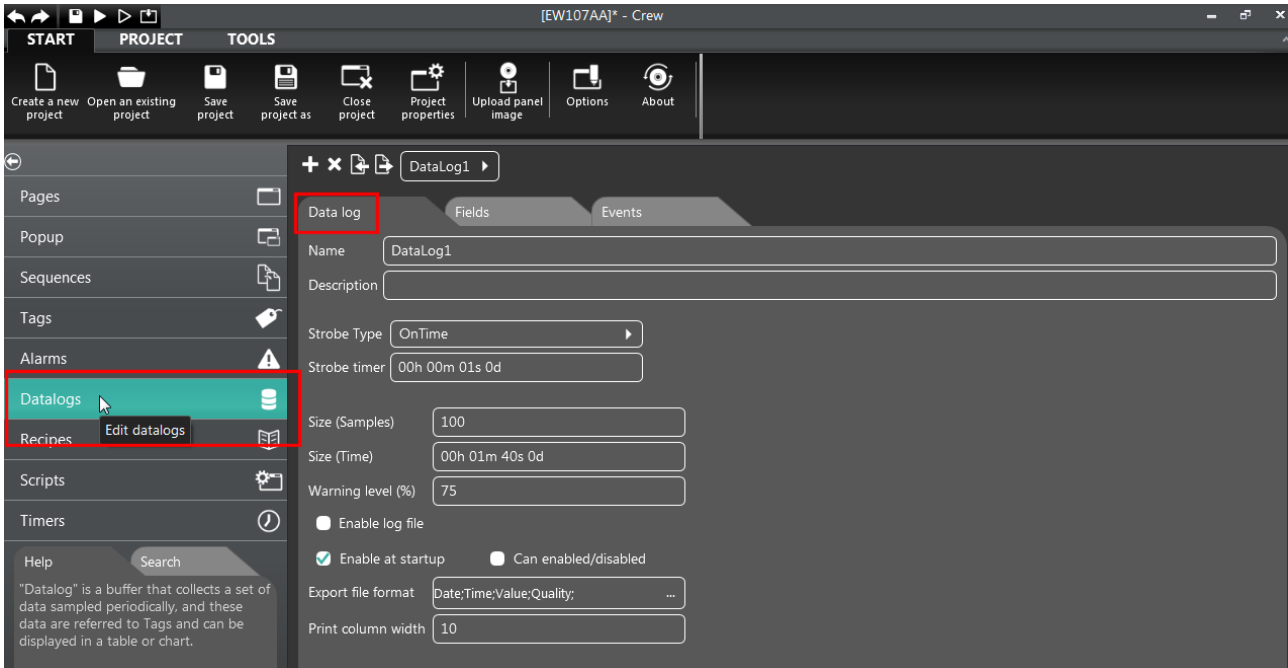
SHOW mode



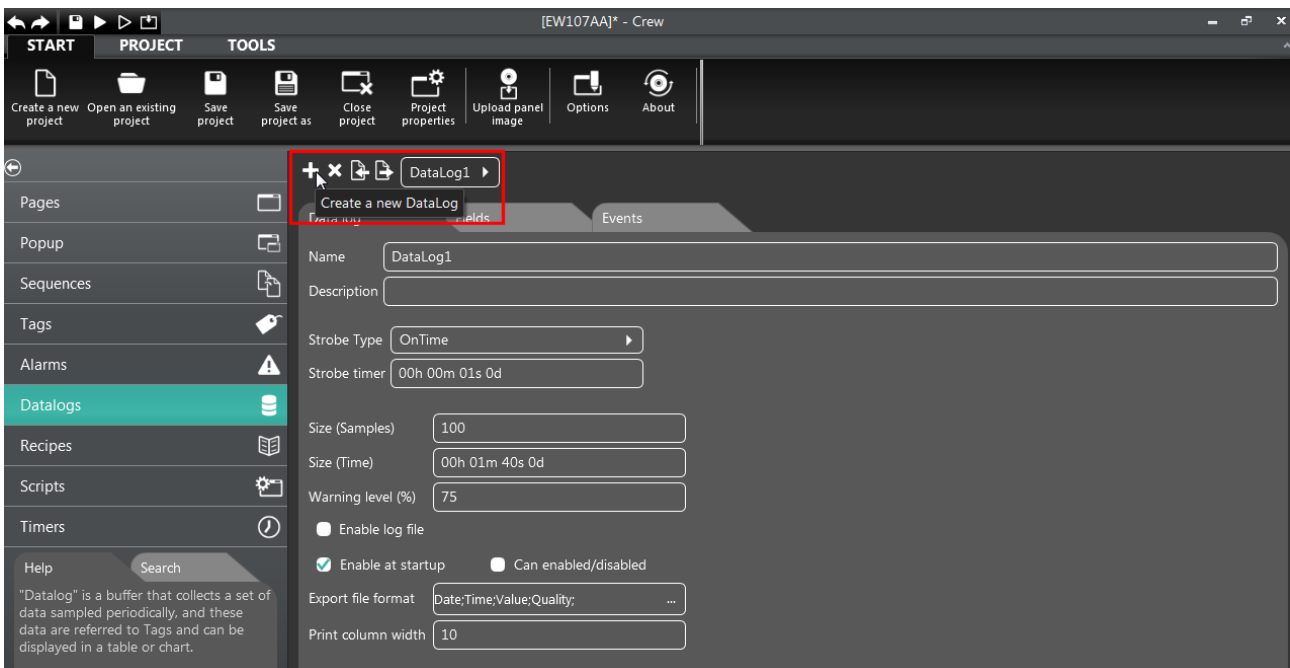
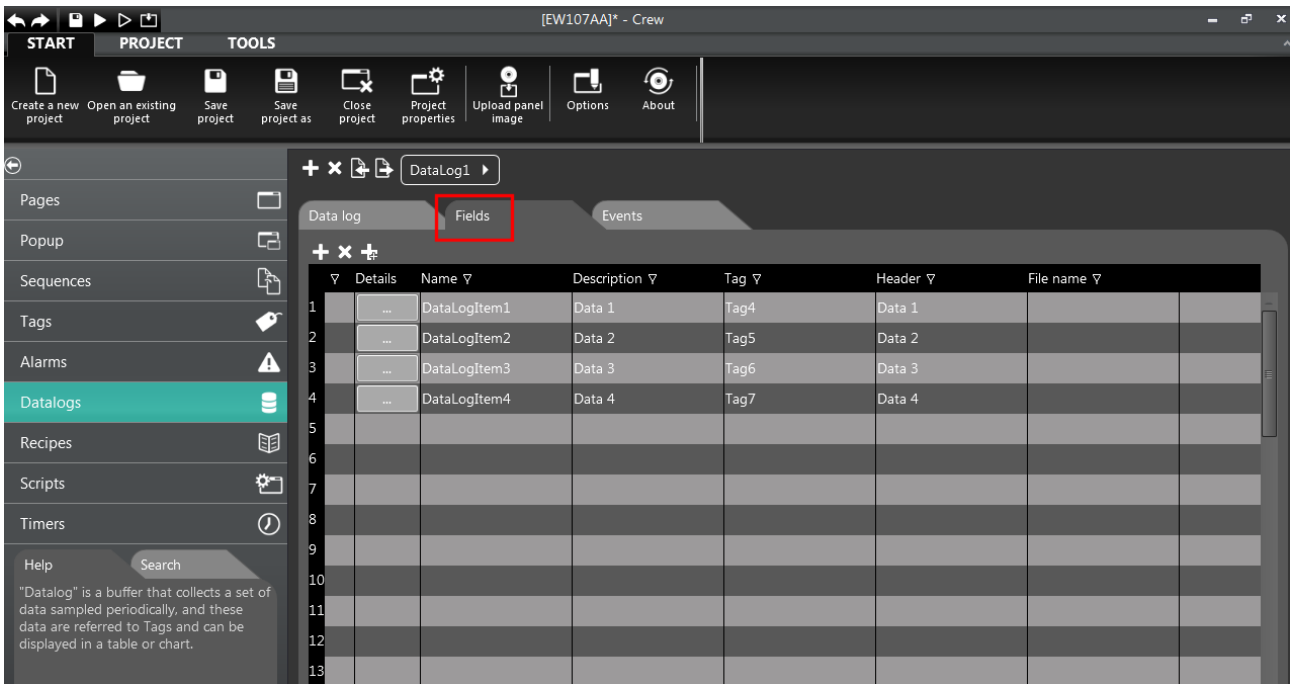
# CREW Manual

## Data Log Configuration - Crew side

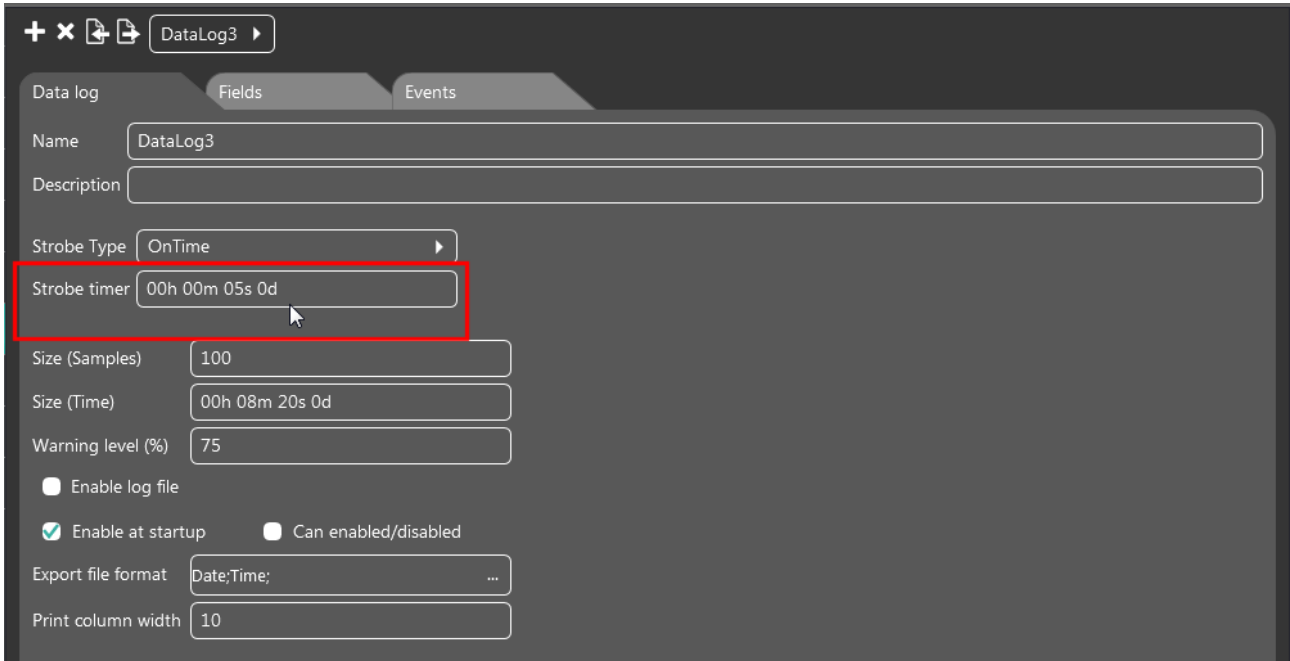
See "[Datalogs](#)" section.



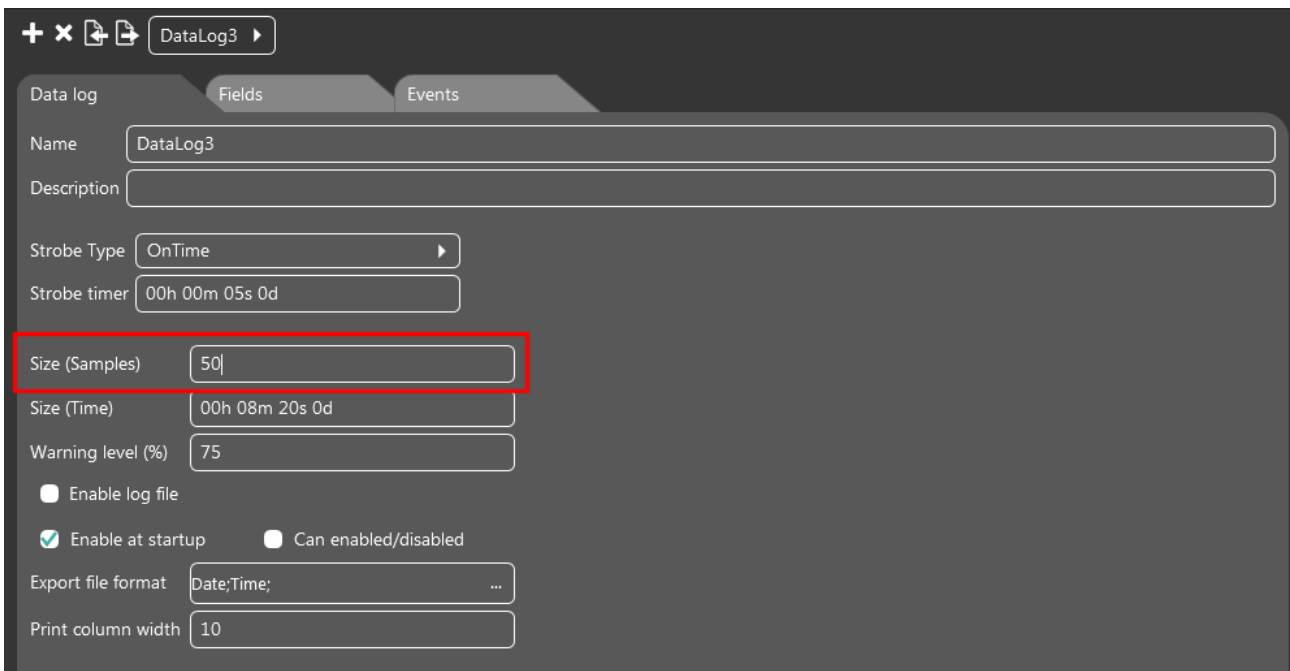
# CREW Manual



# CREW Manual

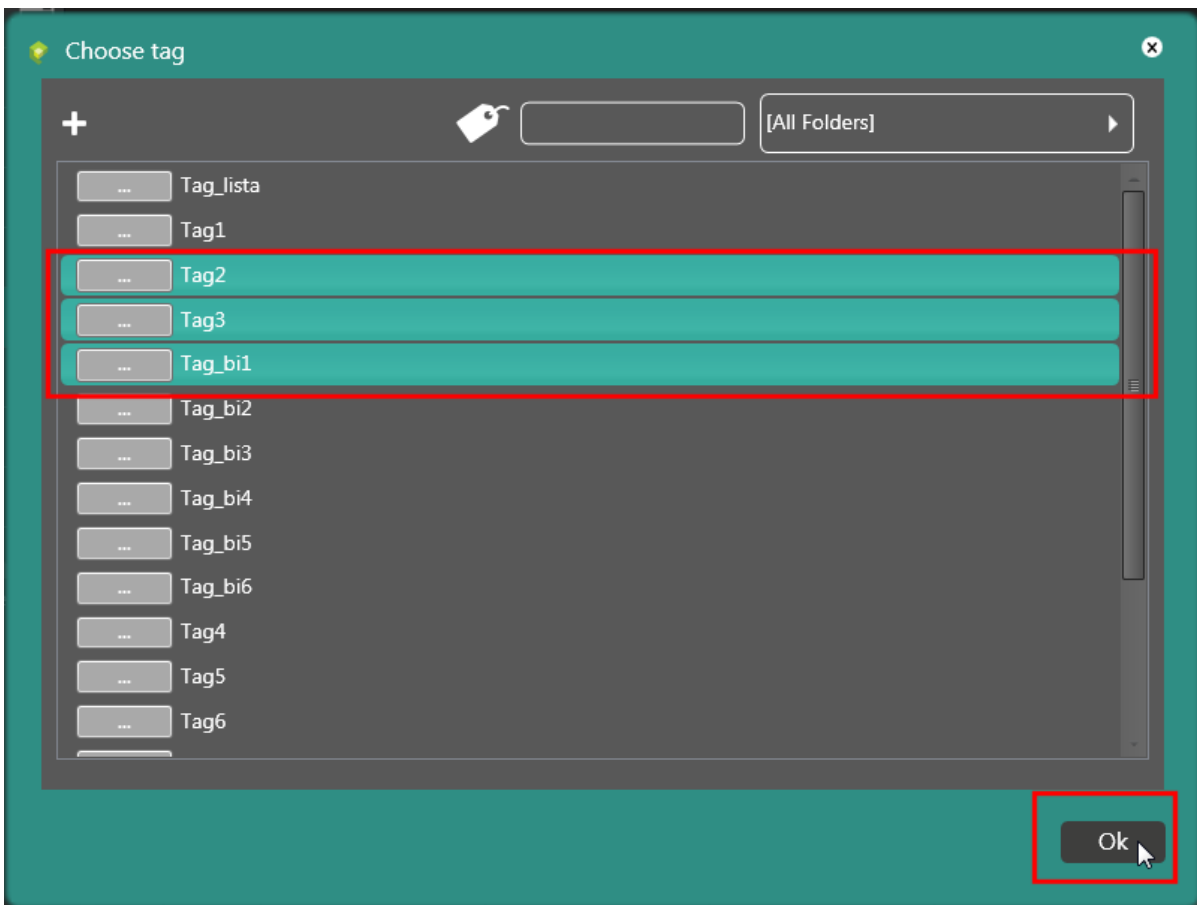


The screenshot shows the configuration window for 'DataLog3'. The 'Fields' tab is selected. The 'Strobe timer' field is highlighted with a red box and contains the value '00h 00m 05s 0d'. Other fields include Name (DataLog3), Description, Strobe Type (OnTime), Size (Samples) (100), Size (Time) (00h 08m 20s 0d), Warning level (%) (75), and checkboxes for 'Enable log file', 'Enable at startup', and 'Can enabled/disabled'. The 'Export file format' is set to 'Date;Time;' and 'Print column width' is 10.

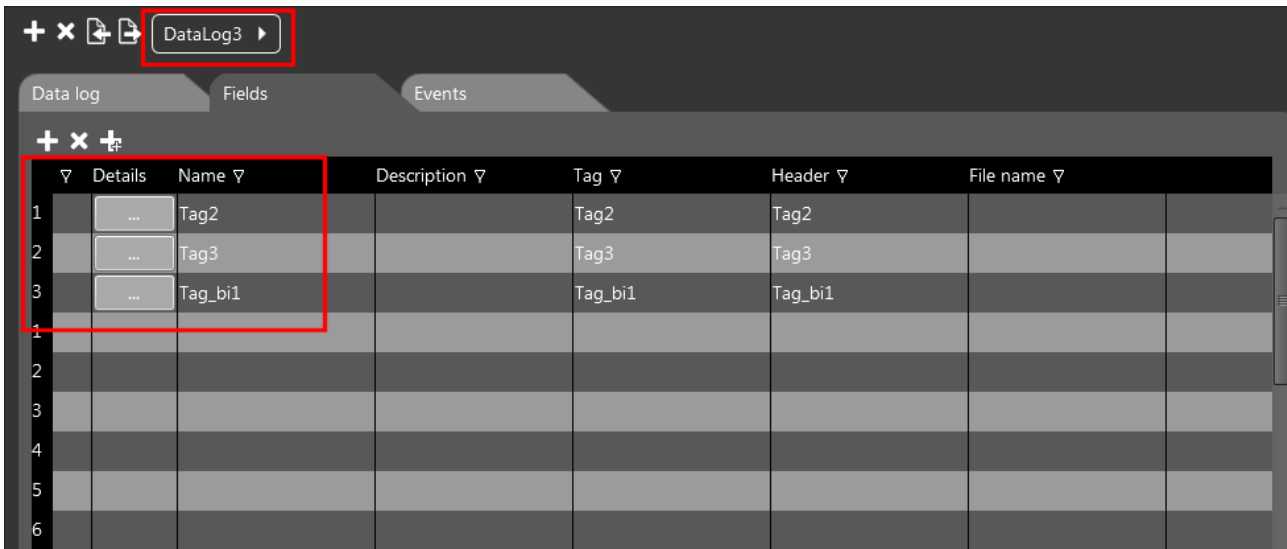


The screenshot shows the same configuration window for 'DataLog3'. The 'Size (Samples)' field is highlighted with a red box and contains the value '50'. The 'Strobe timer' field now contains '00h 00m 05s 0d'. All other fields and settings remain the same as in the previous screenshot.

# CREW Manual

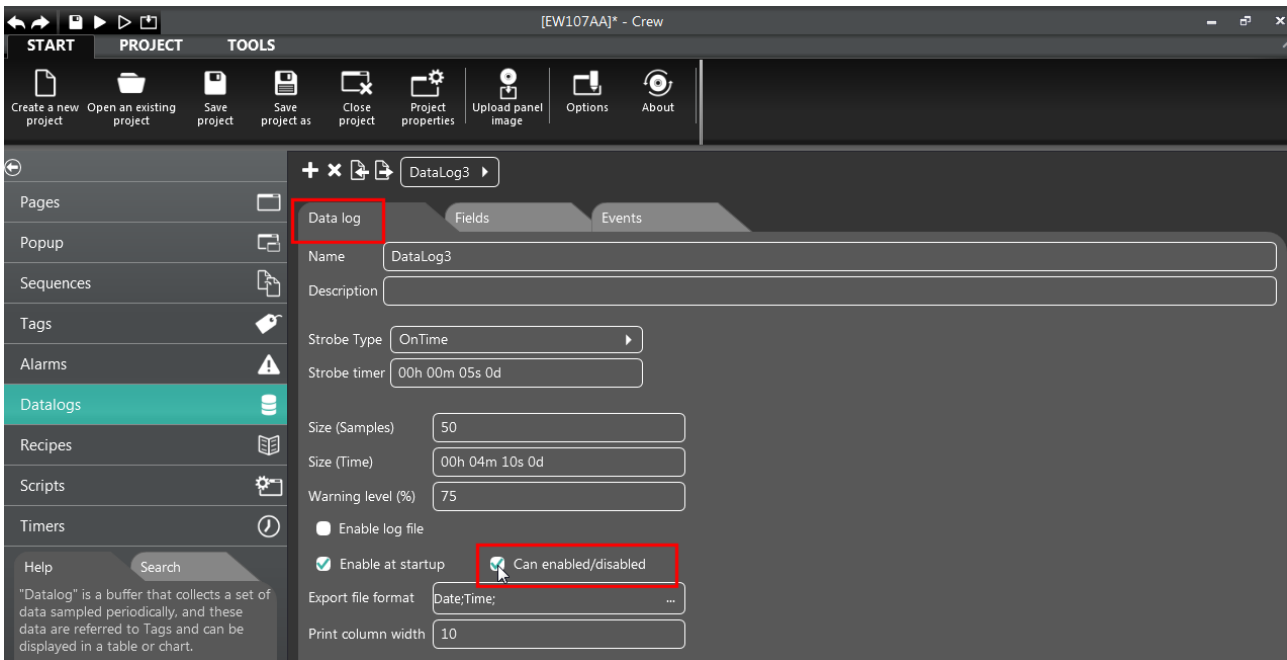


# CREW Manual



The screenshot shows a software window titled "DataLog3" with a table view. The table has columns for "Details", "Name", "Description", "Tag", "Header", and "File name". The first three rows are highlighted with a red box:

Details	Name	Description	Tag	Header	File name
...	Tag2		Tag2	Tag2	
...	Tag3		Tag3	Tag3	
...	Tag_bi1		Tag_bi1	Tag_bi1	

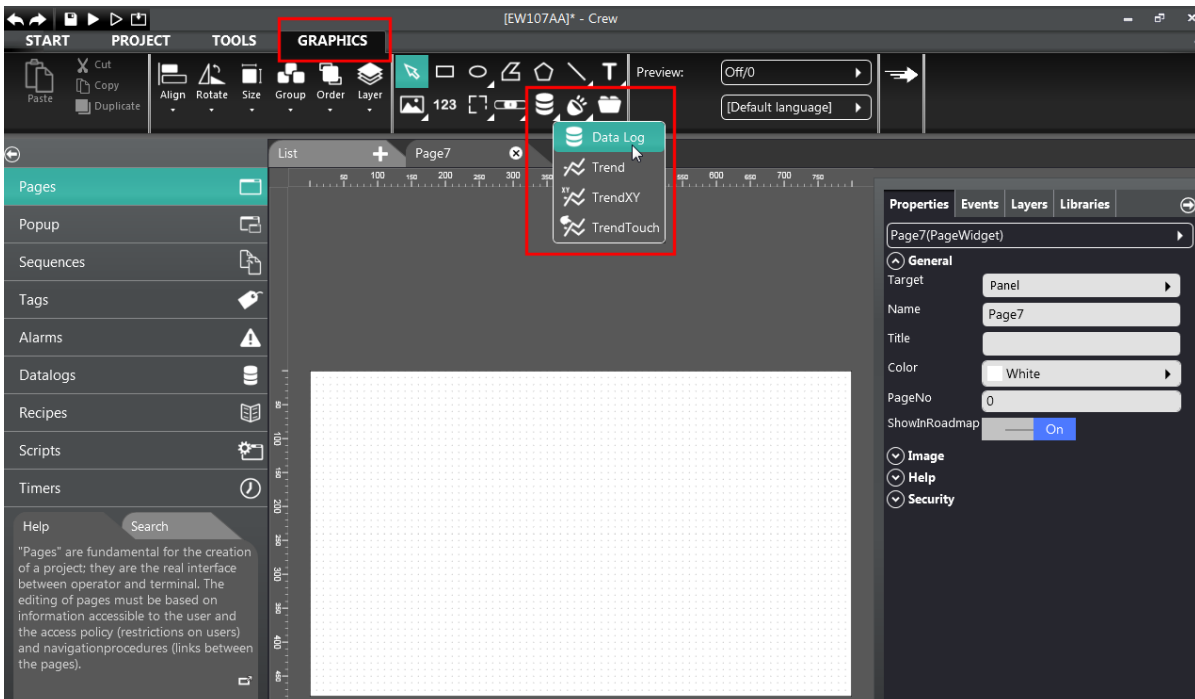
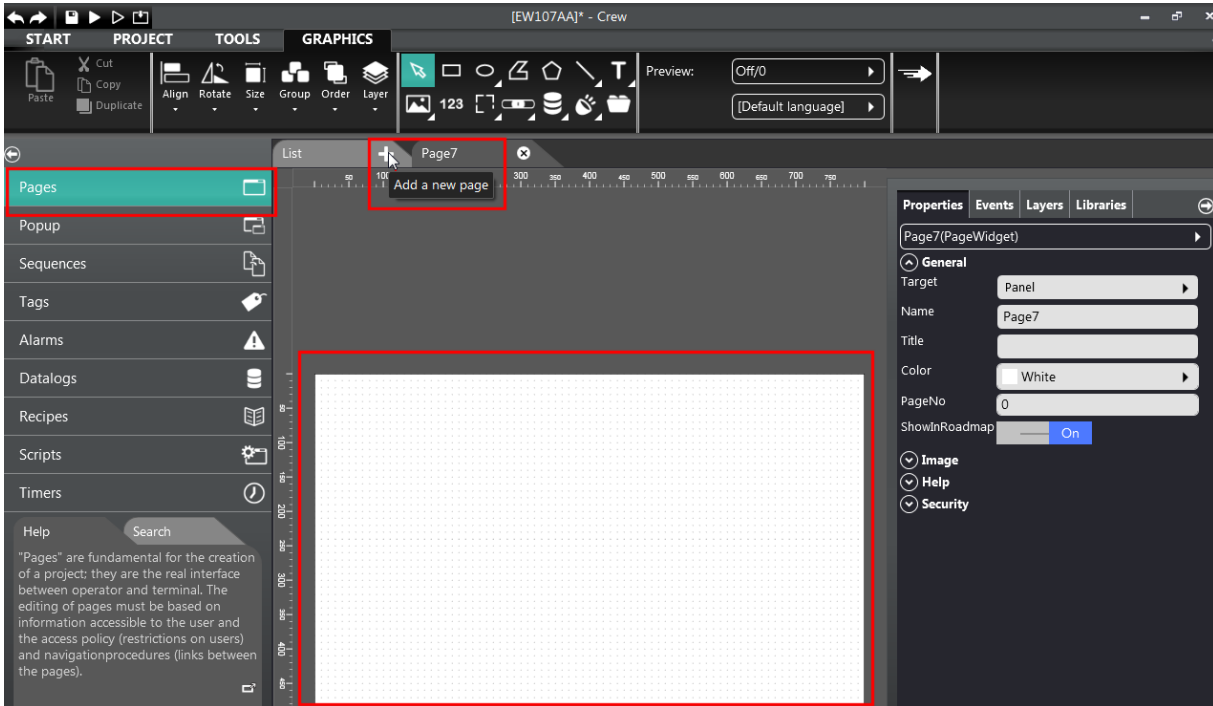


The screenshot shows the configuration window for "DataLog3". The "Data log" tab is selected and highlighted with a red box. The configuration includes the following fields:

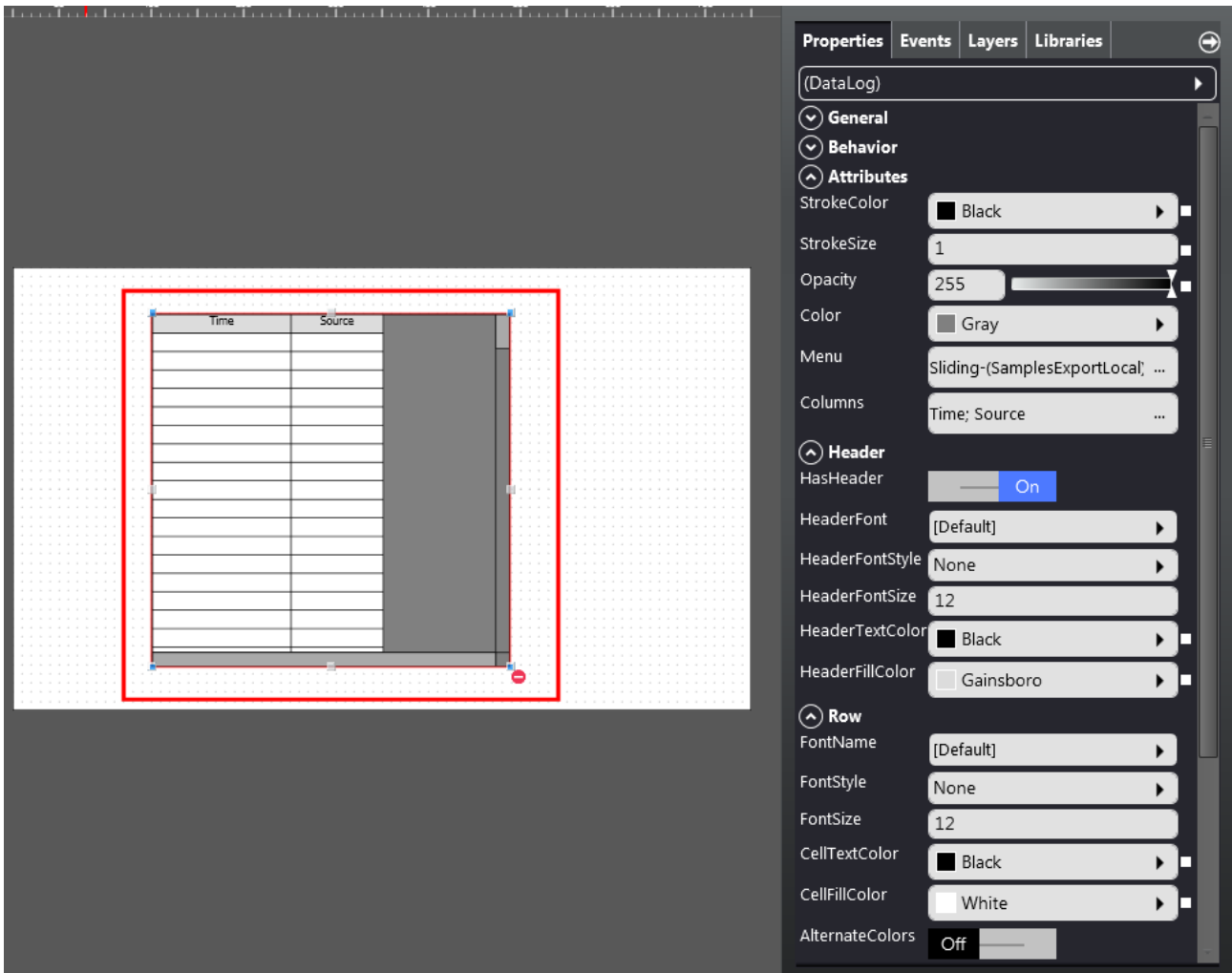
- Name: DataLog3
- Description: (empty)
- Strobe Type: OnTime
- Strobe timer: 00h 00m 05s 0d
- Size (Samples): 50
- Size (Time): 00h 04m 10s 0d
- Warning level (%): 75
- Enable log file:
- Enable at startup:  Can enabled/disabled (highlighted with a red box)
- Export file format: Date;Time; ...
- Print column width: 10

See "[Data Logs](#)" section.

# CREW Manual



# CREW Manual

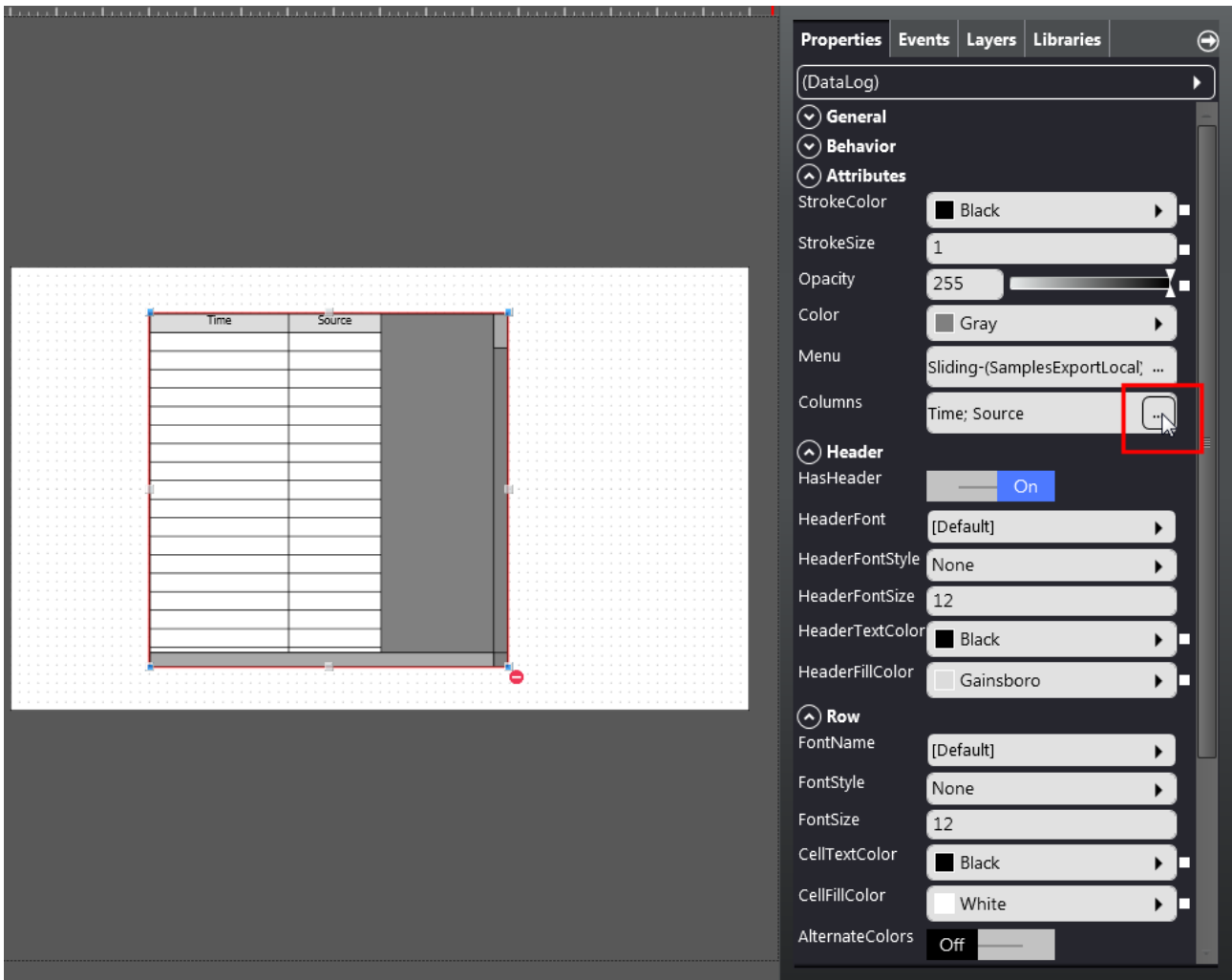


The screenshot displays the CREW software interface. On the left, a DataLog table is shown on a grid background, enclosed in a red selection box. The table has two columns: 'Time' and 'Source'. The 'Time' column contains a list of time entries, and the 'Source' column contains a list of source identifiers. On the right, the Properties panel is open, showing the configuration for the selected DataLog. The panel is titled '(DataLog)' and includes sections for General, Behavior, Attributes, Header, and Row. The Attributes section shows StrokeColor set to Black, StrokeSize set to 1, Opacity set to 255, and Color set to Gray. The Header section shows HasHeader set to On, HeaderFont set to [Default], HeaderFontStyle set to None, HeaderFontSize set to 12, HeaderTextColor set to Black, and HeaderFillColor set to Gainsboro. The Row section shows FontName set to [Default], FontStyle set to None, FontSize set to 12, CellTextColor set to Black, CellFillColor set to White, and AlternateColors set to Off.

See "[Data Log Properties](#)" section.



# CREW Manual

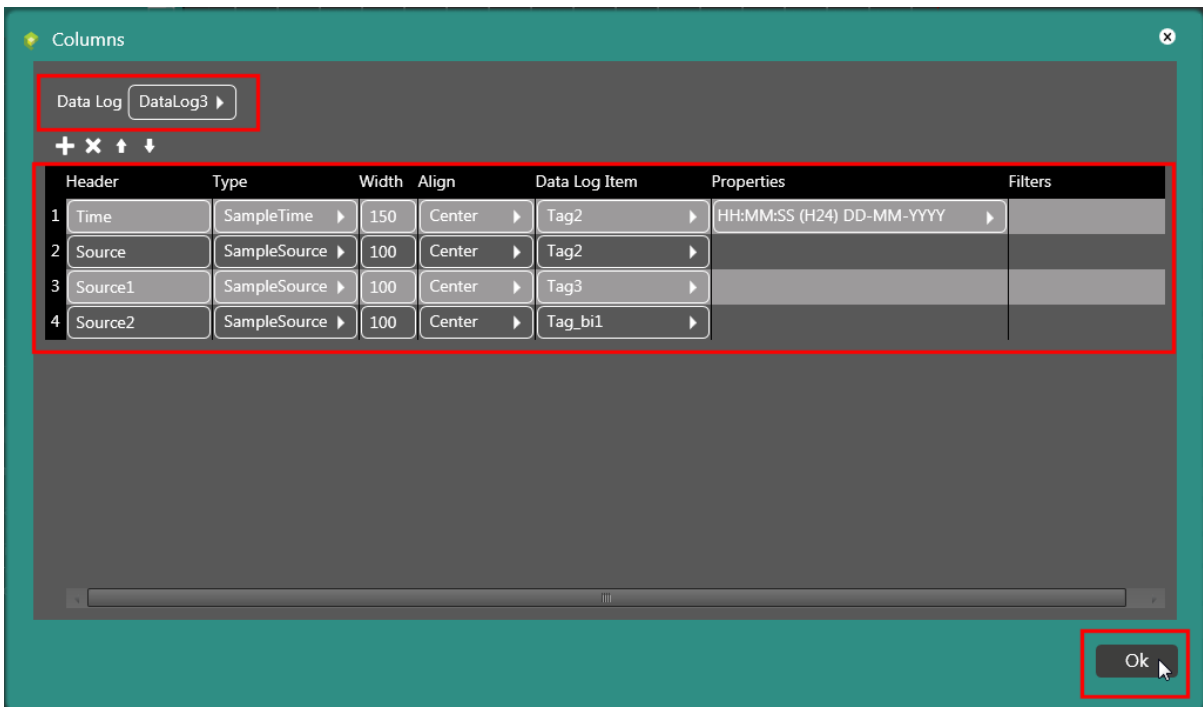
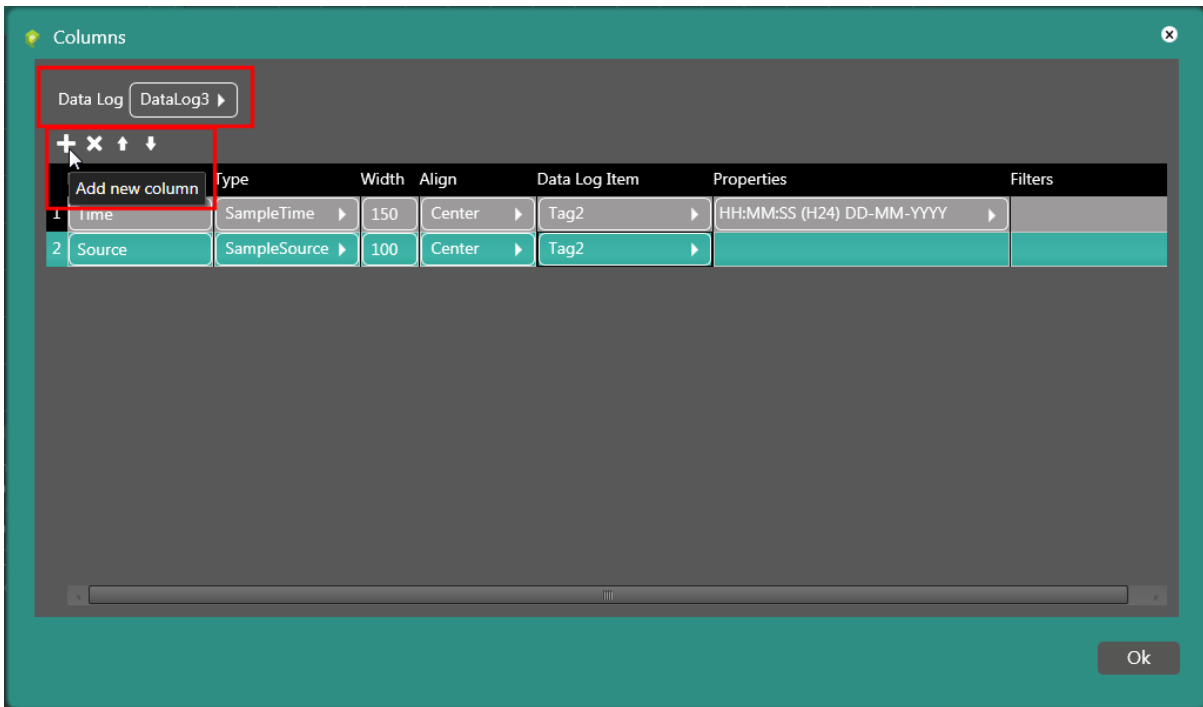


The screenshot displays the CREW software interface. On the left, a data table is shown on a grid background. The table has two columns: 'Time' and 'Source'. The 'Time' column contains a list of times, and the 'Source' column contains a list of sources. The table is currently selected, and its properties are shown in the right-hand panel.

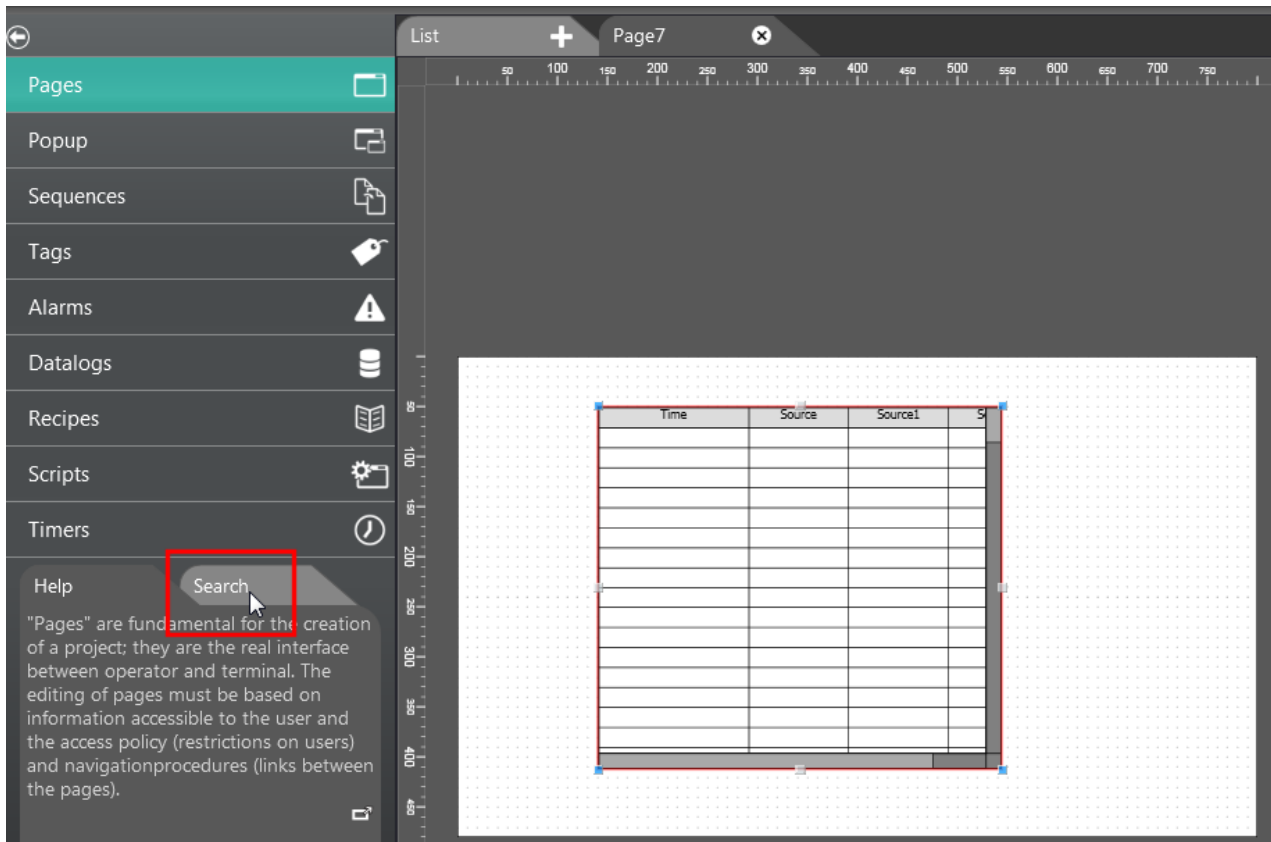
The right-hand panel is titled 'Properties' and contains the following settings:

- General**
- Behavior**
- Attributes**
  - StrokeColor: Black
  - StrokeSize: 1
  - Opacity: 255
  - Color: Gray
  - Menu: Sliding-(SamplesExportLocal', ...
  - Columns: Time; Source (highlighted with a red box)
- Header**
  - HasHeader: On
  - HeaderFont: [Default]
  - HeaderFontStyle: None
  - HeaderFontSize: 12
  - HeaderTextColor: Black
  - HeaderFillColor: Gainsboro
- Row**
  - FontName: [Default]
  - FontStyle: None
  - FontSize: 12
  - CellTextColor: Black
  - CellFillColor: White
  - AlternateColors: Off

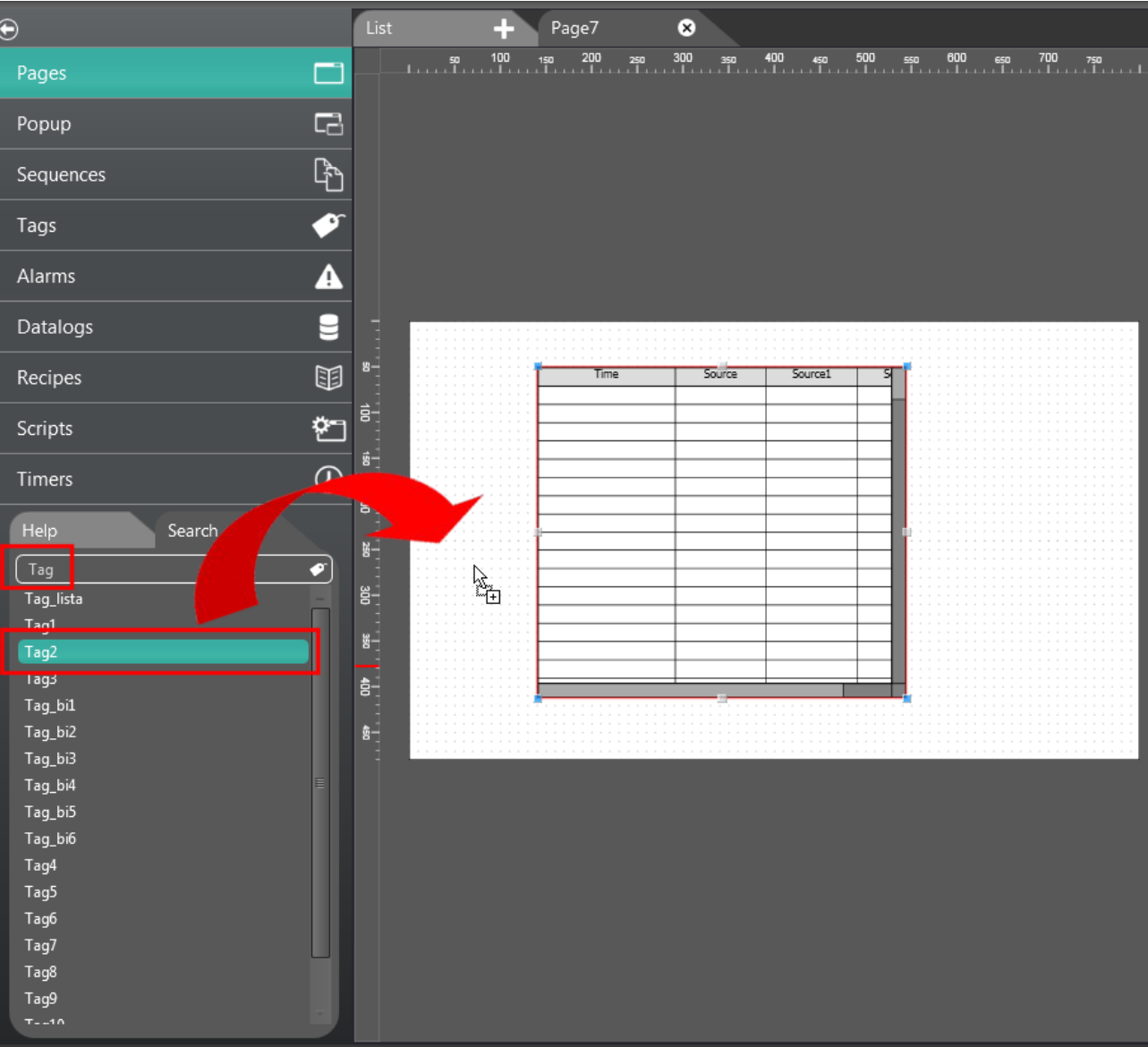
# CREW Manual



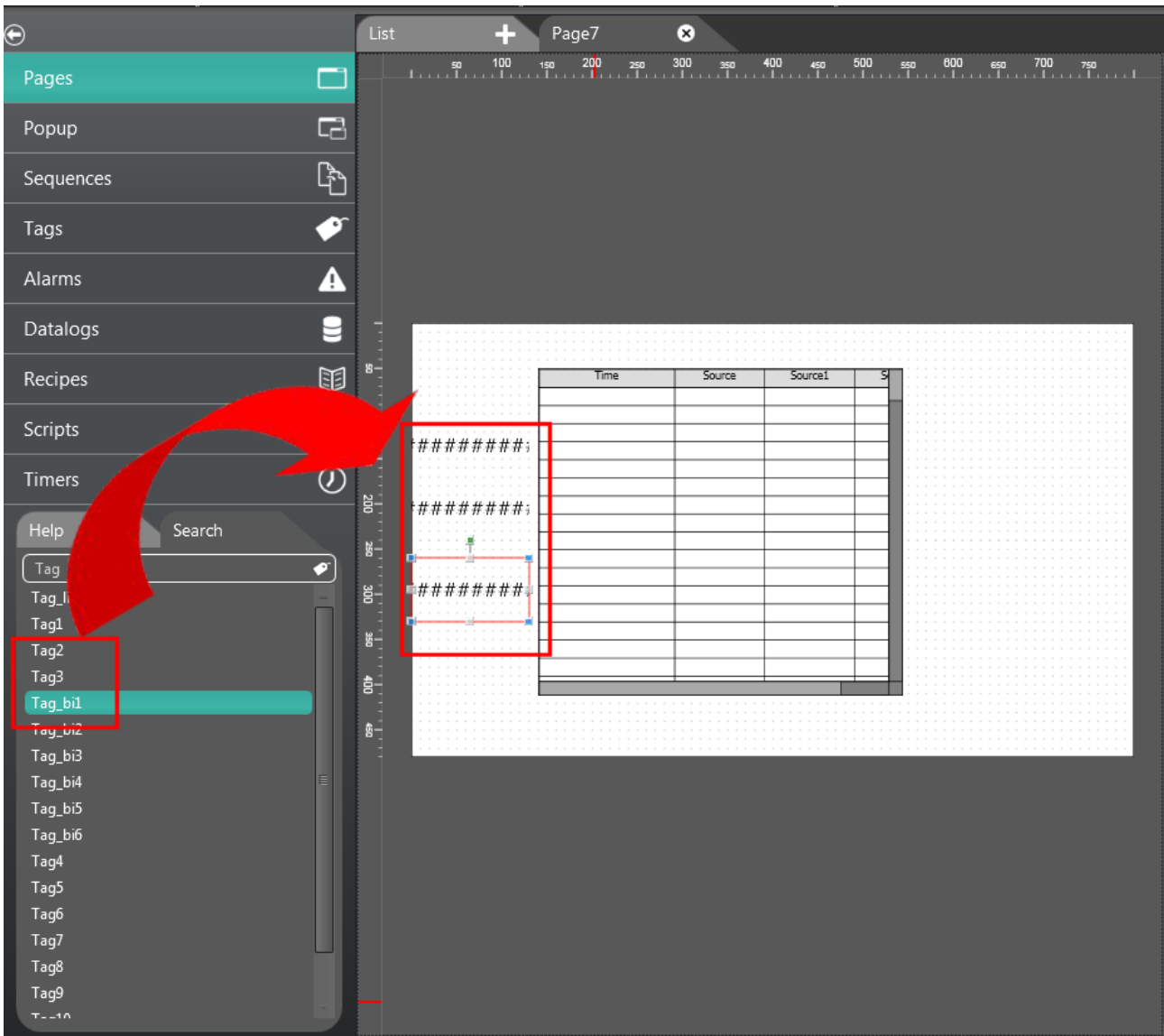
# CREW Manual



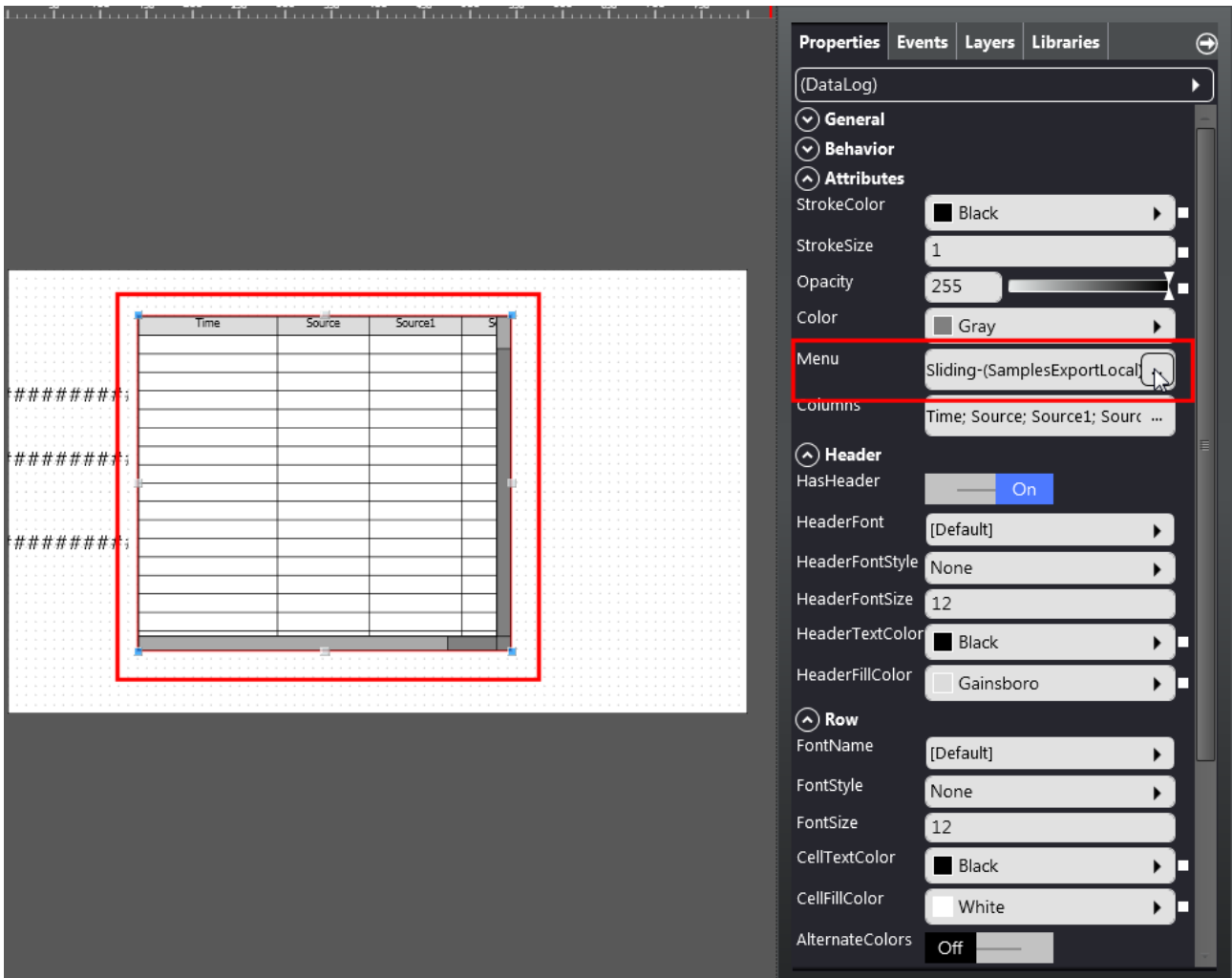
# CREW Manual



# CREW Manual



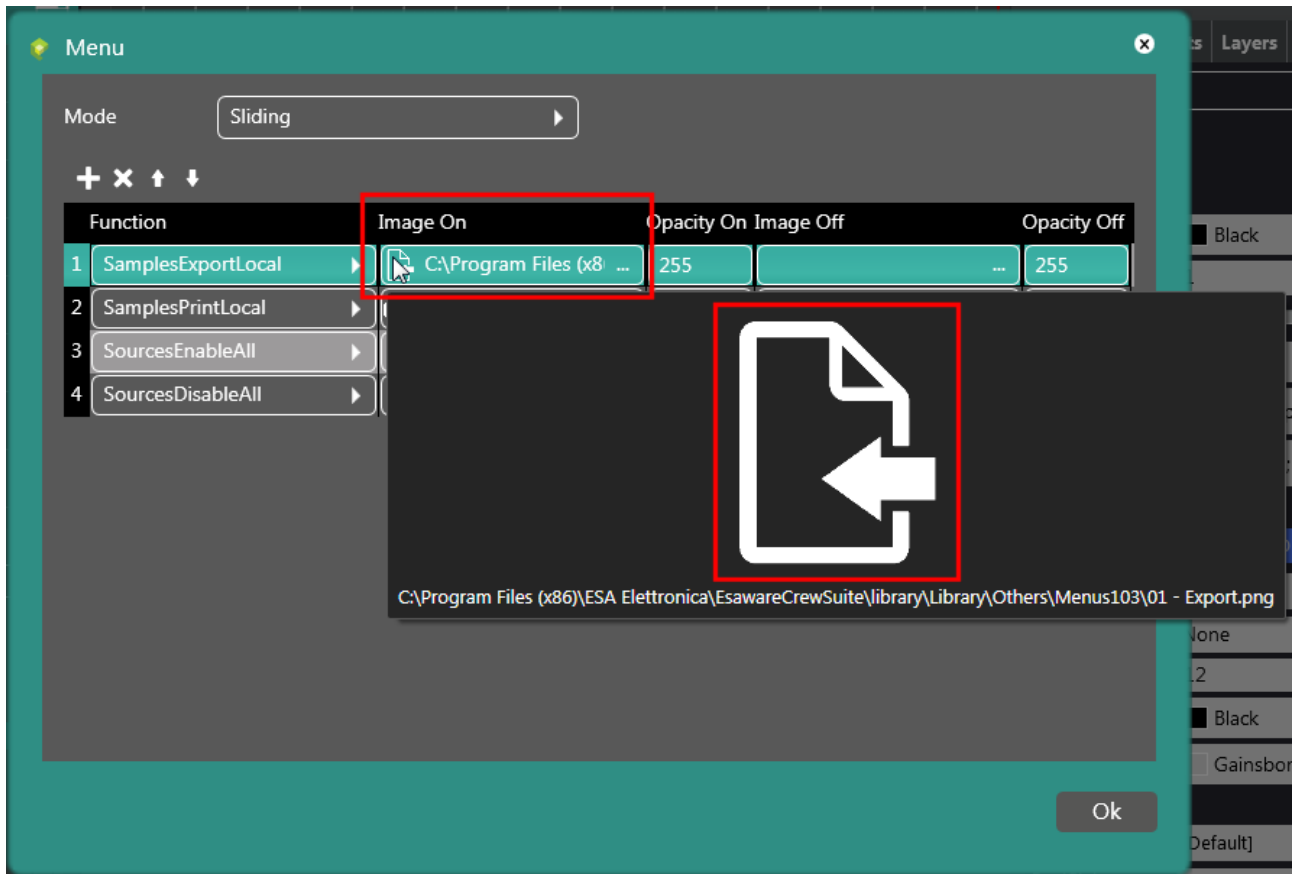
# CREW Manual



The screenshot displays the CREW software interface. On the left, a data table is visible on a grid background, with a red rectangular selection box around it. The table has four columns: "Time", "Source", "Source1", and "Source2". The first three rows are highlighted with a light gray background. To the left of the table, there are three lines of text: "#####"; "#####"; and "#####".

On the right, the "Properties" panel is open, showing various settings for the selected table. The "Menu" property is highlighted with a red box and contains the text "Sliding-(SamplesExportLocal)". Other properties include "StrokeColor" (Black), "StrokeSize" (1), "Opacity" (255), "Color" (Gray), "Columns" (Time; Source; Source1; Sourc ...), "Header" (HasHeader: On, HeaderFont: [Default], HeaderFontStyle: None, HeaderFontSize: 12, HeaderTextColor: Black, HeaderFillColor: Gainsboro), and "Row" (FontName: [Default], FontStyle: None, FontSize: 12, CellTextColor: Black, CellFillColor: White, AlternateColors: Off).

# CREW Manual



# CREW Manual

## Data Log Configuration - EW Terminal Side





# CREW Manual

**DataLog View**

Time	Data1	Data2	Data3	Data4
10:16:41 04-02-2015	0	0	0	0
10:16:42 04-02-2015	0	0	0	0
10:16:43 04-02-2015	29	30	77	1
10:16:44 04-02-2015	5	41	86	79
10:16:45 04-02-2015	5	41	86	79
10:16:46 04-02-2015	6	95	36	52
10:16:47 04-02-2015	47	30	62	65
10:16:48 04-02-2015	82	59	99	91
10:16:49 04-02-2015	82	59	99	91
10:16:50 04-02-2015	24	53	11	100
10:16:51 04-02-2015	10	10	80	28
10:16:52 04-02-2015	30	95	98	40
10:16:53 04-02-2015	30	95	98	40



**Data 1**  
65

**Data 2**  
41

**Data 3**  
41

**Data 4**  
71

**DataLog View**

Time	Data1	Data2	Data3	Data4
10:16:41 04-02-2015	0	0	0	0
10:16:42 04-02-2015	0	0	0	0
10:16:43 04-02-2015	29	30	77	1
10:16:44 04-02-2015	5	41	86	79
10:16:45 04-02-2015	5	41	86	79
10:16:46 04-02-2015	6	95	36	52
10:16:47 04-02-2015	47	30	62	65
10:16:48 04-02-2015	82	59	99	91
10:16:49 04-02-2015	82	59	99	91
10:16:50 04-02-2015	24	53	11	100
10:16:51 04-02-2015	10	10	80	28
10:16:52 04-02-2015	30	95	98	40
10:16:53 04-02-2015	30	95	98	40
10:16:54 04-02-2015	65	41	41	71
10:16:55 04-02-2015	19	58	8	46
10:16:56 04-02-2015	38	29	92	63



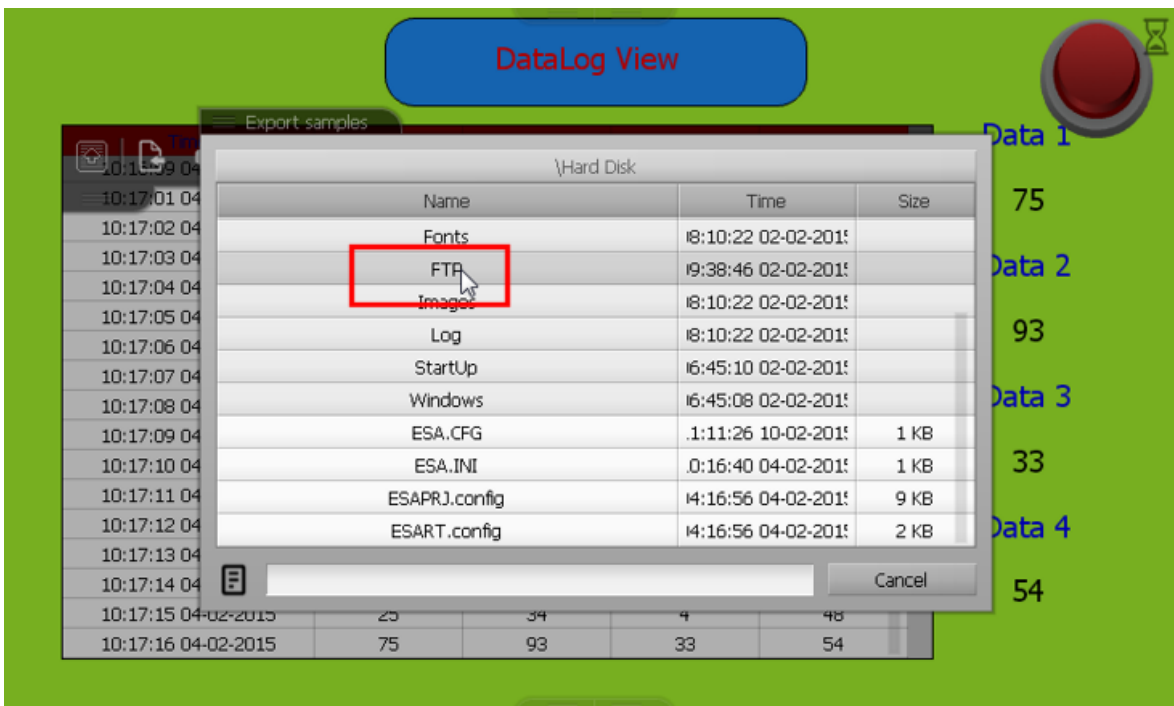
**Data 1**  
38

**Data 2**  
29

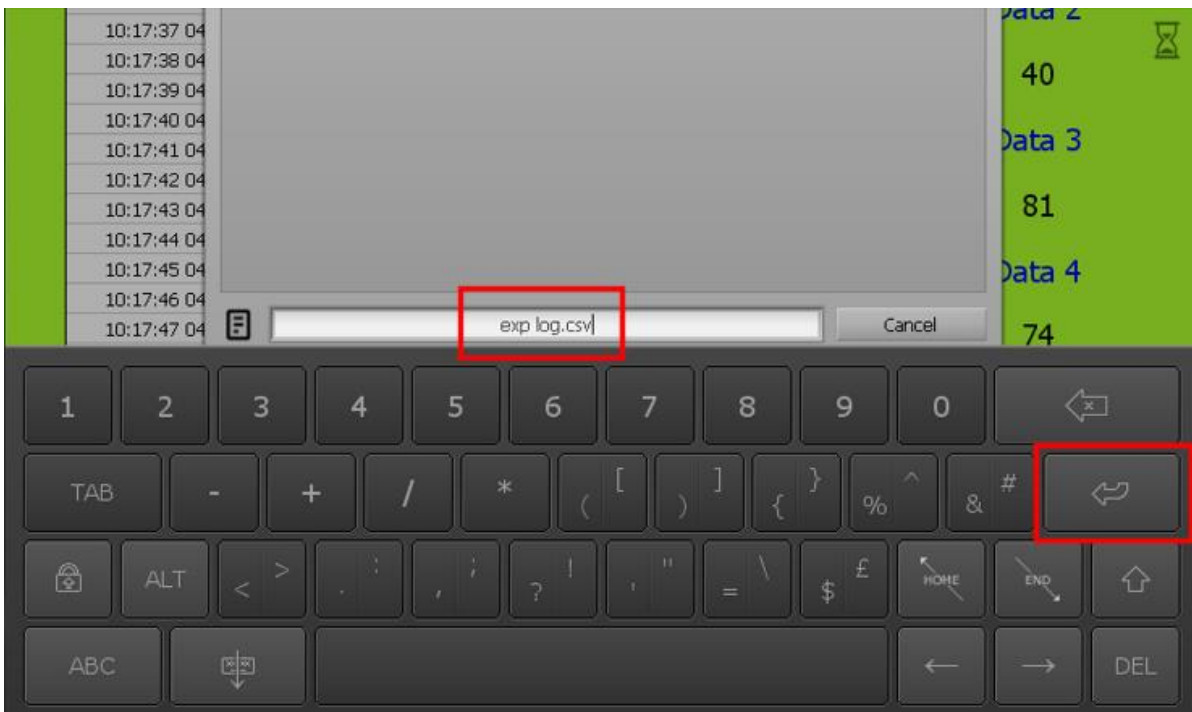
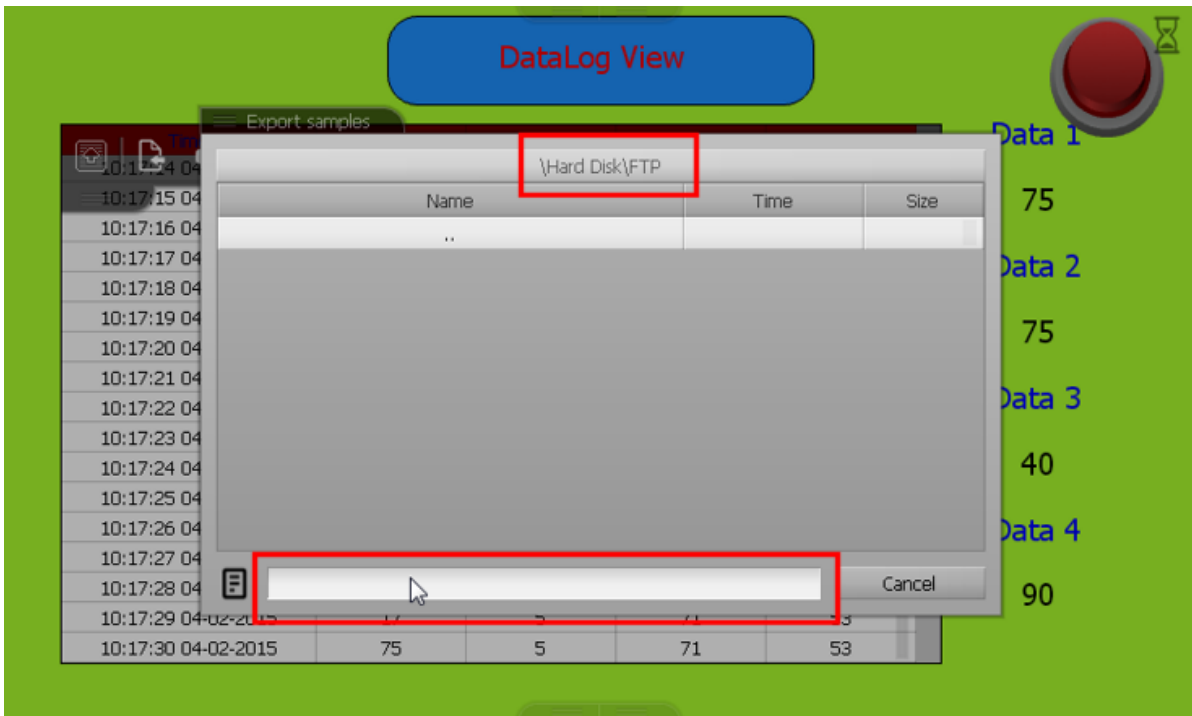
**Data 3**  
92

**Data 4**  
63

# CREW Manual



# CREW Manual



# CREW Manual

DataLog View

📄 Samples <DataLog1> export completed

	Data1	Data2	Data3	Data4
10:17:36 04-02-2015	96	54	49	97
10:17:37 04-02-2015	28	50	14	52
10:17:38 04-02-2015	66	44	69	6
10:17:39 04-02-2015	16	22	33	78
10:17:40 04-02-2015	16	22	33	78
10:17:41 04-02-2015	80	33	97	80
10:17:42 04-02-2015	42	12	95	80
10:17:43 04-02-2015	17	16	51	41
10:17:44 04-02-2015	85	50	19	90
10:17:45 04-02-2015	22	45	24	88
10:17:46 04-02-2015	22	45	24	88
10:17:47 04-02-2015	86	59	93	52
10:17:48 04-02-2015	26	18	35	0
10:17:50 04-02-2015	70	40	81	74
10:17:51 04-02-2015	34	71	31	80
10:17:52 04-02-2015	34	71	31	80
10:17:54 04-02-2015	57	25	86	7

Data 1

57

Data 2

25

Data 3

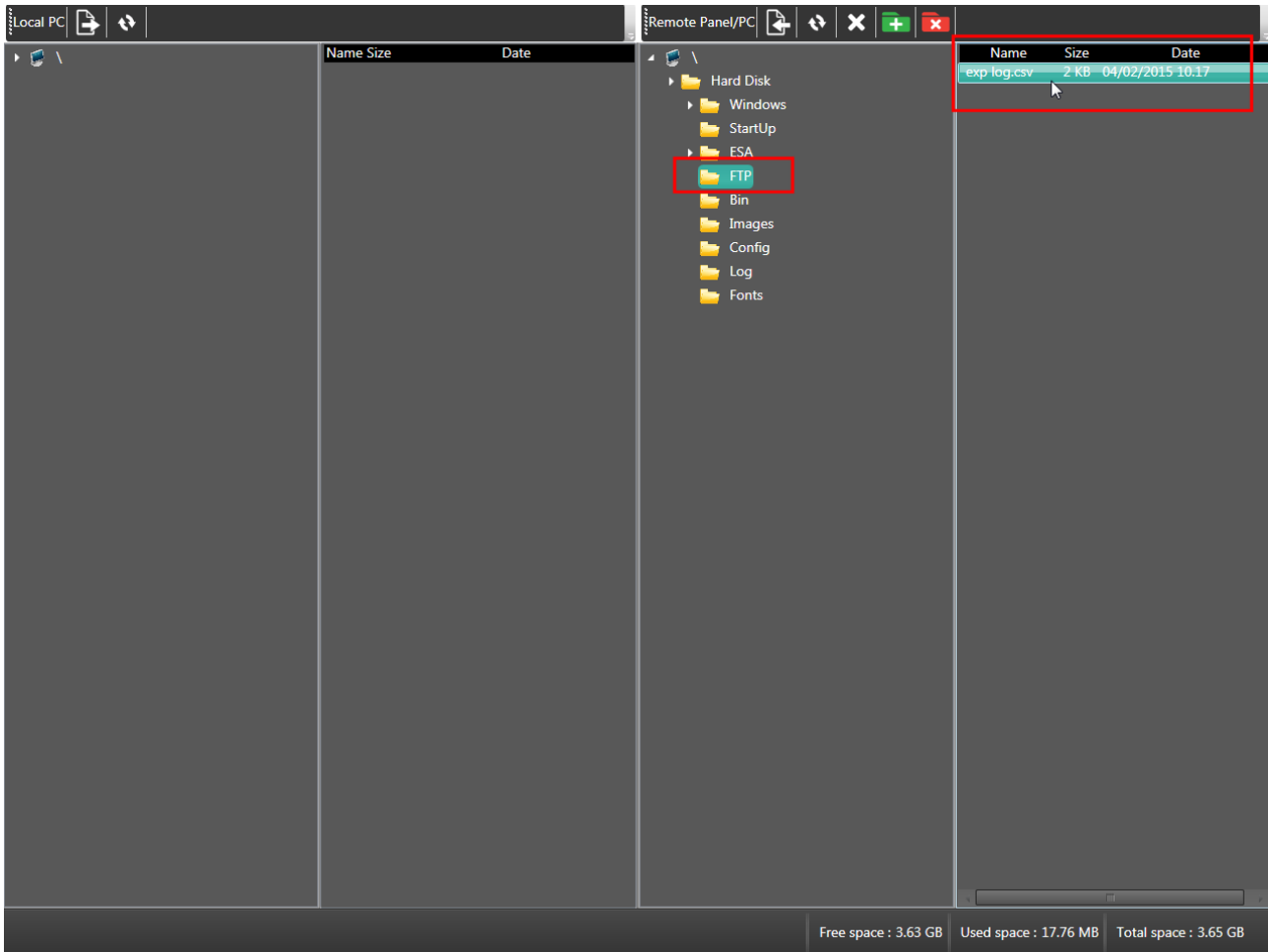
86

Data 4

7

# CREW Manual

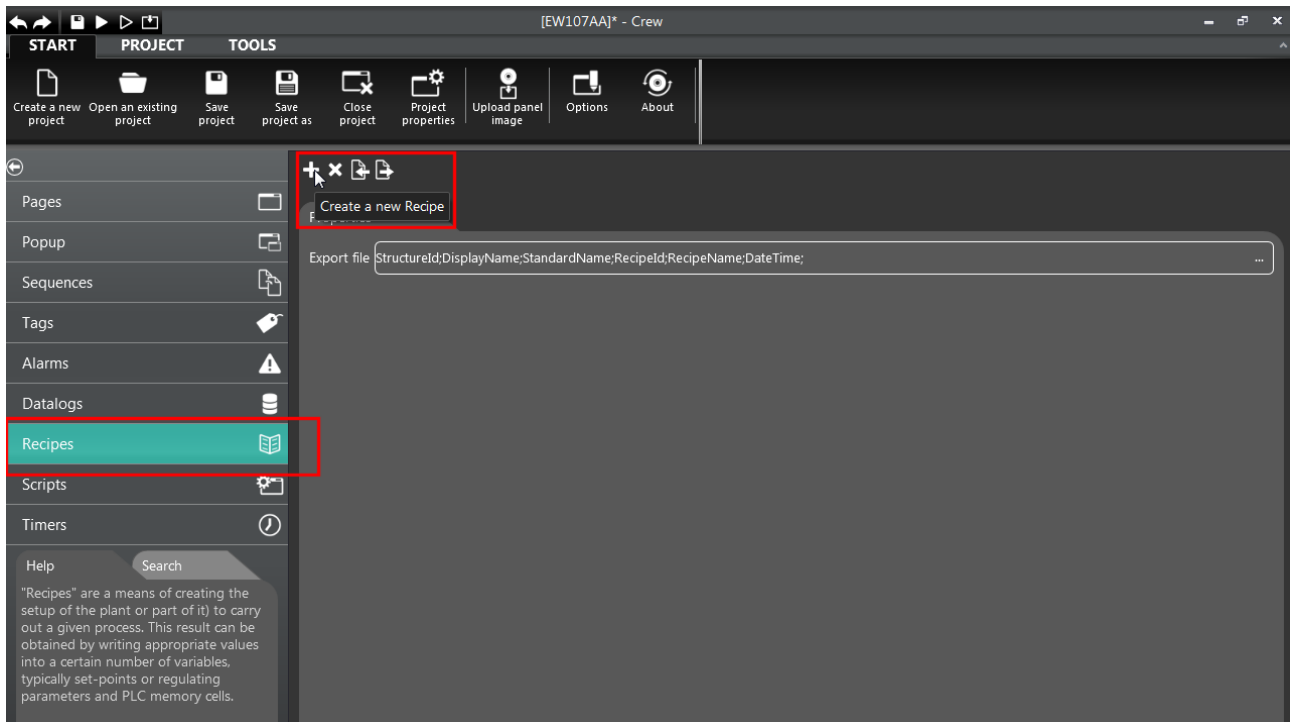
Below are the instructions provided in the "[Online Tools](#)" section to explore the hard disk of the EW terminal so as to verify the presence of the newly-created "exp log.csv" file containing the export of the sample taking of the data created in the example.



# CREW Manual

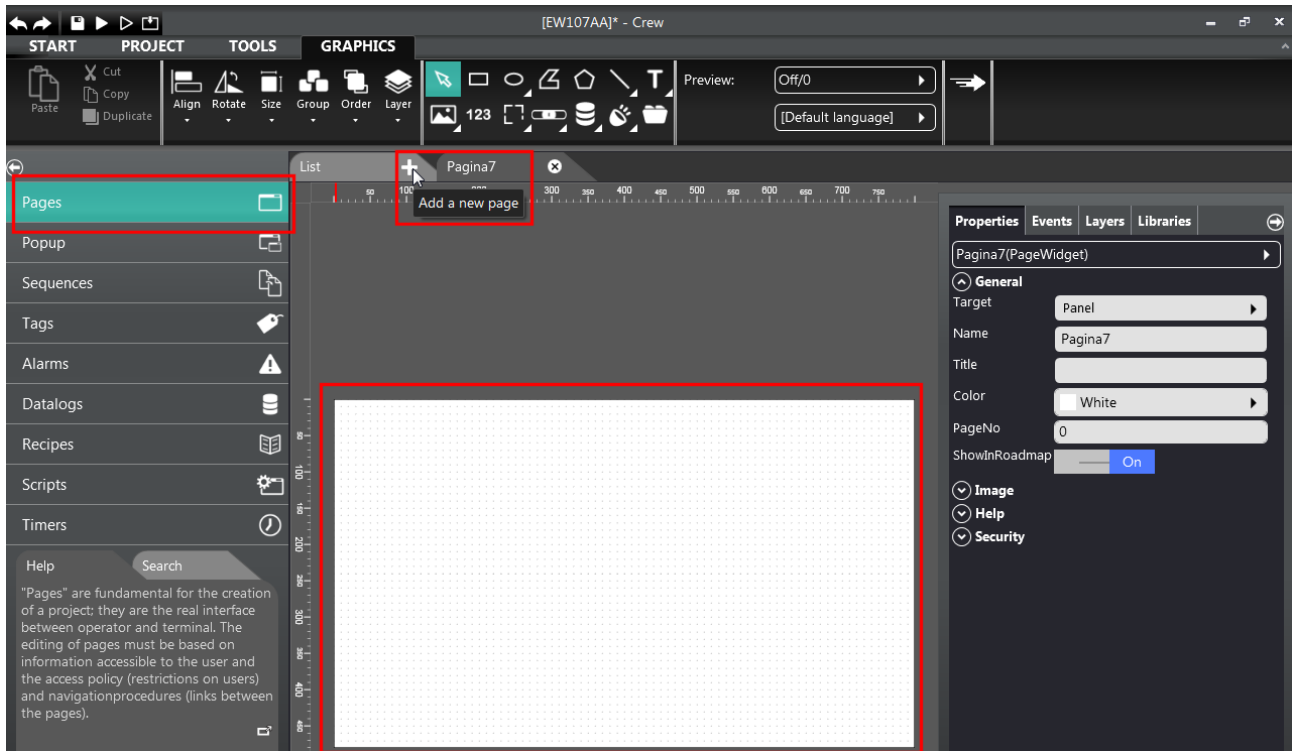
## Recipe Configuration - Crew Side

Creating a new recipe, as shown in the image.



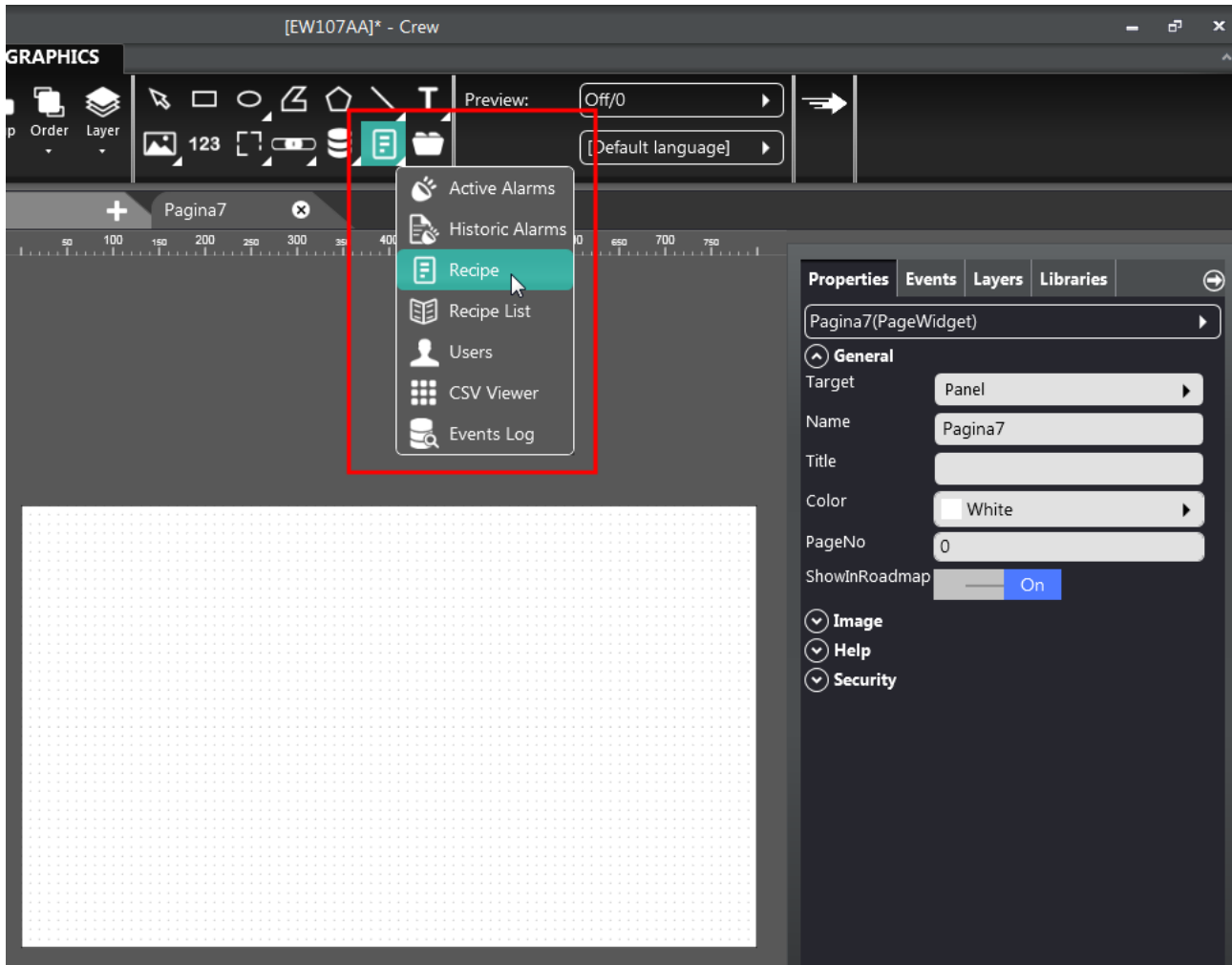
# CREW Manual

Adding a new page to the project.



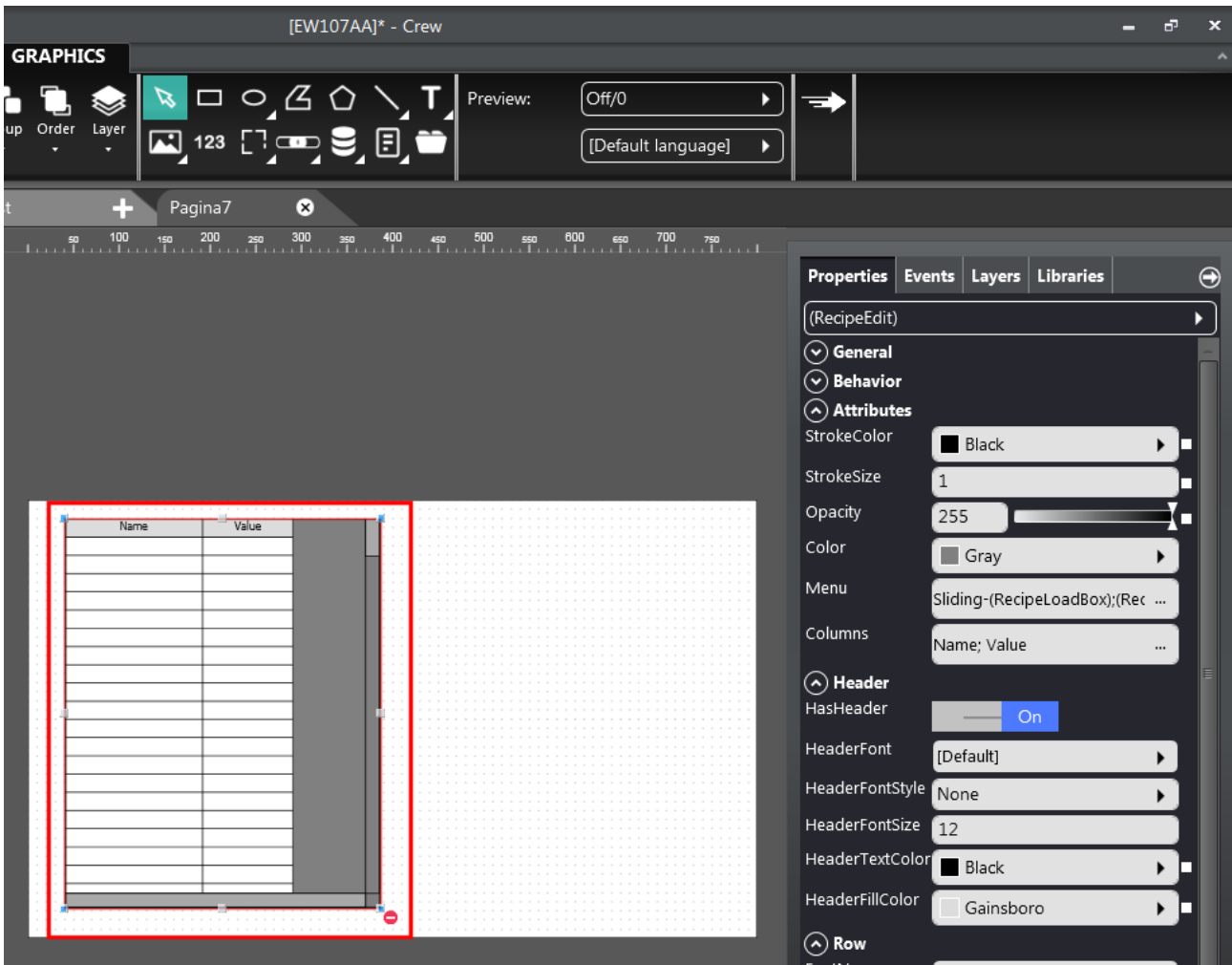
# CREW Manual

Adding a "View Recipe" to the page (for more information see the "[Recipe](#)" section).



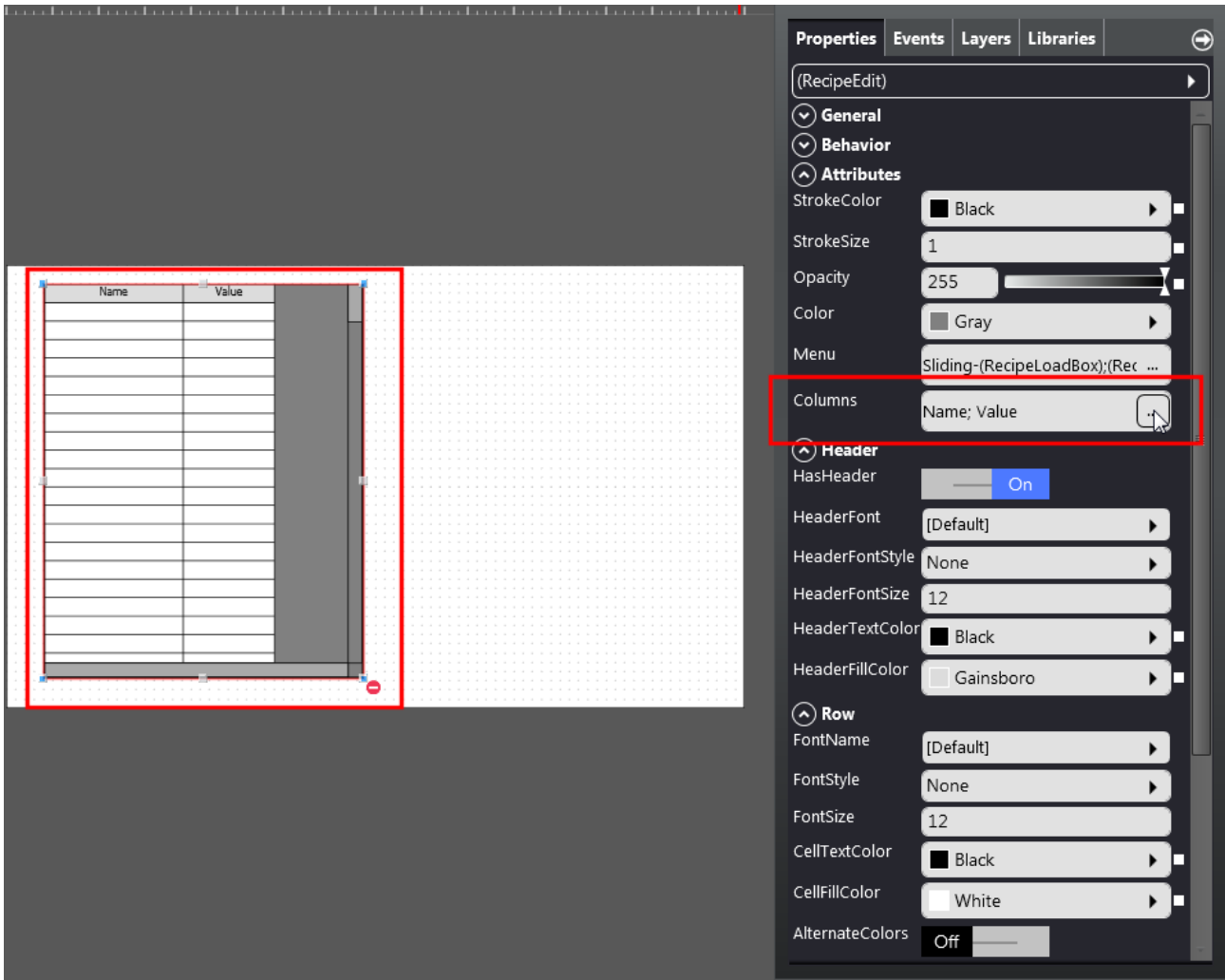


# CREW Manual

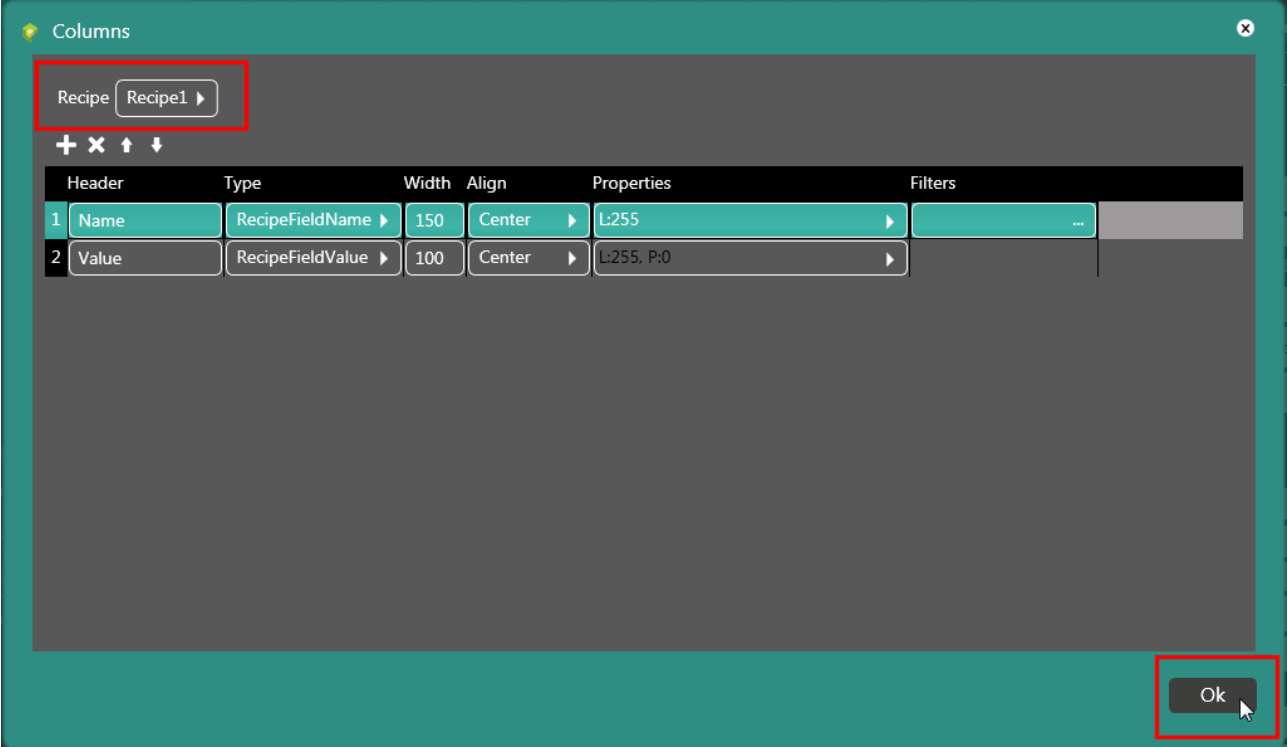


# CREW Manual

Defining the structure of the "Menu" and the "Columns" (for more information see the "[Recipe Properties](#)" section).



# CREW Manual



Columns

Recipe Recipe1 ▾

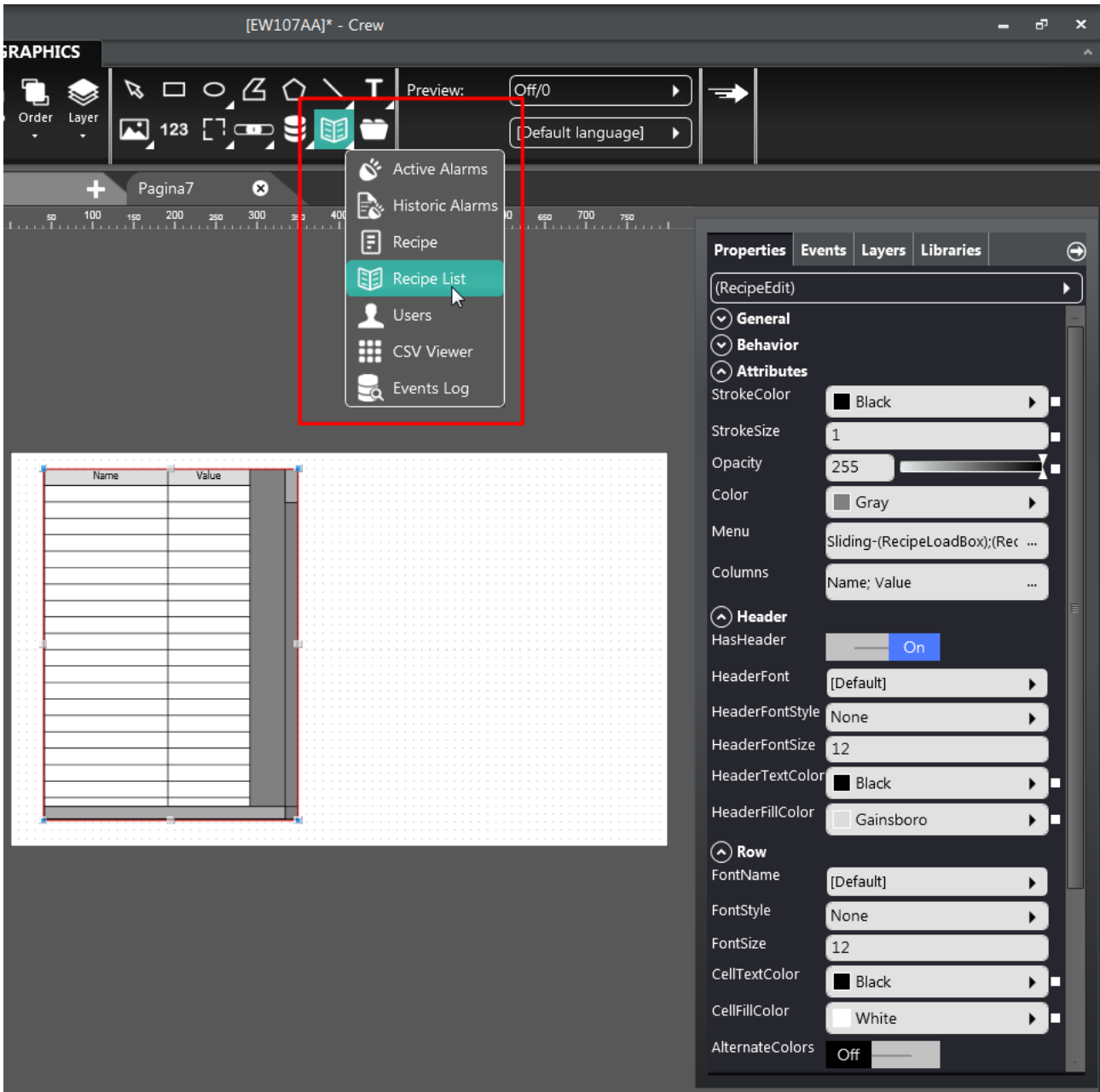
+ × ↑ ↓

	Header	Type	Width	Align	Properties	Filters
1	Name	RecipeFieldName ▾	150	Center ▾	L:255 ▾	...
2	Value	RecipeFieldValue ▾	100	Center ▾	L:255, P:0 ▾	

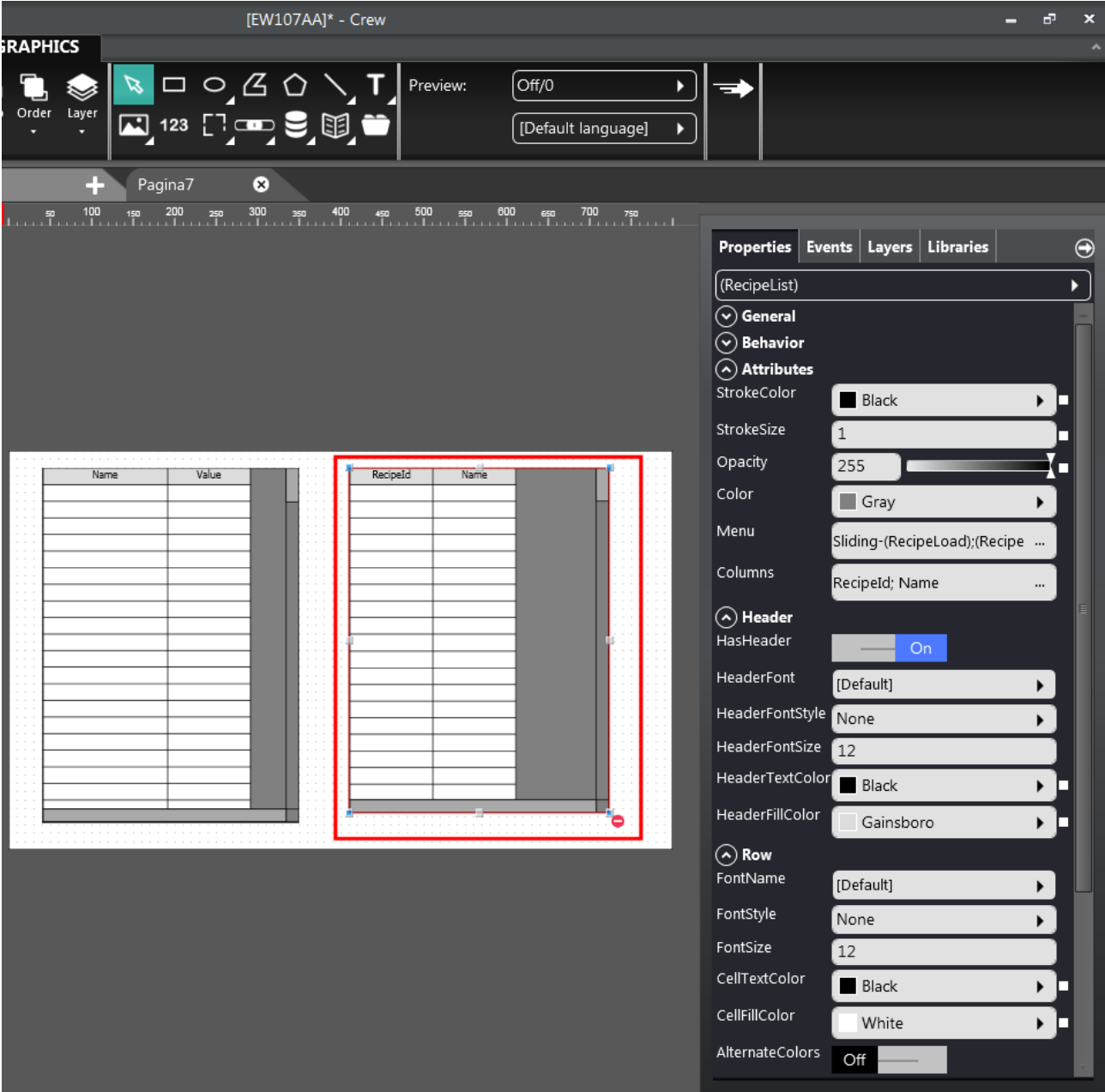
Ok

# CREW Manual

Adding a "Recipe List" view (see "[Recipe List](#)" section).



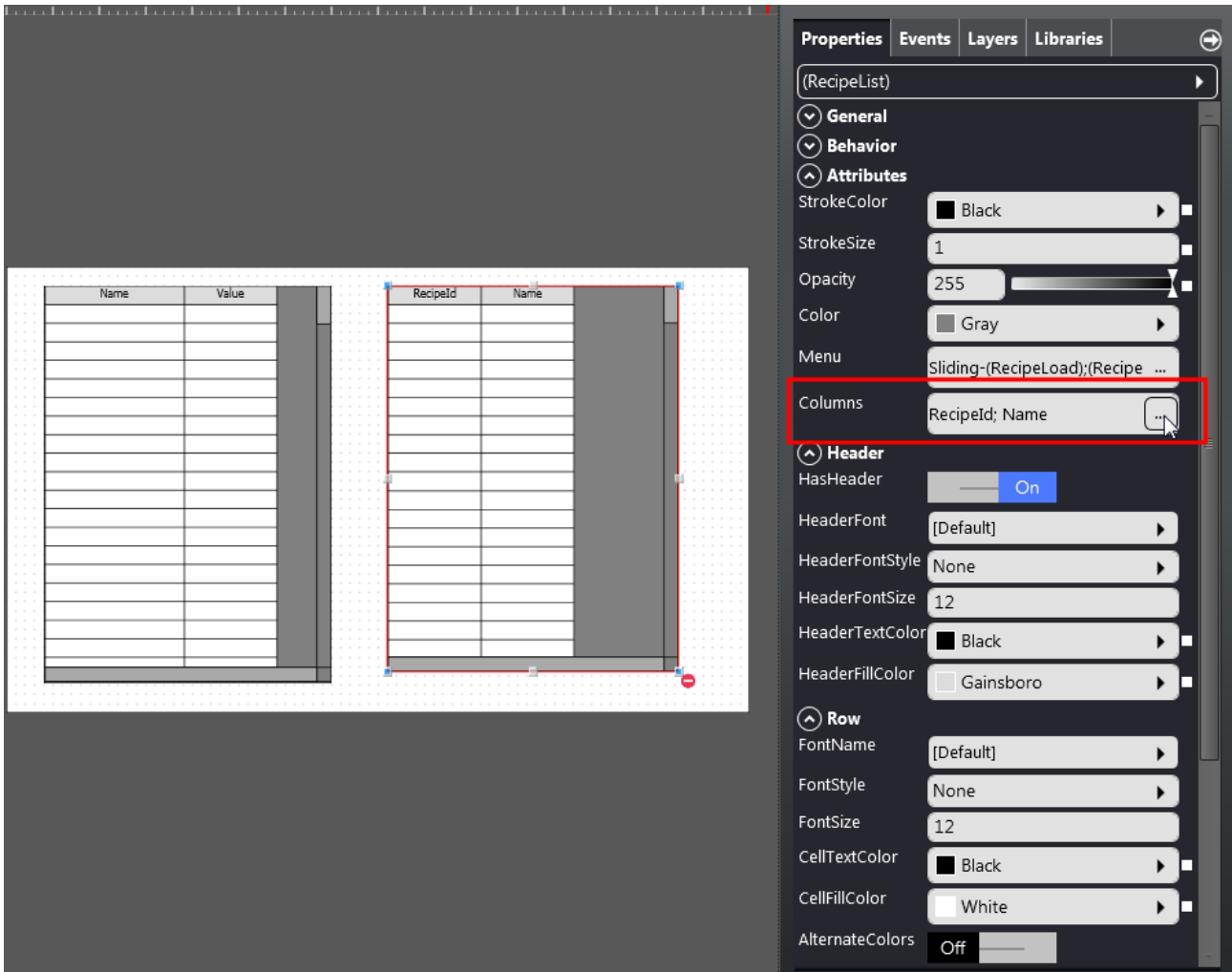
# CREW Manual



The screenshot displays the ESNA AUTOMATION software interface. The top menu bar includes 'GRAPHICS' and a toolbar with various drawing tools. A 'Preview' dropdown is set to 'Off/0' and a language dropdown is set to '[Default language]'. The main workspace shows a table with two columns: 'Name' and 'Value'. A second table is overlaid on the first, with columns 'RecipeId' and 'Name', and is highlighted with a red border. The right-hand side features a 'Properties' panel with tabs for 'Properties', 'Events', 'Layers', and 'Libraries'. The selected object is '(RecipeList)'. The 'Attributes' section includes 'StrokeColor' (Black), 'StrokeSize' (1), 'Opacity' (255), 'Color' (Gray), 'Menu' (Sliding-(RecipeLoad);(Recipe ...), and 'Columns' (RecipeId; Name). The 'Header' section has 'HasHeader' set to 'On', 'HeaderFont' ([Default]), 'HeaderFontSize' (12), 'HeaderTextColor' (Black), and 'HeaderFillColor' (Gainsboro). The 'Row' section includes 'FontName' ([Default]), 'FontSize' (12), 'CellTextColor' (Black), 'CellFillColor' (White), and 'AlternateColors' (Off).

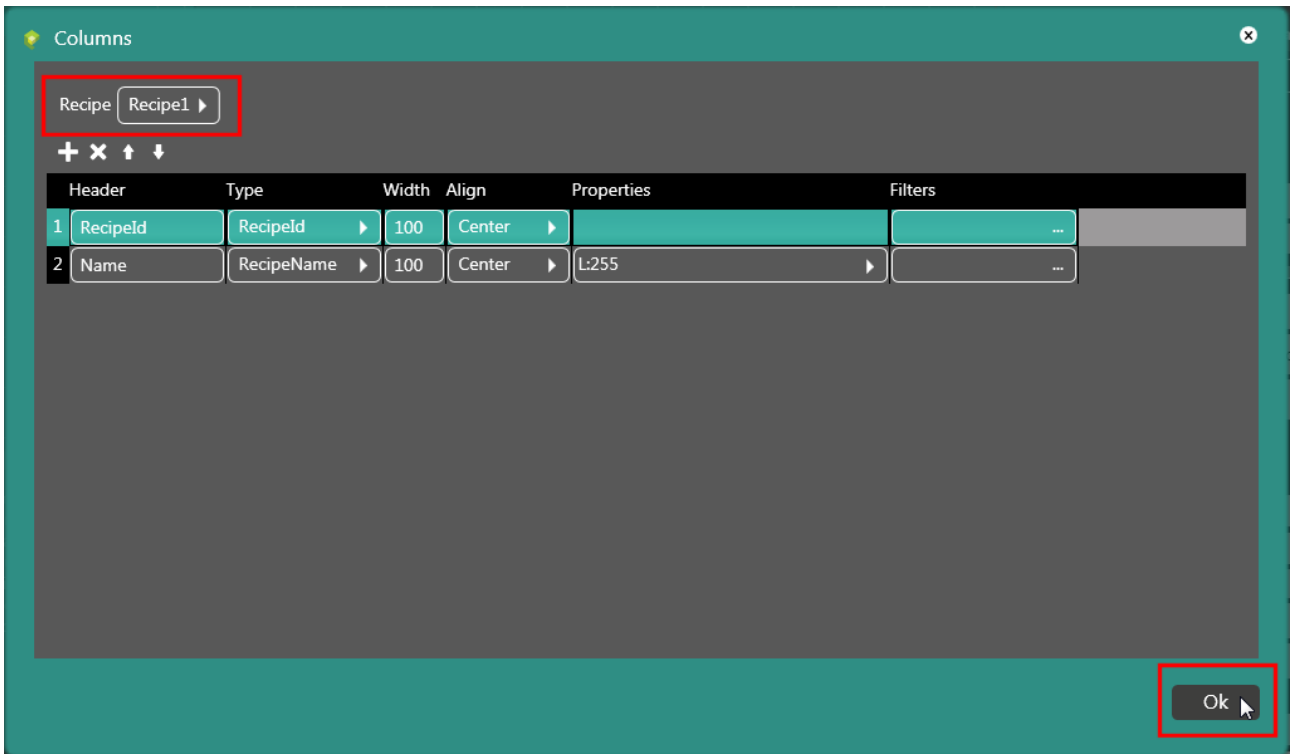
# CREW Manual

Defining the structure of the “Menu” and the “Columns”.

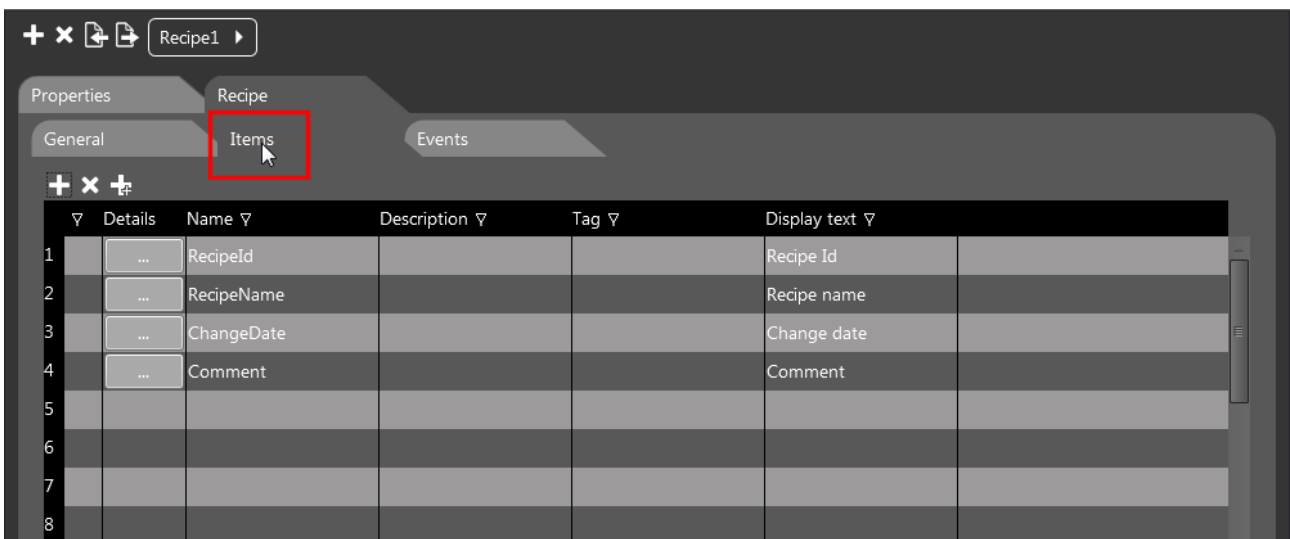


The screenshot displays the CREW software interface. On the left, two table designs are shown side-by-side on a grid background. The first table has two columns: 'Name' and 'Value'. The second table has two columns: 'RecipeId' and 'Name'. On the right, the 'Properties' panel is open, showing settings for a '(RecipeList)' object. The 'Columns' property is highlighted with a red box and contains the text 'RecipeId; Name'. Other visible properties include 'StrokeColor' (Black), 'StrokeSize' (1), 'Opacity' (255), 'Color' (Gray), 'Menu' (Sliding-(RecipeLoad);(Recipe ...), 'Header' (HasHeader: On, HeaderFont: [Default], HeaderFontSize: 12, HeaderFillColor: Gainsboro), and 'Row' (FontName: [Default], FontSize: 12, CellFillColor: White, AlternateColors: Off).

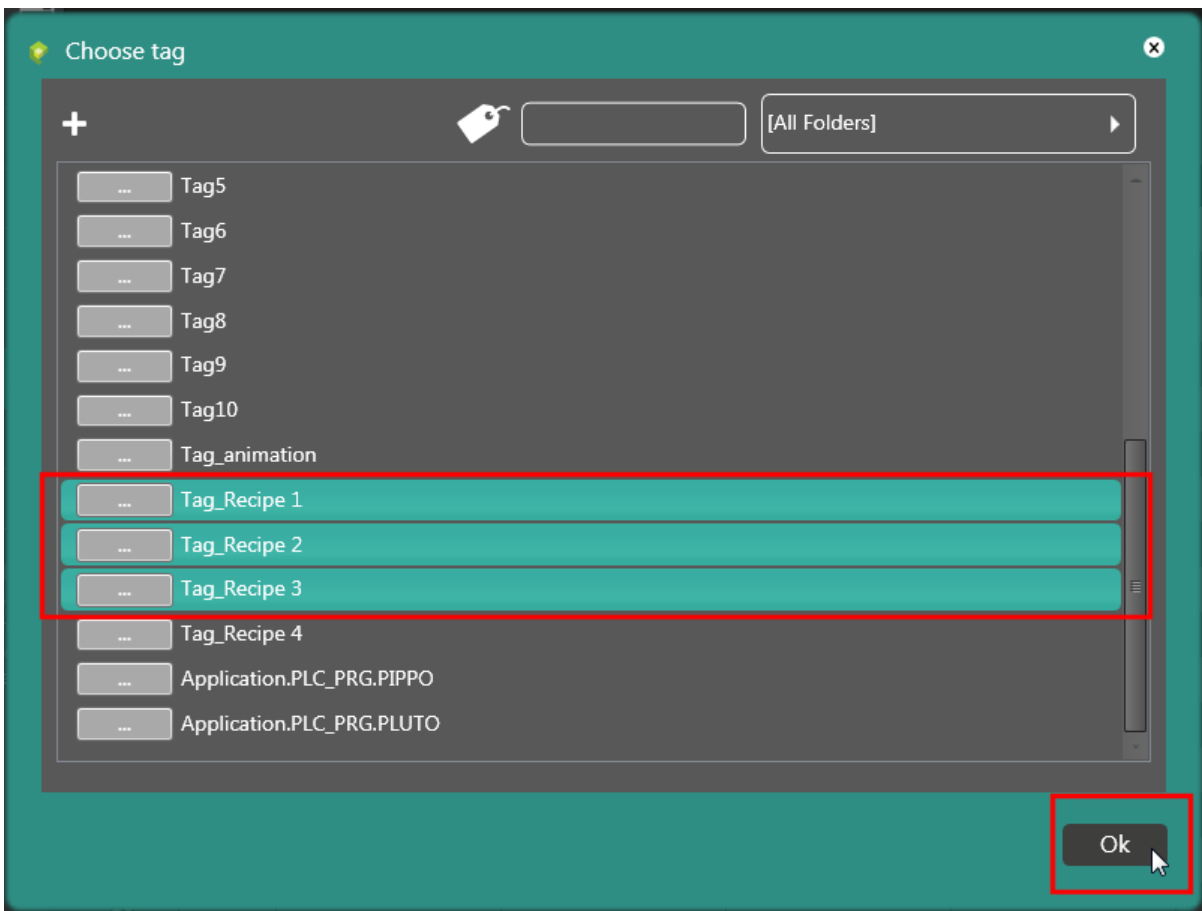
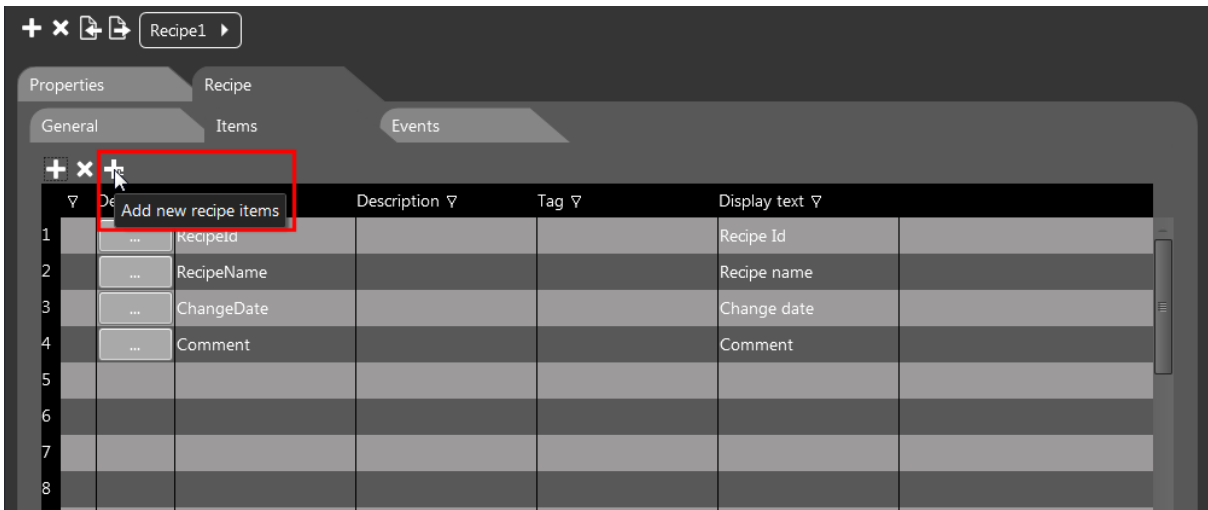
# CREW Manual



Adding elements to the recipe.



# CREW Manual





# CREW Manual

Recipe1

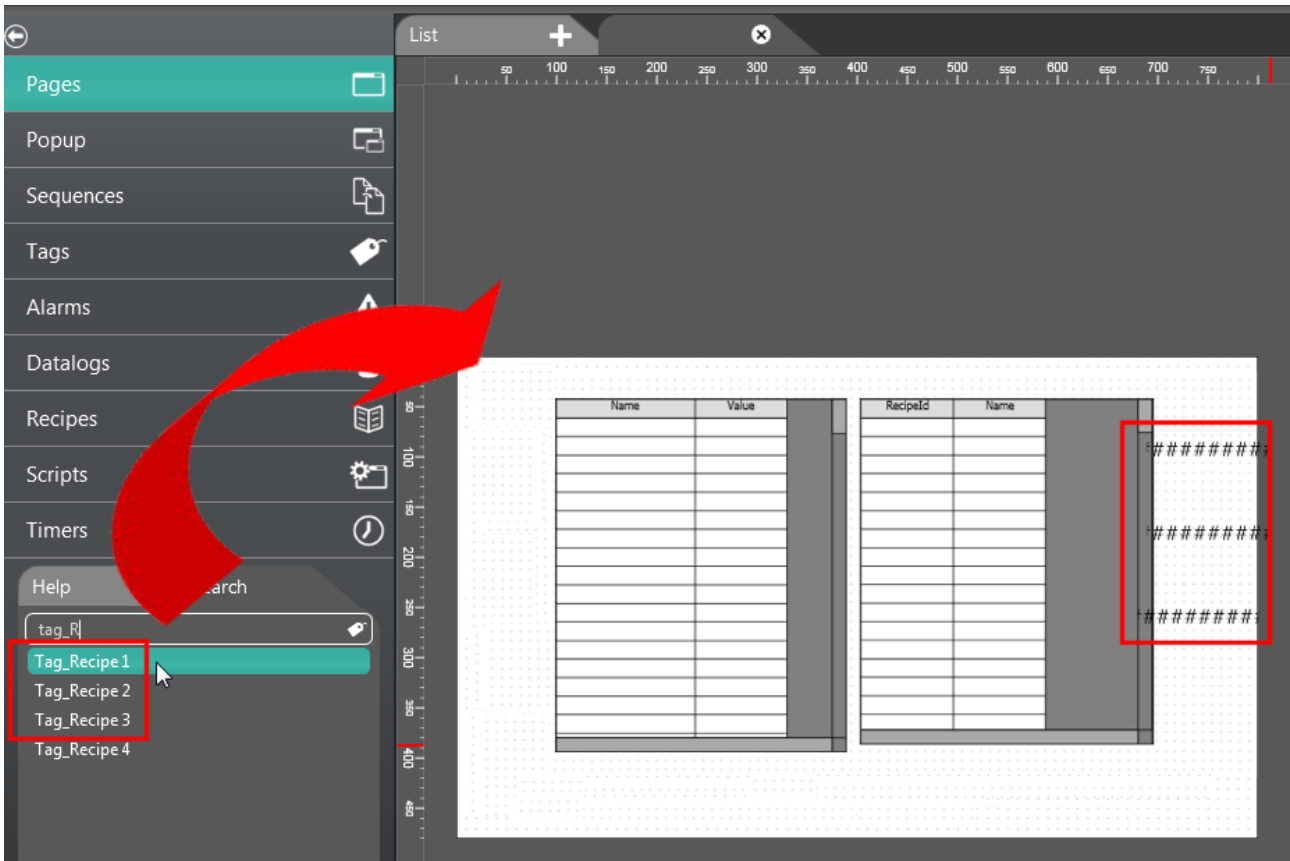
Properties Recipe

General Items Events

	Details	Name	Description	Tag	Display text
1	...	RecipeId			Recipe Id
2	...	RecipeName			Recipe name
3	...	ChangeDate			Change date
4	...	Comment			Comment
5	...	Tag_Recipe 1		Tag_Recipe 1	Tag_Recipe 1
6	...	Tag_Recipe 2		Tag_Recipe 2	Tag_Recipe 2
7	...	Tag_Recipe 3		Tag_Recipe 3	Tag_Recipe 3
8					
9					
10					

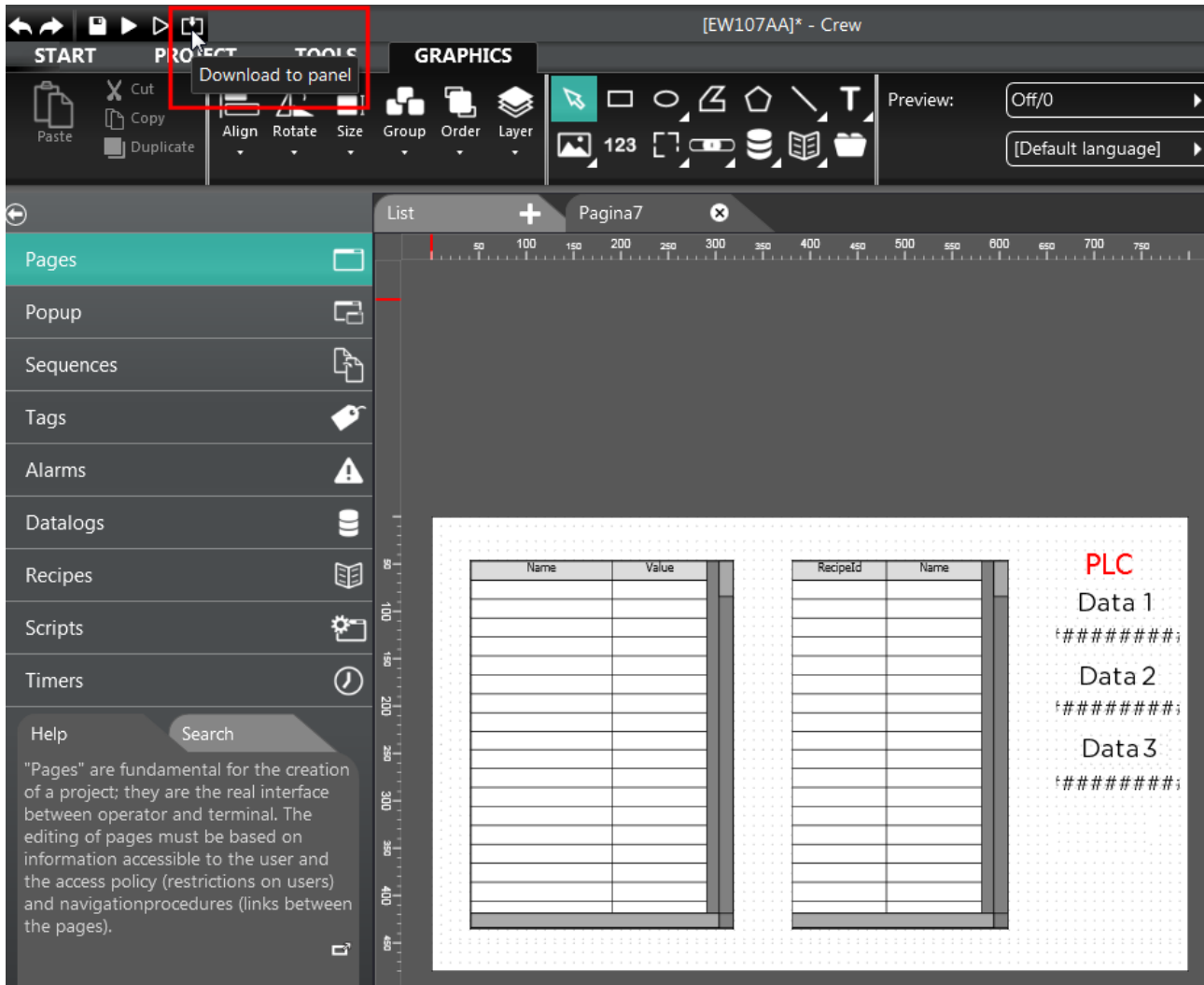
# CREW Manual

Add the "Tag\_Recipe 1", "Tag\_Recipe 2" and "Tag\_Recipe 3" objects to the page, dragging them one at a time with the mouse.



# CREW Manual

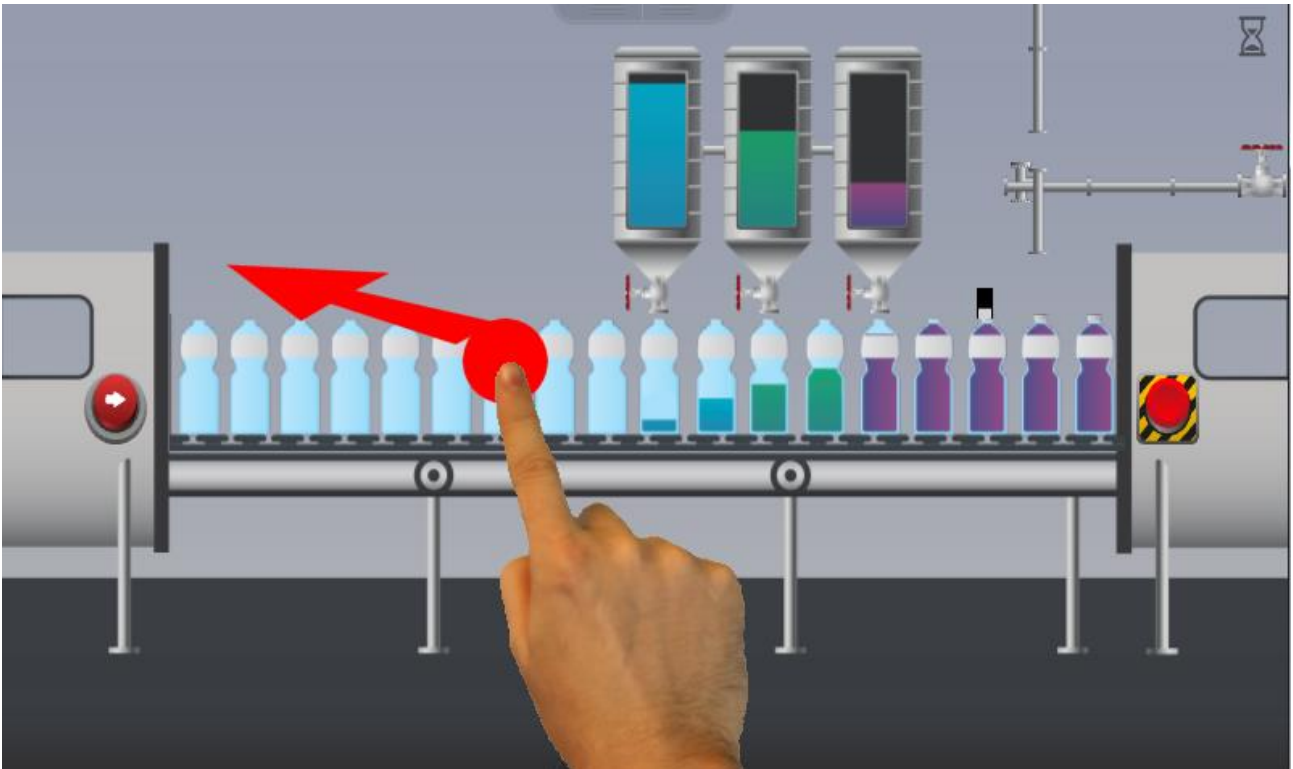
Download the project to the EW terminal.



# CREW Manual

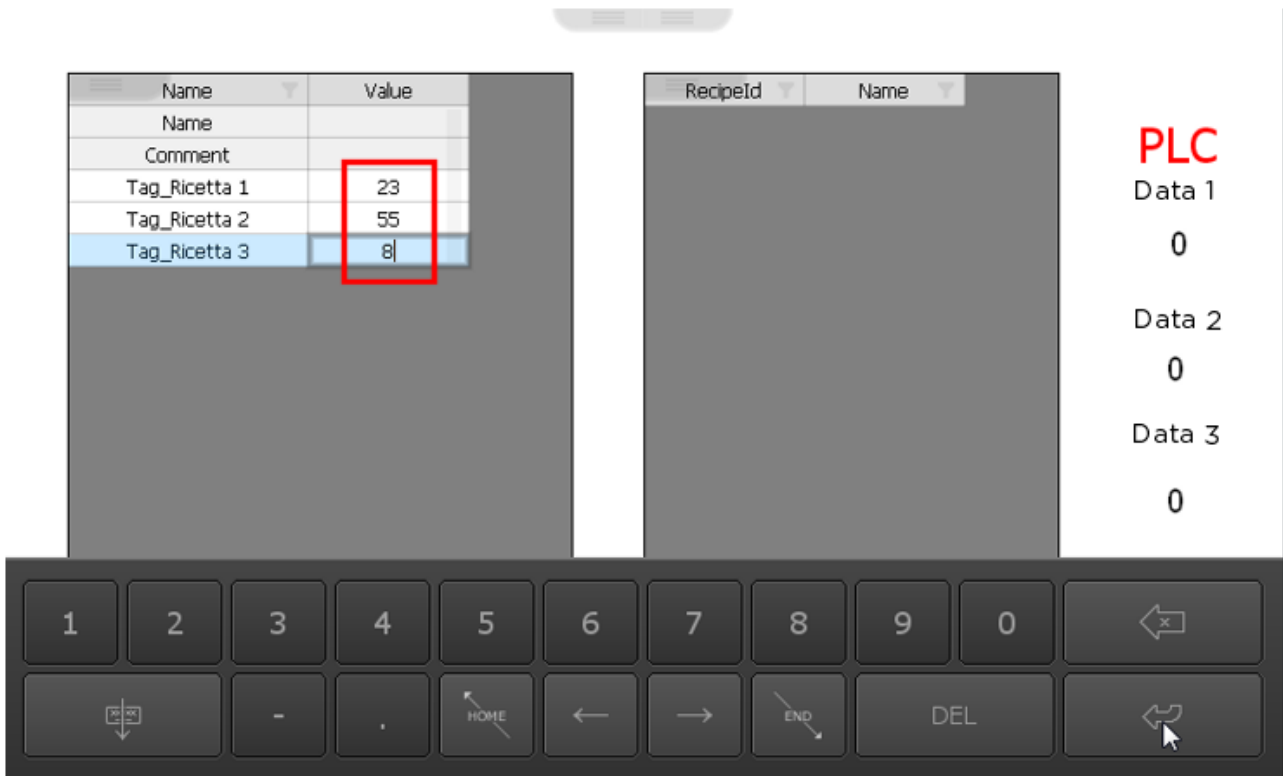
## Recipe Configuration - EW Terminal Side

When the project is downloaded, from the terminal go to the recipes page.



# CREW Manual

Assign values to the "Tag\_Recipe 1", "Tag\_Recipe 2" and "Tag\_Recipe 3" fields (23, 55, 8). To do so, select them one at a time.



The screenshot displays the CREW Manual interface. On the left, a table lists recipe tags and their corresponding values. The values 23, 55, and 8 are highlighted with a red box. On the right, the PLC Data panel shows three data points, each with a value of 0. At the bottom, a numeric keypad is visible, used for entering the values.

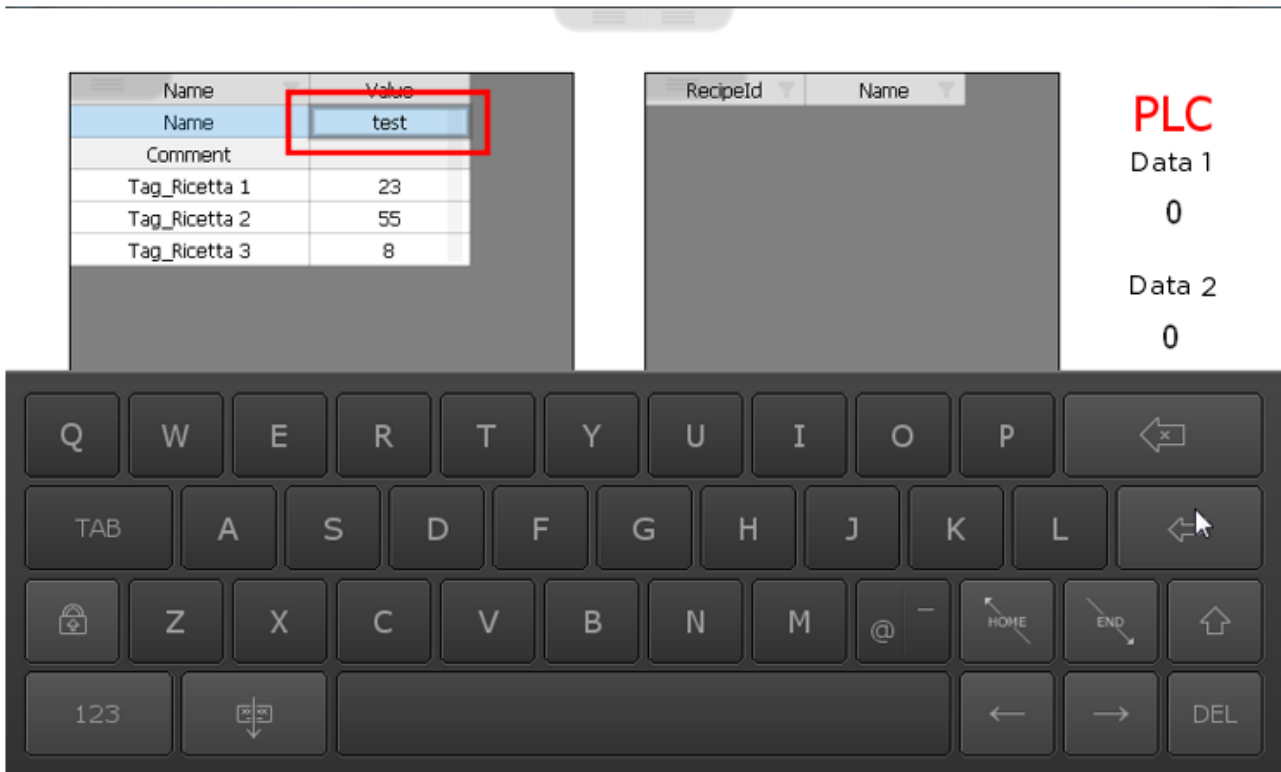
Name	Value
Name	
Comment	
Tag_Ricetta 1	23
Tag_Ricetta 2	55
Tag_Ricetta 3	8

**PLC**  
Data 1  
0  
Data 2  
0  
Data 3  
0

1 2 3 4 5 6 7 8 9 0  
- . HOME ← → END DEL

# CREW Manual

Give the recipe a name (“test”).



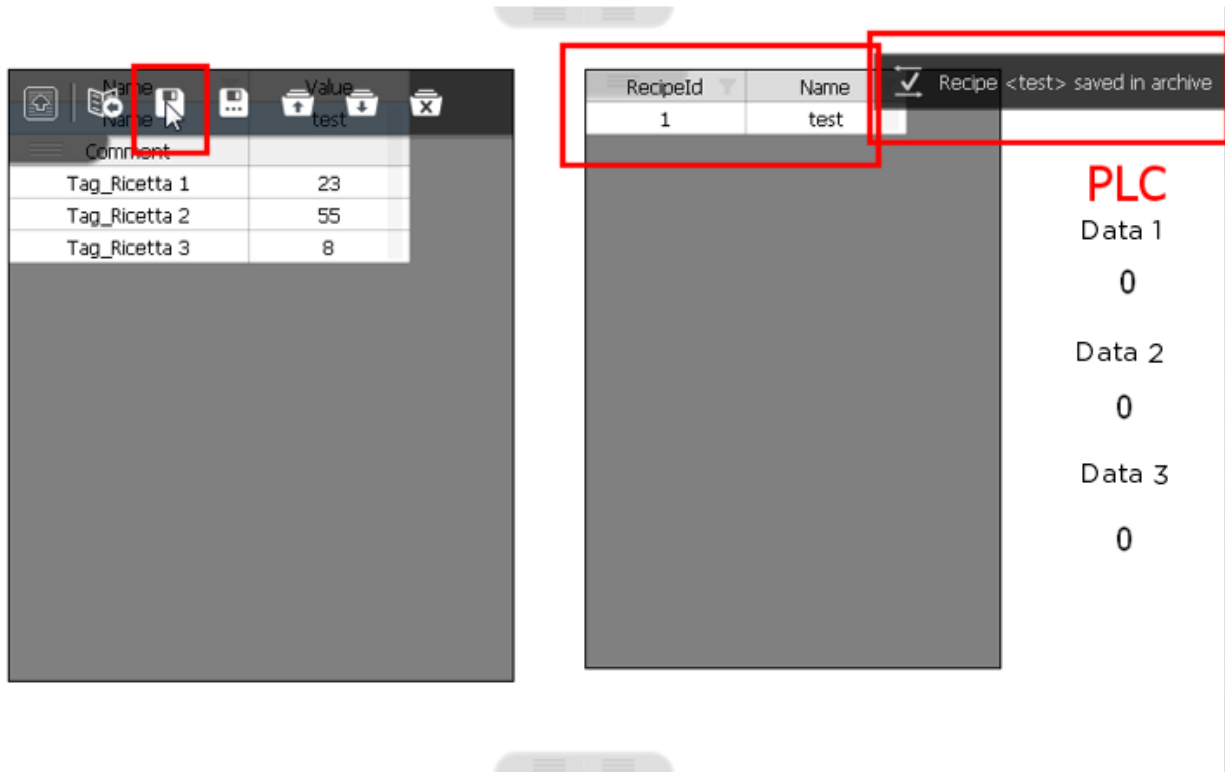
# CREW Manual

Make the drop down menu scroll down.



# CREW Manual

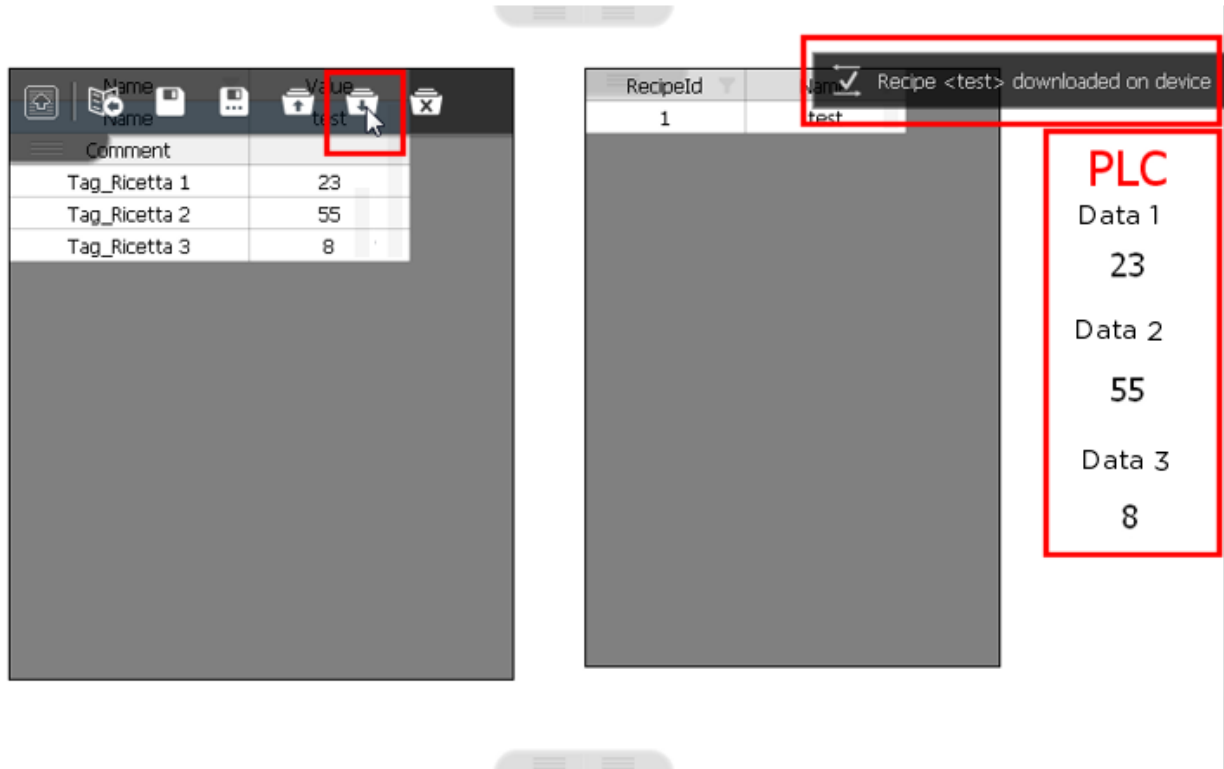
Select the “Save” icon. The saved recipe immediately appears in the recipe view.





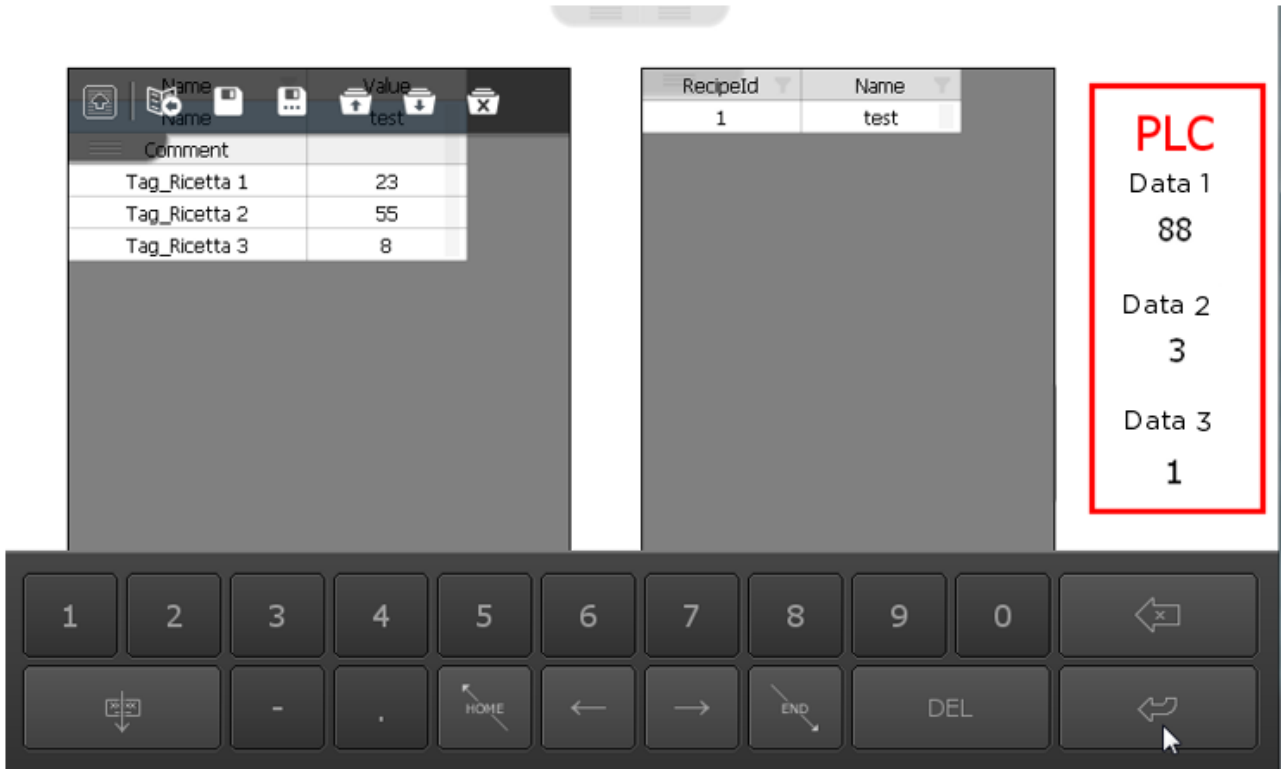
# CREW Manual

Select the "RecipeDownloadBuffer" icon if you wish to transfer the recipe to the PLC data buffer. Accordingly the recipe is immediately downloaded to the PLC.



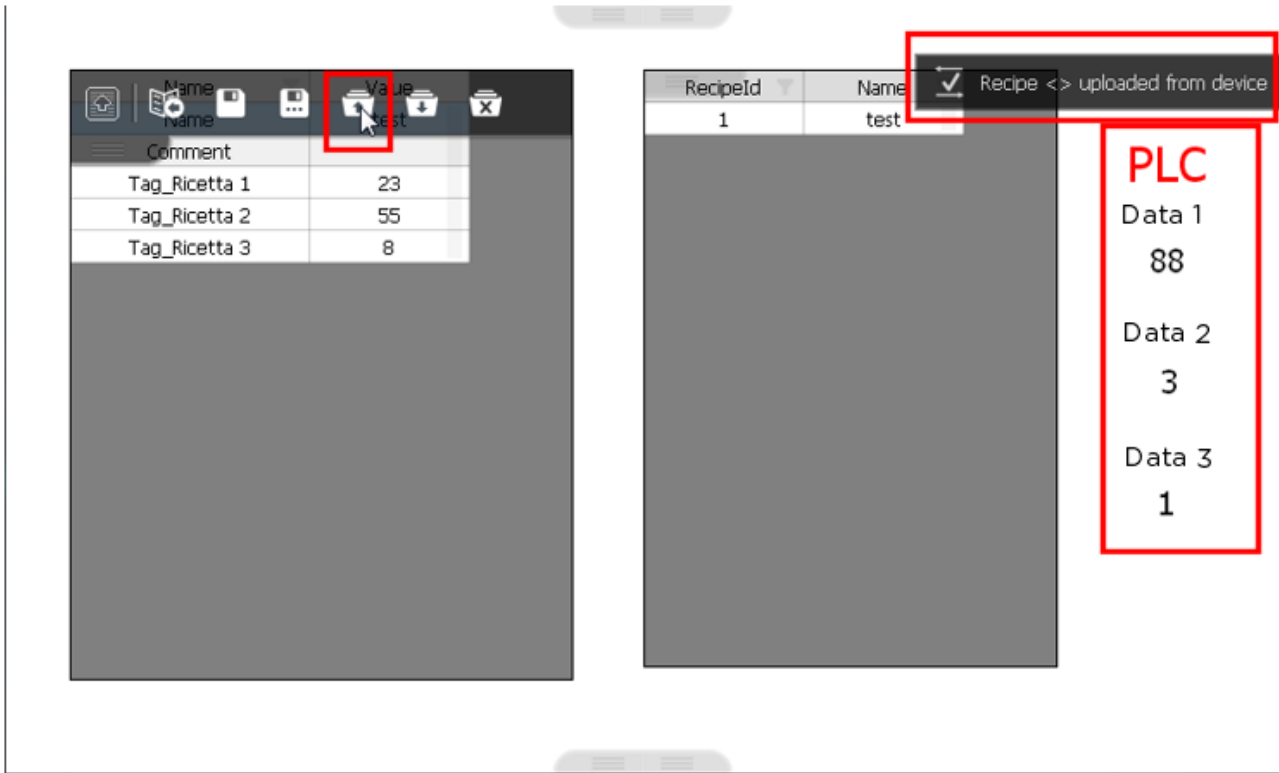
# CREW Manual

Now, if you wish to change the recipe data on the PLC, select the data on the PLC one at a time and change their values (88, 3, 1).



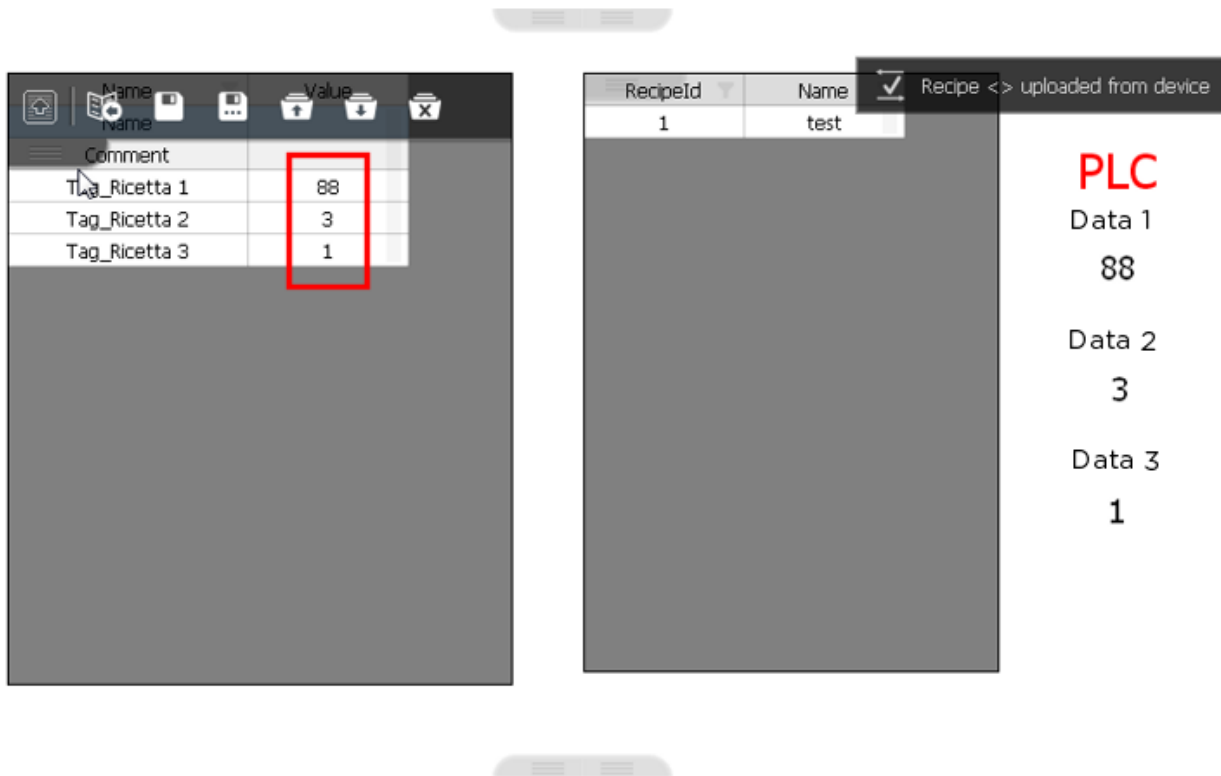
# CREW Manual

To load the new recipe from PLC to terminal, select the "Recipe UploadBuffer" icon.



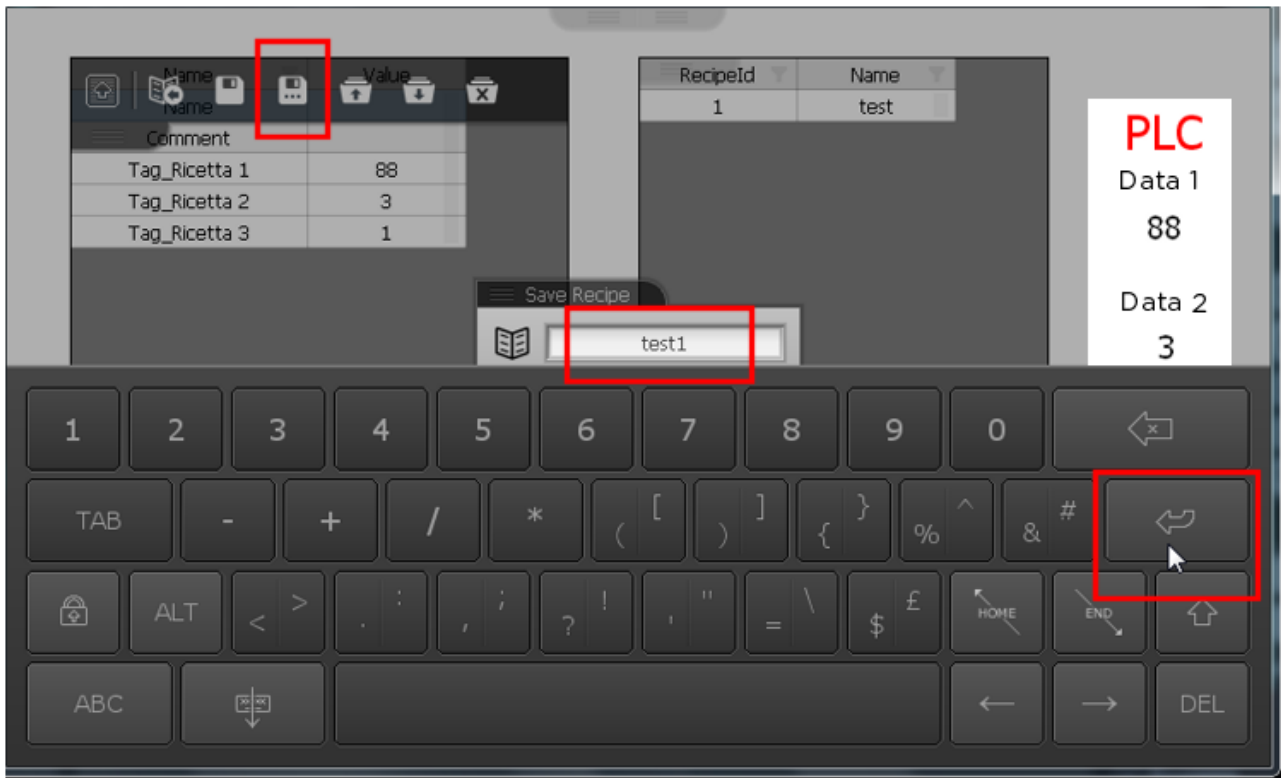
# CREW Manual

The new recipe is immediately imported to the terminal.



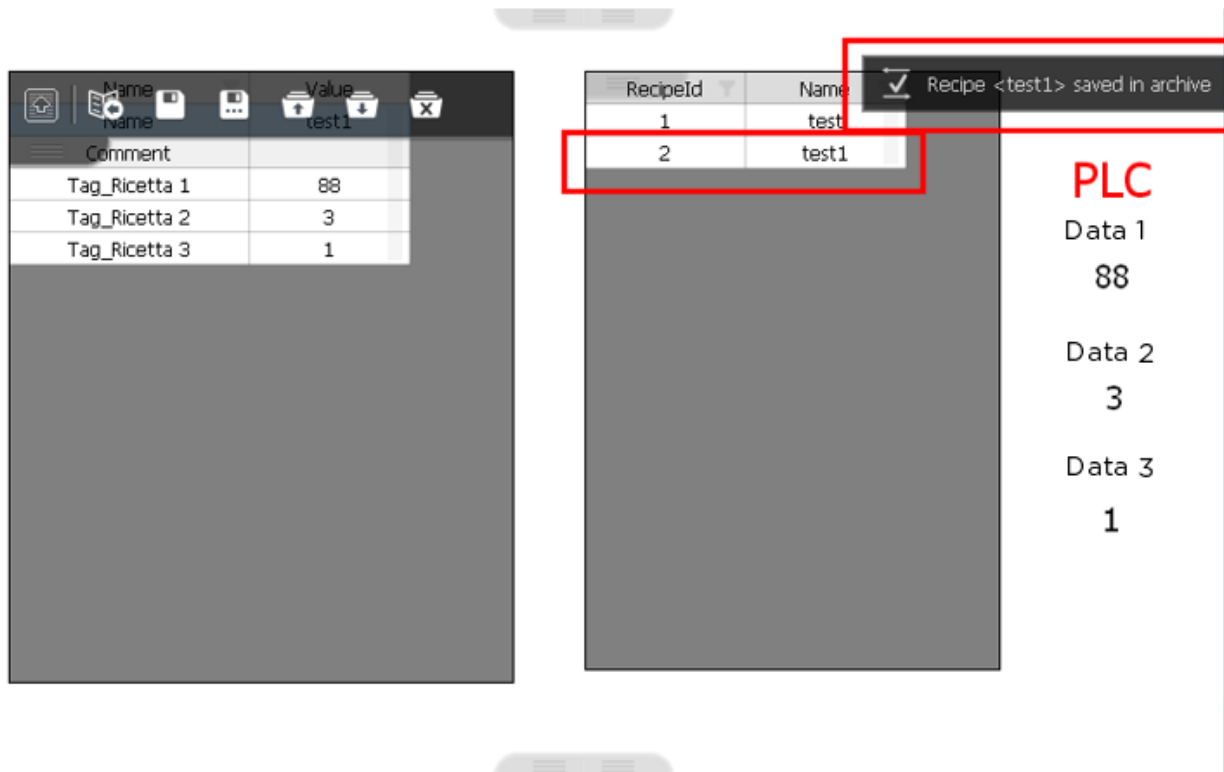
# CREW Manual

To save the new recipe, select the "SaveAs" ("test1") icon.



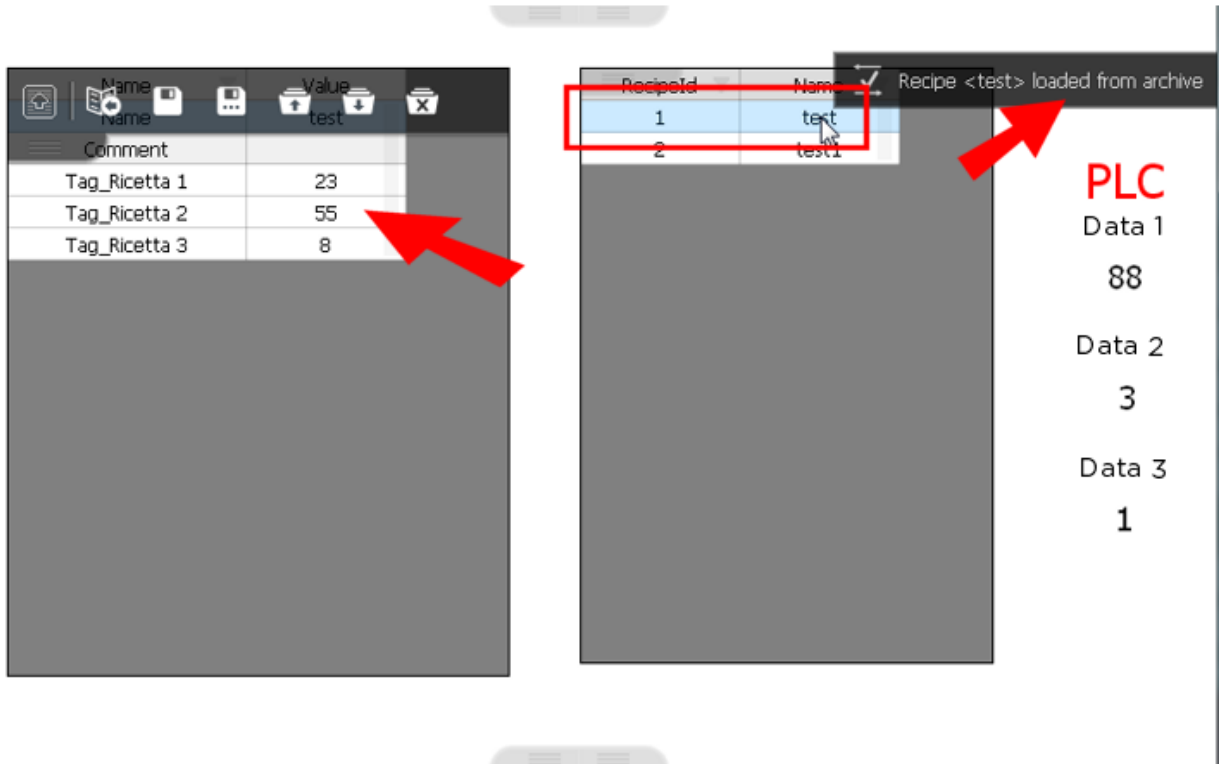
# CREW Manual

Accordingly the new recipe "test1" immediately appears in the recipe view.



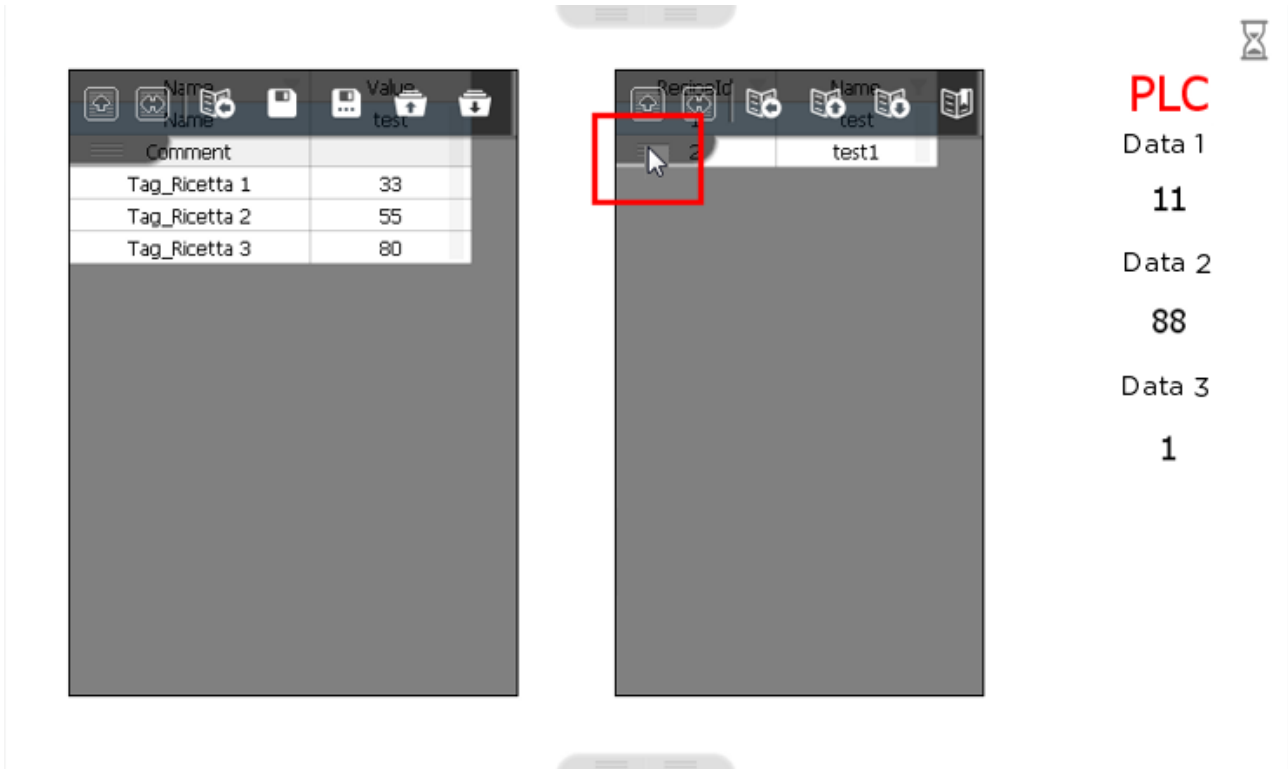
# CREW Manual

To reload the data from the old “test” recipe to the terminal, simply select it from the recipe view and it will immediately appear in the recipe on the terminal.



# CREW Manual

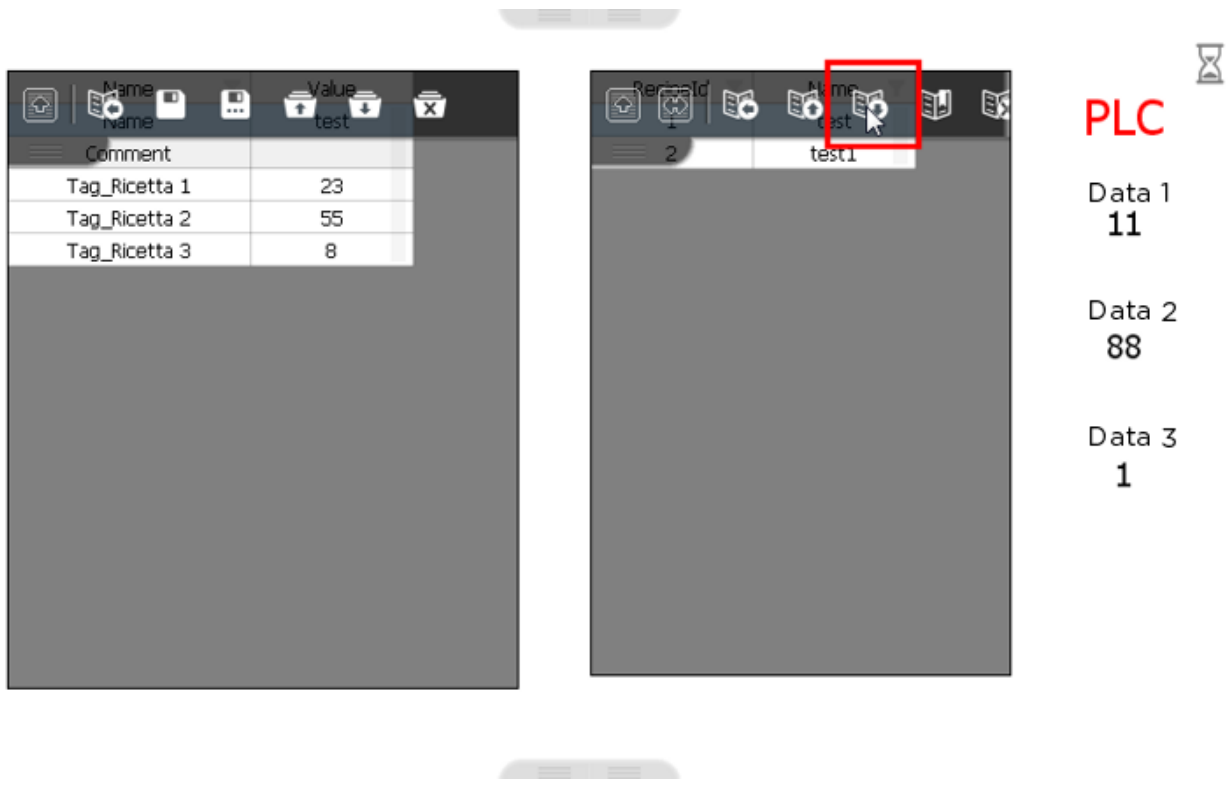
To transfer the “test” recipe from the recipe view to the PLC, scroll down the “Recipe View” drop down menu.





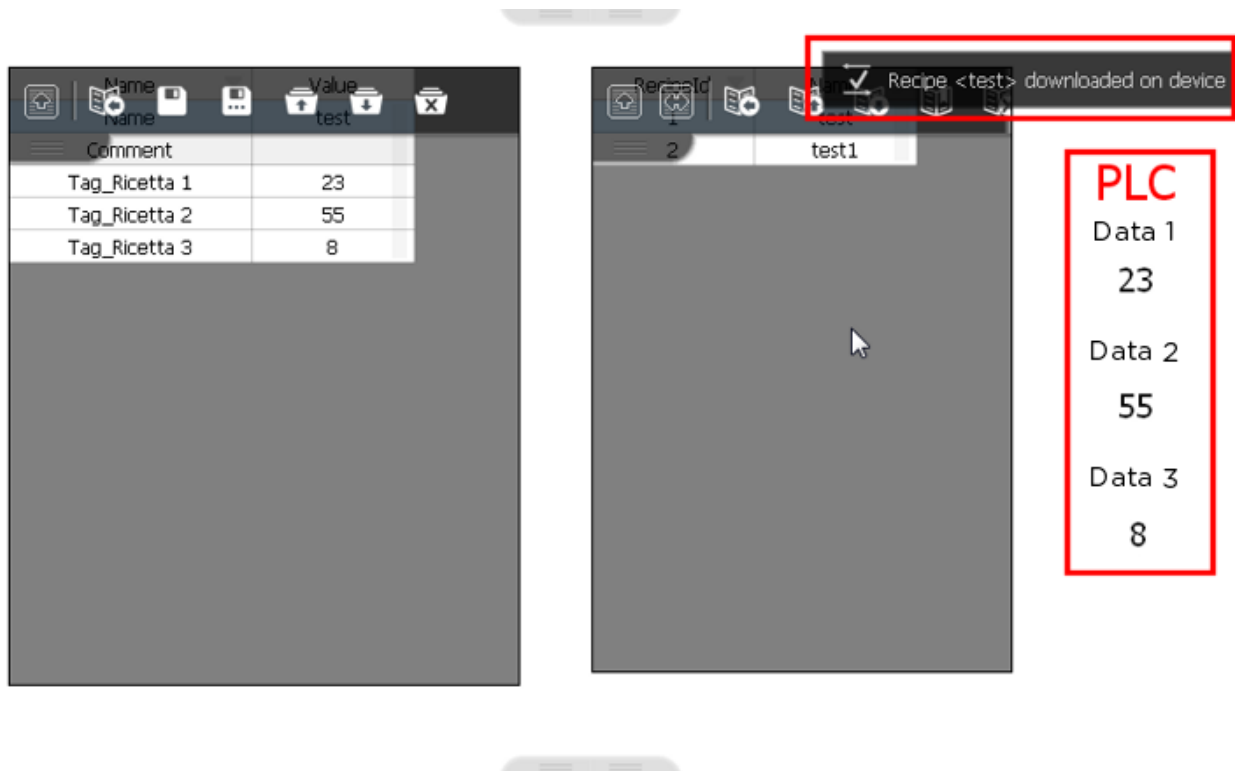
# CREW Manual

Select the "RecipeDownload" icon.



# CREW Manual

The recipe is immediately downloaded to the PLC.



# CREW Manual

---

## Examples of how Script is used

### Example 1 - Analysis of variables and launching events

In this example there is a project where a page, a variable, an alarm and controls associated to the page are configured.

With Crew it is possible to set the objects that are needed during script execution. Import an “Integer” variable calling it “Tag” (the names of the objects assigned by Crew are important as it is their access key from Script), assigned with an initial value of 0.

Then set a generic alarm (“Alarm”) that is triggered when the “Tag” variable has a value of “10”. Remember to set the view of one of the applicable alarm signals in “Alarms” in

the “Alarm Signals” mask (see “Alarm Signals” section).

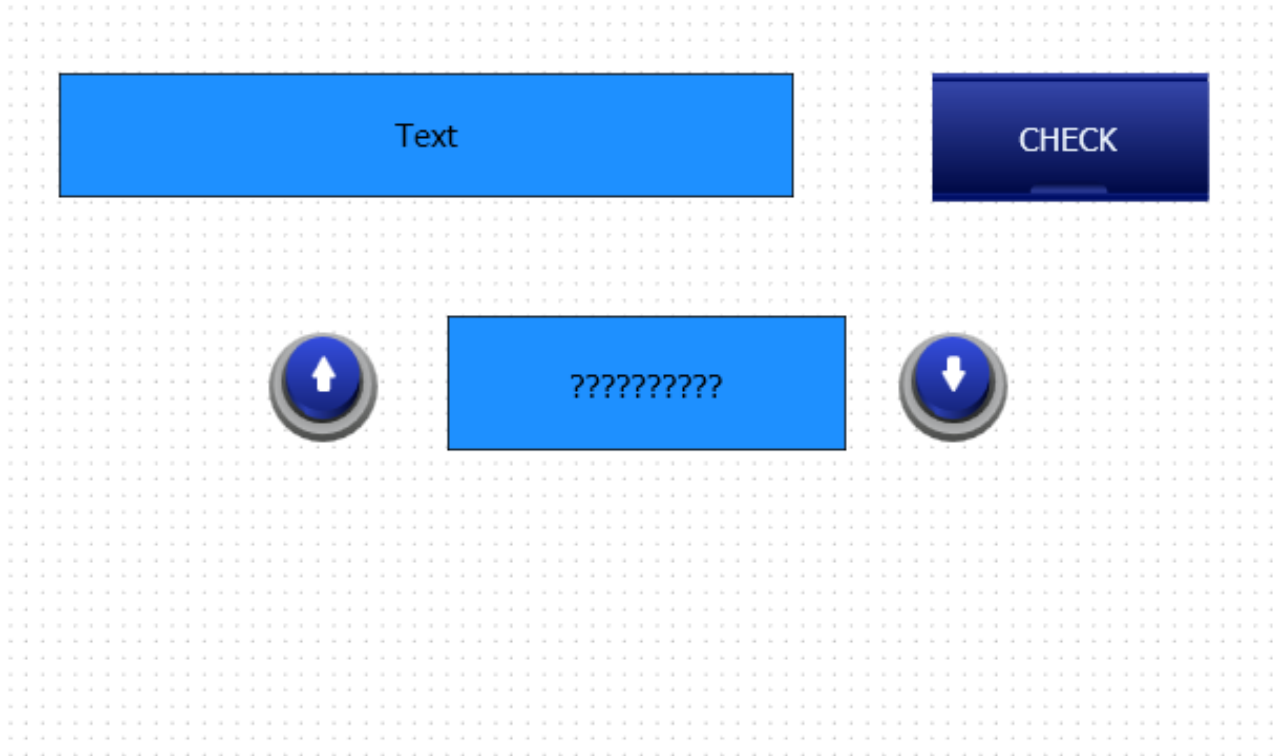
Set a page called “Page” and enter a label (called “Label”) and a Touch button to assign a Script to on the “onReleased” event.

Drag the variable from “Explore Project” to the work area so as to create a dynamic field that shows the value in Runtime (useful for constantly monitoring the value).

Add two buttons and associate the predefined increase-decrease value functions from the “Tag” variable so as to edit the value in Runtime.

# CREW Manual

The created page appears as below.



The script will need to take the value of the “Tag” variable, check whether the value is less than 5 and, if not, launch an alarm, change the layout of the label and the page, and lower the value.

To retrieve the value of the variable, use the ESATAG object and store it in an “a” variable with the following instruction:

```
a=ESAHMI.ESATAG.ReadValue (“Tag”)
```

This runs the analysis of the obtained value. If the value is greater than or equal to 5, the alarm is launched. From Crew the alarm was set to be triggered when a value of 10 was reached, therefore with the certainty that the script activated it. The control and trigger code uses the ESAALARMMGR object as stated in the following lines:

```
If a>4 Then
ESAHMI.ESAALARMMGR.RaiseAlarm("Alarm")
End If
```

# CREW Manual

It is also possible to run other instructions in the same condition so that when the value of the variable is changed and the script is launched, other changes are also applied.

For example, changing the text, colour and label blink (ESACNTRL object, remember

to call the label's Draw method) and the background of the page (ESAPAGE object) as shown below:

If a>4 Then

```
ESAHMI.ESAALARMMGR.RaiseAlarm("Alarm")
```

```
ESAHMI.ESAPAGE("Page").ESACNTRL("Label").TextValue="Error in the value"
```

```
ESAHMI.ESAPAGE("Page").ESACNTRL("Label").AreaColor=RGB (23,123,43)
```

```
ESAHMI.ESAPAGE("Page").ESACNTRL("Label").BorderColor=RGB (54,245,13)
```

```
ESAHMI.ESAPAGE("Page").ESACNTRL("Label").Border-Blink=2
```

```
ESAHMI.ESAPAGE("Page").ESACNTRL("Label").Draw()
```

```
ESAHMI.ESAPAGE("Page").AreaColor=RGB(25,25,25)
```

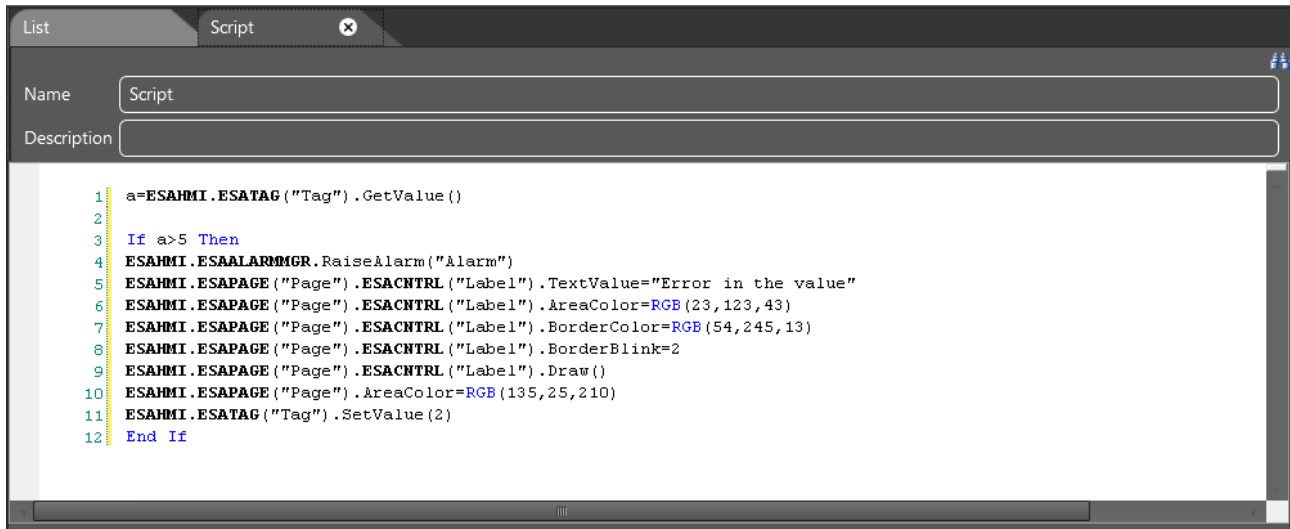
End If

Lastly, re-establish a valid value for the variable with the following instruction:

```
ESAHMI.ESATAG.WriteValue "Tag",2
```

# CREW Manual

The final code entered in the Crew editor is the one shown in the image.



```

1 a=ESAHMI.ESATAG("Tag").GetValue()
2
3 If a>5 Then
4 ESAHMI.ESAALARMGR.RaiseAlarm("Alarm")
5 ESAHMI.ESAPAGE("Page").ESACNTRL("Label").TextValue="Error in the value"
6 ESAHMI.ESAPAGE("Page").ESACNTRL("Label").AreaColor=RGB(23,123,43)
7 ESAHMI.ESAPAGE("Page").ESACNTRL("Label").BorderColor=RGB(54,245,13)
8 ESAHMI.ESAPAGE("Page").ESACNTRL("Label").BorderBlink=2
9 ESAHMI.ESAPAGE("Page").ESACNTRL("Label").Draw()
10 ESAHMI.ESAPAGE("Page").AreaColor=RGB(135,25,210)
11 ESAHMI.ESATAG("Tag").SetValue(2)
12 End If

```

## Example 2 - Page access based on user-level

Another example of Script use is the possibility of managing access to protected pages based on the level of the user currently logged onto the terminal.

On Crew, set the objects that are required during script execution and two user levels (see "Groups" section). For example, assign a password for levels 3 and 8.

Remember that when the project is started, the predefined level is 10, namely, the lowest.

On the default "Page" enter three buttons: one of them that retrieves the script, one the login function and the last one the logout function.

Lastly, set two new pages ("Page\_1" and "Page\_2") that are retrieved by the script based on user level.

To implement the code it is firstly necessary to retrieve the level of the currently logged user, through the USERMGR object:

```
a=ESAHMI.ESAUSERMGR.GetCurrentUserLevel()
```

# CREW Manual

Now create a test condition for this level (the function returns an integer).  
Based on user credentials, the following page will open:

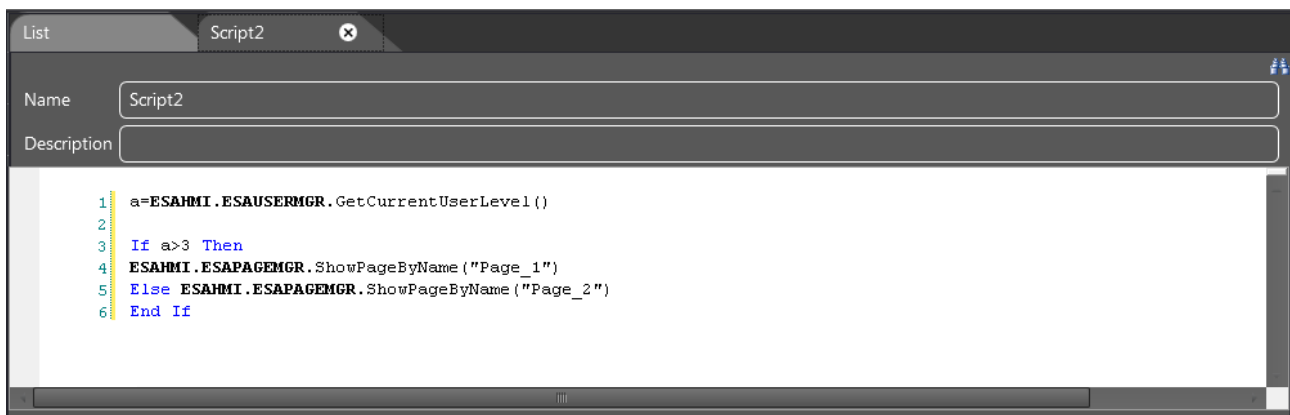
If a>3 Then

ESAHMI.ESAPAGEMGR.ShowPageByName("Page\_1")

Else ESAHMI.ESAPAGEMGR.ShowPageByName("Page\_2")

End If

The complete script code is stated in the following image.



```
1 a=ESAHMI.ESAUSERMGR.GetCurrentUserLevel()  
2  
3 If a>3 Then  
4 ESAHMI.ESAPAGEMGR.ShowPageByName("Page_1")  
5 Else ESAHMI.ESAPAGEMGR.ShowPageByName("Page_2")  
6 End If
```

# CREW Manual

## Example 3 - Exporting Alarms to user-selected file

Another example of how Crew scripts can be used refers to the use of value fields to receive data to be used in invoking dynamic functions.

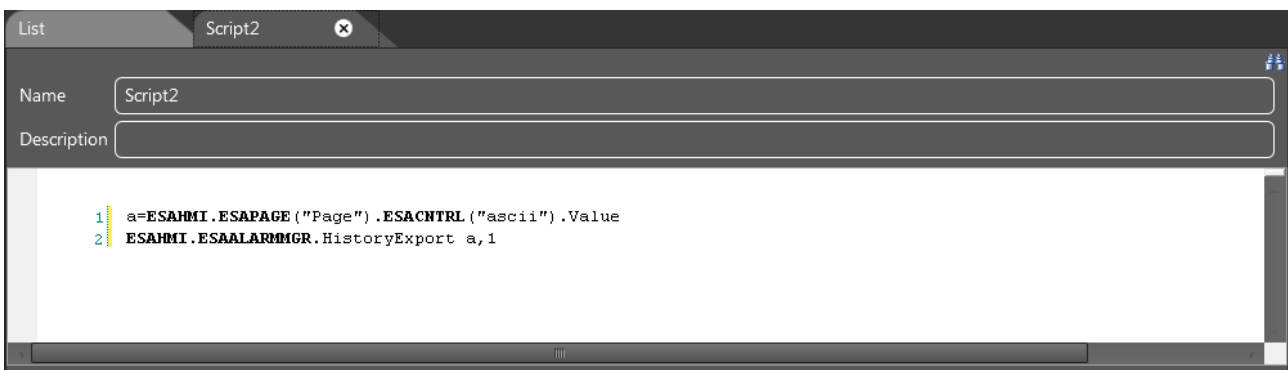
On a page, add a complex field that displays the alarm log, an Ascii field (“ascii” associated to a string variable) and a button with a script associated to it (onReleased event).

The script reads the value of the Ascii field and saves it in a variable “a” with the following instruction:

```
a=ESAHMI.ESAPAGE("Page").ESACNTRL("ascii").Value
```

Then, invoke the alarm exporting function that needs to be given the newly-read string:

```
ESAHMI.ESAALARMGR.HistoryExport a,1
```



This is only an example of the ductility of programming through scripting, which provides the project with a very high level of dynamicity.



# CREW Manual

## Example 4 - Storing Recipe on memory support

This paragraph illustrates how it is possible to force loading, saving and exporting some recipes - among other things - with a script, if a bit on the device is placed high.

To achieve this purpose, associate this script to the OnValoreCambiato/OnValueChanged event of the control bit (in this case the "Control" variable). The PLC raises the status of the bit every X minutes and starts the script.

In the project, create a type of recipe called "Doses" (defined according to preference), which is used in the script.

This example also introduces the use of a function that runs a check on a variable and returns a value. In particular, the values relative to the days, months, hours, minutes and seconds returned by the VBSCRIPT functions can be values of one digit. In order for all of the saved files to have the same format and length, write a function of a few rows that adds a 0 in front of the digit, if this is a value of less than 10.

From Crew create a script with the usual techniques, while, from the general page, it is necessary to assign a name to it ('addzero'), a returned value type (Variant) and an input value (number "value").

The structure of our function has now been created and all that is left to do is write the code:

```
If value<10 Then  
value="0" & value  
End If  
addzero=value
```

If the input value ("value") is less than 10 (i.e. only one digit), add string "0" to the variable and lastly, return the "value" value (if you do not access the if cycle the function simply returns the value received in input).

The following is an application example of this function:

it is invoked with a value of 5, addzero(5), and returns the value of "05".

# CREW Manual

Stay on the main script code:

```
a=ESAHMI.ESATAG("Controllo").GetRawValue()
```

Firstly, read the rough value of the control variable: if it has a value of 1, run the operations (this avoids them from being executed when the bit switches from 1 to 0).

The If cycle is the following:

```
If (a=1) Then
```

```
ESAHMI.ESARECIPEMGR.RecipeBufferUpload "Dosaggi",0
```

```
ESAHMI.ESARECIPEMGR.SaveRecipe "Dosaggi","Ricetta",0
```

```
End If
```

On the first rows of the cycle, upload the recipe loaded on the PLC (“Doses” type), and on the second row save the recipe on the terminal with the name “Recipe”.

The only remaining step is saving, which can be performed with the following instruction:

```
ESAHMI.ESARECIPEMGR.RecipeExport dest,"Dosaggi",
```

```
ESAHMI.ESATAG.WriteValue "Controllo",0
```

Exporting is carried out on all recipes (the third parameter is an empty string) and saving to the file identified in the “dest” string variable that you wish to build. After saving the control bit is placed on 0.

The “dest” string is built by inserting the details relative to date and time of when the operation was carried out. These dates can simply be retrieved using the functions placed at disposal by the VBScript programming language:

```
hours=Now()
```

```
date=Date()
```

```
day=addzero(Day(date))
```

```
month=addzero(Month(date))
```

# CREW Manual

```
year=Year(date)
```

```
hour=addzero(Hour(time))
```

```
minute=addzero(Minute(time))
```

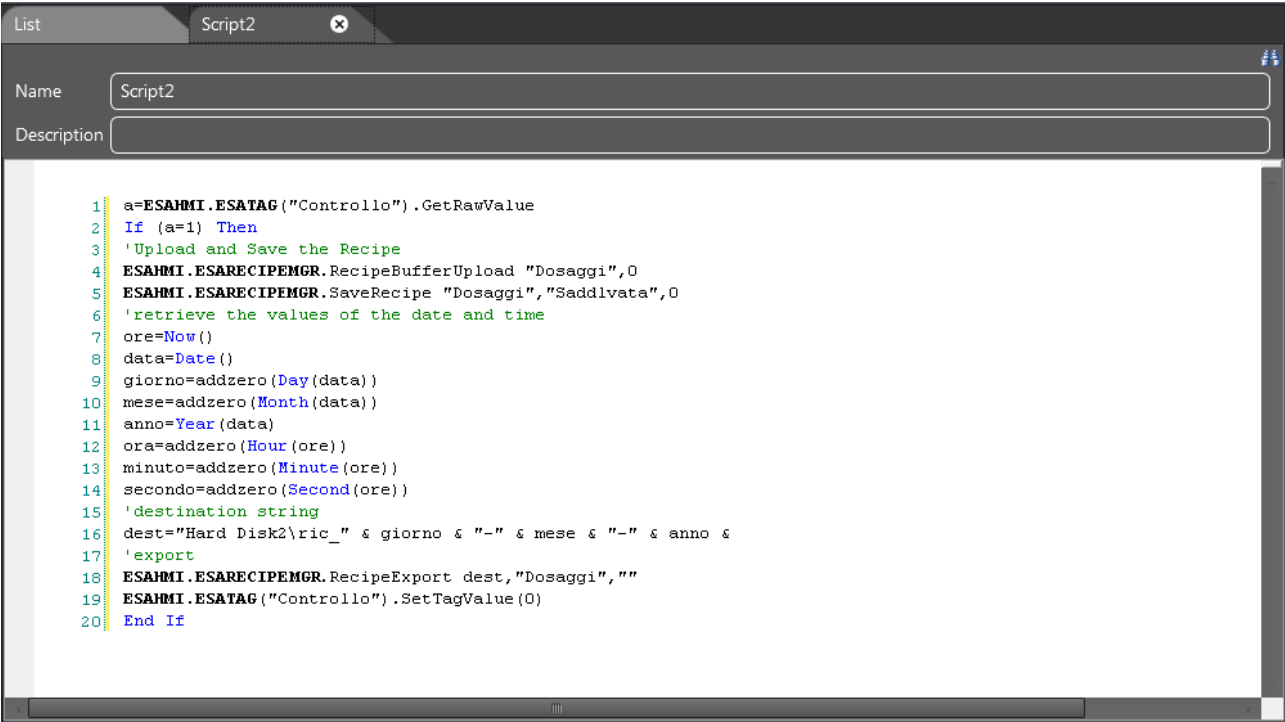
```
second=addzero(Second(time))
```

```
dest="Hard Disk2\ric_" & day & "-" & month & "-" & year & "_h" & hour & "." &
minute & "." & second & ".xml"
```

As you can see, the day, month, hour, minute and second variables are passed onto the previously defined addzero function, where 0s are added to single-digit values.

The last instruction involves building the “dest” string which identifies the path and name of the file that the recipes are exported to. In this case, save on a support named “Hard Disk2” (which can be a USB pendrive for example) with a name like “ric\_02-12-2005\_h12.13.08.xml”. Accordingly you will be sure to have a series of exports distinguished in files with unequivocal names inside the support.

Below is a full view of the newly-configured script.



```

1 a=ESAHMI.ESATAG("Controllo").GetRawValue
2 If (a=1) Then
3 'Upload and Save the Recipe
4 ESAHMI.ESARECIPEMGR.RecipeBufferUpload "Dosaggi",0
5 ESAHMI.ESARECIPEMGR.SaveRecipe "Dosaggi","Saddlvata",0
6 'retrieve the values of the date and time
7 ore=Now()
8 data=Date()
9 giorno=addzero(Day(data))
10 mese=addzero(Month(data))
11 anno=Year(data)
12 ora=addzero(Hour(ore))
13 minuto=addzero(Minute(ore))
14 secondo=addzero(Second(ore))
15 'destination string
16 dest="Hard Disk2\ric_" & giorno & "-" & mese & "-" & anno &
17 'export
18 ESAHMI.ESARECIPEMGR.RecipeExport dest,"Dosaggi",""
19 ESAHMI.ESATAG("Controllo").SetTagValue(0)
20 End If

```

# CREW Manual

## Example 5 - Deleting all of the recipes on EW

Using the methods described in this paragraph, it is possible to build customised functions based on your project needs. This example illustrates how to create a function of just a few rows, that causes the deletion of all of the recipes saved on EW. This can be helpful when you want to avoid manually deleting each single recipe.

Plus, a few code rows are also added, allowing you to “time” the execution time of the entire script (accordingly you will have an identification value of the time required to execute the script).

Code analysis:

```
t=Timer()  
R_Type="Ten_Var"
```

The first line specifies the moment that the script needs to begin (the Timer function returns the number of seconds elapsed after 12:00 AM) and save it in variable (t).

The second line defines the type of recipe to be completely deleted (alternatively it is possible to use this string value as a parameter for the function, as seen in example 4 for the “addzero” function).

Then, retrieve the name of the first recipe and save it in a variable (a):

```
a=ESAHMI.ESARECIPEMGR.GetFirstRecipeName(R_Type)
```

If there are no recipes for the identified type (R\_Type), the function returns an empty string (“”). Therefore cancellation must only be carried out if the returned string is different from “”.

Therefore use a Do While cycle to make the operation recursive.

```
Do While a<>""  
ESAHMI.ESARECIPEMGR.DeleteRecipe R_Type,a,0  
a=ESAHMI.ESARECIPEMGR.GetFirstRecipeName(R_Type)
```

# CREW Manual

## Loop

Note that the While cycle stays open as long as the value of “a” is different from empty string (namely as long as there are saved recipes).

Deletion is carried out for the type identified at the beginning of the script and for the current value of “a” (recipe name). Plus, the value 0 is applied to avoid asking the operator for confirmation.

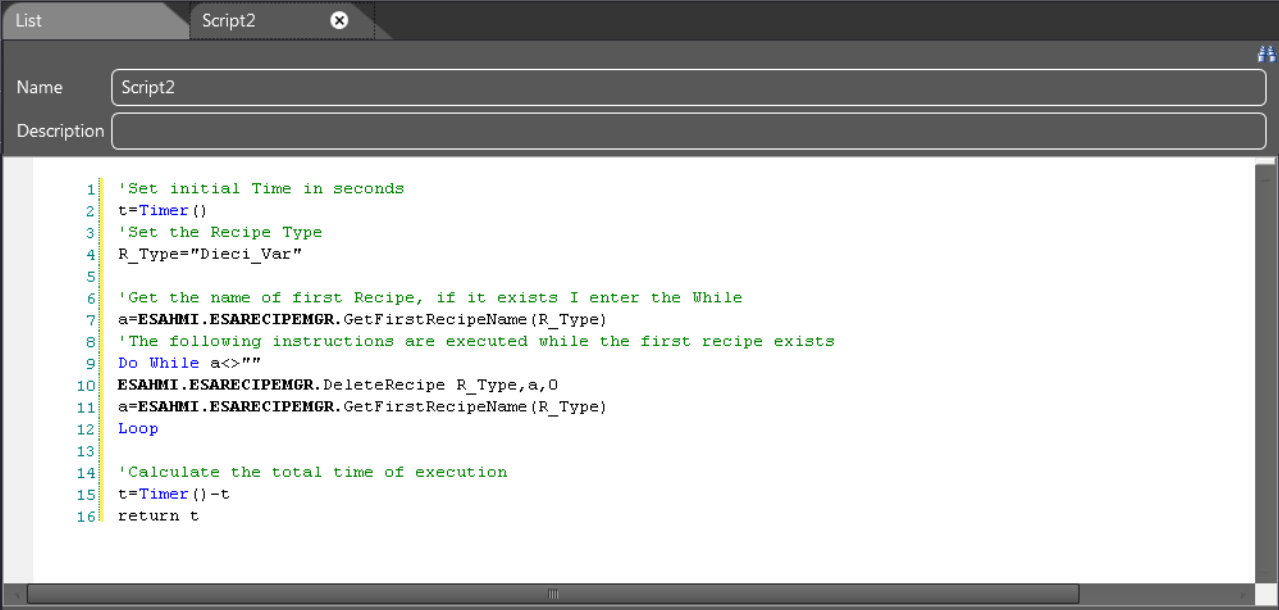
In the cycle, update the value of “a” by retrieving the first new recipe (still using GetFirst instead of GetNext because the deletion operation changed the order of the recipes).

When exiting the While cycle, all of the recipes are deleted and all you need to do is retrieve the time used by the script:

```
t=Timer()-t
return t
```

With this instruction, the value of “t” is updated by subtracting the value retrieved at the beginning of the script (saved in t) from the current Timer() value. Therefore at the end of this instruction “t” contains the number of seconds that elapsed between the beginning and end of elimination.

Below is the complete script code.



```

1 | 'Set initial Time in seconds
2 | t=Timer()
3 | 'Set the Recipe Type
4 | R_Type="Dieci_Var"
5 |
6 | 'Get the name of first Recipe, if it exists I enter the While
7 | a=ESAHMI.ESARECIPEMGR.GetFirstRecipeName(R_Type)
8 | 'The following instructions are executed while the first recipe exists
9 | Do While a<>""
10 | ESAHMI.ESARECIPEMGR.DeleteRecipe R_Type,a,0
11 | a=ESAHMI.ESARECIPEMGR.GetFirstRecipeName(R_Type)
12 | Loop
13 |
14 | 'Calculate the total time of execution
15 | t=Timer()-t
16 | return t

```

# CREW Manual

## Example 6: Print recipe list in paper

The following example illustrates the use of the print functions. In the example we want to print the list of recipes contained in the terminal's memory onto a sheet of paper. The recipe search logic is similar to the one used in the last example.

Firstly, initialise the printing session with the Start method. With parameter 1, in Runtime, the printing options box is shown before it starts.

It is necessary to treat the case when the user deletes the printing operation by clicking X in the box. This is executed with a If cycle that controls - and possibly prevents - the execution of all of the other code rows:

```
if (ESAHMI.ESAPRN.Start(1)=1) Then
```

Now create a page header with a title and with two white rows to separate title and content.

To leave the rows white, use the WriteLN method and apply an empty string. Before writing the title, set the font at a higher value, and then set it at a lower one for the rest of the page.

```
ESAHMI.ESAPRN.FontSize=16
ESAHMI.ESAPRN.WriteLN("List of recipes on EW")
ESAHMI.ESAPRN.WriteLN("")
ESAHMI.ESAPRN.WriteLN("")
ESAHMI.ESAPRN.FontSize=12
```

Now create the read cycle of the recipes saved on EW, using the GetFirstRecipeName and GetNextRecipeName methods. In the cycle use the PrintLN method to have the name of a recipe in each line.

```
R_Type="Type_Recipe_1"
a=ESAHMI.ESARECIPEMGR.GetFirstRecipeName(R_Type)Do While a<>""
ESAHMI.ESAPRN.WriteLN(a)
a=ESAHMI.ESARECIPEMGR.GetNextRecipeName(R_Type)
Loop
```

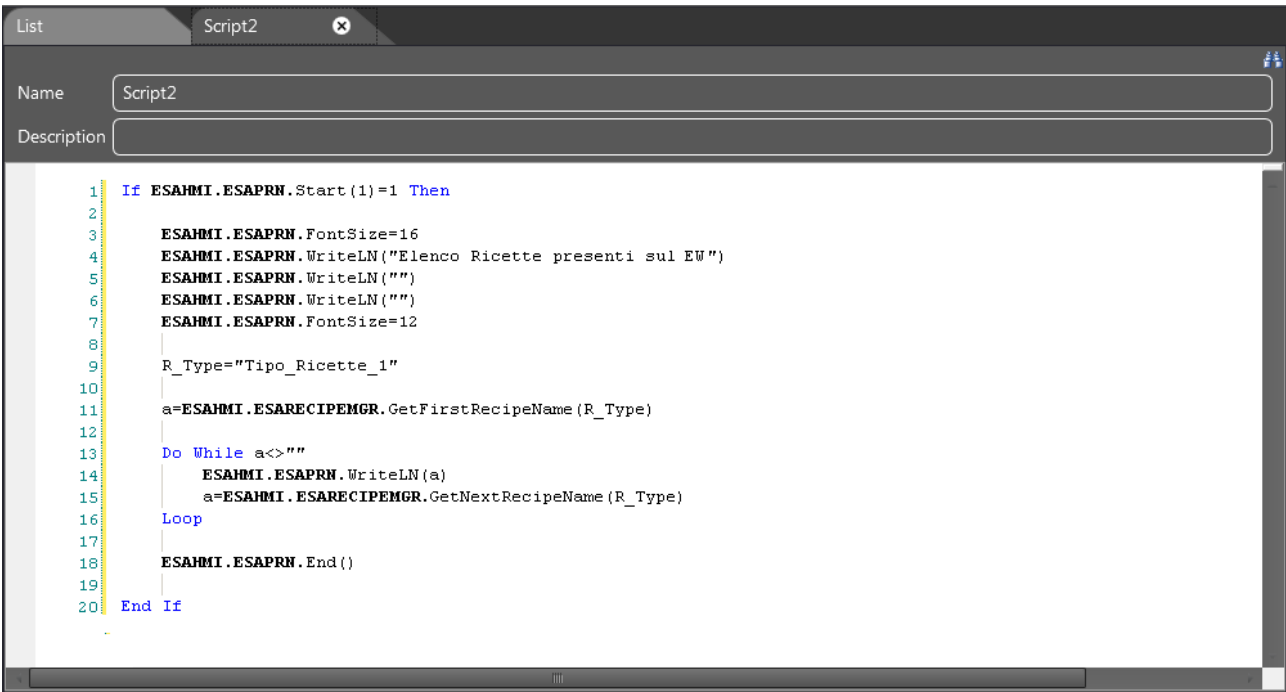
# CREW Manual

Up to this point we have prepared the contents of the page, but now it is necessary to launch the command that will make printing actually start:

```
ESAHMI.ESAPRN.End()
```

```
End If
```

With the execution of this method, the printing process starts. Below is the complete Script text.



```
List Script2 x
Name Script2
Description
1 If ESAHMI.ESAPRN.Start(1)=1 Then
2
3     ESAHMI.ESAPRN.FontSize=16
4     ESAHMI.ESAPRN.WriteLine("Elenco Ricette presenti sul EW")
5     ESAHMI.ESAPRN.WriteLine("")
6     ESAHMI.ESAPRN.WriteLine("")
7     ESAHMI.ESAPRN.FontSize=12
8
9     R_Type="Tipo_Ricette_1"
10
11     a=ESAHMI.ESARECIPEMGR.GetFirstRecipeName(R_Type)
12
13     Do While a<>""
14         ESAHMI.ESAPRN.WriteLine(a)
15         a=ESAHMI.ESARECIPEMGR.GetNextRecipeName(R_Type)
16     Loop
17
18     ESAHMI.ESAPRN.End()
19
20 End If
```

# CREW Manual

---

## Everyware

### What is Everyware

Everyware is a software that makes it possible, through access to a personal domain, to control, interact or edit the programmed systems as though the user were in front of them.

In fact, thanks to this remote assistance platform the distance between operator and applications is eliminated and it is thereby possible to drastically cut maintenance costs and times.

Everyware uses a normal internet connection through common private IPs to connect the user directly to the systems. In this way it makes it possible to run, edit and process the applications in a synchronous or asynchronous manner wherever they are.

Everyware does not interfere with firewalls and does not require a public IP for its connection.



# CREW Manual

## Main characteristics of Everyware

The Everyware platform is very quick and immediate. The connection is managed automatically through the geo-localisation system that guarantees the best available data traffic line, and its reliability is guaranteed by a redundant server system that ensures continuity of the service.

Everyware starts an encrypted VPN connection with the selected system.

Data protection and connection security are guaranteed by the use of a digital certificate with TLS 1.2 connection, that verifies the identity according to a client-server logic.

## What can be done with Everyware

- Synchronous remote control.
- Asynchronous remote control.
- Remote programming and debugging of the HMI application.
- Remote programming and debugging of devices such as PLCs and Drives.
- Preventive maintenance through malfunction analysis.
- Chat Service file transfer.
- Remote management of system processes.

Everyware operates on a simple principle: when a profile is created, the user has a personal account and domain, where he/she can view his/her devices from.

It is also possible to decide whether to offer the operator and other users single or multiple access to domains. This makes it possible to view and manage data simultaneously or one datum at a time, to share it with other users or manage it individually, and to punctually interact with the system or with the individual machine, no matter where they are located.

Having an Everyware domain also means that two or more branches of the same company can operate simultaneously on the same project.

# CREW Manual

## Types of licences

Single access/Single domain: one user can connect to the domain at a time, to view and manage one application at a time. There are however many applications as have been created.

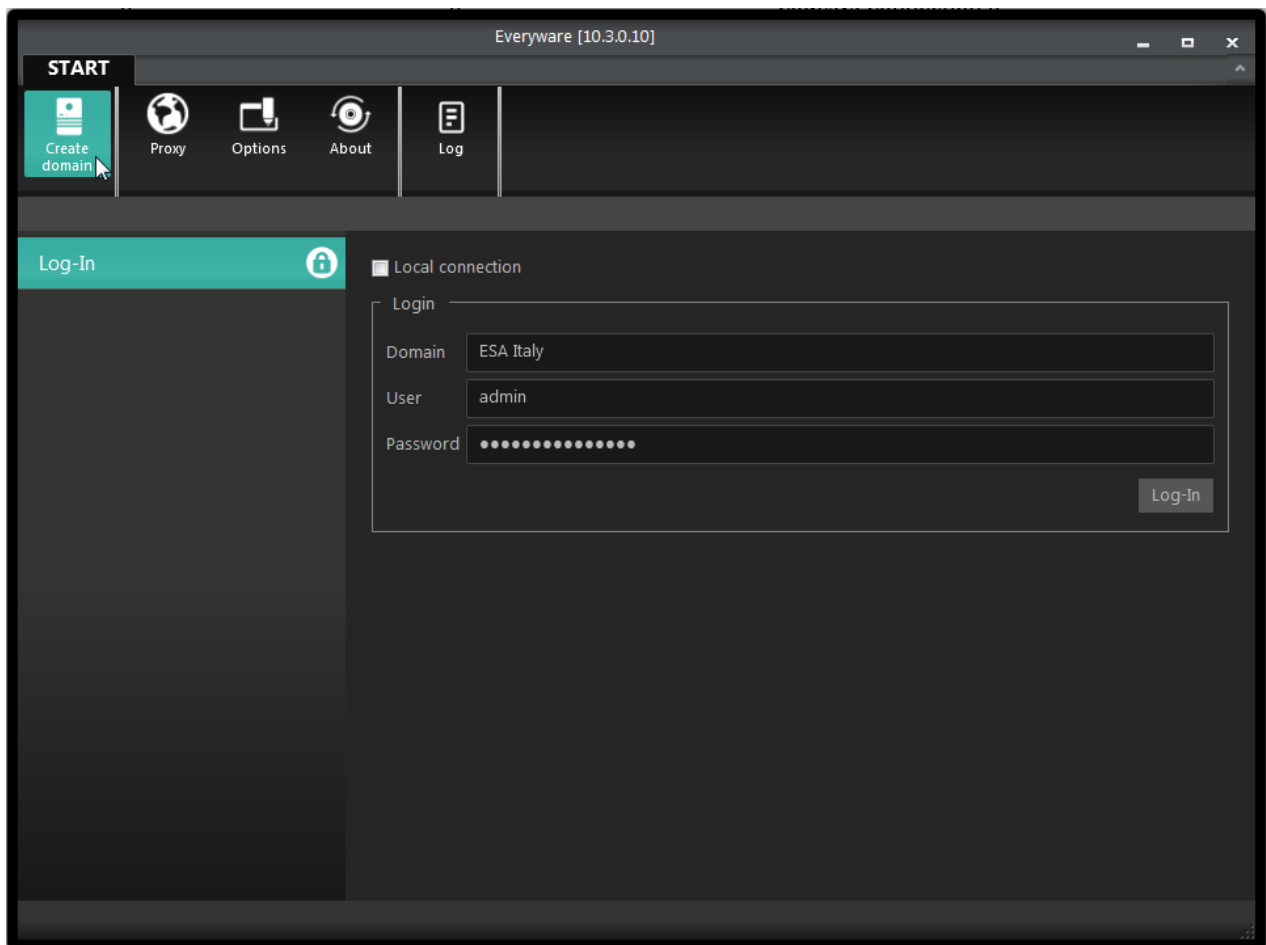
Multi access/Single domain: multiple users have access to the domain and can work simultaneously. They can see various devices and manage one or more applications at a time.

Multi access/Multi domain: full freedom in organising applications. It is possible to create various domains, connect to however many devices you need to and manage them in hierarchical account structures. Plus, it is possible to manage the various applications and user accesses that can connect to the multiple domains at the same time.

# CREW Manual

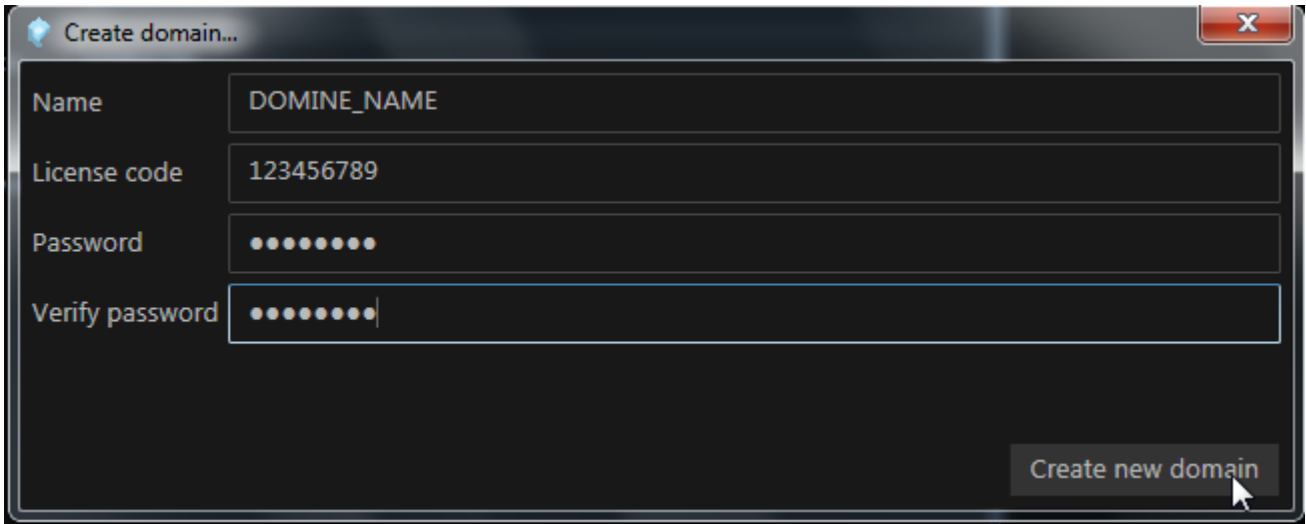
## Creating domains

The first operation that needs to be done after installing Everyware is the creation of the domain. The domain is created by the administrator who can add or eliminate panels, and create, eliminate or edit users.



# CREW Manual

Enter domain name, licence number and password, then click “Create new domain”.

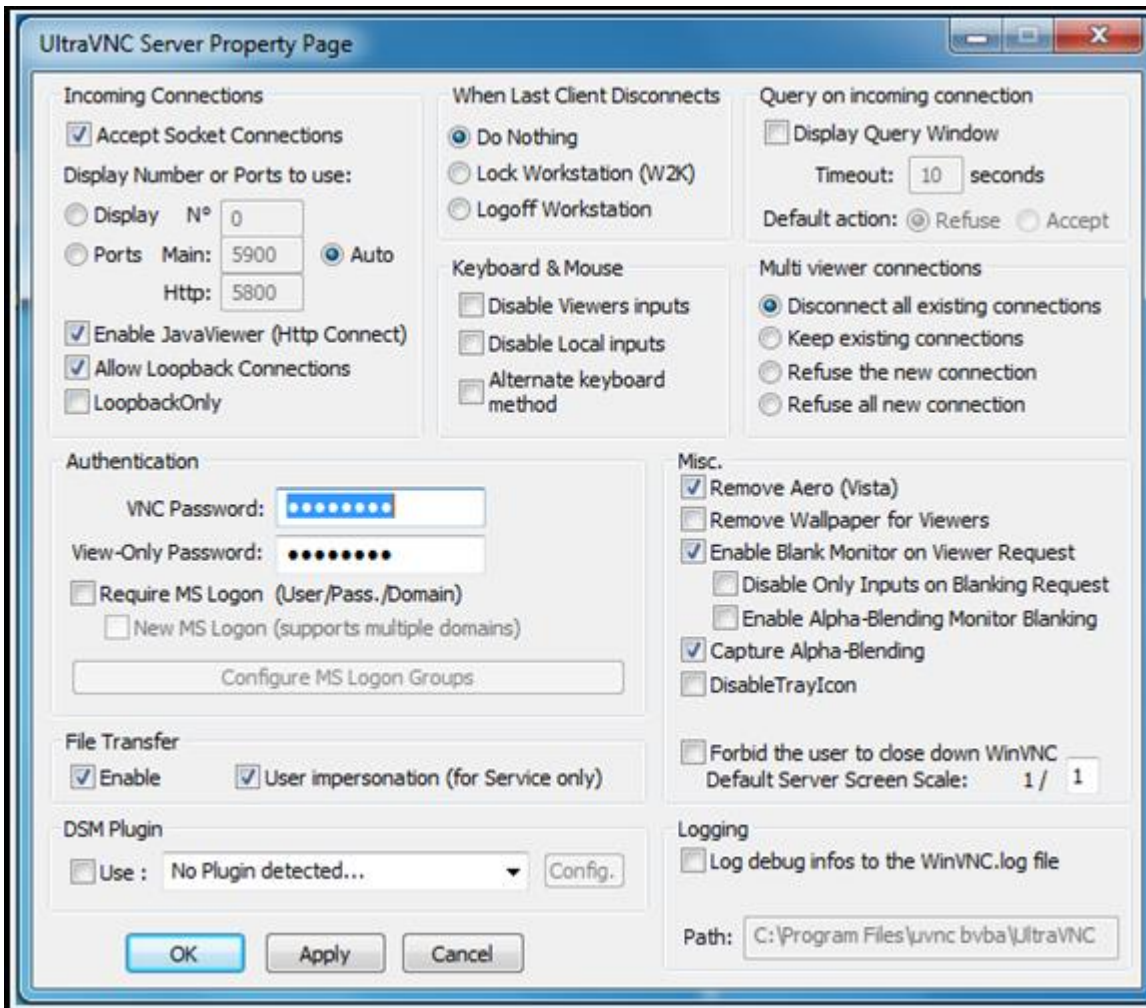


**IMPORTANT:** when the domain is created it is possible to change the password but not the name. The password entered in this phase is the one required to connect to one's domain.

**IMPORTANT:** for Everyware installation on PC (EW2xx), it is necessary to install a VNC Server software (it can be downloaded for free from the internet). ESA Automation recommends “ULTRAVNC SERVER”.

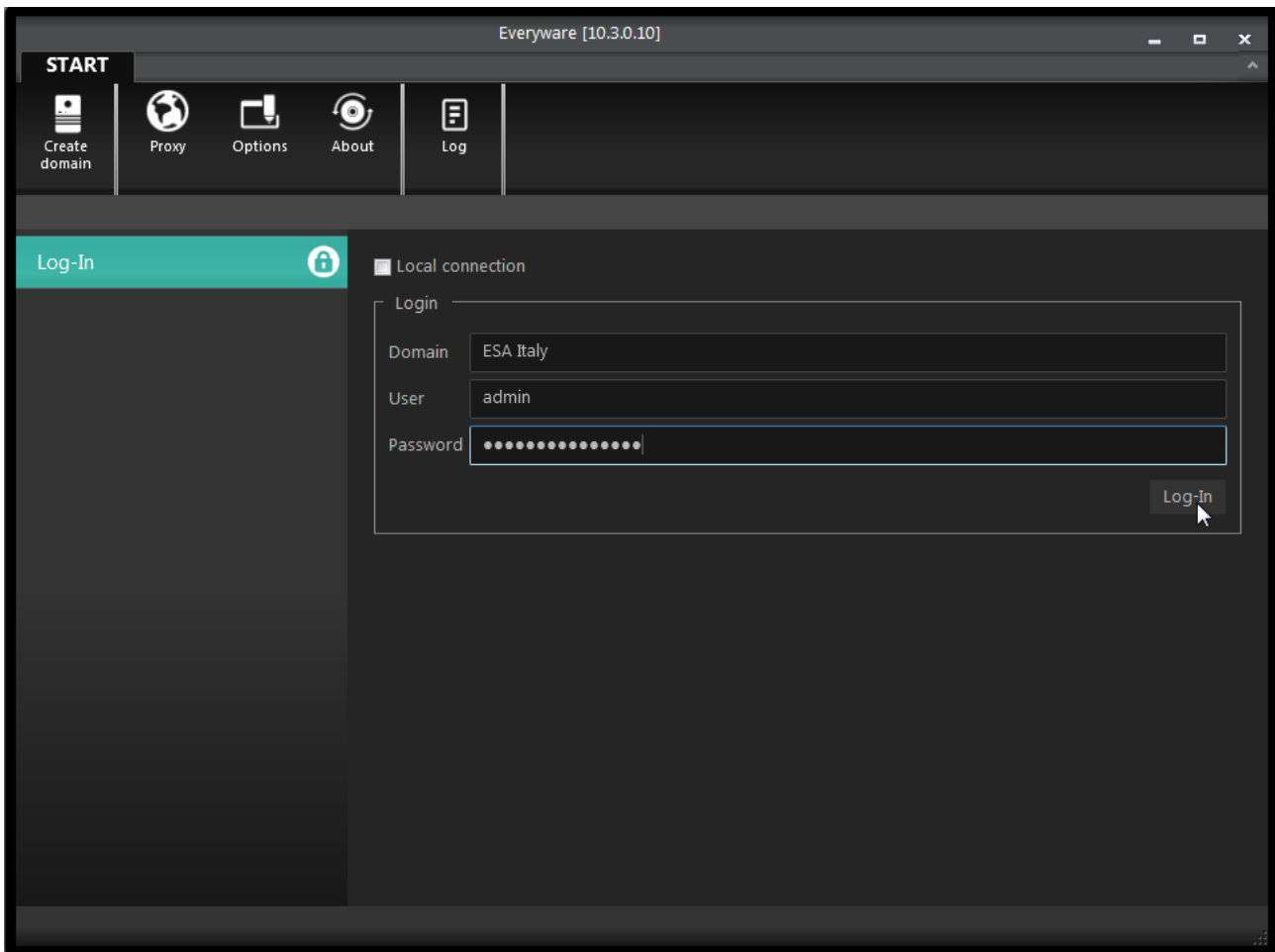
Note: In the VNC SERVER software settings you need to enter the same password that will be used later on in “REMOTE DESKTOP”.

# CREW Manual



# CREW Manual

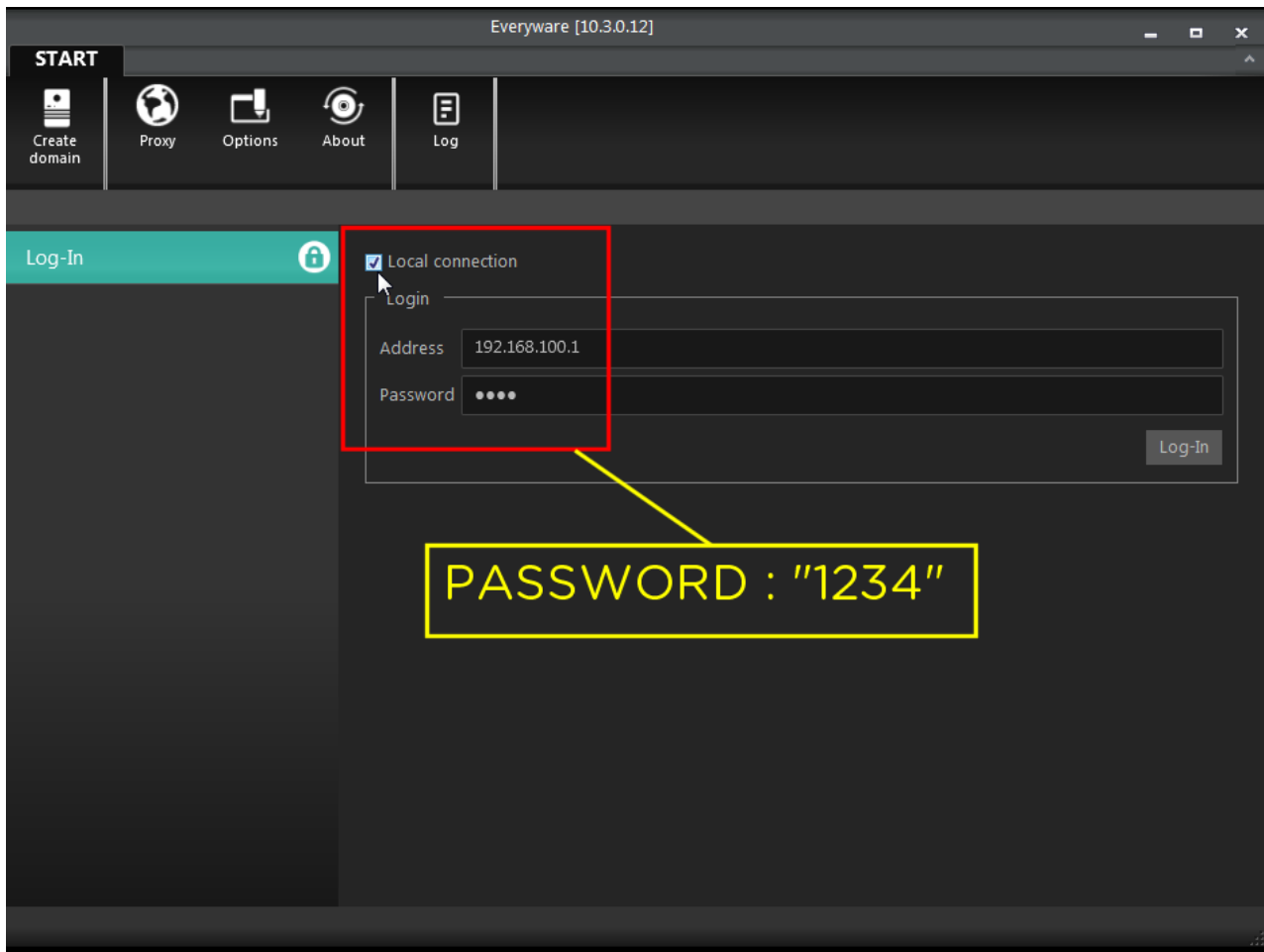
Log in.



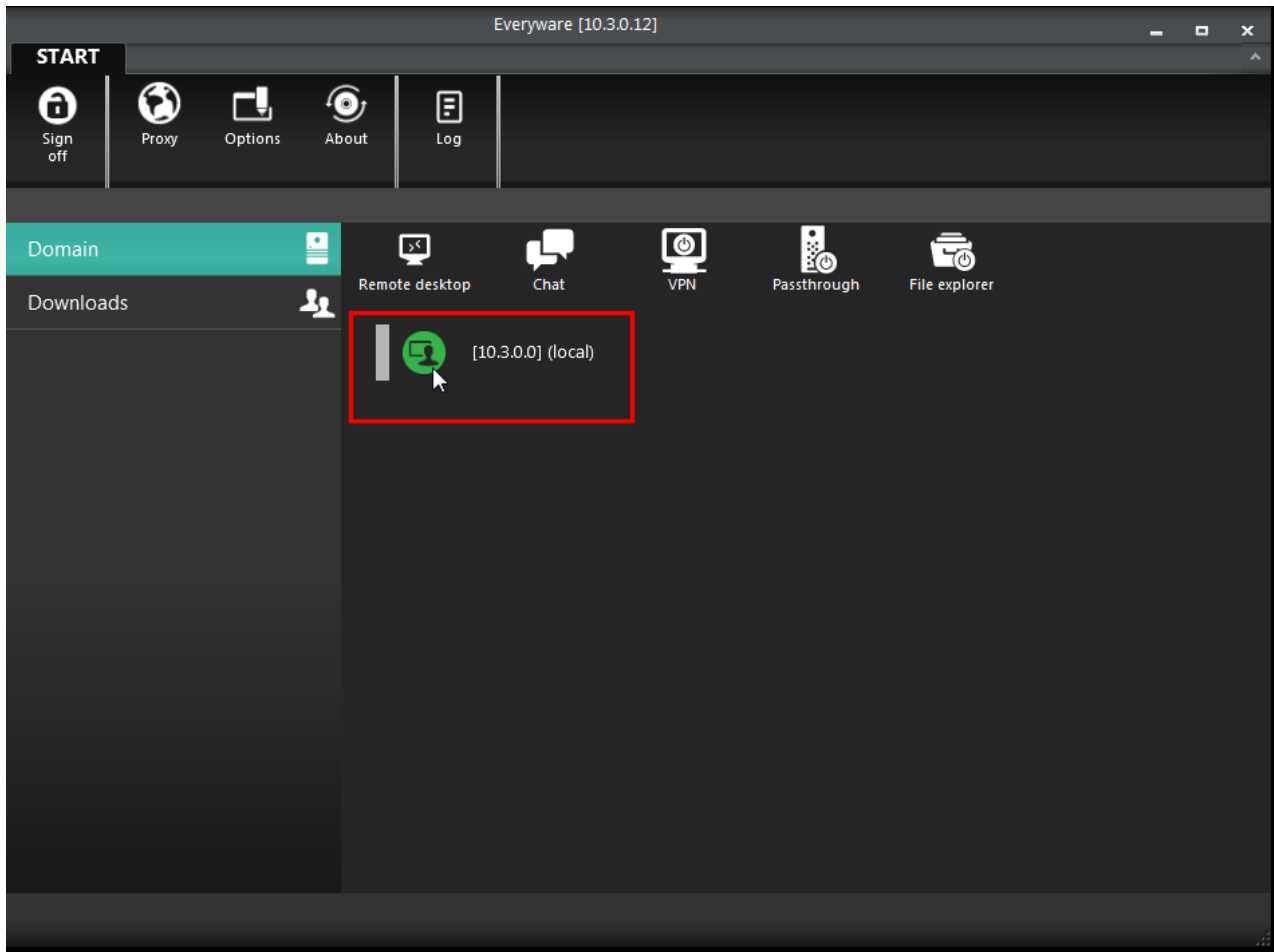
# CREW Manual

## Local Connection

The “Local” networked connection is set up with LAN intranet (it does not use the internet connection).



# CREW Manual

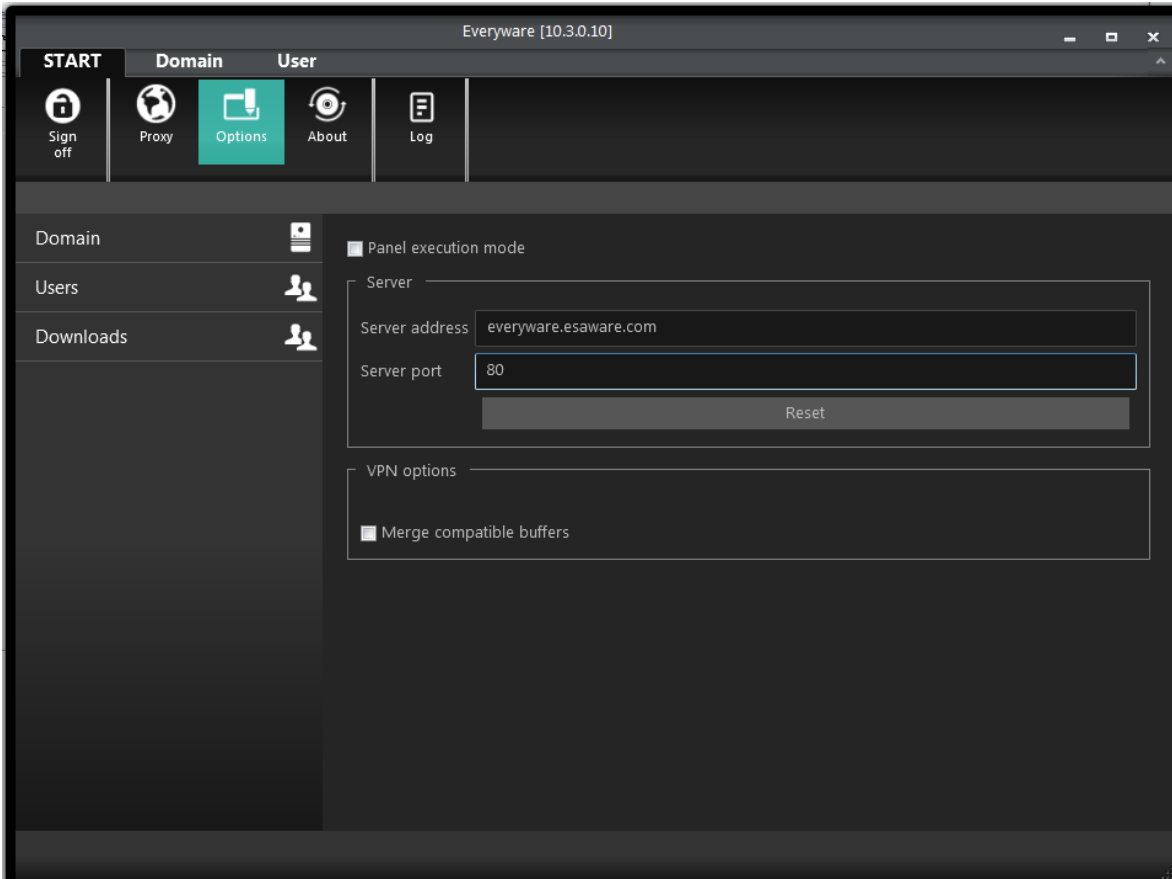




# CREW Manual

## Proxy Settings

Select the Proxy setting options then enter: Proxy address, port number, name of company network domain, company user name and company user password.

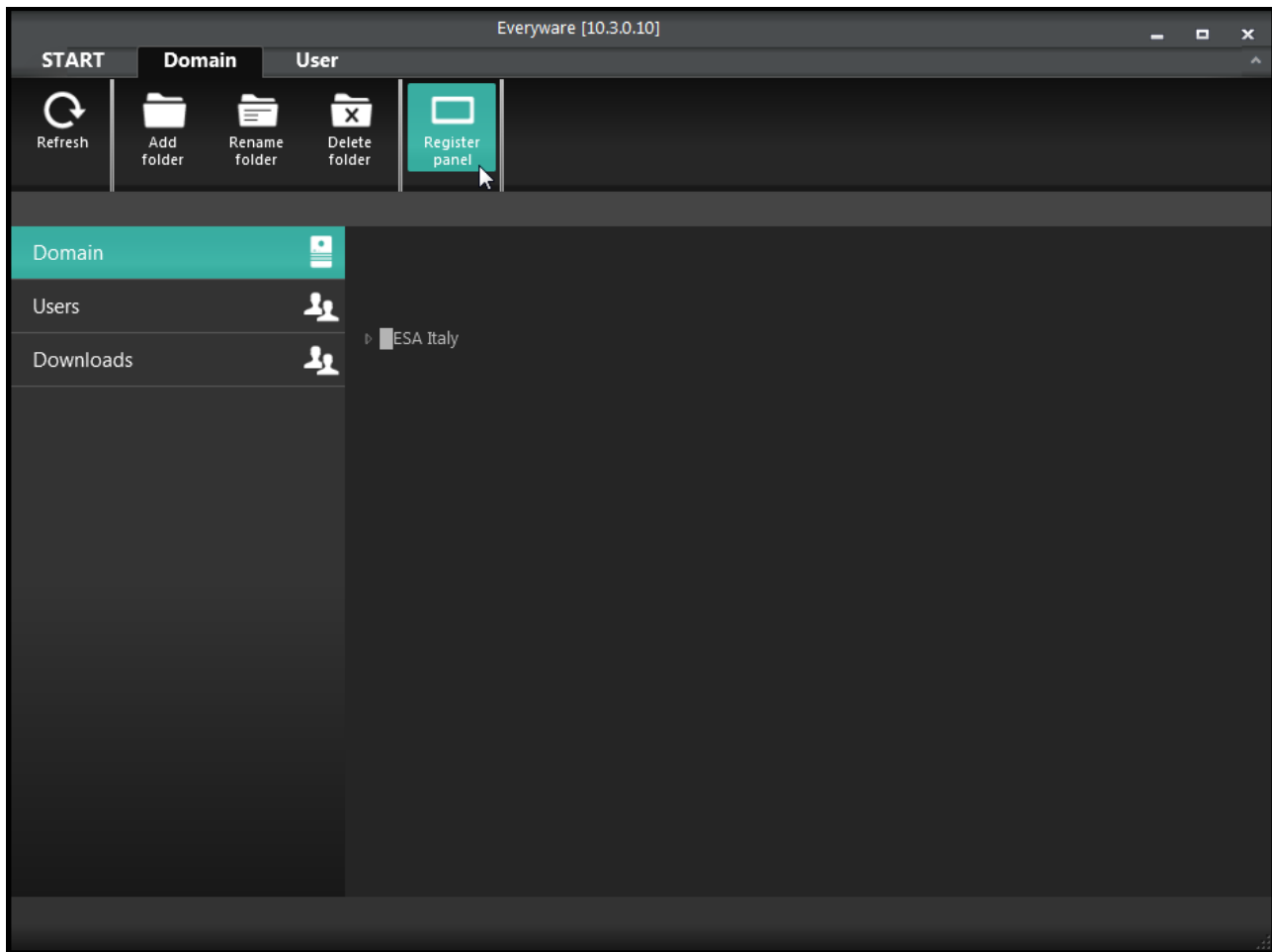


Note: The “Panel execution mode” checkbox makes it possible to use the PC on the network (EW2xx) like any other terminal (EW1xx).

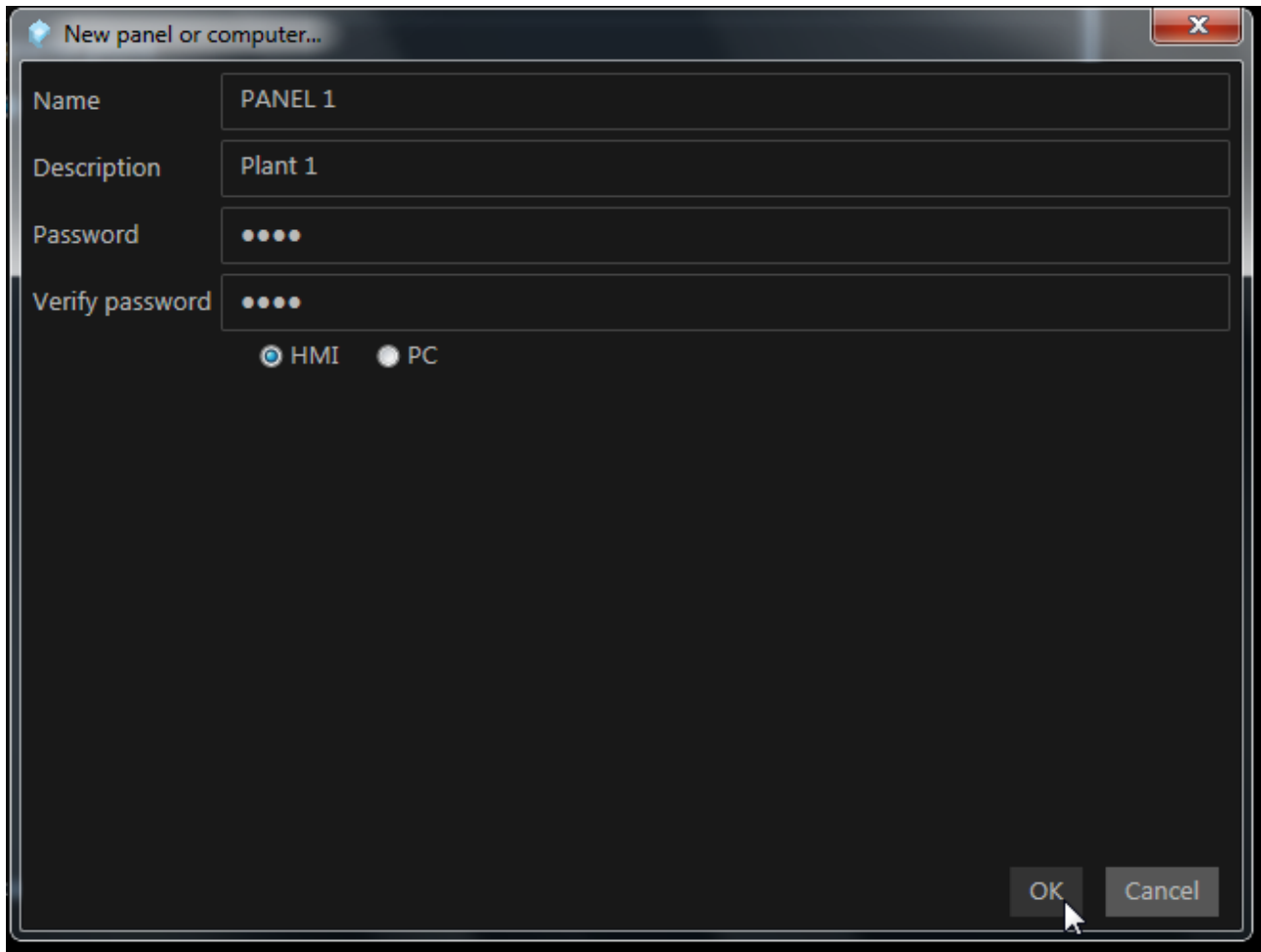
# CREW Manual

## Adding a panel

Click the relative icon to enter one or more terminals in the network of your domain.



# CREW Manual



New panel or computer...

Name: PANEL 1

Description: Plant 1

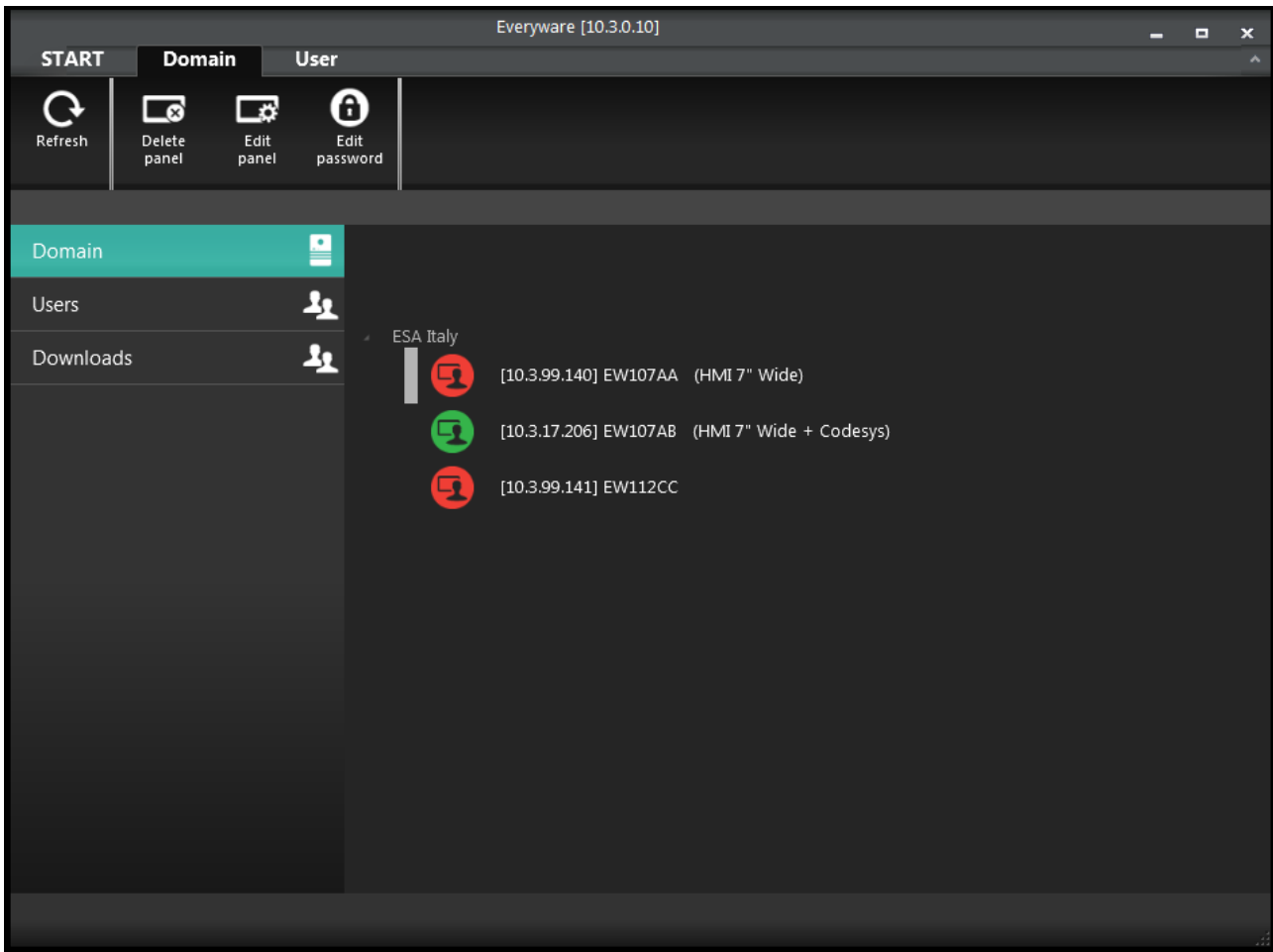
Password: ●●●●

Verify password: ●●●●

HMI  PC

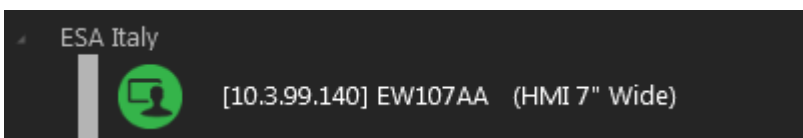
OK Cancel

# CREW Manual



Note: As shown in the previous image, the red box to the left of the panel name tells you that the connection with said panel has not been set up yet. It turns green when the settings on the panel side have been entered.

Enter the terminal settings with reference to the “Panel side settings”, “Everyware Configuration” and “Network Settings” points. At the end, the following image needs to appear on the PC where Everyware was installed (connection with terminal established, the box has turned green):

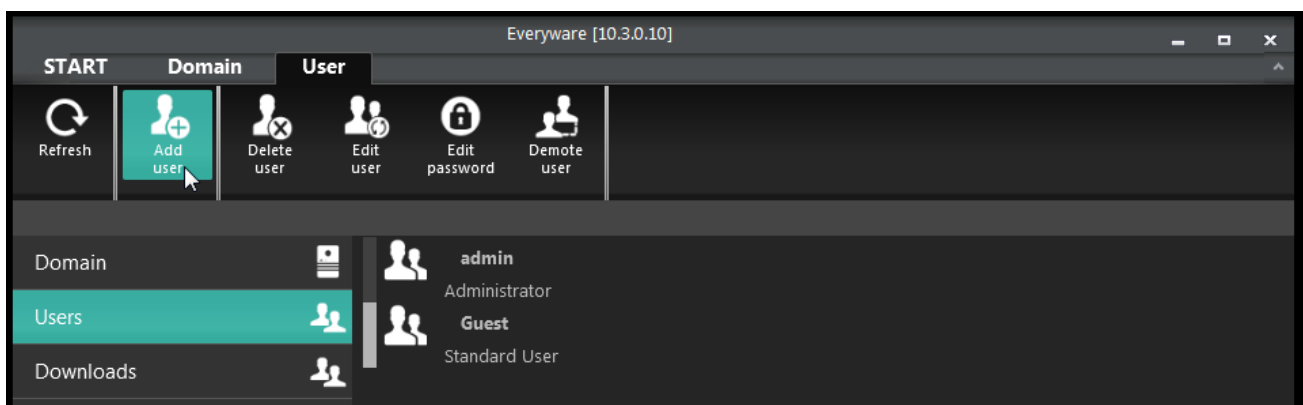


# CREW Manual

## Adding a user

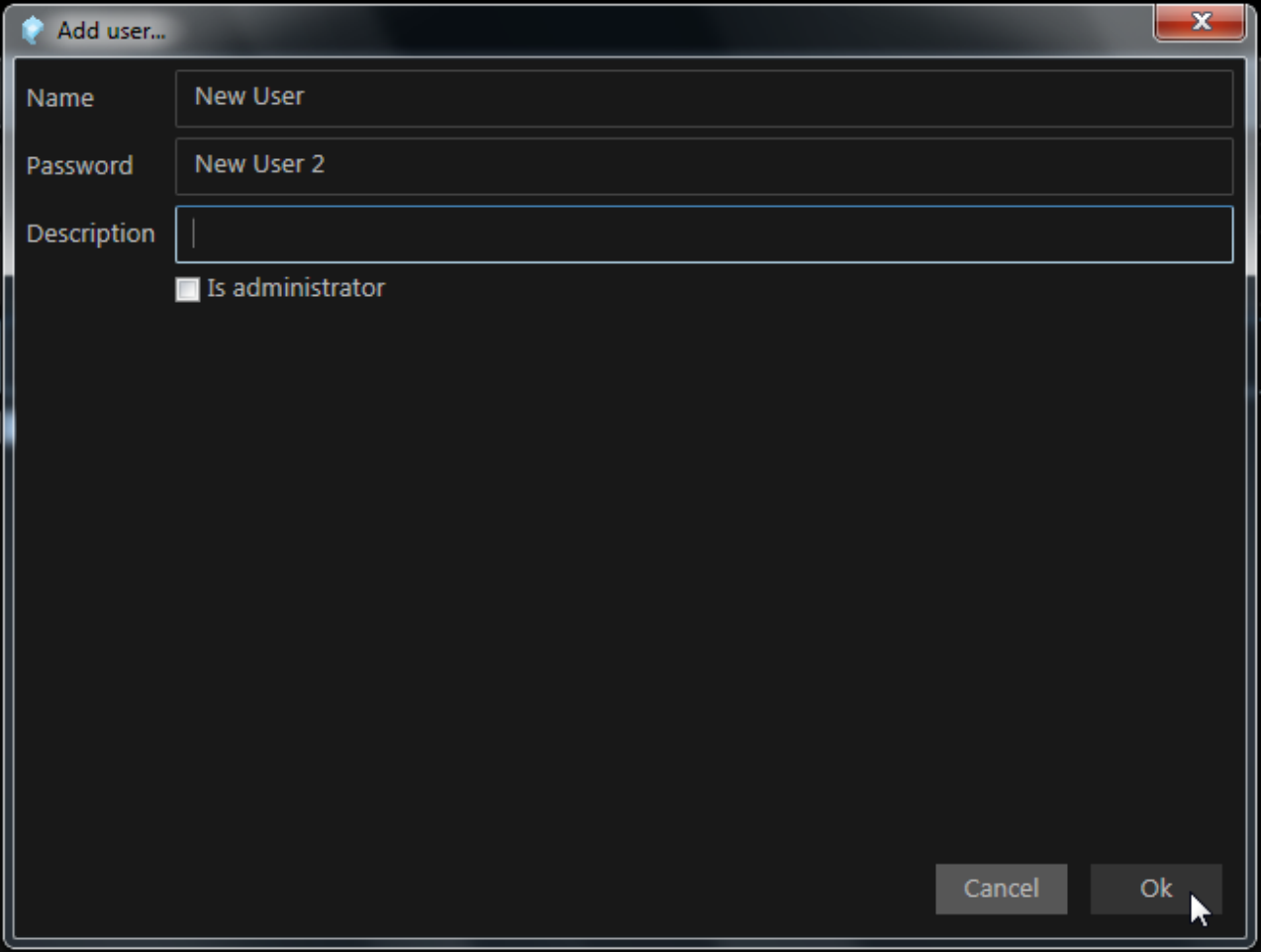
Note: It is only possible to add users to the domain if you are logged on as administrator.

Click the relative icon to add a new user to the domain.



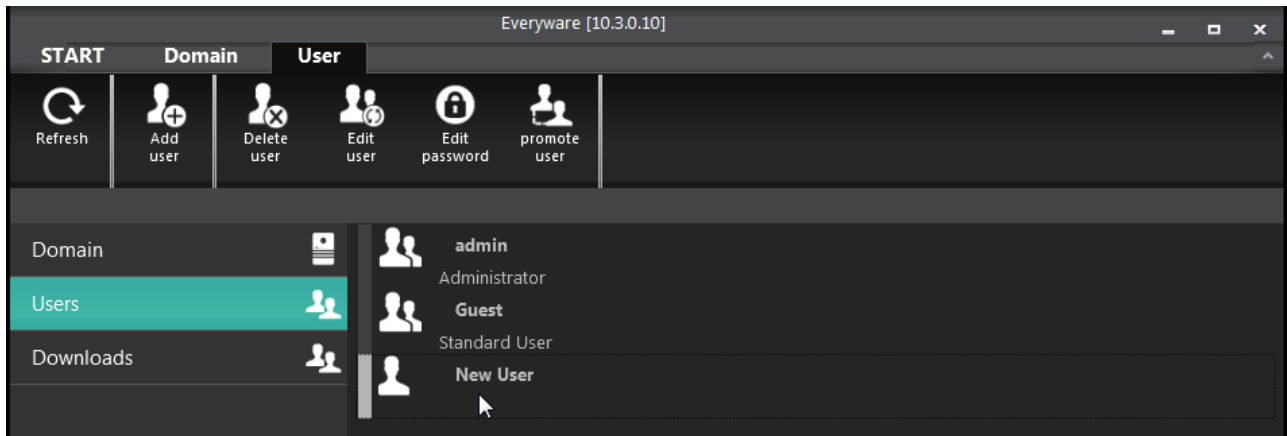
# CREW Manual

Enter name, password and any description of the new user. Enable the relative “checkbox” to assign administrator credentials to the newly created user.



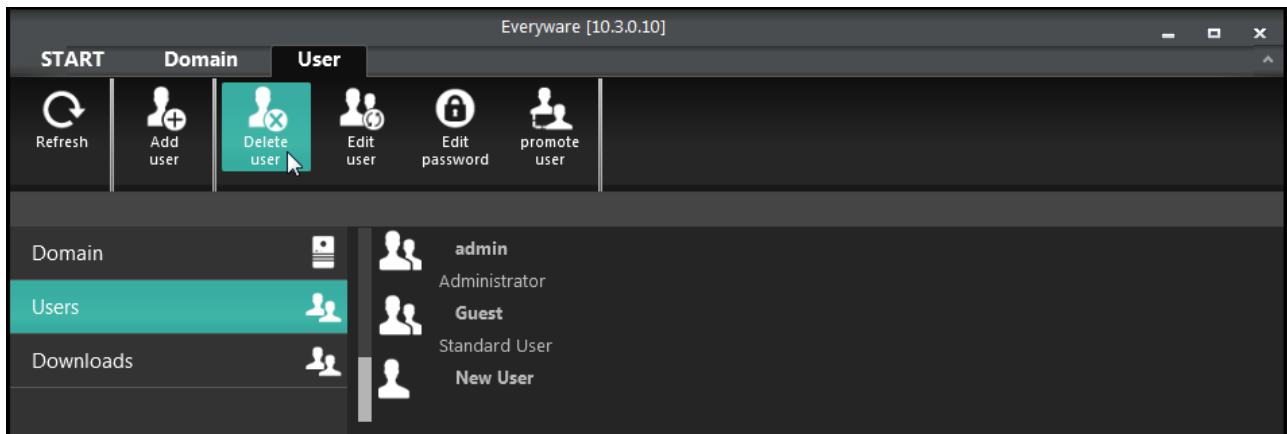
The screenshot shows a dialog box titled "Add user...". It contains three text input fields: "Name" with the value "New User", "Password" with the value "New User 2", and "Description" which is currently empty. Below the "Description" field is a checkbox labeled "Is administrator" which is currently unchecked. At the bottom right of the dialog box are two buttons: "Cancel" and "Ok". A mouse cursor is pointing at the "Ok" button.

# CREW Manual



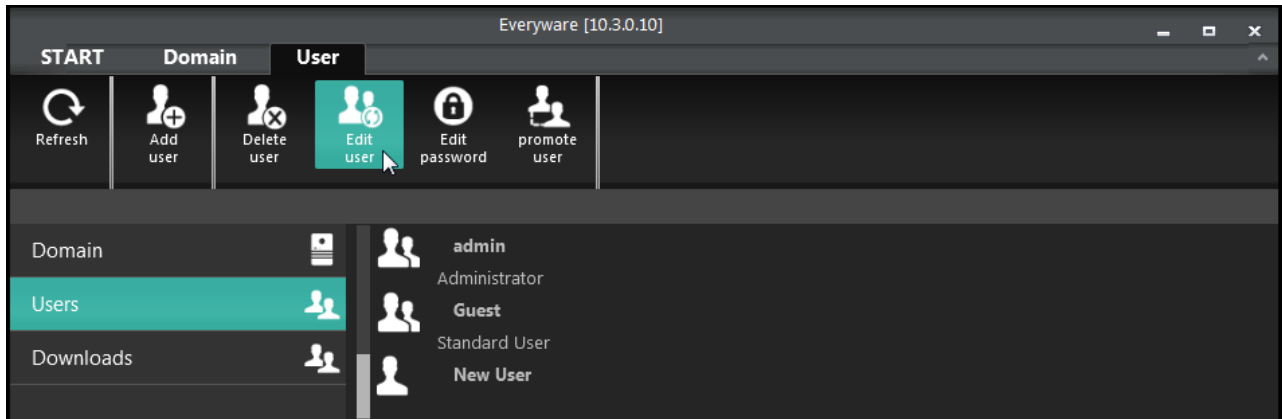
Use the relative keys to:

- eliminate users.

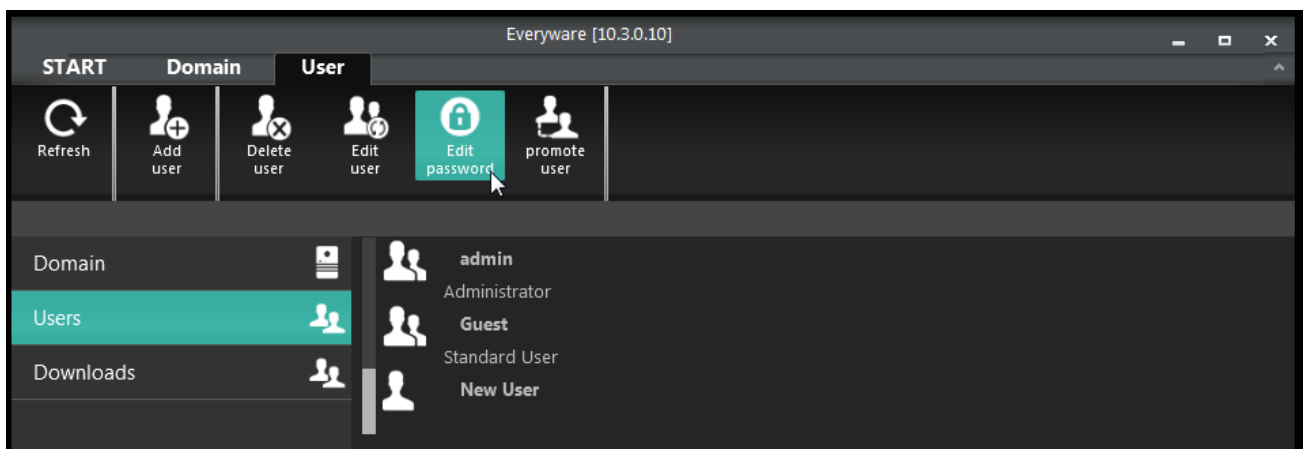


# CREW Manual

- edit users (name, description and possibility of making him/her administrator).



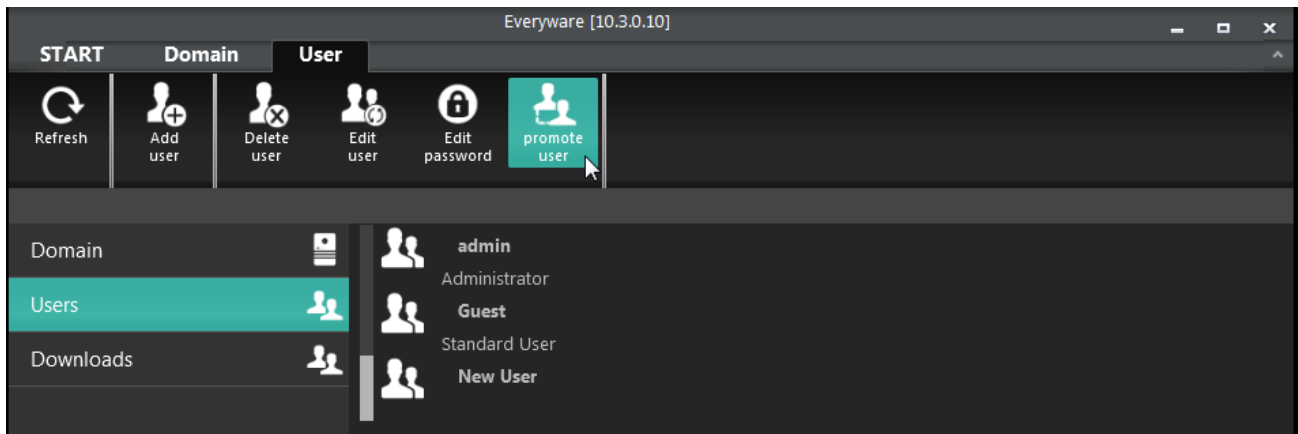
- change user passwords.





# CREW Manual

- promote a user to administrator (a user with administrator credentials can, in turn, create new users in the domain).



## Features of Everyware

When the connection is established correctly, the following icons appear on the main Everyware bar.



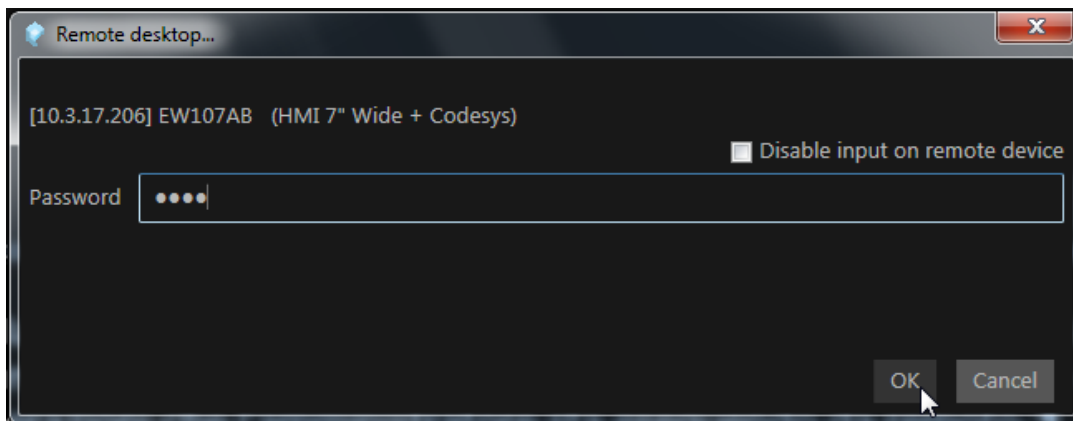
# CREW Manual

## Remote desktop

Click the relative icon to start the Remote desktop.



In the box that appears, enter the password required to start the remote Desktop (which was entered in “ID Settings”, Remote Access Password) on the panel side, then confirm with “Ok”.



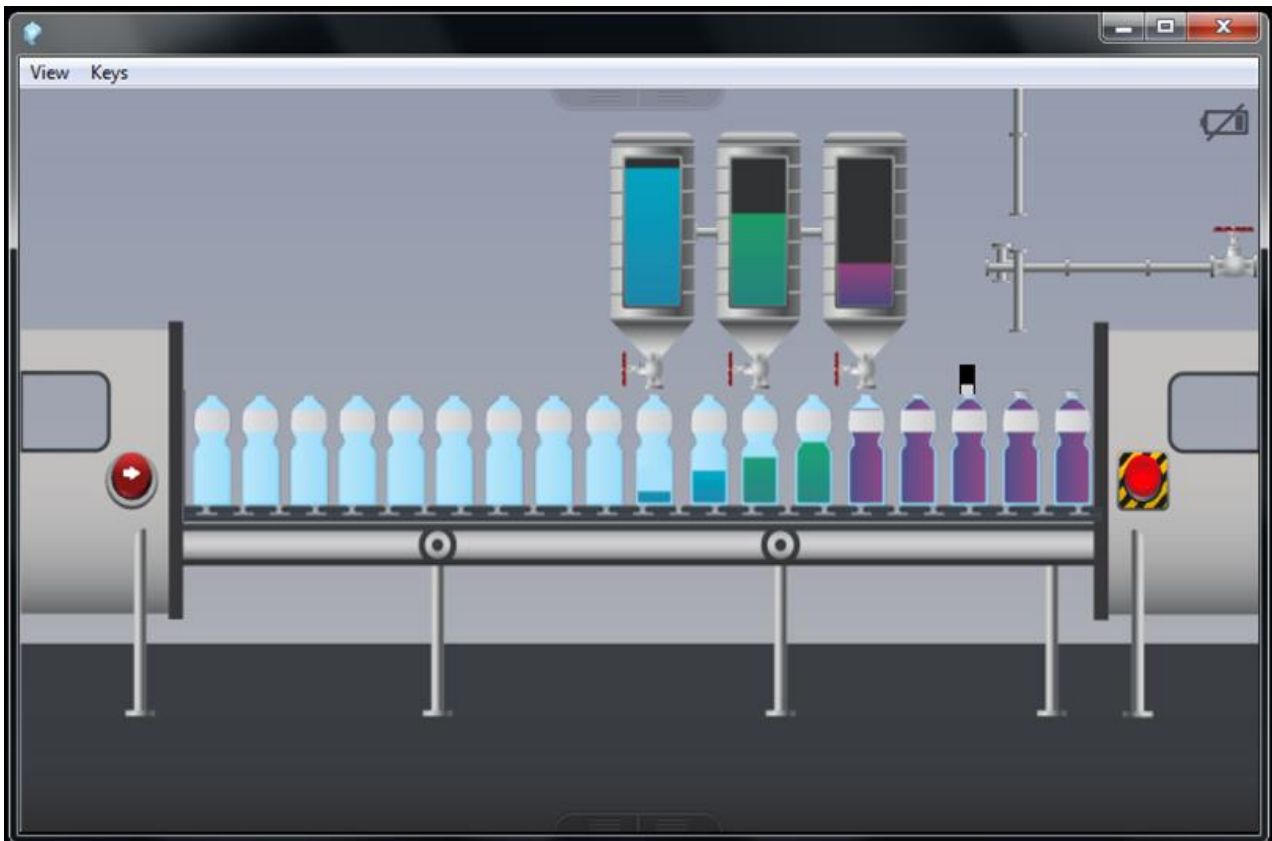
## CREW Manual

The “Blank remote screen” option is used to show an image on the terminal being checked that stops the operator from seeing the operations that the administrator is performing remotely.



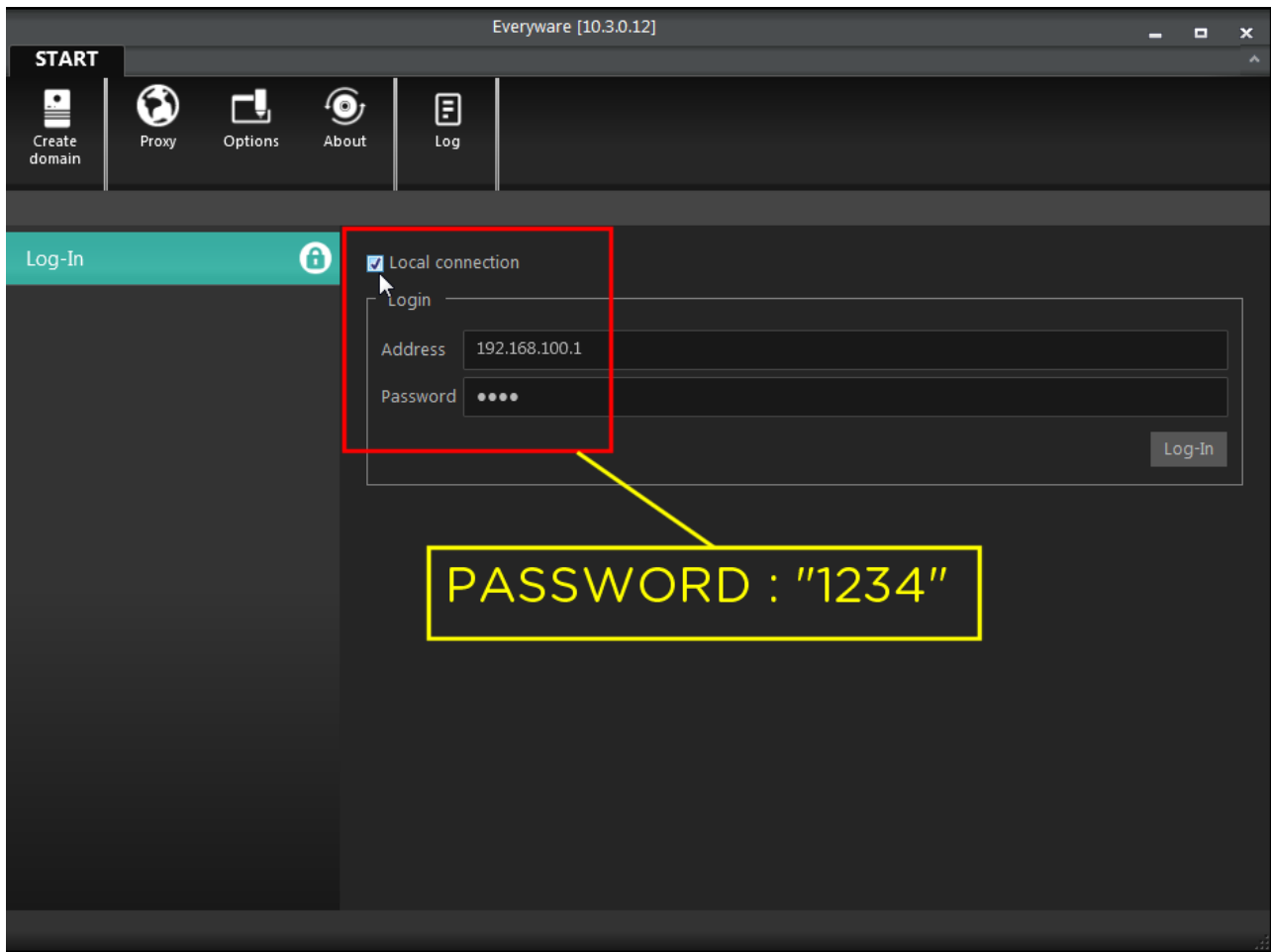
## CREW Manual

When the operator only sees the previous image, on the PC where Everywhere is installed, the administrator has access to the terminal as though he/she were physically in front of it and is free to perform all necessary operations.

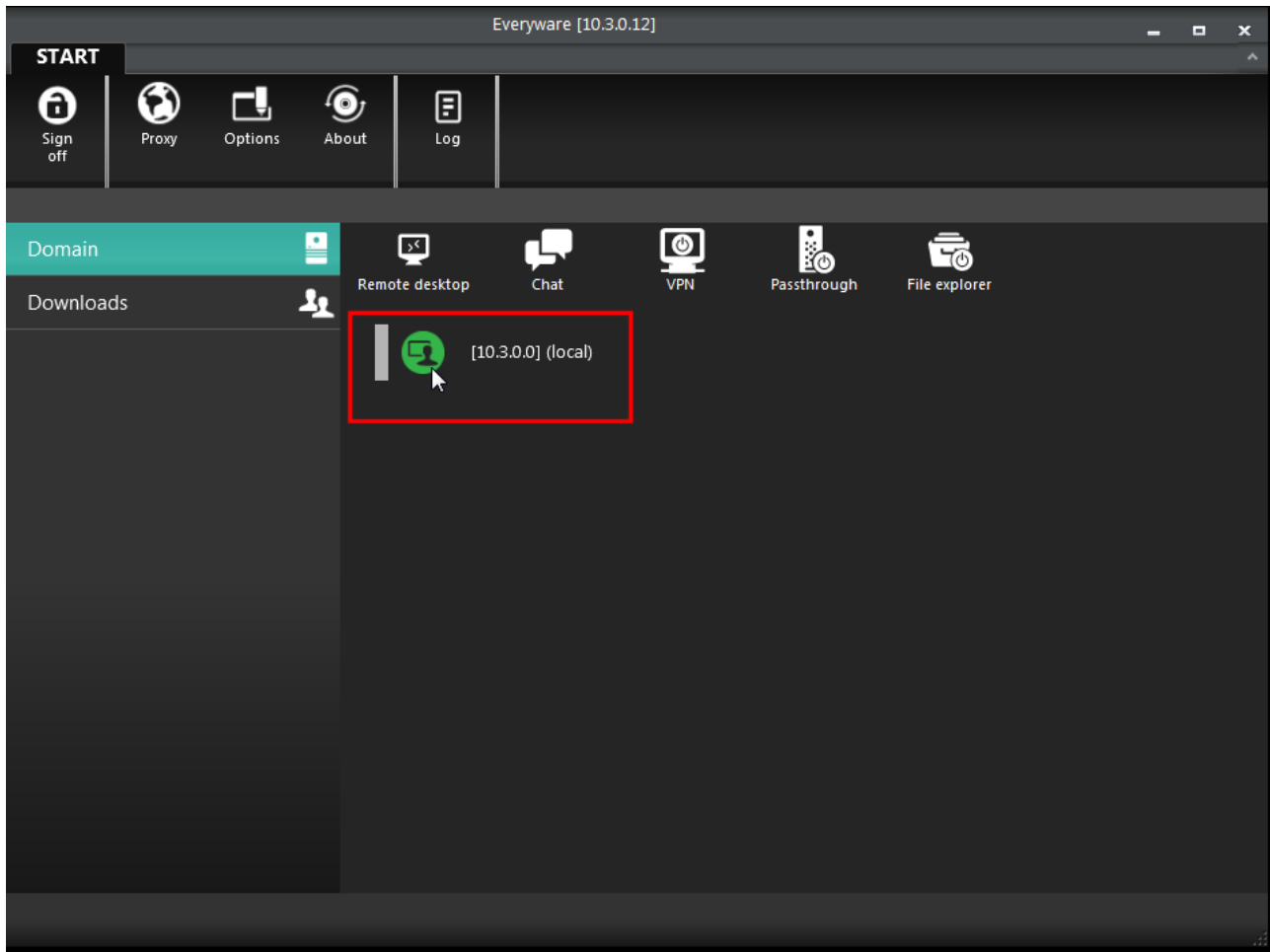


# CREW Manual

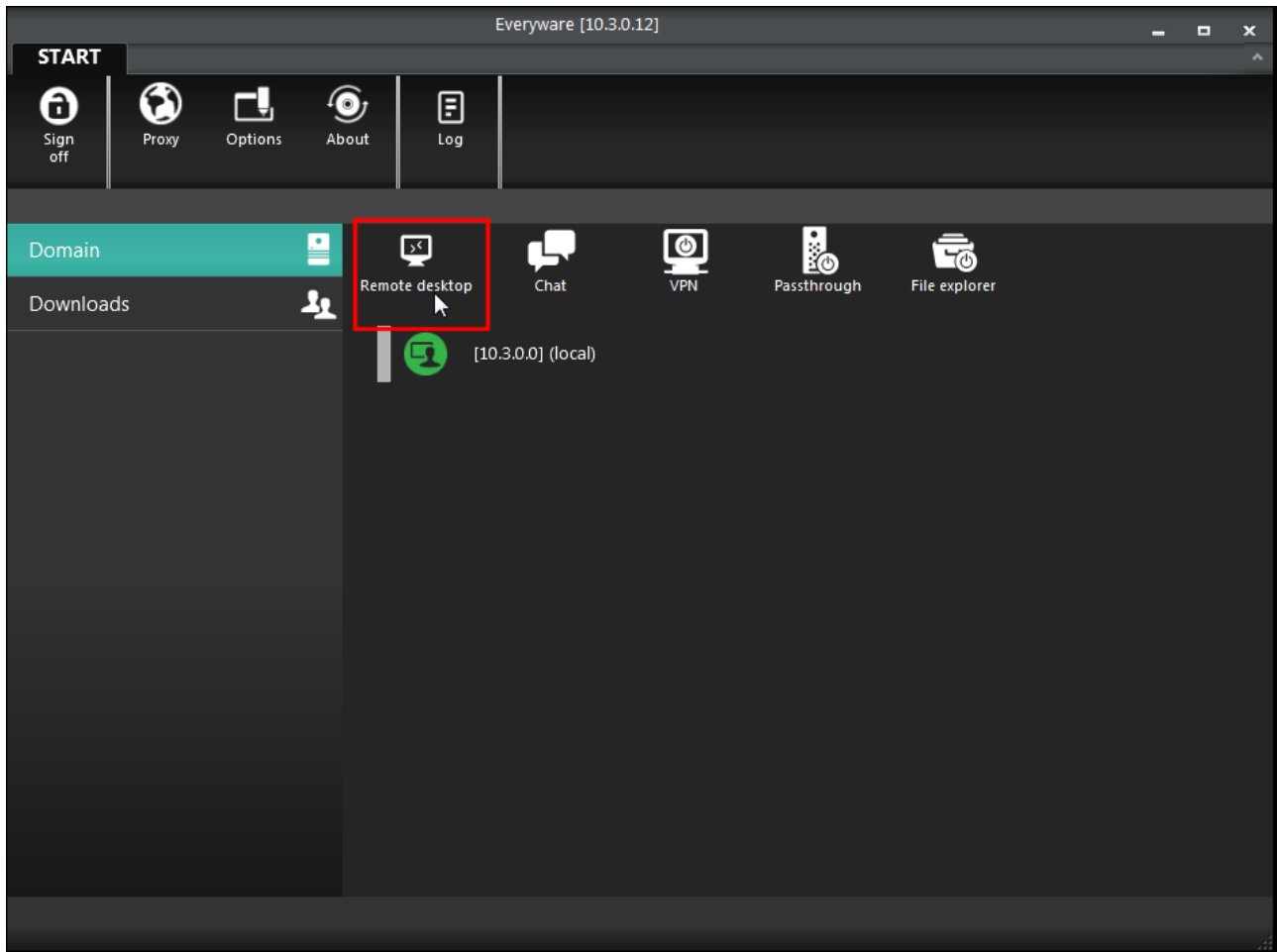
## Remote desktop (Local connection)



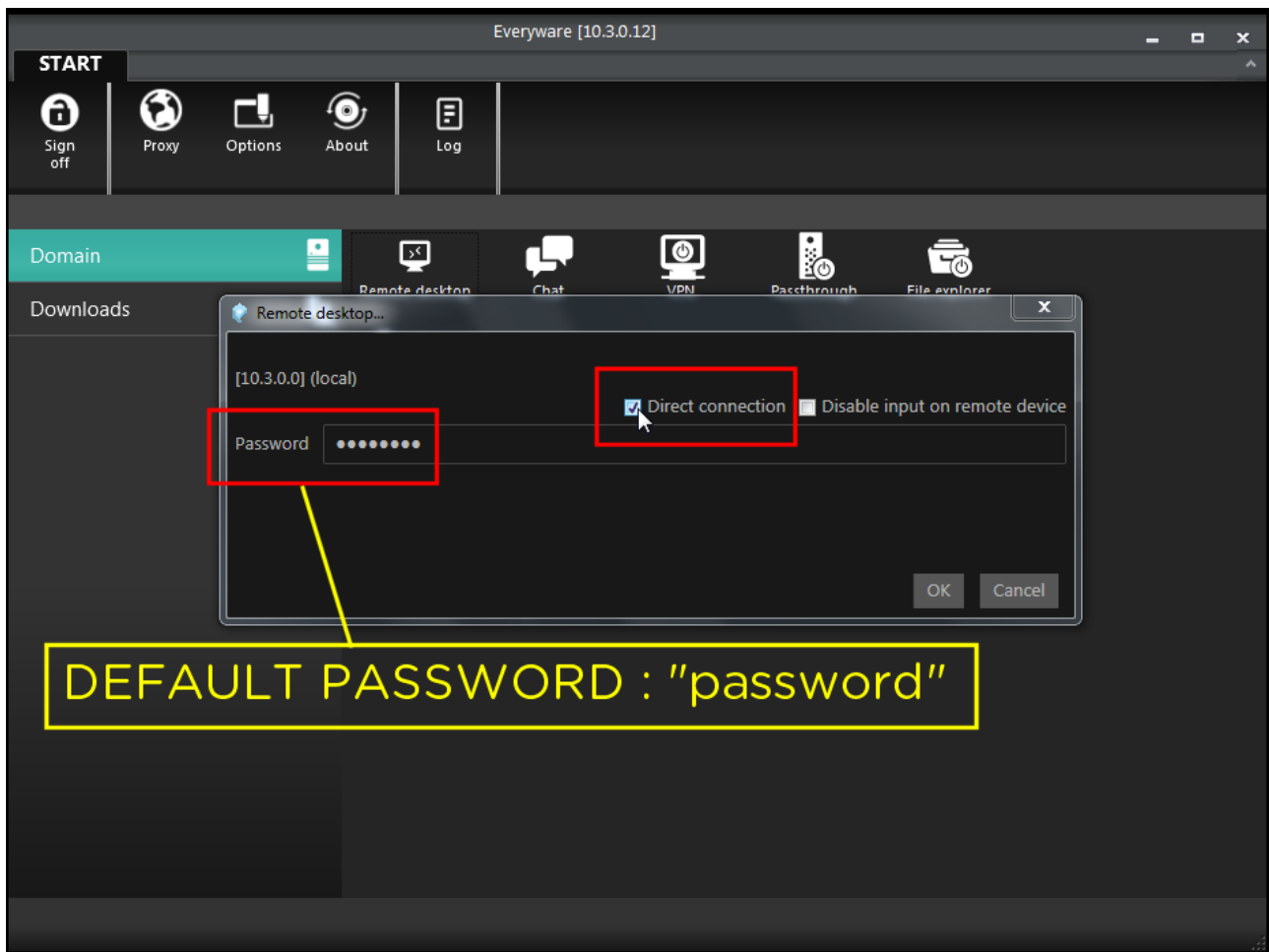
# CREW Manual



# CREW Manual

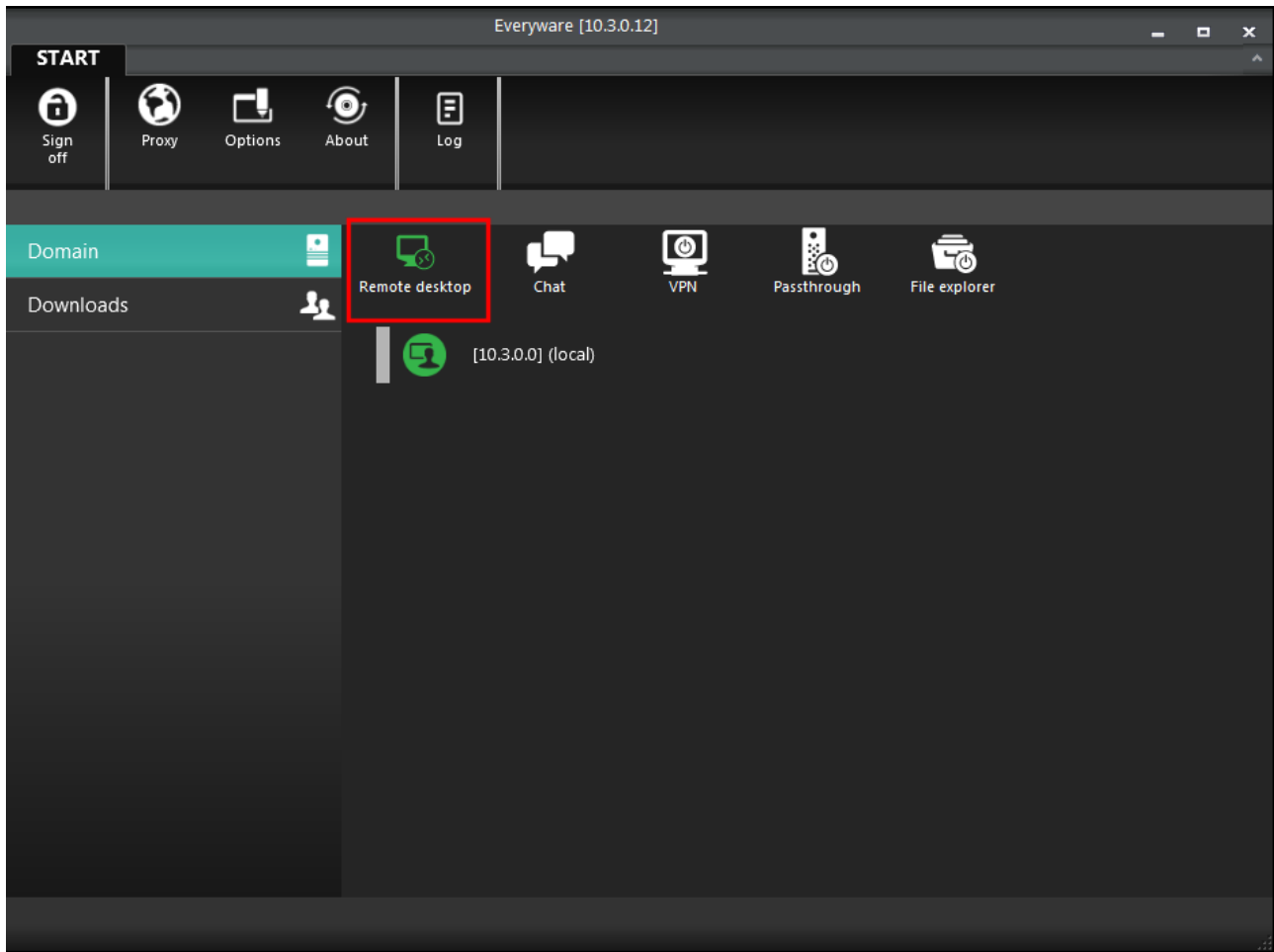


# CREW Manual

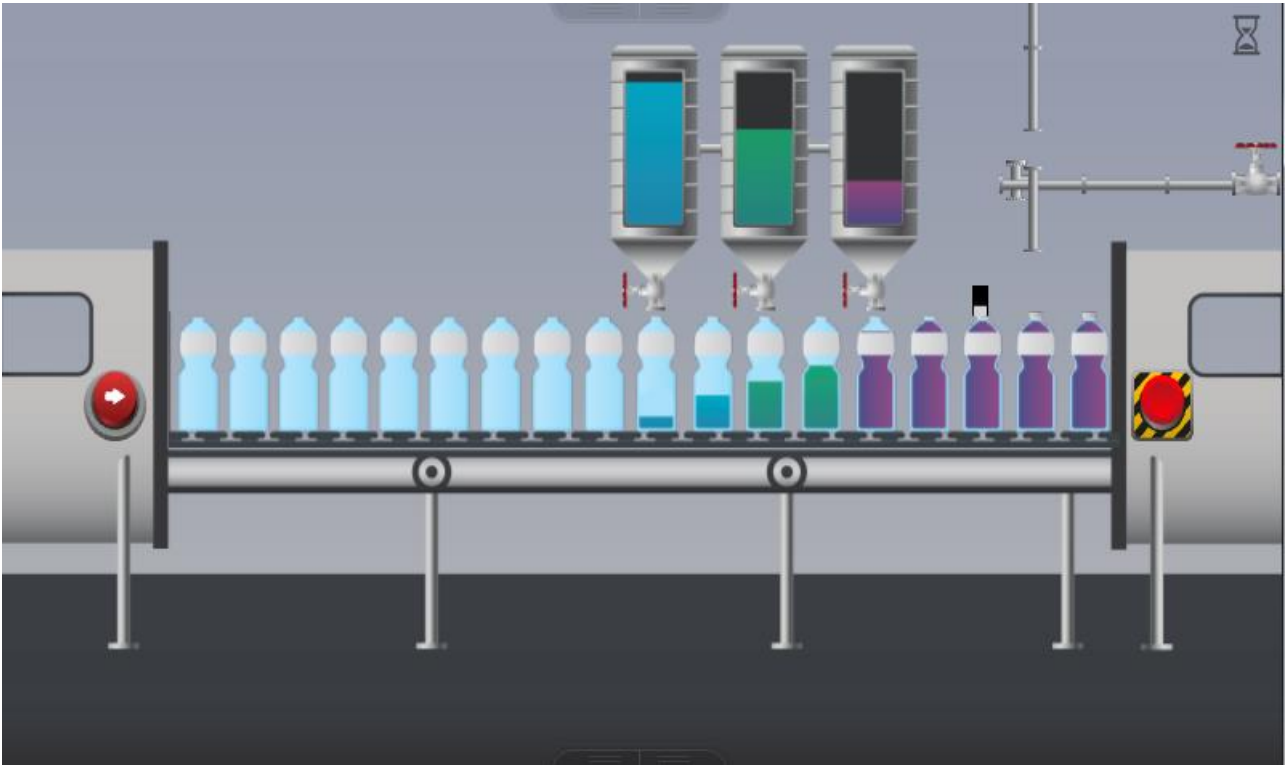




# CREW Manual



# CREW Manual



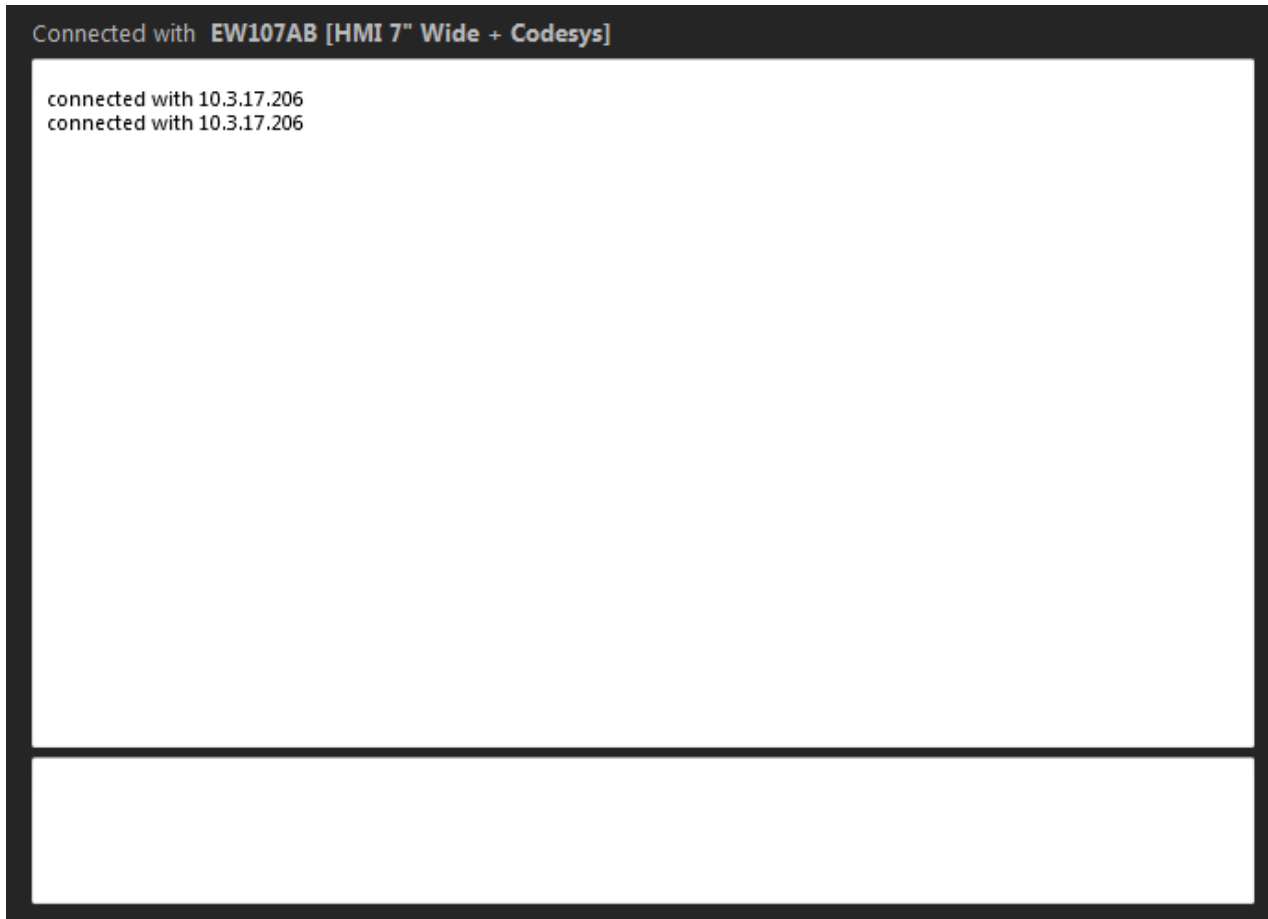
## Chat service

Click the relative icon to start the Chat service.



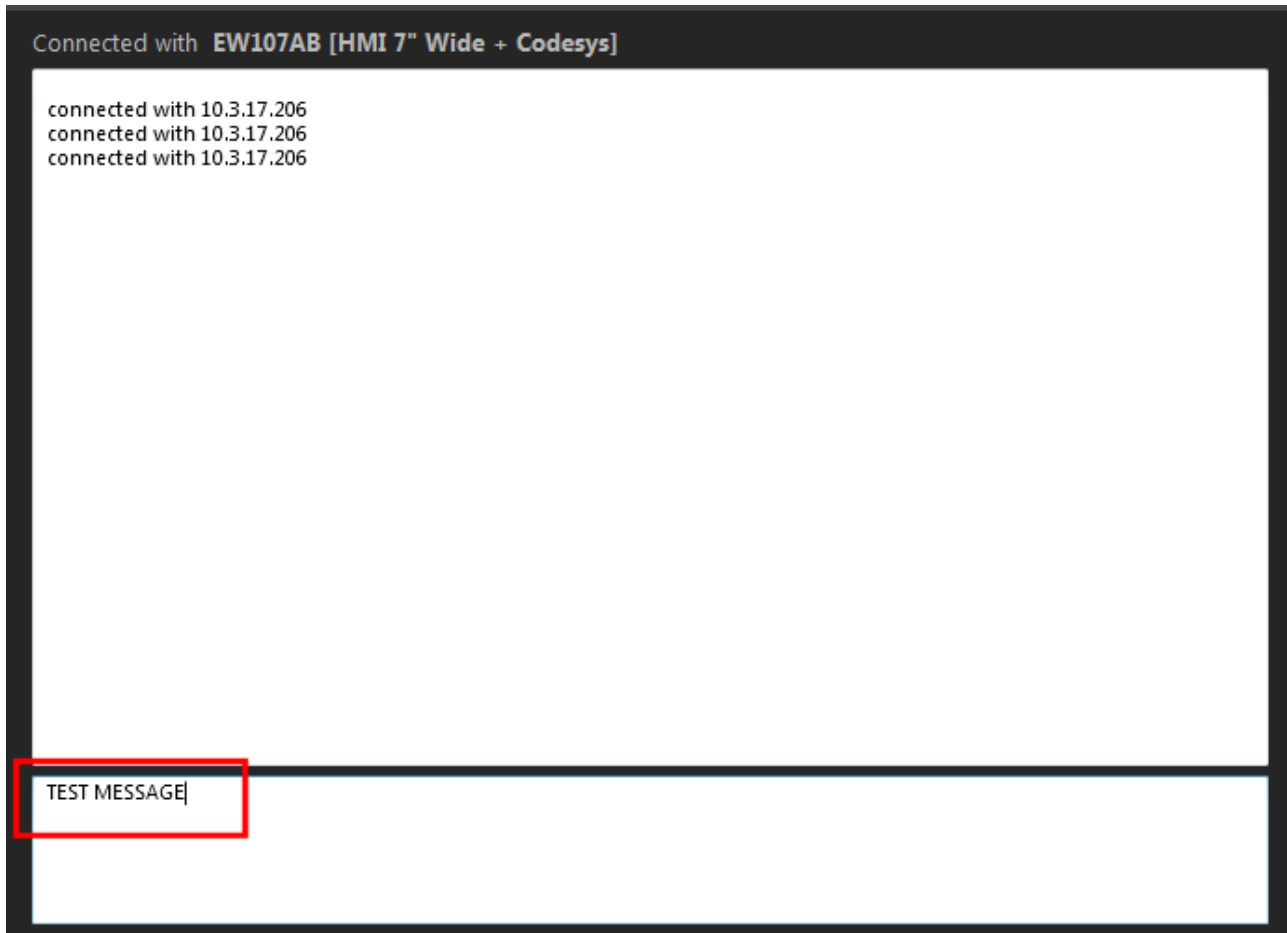
# CREW Manual

This displays the page for successful connection with the terminal.



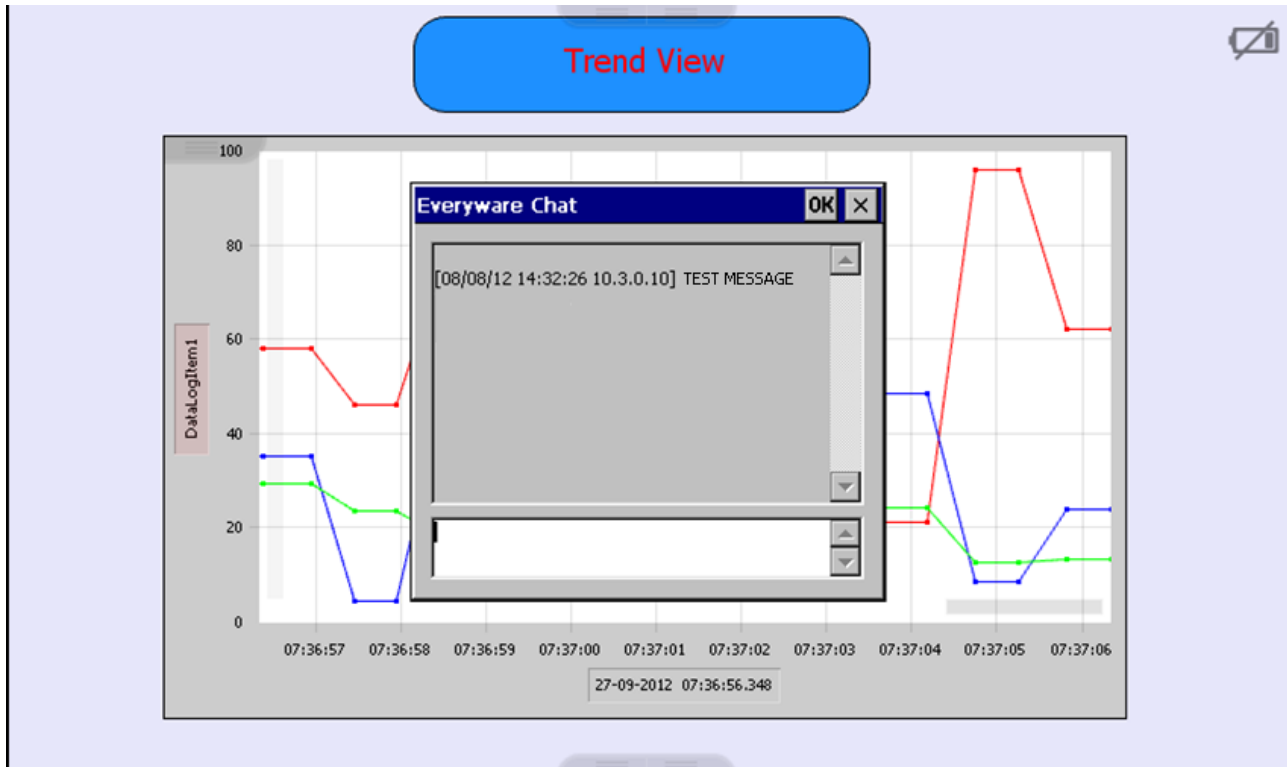
# CREW Manual

From the above page it is possible to enter text messages that will be shown on the terminal.



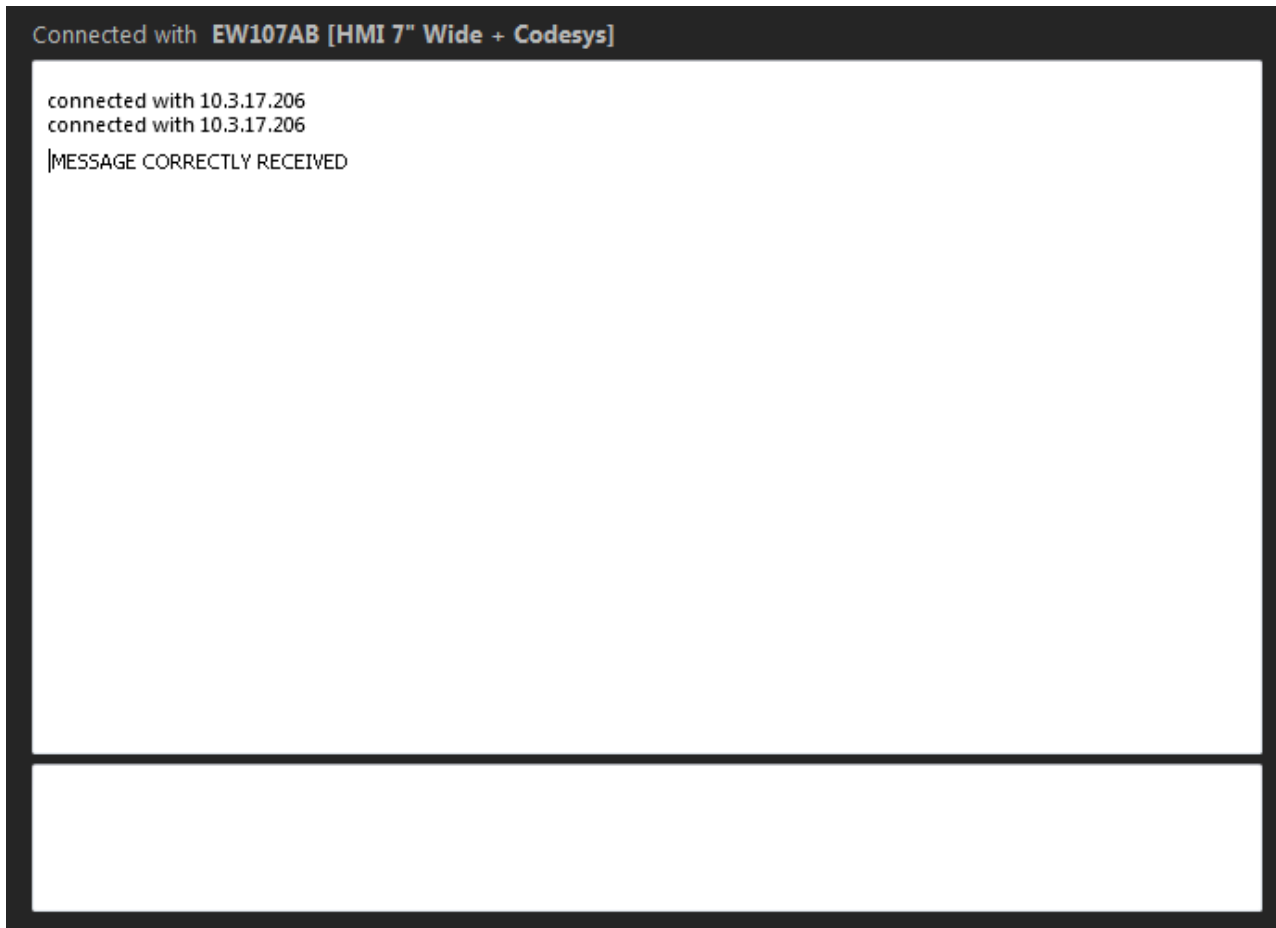
# CREW Manual

After pressing “Enter” on the administrator keyboard, the message appears on the remote terminal in use.



# CREW Manual

To reply, the user in front of the terminal needs to enter the text and send it by pressing the “Enter” key of the popup keyboard that appears on the terminal. The administrator immediately receives the reply.

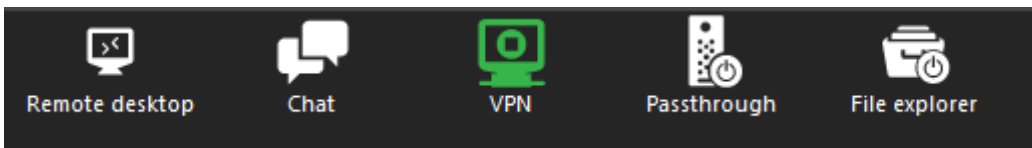


# CREW Manual

## VPN (Virtual Port Network)

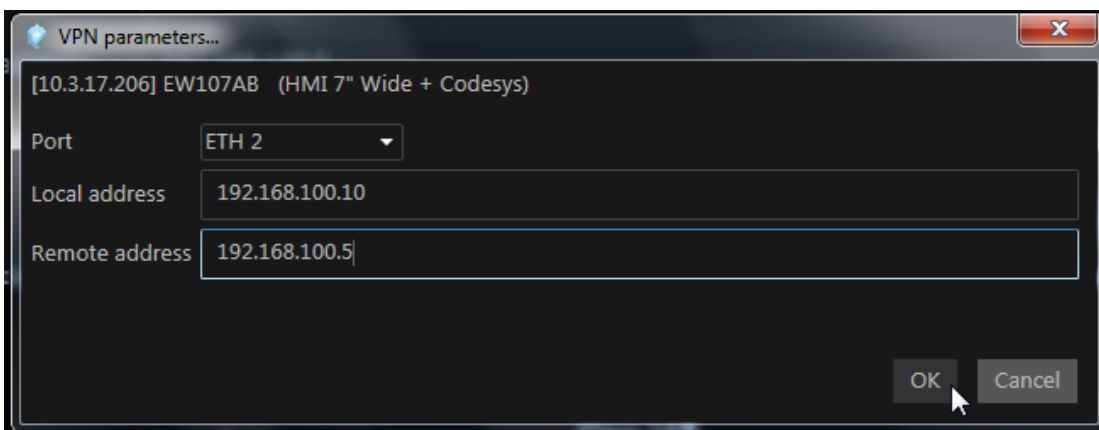
Note: The VPN is created automatically during installation of the Everyware application on the administrator's PC.

Click the relative icon to set the VPN parameters.



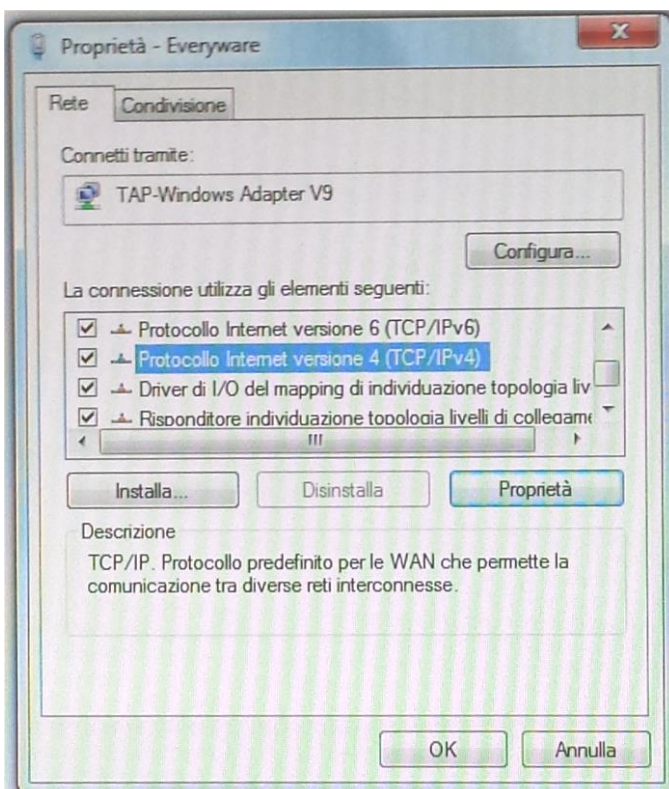
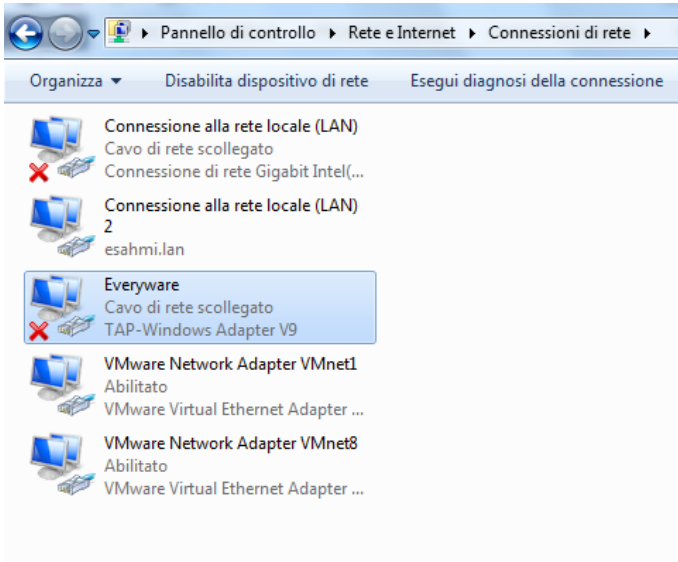
In the box that appears, two compatible IP addresses need to be entered and the Ethernet port to be used on the terminal needs to be selected:

- 1) Local Address: the IP address of the virtual Everyware port.
- 2) Remote Address: the IP address of the panel (to interact with a terminal) or the IP address of a PLC (to work on a PLC with dedicated software).



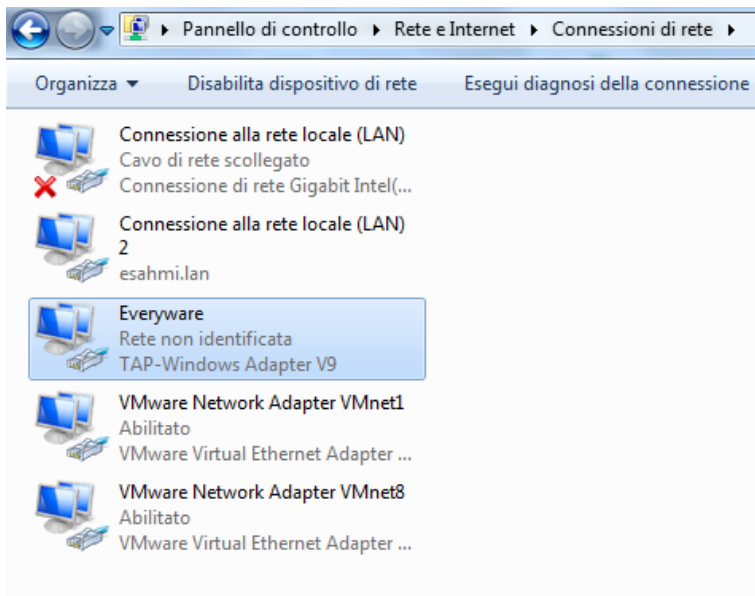
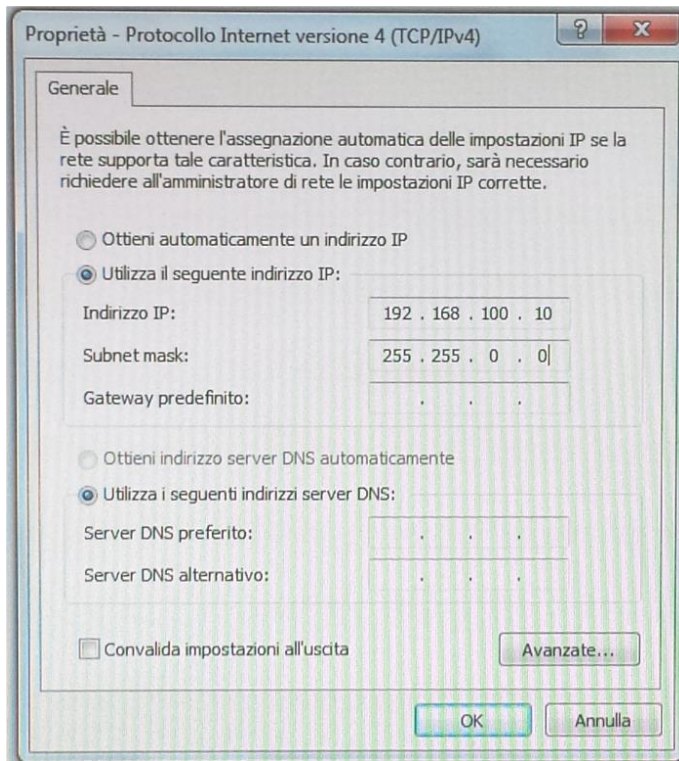
# CREW Manual

Below, the sequence of operations to be carried out on the administrator PC to set the IP address of the virtual Everyware port.



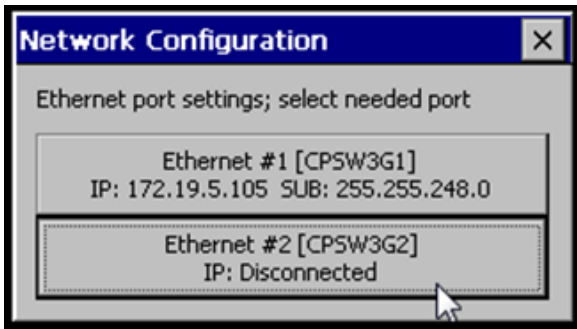


# CREW Manual



# CREW Manual

Operations to be carried out to set the IP address on the terminal.



# CREW Manual

## Passthrough

With the Everyware Passthrough function it is possible to communicate remotely with a PLC through a serial port located on the panel. For example, the administrator can download or upload a PLC program and stay connected to the panel with the PC through an Ethernet port, while Everyware creates a virtual serial port for communication with the PLC.

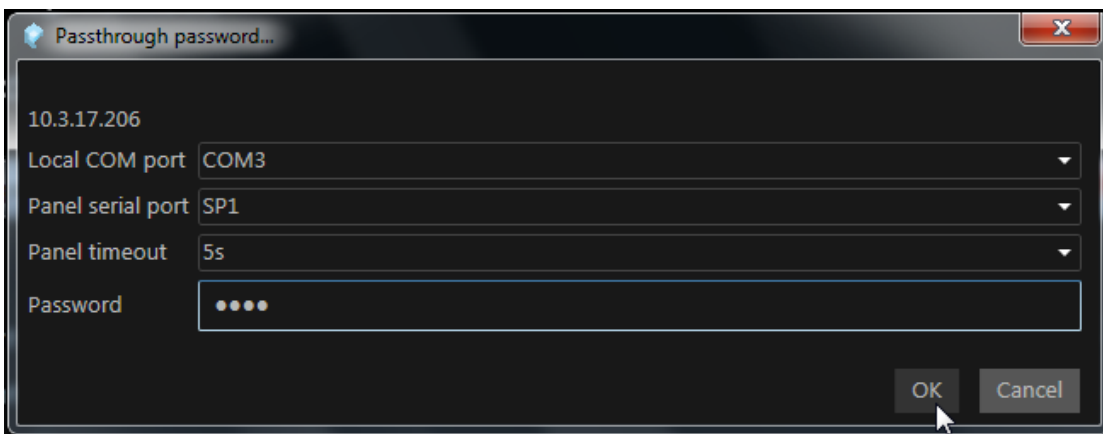
Click the relative icon to start Passthrough.



The mask below asks you to:

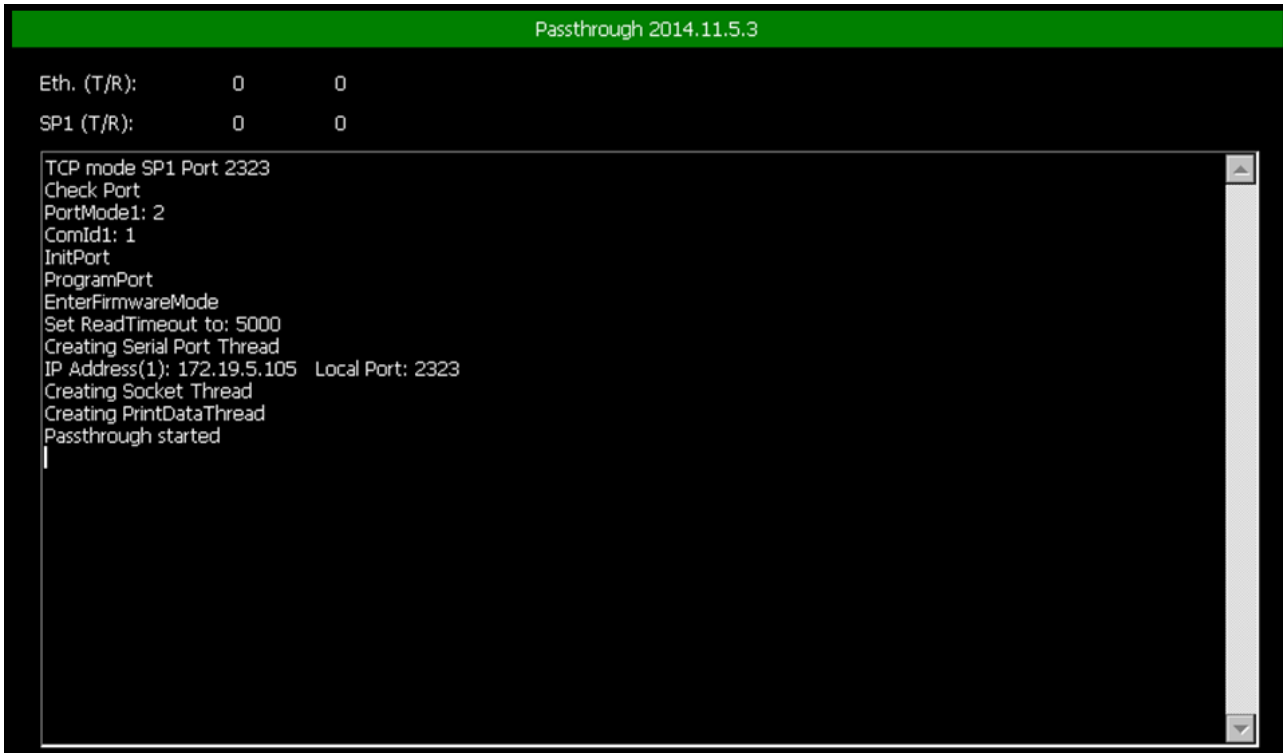
- Choose which virtual COM port to use.
- Choose which panel COM port to use.
- Decide on the duration of panel time-out.
- Enter the password for the connection (the same password that is used for the remote desktop).

At the end click “Ok”.



# CREW Manual

The correct Passthrough start-up mask appears on the terminal.



```
Passthrough 2014.11.5.3  
Eth. (T/R):      0      0  
SP1 (T/R):      0      0  
  
TCP mode SP1 Port 2323  
Check Port  
PortMode1: 2  
ComId1: 1  
InitPort  
ProgramPort  
EnterFirmwareMode  
Set ReadTimeout to: 5000  
Creating Serial Port Thread  
IP Address(1): 172.19.5.105  Local Port: 2323  
Creating Socket Thread  
Creating PrintDataThread  
Passthrough started  
|
```

## Explore files

With Everyware’s “Explore files” function it is possible to explore the folders contained in the terminal to

all, eliminate or edit the file from one’s remote control workstation.

Click the relative icon to start the Explore files function.



On the left side of the following screen it is possible to browse the folders on the local computer (administrator) and possibly on the network connected to it.

# CREW Manual

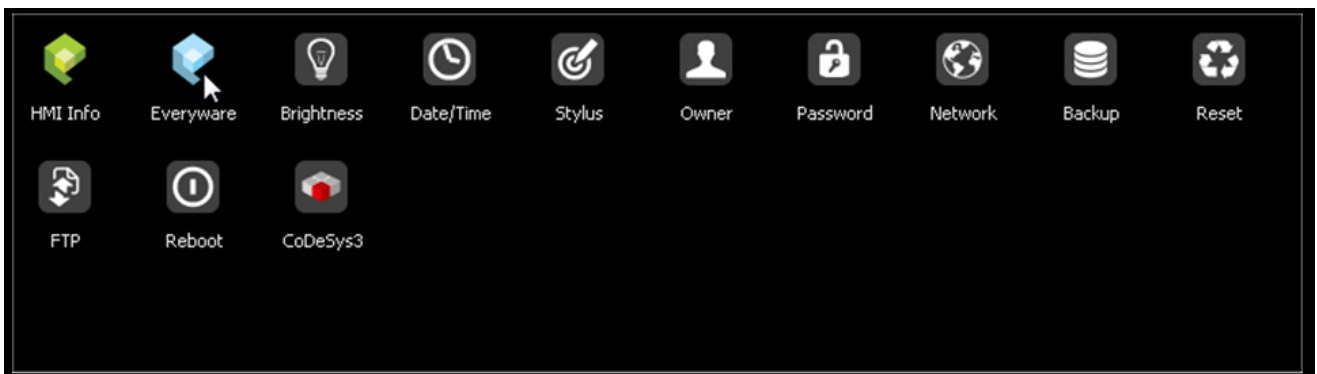
The right side displays the folders contained on the terminal. For example, if you wish to copy a file in the terminal, simply select it in the left side and drag it to the right side.



# CREW Manual

## Settings on panel side

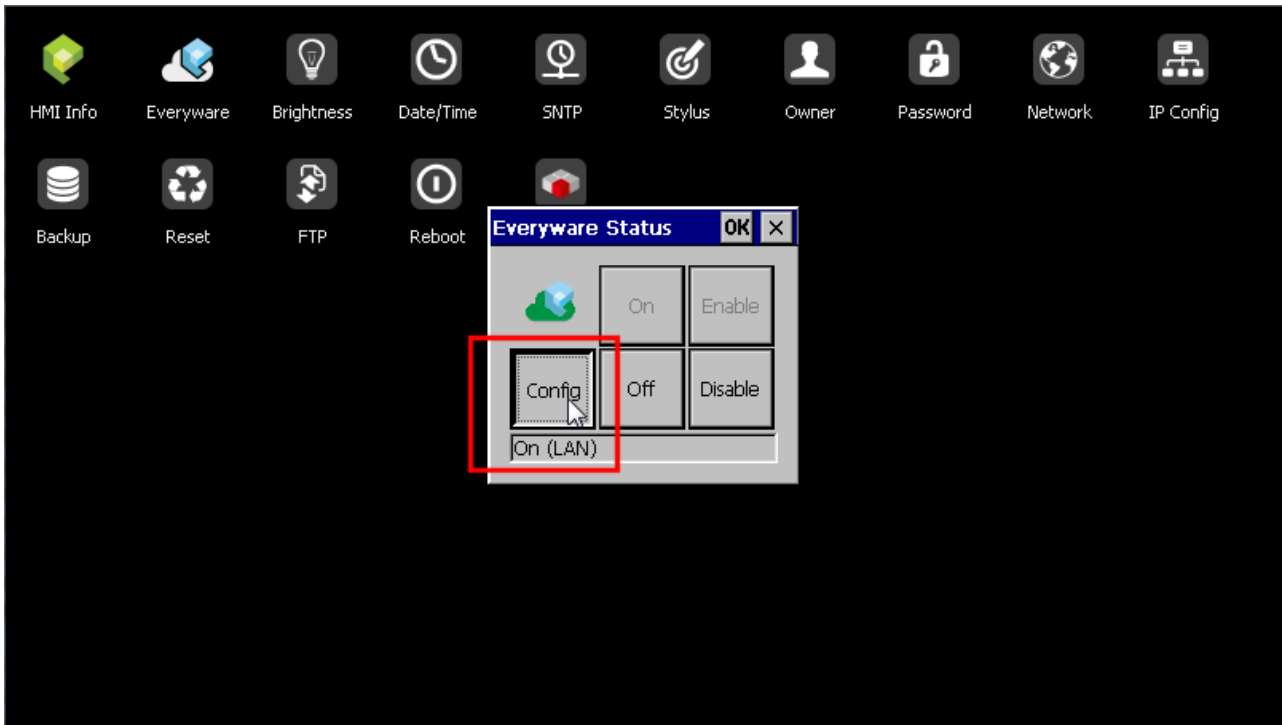
Power the terminal, connect the network cable to the terminal's Ethernet port (ETH1, the port closest to the power supply connector), enter "Control panel" and select the "Everyware" icon.



# CREW Manual

## Everyware Configuration

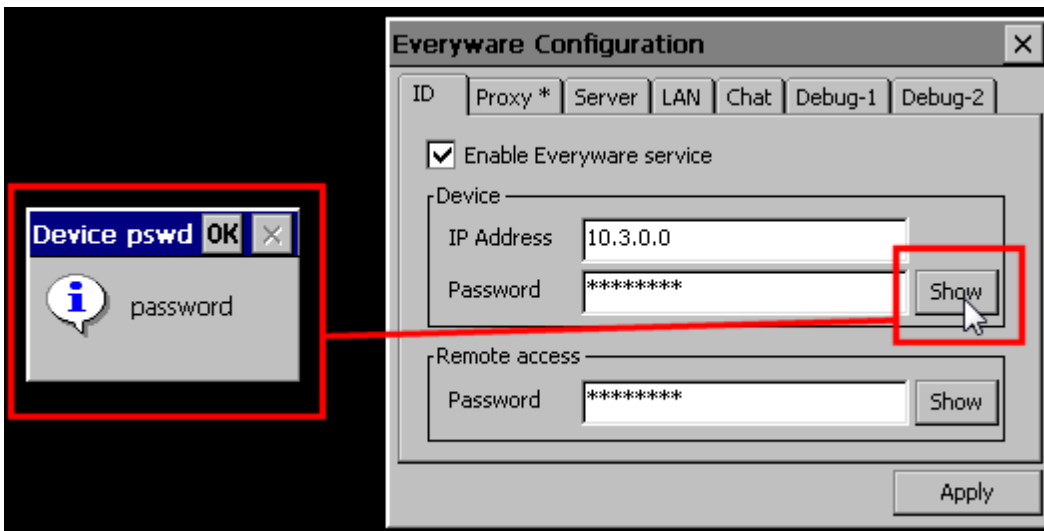
Click the “Config” key.



# CREW Manual

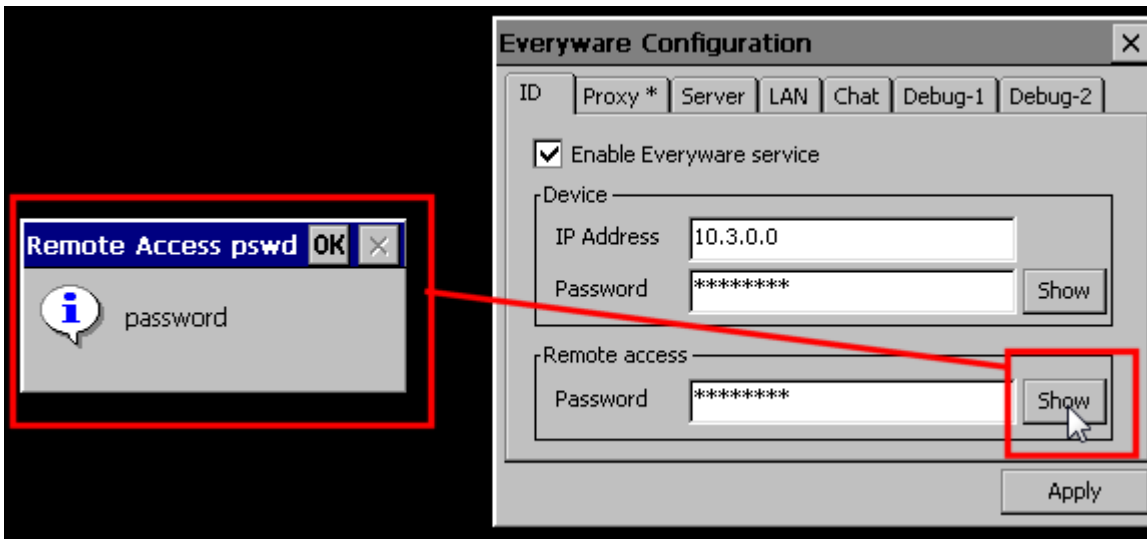


"ID" Settings:

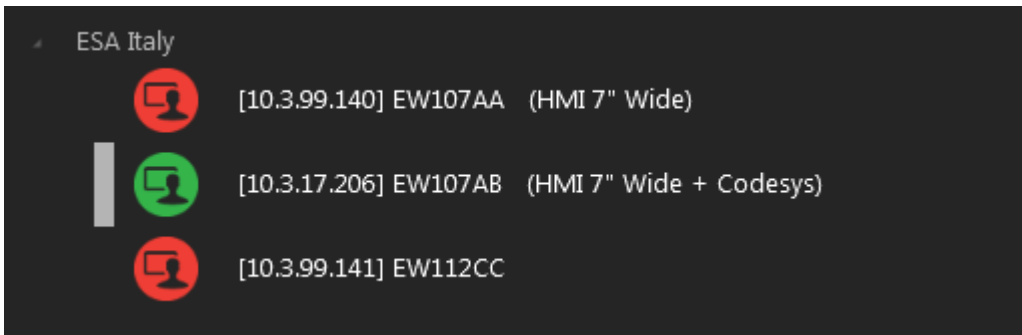




# CREW Manual



- Ensure that the “Enable Everyware service” checkbox is checked.
- Enter the terminal’s IP Address (enter the exact address assigned automatically by the Everyware server).



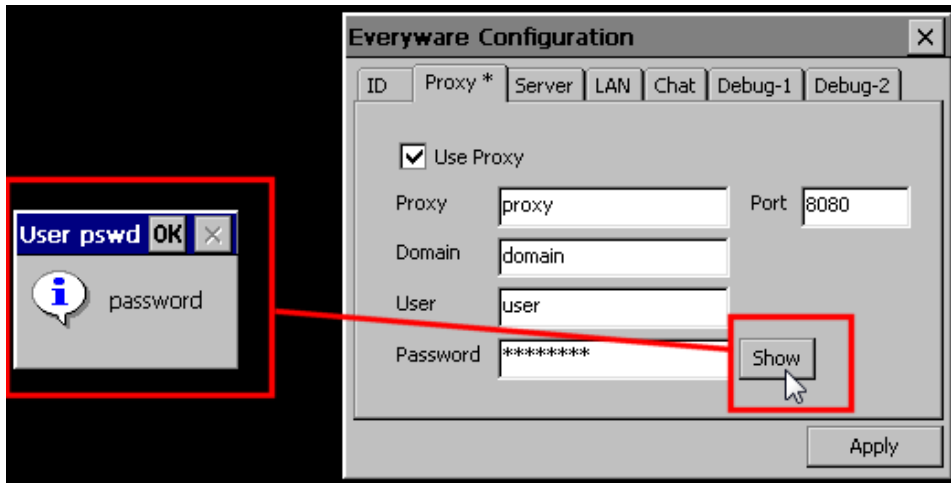
- Assign a password for the connection to the terminal and one to use the “Remote desktop” function.

IMPORTANT: Memorise the passwords for future actions.

- Click “Apply” to implement the settings.

# CREW Manual

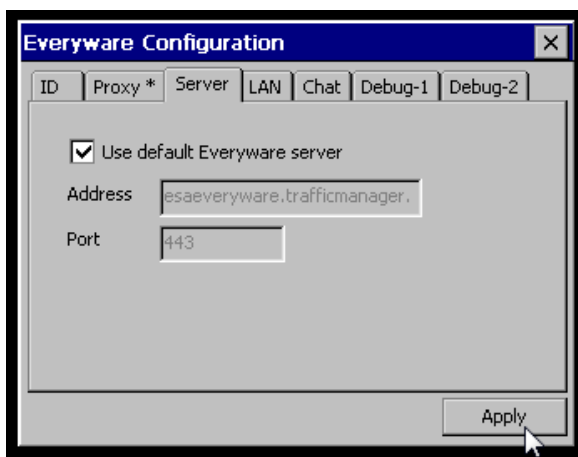
“PROXY” settings:



- If the company network uses a Proxy, select the “Proxy” check box and enter the same data as in “Proxy settings”.

- Click “Apply” to implement the settings.

“SERVER” settings:

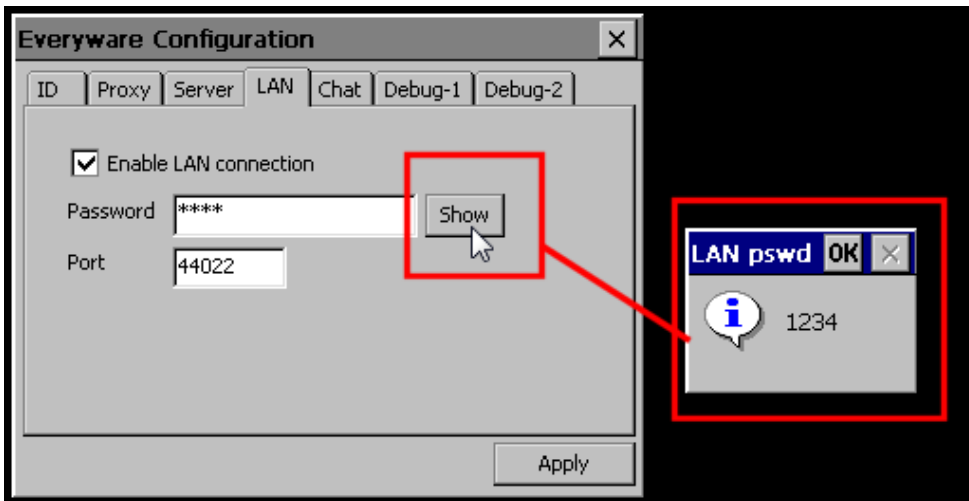


- Ensure that the “Use the default Everyware server” checkbox is checked.

- Click “Apply” to implement the settings.

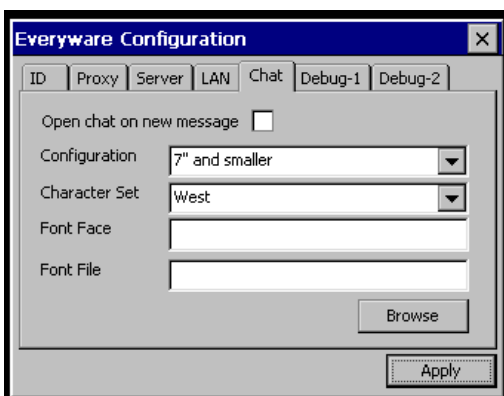
“LAN” settings:

# CREW Manual



- Ensure that the “Enable LAN connection” checkbox is checked.
- Click “Apply” to implement the settings.

“Chat” settings:



- It is possible to automatically open the “Chat” service the first time the message appears.
- It is possible to configure the sizes of the “Chat” boxes based on the type of terminal (7” and smaller than 7”, 7” and larger than 7”).
- It is possible to define the set of characters to be used (European or Cyrillic).
- It is possible to establish what type of font to use.

# CREW Manual

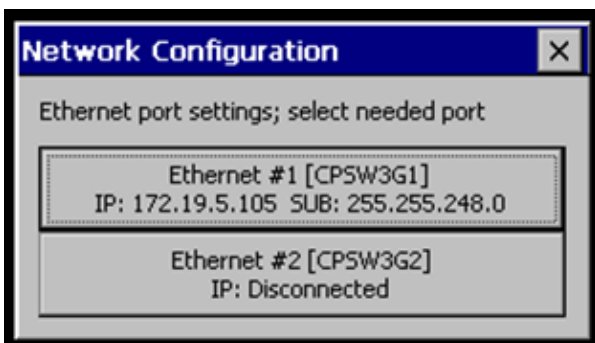
- Click “Apply” to implement the settings.

## “Network” settings

From the control panel of the terminal, click the “Network” icon.

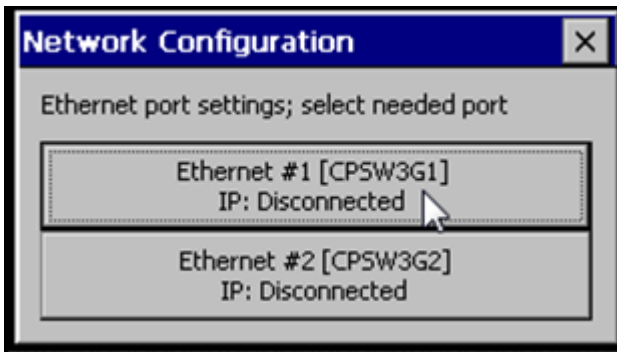


If the settings are correct, you will see the following image (Ethernet port 1 connected to the network correctly).

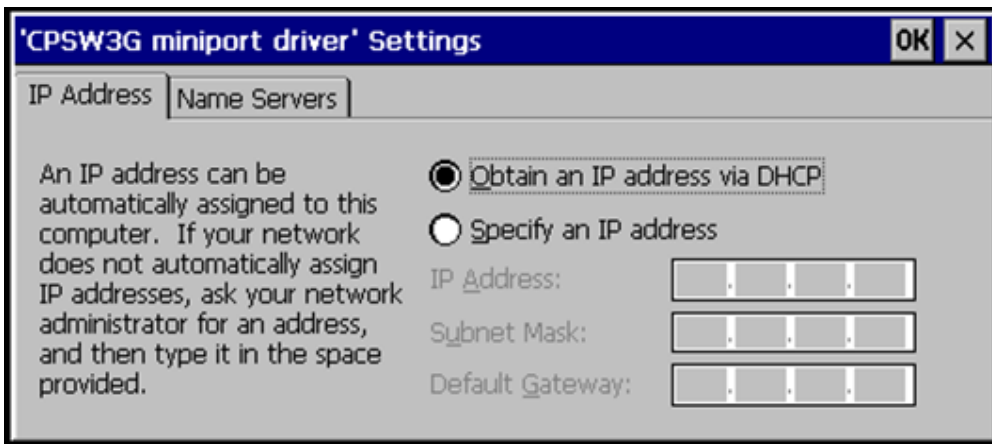


If the settings are not correct or if the Ethernet cable is not connected to the network, you will see the following image (Ethernet port 1 not connected to the network).

# CREW Manual



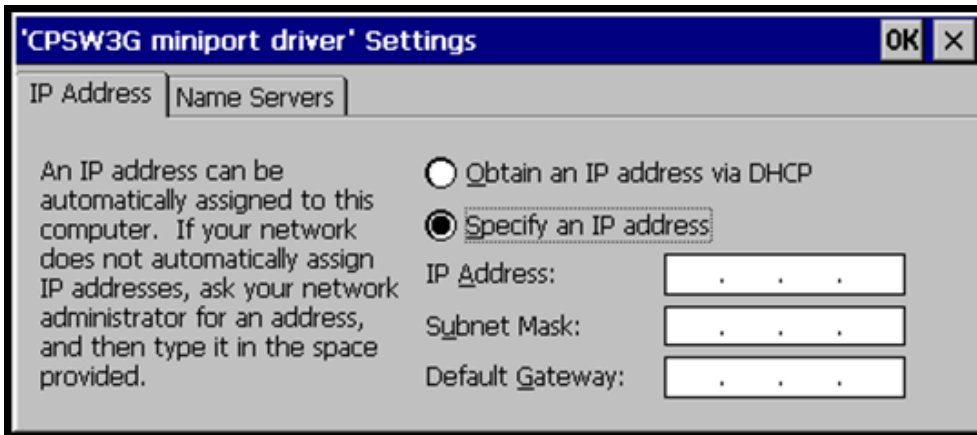
Click the used Ethernet port and enter one of the two settings shown below.



Select the option shown in the image above, to automatically assign the IP address.

# CREW Manual

Select the option shown in the image below, to have the user manually enter a valid IP address.



If all of the selected settings are correct, when you go back to the terminal's control panel and select the Everyware icon again, the icon shown in the image below turns green. The terminal is now connected to the network correctly.



Note: The blue Everyware icon means that the terminal is offline. Press the “On” key to connect the terminal to the network (the icon turns green).

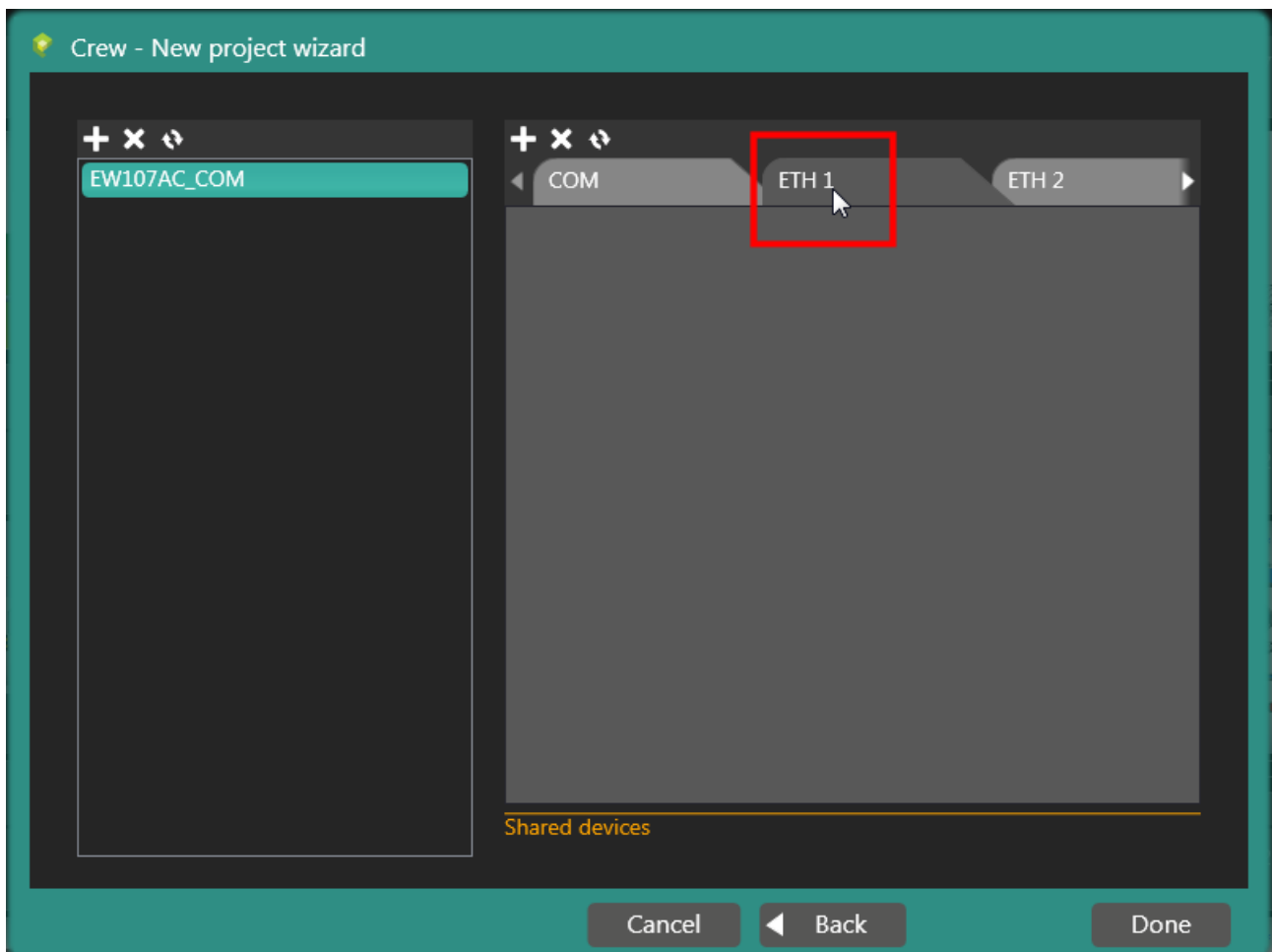
Note: The red Everyware icon means that the terminal is not able to communicate with the Everyware server, the problem may be due to:

- Incorrect data setting.
- No network.
- Faulty Ethernet connection (cable not connected correctly).
- Other hardware problem.

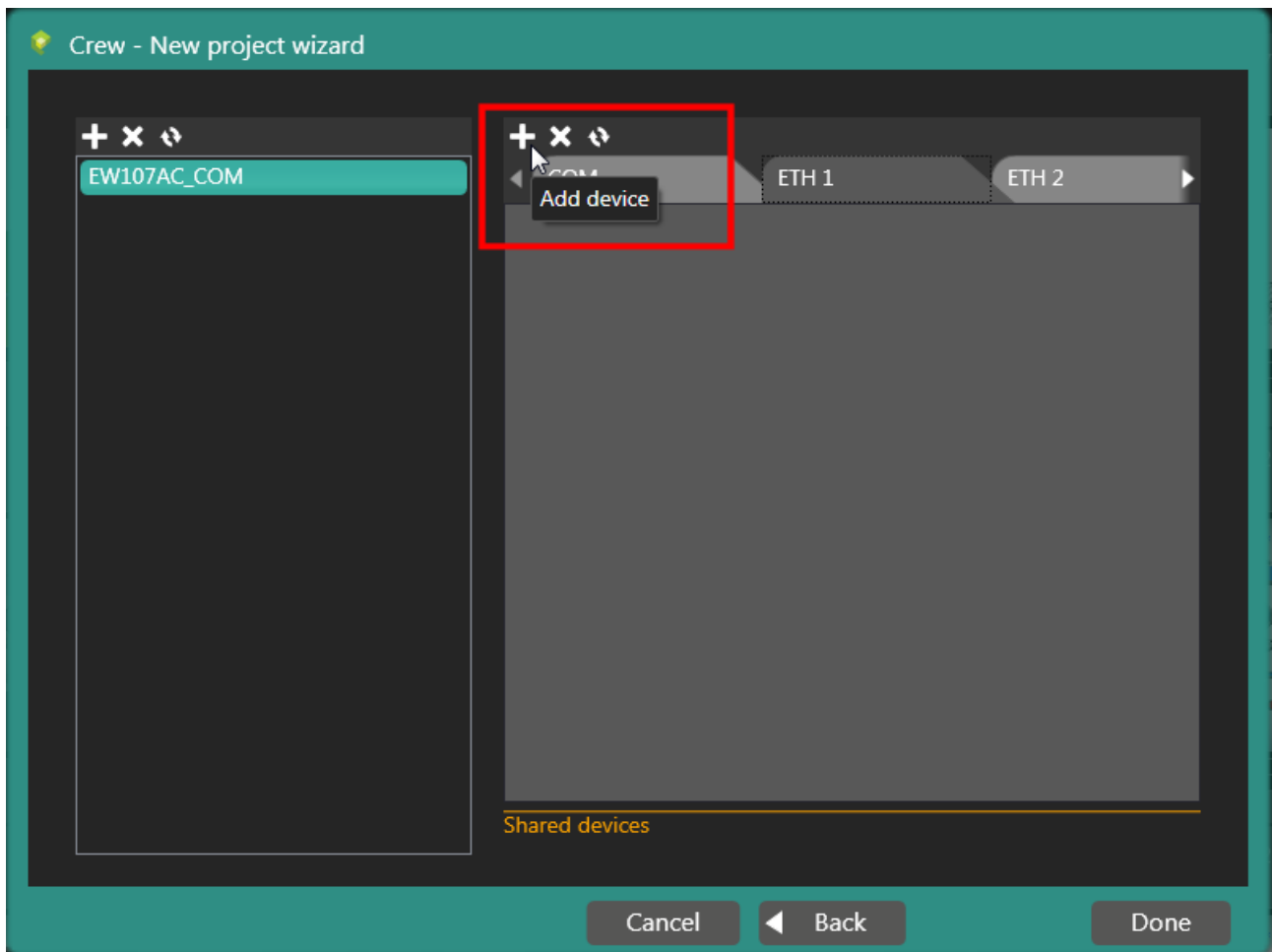
# CREW Manual

## Communication Set Up - Examples

Allen Bradley -Rockwell ControlLogix Ethernet

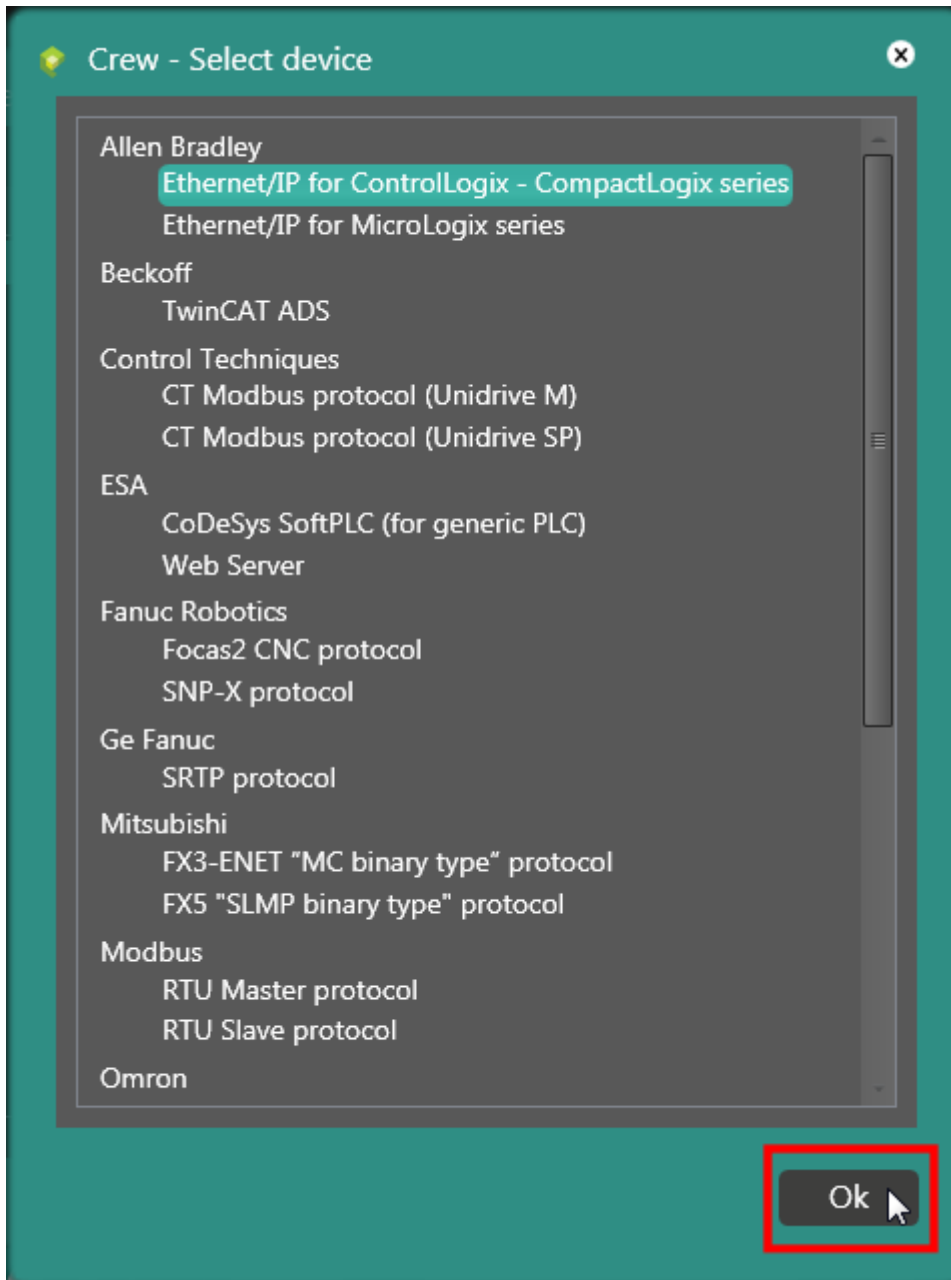


# CREW Manual





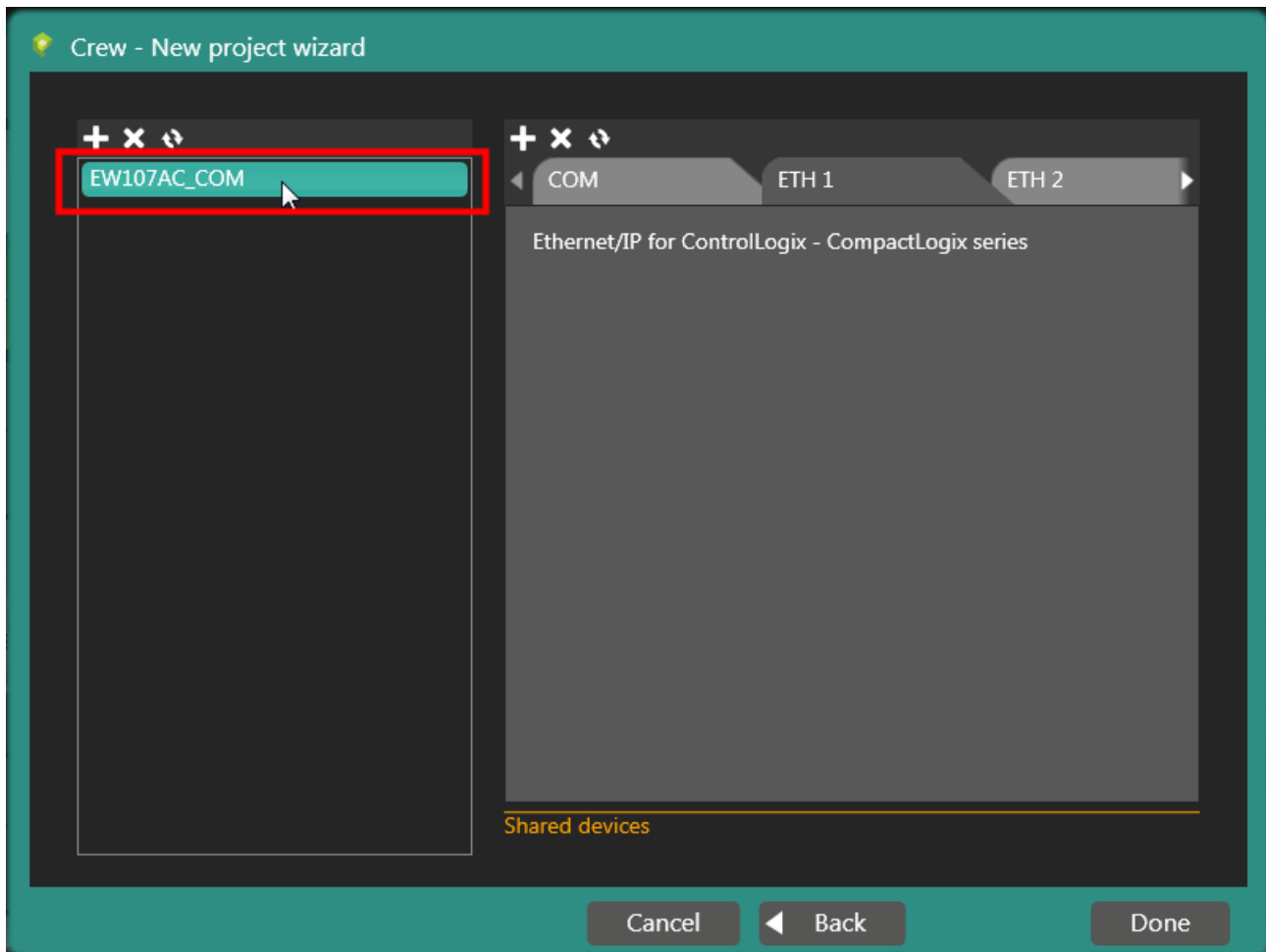
# CREW Manual



# CREW Manual

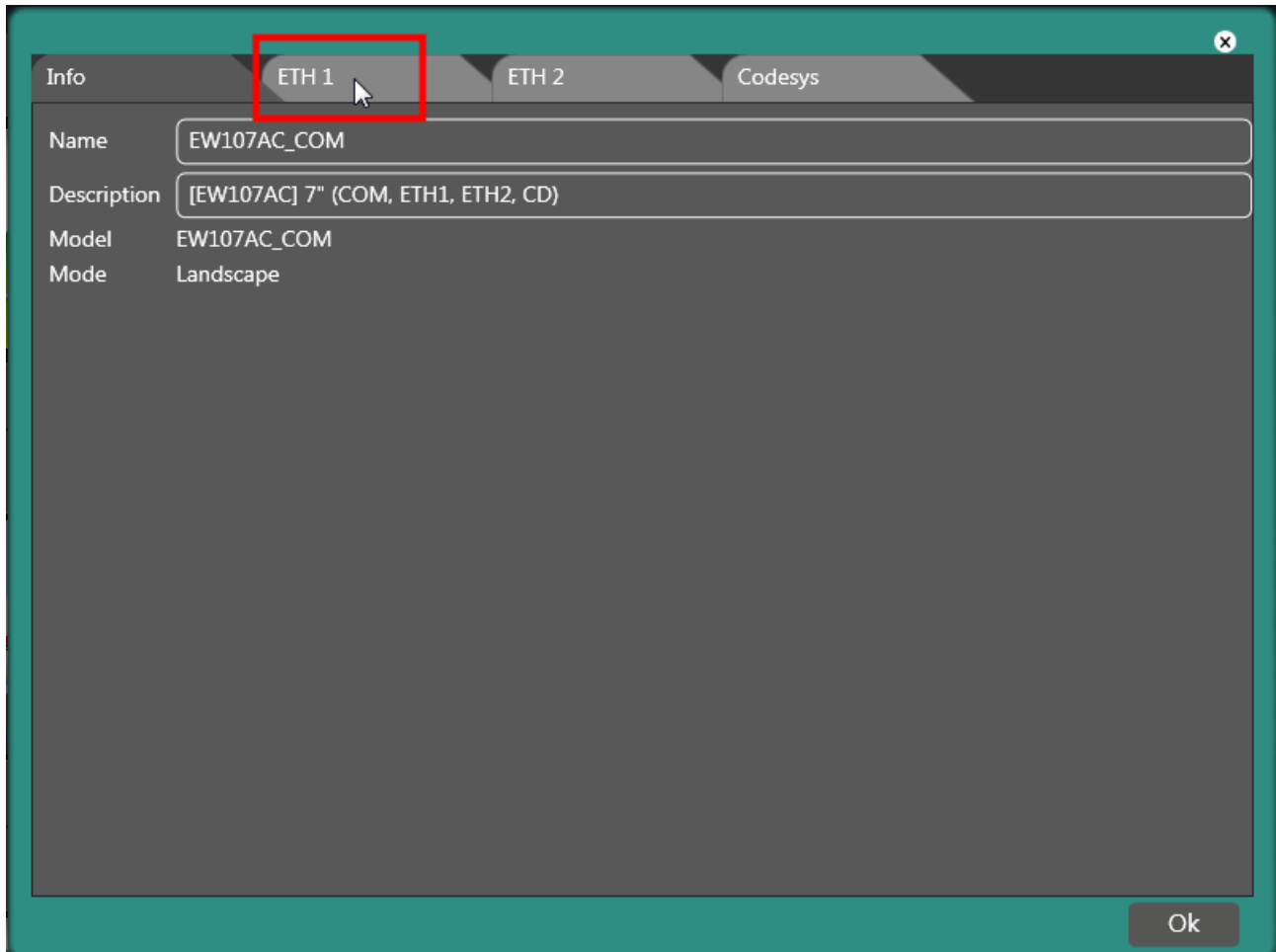
Settings on EW side:

Double click on the name of the terminal.

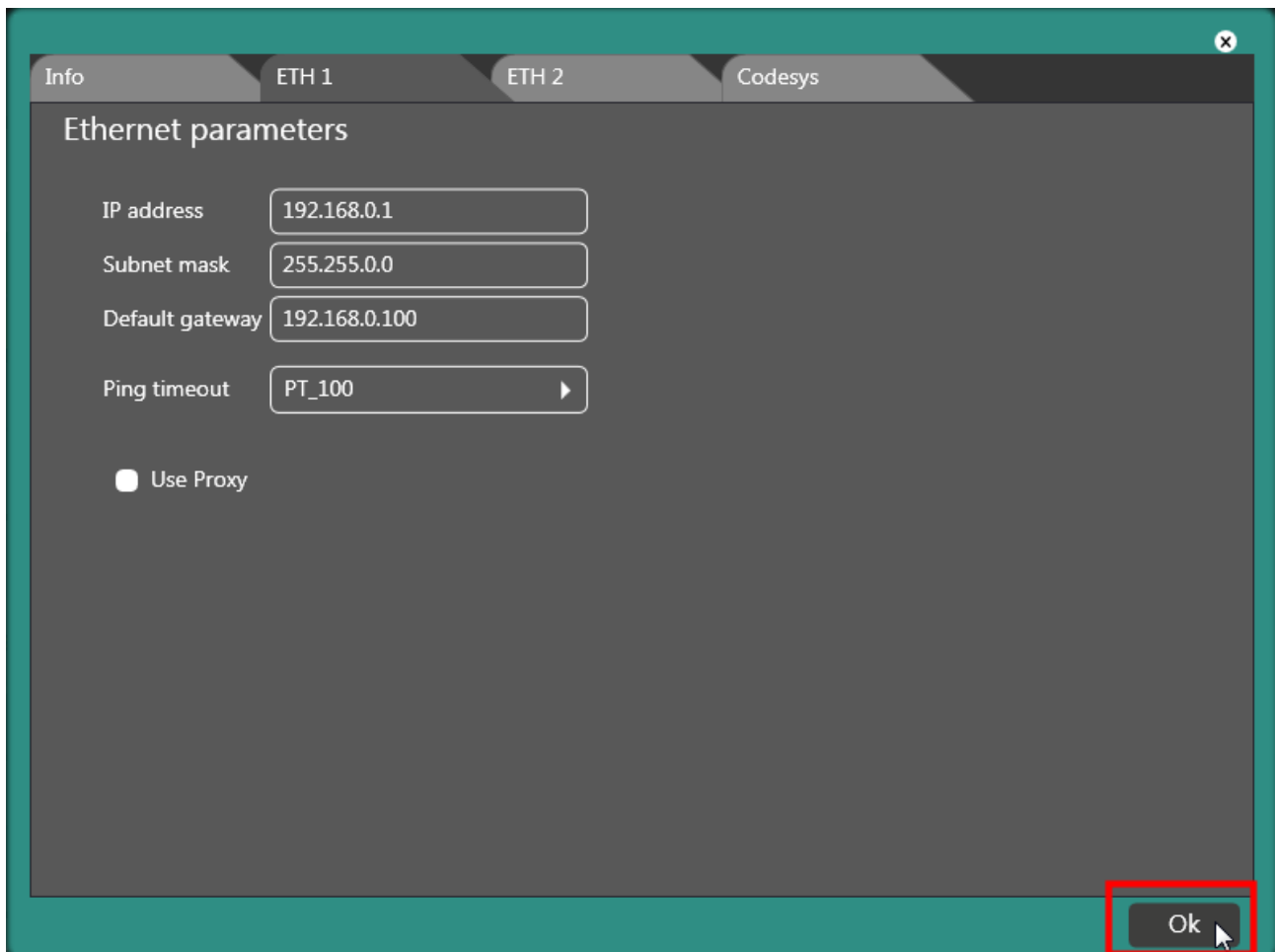


# CREW Manual

Ethernet Parameters:

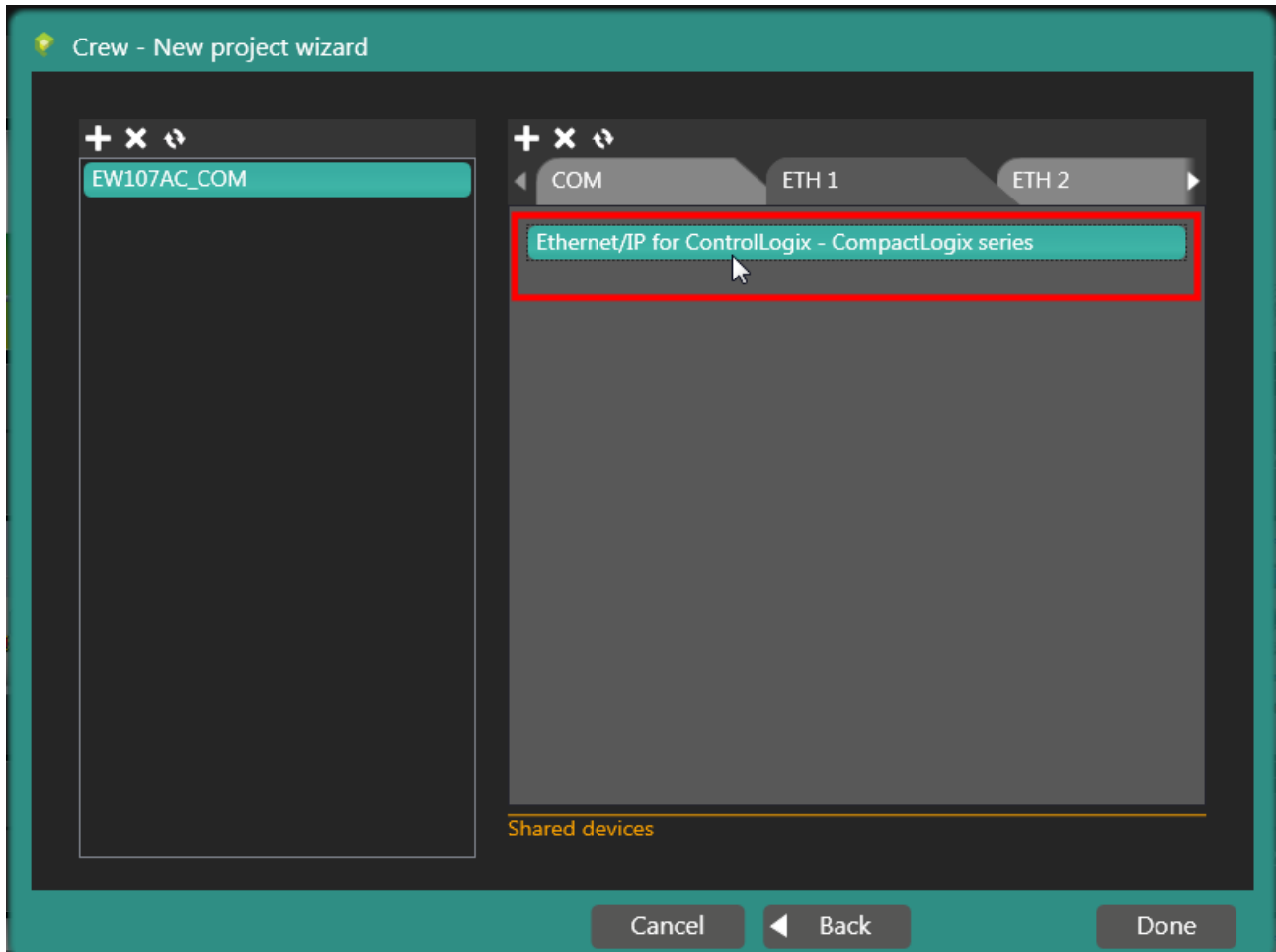


# CREW Manual



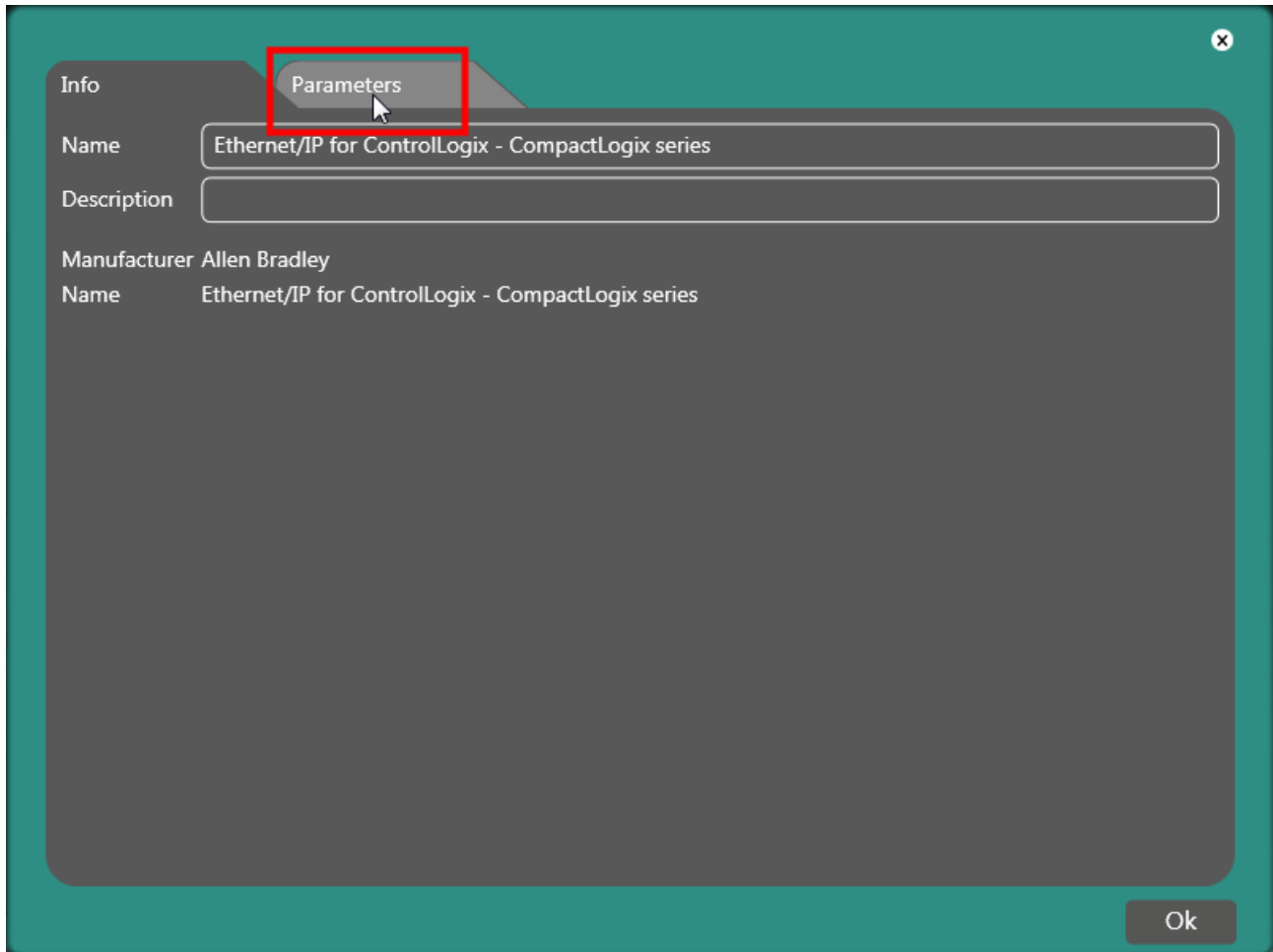
# CREW Manual

Settings on Device side. Double click on the name of the device.

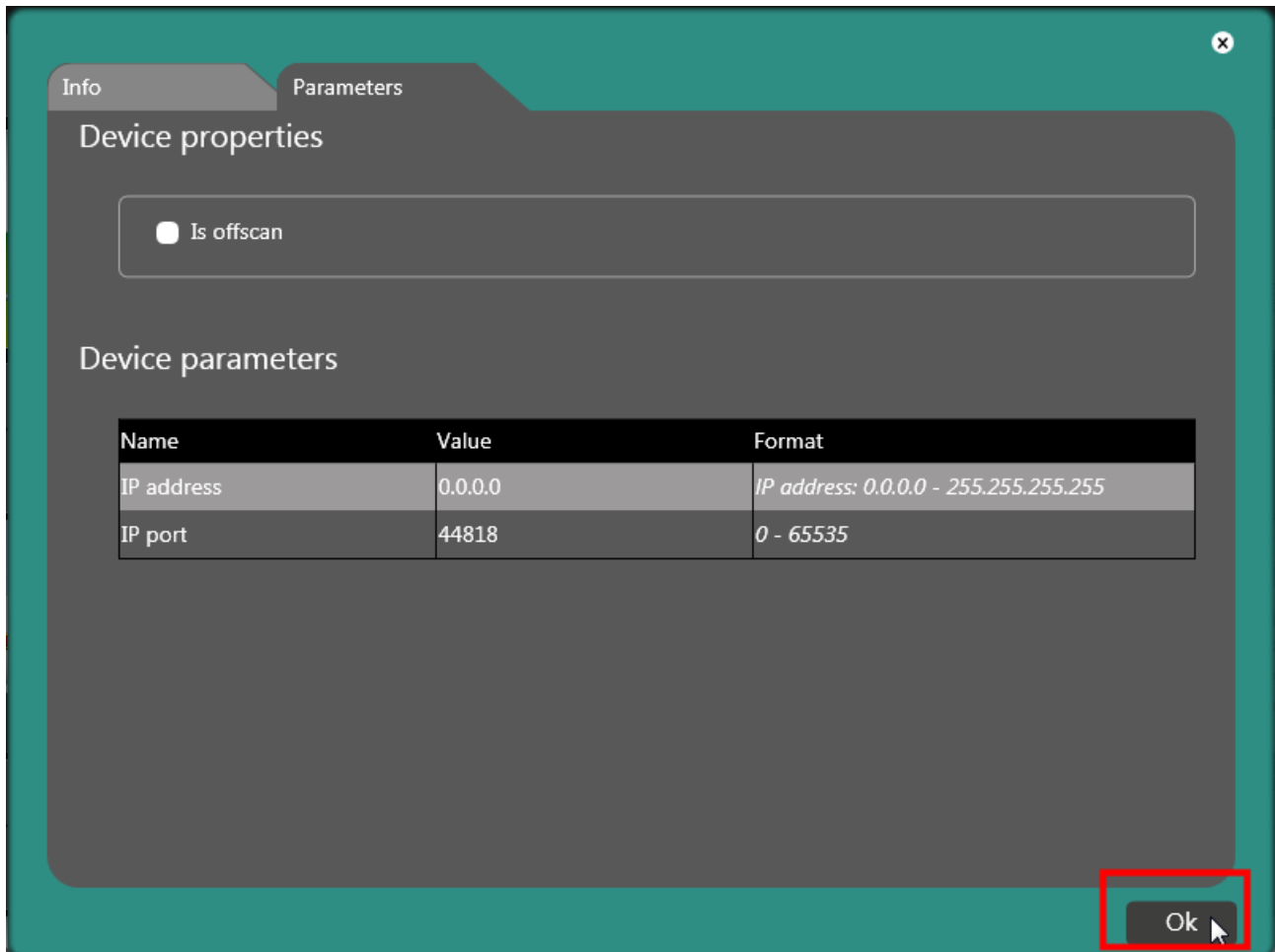


# CREW Manual

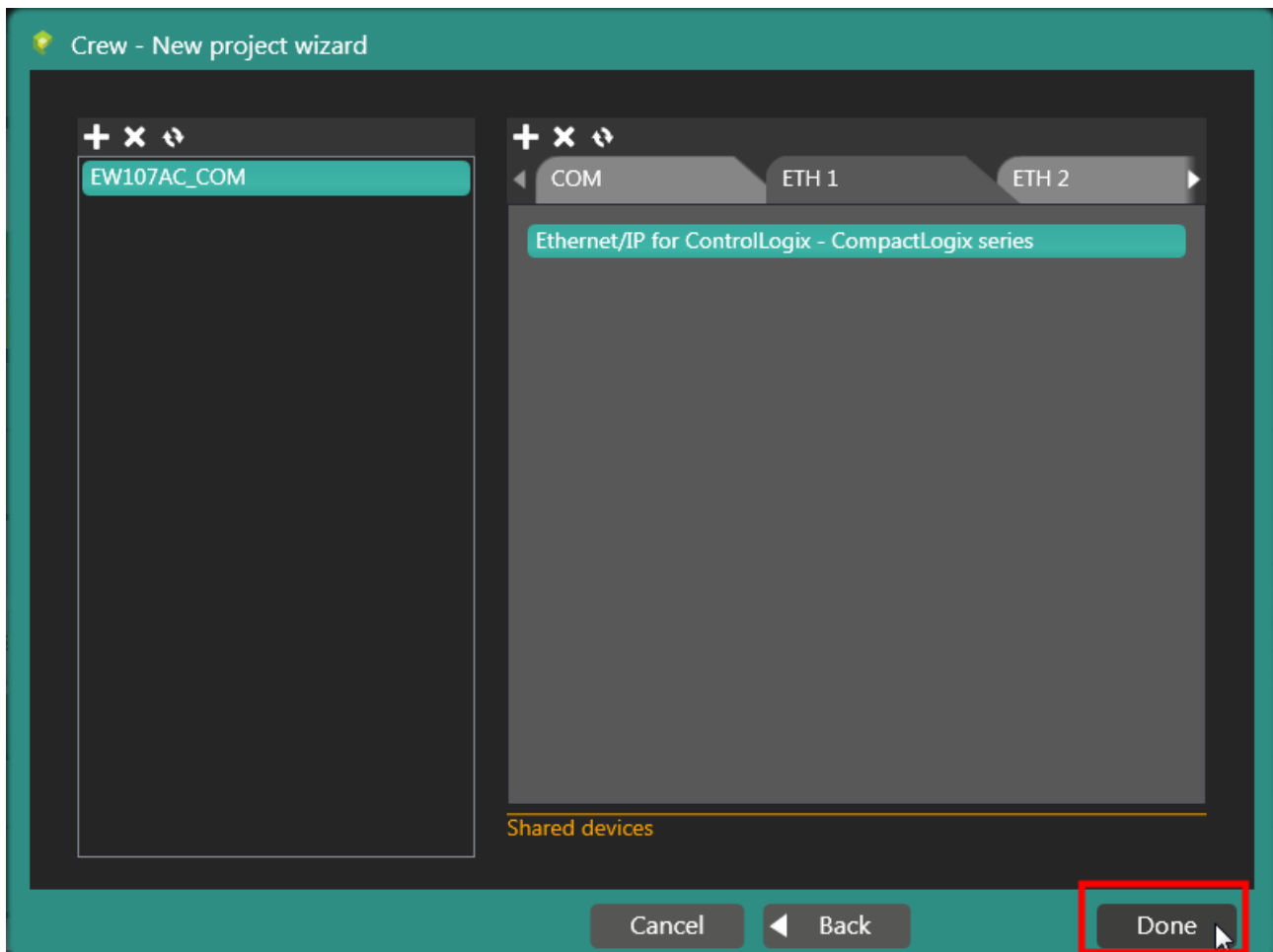
Device Parameters:



# CREW Manual



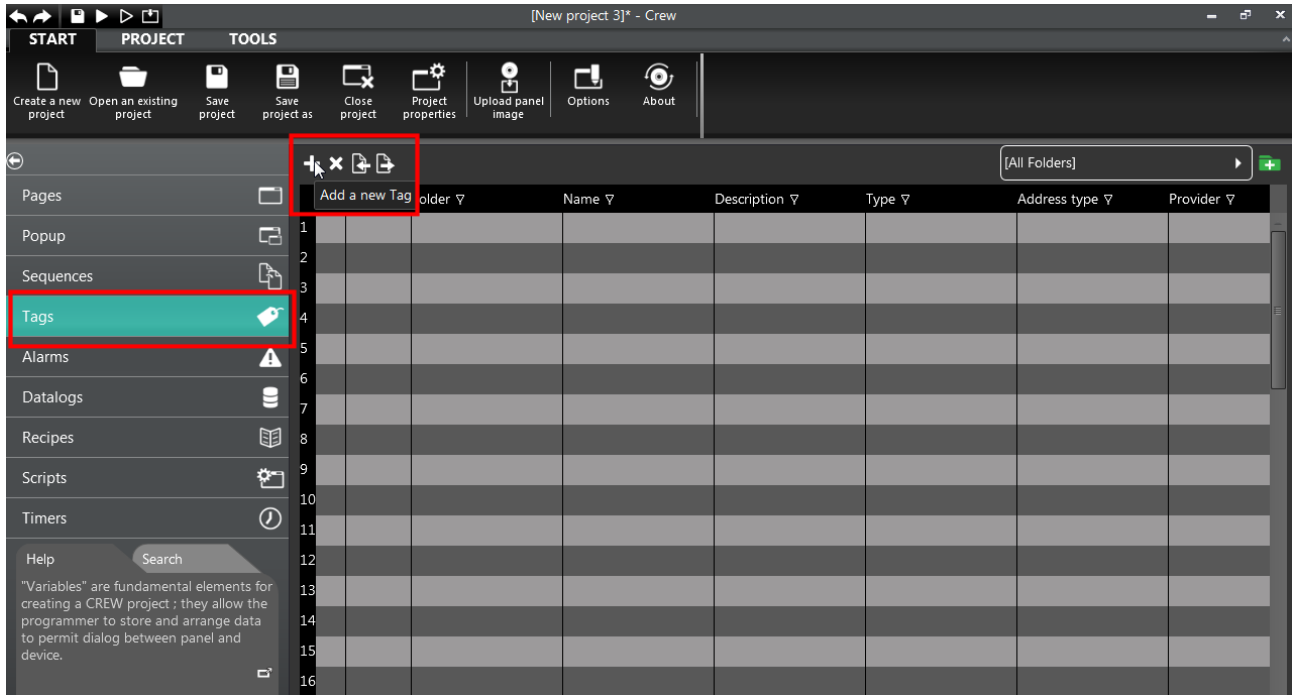
# CREW Manual





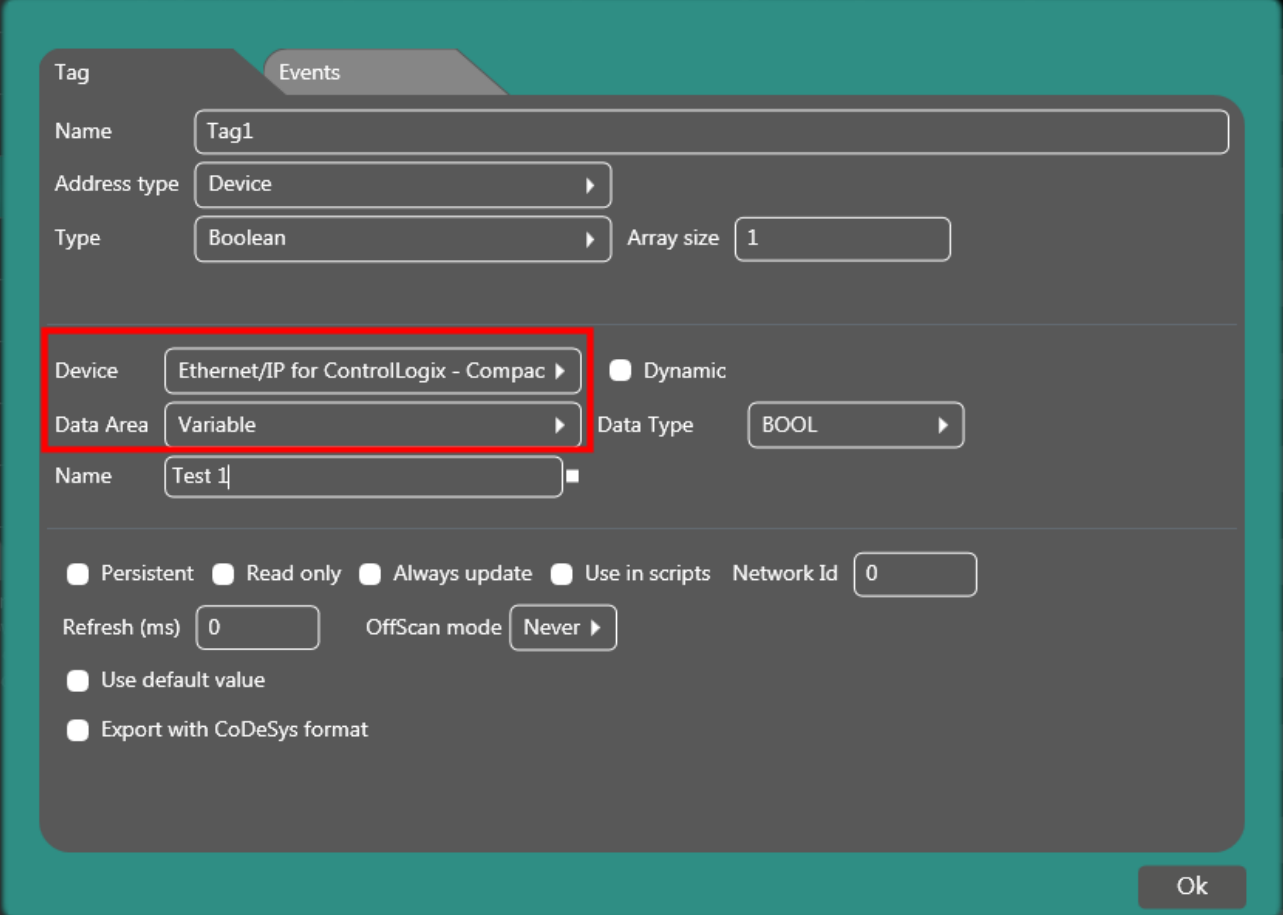
# CREW Manual

## Parameters Set up - Allen Bradley -Rockwell



# CREW Manual

Data Area: "Variable".



The screenshot shows the 'Tag' configuration dialog box with the following settings:

- Tag: Events
- Name: Tag1
- Address type: Device
- Type: Boolean
- Array size: 1
- Device: Ethernet/IP for ControlLogix - Compac
- Dynamic:
- Data Area: Variable (highlighted with a red box)
- Data Type: BOOL
- Name: Test 1
- Persistent:  Read only:  Always update:  Use in scripts:  Network Id: 0
- Refresh (ms): 0
- OffScan mode: Never
- Use default value:
- Export with CoDeSys format:

Ok

# CREW Manual

Type:

In the “Type” mask is used to designate the type of datum that the tag is destined to contain. The expected data types are represented in the following table.

Type	Description	Range
Char	8-bit signed Integer	-128 to 127
Byte	8-bit unsigned Integer	0 to 255
Boolean	Single bit	True (1) or False(0)
Integer	16-bit signed Integer	-32,768 to 32,767
Unsigned Integer	16-bit unsigned Integer	0 to 0xFFFF
Long	32-bit signed Integer	-2,147,483,648 to 2,147,483,647
Unsigned Long	32-bit unsigned Integer	0 to 0xFFFFFFFF
Real	IEEE Floating point 32-bit single precision	-3.402823E38 to -1.401298E-45 for negative values; 1.401298E-45 to 3.402823E38 for positive values
Double	IEEE Floating point 64-bit double precision	-1.7976931348623E308 a -4.9406564584124E-324 for negative values; 4.9406564584124E-324 a 1.7976931348623E308 for positive values
String	ASCII string	ASCII String (max length 0x7FFF characters)
Array of Char	Array of 8-bit signed Integers	1 a 1024
Array of Byte	Array of 8-bit unsigned Integers	1 a 1024
Array of Boolean	Array of Single bits	1 a 1024
Array of Integer (WORD)	Array of unsigned integers	1 a 1024
Array of Unsigned Integer (WORD)	Array of unsigned integers	1 a 1024
Array of Long (DWORD)	Array of 32-bit signed Integers	1 a 1024
Array of Unsigned Long (DWORD)	Array of 32-bit unsigned Integers	1 a 1024
Array of Real	Array of IEEE Floating points 32-bit single precision	1 a 1024
Array of Double	Array of IEEE Floating points 64-bit double precision	1 a 1024
Array of String	Array of ASCII Strings	1 a 1024

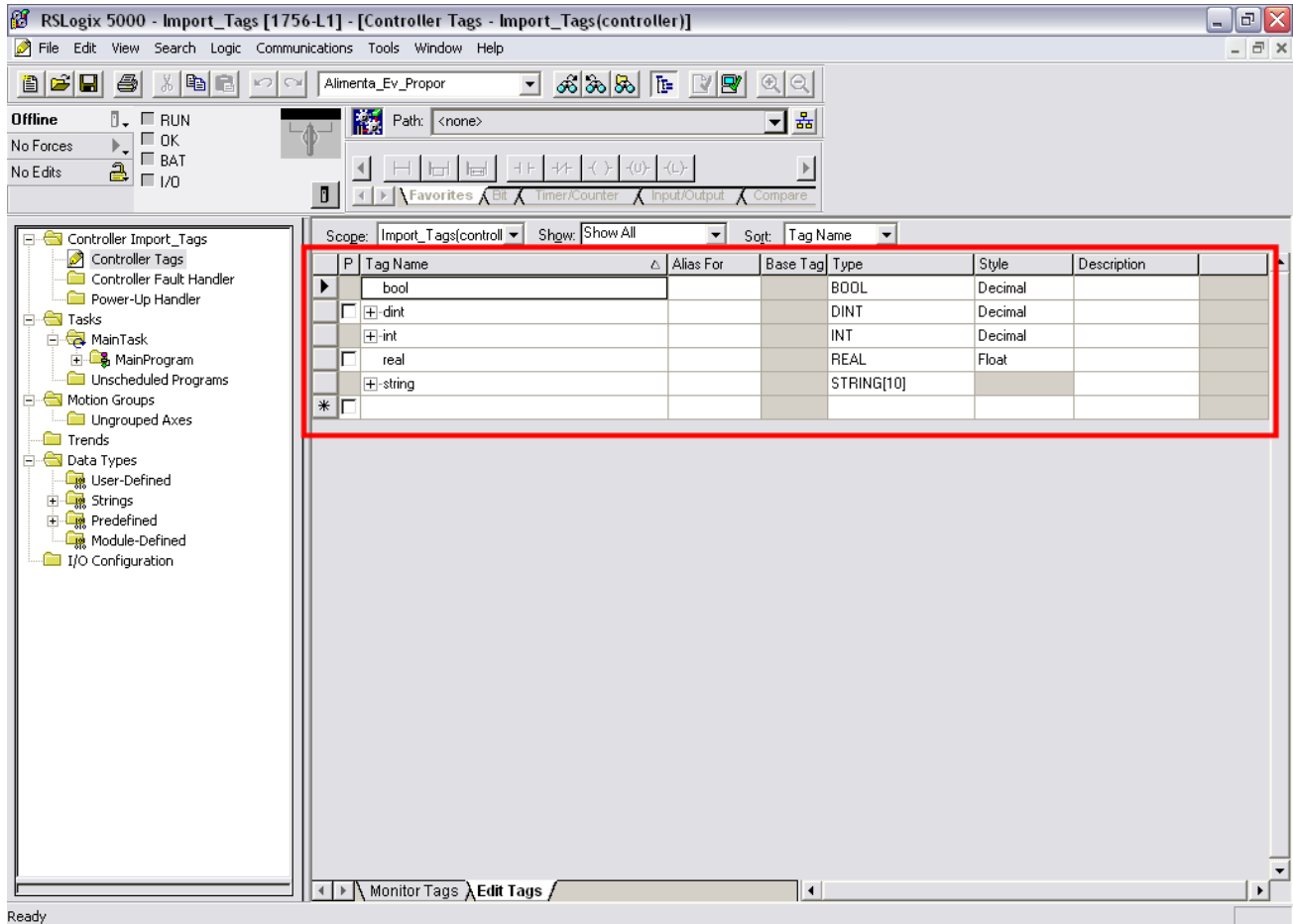
# CREW Manual

For additional information on the protocol, refer to the Drivers section, "[Allen Bradley - Ethernet-IP for ControlLogix](#)".

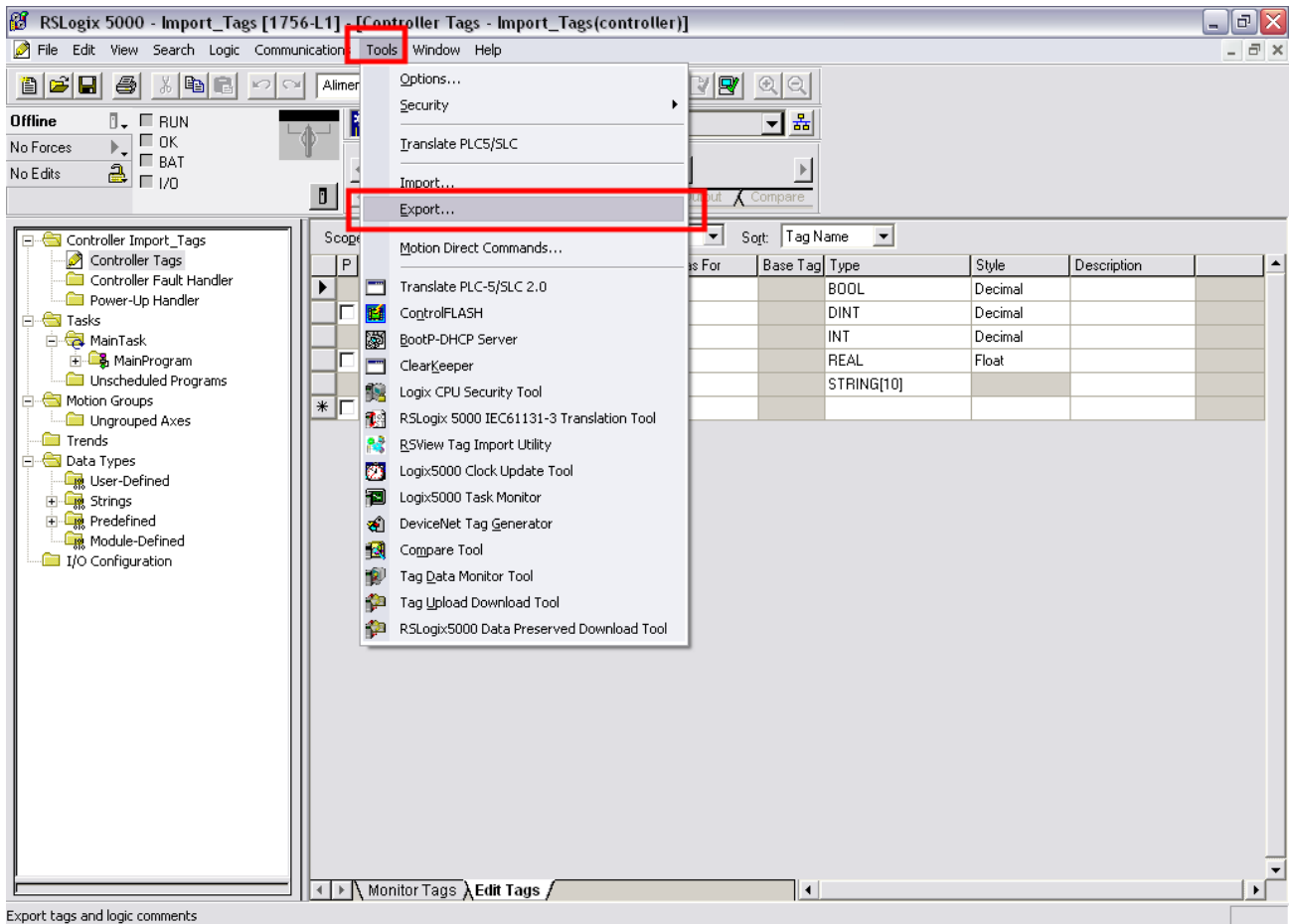
For more information on the variables (tags), refer to section "[Tags](#)".

# CREW Manual

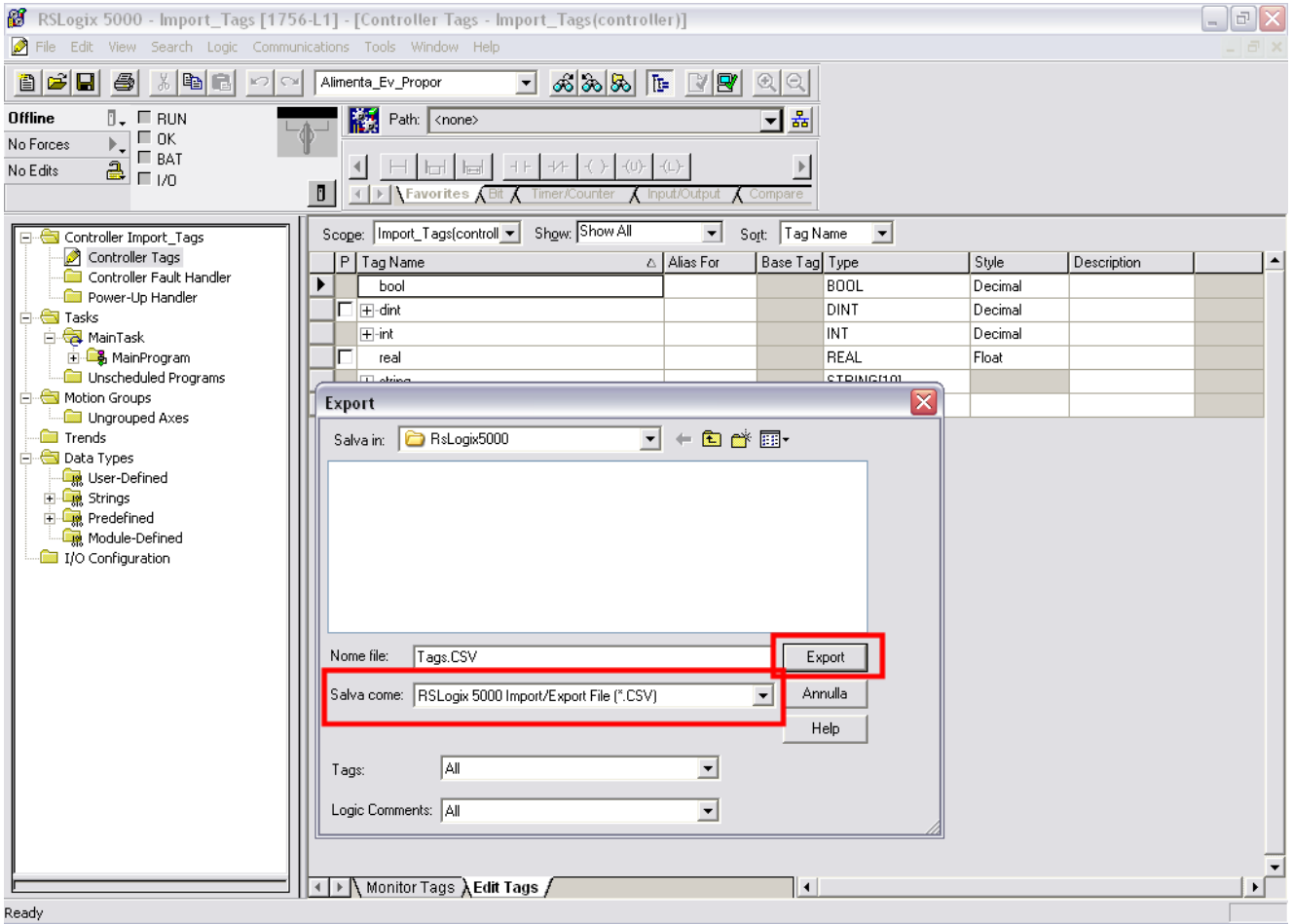
## Exporting Tags from RSLogix



# CREW Manual

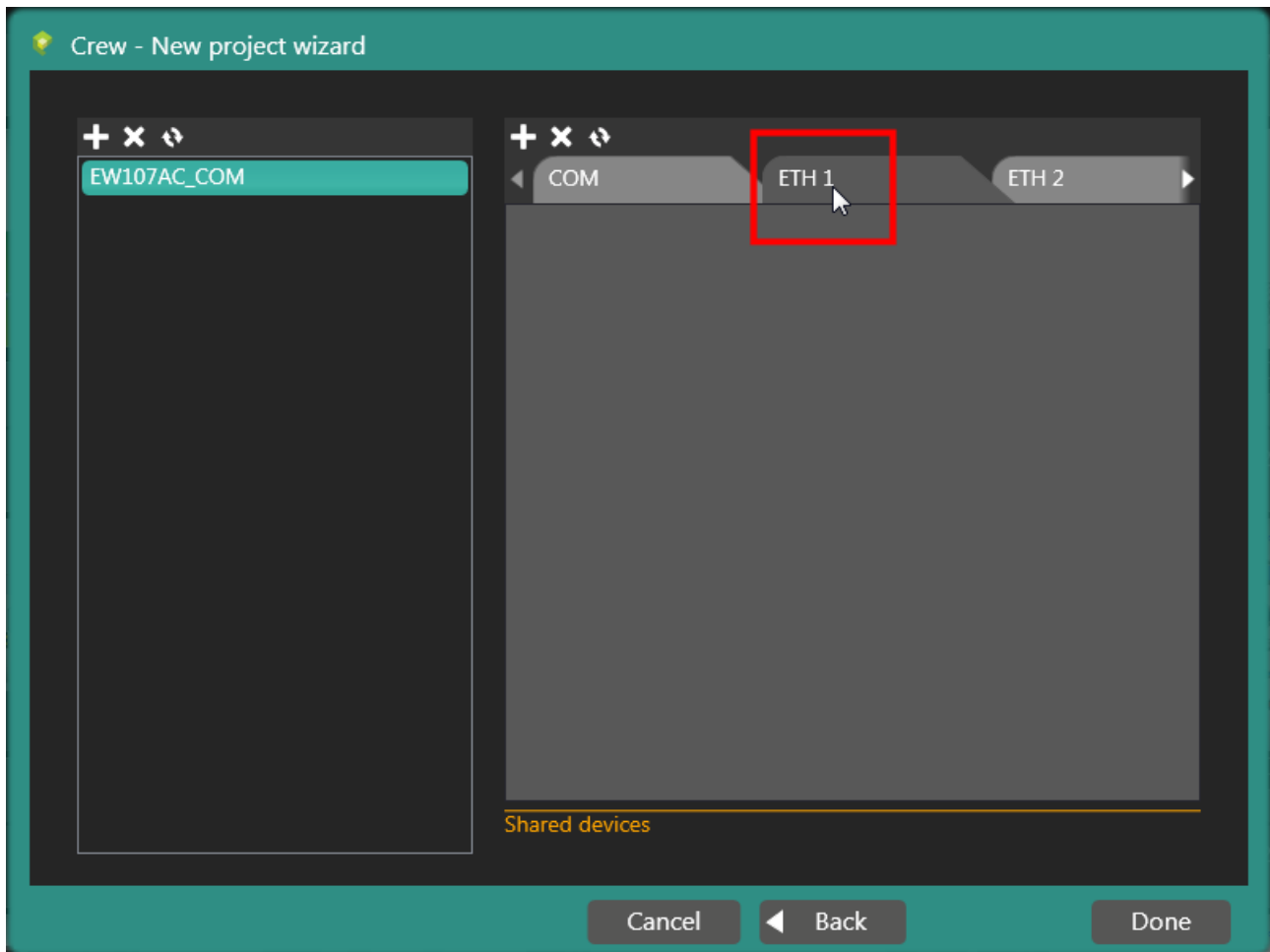


# CREW Manual



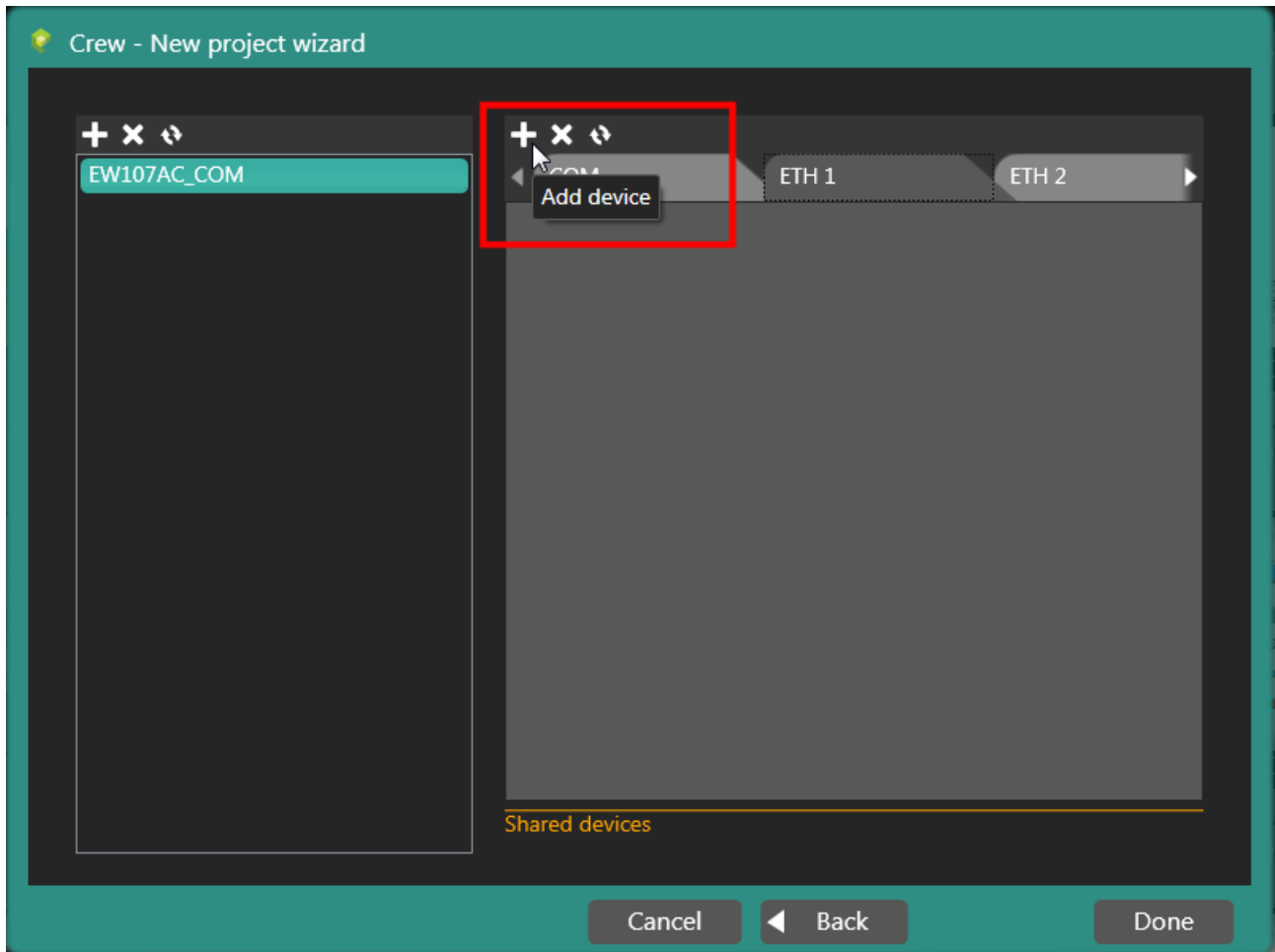
# CREW Manual

## Importing RSLogix Tags

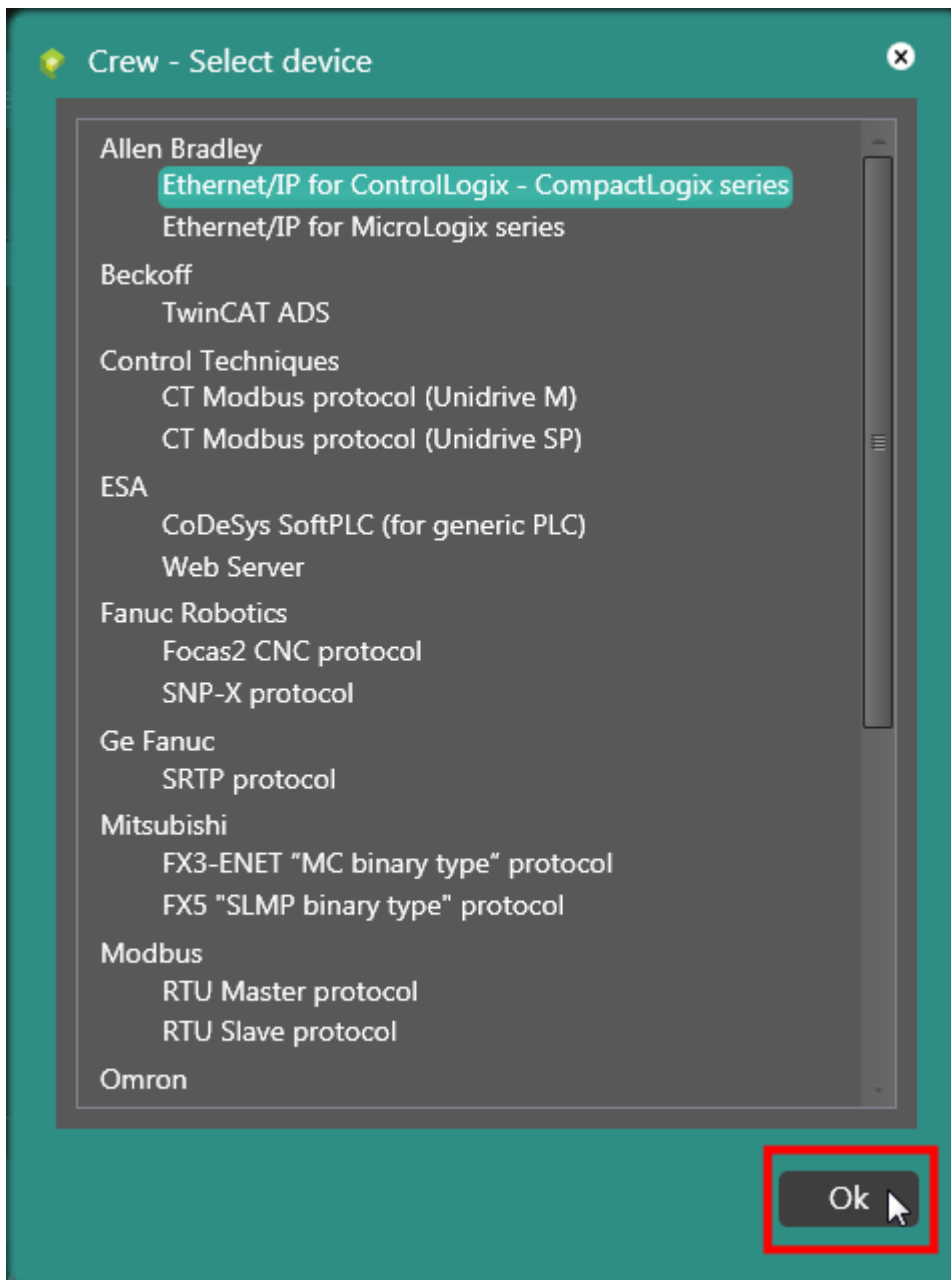




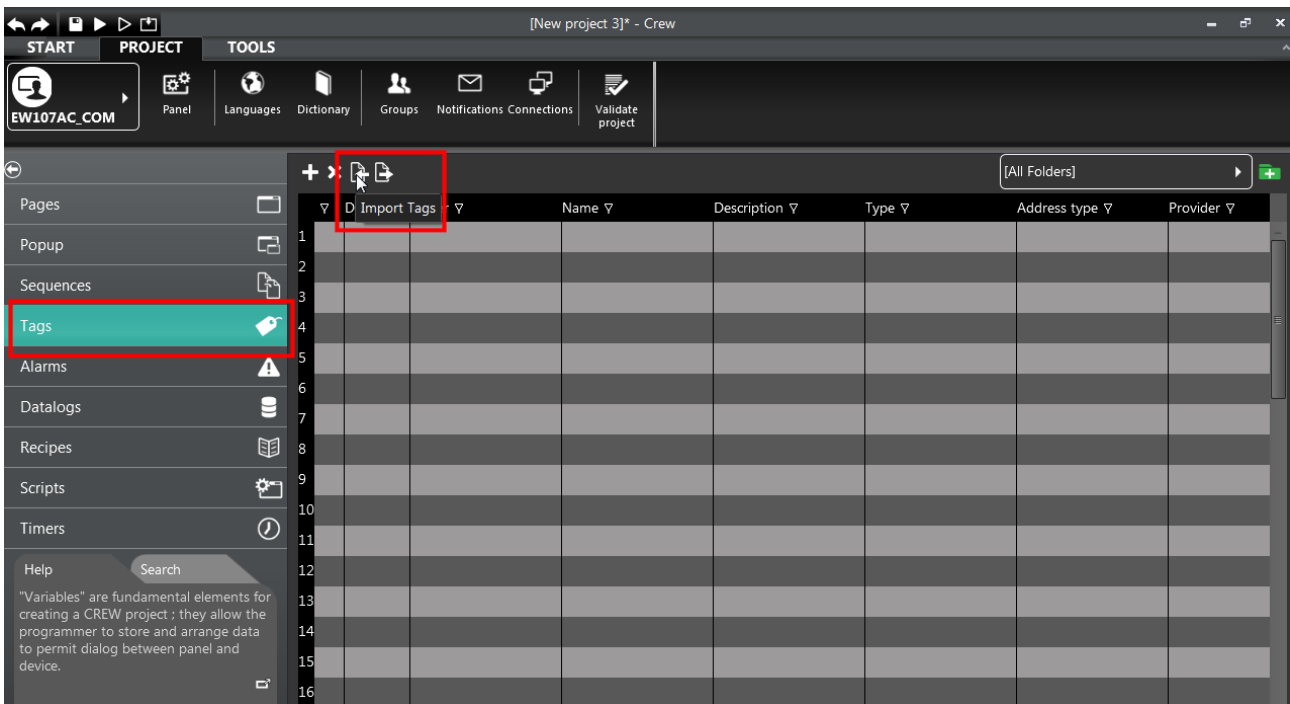
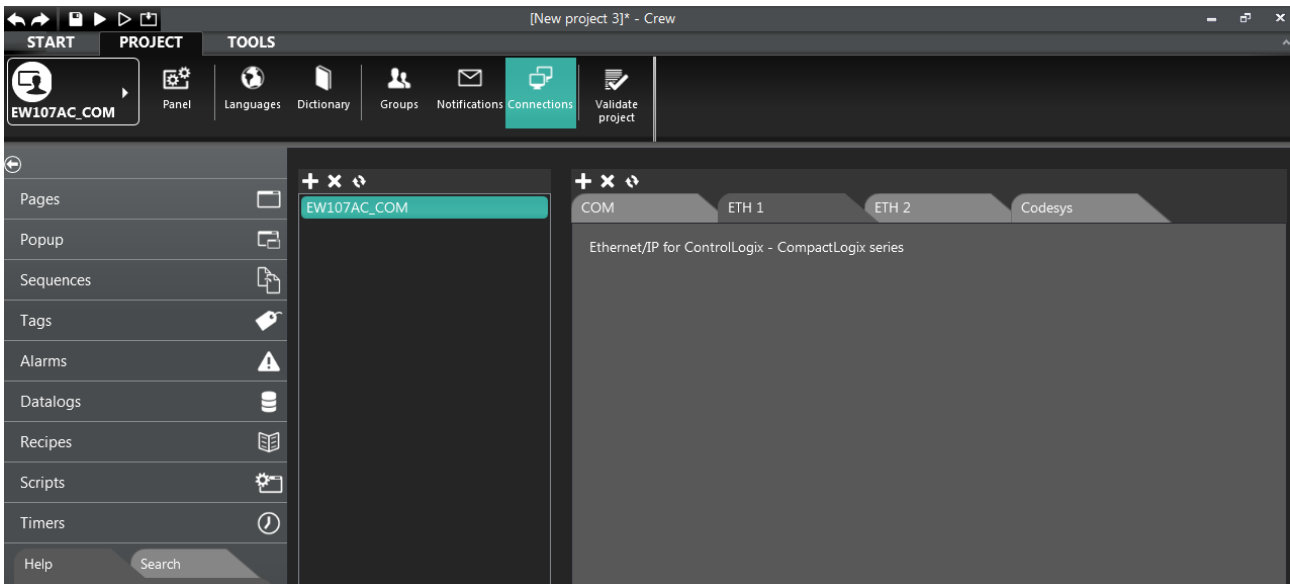
# CREW Manual



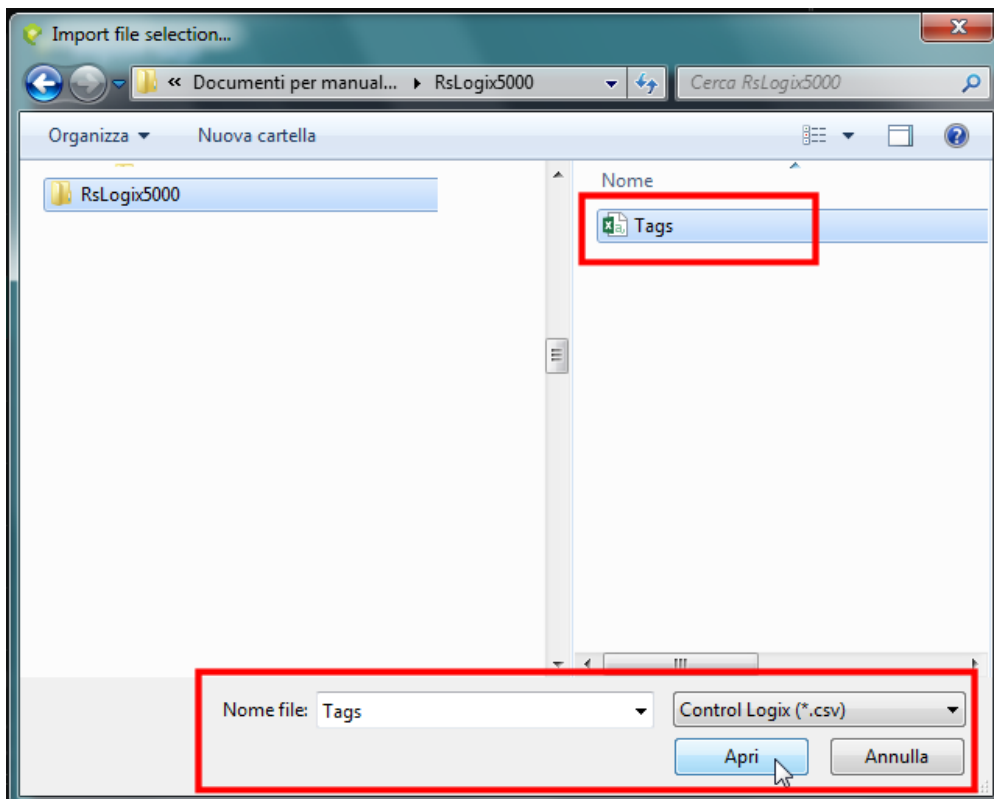
# CREW Manual



# CREW Manual



# CREW Manual



# CREW Manual

Import tags

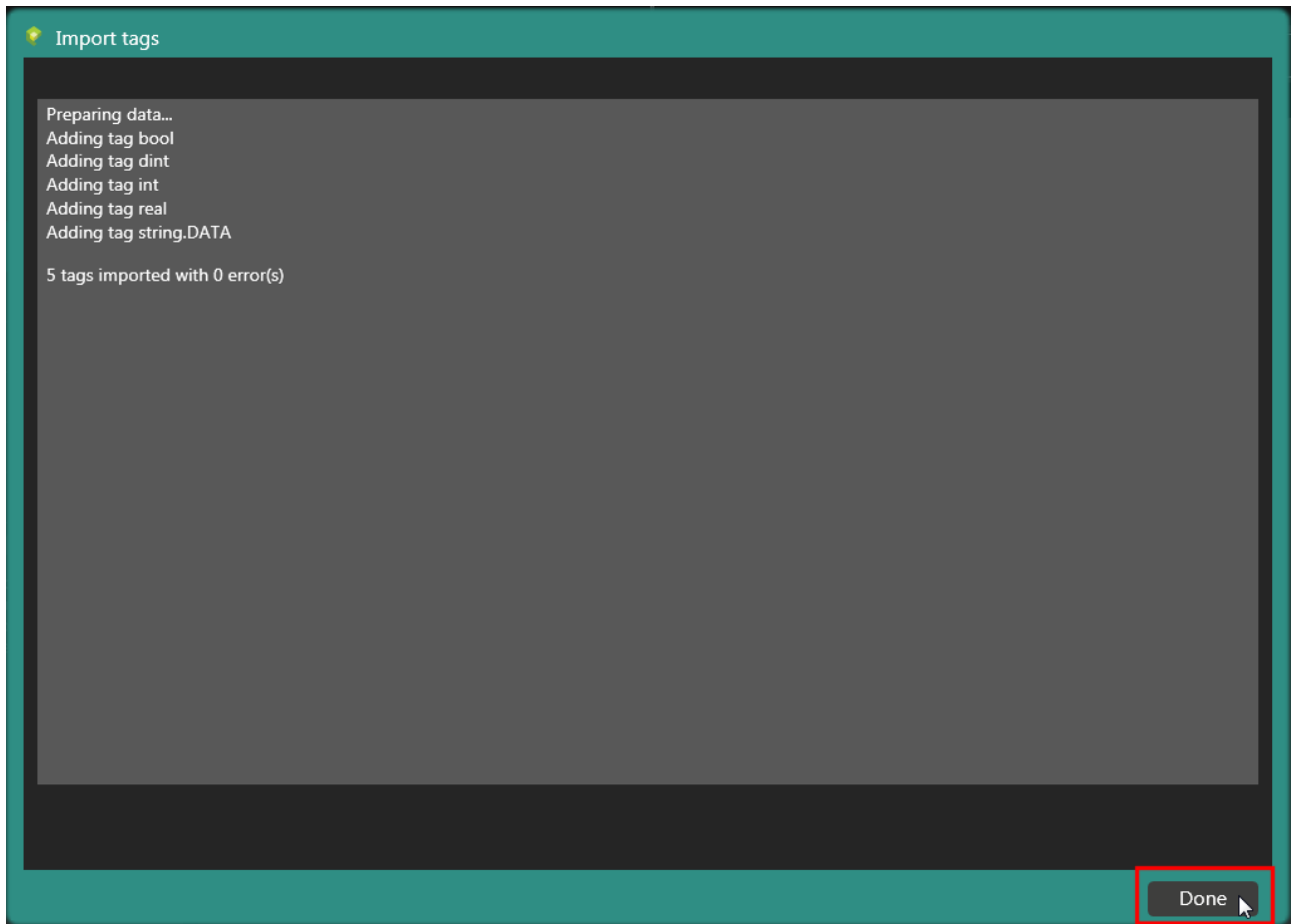
Tags to import

Name	Folder	TagType	AddressType	Description
<input checked="" type="checkbox"/> bool		Boolean	Device	
<input checked="" type="checkbox"/> dint		Long	Device	
<input checked="" type="checkbox"/> int		Integer	Device	
<input checked="" type="checkbox"/> real		Real	Device	
<input checked="" type="checkbox"/> string.DATA		String	Device	

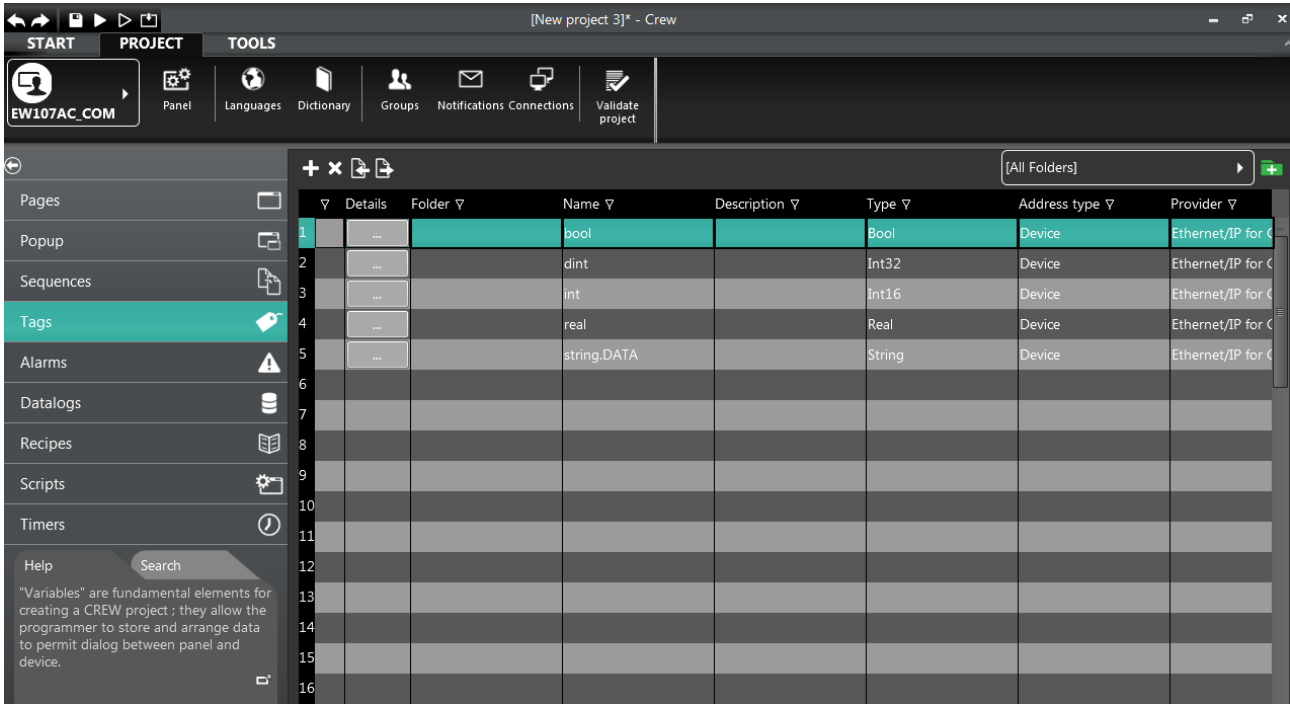
Select all   Unselect all   Destination device: Ethernet/IP for ControlLogix - CompactLogix ▶    Override the existing objects

Cancel   Next ▶

# CREW Manual



# CREW Manual



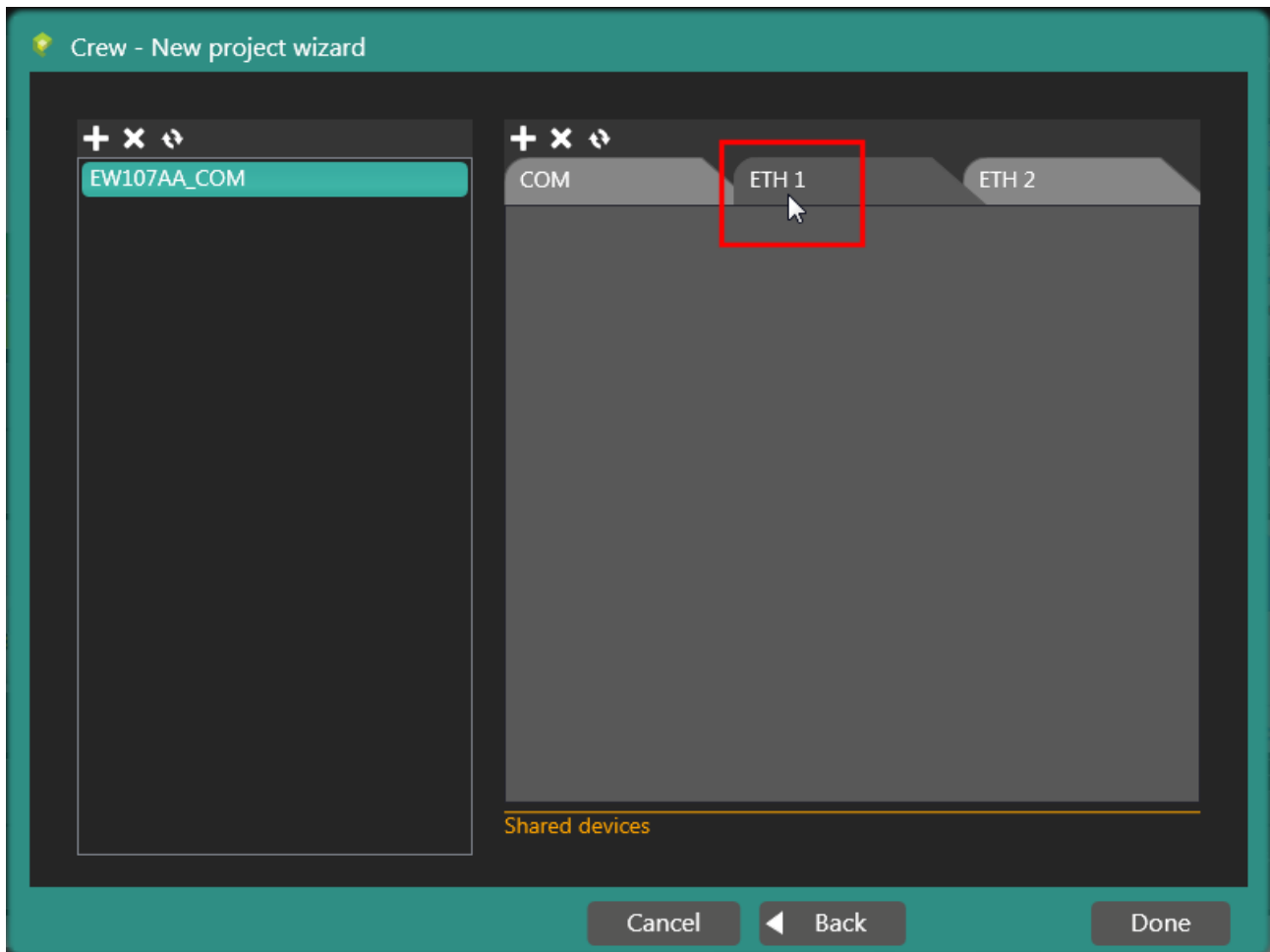
The screenshot shows the ESNA AUTOMATION software interface. The top menu bar includes 'START', 'PROJECT', and 'TOOLS'. Below the menu is a toolbar with icons for 'Panel', 'Languages', 'Dictionary', 'Groups', 'Notifications', 'Connections', and 'Validate project'. The main workspace is a table with columns: 'Details', 'Folder', 'Name', 'Description', 'Type', 'Address type', and 'Provider'. The table contains five rows of data, with the first row highlighted in green.

	Details	Folder	Name	Description	Type	Address type	Provider
1	...		bool		Bool	Device	Ethernet/IP for C
2	...		dint		Int32	Device	Ethernet/IP for C
3	...		int		Int16	Device	Ethernet/IP for C
4	...		real		Real	Device	Ethernet/IP for C
5	...		string.DATA		String	Device	Ethernet/IP for C
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

The left sidebar contains a list of project elements: Pages, Popup, Sequences, Tags (highlighted), Alarms, Datalogs, Recipes, Scripts, and Timers. Below this is a 'Help' section with a search bar and a text block: "Variables" are fundamental elements for creating a CREW project ; they allow the programmer to store and arrange data to permit dialog between panel and device.

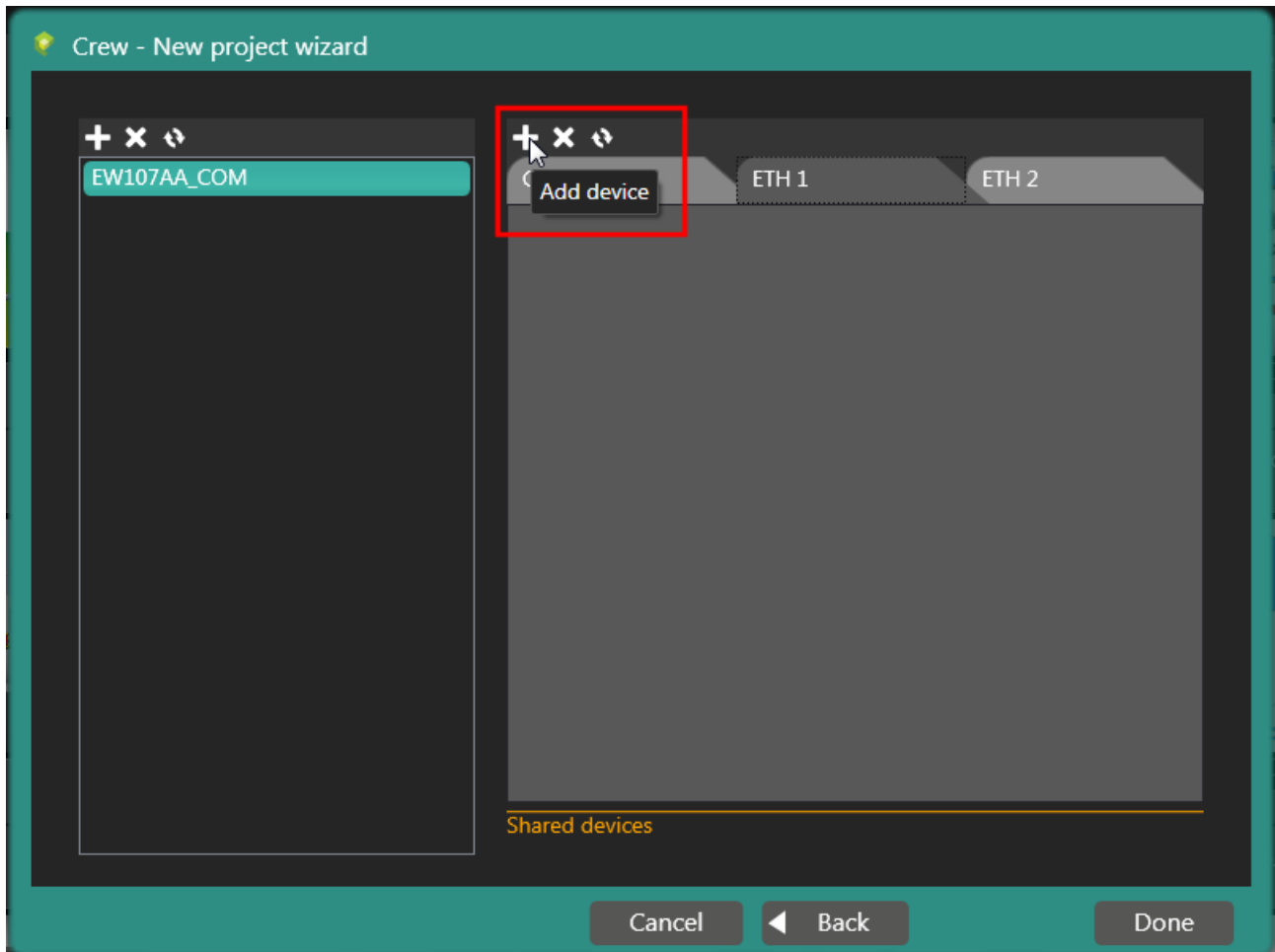
# CREW Manual

## CODESYS SoftPLC - For generic PLCs

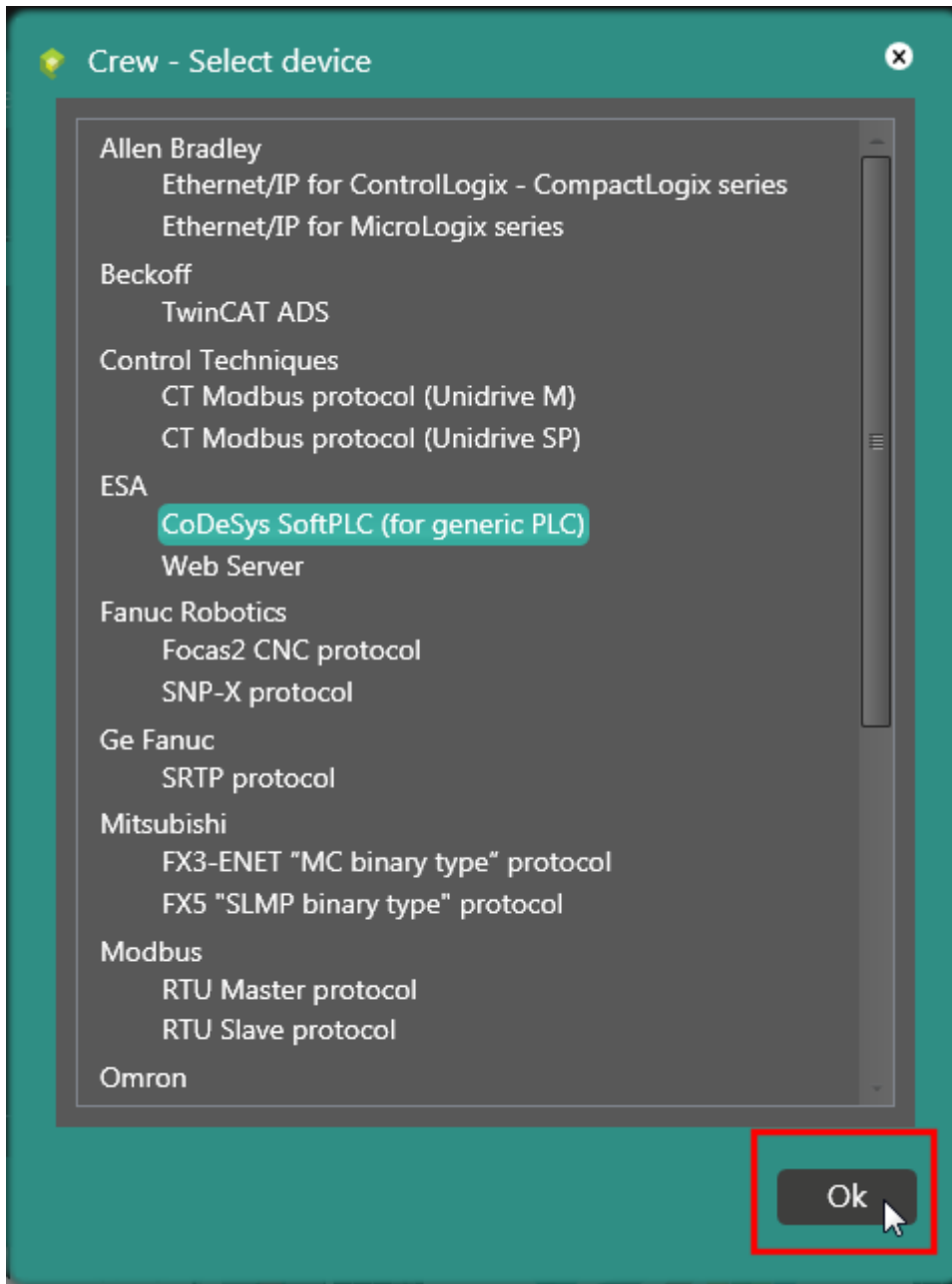




# CREW Manual



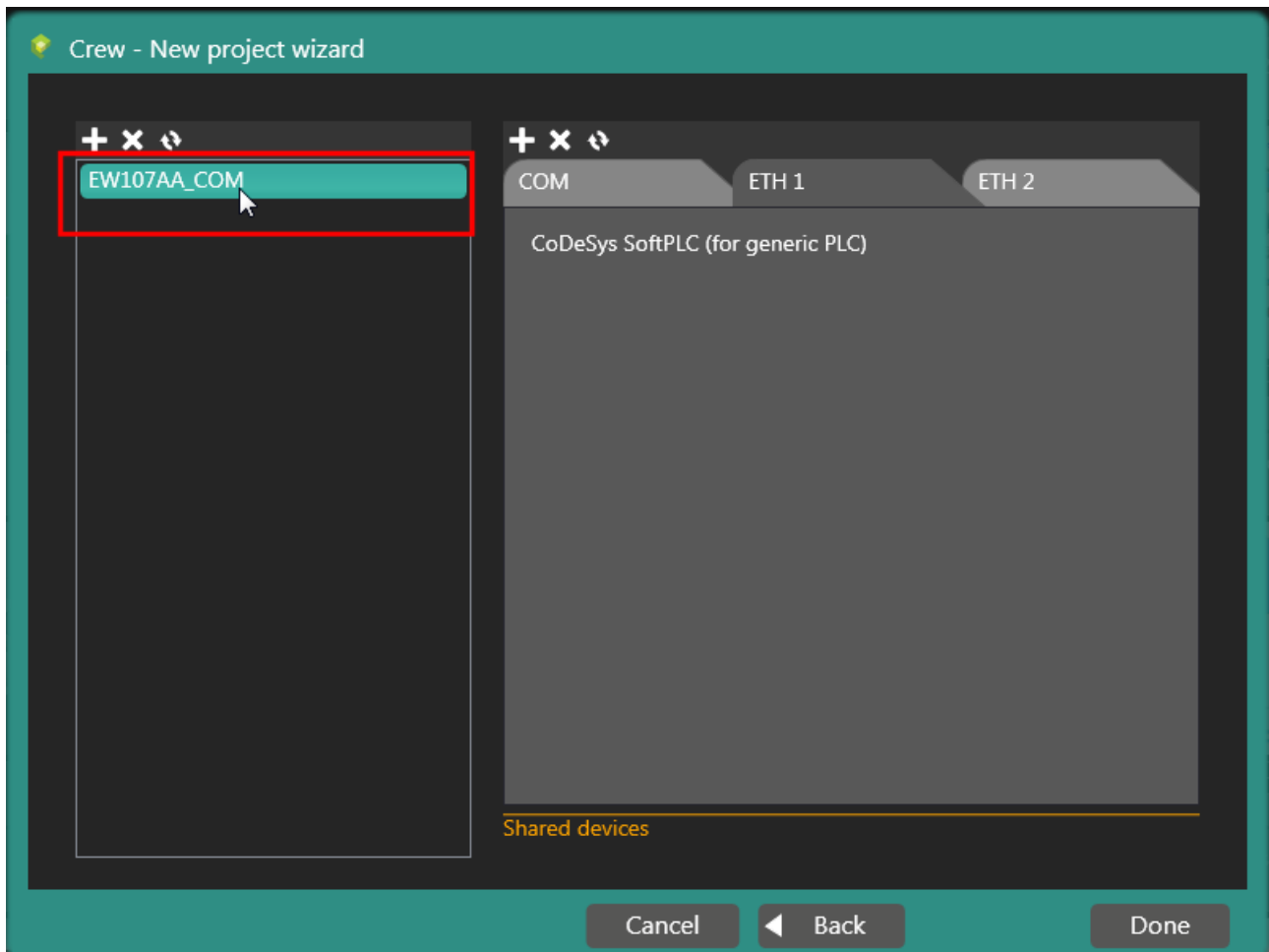
# CREW Manual



# CREW Manual

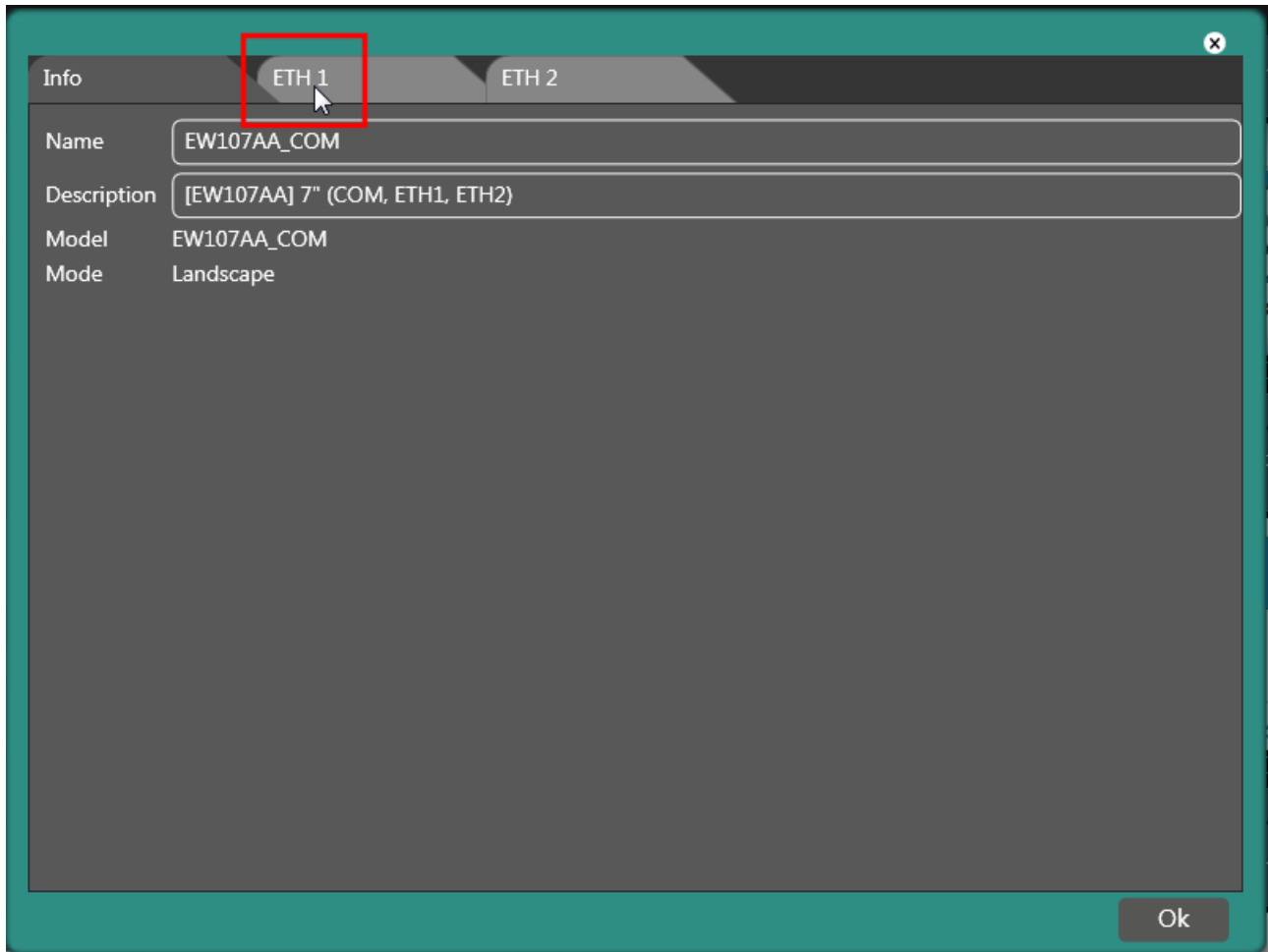
Settings on EW side:

Double click on the name of the terminal.

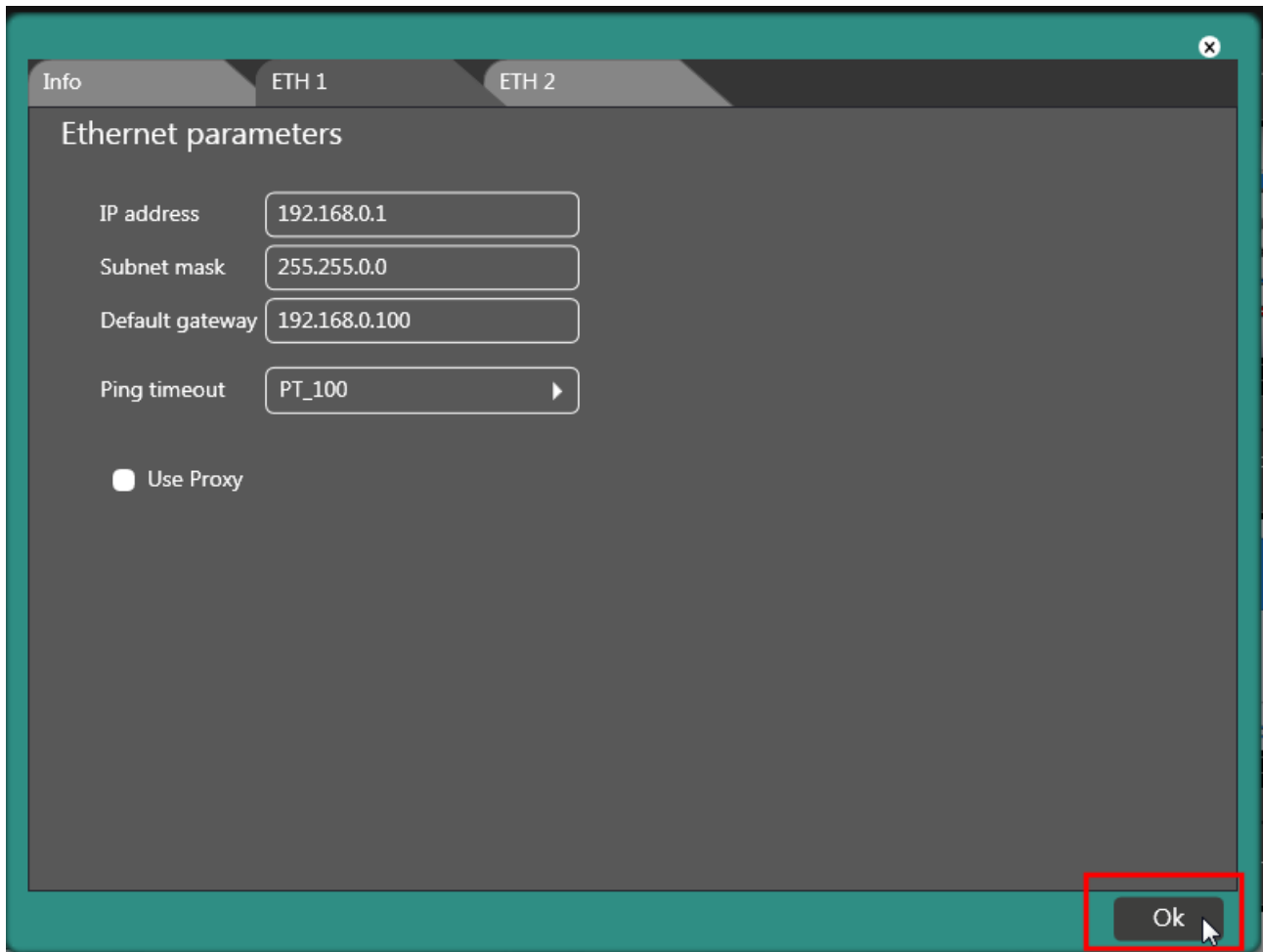


# CREW Manual

Ethernet Parameters:



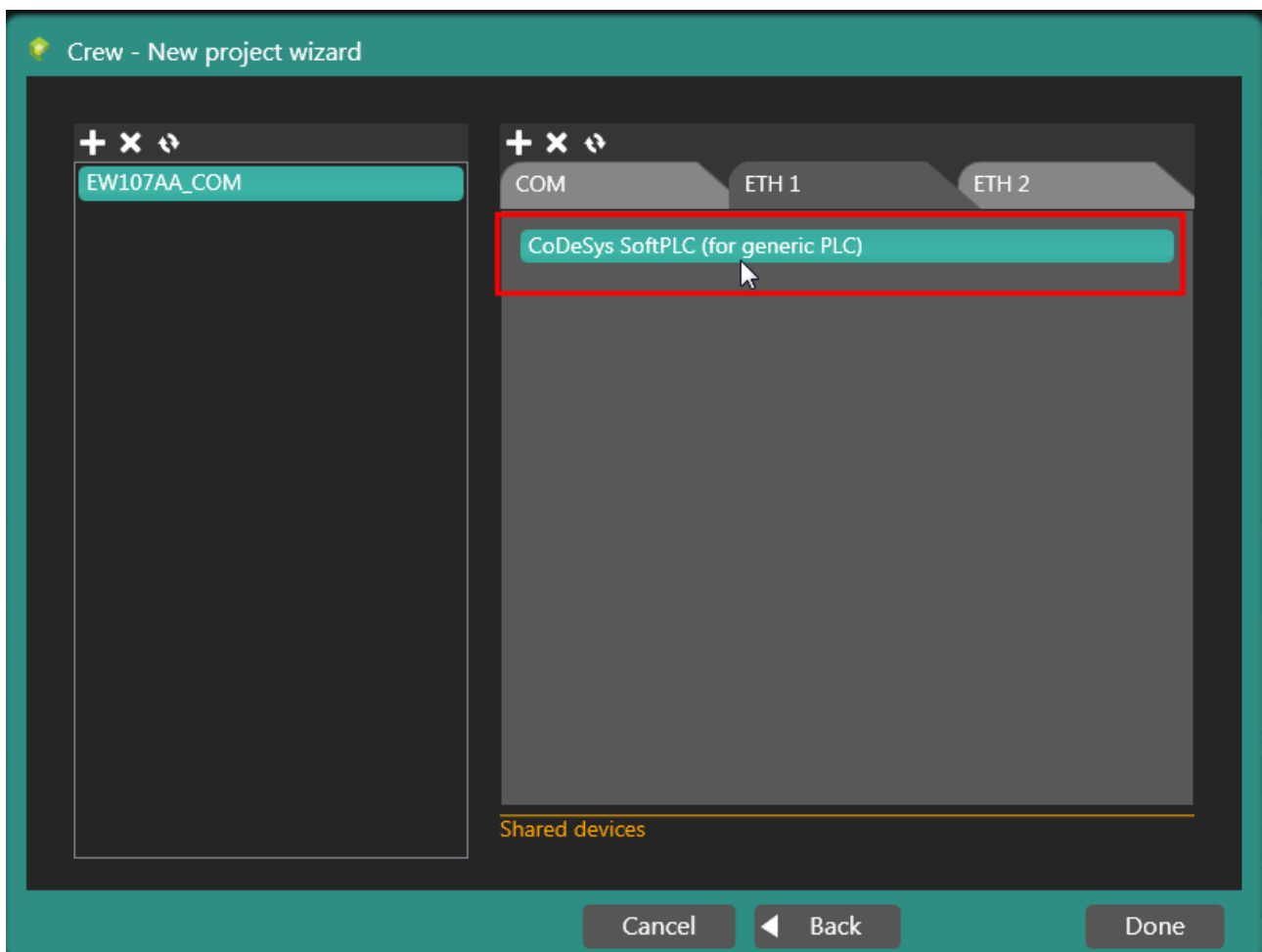
# CREW Manual



# CREW Manual

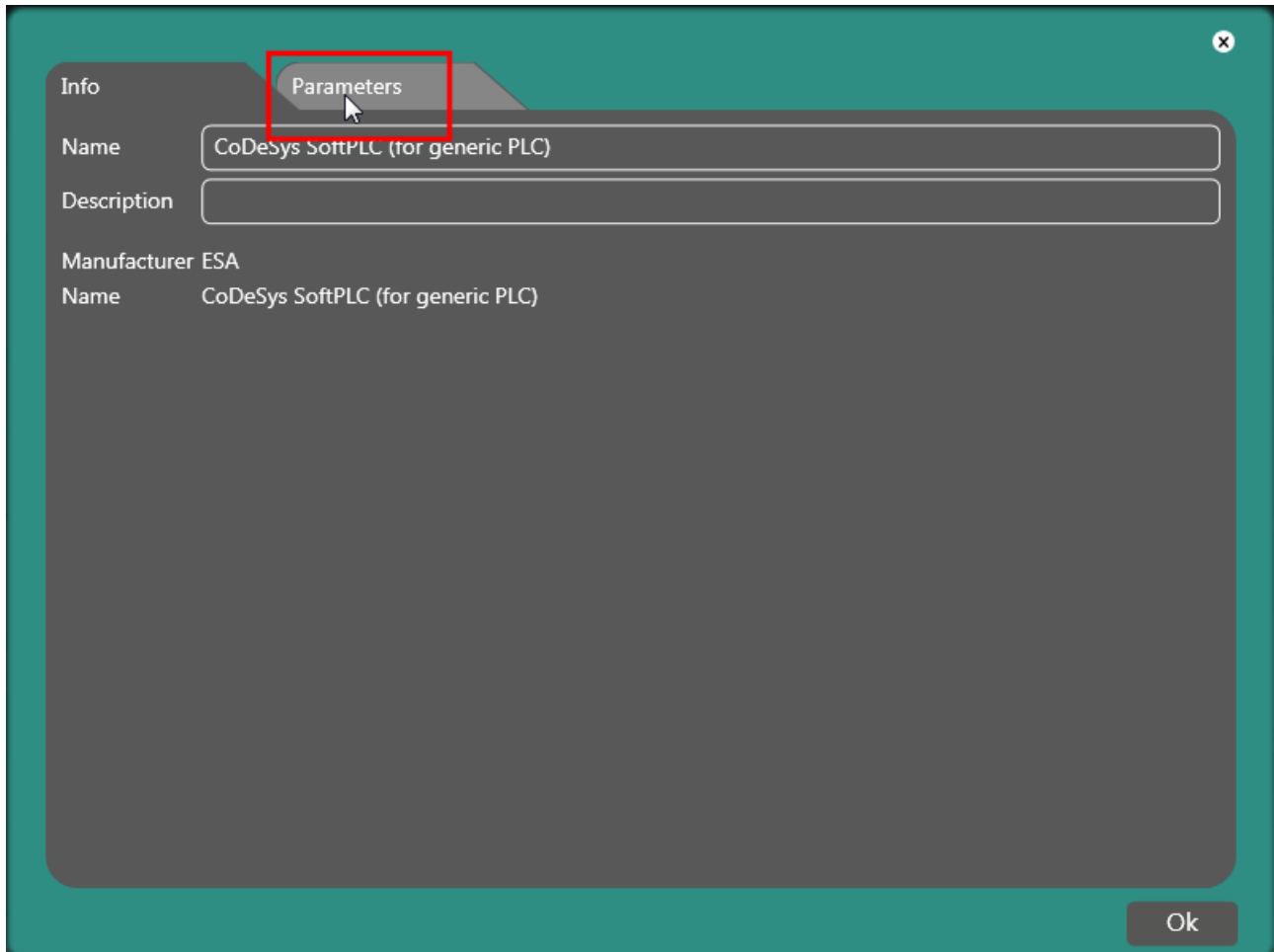
Settings on device side:

Double click on the name of the device.



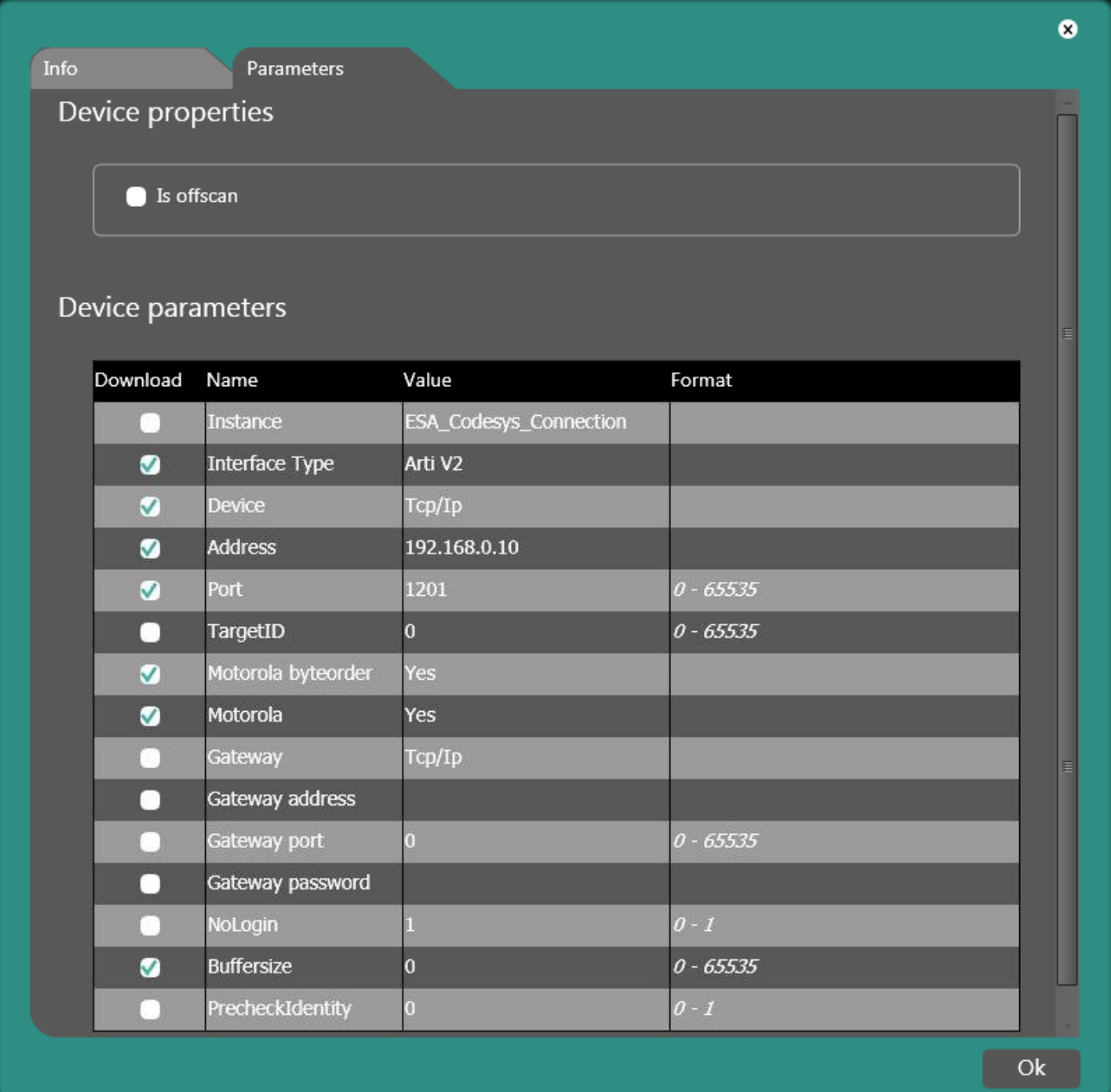
# CREW Manual

Device parameters:



# CREW Manual

Generic CODESYS parameters with ABB PM564:



Info Parameters

Device properties

Is offscan

Device parameters

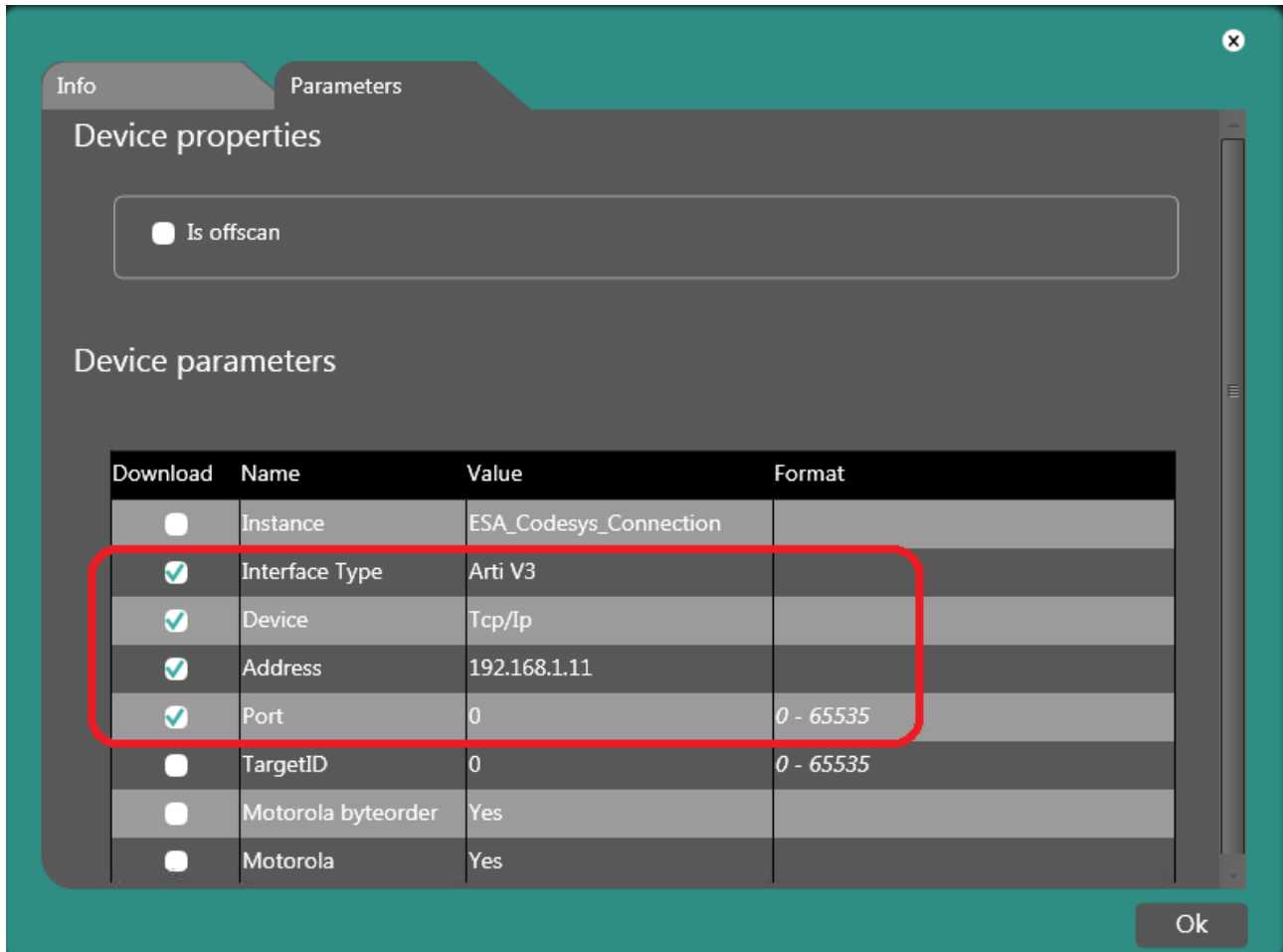
Download	Name	Value	Format
<input type="checkbox"/>	Instance	ESA_Codesys_Connection	
<input checked="" type="checkbox"/>	Interface Type	Arti V2	
<input checked="" type="checkbox"/>	Device	Tcp/Ip	
<input checked="" type="checkbox"/>	Address	192.168.0.10	
<input checked="" type="checkbox"/>	Port	1201	0 - 65535
<input type="checkbox"/>	TargetID	0	0 - 65535
<input checked="" type="checkbox"/>	Motorola byteorder	Yes	
<input checked="" type="checkbox"/>	Motorola	Yes	
<input type="checkbox"/>	Gateway	Tcp/Ip	
<input type="checkbox"/>	Gateway address		
<input type="checkbox"/>	Gateway port	0	0 - 65535
<input type="checkbox"/>	Gateway password		
<input type="checkbox"/>	NoLogin	1	0 - 1
<input checked="" type="checkbox"/>	Buffersize	0	0 - 65535
<input type="checkbox"/>	PrecheckIdentity	0	0 - 1

Ok



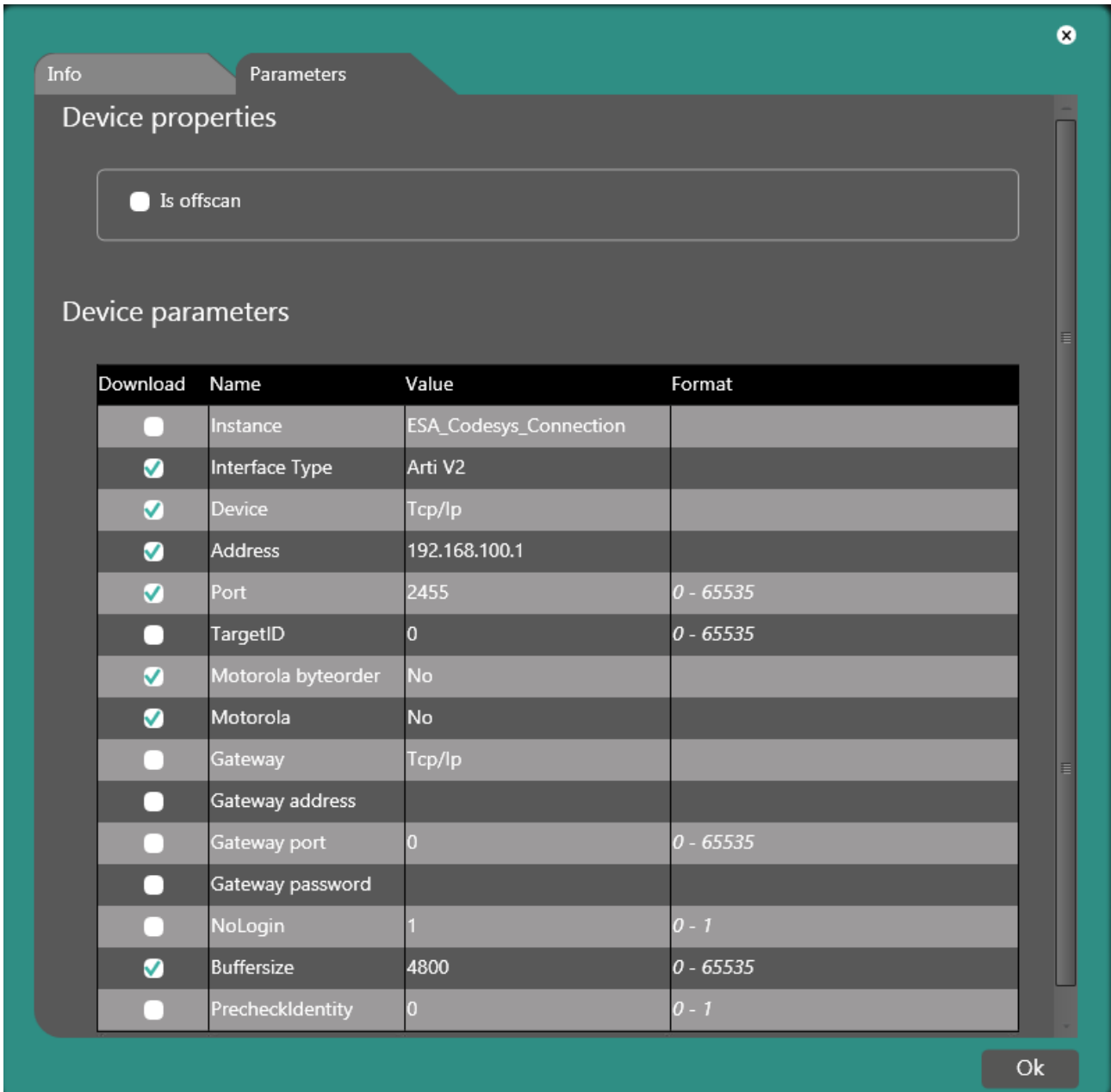
# CREW Manual

Generic CODESYS parameters with BOSCH - REXROTH L45:



# CREW Manual

Generic CODESYS parameters with WAGO 750-841:



The screenshot shows a software window with two tabs: 'Info' and 'Parameters'. The 'Parameters' tab is active and displays 'Device properties' and 'Device parameters'.

**Device properties**

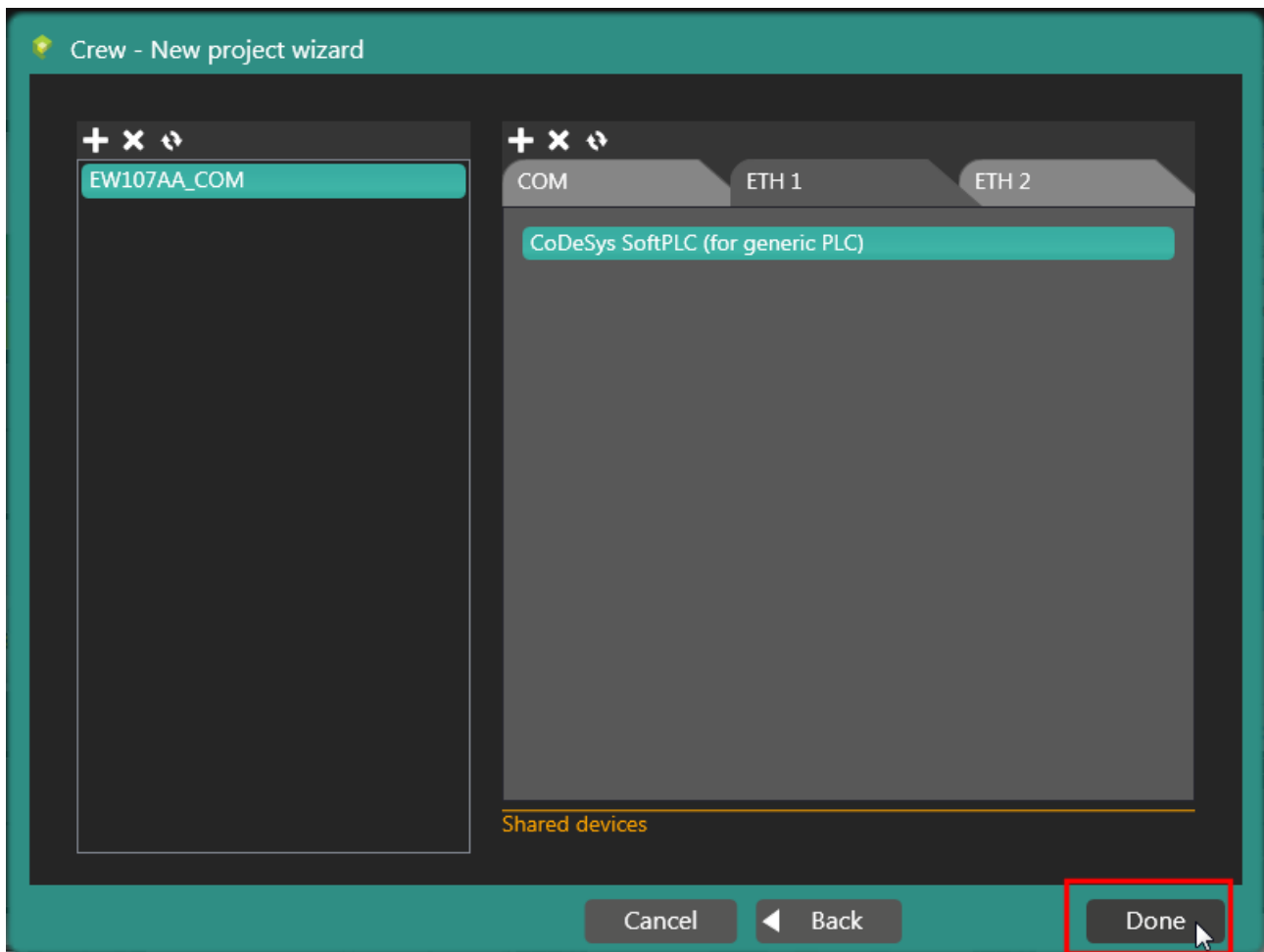
- Is offscan

**Device parameters**

Download	Name	Value	Format
<input type="checkbox"/>	Instance	ESA_Codesys_Connection	
<input checked="" type="checkbox"/>	Interface Type	Arti V2	
<input checked="" type="checkbox"/>	Device	Tcp/lp	
<input checked="" type="checkbox"/>	Address	192.168.100.1	
<input checked="" type="checkbox"/>	Port	2455	0 - 65535
<input type="checkbox"/>	TargetID	0	0 - 65535
<input checked="" type="checkbox"/>	Motorola byteorder	No	
<input checked="" type="checkbox"/>	Motorola	No	
<input type="checkbox"/>	Gateway	Tcp/lp	
<input type="checkbox"/>	Gateway address		
<input type="checkbox"/>	Gateway port	0	0 - 65535
<input type="checkbox"/>	Gateway password		
<input type="checkbox"/>	NoLogin	1	0 - 1
<input checked="" type="checkbox"/>	Buffersize	4800	0 - 65535
<input type="checkbox"/>	PrecheckIdentity	0	0 - 1

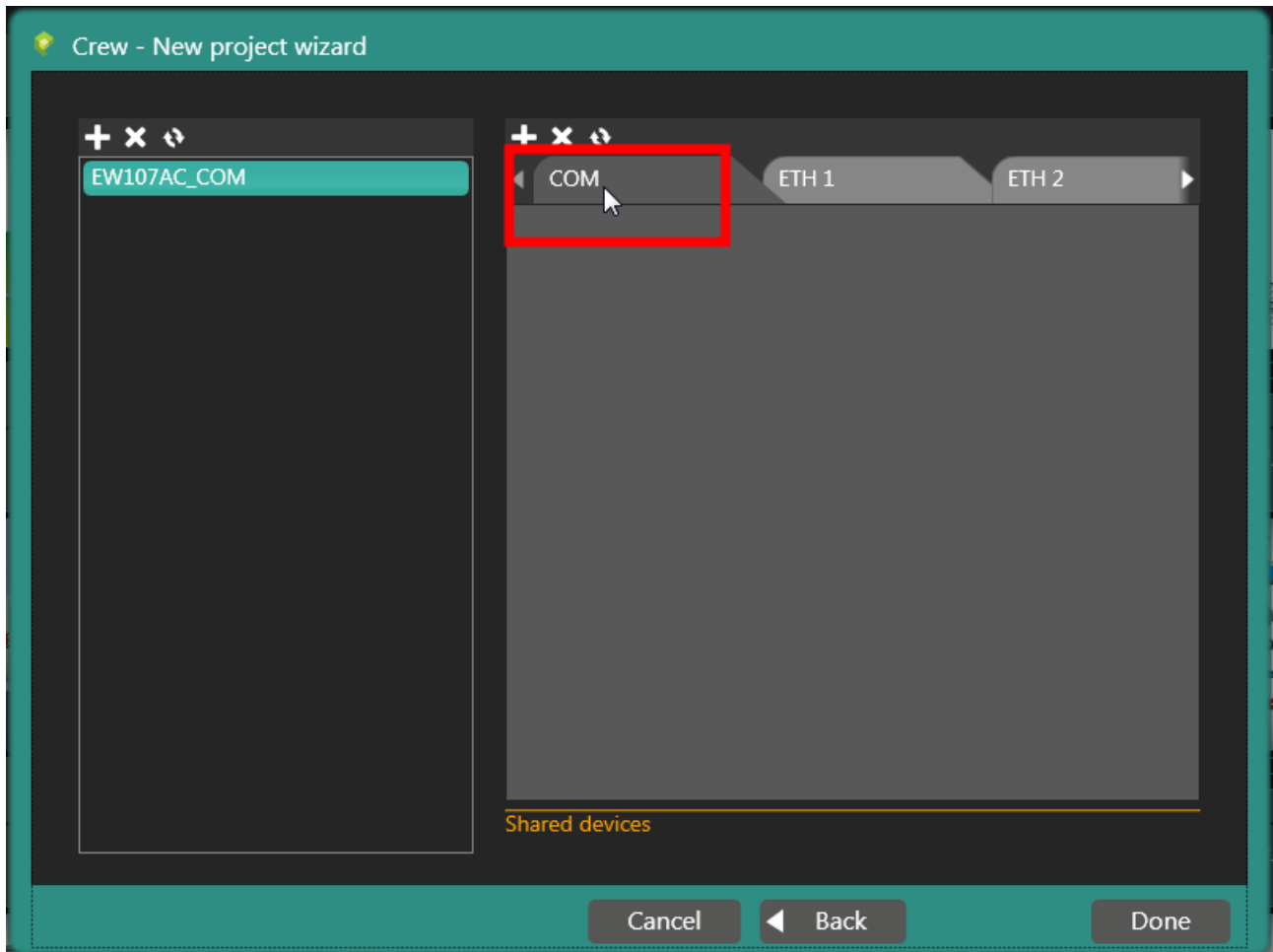
Ok

# CREW Manual

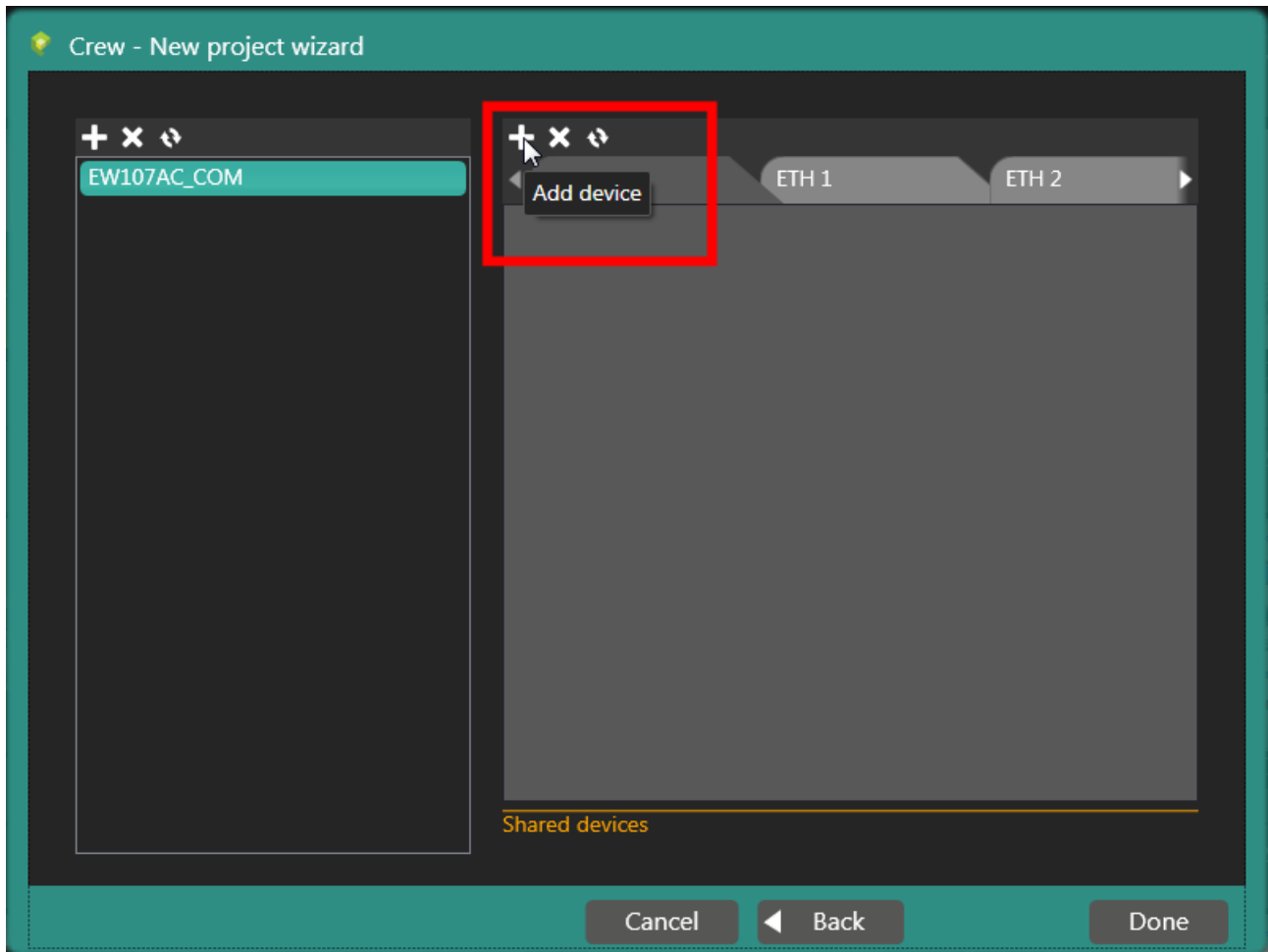


# CREW Manual

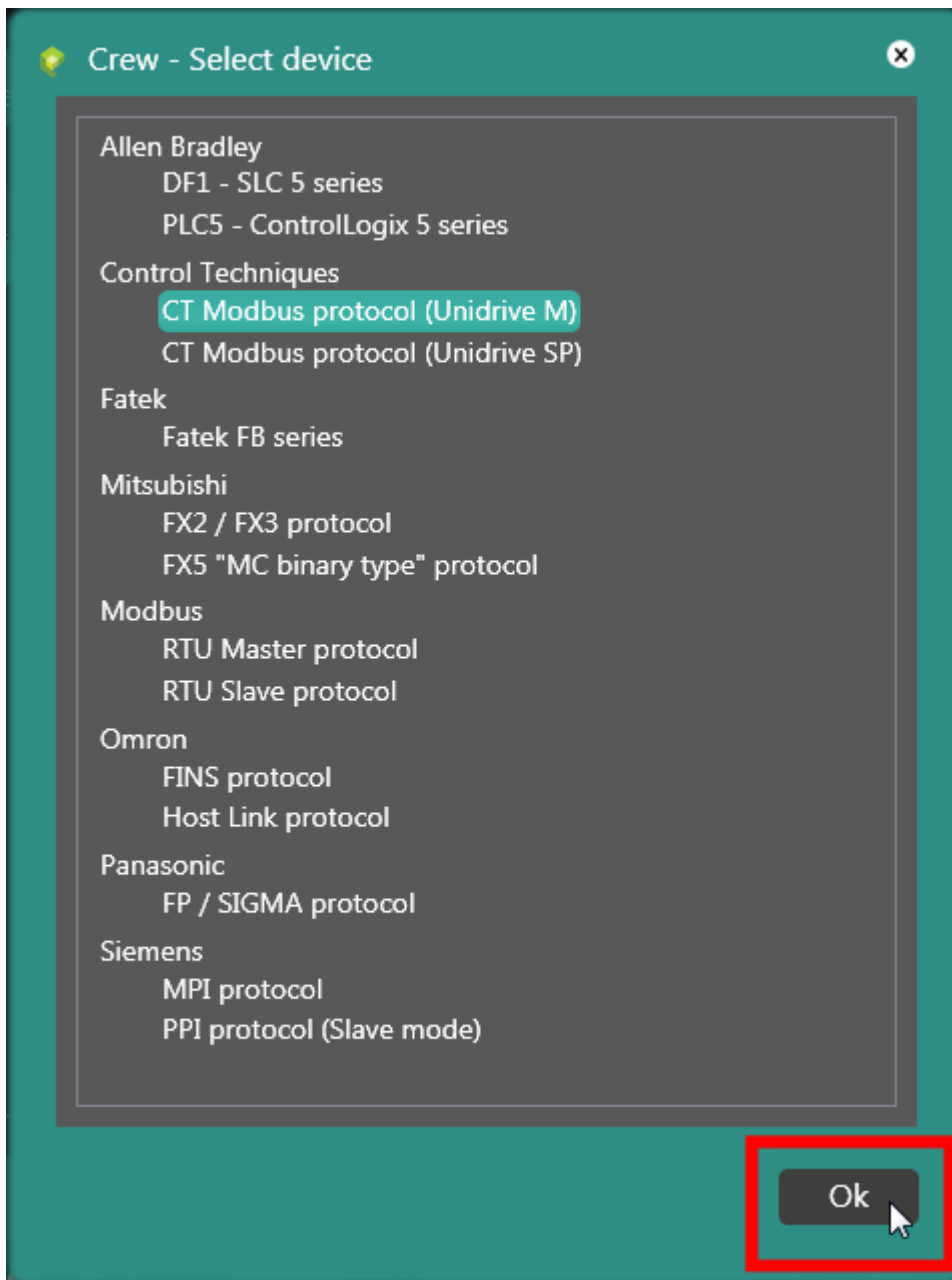
Emerson - Control Techniques (CT Modbus protocol Unidrive M)



# CREW Manual



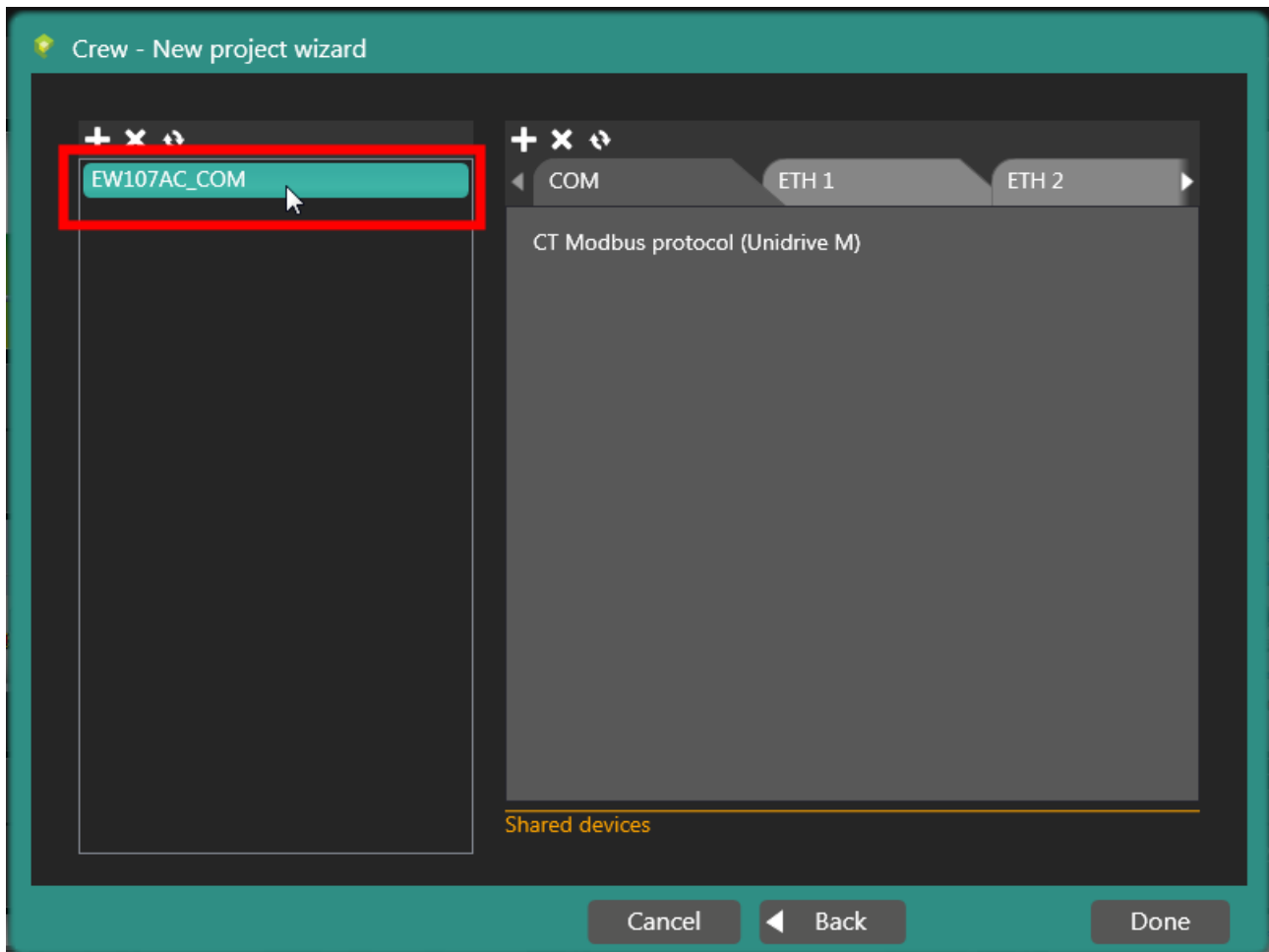
# CREW Manual



# CREW Manual

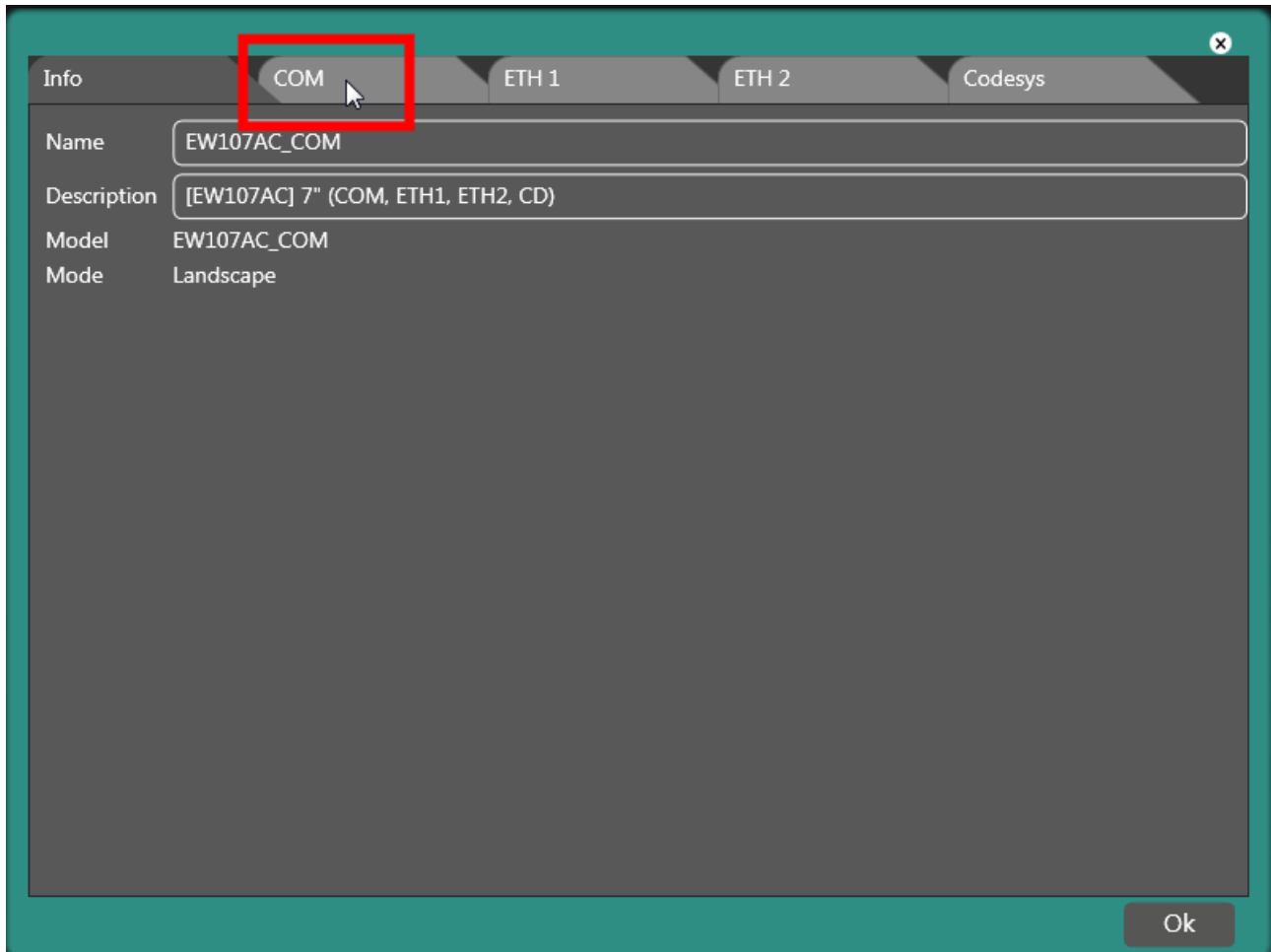
Settings on EW side:

Double click on the name of the terminal.



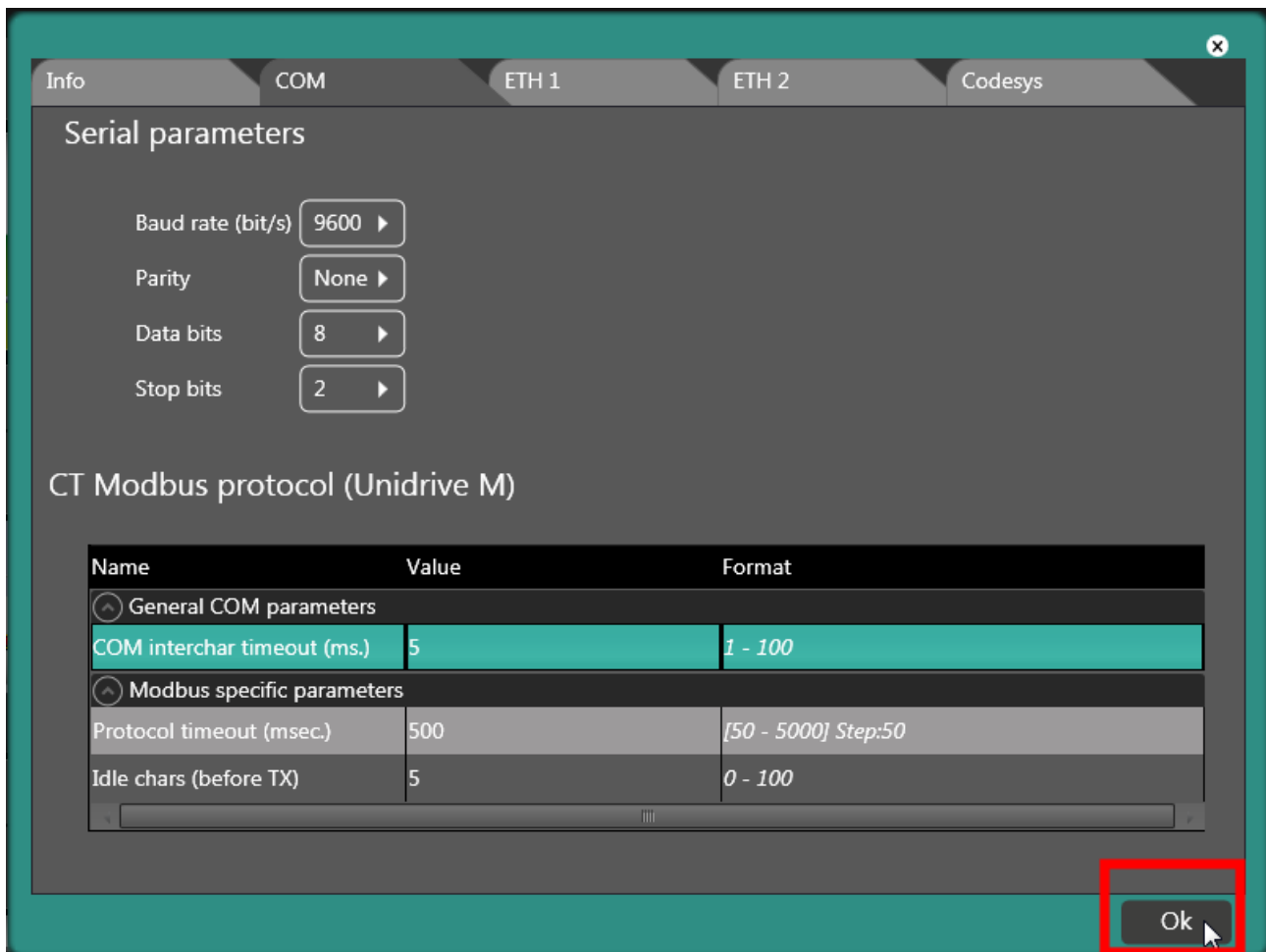
# CREW Manual

Serial parameters:





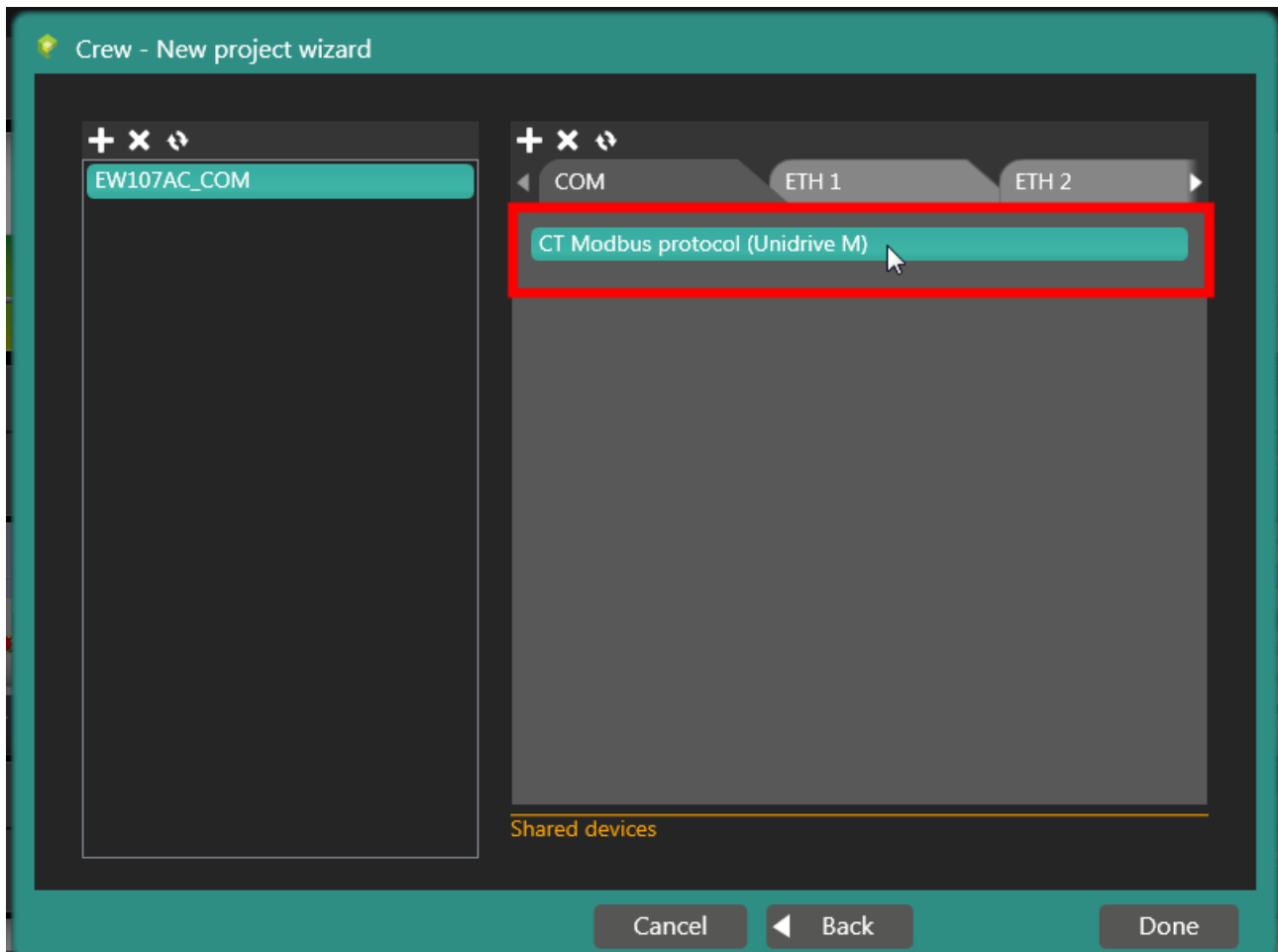
# CREW Manual



# CREW Manual

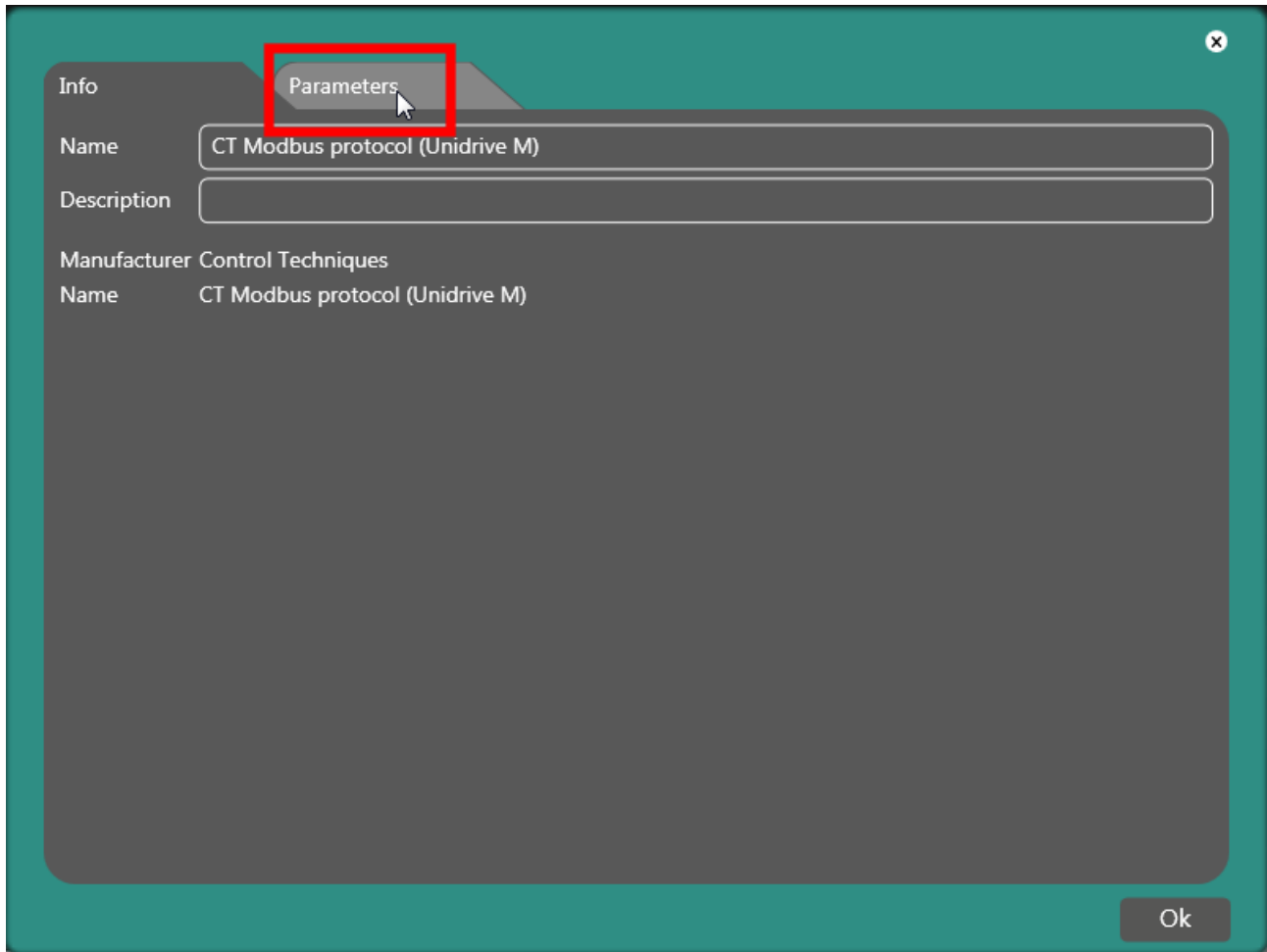
Settings on device side:

Double click on the name of the device.

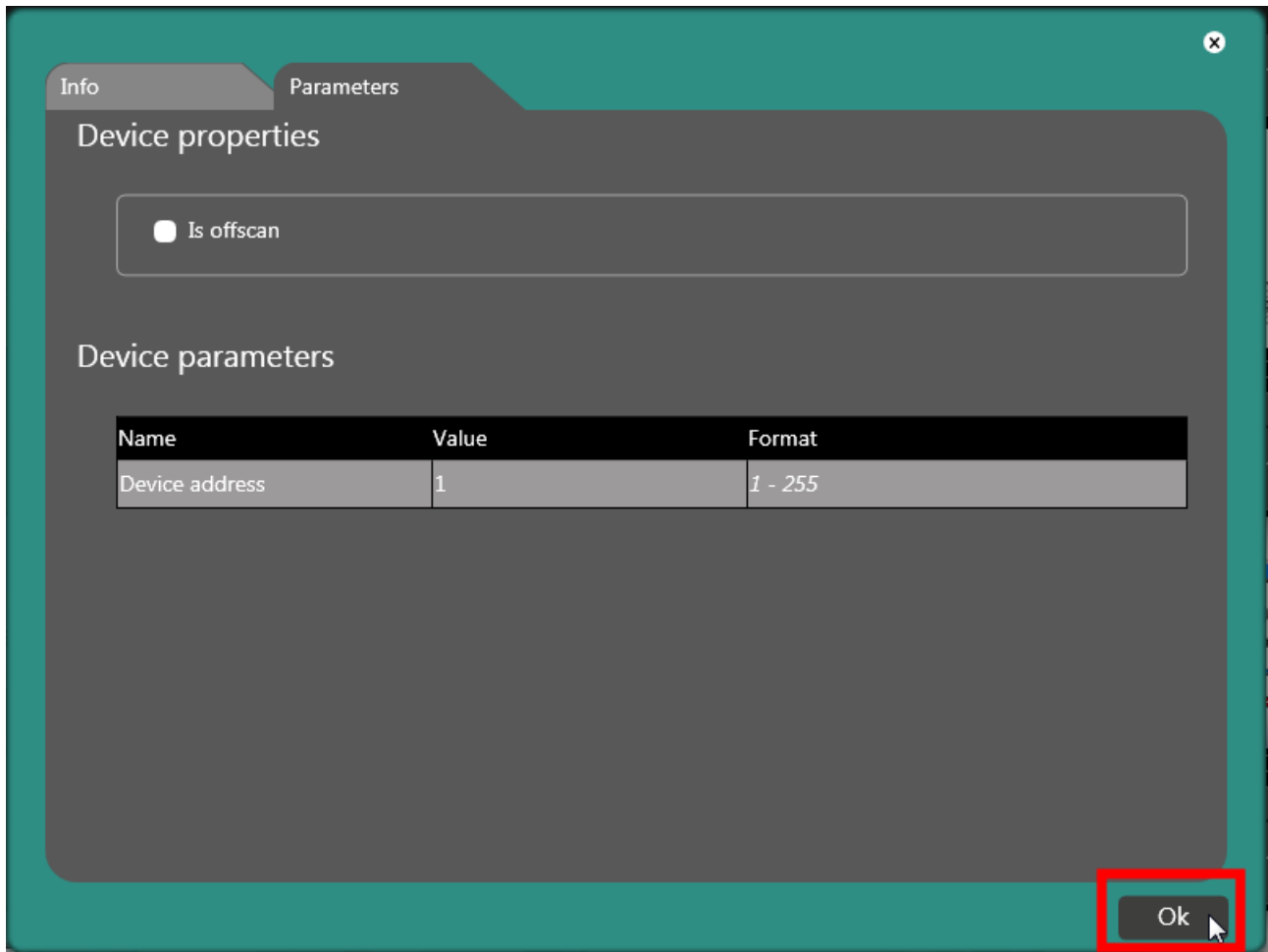


# CREW Manual

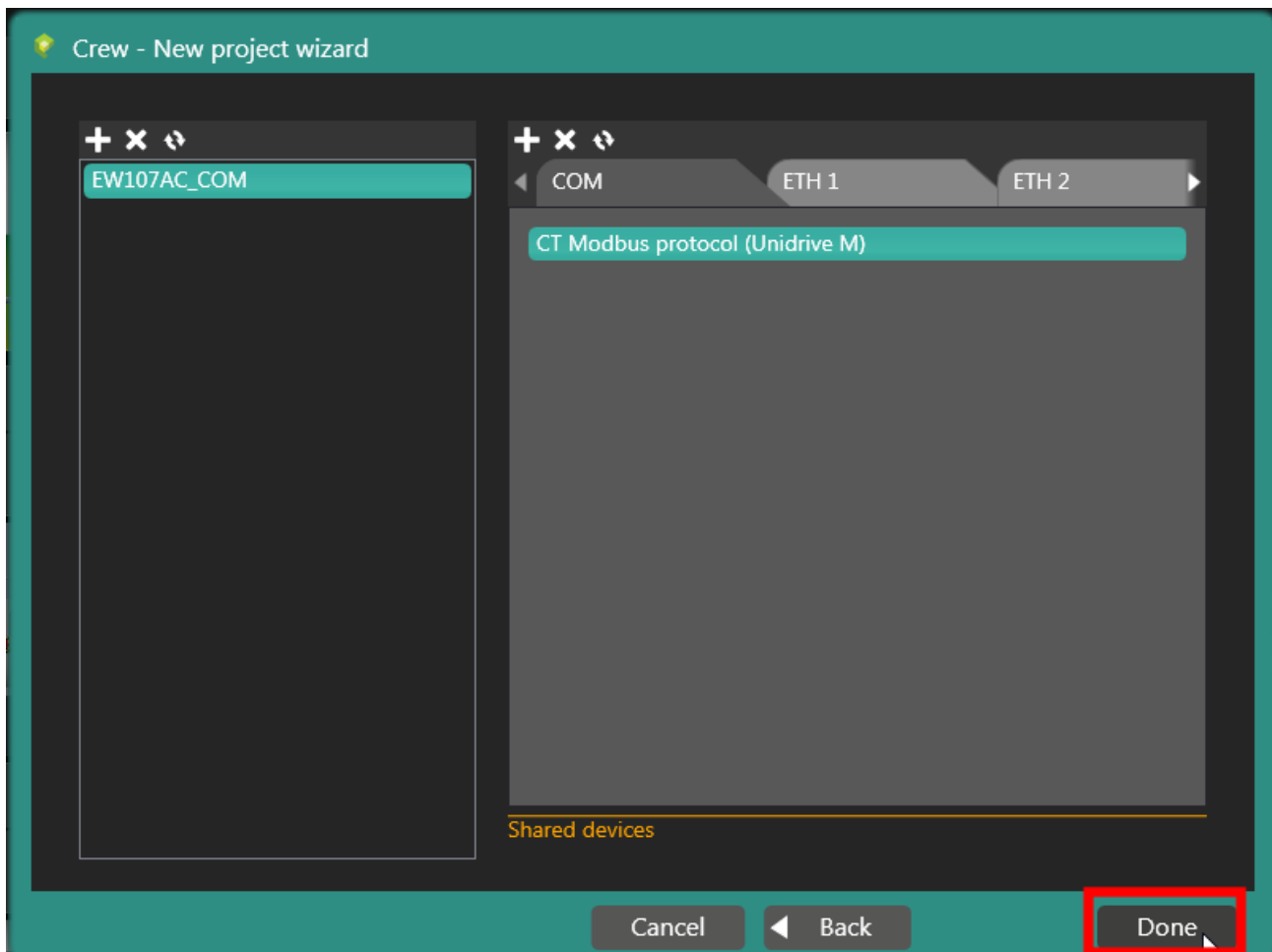
Device parameters:



# CREW Manual

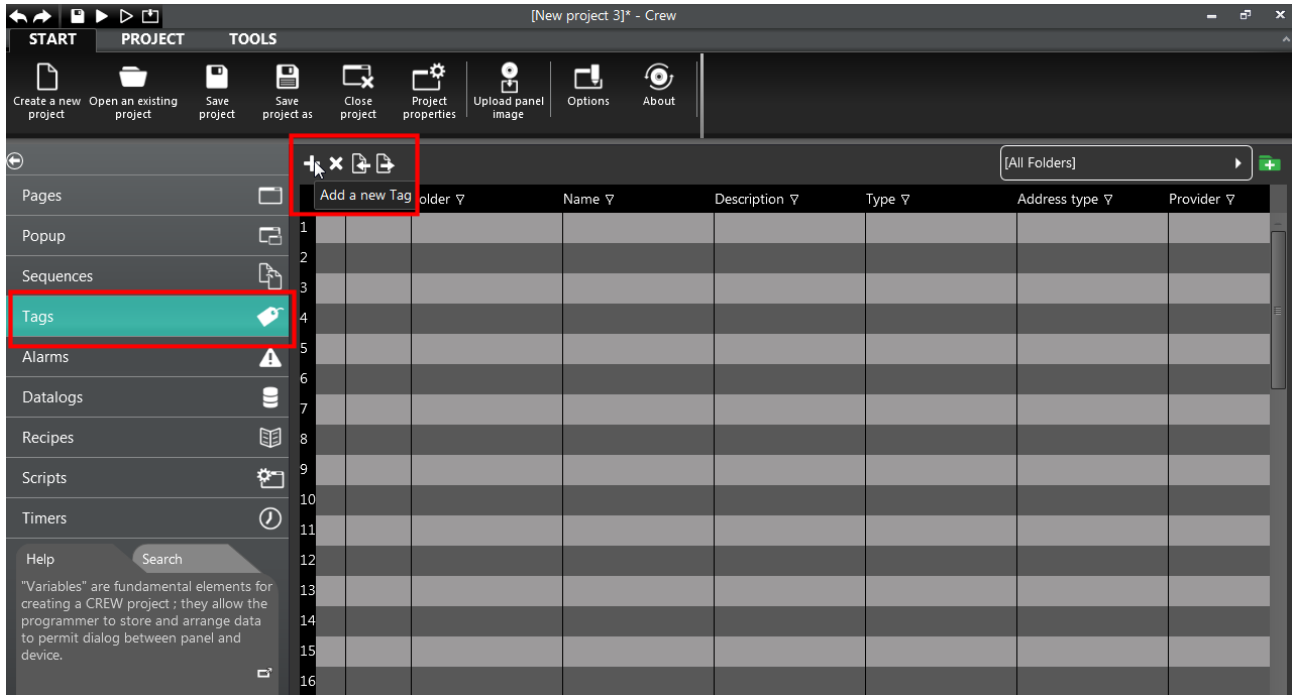


# CREW Manual



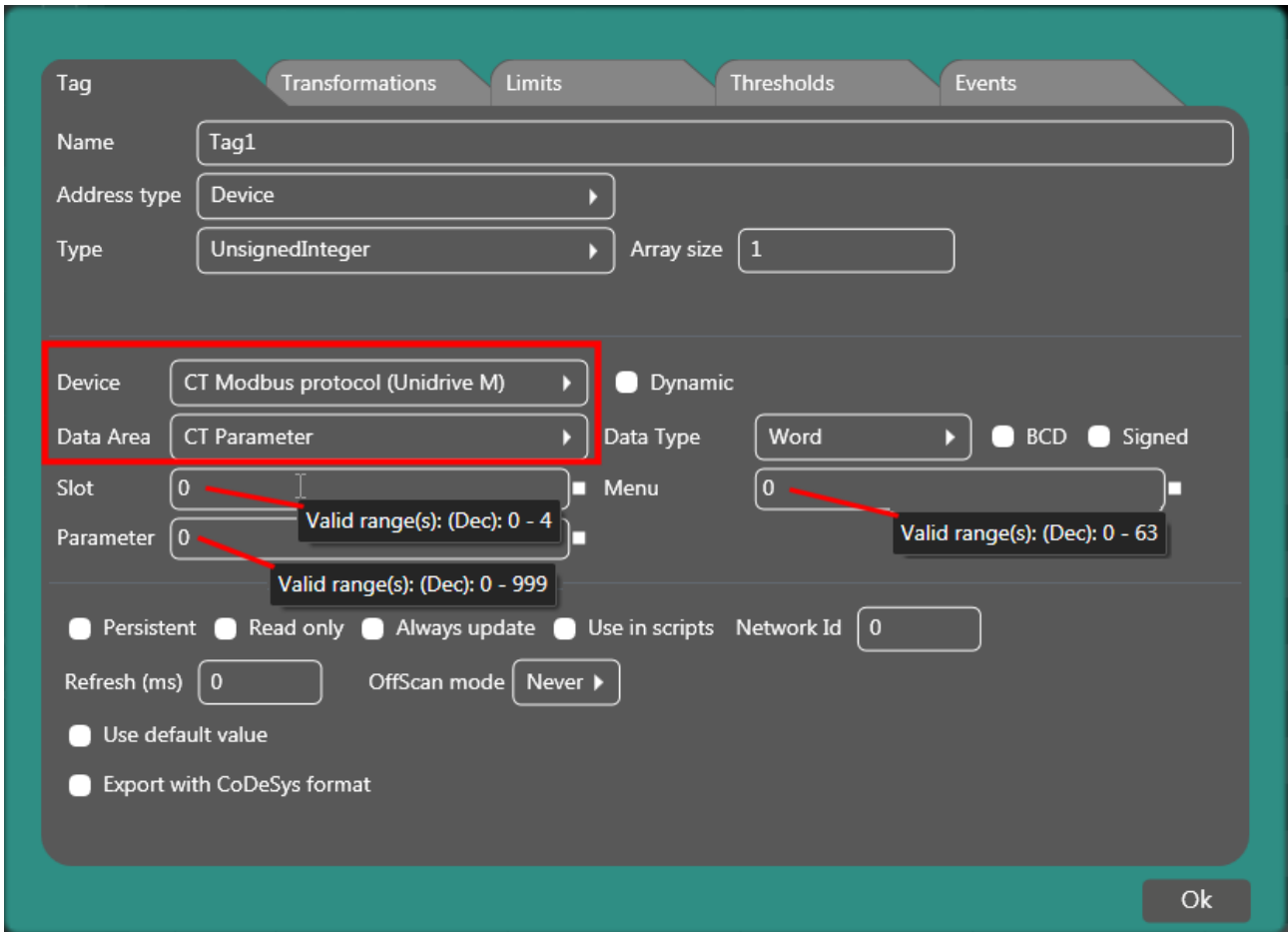
# CREW Manual

## Set up Parameters - Emerson Control Techniques



# CREW Manual

"CT Parameter" Data Area:



Tag

Transformations Limits Thresholds Events

Name Tag1

Address type Device

Type UnsignedInteger Array size 1

Device CT Modbus protocol (Unidrive M)  Dynamic

Data Area CT Parameter Data Type Word  BCD  Signed

Slot 0  Menu 0

Parameter 0

Valid range(s): (Dec): 0 - 4

Valid range(s): (Dec): 0 - 63

Valid range(s): (Dec): 0 - 999

Persistent  Read only  Always update  Use in scripts Network Id 0

Refresh (ms) 0 OffScan mode Never

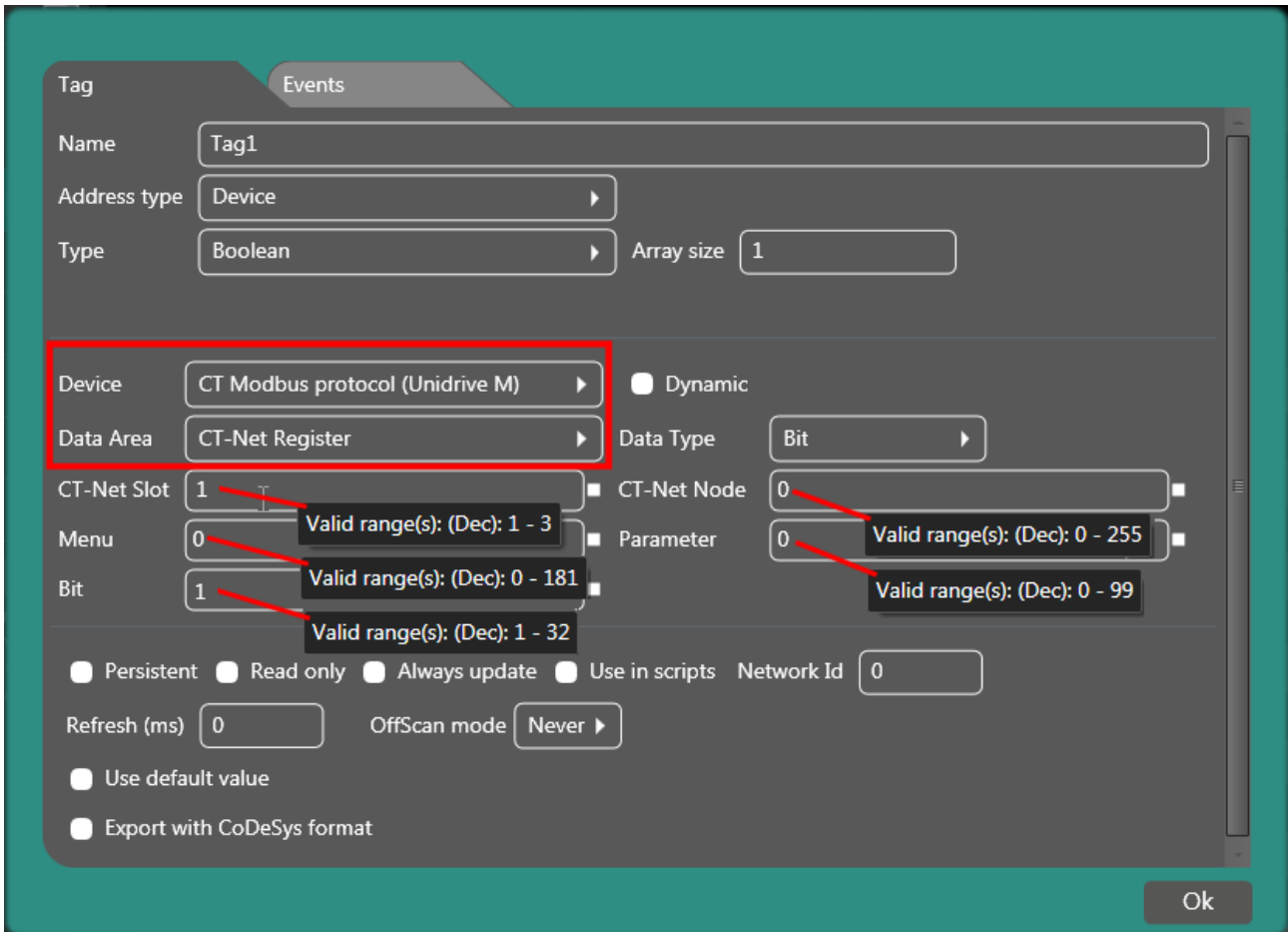
Use default value

Export with CoDeSys format

Ok

# CREW Manual

"CT-Net Register" Data Area:



Tag

Events

Name Tag1

Address type Device

Type Boolean Array size 1

Device CT Modbus protocol (Unidrive M)  Dynamic

Data Area CT-Net Register Data Type Bit

CT-Net Slot 1  CT-Net Node 0

Menu 0  Parameter 0

Bit 1

Persistent  Read only  Always update  Use in scripts Network Id 0

Refresh (ms) 0 OffScan mode Never

Use default value

Export with CoDeSys format

Ok



# CREW Manual

Type:

In the “Type” mask is used to designate the type of datum that the tag is destined to contain. The expected data types are represented in the following table.

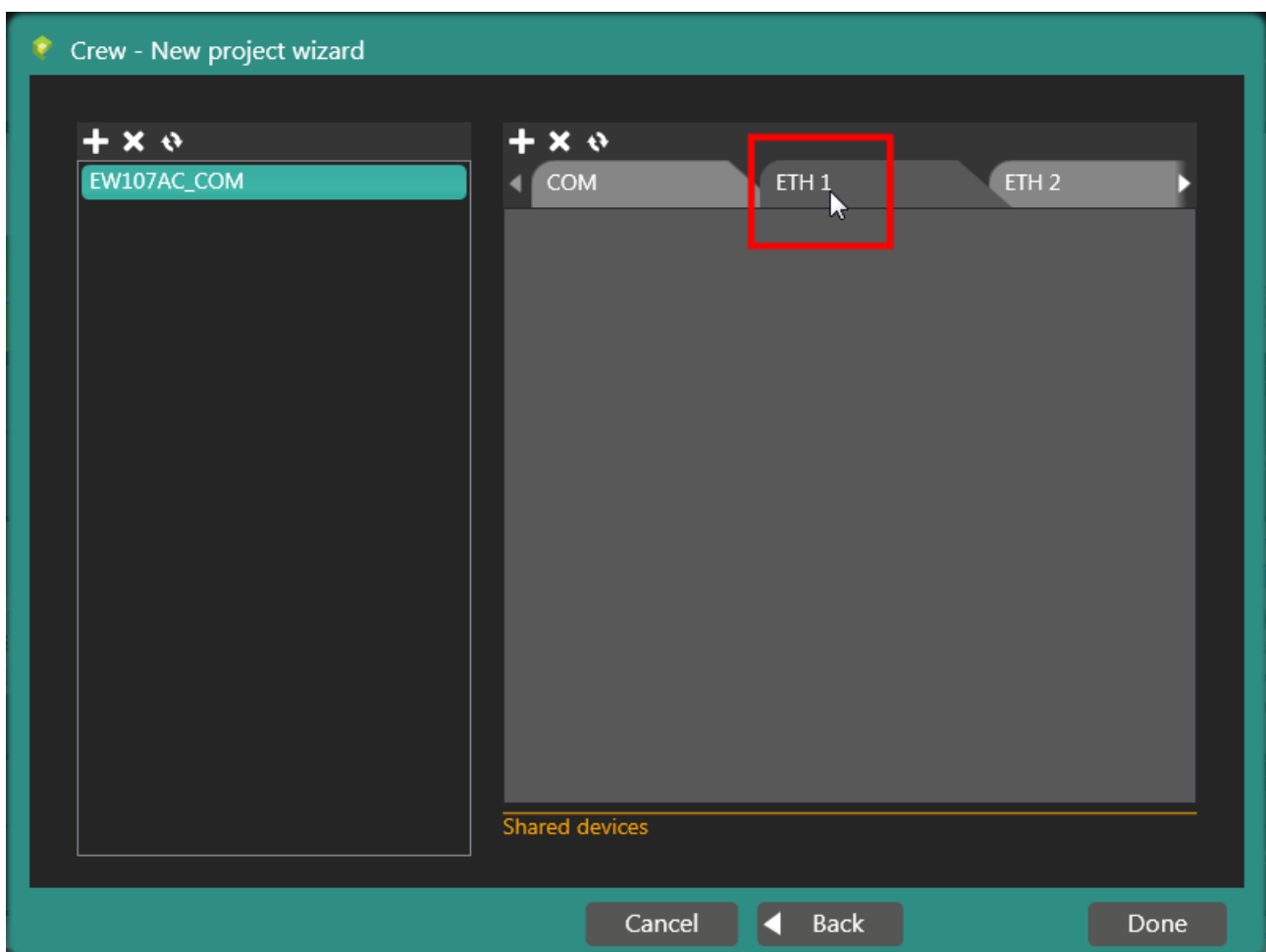
Type	Description	Range
Char	8-bit signed Integer	-128 to 127
Byte	8-bit unsigned Integer	0 to 255
Boolean	Single bit	True (1) or False(0)
Integer	16-bit signed Integer	-32,768 to 32,767
Unsigned Integer	16-bit unsigned Integer	0 to 0xFFFF
Long	32-bit signed Integer	-2,147,483,648 to 2,147,483,647
Unsigned Long	32-bit unsigned Integer	0 to 0xFFFFFFFF
Real	IEEE Floating point 32-bit single precision	-3.402823E38 to -1.401298E-45 for negative values; 1.401298E-45 to 3.402823E38 for positive values
Double	IEEE Floating point 64-bit double precision	-1.7976931348623E308 a -4.9406564584124E-324 for negative values; 4.9406564584124E-324 a 1.7976931348623E308 for positive values
String	ASCII string	ASCII String (max length 0x7FFF characters)
Array of Char	Array of 8-bit signed Integers	1 a 1024
Array of Byte	Array of 8-bit unsigned Integers	1 a 1024
Array of Boolean	Array of Single bits	1 a 1024
Array of Integer (WORD)	Array of unsigned integers	1 a 1024
Array of Unsigned Integer (WORD)	Array of unsigned integers	1 a 1024
Array of Long (DWORD)	Array of 32-bit signed Integers	1 a 1024
Array of Unsigned Long (DWORD)	Array of 32-bit unsigned Integers	1 a 1024
Array of Real	Array of IEEE Floating points 32-bit single precision	1 a 1024
Array of Double	Array of IEEE Floating points 64-bit double precision	1 a 1024
Array of String	Array of ASCII Strings	1 a 1024

# CREW Manual

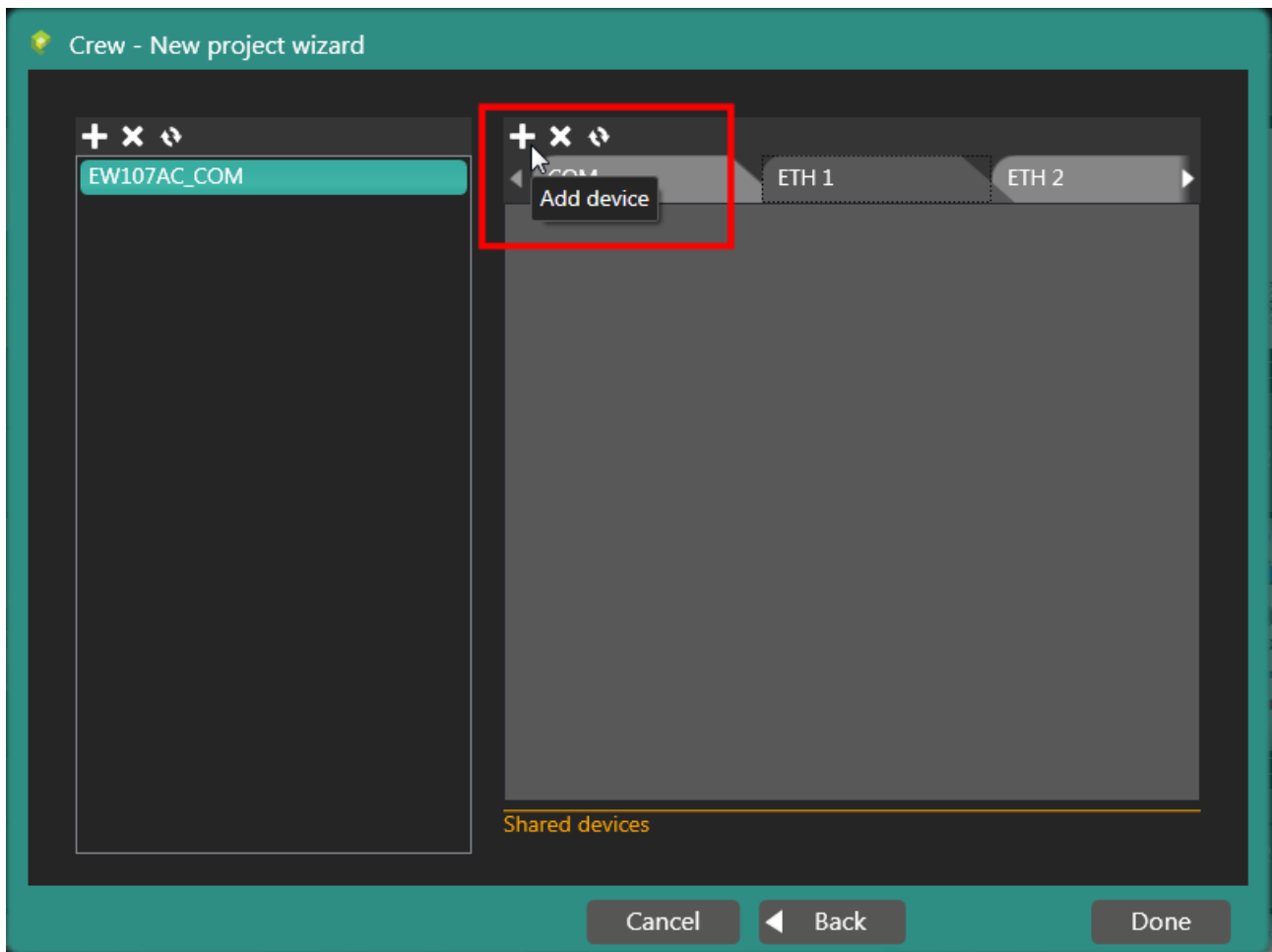
For additional information on the protocol, refer to section, "[Control Techniques CT Modbus](#)"

For more information on the variables (Tags), refer to section "[Tags](#)"

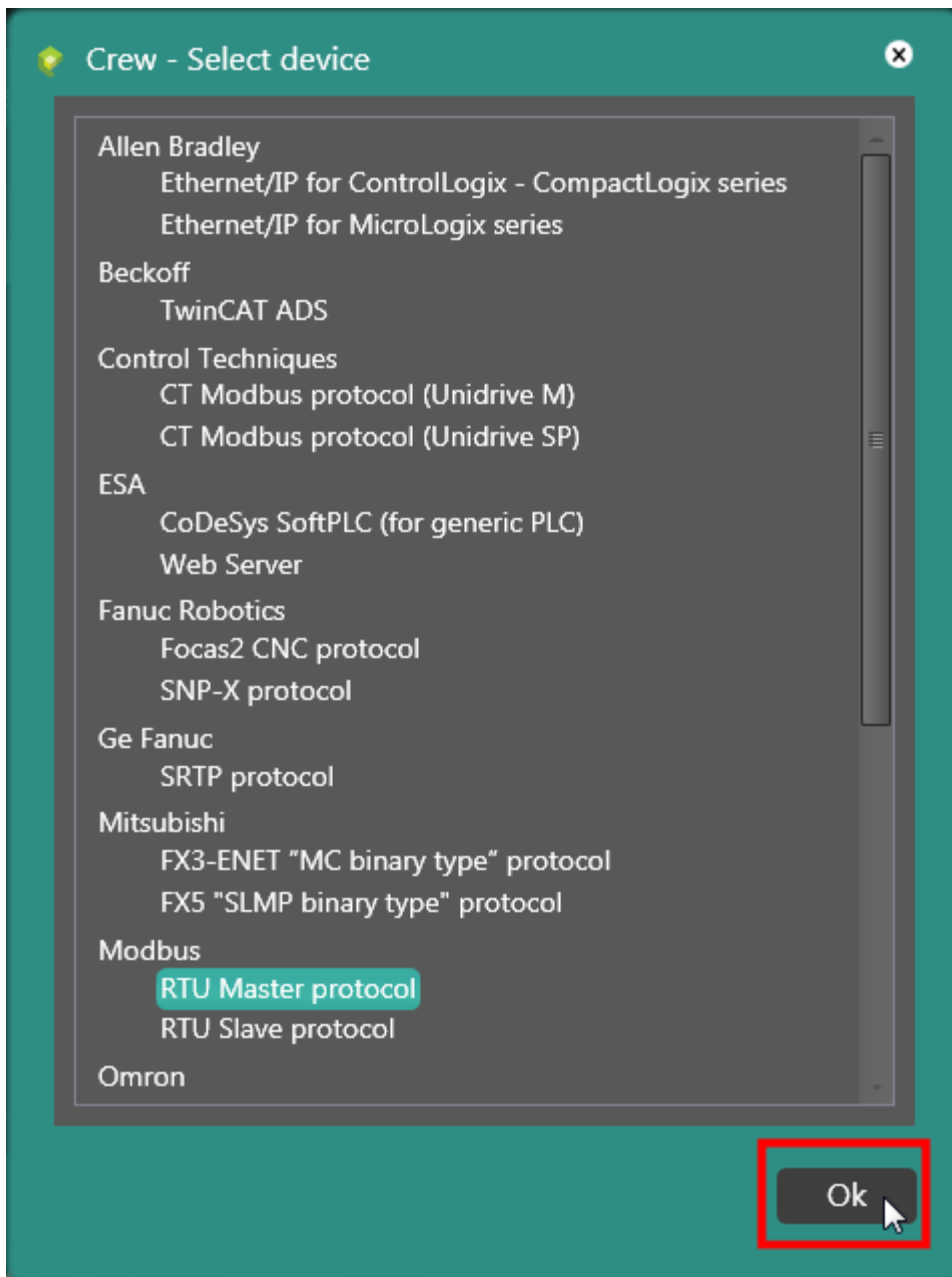
## Modbus RTU Master Ethernet



# CREW Manual



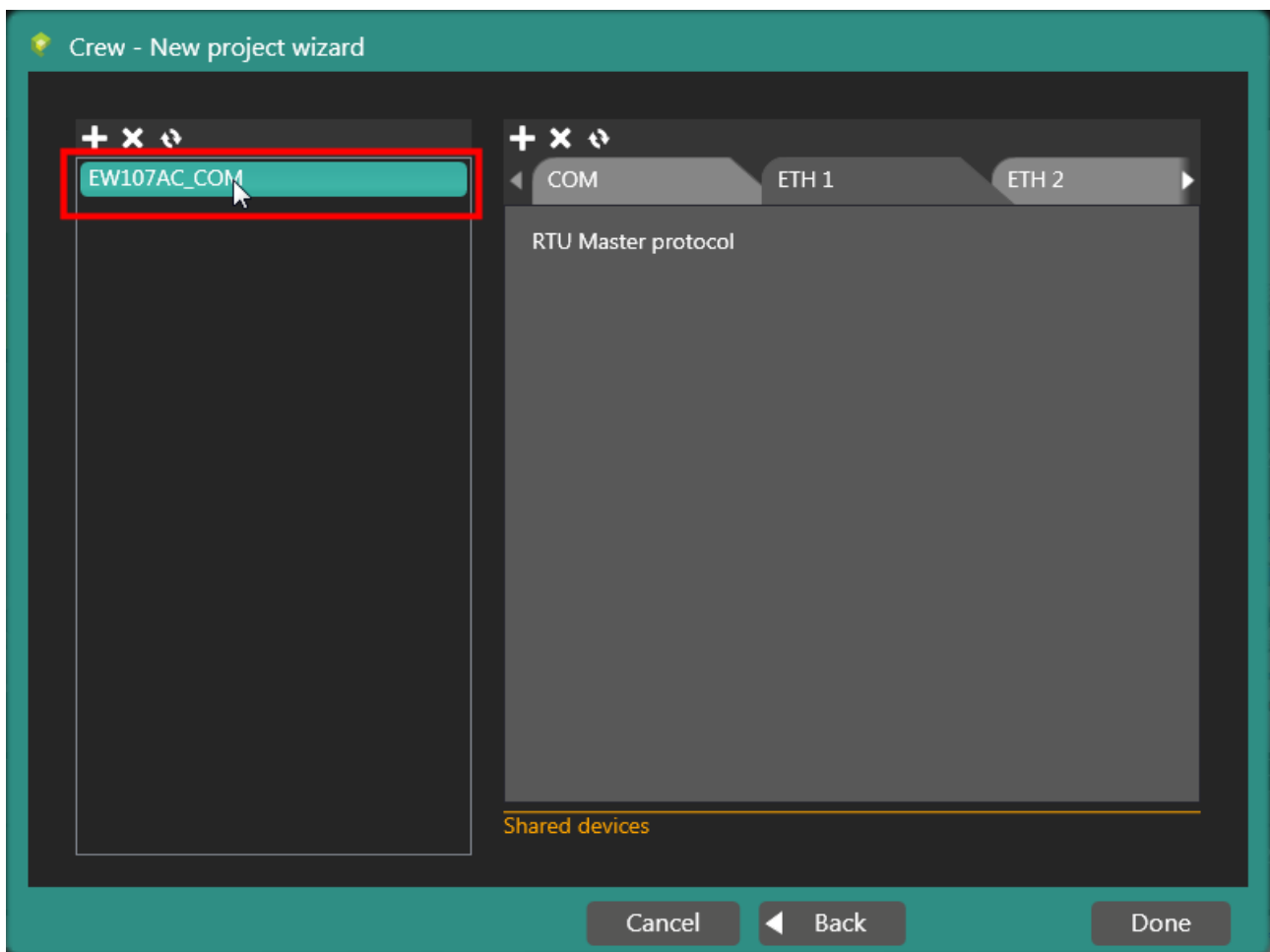
# CREW Manual



# CREW Manual

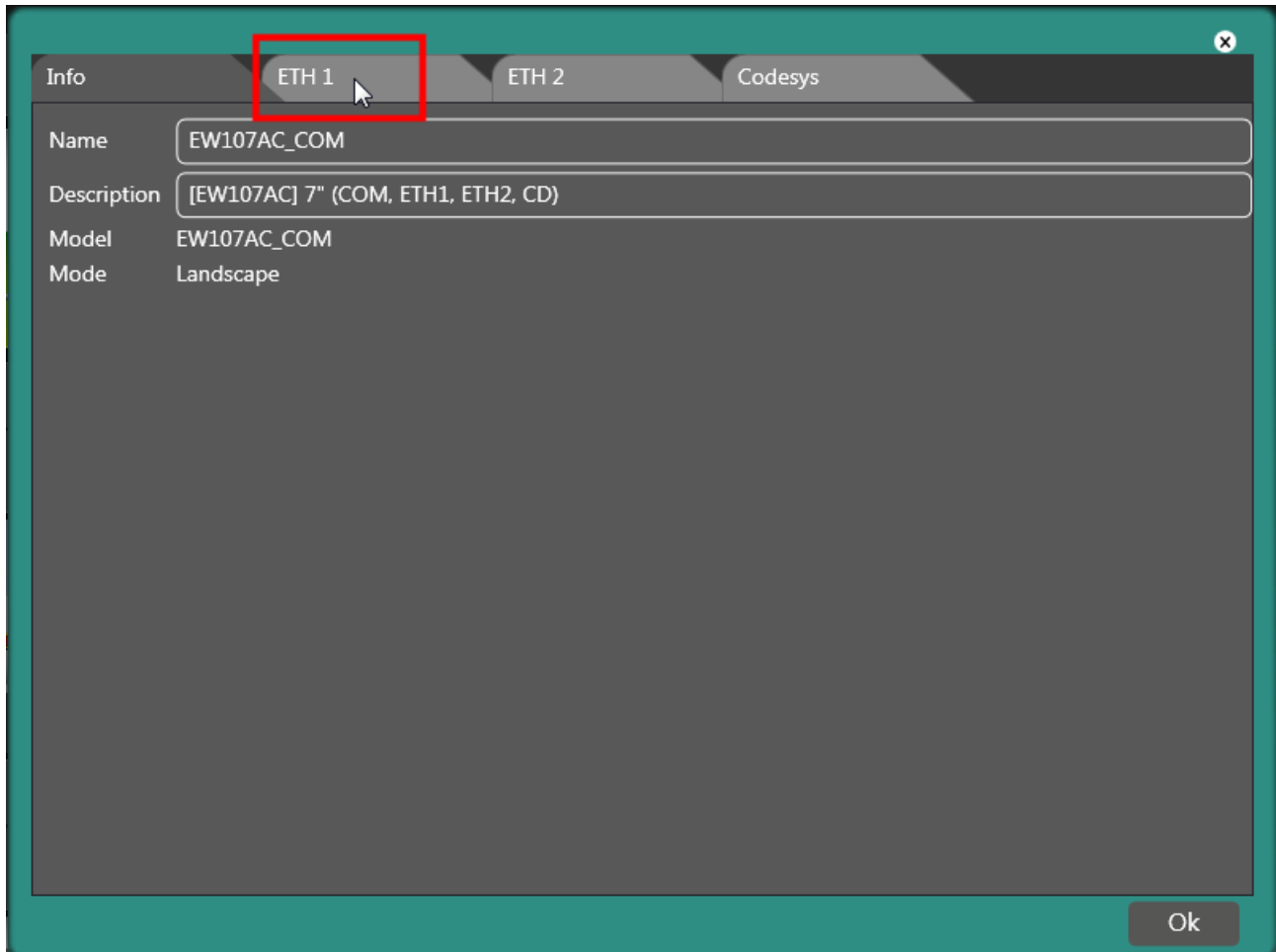
Settings on EW side:

Double click on the name of the terminal.

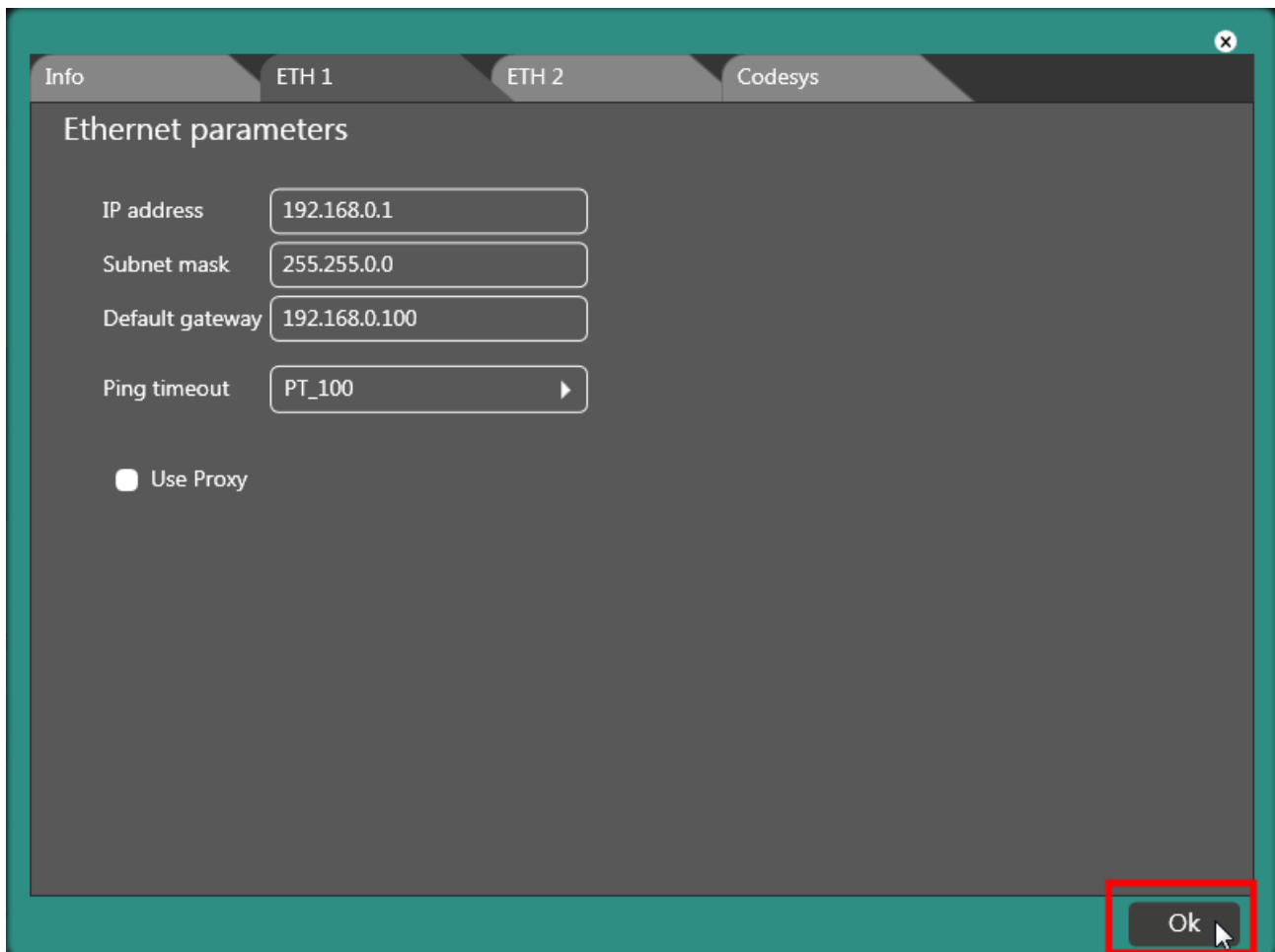


# CREW Manual

Ethernet Parameters:



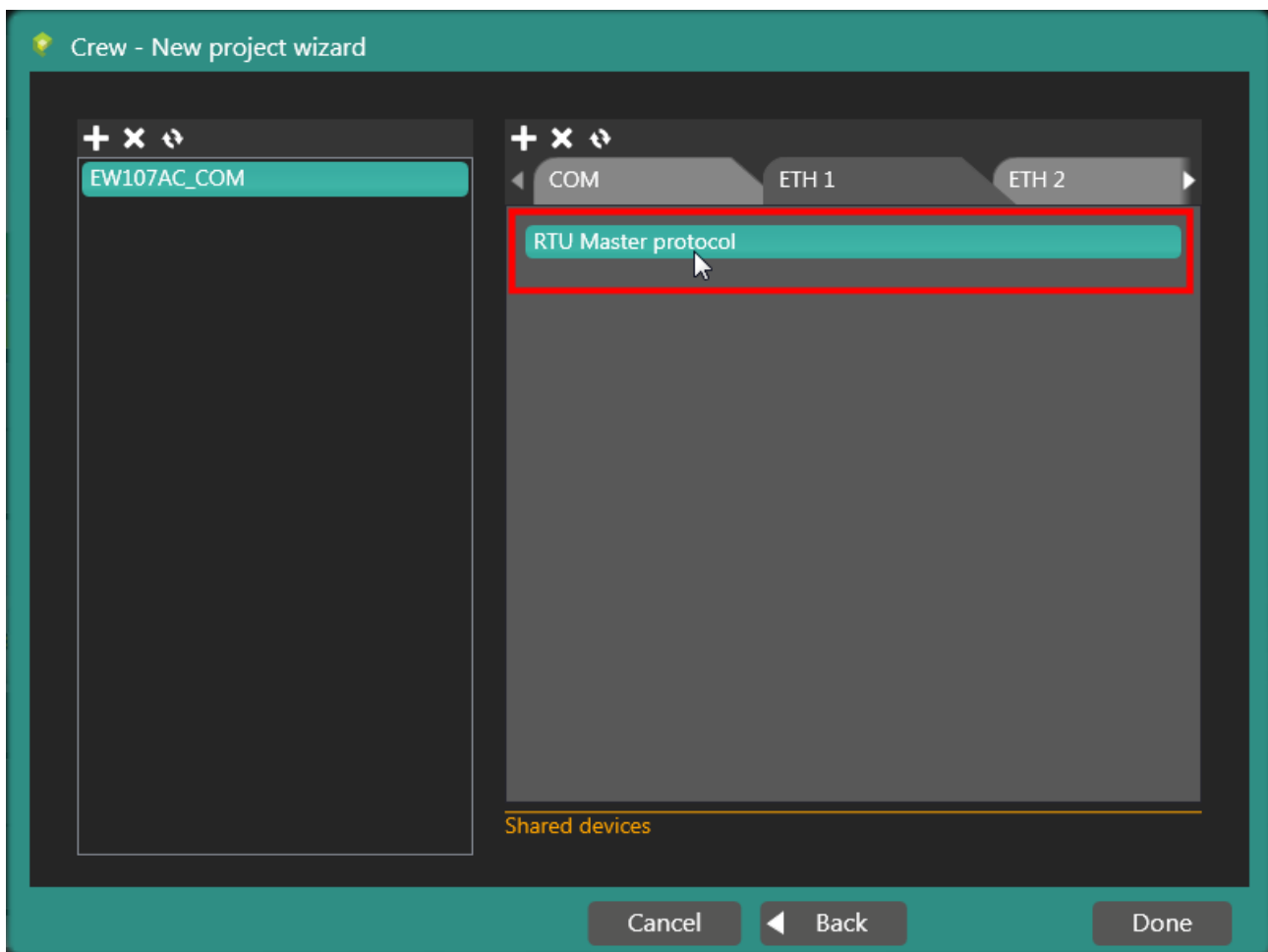
# CREW Manual



# CREW Manual

Settings on device side.

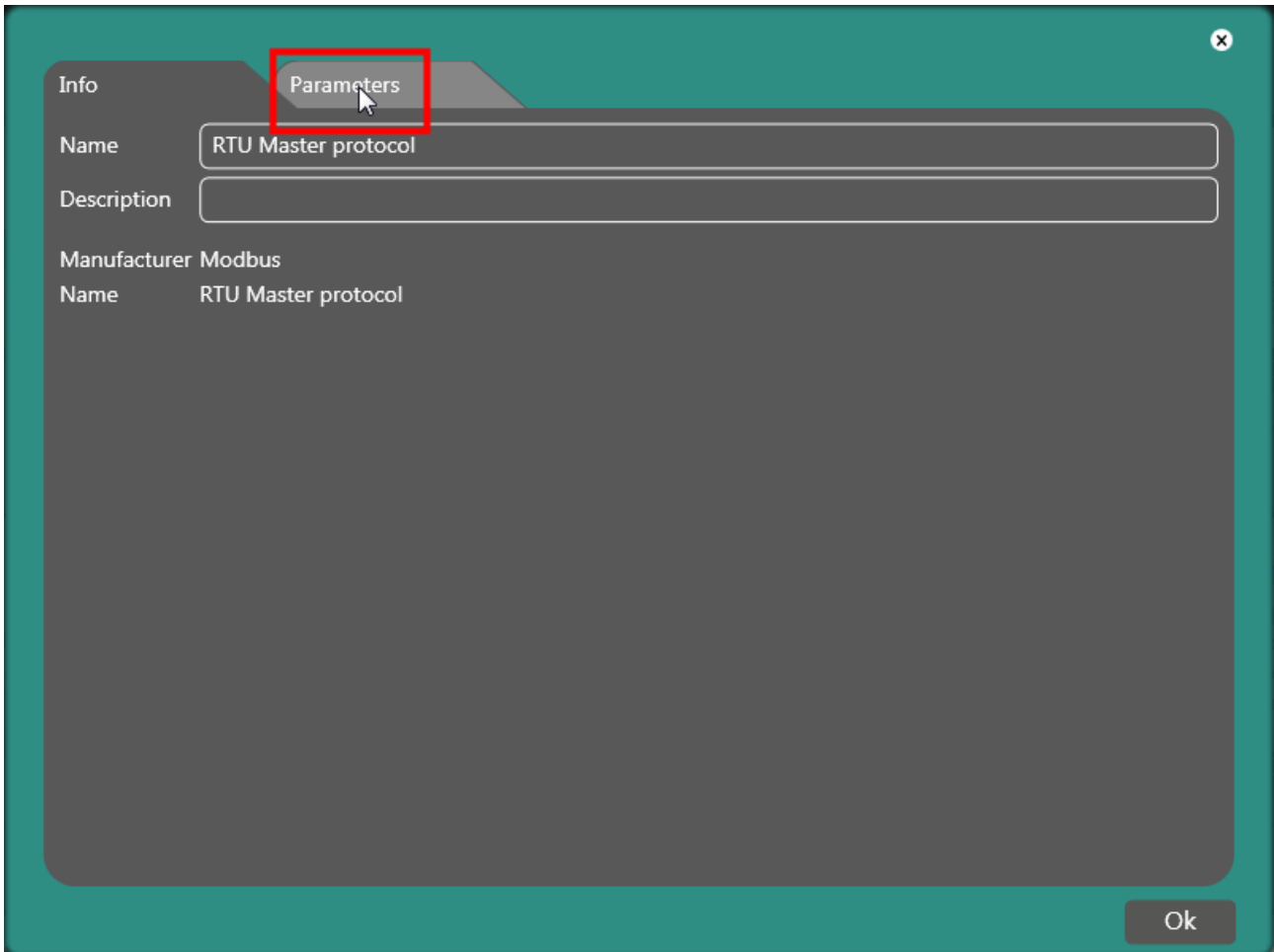
Double click on the name of the device.





# CREW Manual

Device parameters:

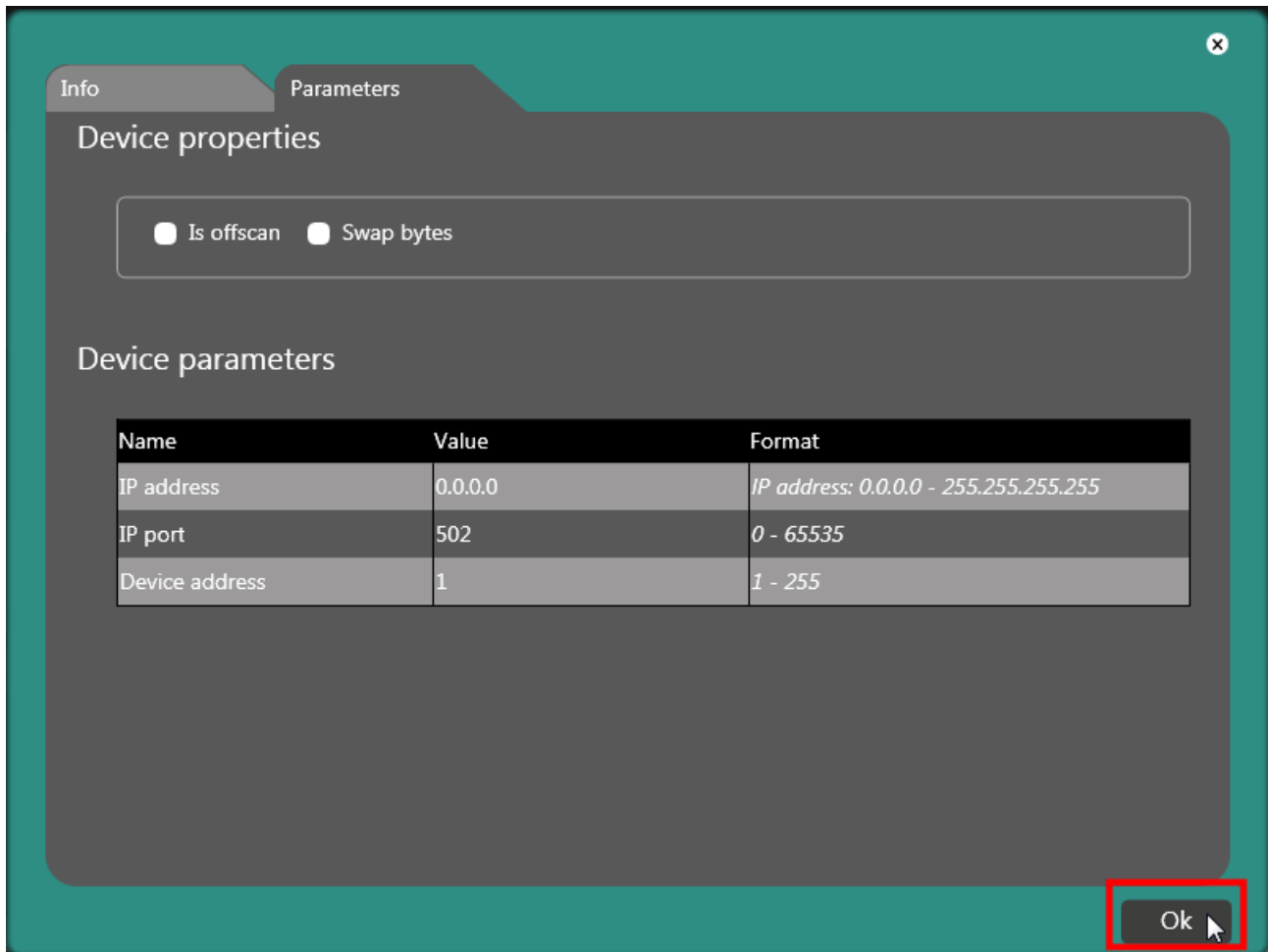


The screenshot shows a software window with a teal header and a dark grey body. At the top left, there are two tabs: 'Info' and 'Parameters'. The 'Parameters' tab is selected and highlighted with a red rectangular box. Below the tabs, there are several input fields and labels:

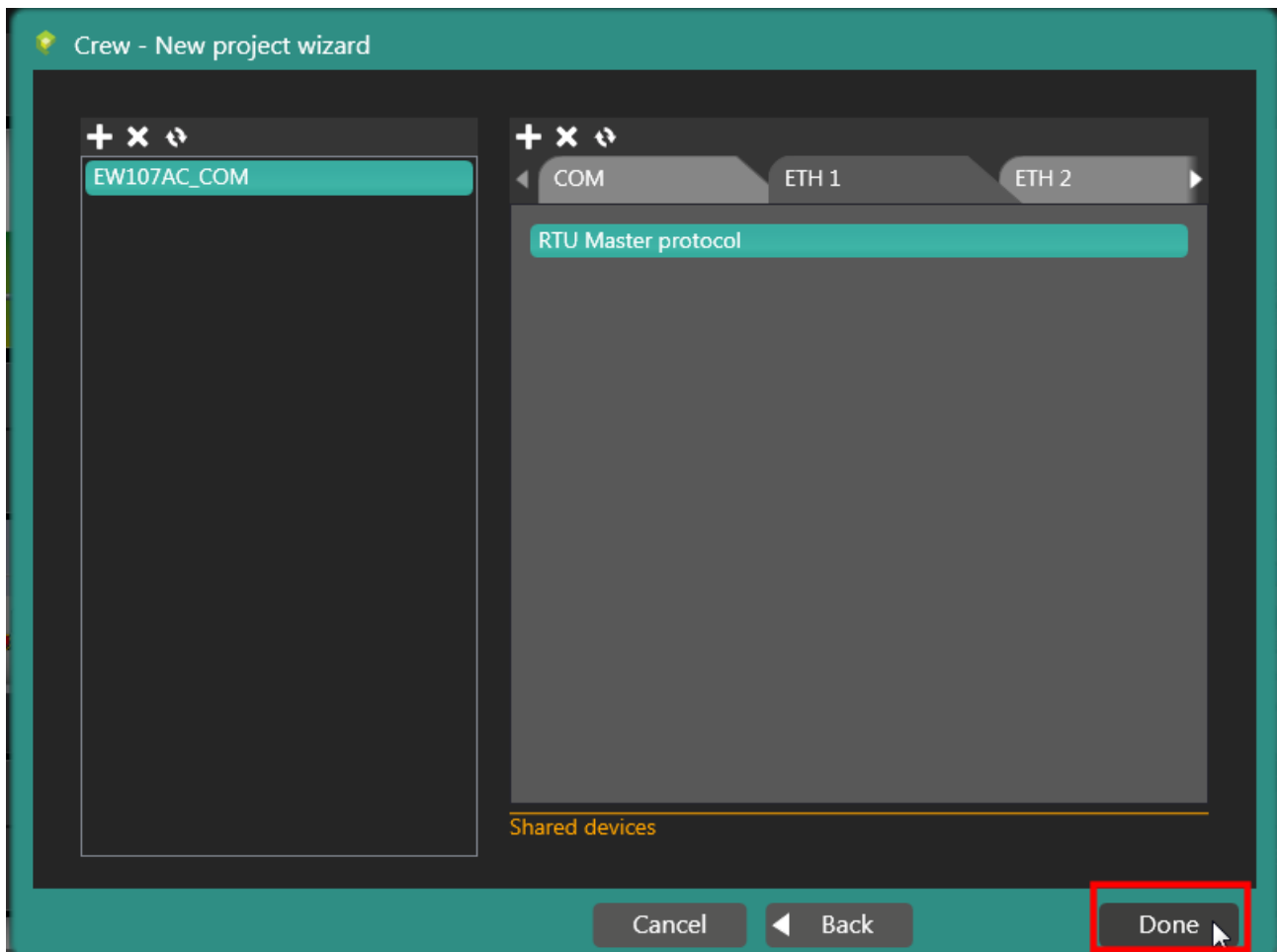
- Name:** A text box containing 'RTU Master protocol'.
- Description:** An empty text box.
- Manufacturer Modbus:** A section header.
- Name:** A text box containing 'RTU Master protocol'.

An 'Ok' button is located at the bottom right of the window. A close button (marked with an 'x') is in the top right corner.

# CREW Manual

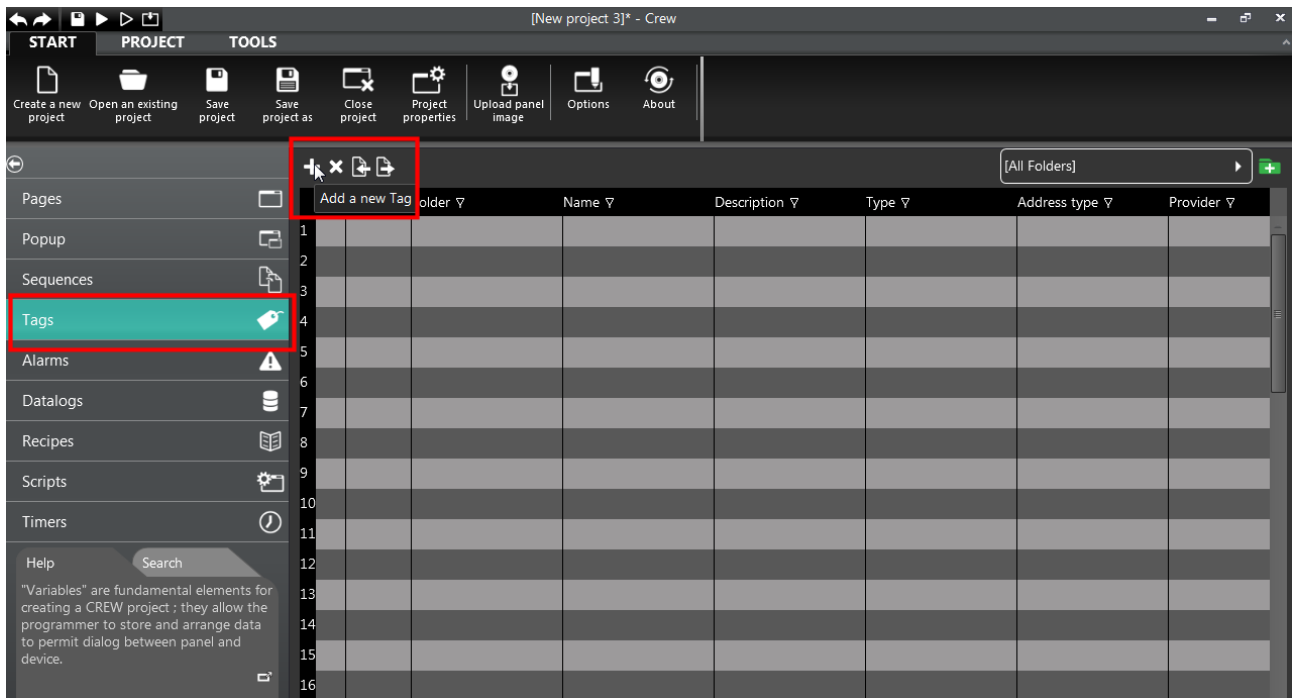


# CREW Manual



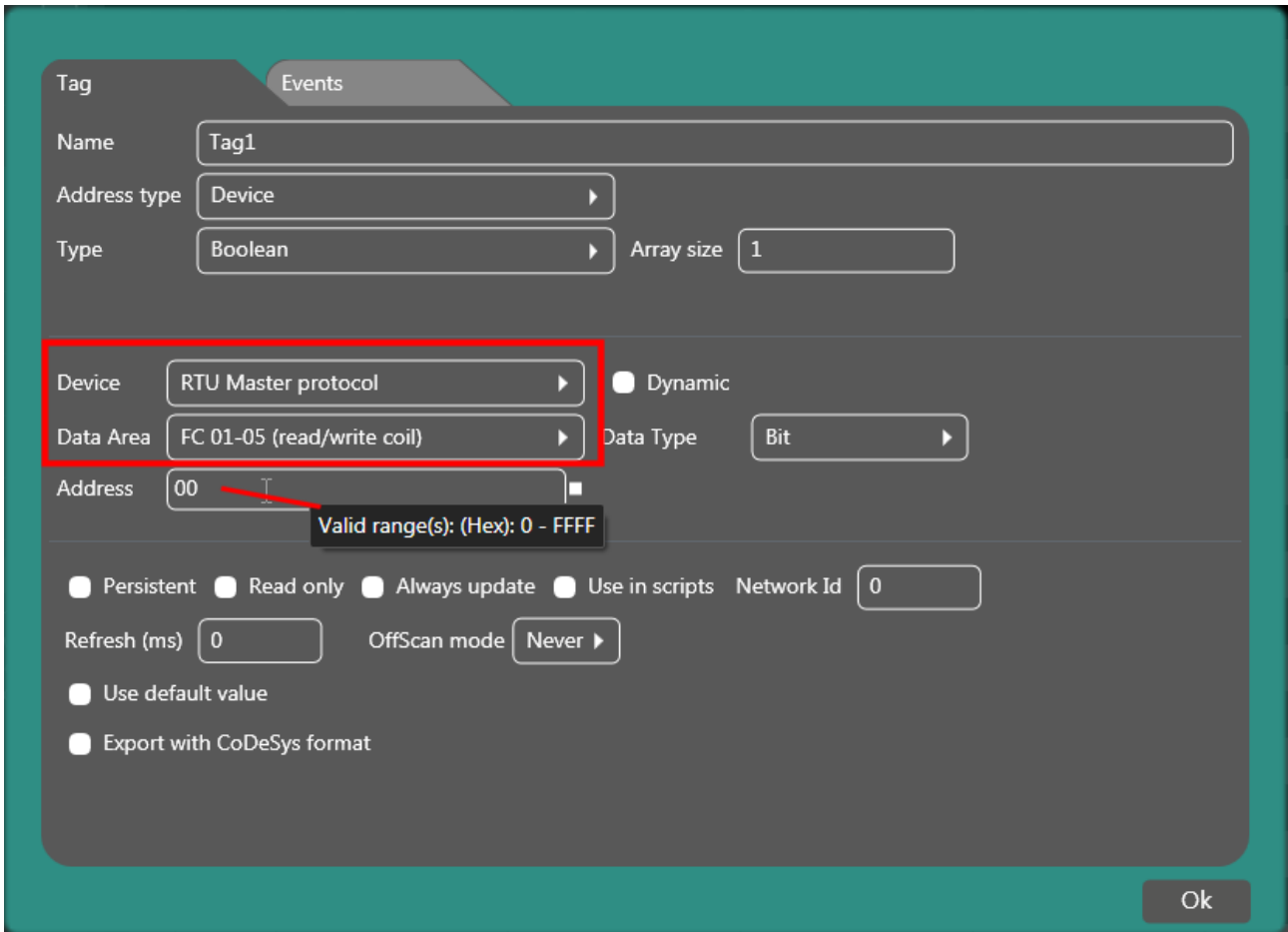
# CREW Manual

## Set up Parameters - Modbus RTU Master Ethernet



# CREW Manual

Data Area: "FC 01-05 (read/write coil)":



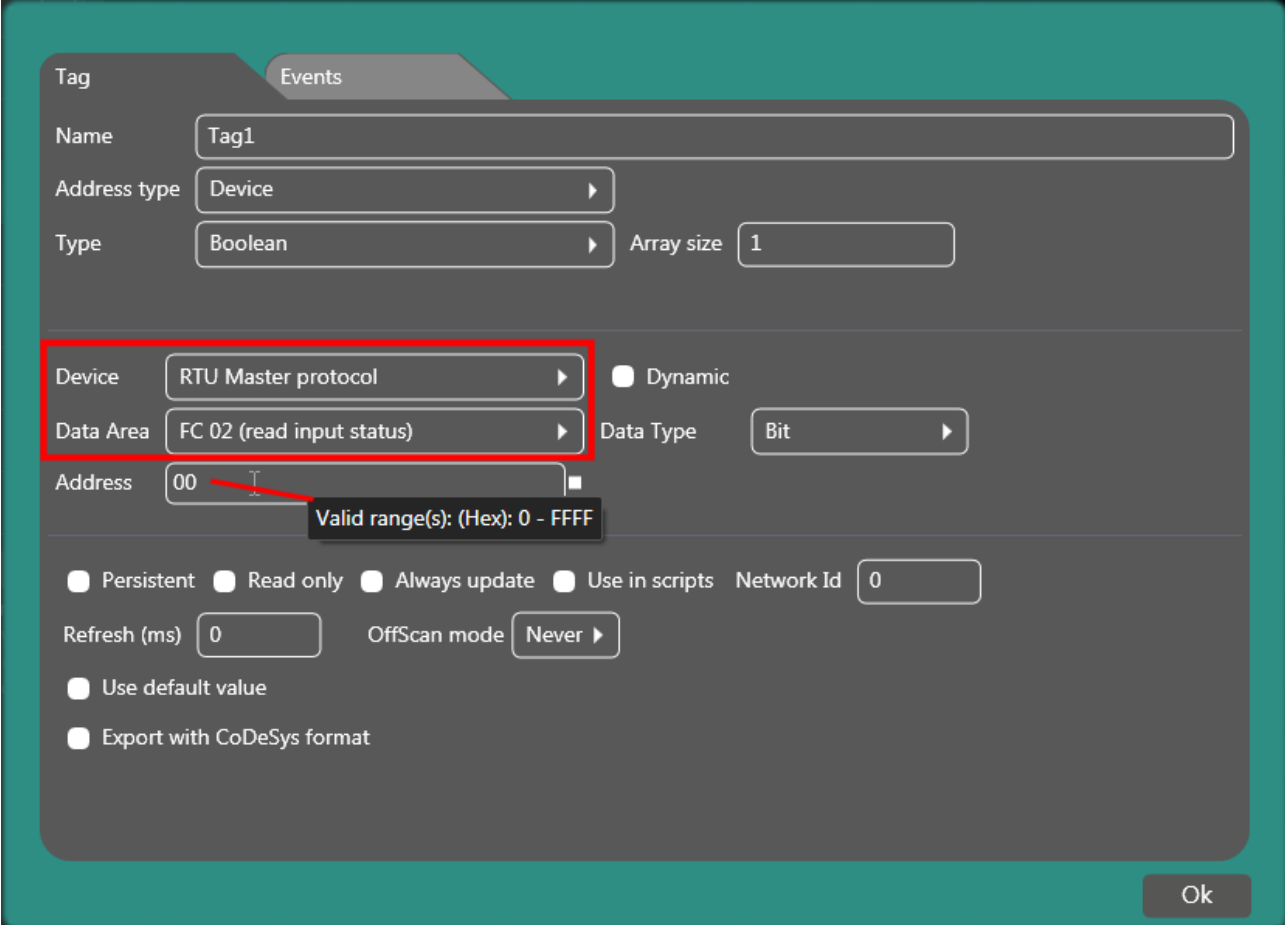
The screenshot shows a configuration window for a tag named "Tag1". The "Tag" tab is active, and the "Events" sub-tab is selected. The configuration fields are as follows:

- Name: Tag1
- Address type: Device
- Type: Boolean
- Array size: 1
- Device: RTU Master protocol
- Dynamic:
- Data Area: FC 01-05 (read/write coil)
- Data Type: Bit
- Address: 00
- Valid range(s): (Hex): 0 - FFFF
- Persistent:
- Read only:
- Always update:
- Use in scripts:
- Network Id: 0
- Refresh (ms): 0
- OffScan mode: Never
- Use default value:
- Export with CoDeSys format:

An "Ok" button is located at the bottom right of the dialog.

# CREW Manual

Data Area: "FC 02 (read input status)":



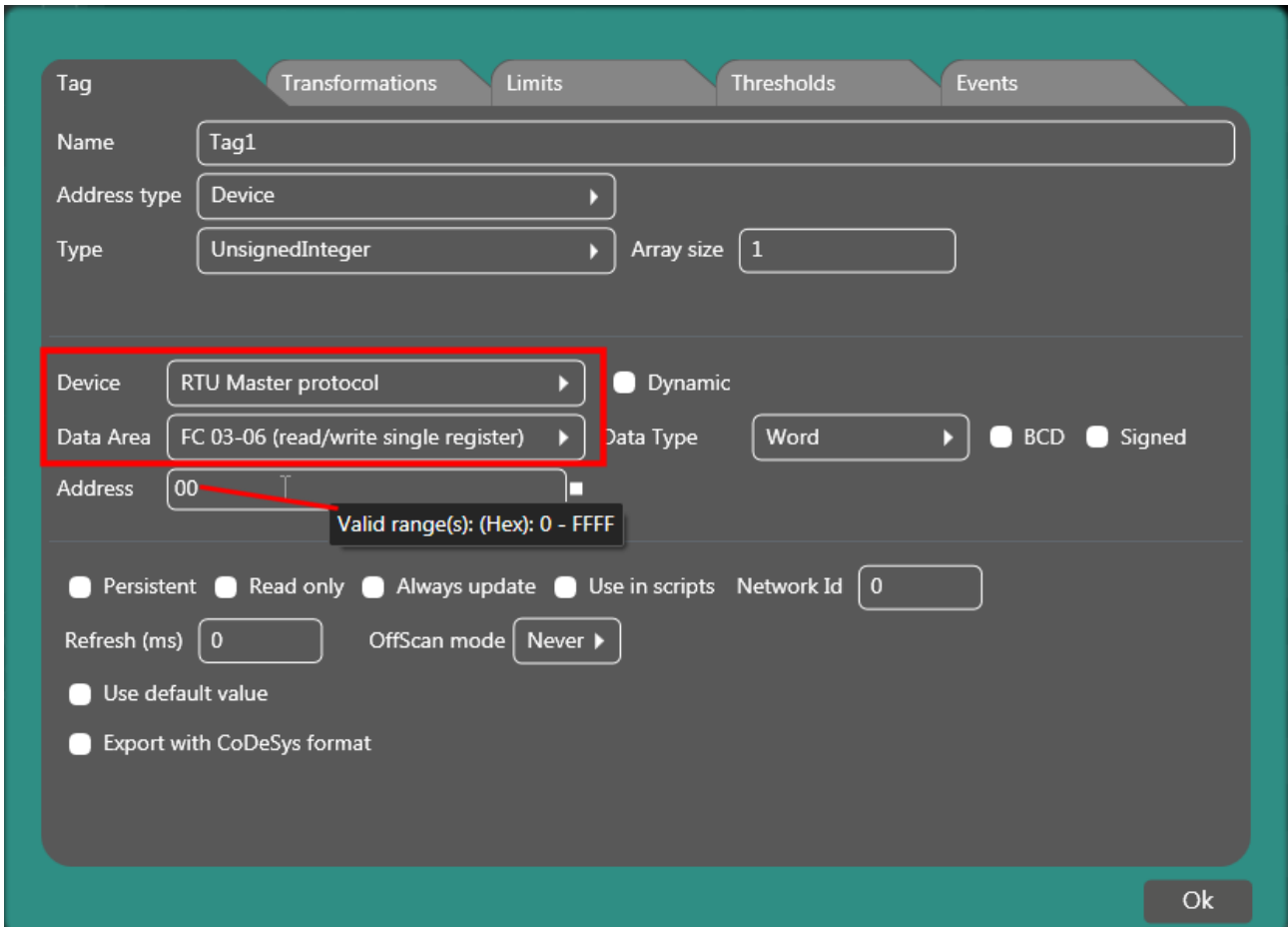
The screenshot shows a configuration dialog box for a Tag. The 'Tag' tab is active, and the 'Events' sub-tab is selected. The configuration is as follows:

- Name: Tag1
- Address type: Device
- Type: Boolean
- Array size: 1
- Device: RTU Master protocol
- Dynamic:
- Data Area: FC 02 (read input status)
- Data Type: Bit
- Address: 00
- Valid range(s): (Hex): 0 - FFFF
- Persistent:
- Read only:
- Always update:
- Use in scripts:
- Network Id: 0
- Refresh (ms): 0
- OffScan mode: Never
- Use default value:
- Export with CoDeSys format:

An 'Ok' button is located at the bottom right of the dialog.

# CREW Manual

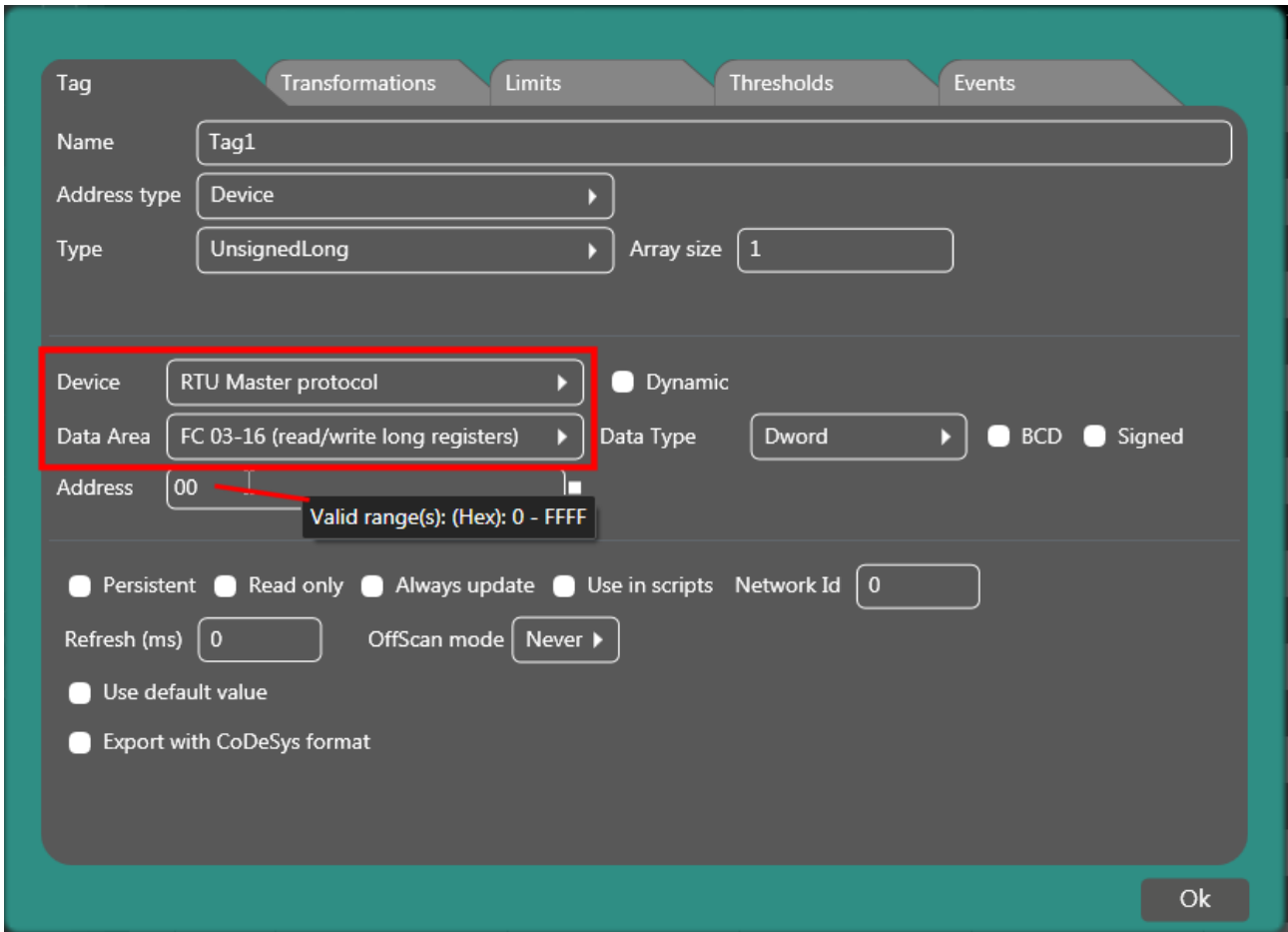
Data Area: "FC 03-06 (read/write single register)":



The screenshot displays the configuration window for a tag named 'Tag1'. The 'Tag' tab is active, showing various settings. A red box highlights the 'Device' dropdown menu (set to 'RTU Master protocol') and the 'Data Area' dropdown menu (set to 'FC 03-06 (read/write single register)'). Below these, the 'Address' field is set to '00', with a tooltip indicating a valid range of '0 - FFFF'. Other settings include 'Type' as 'UnsignedInteger', 'Array size' as '1', 'Data Type' as 'Word', and several checkboxes for 'Persistent', 'Read only', 'Always update', 'Use in scripts', 'Use default value', and 'Export with CoDeSys format'. The 'Network Id' is set to '0', 'Refresh (ms)' is '0', and 'OffScan mode' is 'Never'. An 'Ok' button is located at the bottom right.

# CREW Manual

Data Area: "FC 03-16 (read/write long register)":



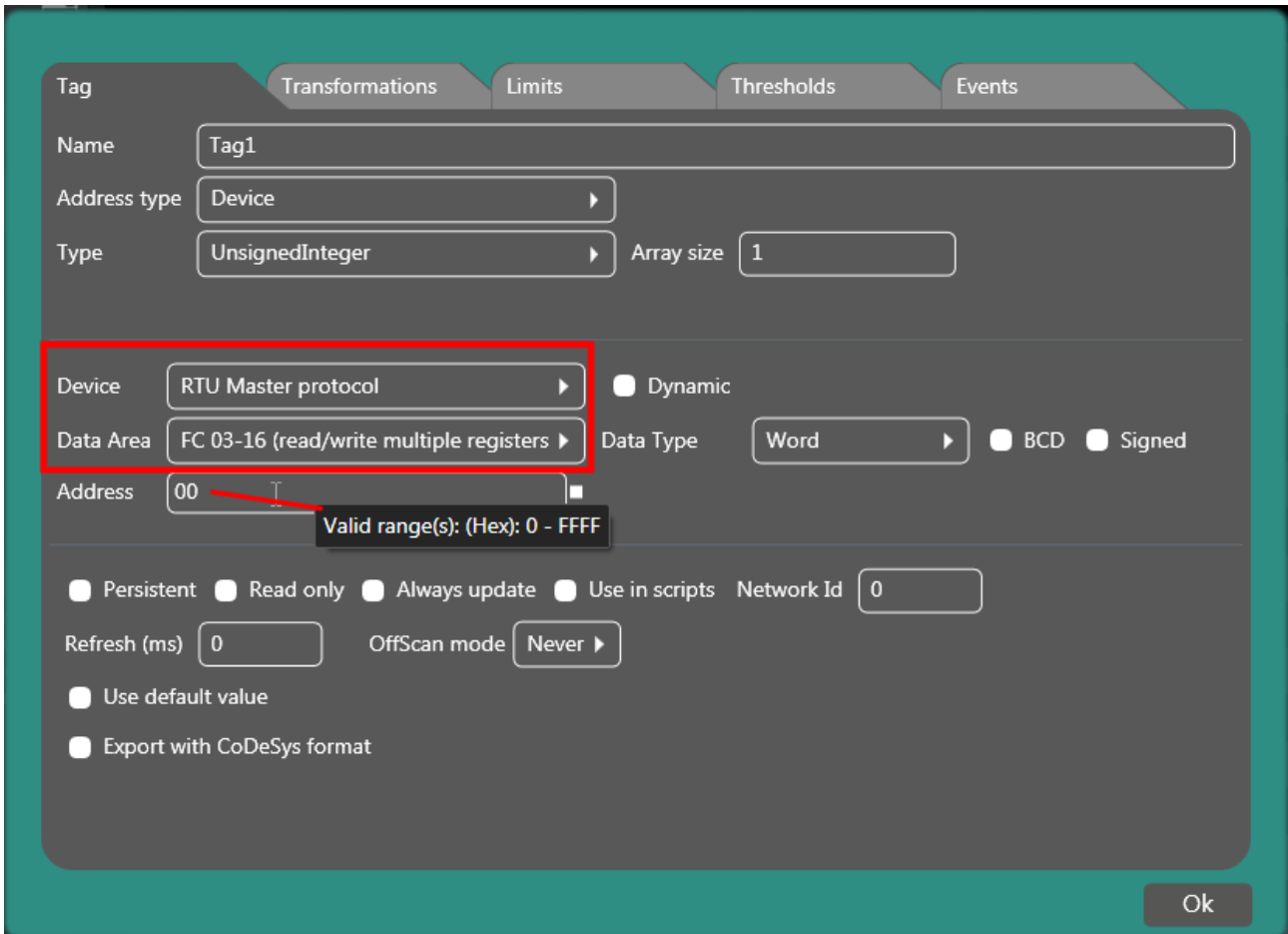
The screenshot displays the configuration window for a tag named 'Tag1'. The 'Tag' tab is active, showing various settings. A red box highlights the 'Device' and 'Data Area' fields. The 'Device' is set to 'RTU Master protocol' and the 'Data Area' is set to 'FC 03-16 (read/write long registers)'. Other visible settings include 'Address type' set to 'Device', 'Type' set to 'UnsignedLong', 'Array size' set to '1', 'Data Type' set to 'Dword', and 'Address' set to '00'. A tooltip indicates the valid range for the address is '(Hex): 0 - FFFF'. At the bottom right, there is an 'Ok' button.

Field	Value
Name	Tag1
Address type	Device
Type	UnsignedLong
Array size	1
Device	RTU Master protocol
Data Area	FC 03-16 (read/write long registers)
Data Type	Dword
Address	00
Dynamic	<input type="checkbox"/>
BCD	<input type="checkbox"/>
Signed	<input type="checkbox"/>
Persistent	<input type="checkbox"/>
Read only	<input type="checkbox"/>
Always update	<input type="checkbox"/>
Use in scripts	<input type="checkbox"/>
Network Id	0
Refresh (ms)	0
OffScan mode	Never
Use default value	<input type="checkbox"/>
Export with CoDeSys format	<input type="checkbox"/>



# CREW Manual

Data Area: "FC 03-16 (read/write multiple register)":



The screenshot shows the configuration window for a tag named "Tag1". The "Tag" tab is active, and the "Data Area" is set to "FC 03-16 (read/write multiple registers)". A red box highlights the "Device" dropdown (set to "RTU Master protocol"), the "Data Area" dropdown, and the "Address" field (set to "00"). A tooltip below the address field indicates "Valid range(s): (Hex): 0 - FFFF". Other settings include "Type" as "UnsignedInteger", "Array size" as "1", "Data Type" as "Word", and various checkboxes for "Dynamic", "Persistent", "Read only", "Always update", "Use in scripts", "Use default value", and "Export with CoDeSys format". The "Network Id" is set to "0". An "Ok" button is visible at the bottom right.

Tag

Transformations Limits Thresholds Events

Name Tag1

Address type Device

Type UnsignedInteger Array size 1

Device RTU Master protocol  Dynamic

Data Area FC 03-16 (read/write multiple registers) Data Type Word  BCD  Signed

Address 00 Valid range(s): (Hex): 0 - FFFF

Persistent  Read only  Always update  Use in scripts Network Id 0

Refresh (ms) 0 OffScan mode Never

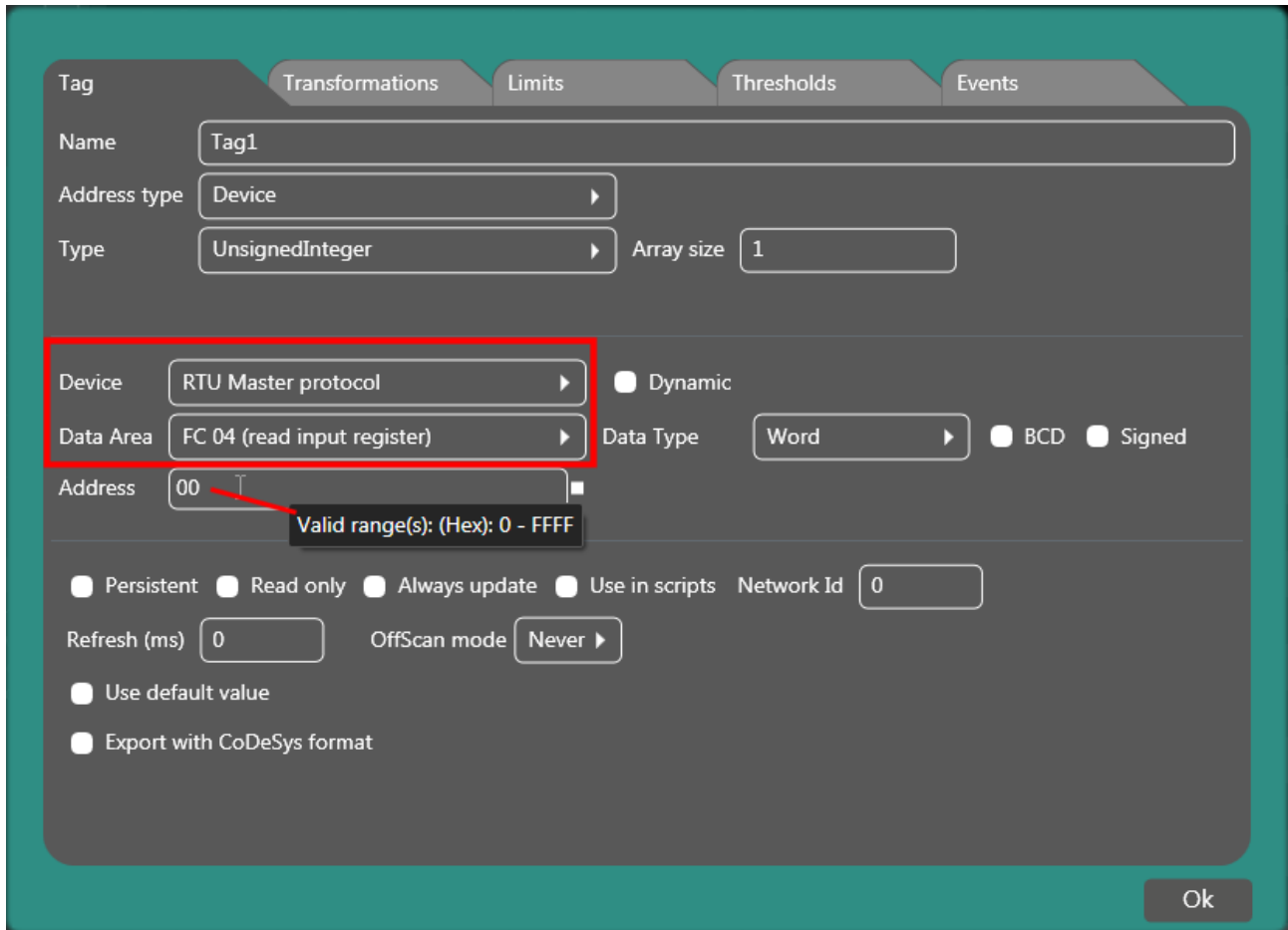
Use default value

Export with CoDeSys format

Ok

# CREW Manual

Data Area: "FC 04 (read input register)":



The screenshot shows the configuration window for a tag named "Tag1". The window has several tabs: "Tag", "Transformations", "Limits", "Thresholds", and "Events". The "Tag" tab is active. The configuration fields are as follows:

- Name: Tag1
- Address type: Device
- Type: UnsignedInteger
- Array size: 1
- Device: RTU Master protocol
- Data Area: FC 04 (read input register)
- Address: 00
- Data Type: Word
- Dynamic:
- BCD:
- Signed:
- Persistent:
- Read only:
- Always update:
- Use in scripts:
- Network Id: 0
- Refresh (ms): 0
- OffScan mode: Never
- Use default value:
- Export with CoDeSys format:

A red box highlights the "Device", "Data Area", and "Address" fields. A tooltip below the "Address" field indicates "Valid range(s): (Hex): 0 - FFFF". An "Ok" button is located at the bottom right of the window.

# CREW Manual

Type:

In the “Type” mask is used to designate the type of datum that the tag is destined to contain. The expected data types are represented in the following table.

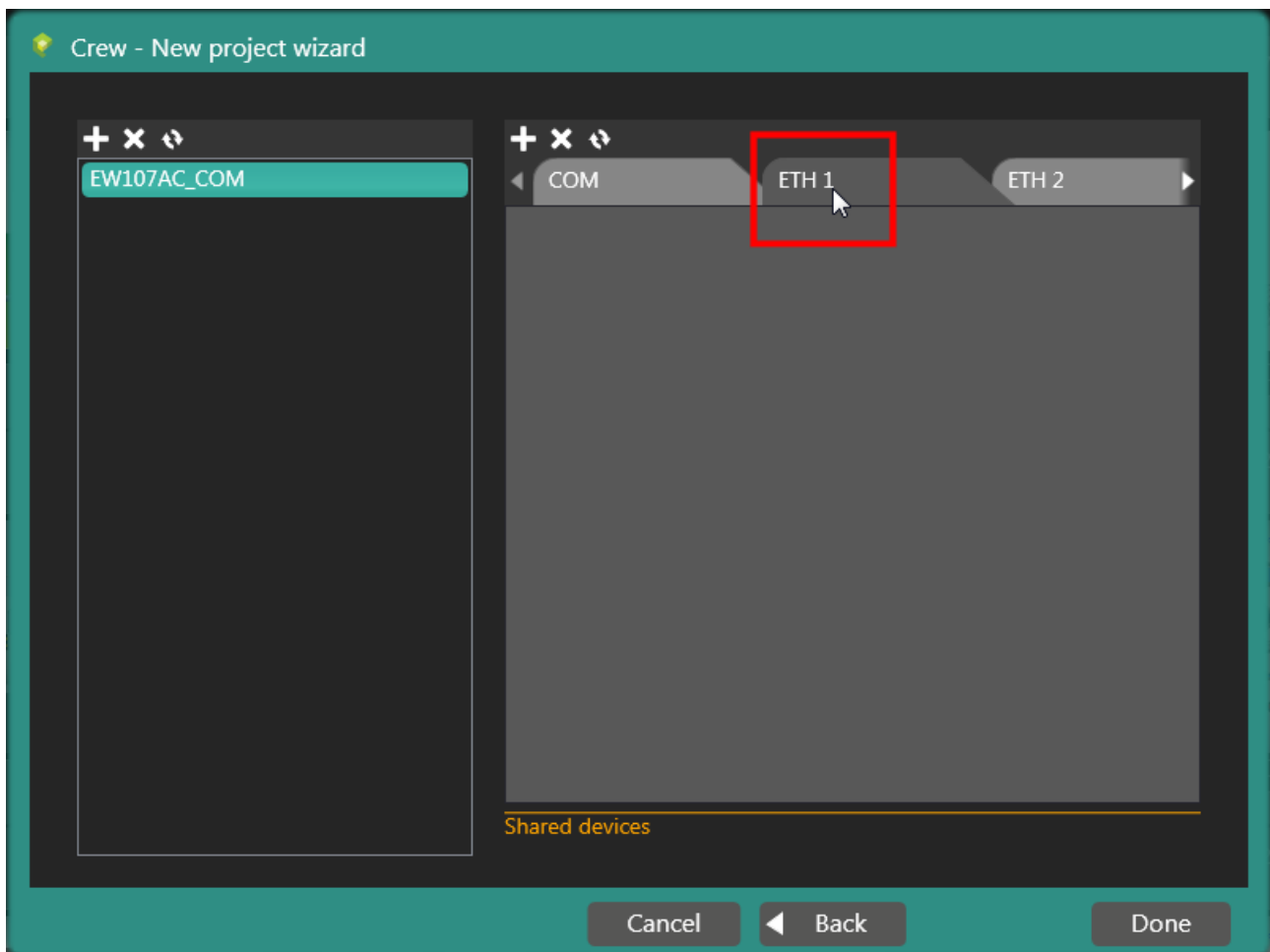
Type	Description	Range
Char	8-bit signed Integer	-128 to 127
Byte	8-bit unsigned Integer	0 to 255
Boolean	Single bit	True (1) or False(0)
Integer	16-bit signed Integer	-32,768 to 32,767
Unsigned Integer	16-bit unsigned Integer	0 to 0xFFFF
Long	32-bit signed Integer	-2,147,483,648 to 2,147,483,647
Unsigned Long	32-bit unsigned Integer	0 to 0xFFFFFFFF
Real	IEEE Floating point 32-bit single precision	-3.402823E38 to -1.401298E-45 for negative values; 1.401298E-45 to 3.402823E38 for positive values
Double	IEEE Floating point 64-bit double precision	-1.7976931348623E308 a -4.9406564584124E-324 for negative values; 4.9406564584124E-324 a 1.7976931348623E308 for positive values
String	ASCII string	ASCII String (max length 0x7FFF characters)
Array of Char	Array of 8-bit signed Integers	1 a 1024
Array of Byte	Array of 8-bit unsigned Integers	1 a 1024
Array of Boolean	Array of Single bits	1 a 1024
Array of Integer (WORD)	Array of unsigned integers	1 a 1024
Array of Unsigned Integer (WORD)	Array of unsigned integers	1 a 1024
Array of Long (DWORD)	Array of 32-bit signed Integers	1 a 1024
Array of Unsigned Long (DWORD)	Array of 32-bit unsigned Integers	1 a 1024
Array of Real	Array of IEEE Floating points 32-bit single precision	1 a 1024
Array of Double	Array of IEEE Floating points 64-bit double precision	1 a 1024
Array of String	Array of ASCII Strings	1 a 1024

For additional information on the protocol, refer to Drivers section, "[Modbus Master RTU Tcp](#)".

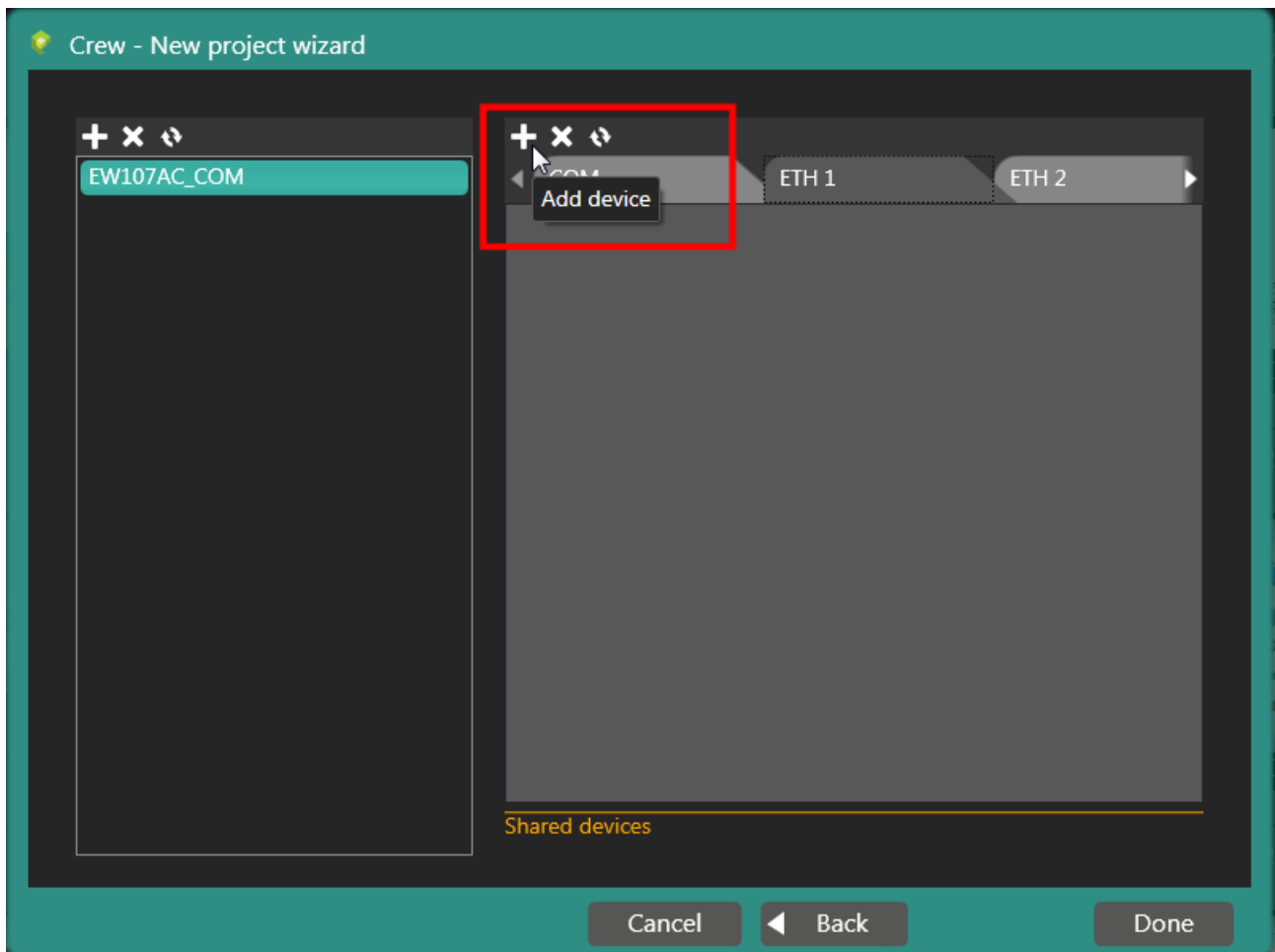
# CREW Manual

For more information on the variables (tags), refer to the "[Tags](#)" section.

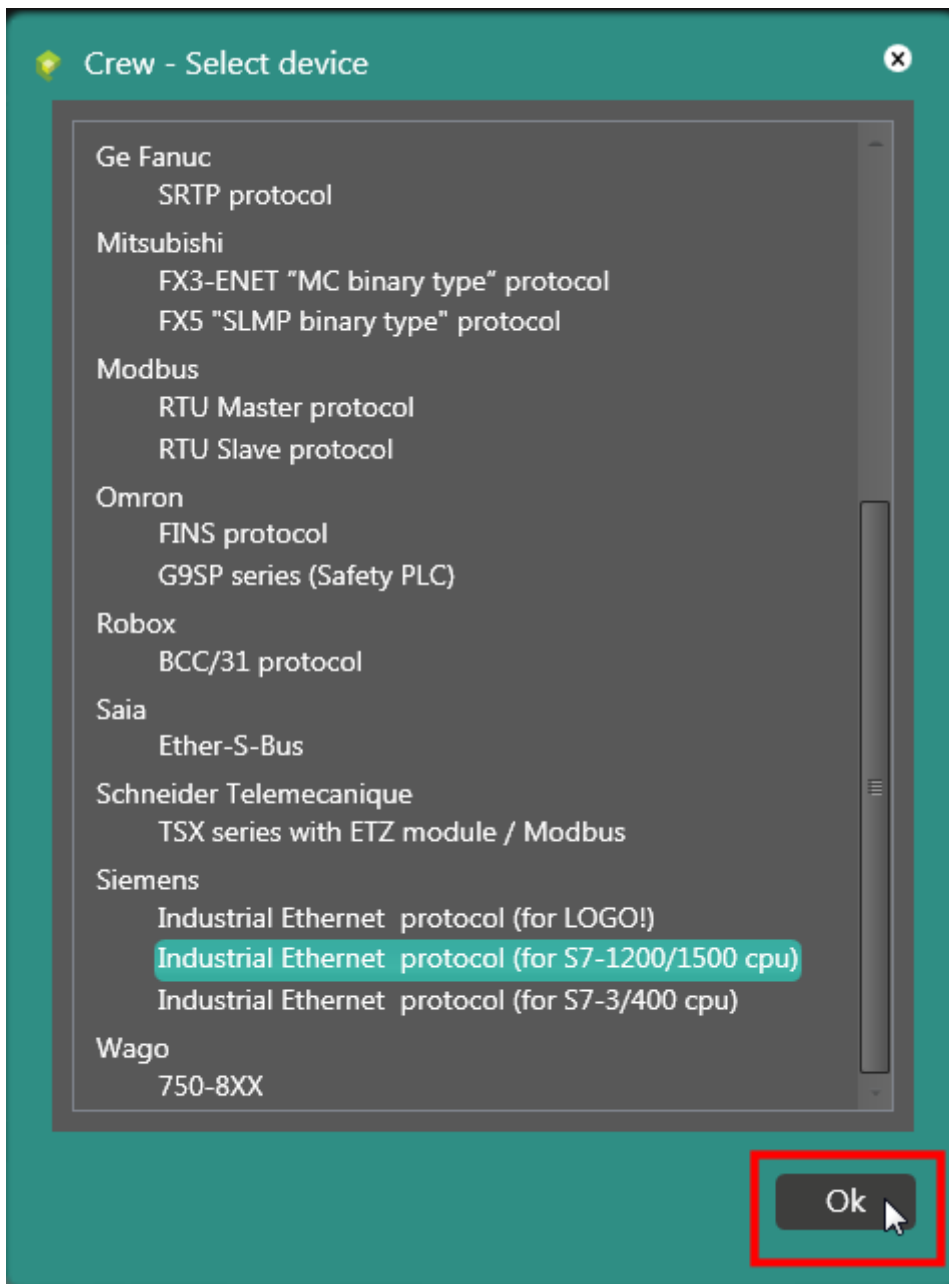
## Siemens S7-1200 1500 Ethernet



# CREW Manual



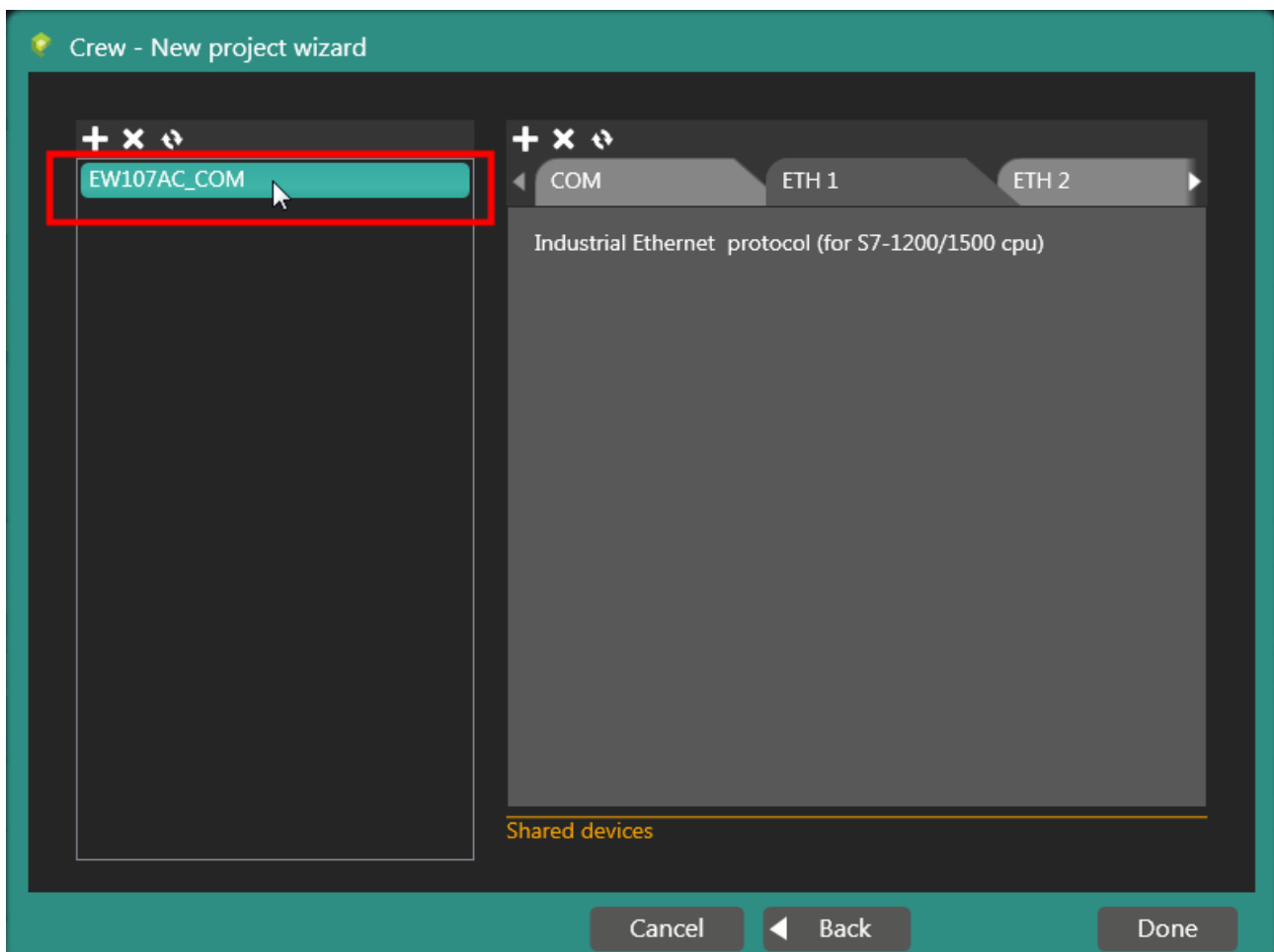
# CREW Manual



# CREW Manual

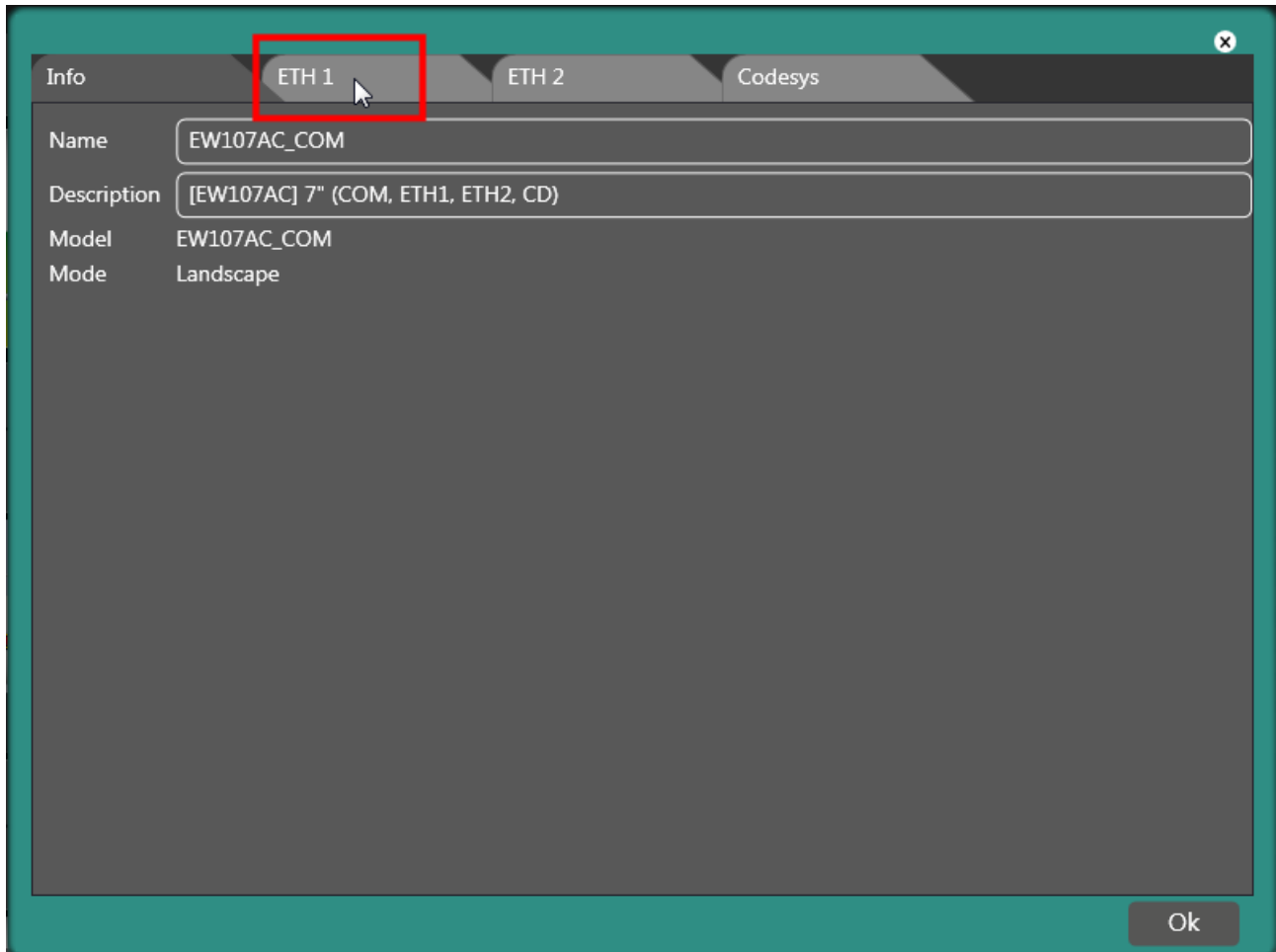
Settings on EW side:

Double click on the name of the terminal.



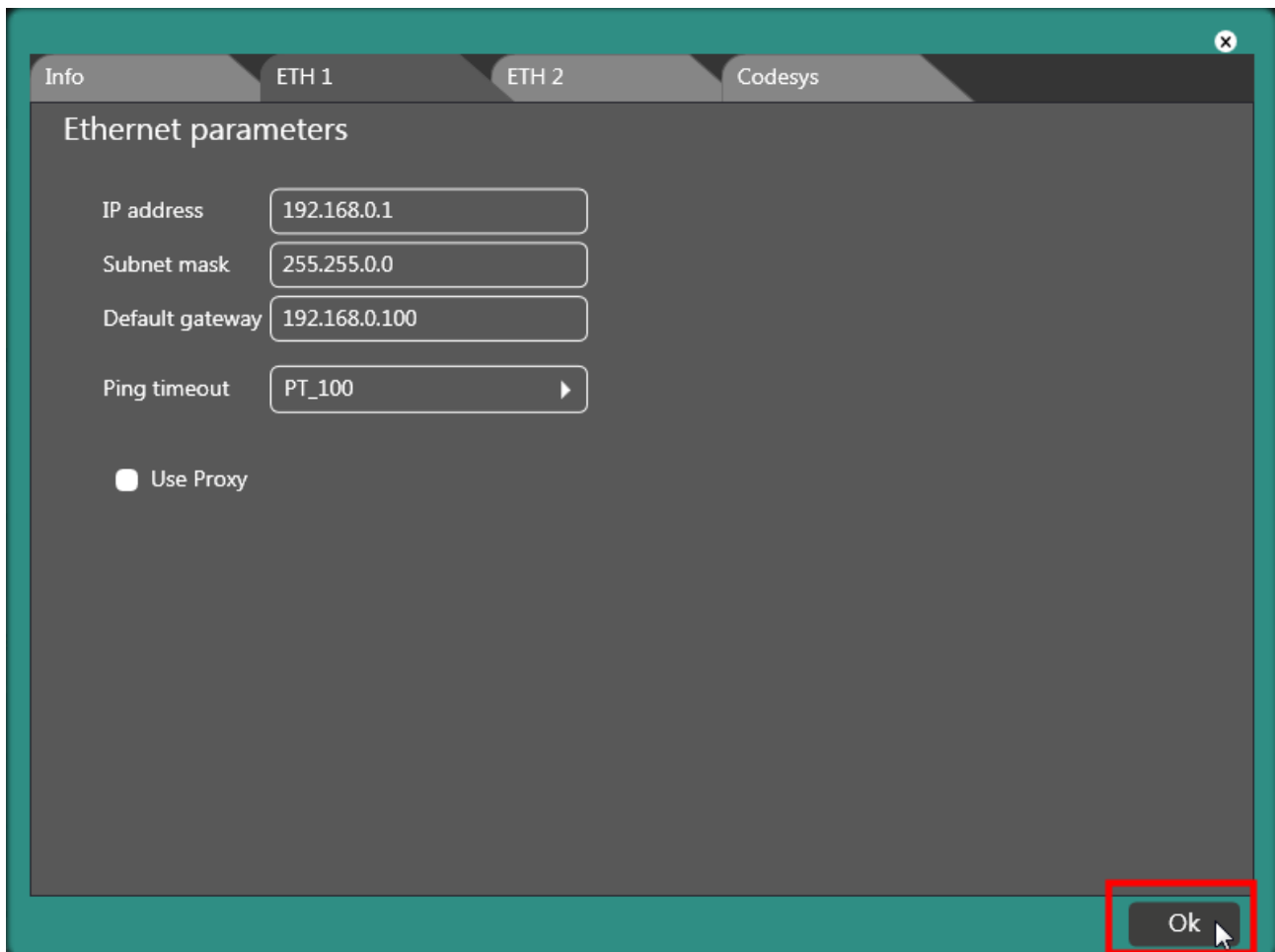
# CREW Manual

Ethernet Parameters:





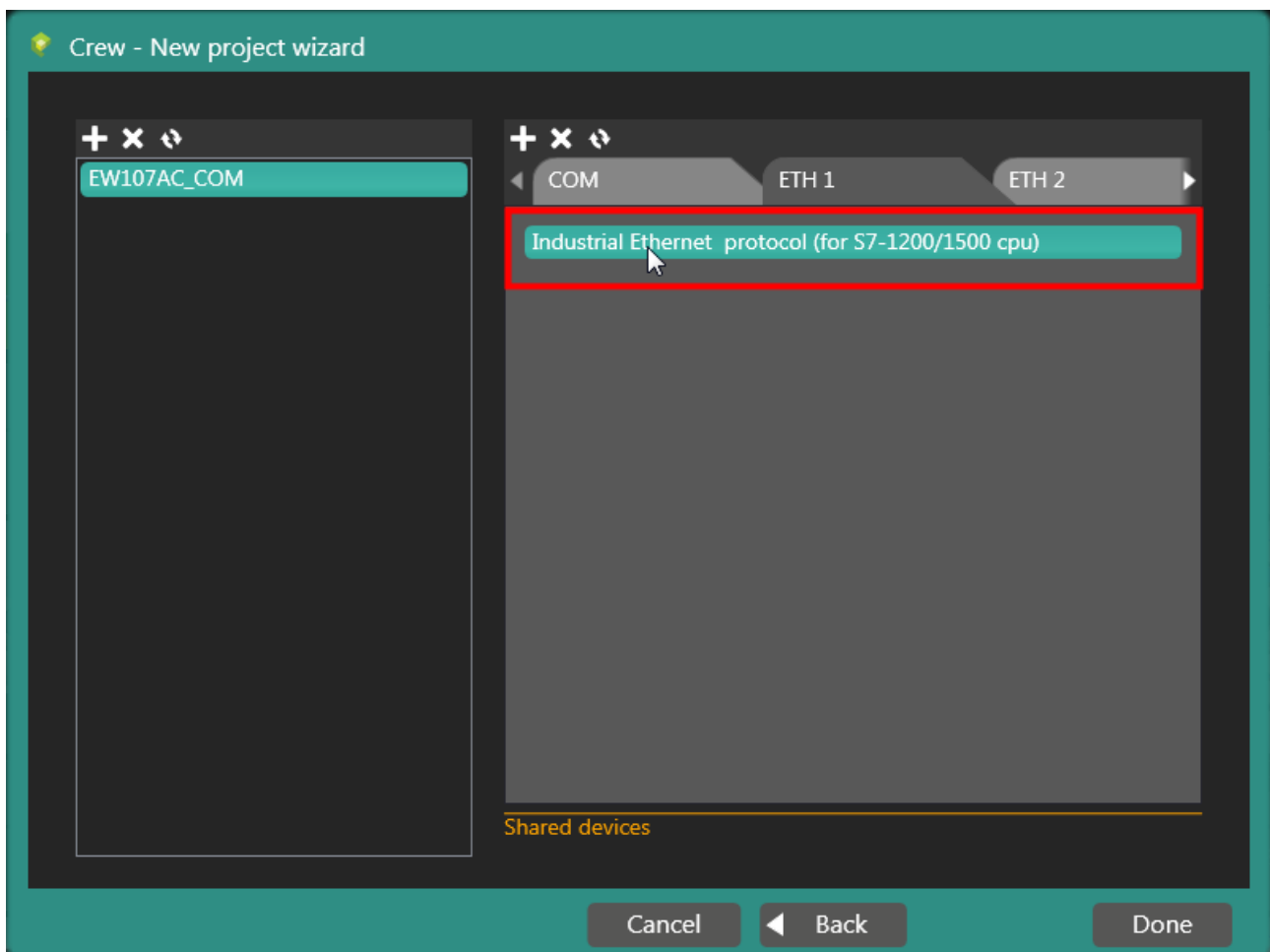
# CREW Manual



# CREW Manual

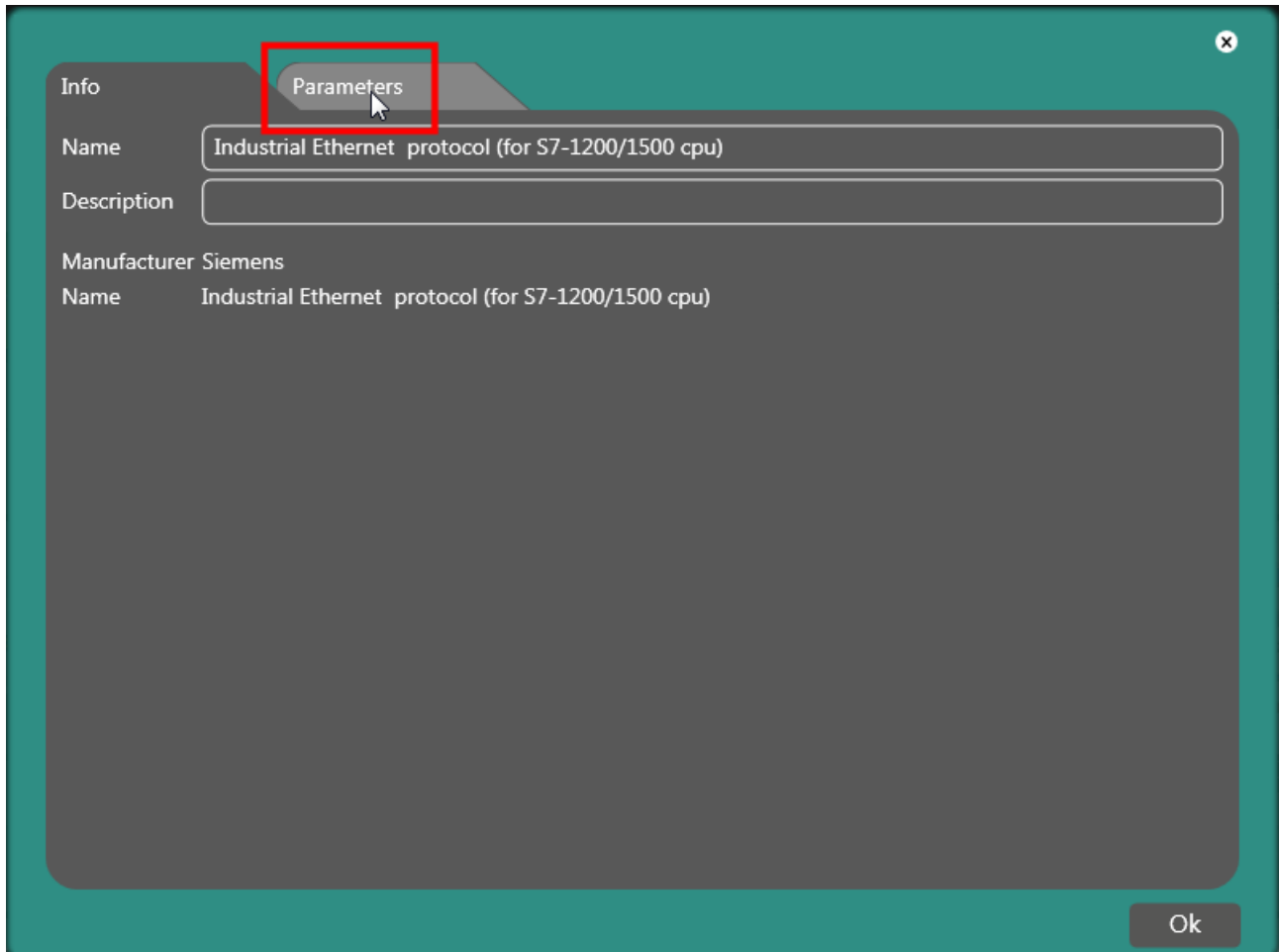
Settings on device side:

Double click on the name of the device.

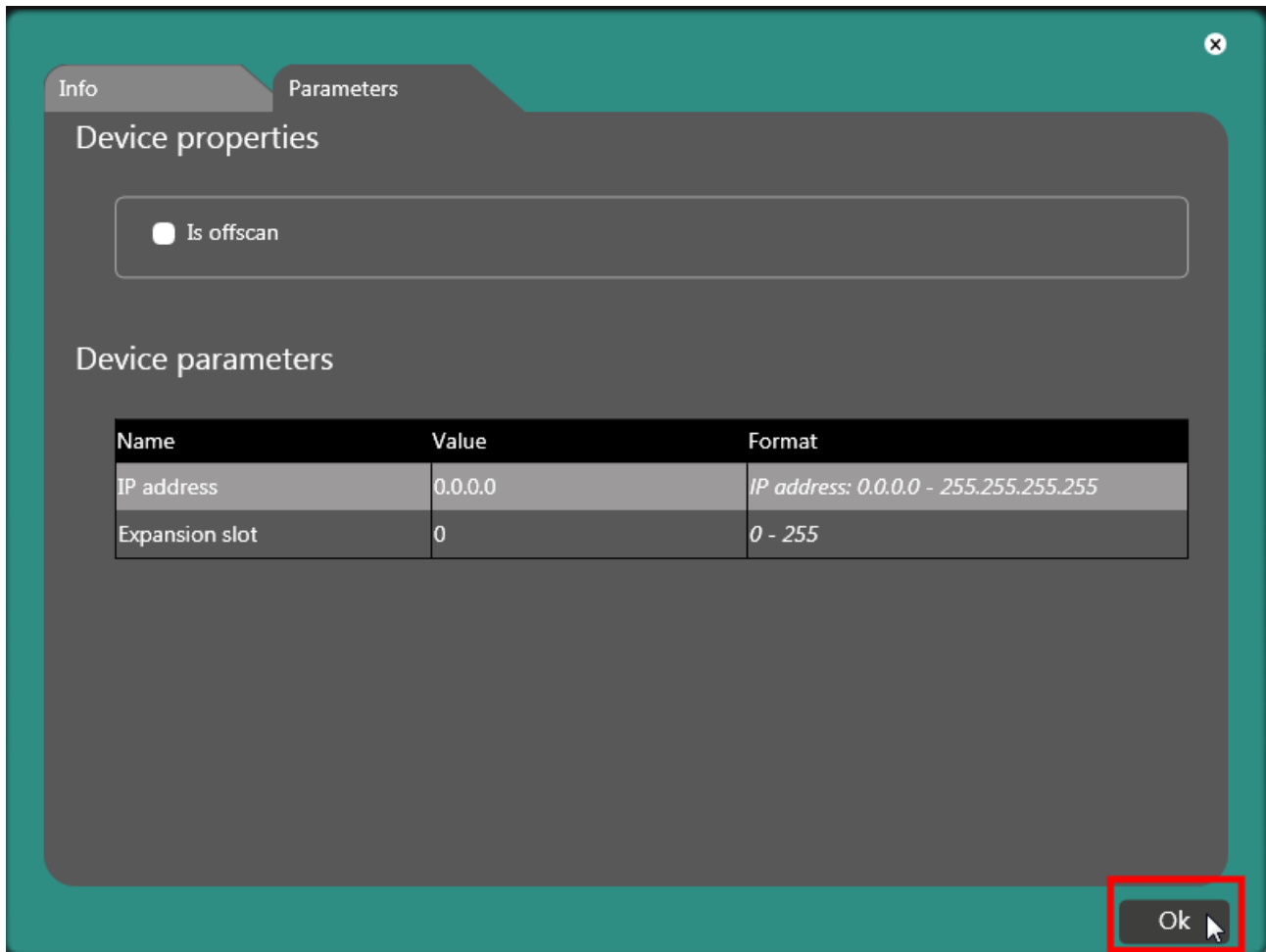


# CREW Manual

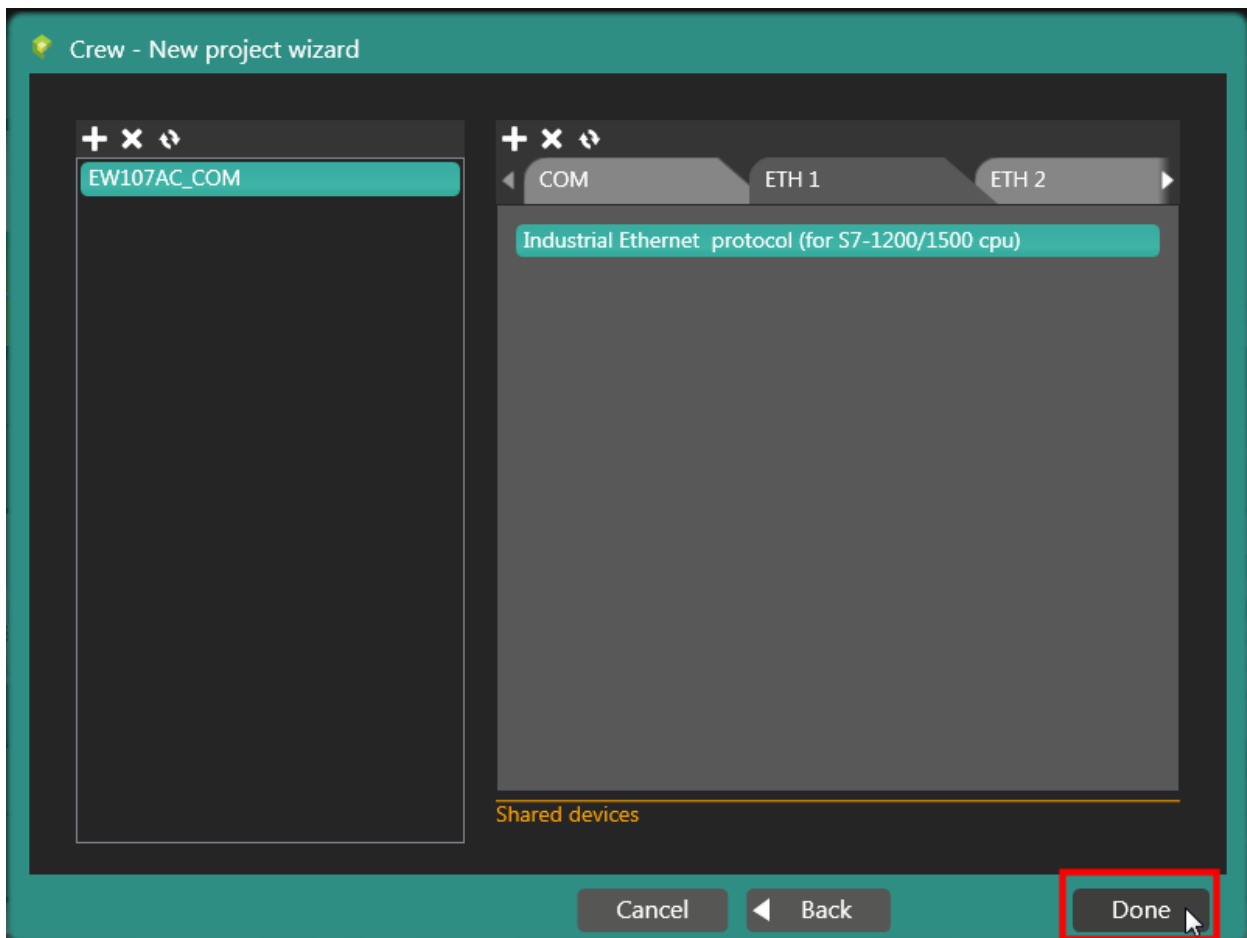
Device parameters:



# CREW Manual

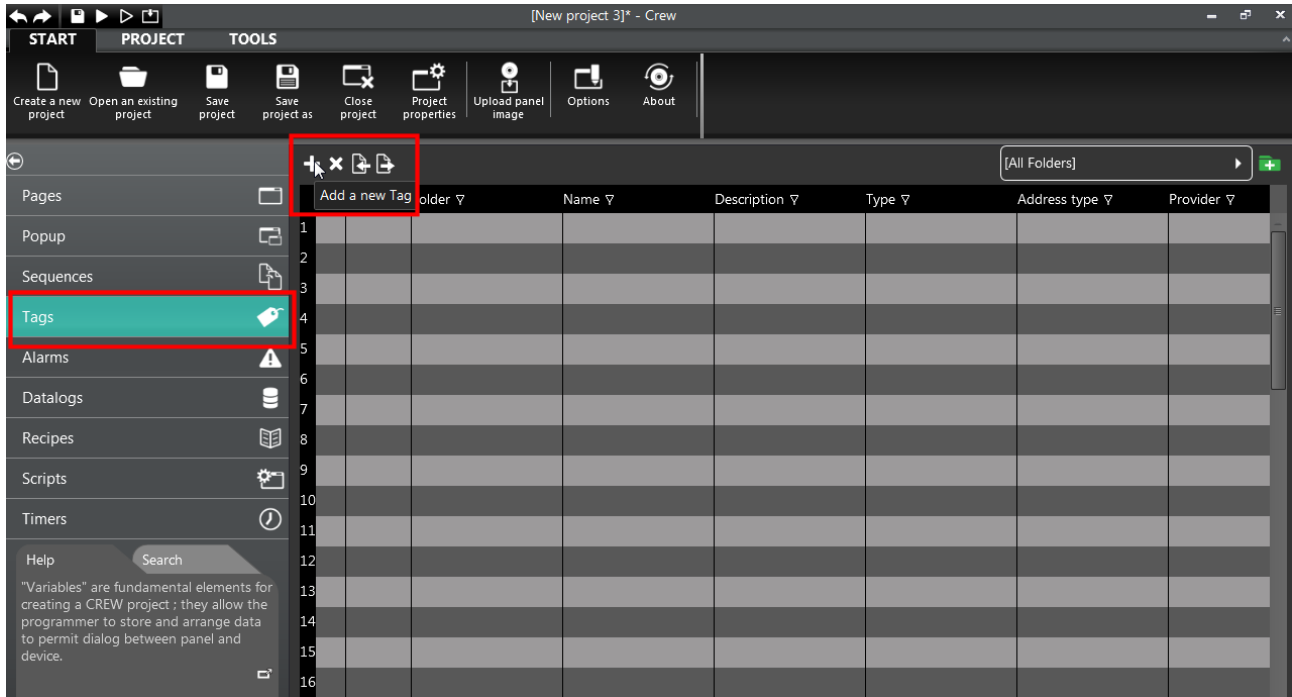


# CREW Manual



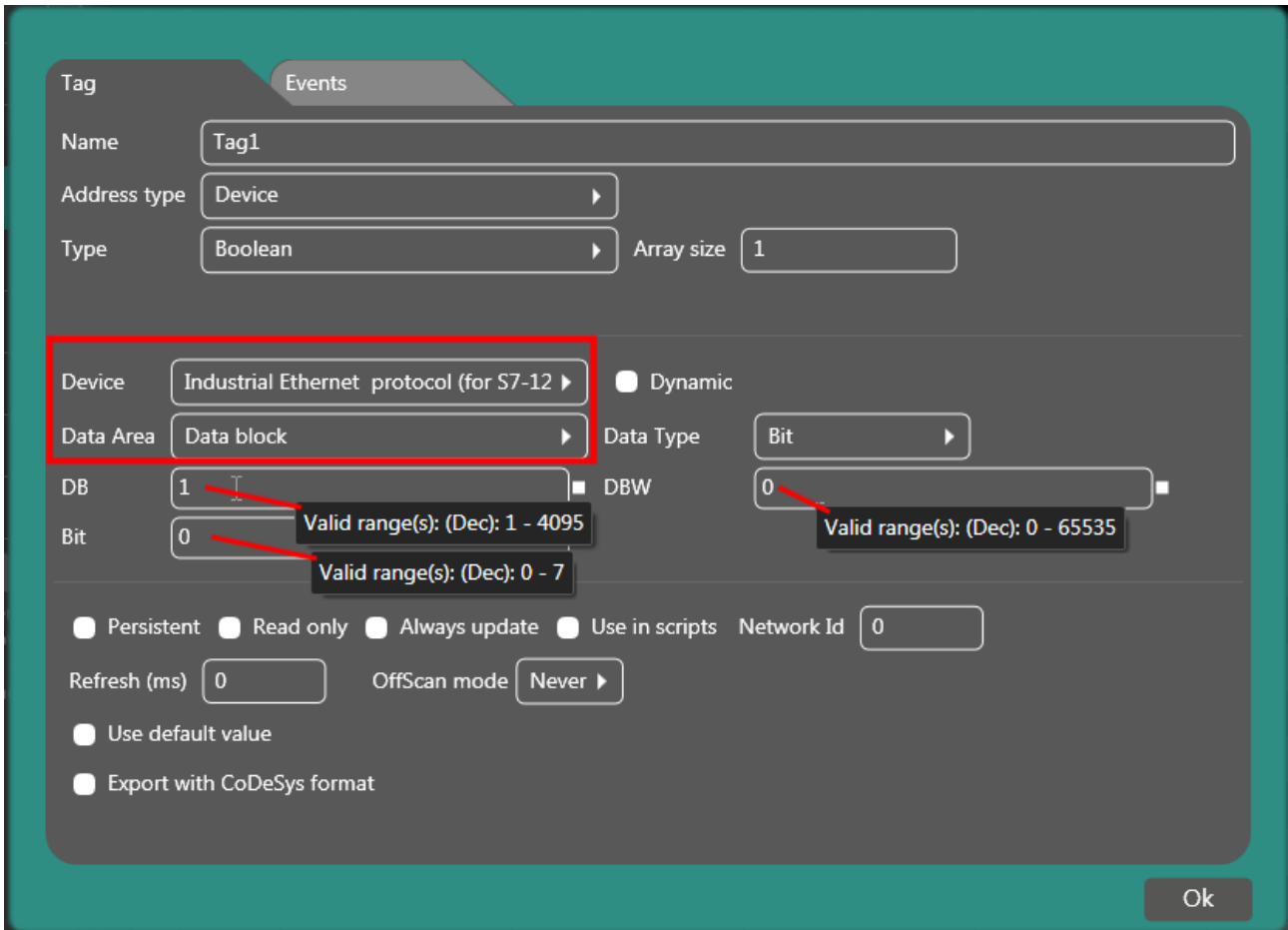
# CREW Manual

## Parameters Set up - Siemens S7-1200 1500 Ethernet



# CREW Manual

"Data block" Data Area:



Tag

Events

Name Tag1

Address type Device

Type Boolean Array size 1

Device Industrial Ethernet protocol (for S7-12)  Dynamic

Data Area Data block Data Type Bit

DB 1 DBW 0

Bit 0

Valid range(s): (Dec): 1 - 4095

Valid range(s): (Dec): 0 - 65535

Valid range(s): (Dec): 0 - 7

Persistent  Read only  Always update  Use in scripts Network Id 0

Refresh (ms) 0 OffScan mode Never

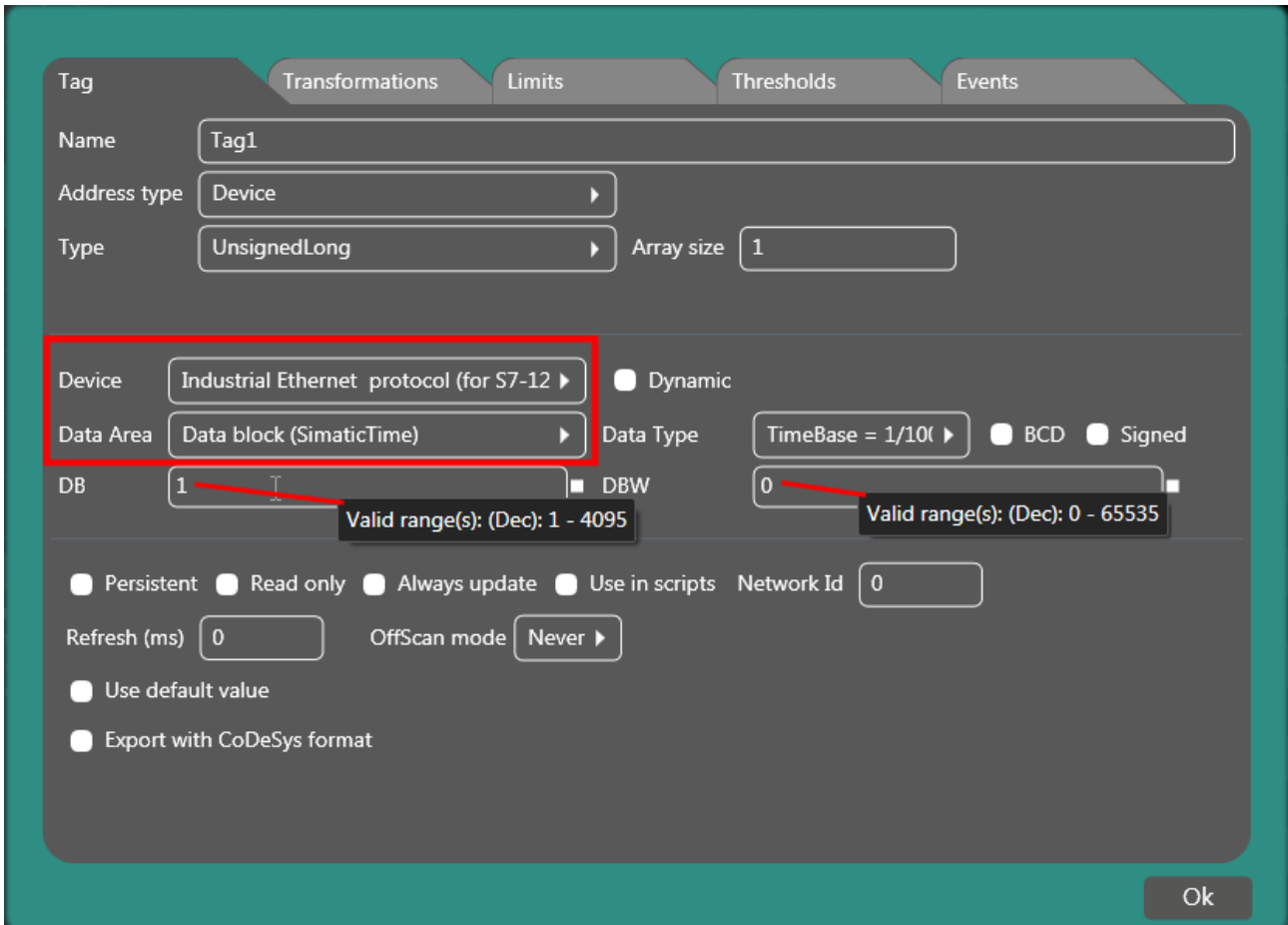
Use default value

Export with CoDeSys format

Ok

# CREW Manual

"Data block (SimaticTime)" Data Area:

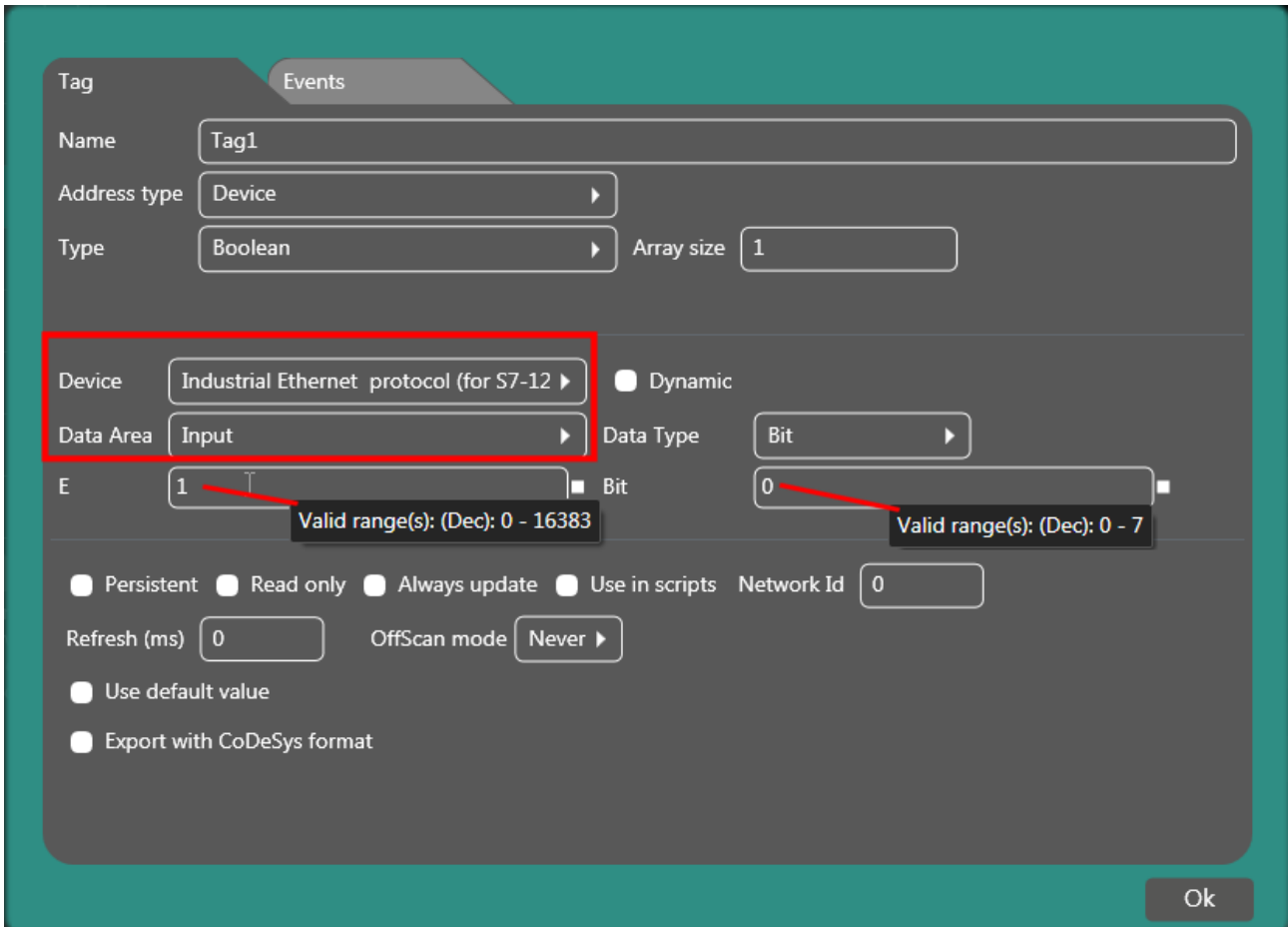


The screenshot displays the configuration window for a tag named "Tag1". The "Data Area" is set to "Data block (SimaticTime)", which is highlighted with a red box. The "DB" field is set to 1, with a tooltip indicating a valid range of 1 to 4095. The "DBW" field is set to 0, with a tooltip indicating a valid range of 0 to 65535. Other configuration options include "Device" (Industrial Ethernet protocol for S7-12), "Data Type" (TimeBase = 1/10), and various checkboxes for "Persistent", "Read only", "Always update", "Use in scripts", "Use default value", and "Export with CoDeSys format".



# CREW Manual

"Input" Data Area:



Tag

Events

Name Tag1

Address type Device

Type Boolean Array size 1

Device Industrial Ethernet protocol (for S7-12)  Dynamic

Data Area Input Data Type Bit

E 1 Bit 0

Valid range(s): (Dec): 0 - 16383

Valid range(s): (Dec): 0 - 7

Persistent  Read only  Always update  Use in scripts Network Id 0

Refresh (ms) 0 OffScan mode Never

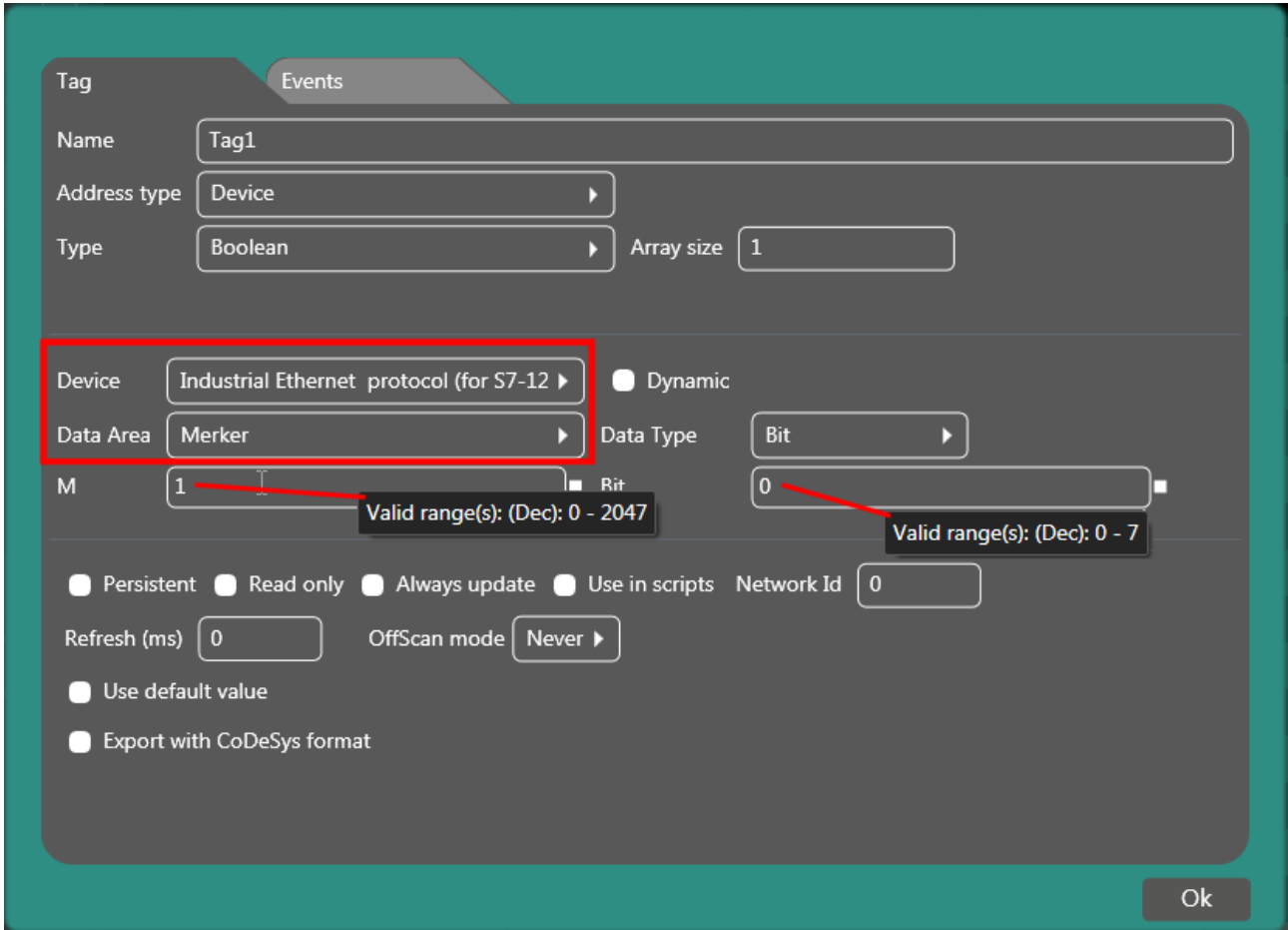
Use default value

Export with CoDeSys format

Ok

# CREW Manual

"Merker "Data Area:



Tag

Events

Name

Address type

Type  Array size

Device   Dynamic

Data Area  Data Type

M  Bit

Valid range(s): (Dec): 0 - 2047

Valid range(s): (Dec): 0 - 7

Persistent  Read only  Always update  Use in scripts Network Id

Refresh (ms)  OffScan mode

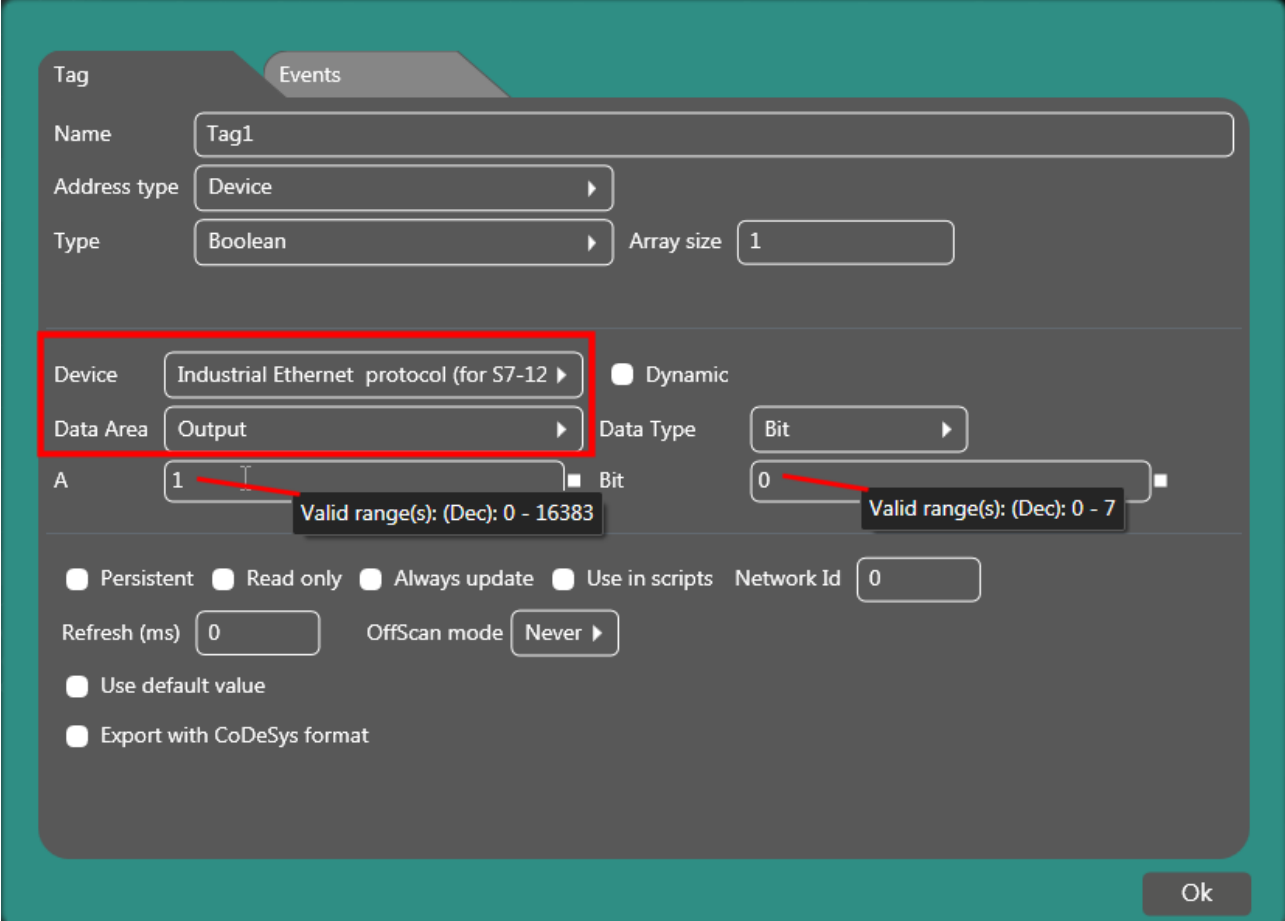
Use default value

Export with CoDeSys format

Ok

# CREW Manual

"Output" Data Area:



Tag

Events

Name

Address type

Type  Array size

Device   Dynamic

Data Area  Data Type

A   Bit

Valid range(s): (Dec): 0 - 16383

Valid range(s): (Dec): 0 - 7

Persistent  Read only  Always update  Use in scripts Network Id

Refresh (ms)  OffScan mode

Use default value

Export with CoDeSys format

Ok

# CREW Manual

Type:

In the “Type” mask is used to designate the type of datum that the tag is destined to contain. The expected data types are represented in the following table.

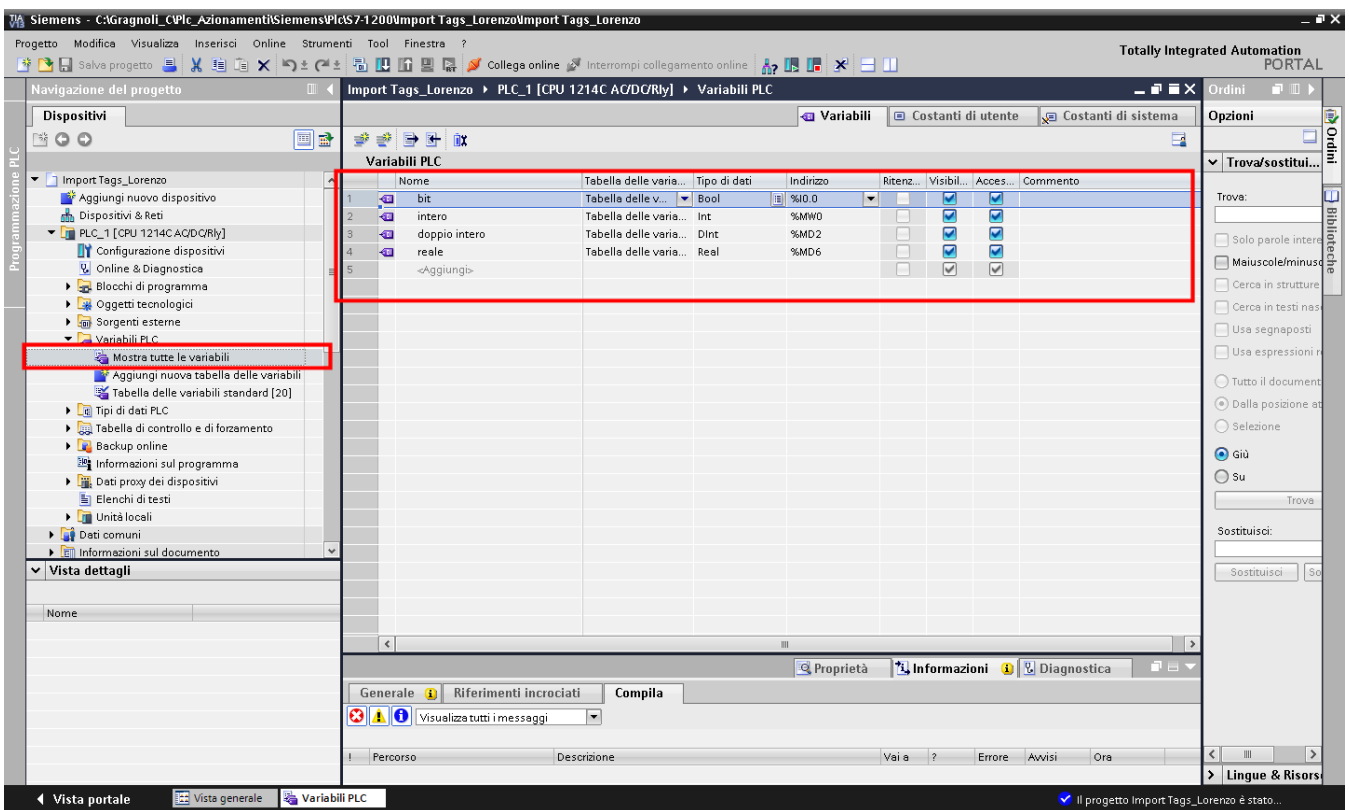
Type	Description	Range
Char	8-bit signed Integer	-128 to 127
Byte	8-bit unsigned Integer	0 to 255
Boolean	Single bit	True (1) or False(0)
Integer	16-bit signed Integer	-32,768 to 32,767
Unsigned Integer	16-bit unsigned Integer	0 to 0xFFFF
Long	32-bit signed Integer	-2,147,483,648 to 2,147,483,647
Unsigned Long	32-bit unsigned Integer	0 to 0xFFFFFFFF
Real	IEEE Floating point 32-bit single precision	-3.402823E38 to -1.401298E-45 for negative values; 1.401298E-45 to 3.402823E38 for positive values
Double	IEEE Floating point 64-bit double precision	-1.7976931348623E308 a -4.9406564584124E-324 for negative values; 4.9406564584124E-324 a 1.7976931348623E308 for positive values
String	ASCII string	ASCII String (max length 0x7FFF characters)
Array of Char	Array of 8-bit signed Integers	1 a 1024
Array of Byte	Array of 8-bit unsigned Integers	1 a 1024
Array of Boolean	Array of Single bits	1 a 1024
Array of Integer (WORD)	Array of unsigned integers	1 a 1024
Array of Unsigned Integer (WORD)	Array of unsigned integers	1 a 1024
Array of Long (DWORD)	Array of 32-bit signed Integers	1 a 1024
Array of Unsigned Long (DWORD)	Array of 32-bit unsigned Integers	1 a 1024
Array of Real	Array of IEEE Floating points 32-bit single precision	1 a 1024
Array of Double	Array of IEEE Floating points 64-bit double precision	1 a 1024
Array of String	Array of ASCII Strings	1 a 1024

# CREW Manual

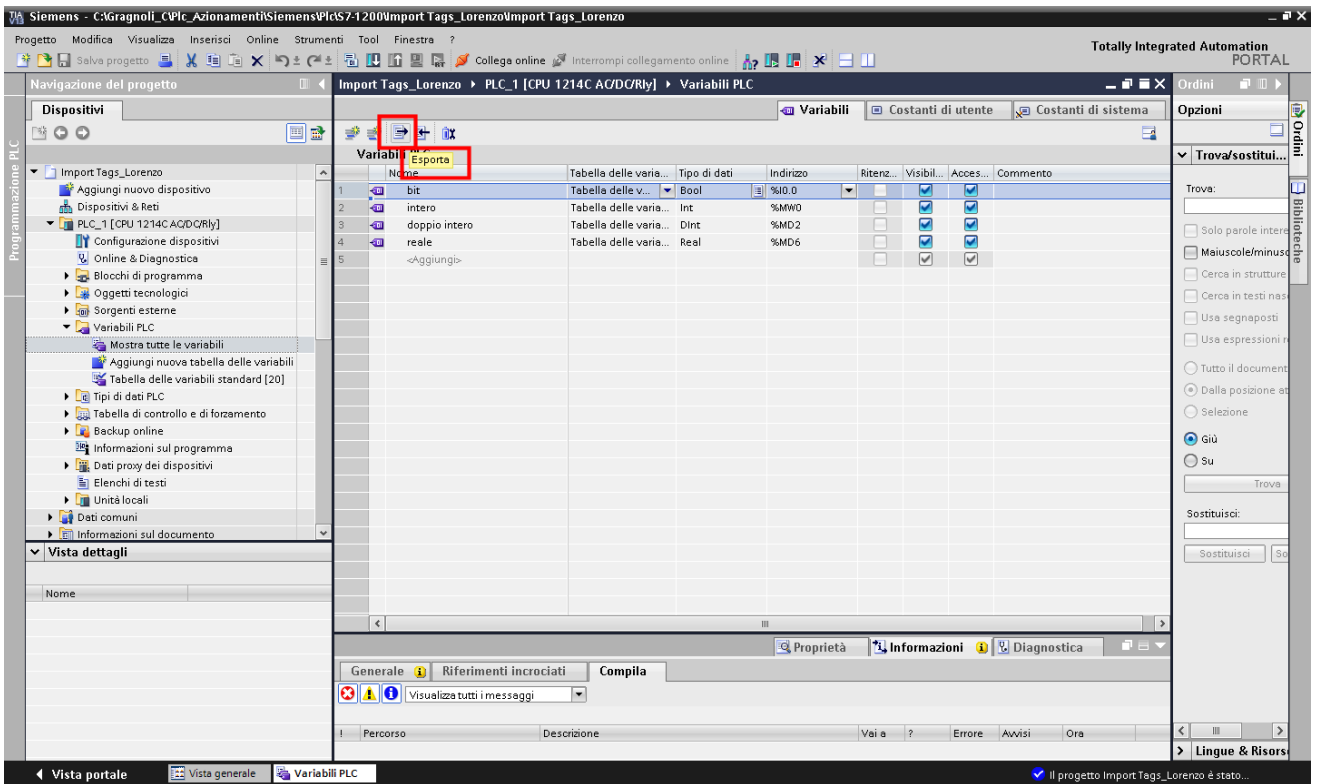
For additional information on the protocol, refer to the "[Siemens Industrial Ethernet](#)".

For more information on the variables (tags), refer to section "[Tags](#)".

## Exporting Tags from TIA PORTAL



# CREW Manual

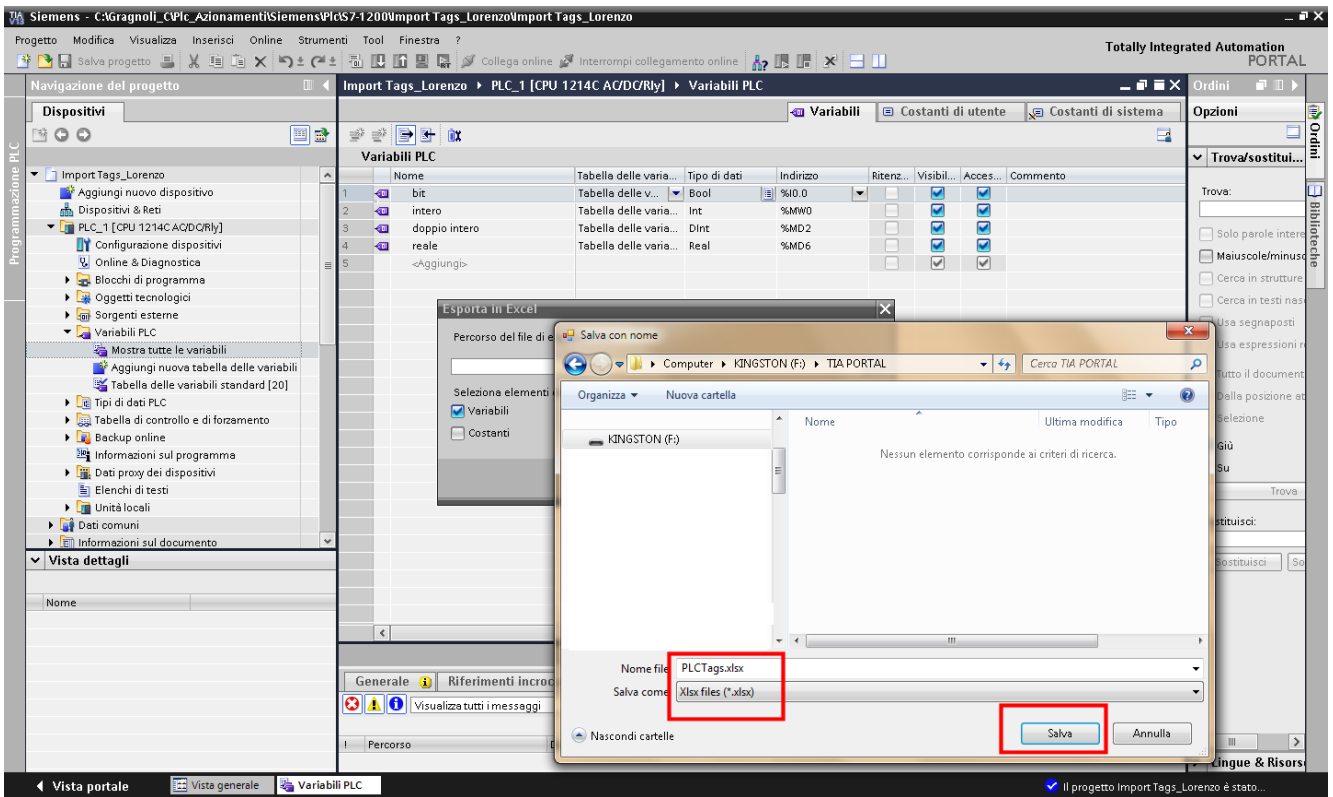


The screenshot shows the Siemens TIA Portal interface for a PLC project. The main window displays a table of PLC variables. A red box highlights the 'Variabili' and 'Esporta' buttons in the top-left corner of the table area.

Nome	Tabella delle varia...	Tipo di dati	Indirizzo	Ritenz...	Visibil...	Acces...	Commento
1 bit	Tabella delle varia...	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2 intero	Tabella delle varia...	Int	%MW0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3 doppio intero	Tabella delle varia...	Dint	%MD2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4 reale	Tabella delle varia...	Real	%MD6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5 <Aggiungi>							

At the bottom of the interface, there are tabs for 'Generale', 'Riferimenti incrociati', and 'Compila'. A status bar at the very bottom indicates 'Il progetto Import Tags\_Lorenzo è stato...'.

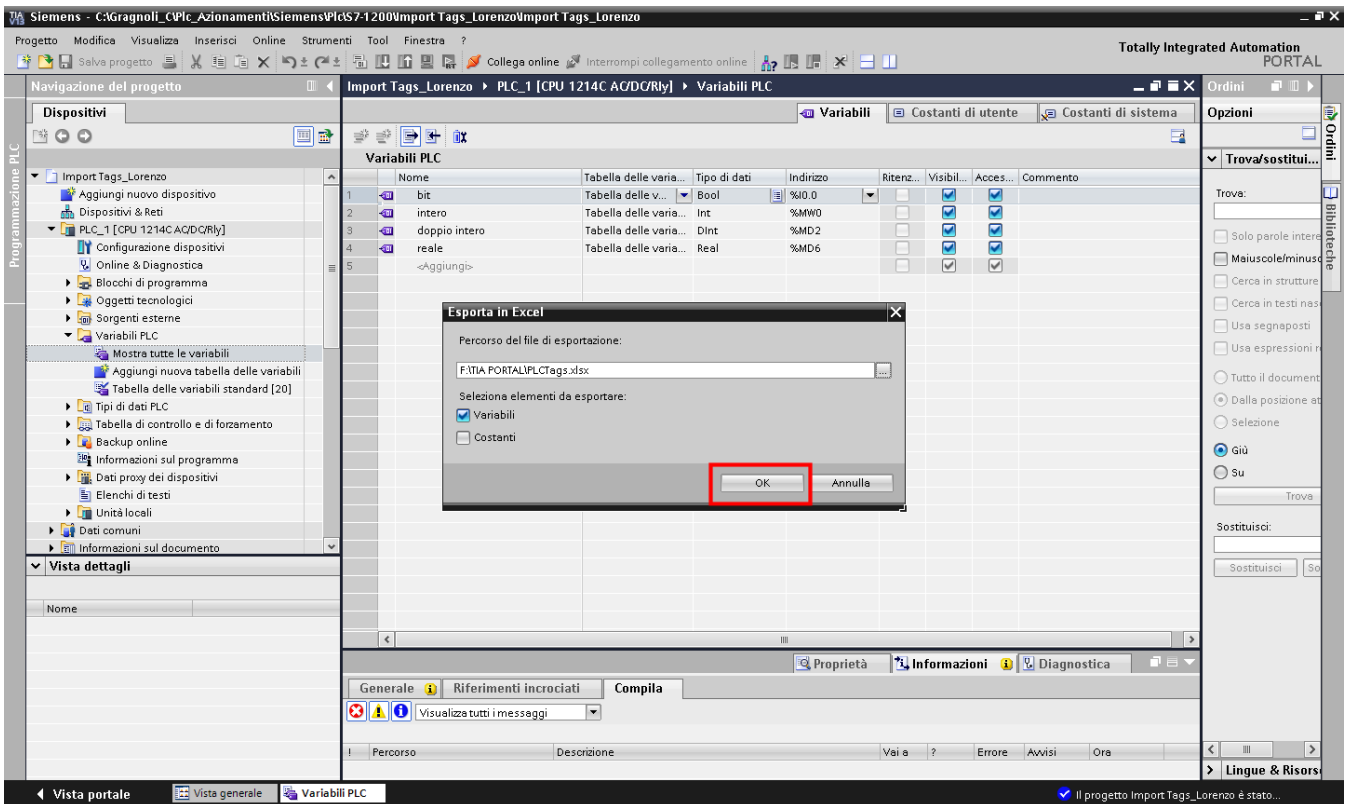
# CREW Manual



The screenshot shows the Siemens TIA Portal interface. The main window displays the 'Variabili PLC' table for a project named 'Import Tags\_Lorenzo'. The table lists variables with their names, data types, addresses, and other properties. A dialog box titled 'Esporta in Excel' is open, showing the 'Salva con nome' (Save as) window. The file name is 'PLCTags.xlsx' and the file type is 'Xlsx files (\*.xlsx)'. The 'Salva' button is highlighted with a red box.

Nome	Tabella delle varia...	Tipo di dati	Indirizzo	Ritenz...	Visibil...	Acces...	Commento
1 bit	Tabella delle v...	Bool	%I0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2 intero	Tabella delle varia...	Int	%MW0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3 doppio intero	Tabella delle varia...	DInt	%MD2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4 reale	Tabella delle varia...	Real	%MD6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5 <Aggiungi>							

# CREW Manual



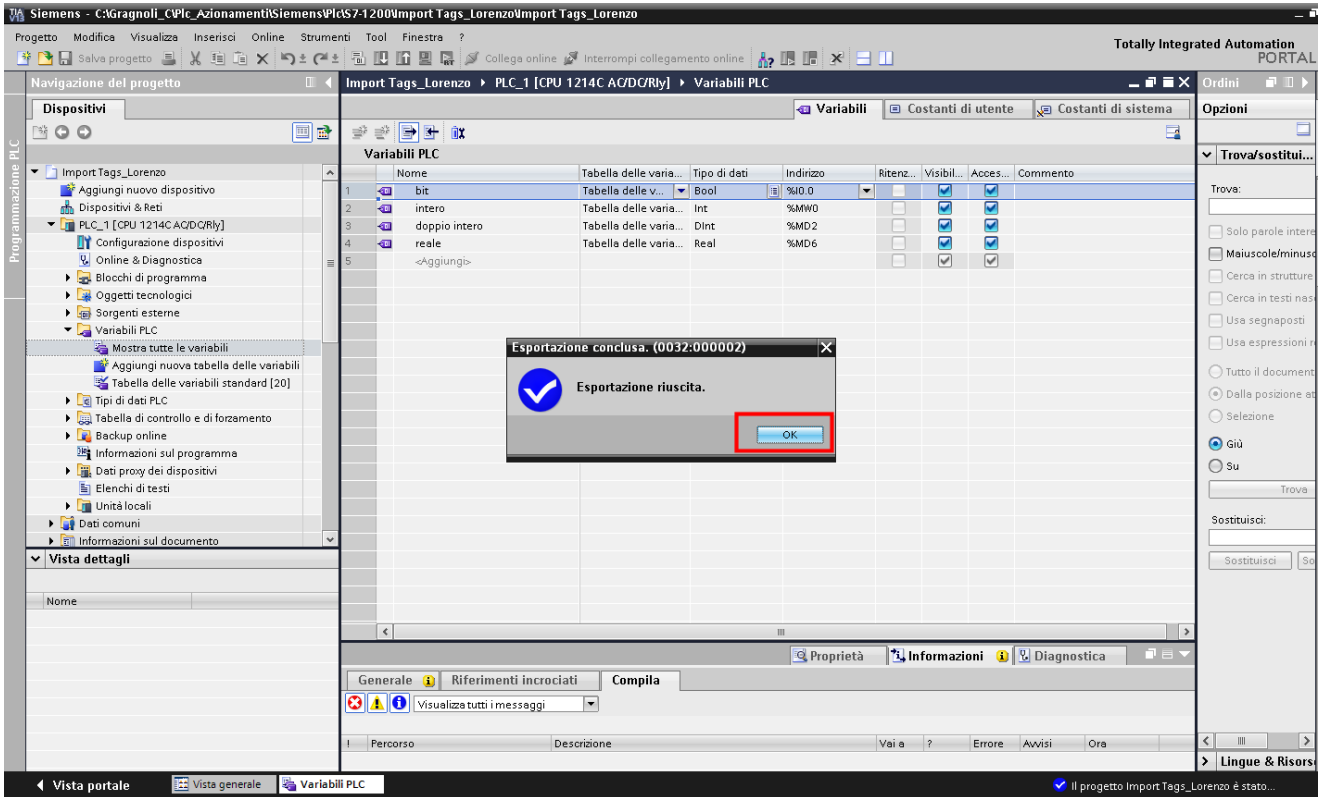
The screenshot shows the Siemens TIA Portal interface. The main window displays the 'Variabili PLC' table with the following data:

	Nome	Tabella delle varia...	Tipo di dati	Indirizzo	Ritenz...	Visibil...	Acces...	Commento
1	bit	Tabella delle v...	Bool	%I0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	intero	Tabella delle varia...	Int	%MW0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	doppio intero	Tabella delle varia...	Dint	%MD2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	reale	Tabella delle varia...	Real	%MD6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	<Aggiungi>					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

An 'Esporta in Excel' dialog box is open, showing the file path 'F:\TIA PORTAL\PLCTags.xlsx' and the option 'Variabili' selected under 'Selezione elementi da esportare'. The 'OK' button is highlighted with a red rectangle.



# CREW Manual

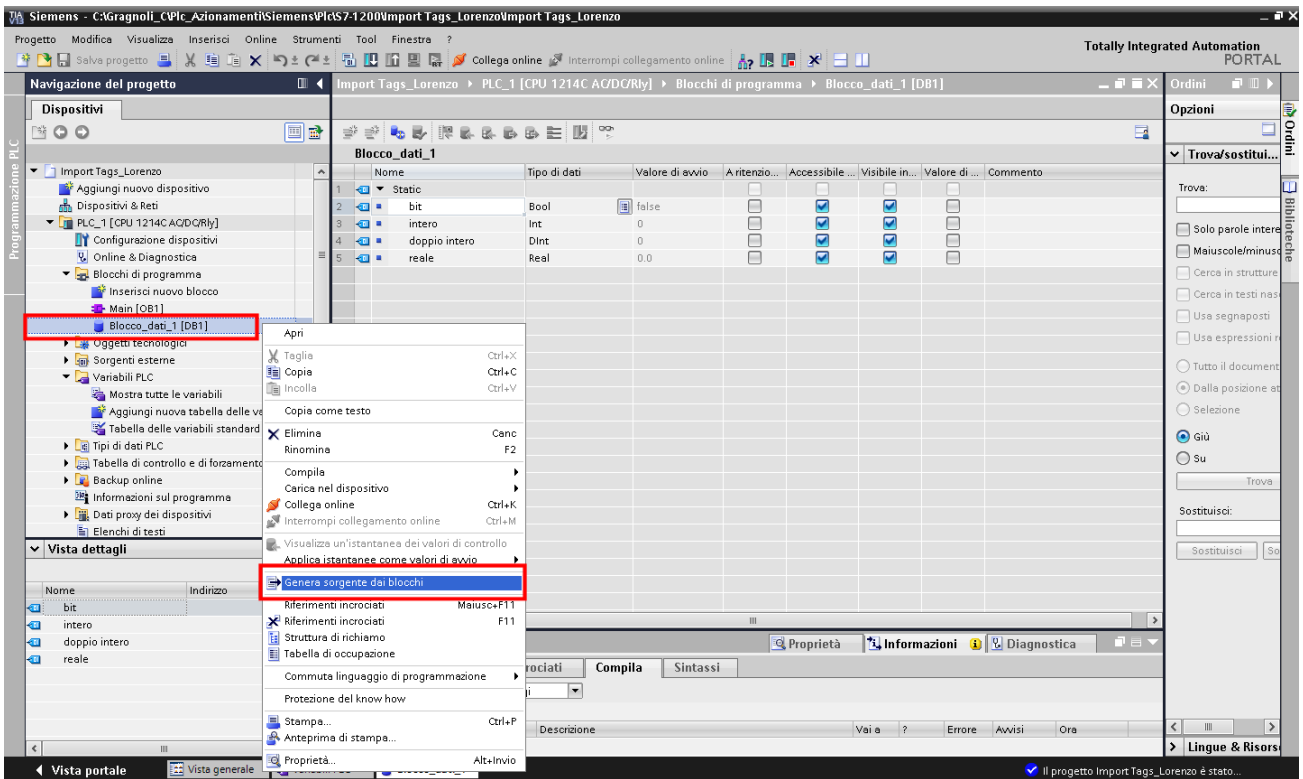


The screenshot shows the Siemens TIA Portal interface. The main window displays the 'Variabili PLC' table for the project 'Import Tags\_Lorenzo'. The table contains the following data:

Nome	Tabella delle varia...	Tipo di dati	Indirizzo	Ritenz...	Visibil...	Acces...	Commento
1 bit	Tabella delle varia...	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2 intero	Tabella delle varia...	Int	%MW0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3 doppio intero	Tabella delle varia...	Dint	%MD2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4 reale	Tabella delle varia...	Real	%MD6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5 <-Aggiungi>				<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

A dialog box titled 'Esportazione conclusa. (0032:000002)' is displayed in the center of the screen, indicating a successful export. The dialog contains a blue checkmark icon and the text 'Esportazione riuscita.' with an 'OK' button highlighted by a red rectangle.

# CREW Manual

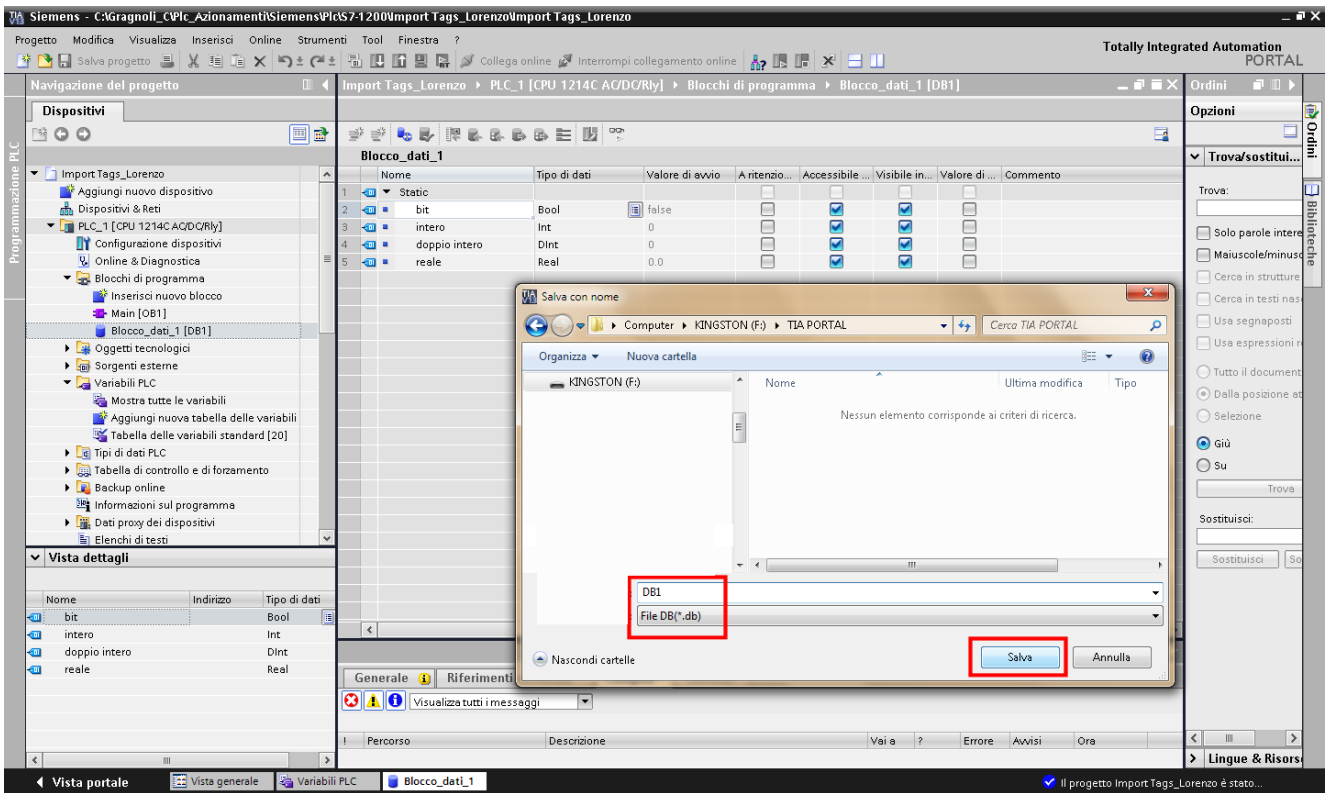


The screenshot shows the Siemens SIMATIC Manager interface. The main window displays the configuration for the 'Blokco\_dati\_1' data block. The table below represents the data types defined in this block:

Nome	Tipo di dati	Valore di avvio	A ritenzio...	Accessibile...	Visibile in...	Valore di...	Commento
1	Static						
2	bit	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	intero	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	doppio intero	DInt	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	reale	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

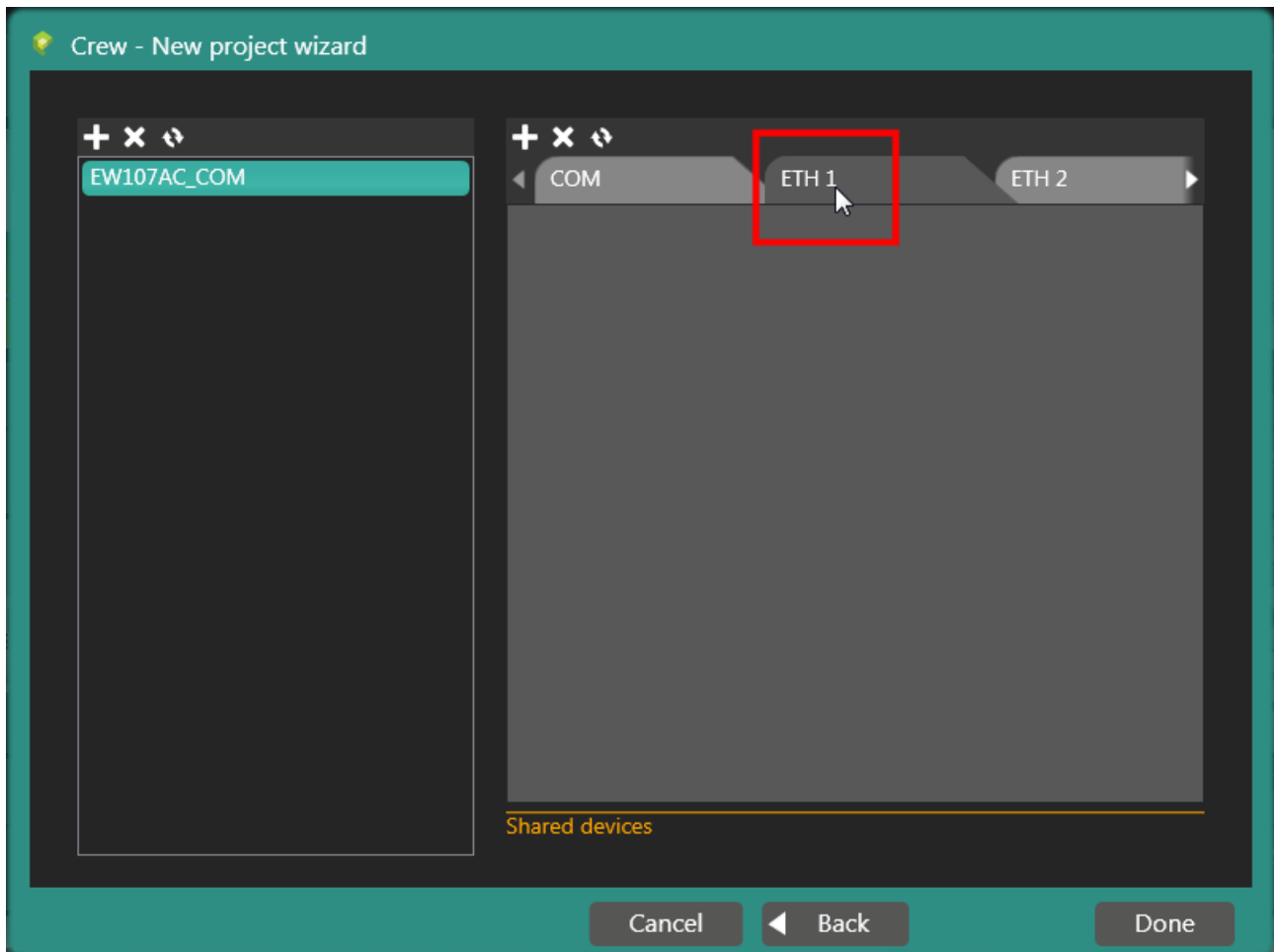
The context menu is open over the 'Blokco\_dati\_1 [DB1]' entry in the project tree. The 'Genera sorgente dai blocchi' option is highlighted. Other menu items include 'Apri', 'Taglia', 'Copia', 'Incolla', 'Copia come testo', 'Elimina', 'Rinomina', 'Compila', 'Carica nel dispositivo', 'Collega online', 'Interrompi collegamento online', 'Visualizza un'istananea dei valori di controllo', 'Applica istantanea come valori di avvio', 'Riferimenti incrociati', 'Struttura di richiamo', 'Tabella di occupazione', 'Commuta linguaggio di programmazione', 'Protezione del know how', 'Stampa...', and 'Anteprima di stampa...'.

# CREW Manual

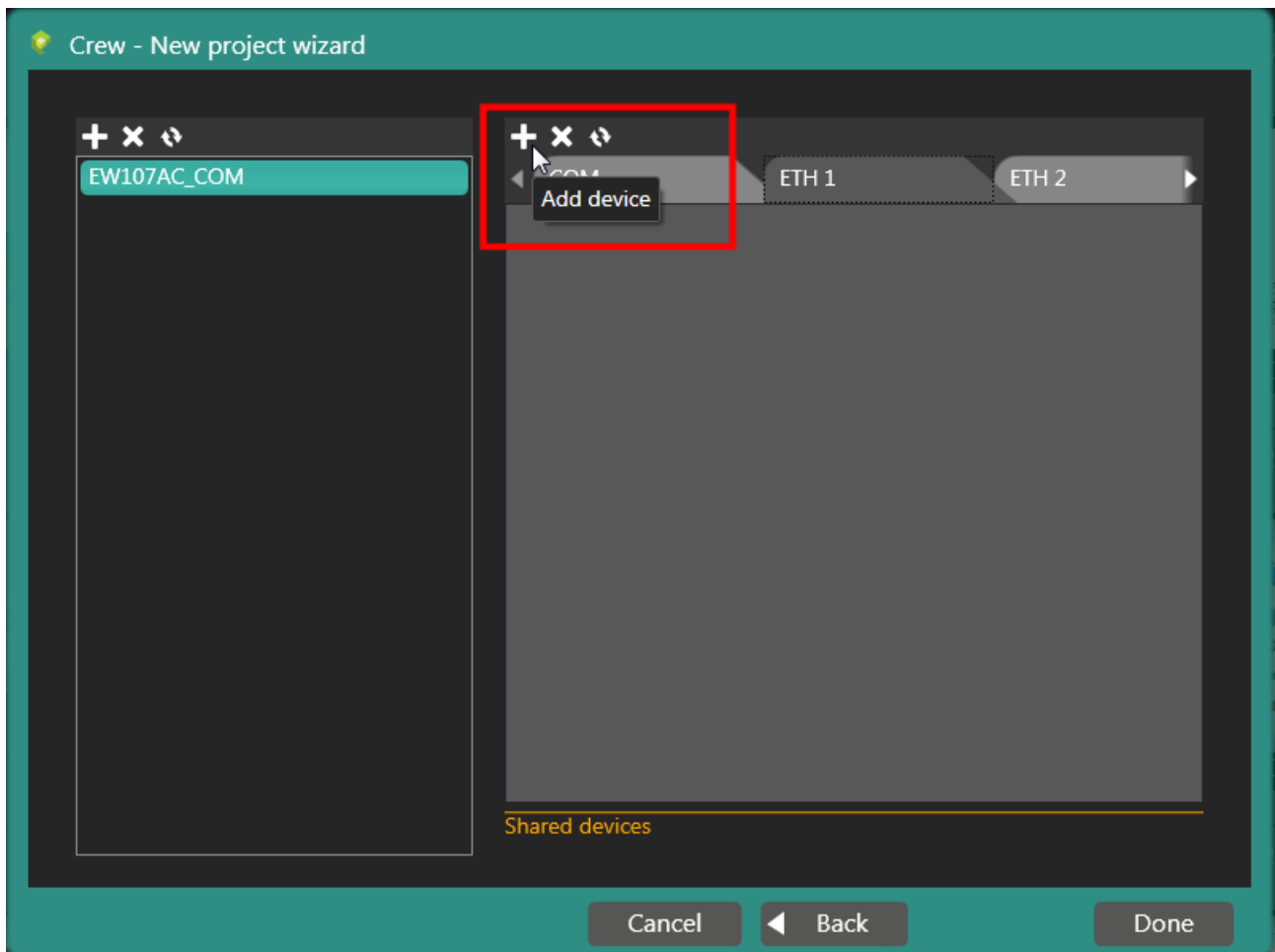


# CREW Manual

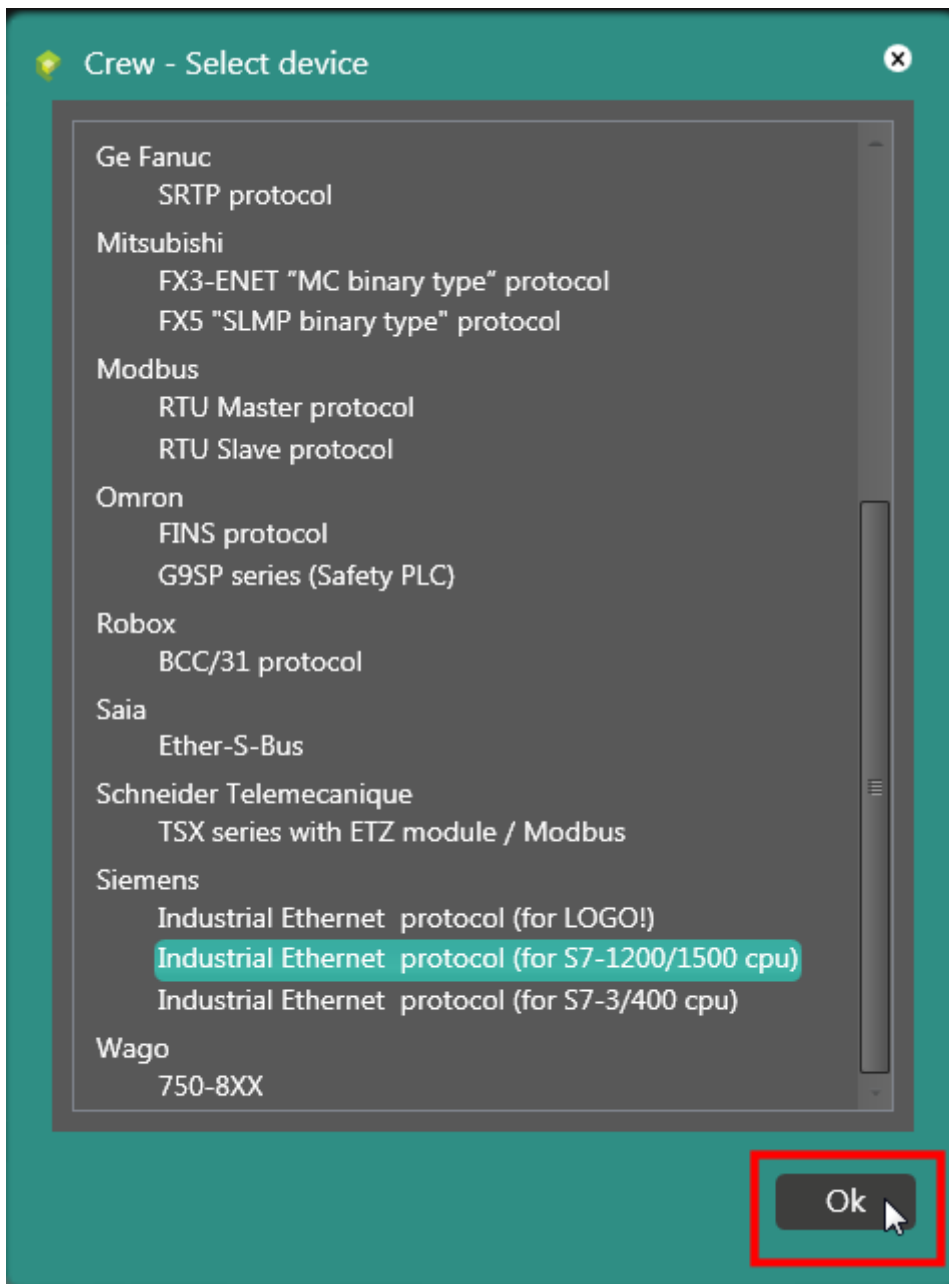
## Importing TIA PORTAL Tags



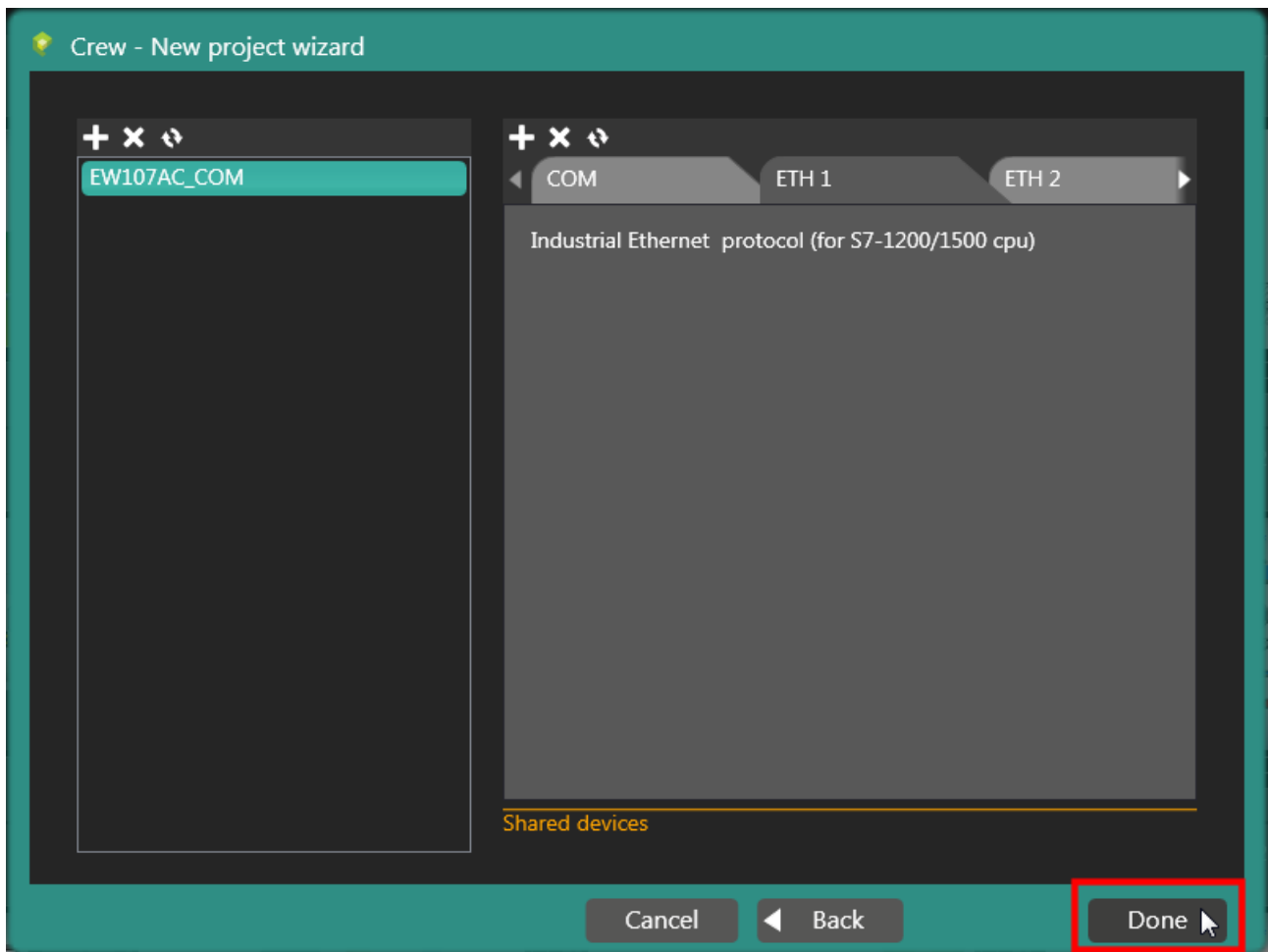
# CREW Manual



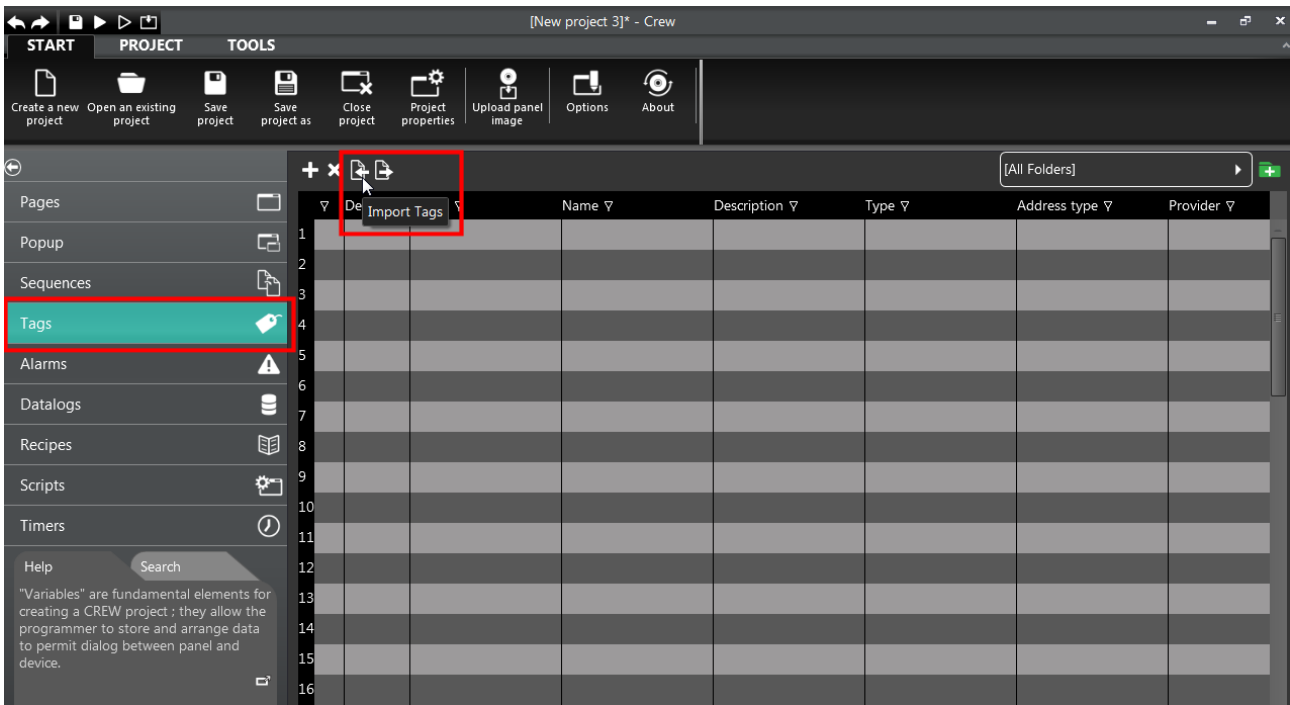
# CREW Manual



# CREW Manual

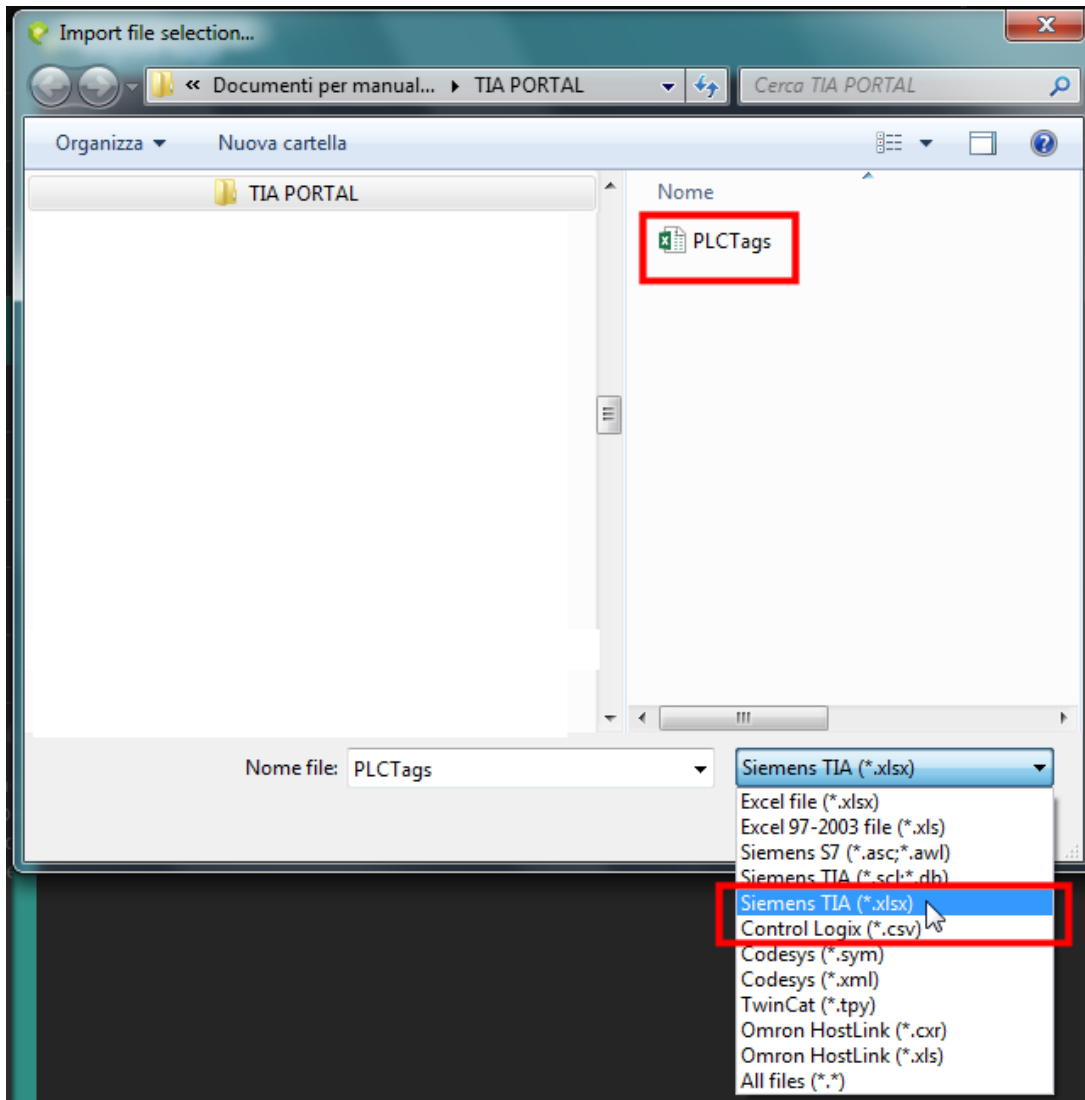


# CREW Manual





# CREW Manual



# CREW Manual

Import tags

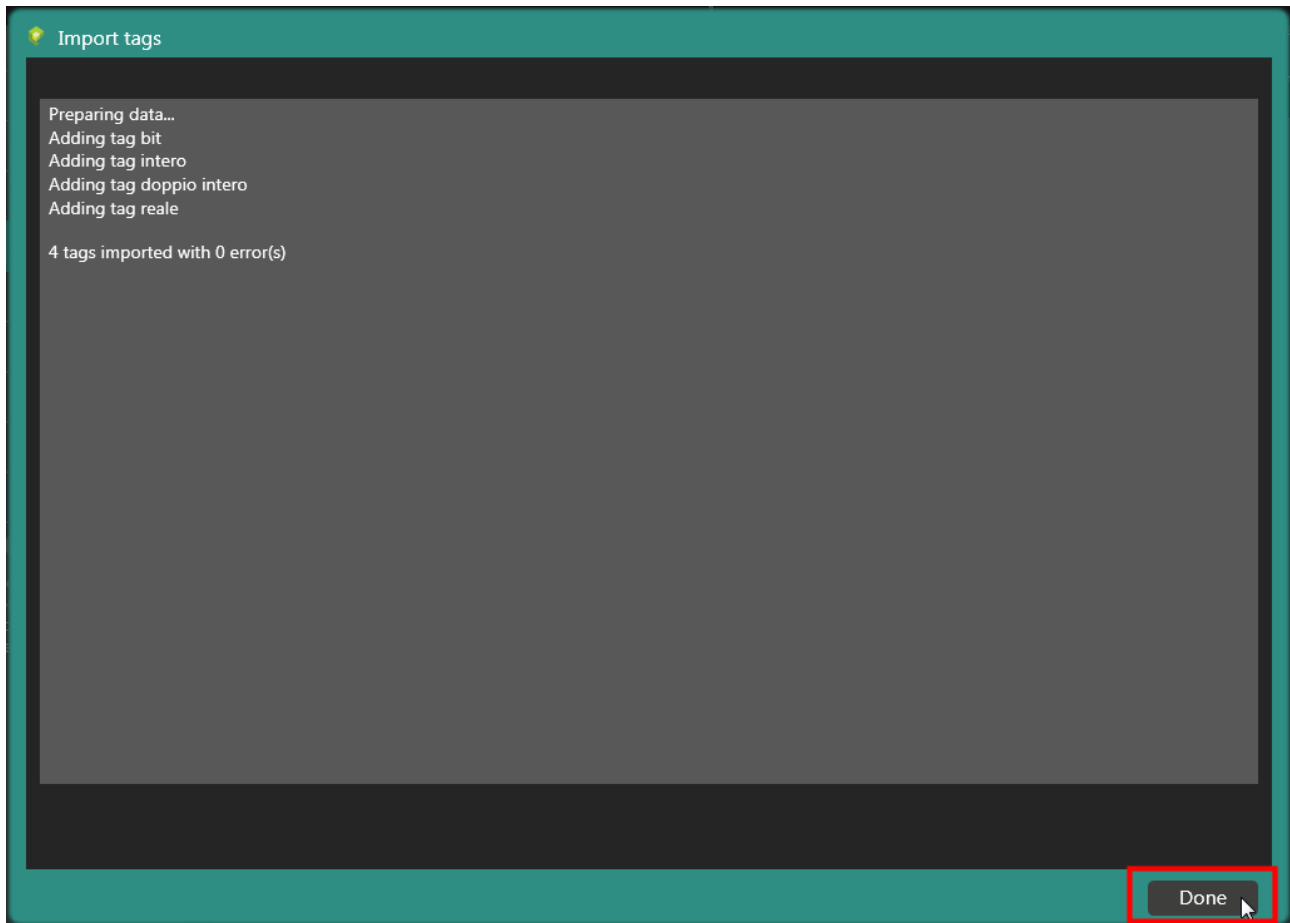
Tags to import

Name	Folder	TagType	AddressType	Description
<input checked="" type="checkbox"/> bit		Boolean	Device	
<input checked="" type="checkbox"/> intero		Integer	Device	
<input checked="" type="checkbox"/> doppio intero		Long	Device	
<input checked="" type="checkbox"/> reale		Real	Device	

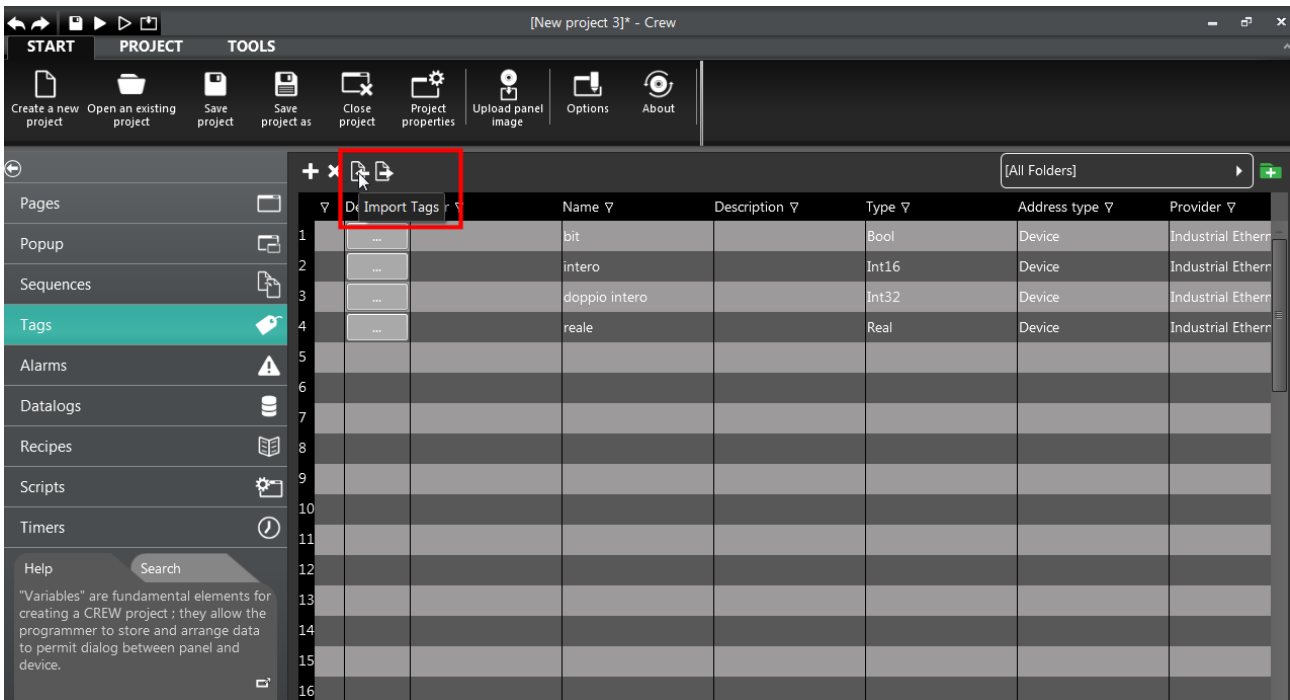
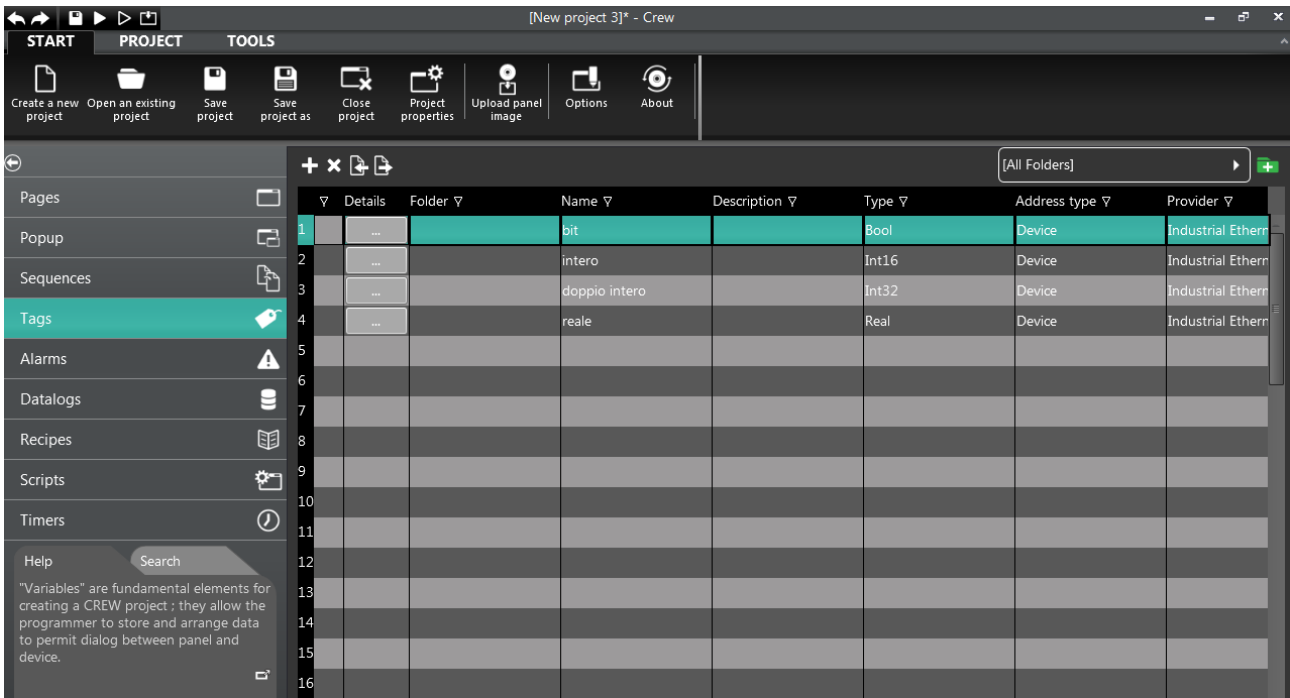
Select all   Unselect all   Destination device: Industrial Ethernet protocol (for S7-1200/1)    Override the existing objects

Cancel   Next

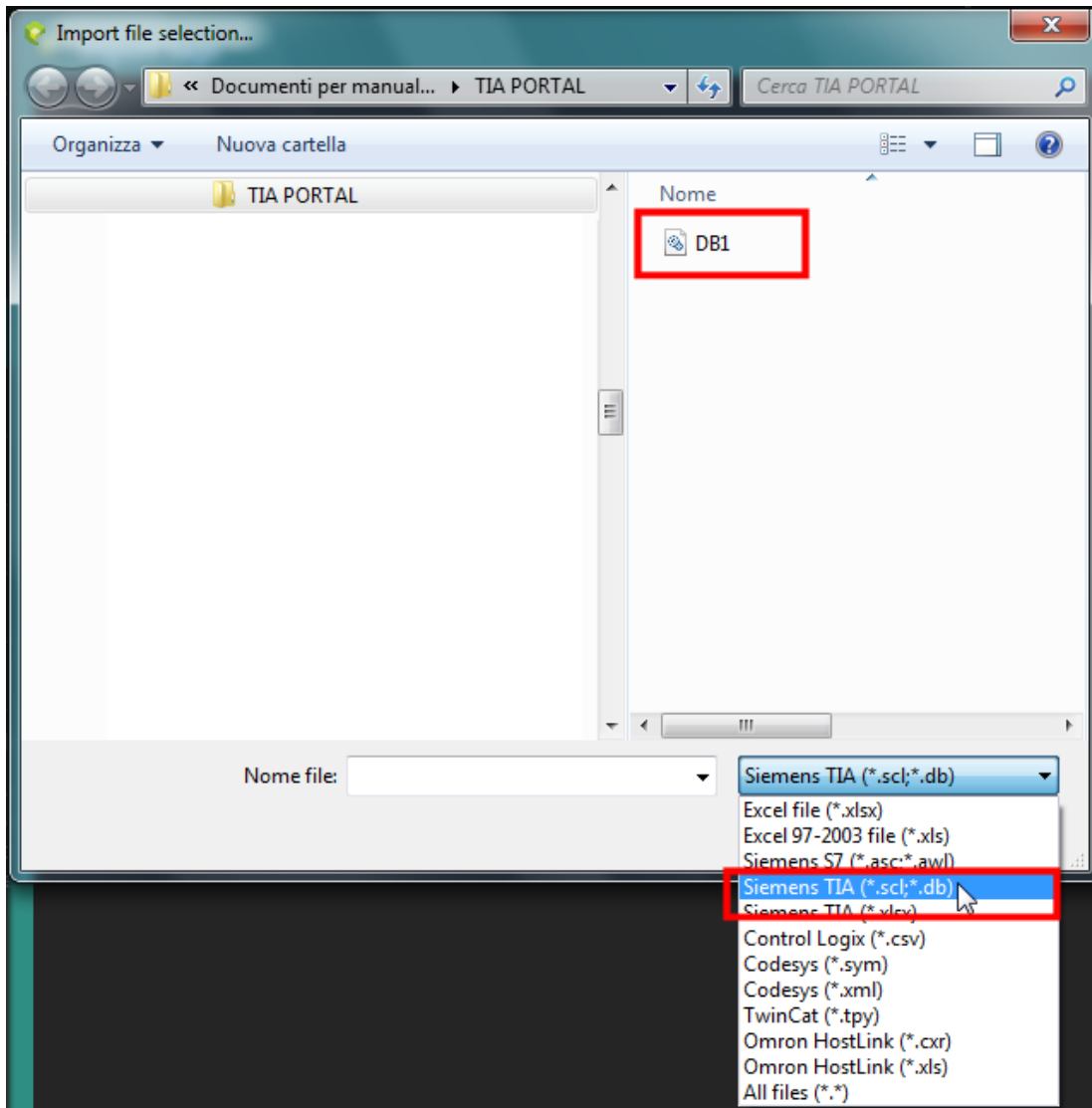
# CREW Manual



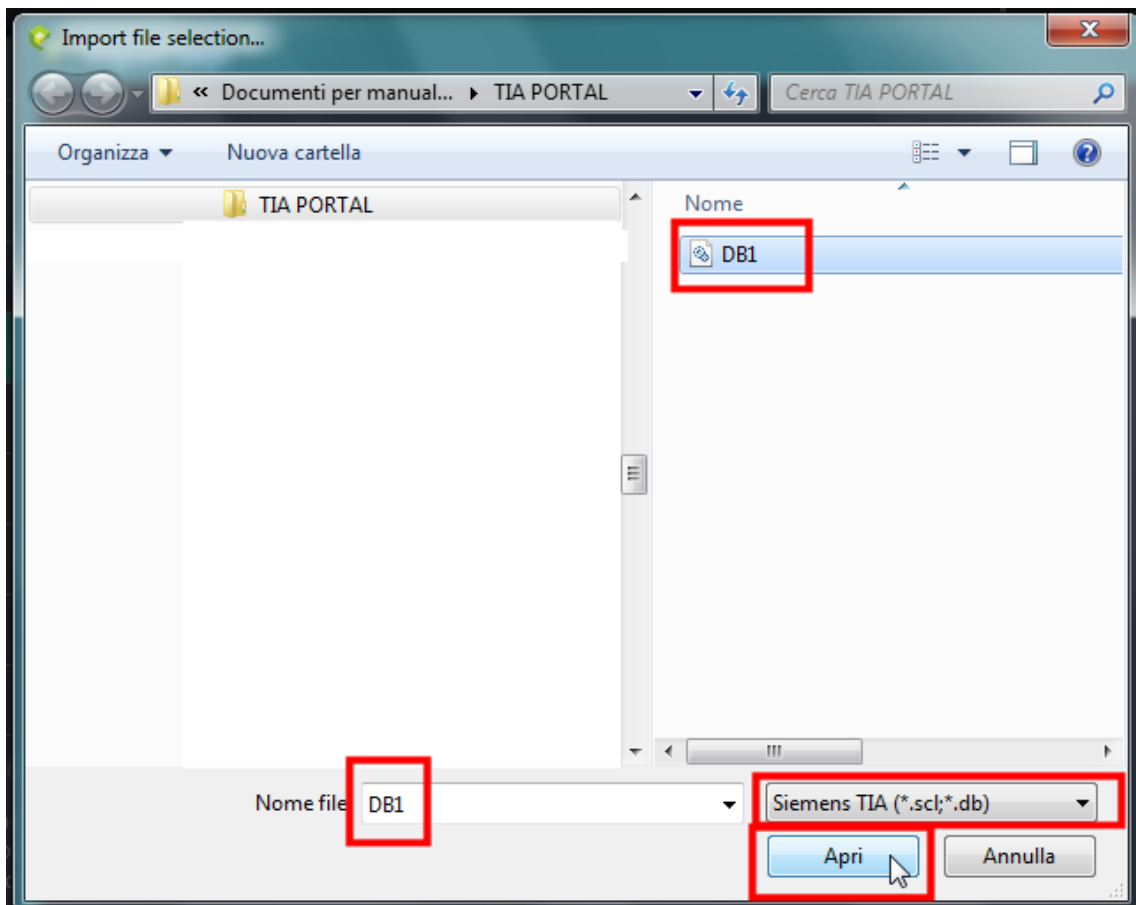
# CREW Manual



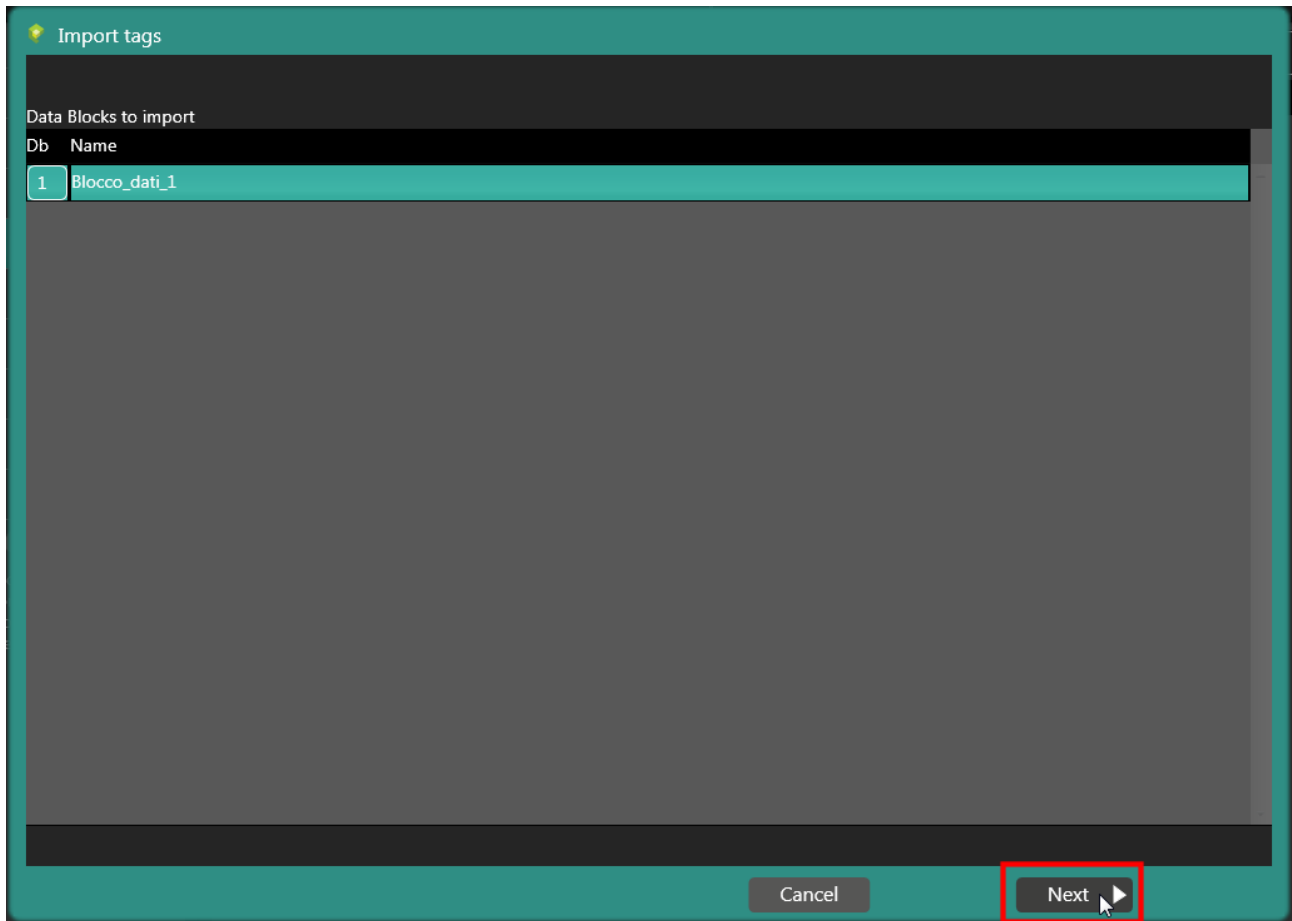
# CREW Manual



# CREW Manual



# CREW Manual



# CREW Manual

Import tags

Tags to import

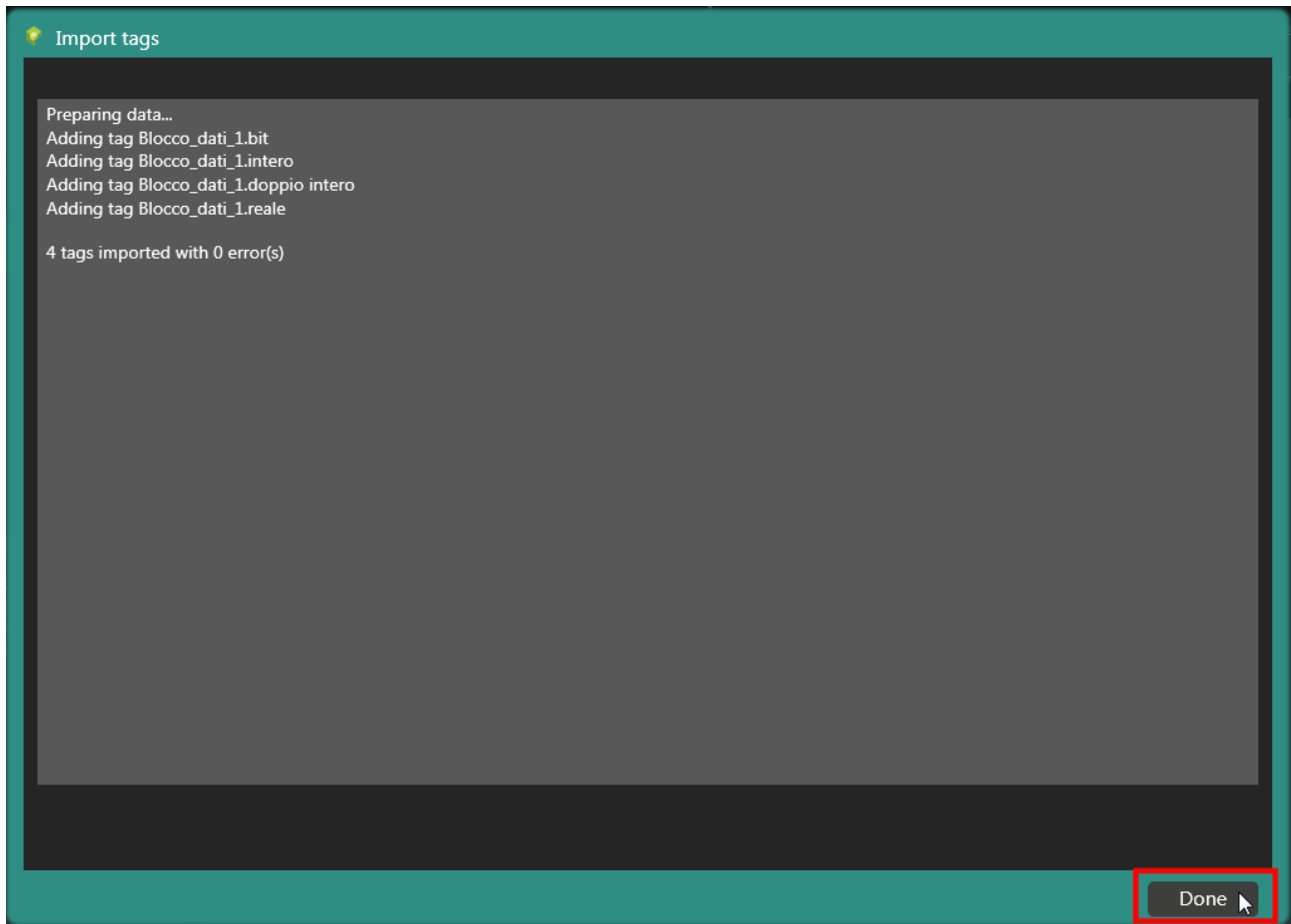
Name	Folder	TagType	AddressType	Description
<input checked="" type="checkbox"/> Blocco_dati_1.bit		Boolean	Device	
<input checked="" type="checkbox"/> Blocco_dati_1.intero		Integer	Device	
<input checked="" type="checkbox"/> Blocco_dati_1.doppio intero		Long	Device	
<input checked="" type="checkbox"/> Blocco_dati_1.reale		Real	Device	

Select all   Unselect all   Destination device: Industrial Ethernet protocol (for S7-1200/1)    Override the existing objects

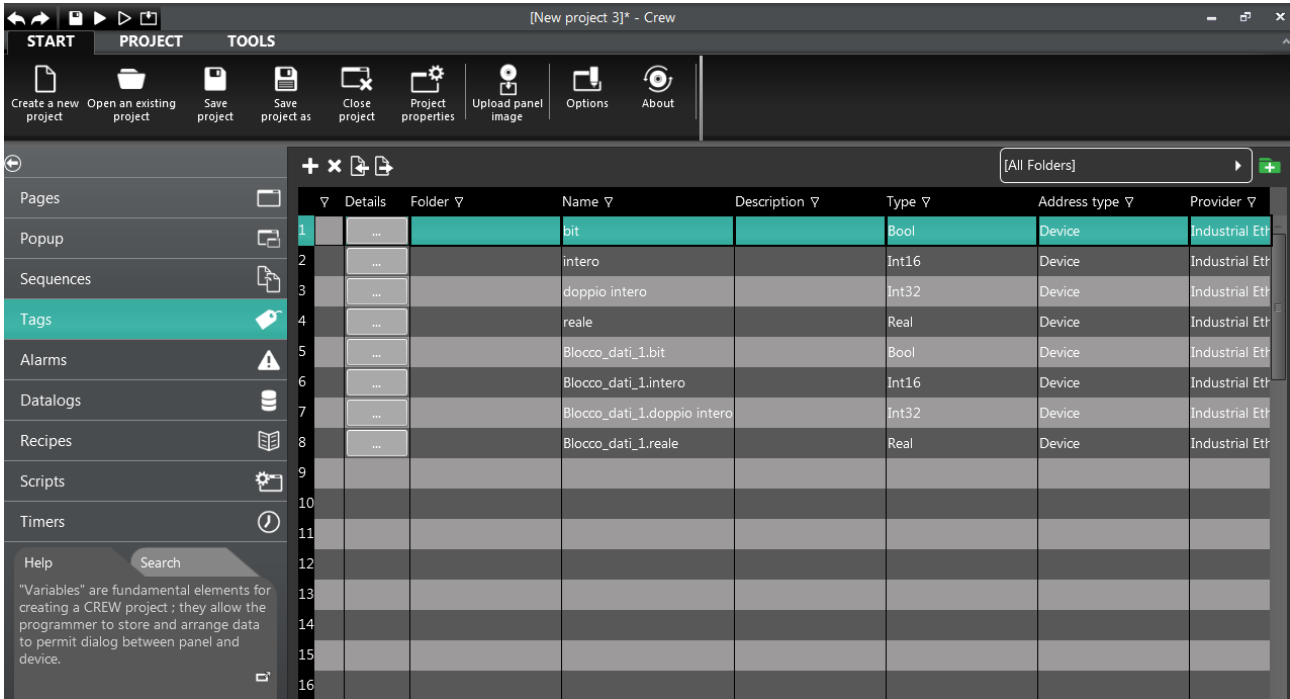
Cancel   Next



# CREW Manual



# CREW Manual



The screenshot shows the CREW software interface with a table of variables. The table has columns for Details, Folder, Name, Description, Type, Address type, and Provider. The 'Tags' menu item is highlighted in the left sidebar.

	Details	Folder	Name	Description	Type	Address type	Provider
1	...		bit		Bool	Device	Industrial Etr
2	...		intero		Int16	Device	Industrial Etr
3	...		doppio intero		Int32	Device	Industrial Etr
4	...		reale		Real	Device	Industrial Etr
5	...		Blocco_dati_1.bit		Bool	Device	Industrial Etr
6	...		Blocco_dati_1.intero		Int16	Device	Industrial Etr
7	...		Blocco_dati_1.doppio intero		Int32	Device	Industrial Etr
8	...		Blocco_dati_1.reale		Real	Device	Industrial Etr
9							
10							
11							
12							
13							
14							
15							
16							

# CREW Manual

---

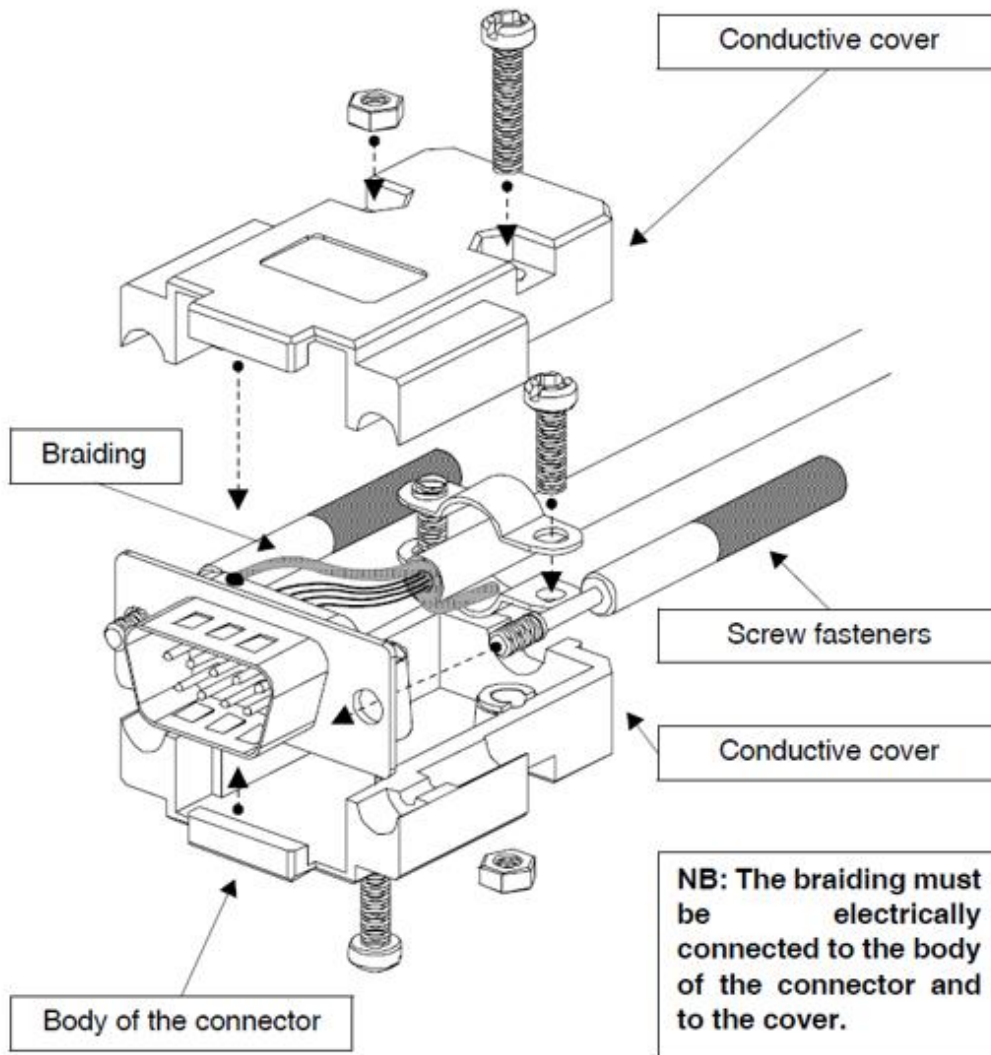
## Connection cables

### Connection of the cable shield

The correct shielding of the interface cables between EW and device is crucial for setting up serial communication without external interference. Therefore all of the cables mentioned in this manual strictly need to be shielded and the D-sub connectors on the EW side and device side need to have a metal or conductive plastic case.

# CREW Manual

The correct shielding connection method is shown in the lay-out below.



The interface cable shield must be electrically connected to the case and to the body of the connector itself from both sides of the cable.

If it is not possible to connect the device side shield due to the type of particular serial connector, the shielding itself must be taken externally to the connector and connected to the earth clamp.

# CREW Manual

The same operation must also be performed if the body of the device serial connector, even if standard, is not electrically connected to the earth clamp of the PLC itself.

It is understood that, also in this condition, the shield must be connected to both case and body of the connector.

Some cable shields have the pin configuration of the device side shielding signals. In these cases, considering the above, the shield must be connected.

In all cases the connection of the EW side shield (pin 1) must never be set up.



Note: Earth potentials obtained from DIN guides, machine metalwork, electrical panel doors, etc. are not allowed, and it is a good idea to avoid equipotential bars where earth from inverter, driver, stepper motor loads, and all loads that generally represent a source of major disturbance, converge.

The failure to comply with these indications can jeopardise the compatibility of the EW-PLC system with EMC regulations in force.

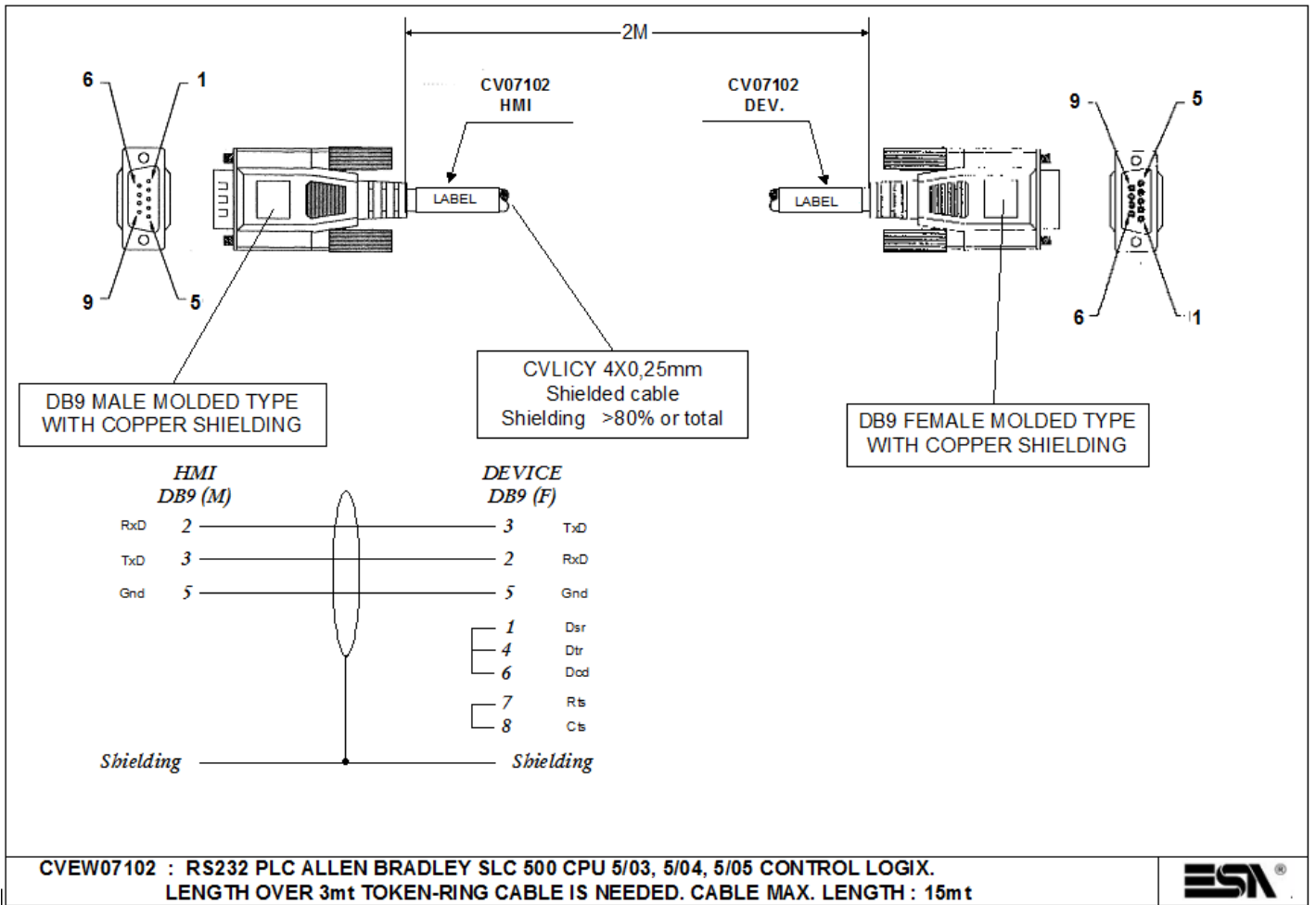
# CREW Manual

## RS232 connection cables

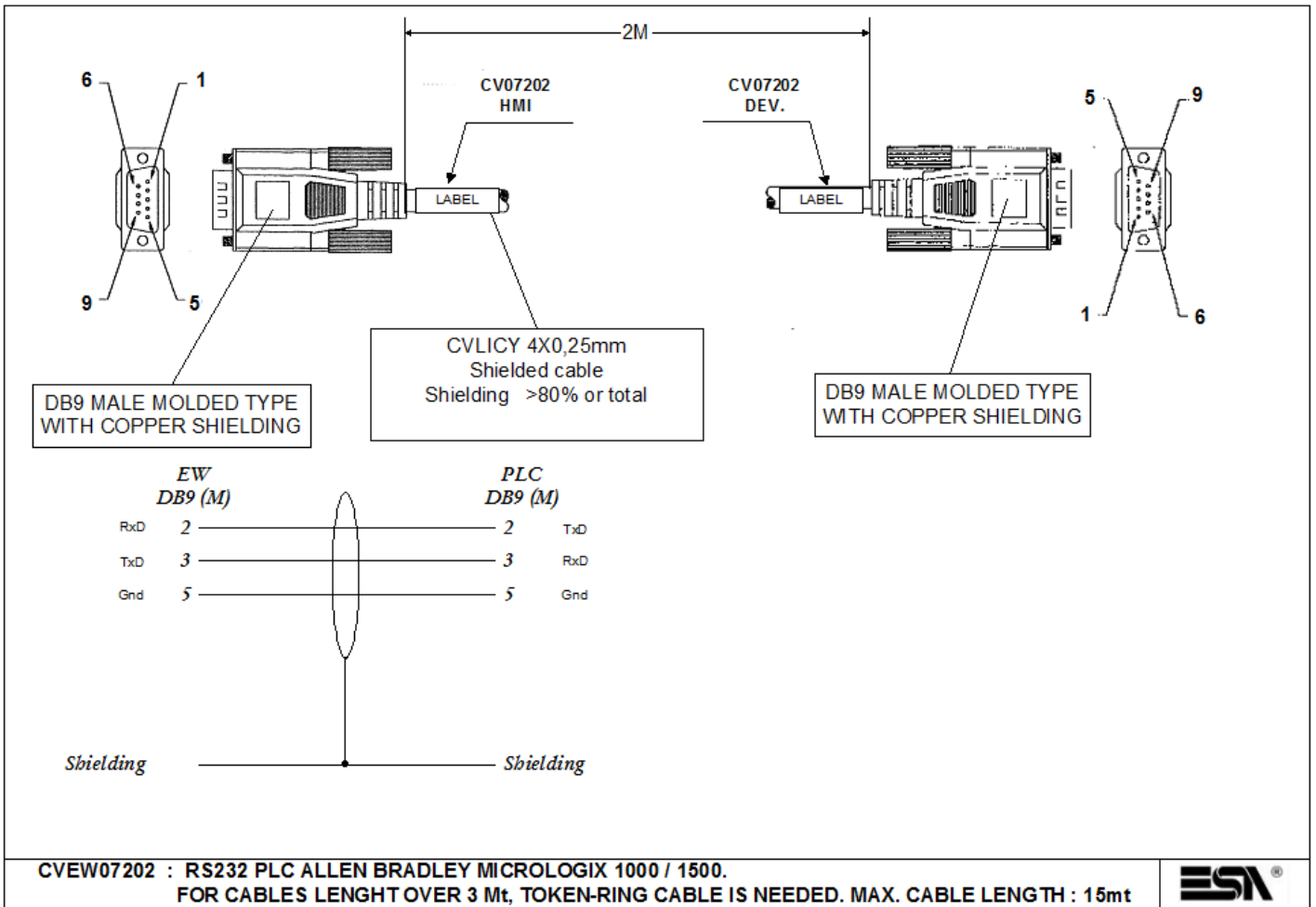
### Cables for PLC Allen Bradley - Rockwell (RS232)

<b>MicroLogix - All devices</b>	<b>ControlLogix 5 series</b>
<p>(RS232)</p> <p>SET MICROLOGIX SERIAL DOOR DF1 mode full duplex (MICRO)</p> <p>Baud rate : 9600 - 38400* ACK timeout : 50 Parity : NONE Error detect : CRC NAK retries : 3 ENQ retries : 3 Embedded responses : ENABLED Duplicate packed detect : NO</p> <p>(*only for CPU1500)</p> <p>The Db 9 male poles connector must be connected to the A-B 1761-CBL-PM02, SER cable, A of the MICROLOGIX PLC.</p> <p>(See "Chapter -&gt; Connection of the cable shield")</p>	<p>(RS232)</p> <p>SET "CH0" System SERIAL PORT DF1 mode full duplex (Point to point)</p> <p>Baud rate : 19200 ACK timeout : 50 Stop Bits : 1 Parity : NONE Control Line : NO HANDSHAKING Error detect : BCC NAK retries : 3 ENQ retries : 3 Embedded responses : ENABLED Duplicate Detect : DISABLED</p> <p><b>N.B. Allen-Bradley V7.00 programming SW onwards is necessary</b></p> <p>(See "Chapter -&gt; Connection of the cable shield")</p>

# CREW Manual

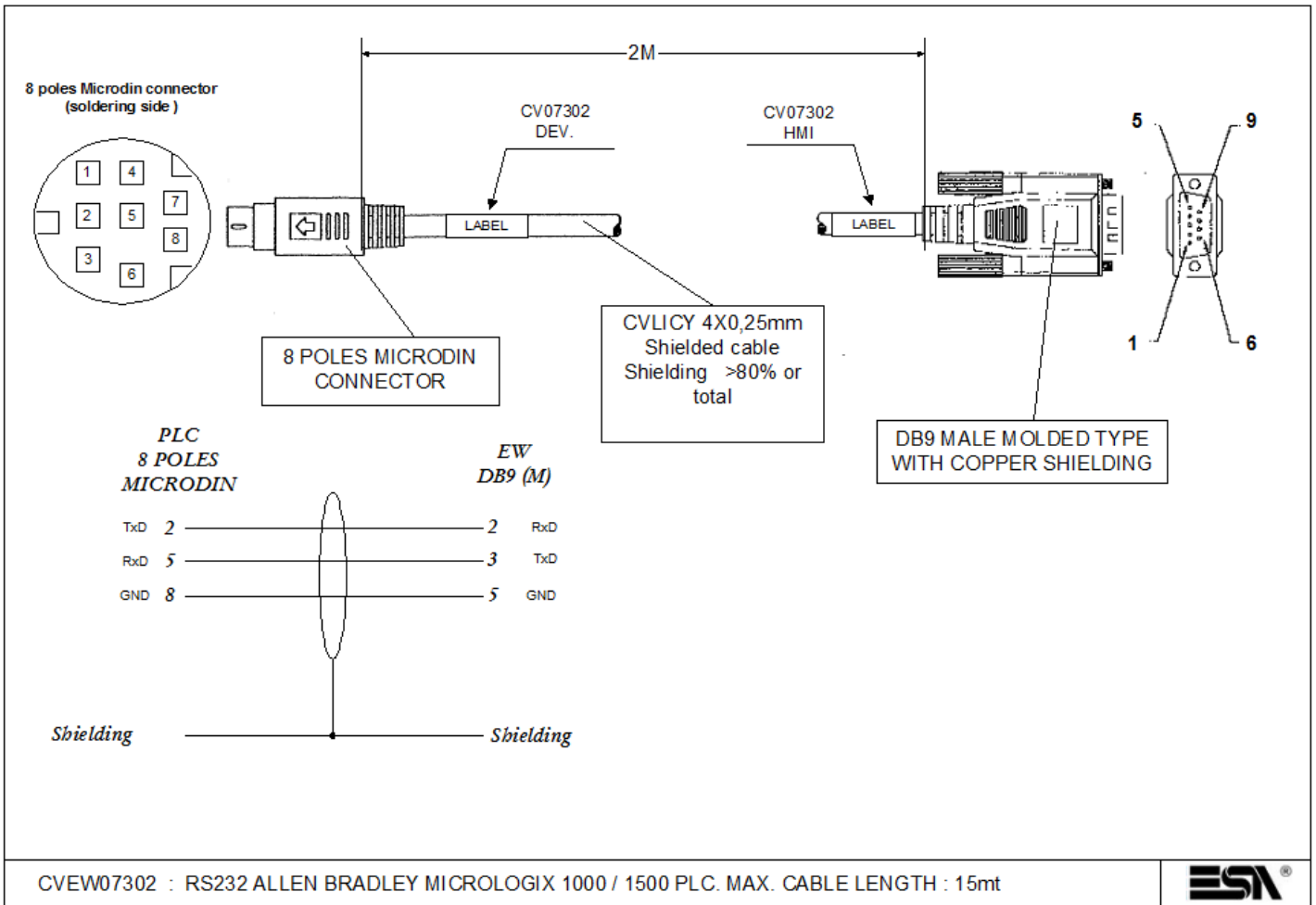


# CREW Manual





# CREW Manual



# CREW Manual

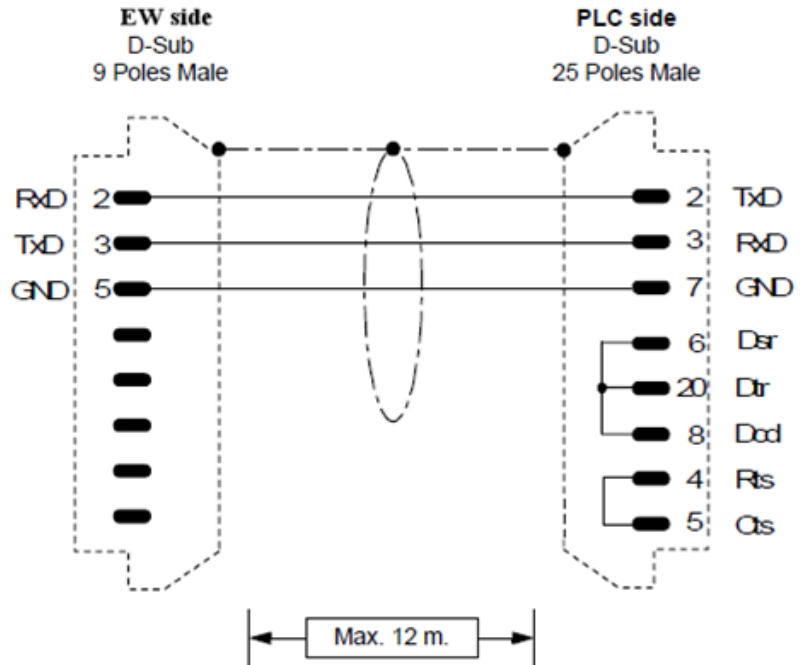
**ControlLogix PLC 5**  
(RS232)

"CH0" SERIAL PORT SETChannel 0: SYSTEM  
DF1 full duplex Mode (Point to point)

Baud rate : 19200  
ACK timeout : 50  
Stop Bits : 1  
Parity : NONE  
Control Line : NO HANDSHAKING  
Error detect : BCC  
NAK retries : 3  
ENQ retries : 3  
Embedded responses : ENABLED  
Duplicate Detect : DISABLED

**N.B. : Allen-Bradley V7.00 or higher programming software needed**

(See "Chapter -> Connection of the cable shield")



# CREW Manual

## Cable for PLC Omron (RS232)

**H series, CS1**  
**CQM1 CPU 21-E and above**  
**CVM1 and C200HS/HE/HG/HX/Hα (integrated serial)**  
**CPM1 through CPM1-CIF01 interface**  
**CPM2A/2C**

(RS232)

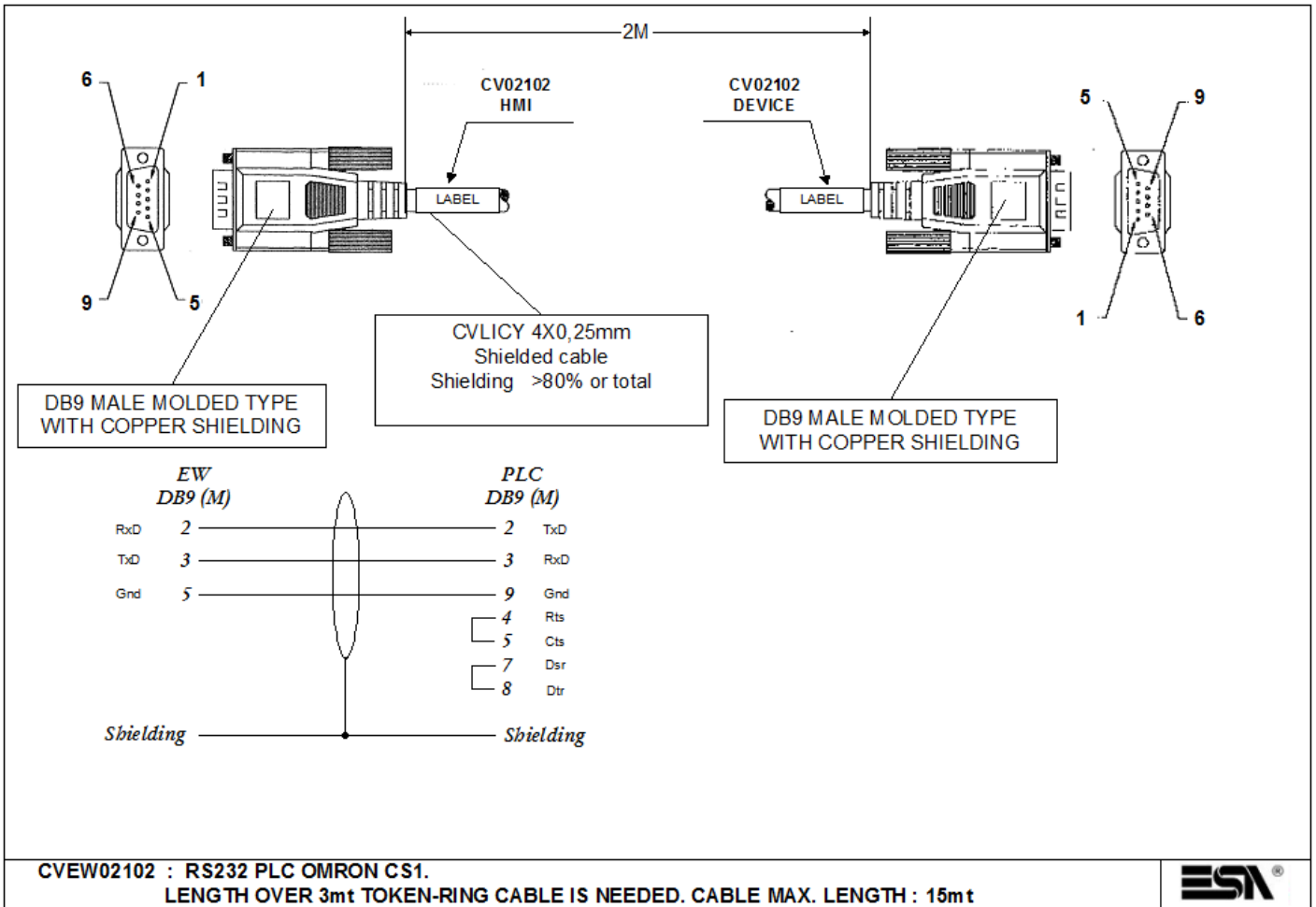
Bridge the **GR** and **LG** clamps on the PLC terminal board.

(See "Chapter -> Connection of the cable shield")

### SETTING JUMPERS CPM1-CIF01

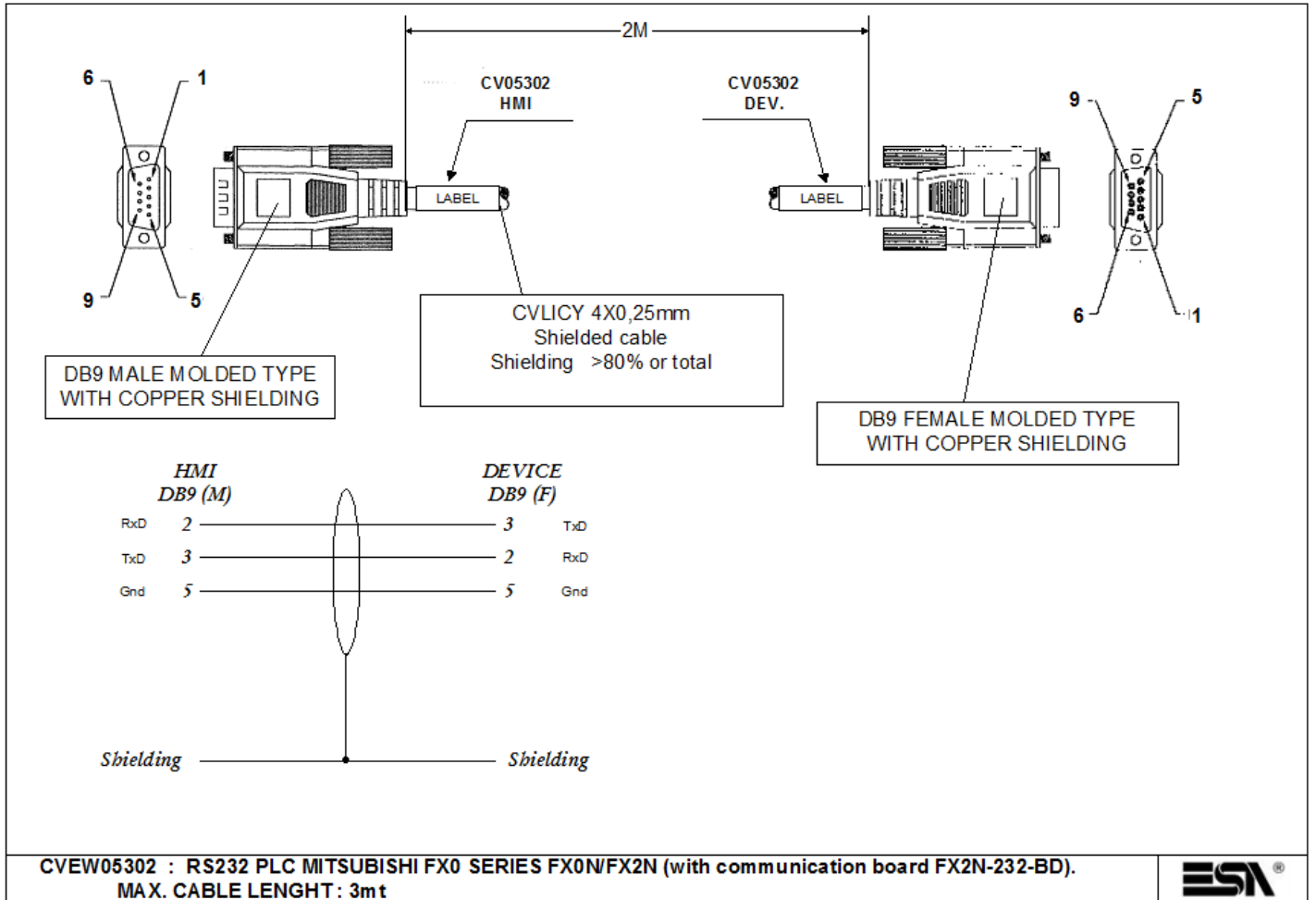
HOST   
NT

# CREW Manual

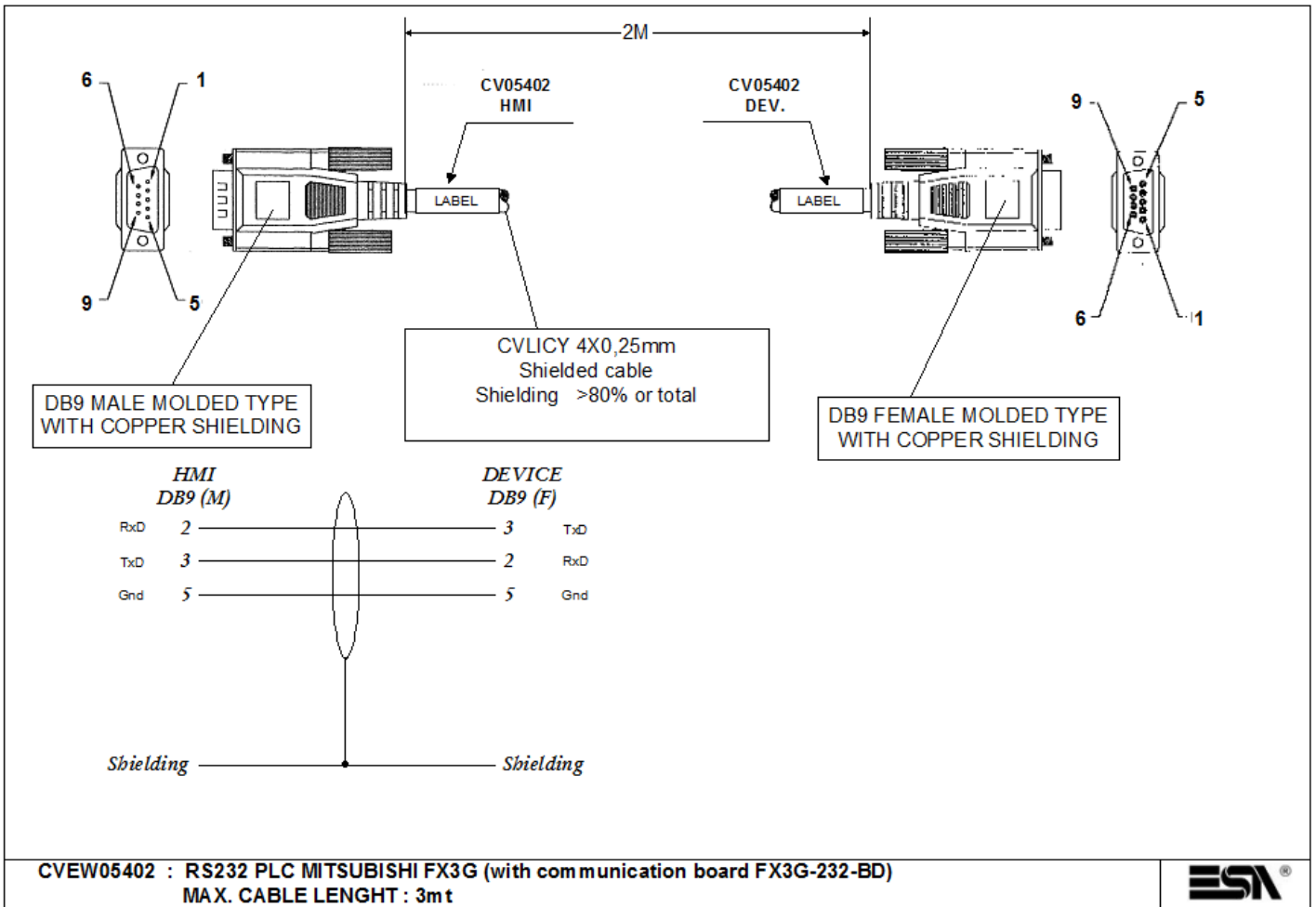


# CREW Manual

## Cables for PLC Mitsubishi (RS232)



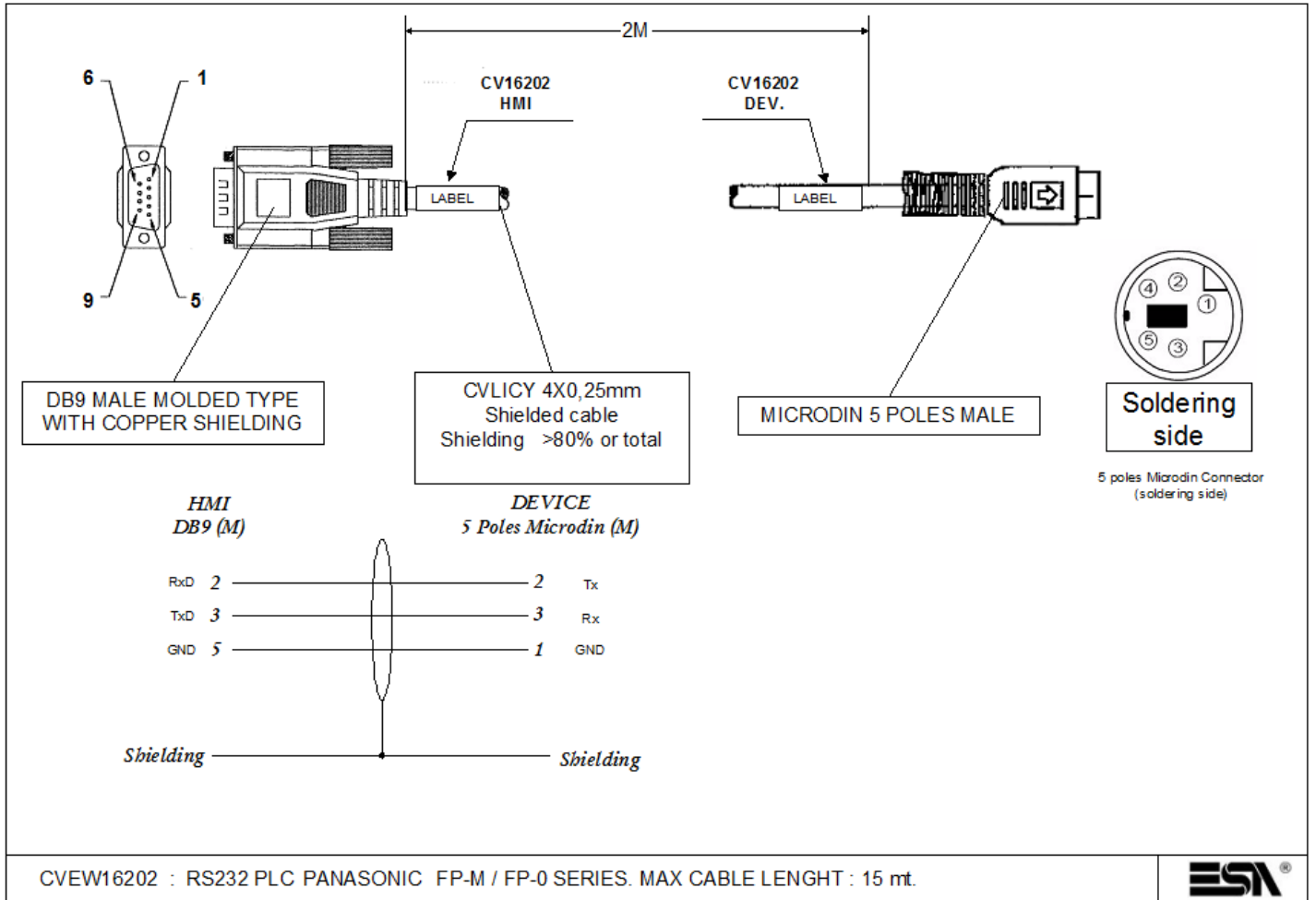
# CREW Manual



CVEW05402 : RS232 PLC MITSUBISHI FX3G (with communication board FX3G-232-BD)  
MAX. CABLE LENGHT : 3m

# CREW Manual

## Cable for PLC Panasonic (RS232)



CVEW16202 : RS232 PLC PANASONIC FP-M / FP-0 SERIES. MAX CABLE LENGTH : 15 mt.

# CREW Manual

## RS485 connection cables

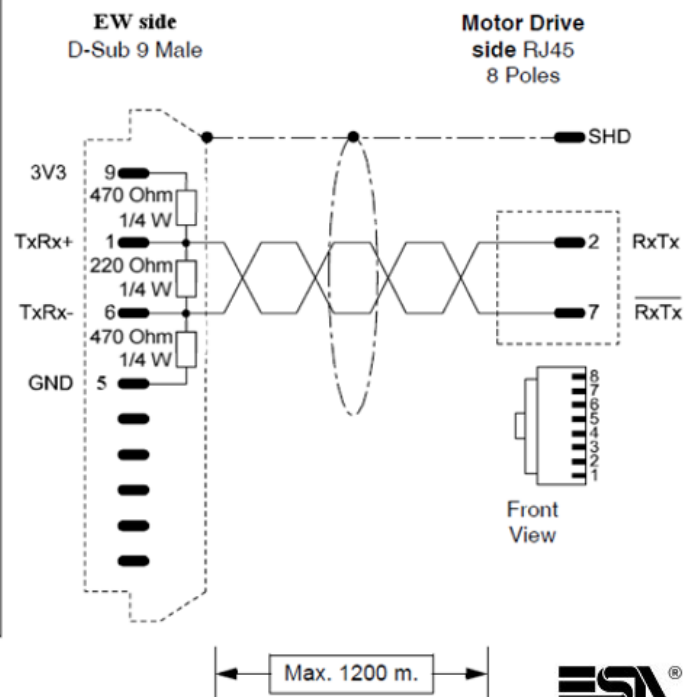
### Cable for Emerson - Control Techniques

#### SP, SK, Uni M200 - 700 (RS485)

(RS485)

Set the parameter 41 = ANSI  
Set the parameter 42 = (Communication speed)  
Set the parameter 43 = (Address)

**NOTE:**  
Register addressed with Menu 0 Parameter 0 cannot be accessed by the panel both in read and write mode.





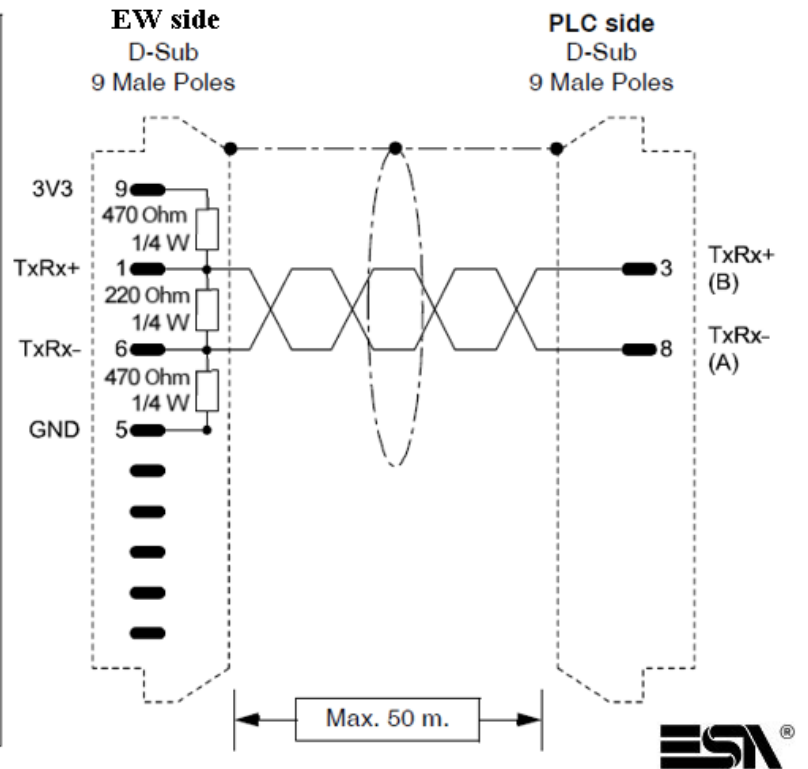
# CREW Manual

## Cable for PLC SIEMENS S7 (RS485)

**SIMATIC S7**  
(RS485)

For contemporary communications between programming suitcase PLC S7 and IT, it is recommended to use the SIEMENS 6ES7972-OBB20-OXAO passing connector.

**N.B.: Max. length 50m without repeater**  
(See "Chapter -> Connection of the cable shield")



# CREW Manual

---

## Backup and Restore - Example

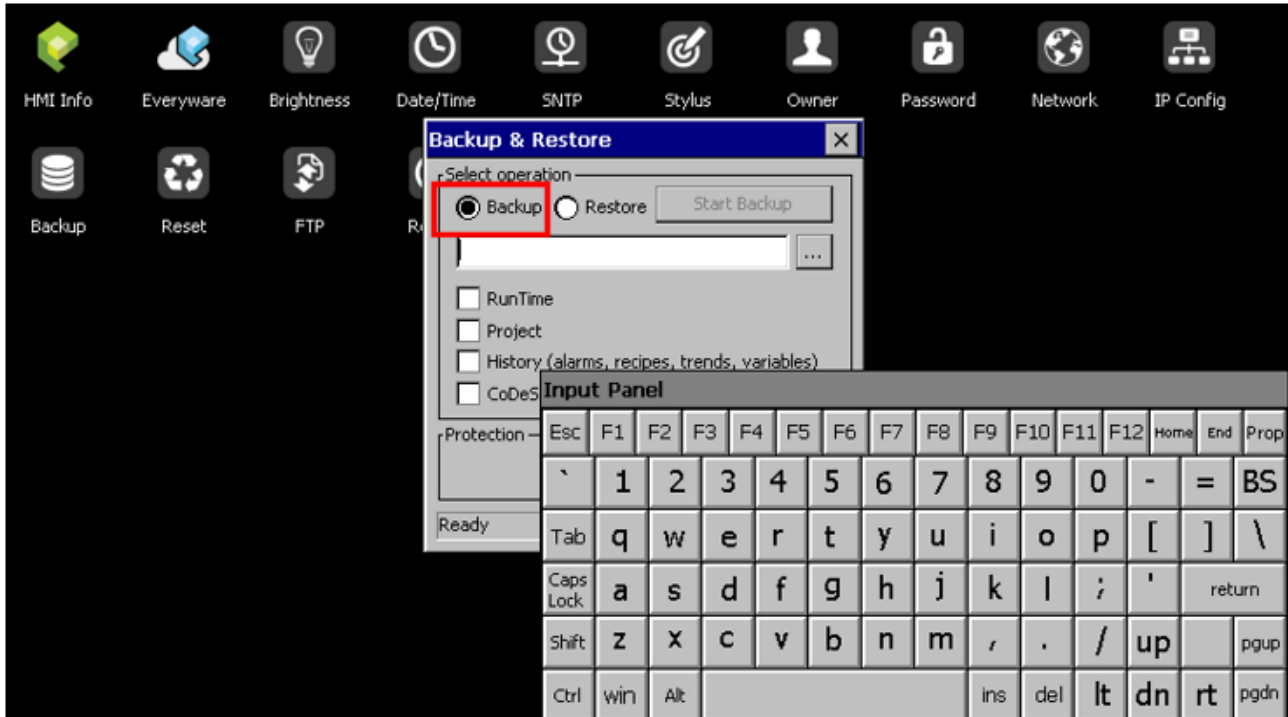
### BACKUP:

From the control panel of the EW terminal, select the “Backup” icon.



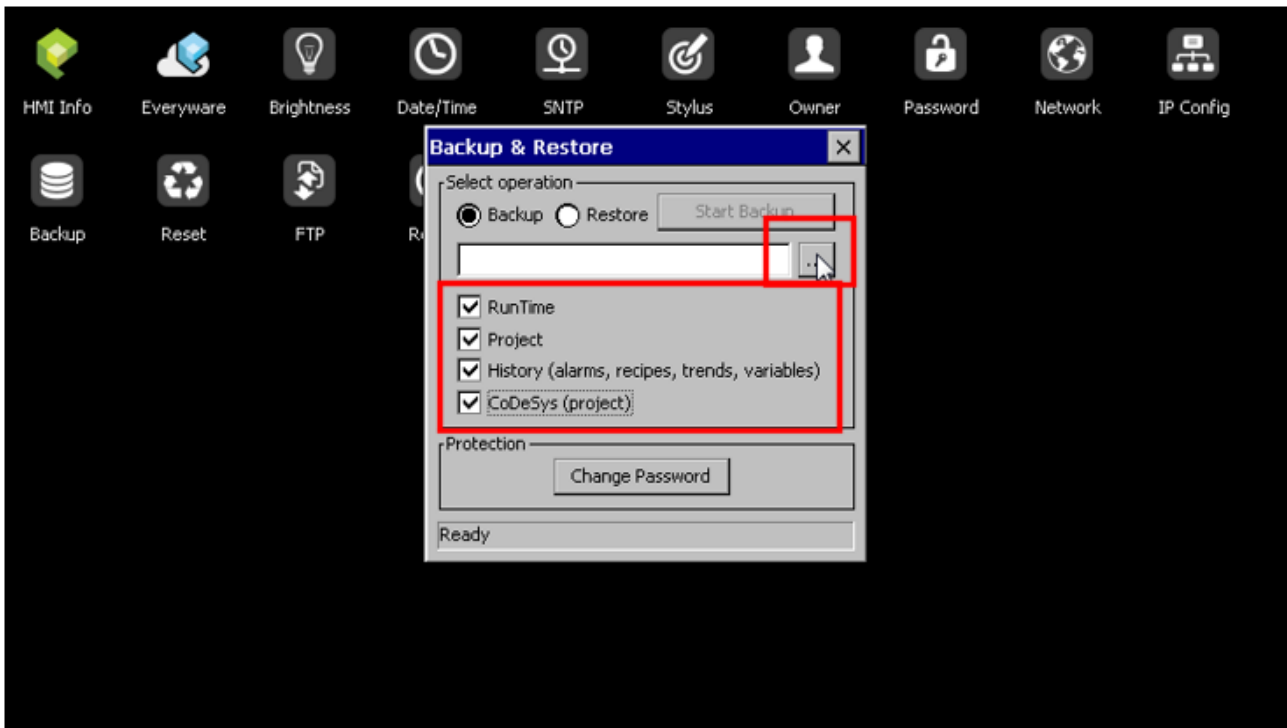
# CREW Manual

Select the “Backup” option.



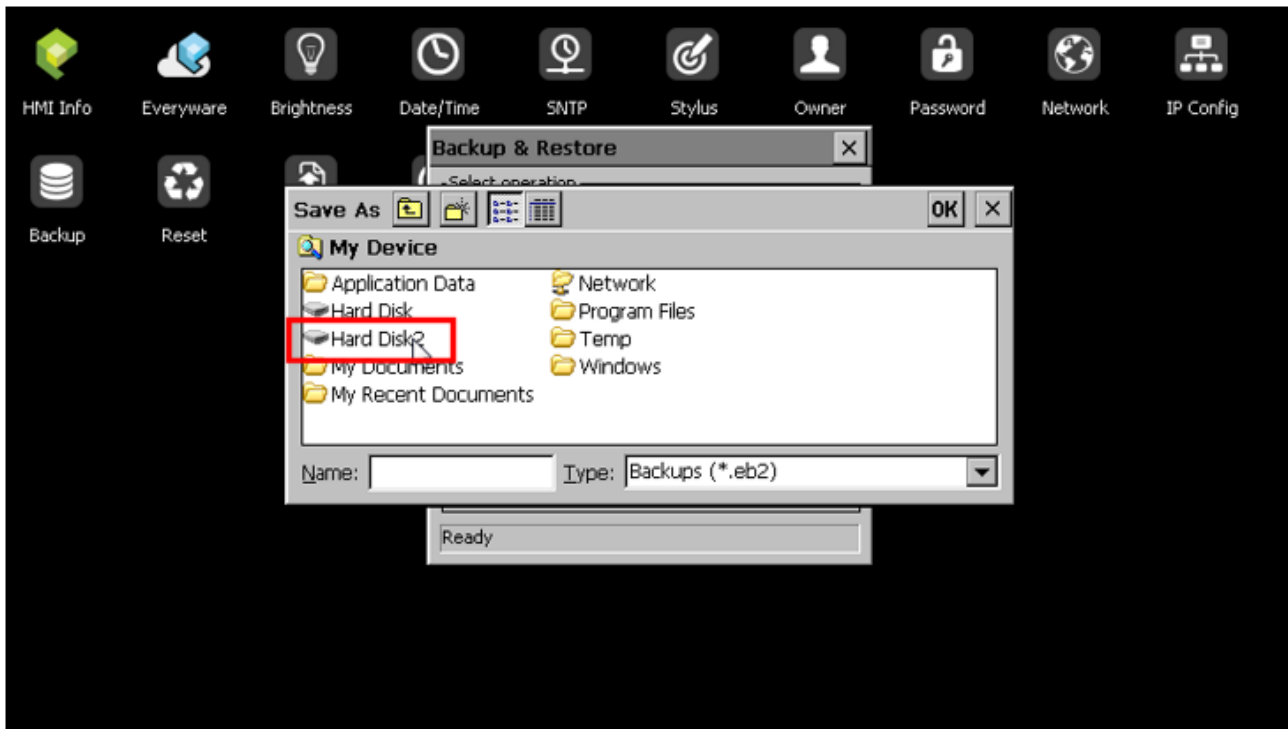
# CREW Manual

Select the "check boxes" relative to the parts of the project that you wish to save (Runtime, Project, Logs or CODESYS project), and click the "Browse" button.



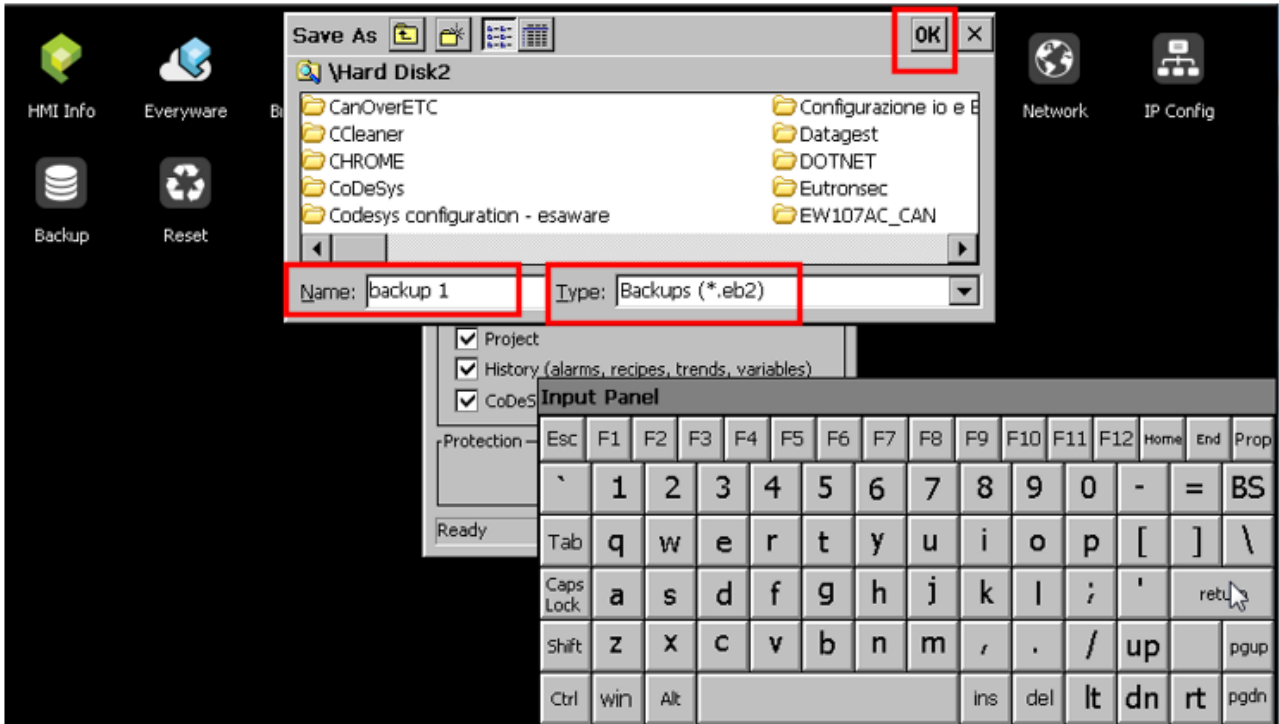
# CREW Manual

Select the path you wish to save the Backup file in.



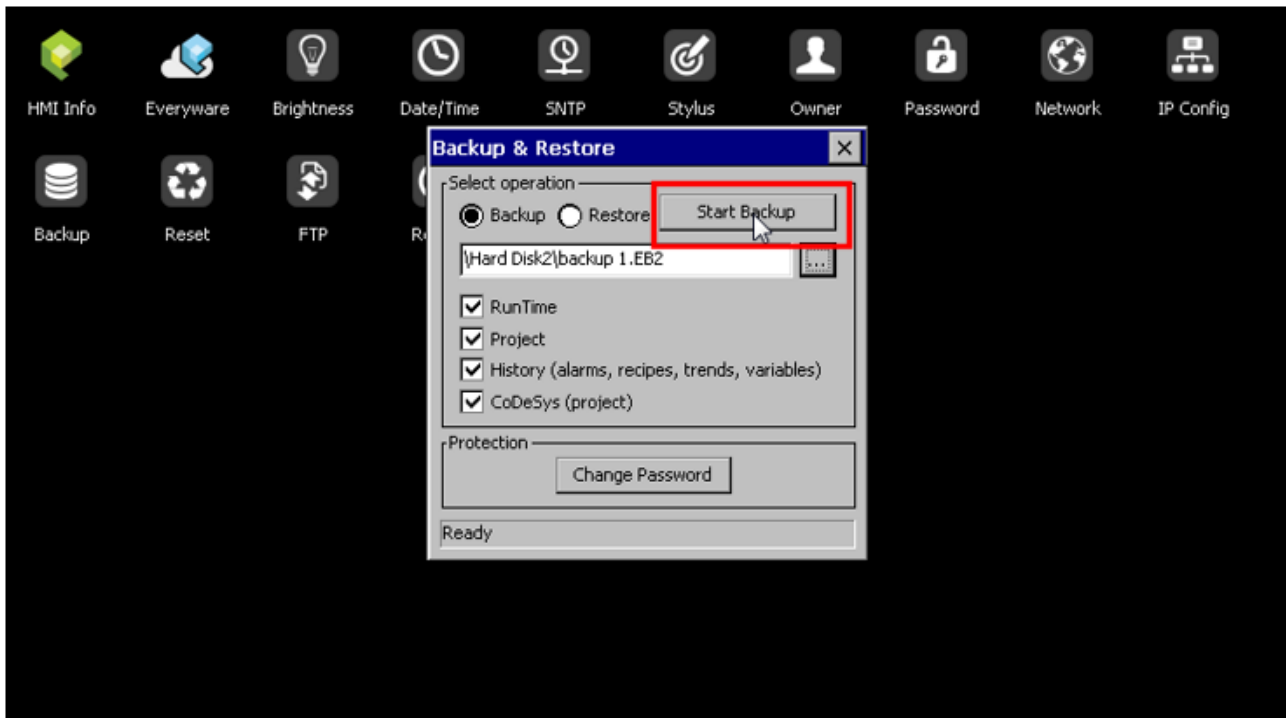
# CREW Manual

Assign a name to the Backup file and click “Ok”.



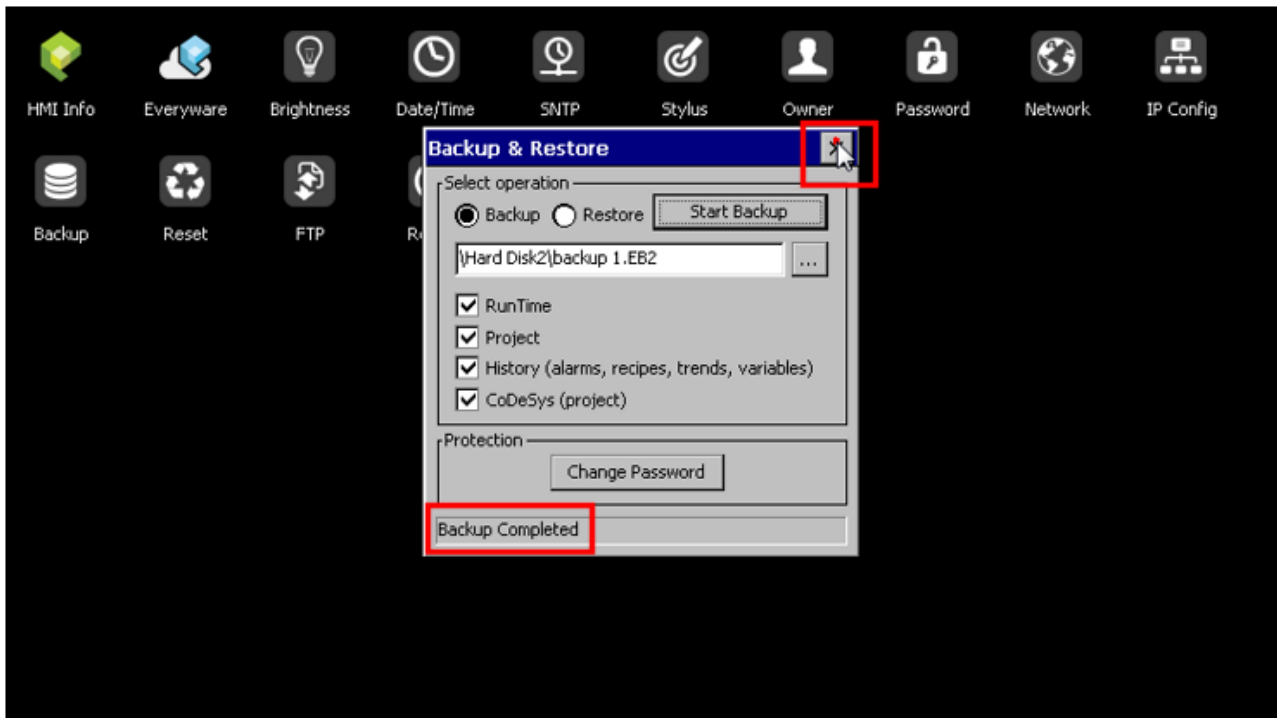
# CREW Manual

Click the “Start Backup” button.



# CREW Manual

When the Backup is finished, close the "Backup & Restore" box.





# CREW Manual

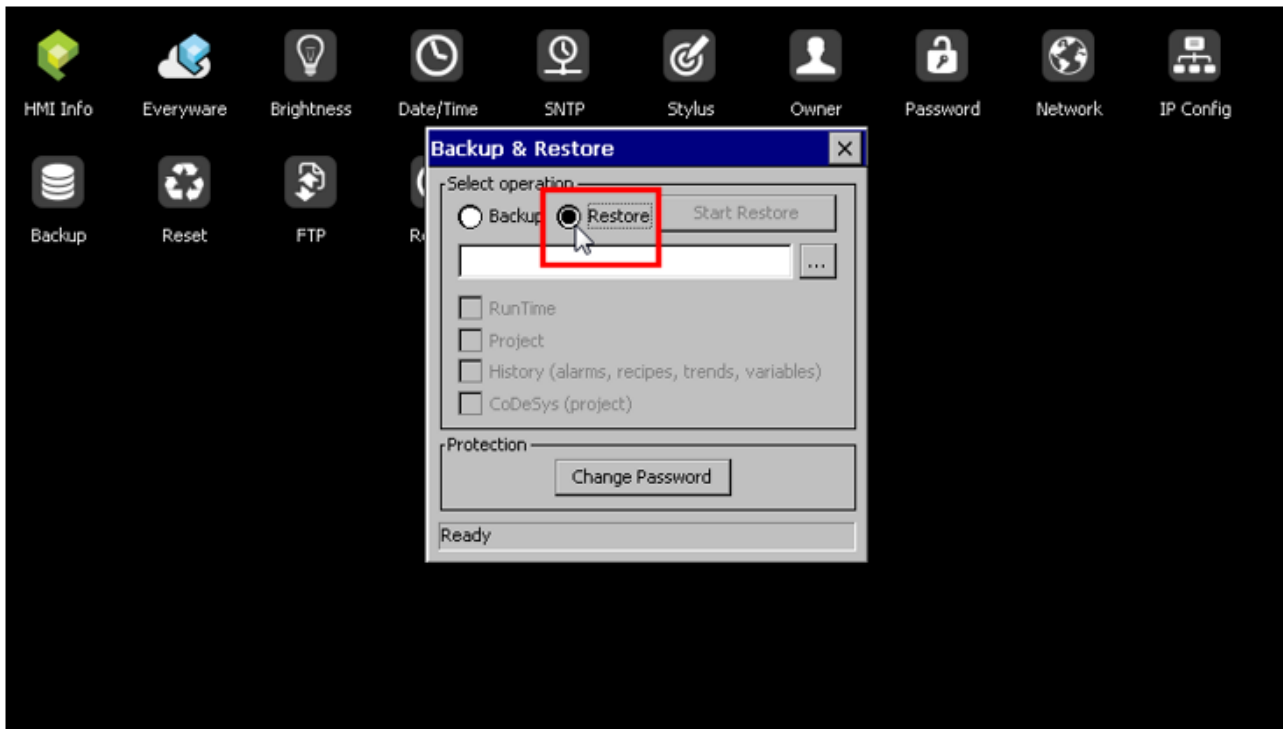
## RESTORE:

From the control panel of the EW terminal, select the “Backup” icon.



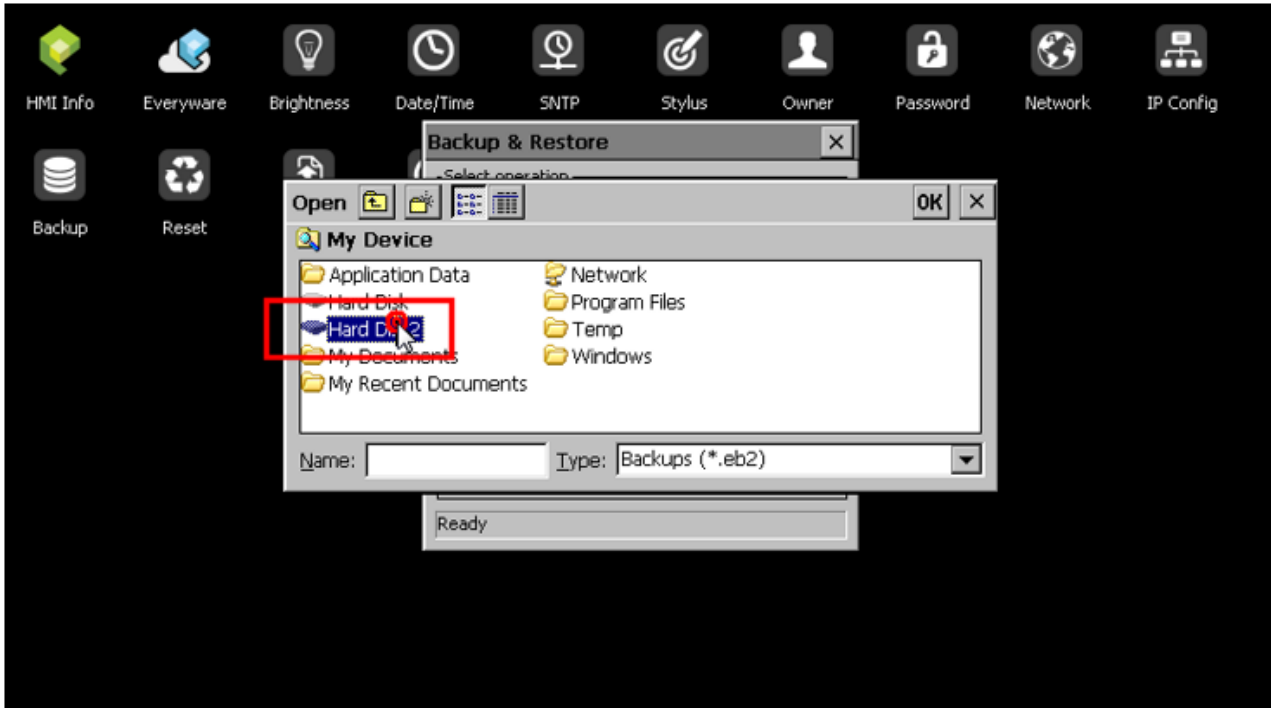
# CREW Manual

Select the “Restore” option.



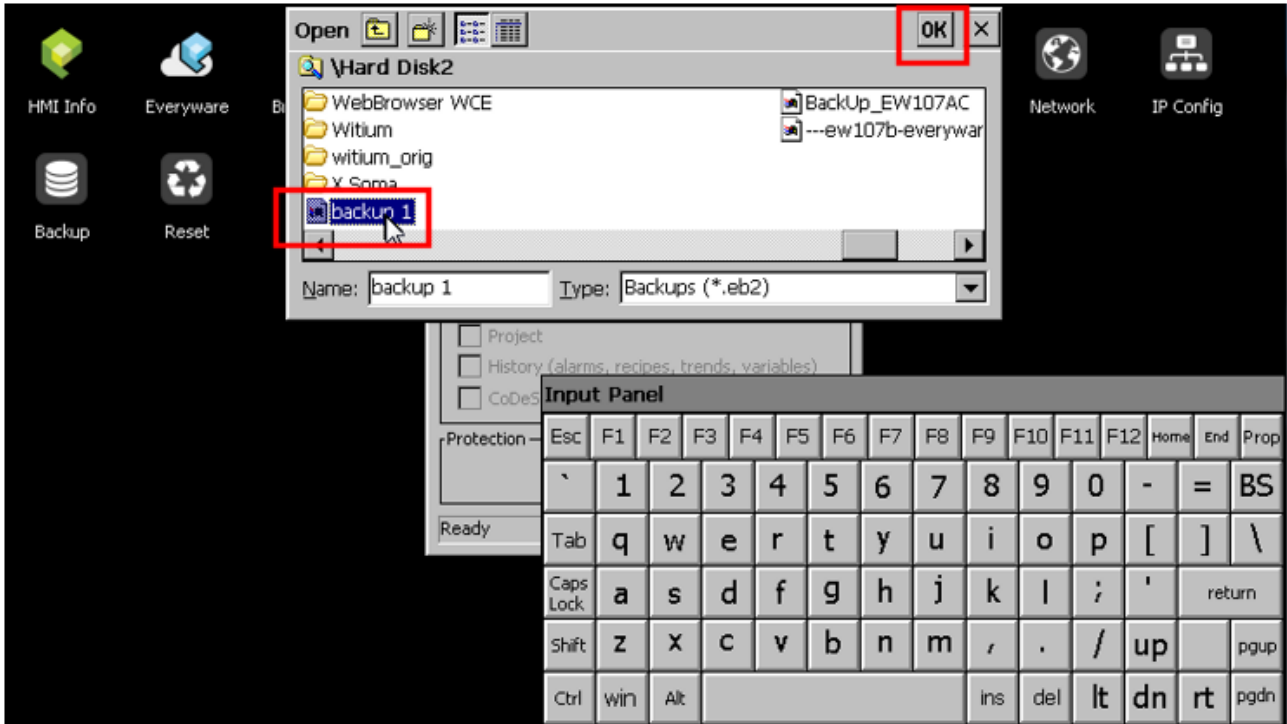
# CREW Manual

Select the path of the restore file to be used.



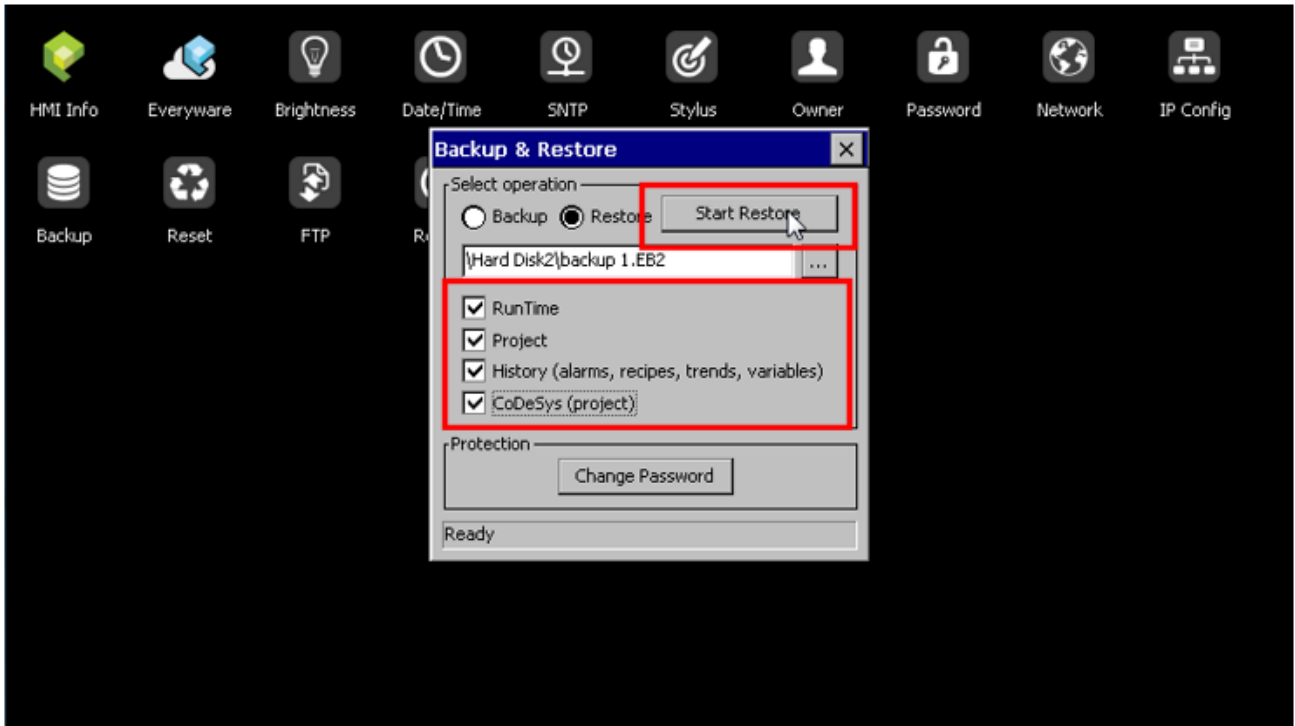
# CREW Manual

Select the file and click “Ok”.



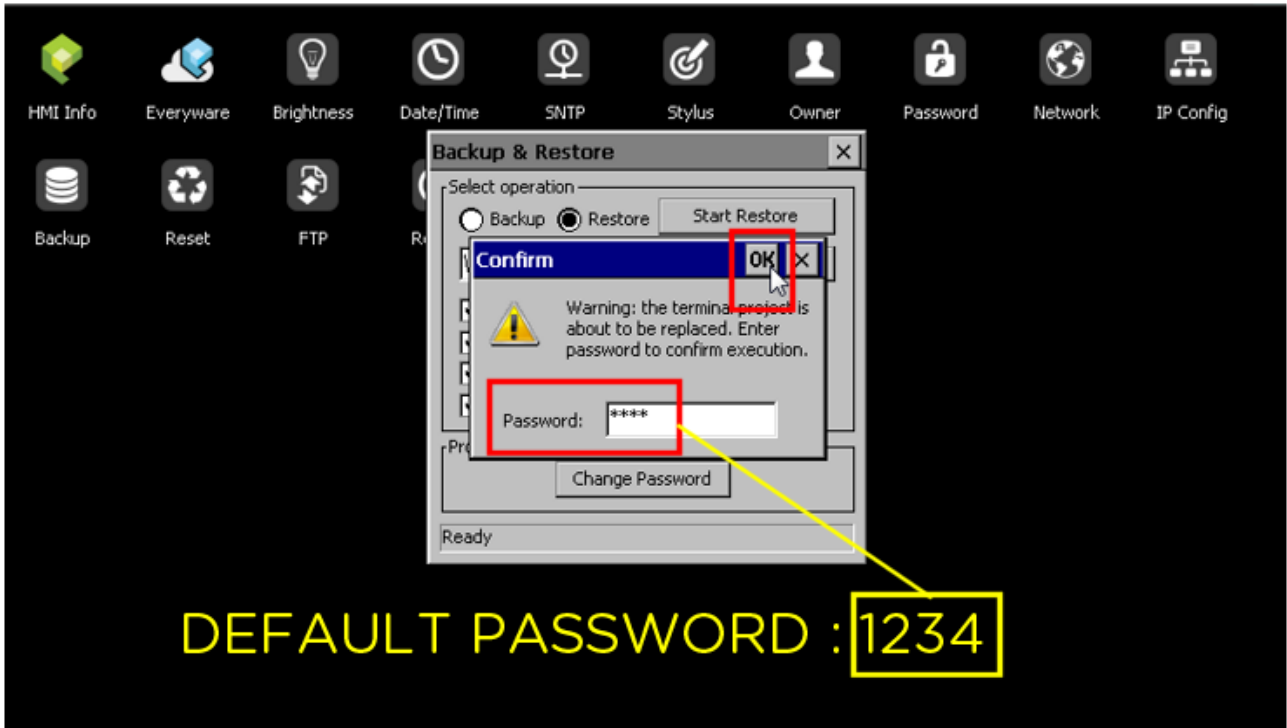
# CREW Manual

Select the "check boxes" relative to the parts of the project that you wish to restore (Runtime, Project, Logs or CODESYS project), and click the "Start Restore" button.



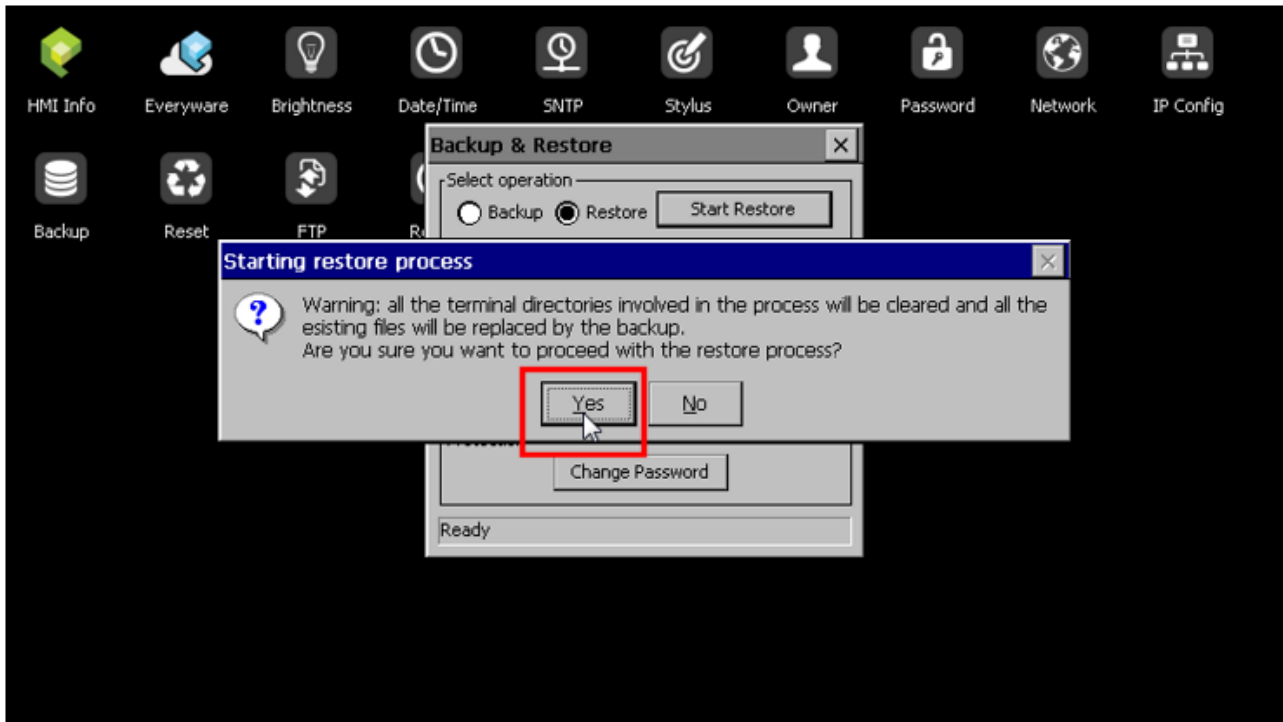
# CREW Manual

Enter the password (the default password is: "1234") and click "Ok".



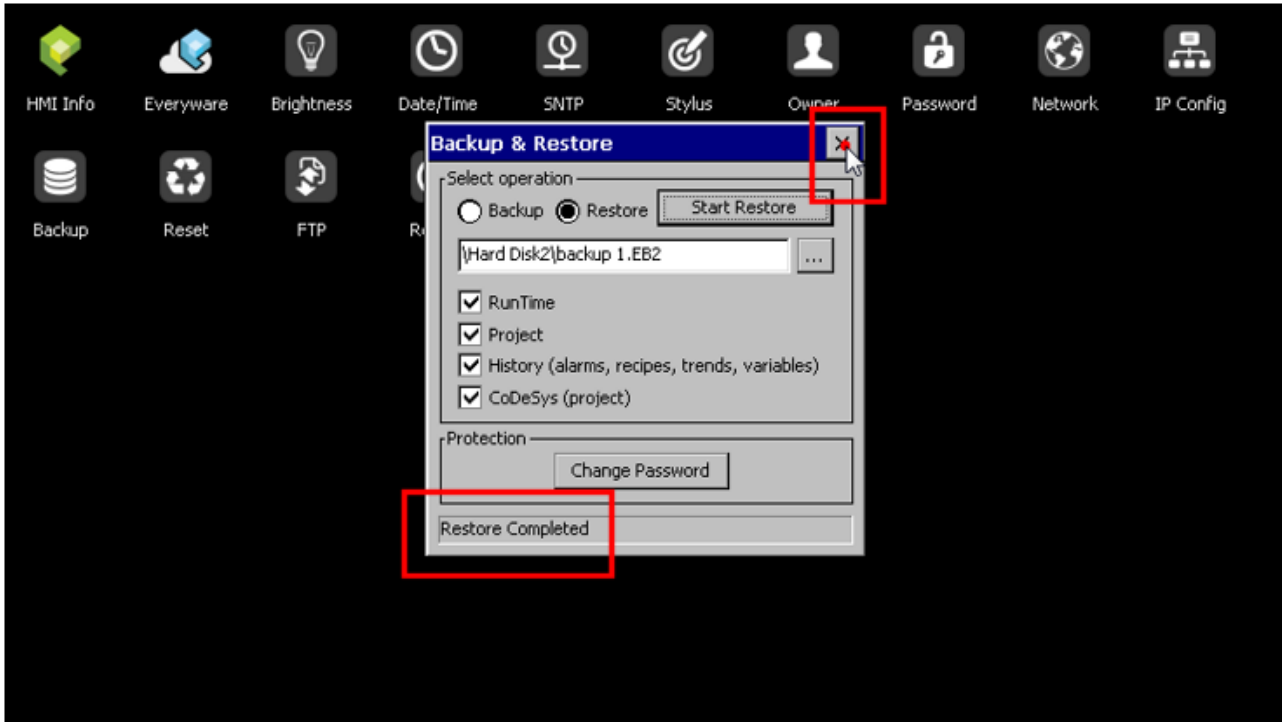
# CREW Manual

All of the files contained in the EW panel will be replaced, click “YES” to continue.



# CREW Manual

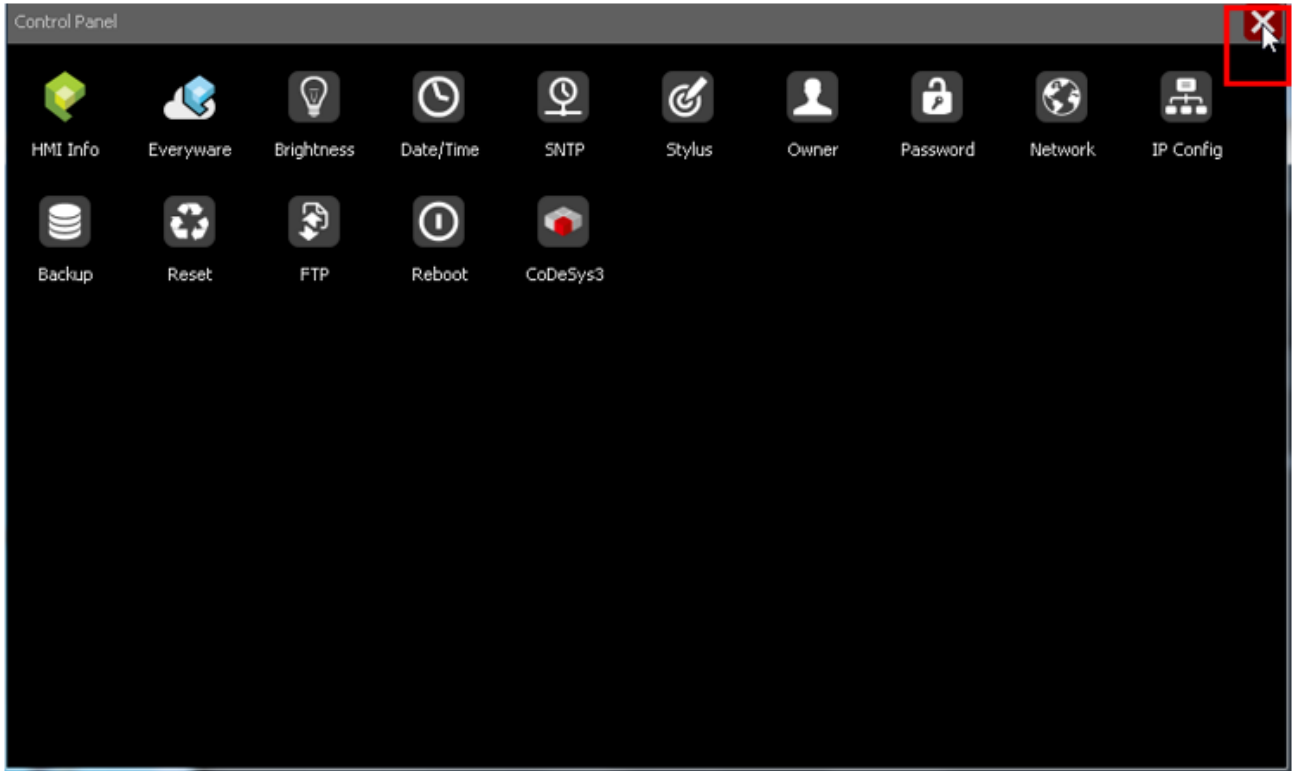
When restore is finished, close the box.





# CREW Manual

Close the EW terminal's control panel.

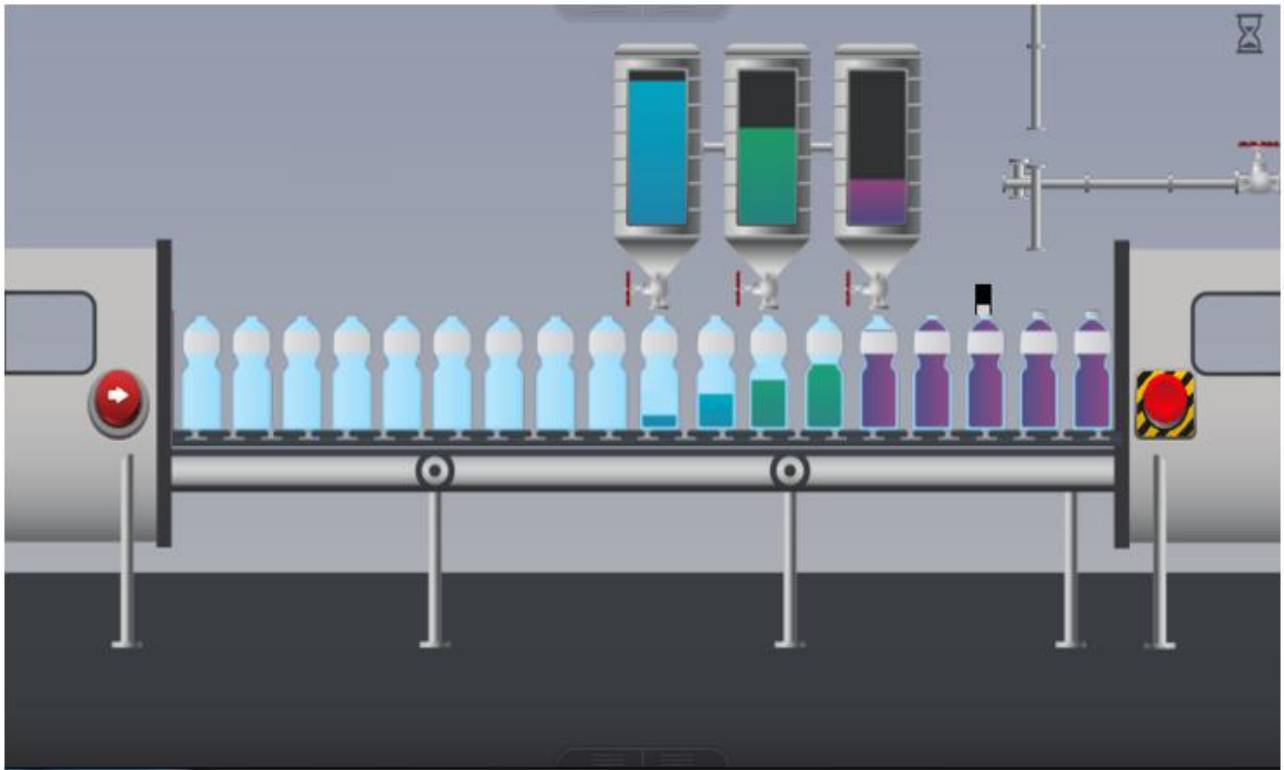


# CREW Manual

Start the project back up again.



# CREW Manual

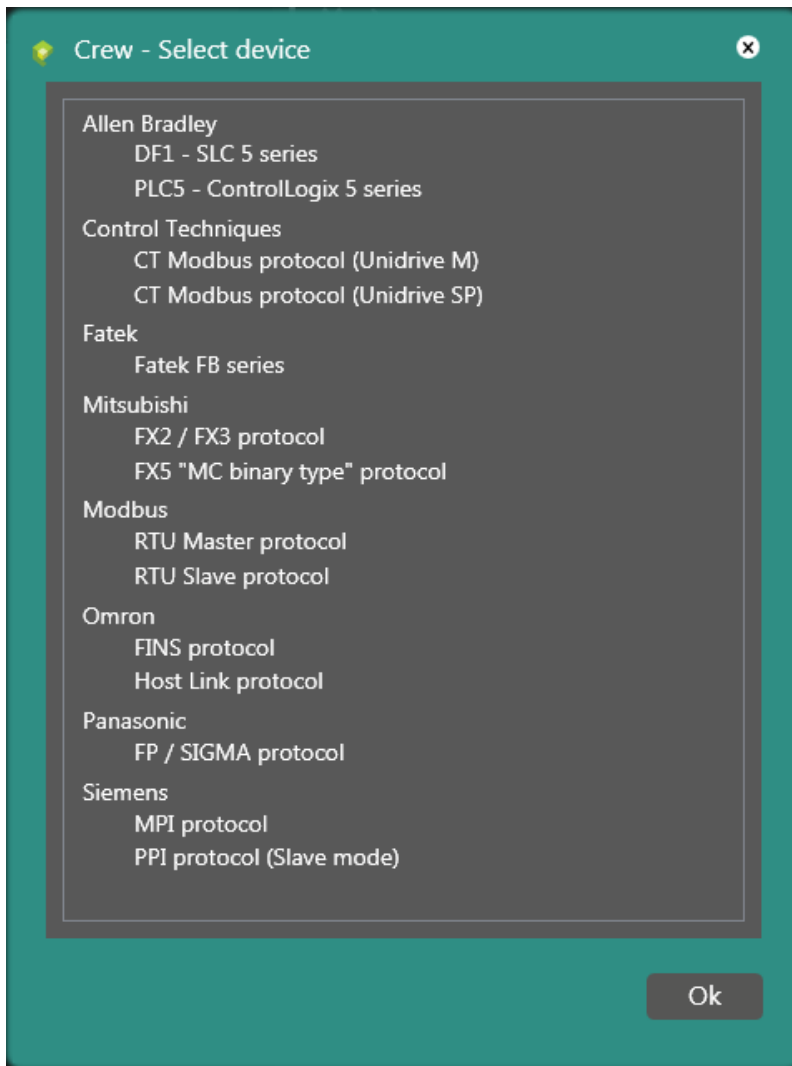


CREW Manual

# Drivers

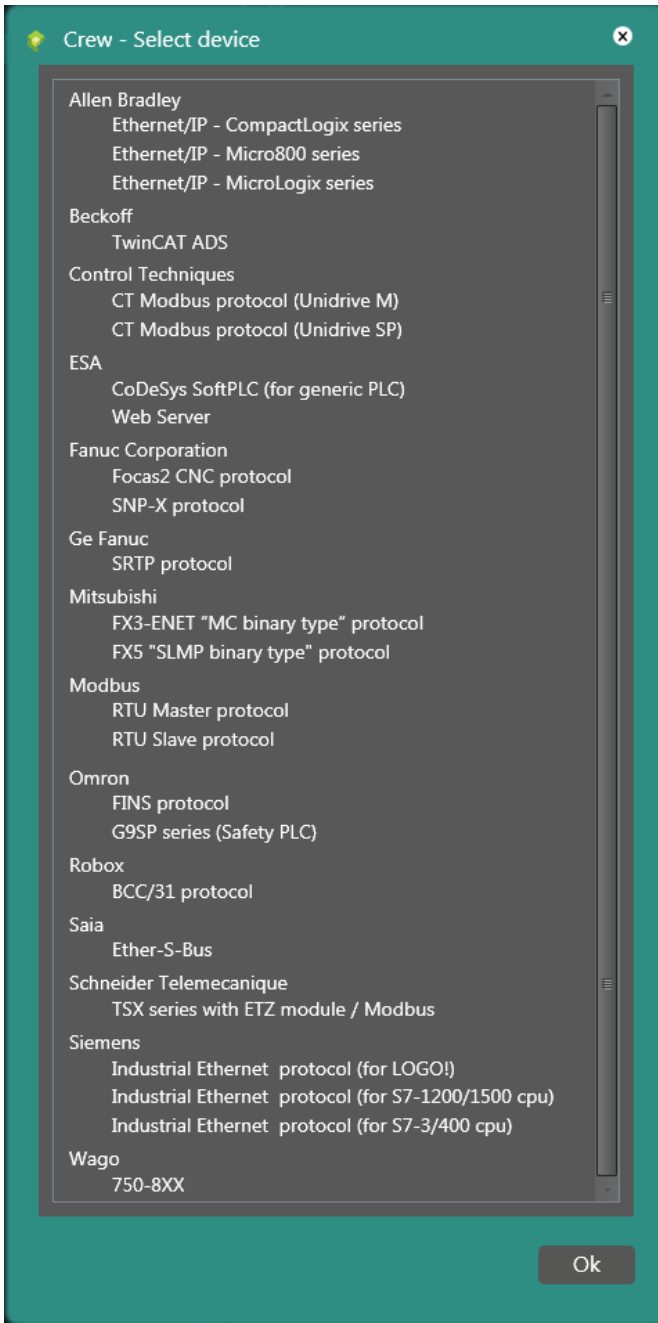
# CREW Manual

List of serial Drivers :



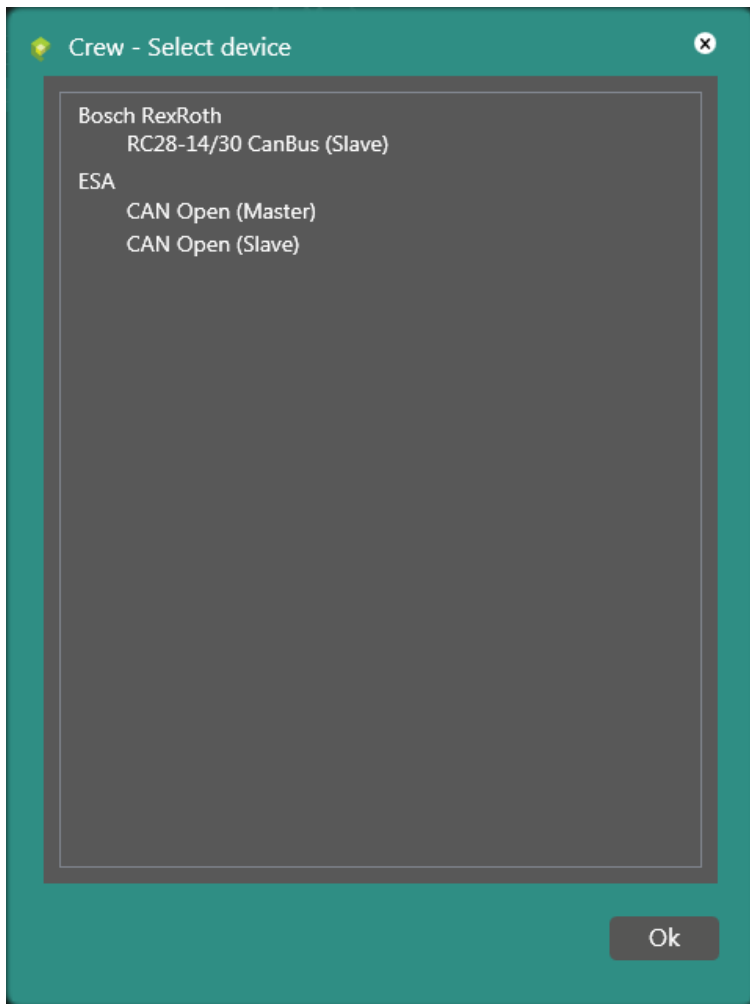
# CREW Manual

## List of Ethernet Drivers :



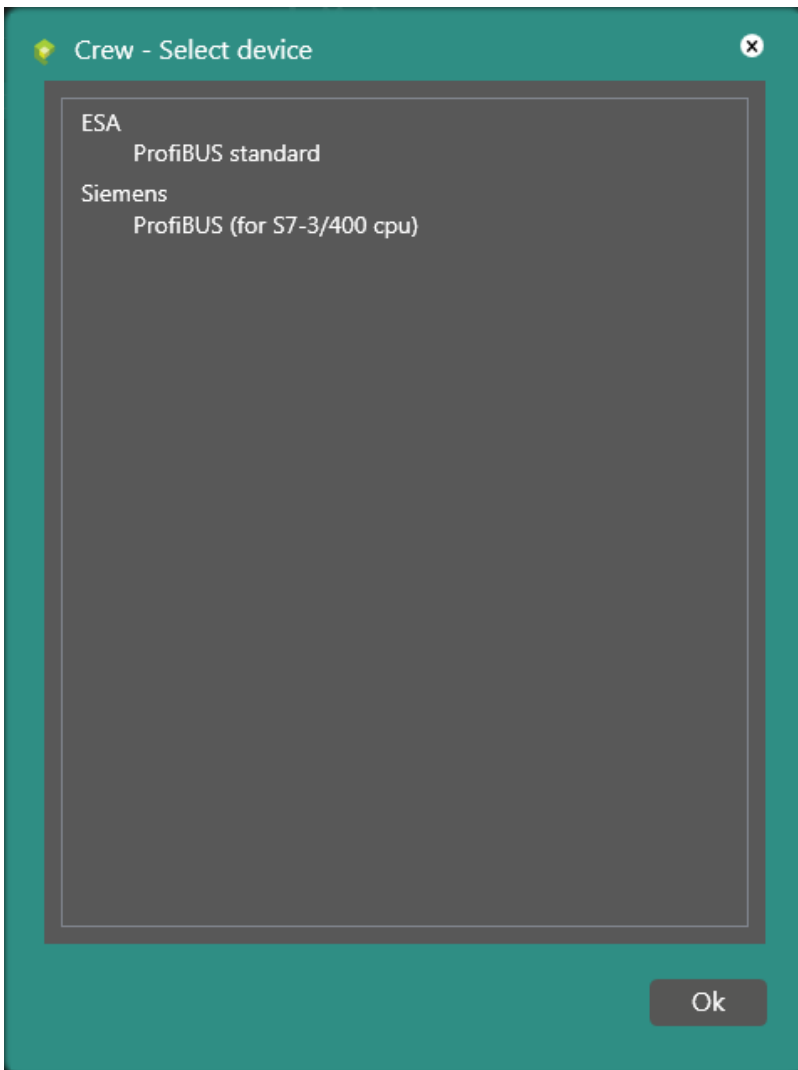
# CREW Manual

List of CAN Open Drivers :



# CREW Manual

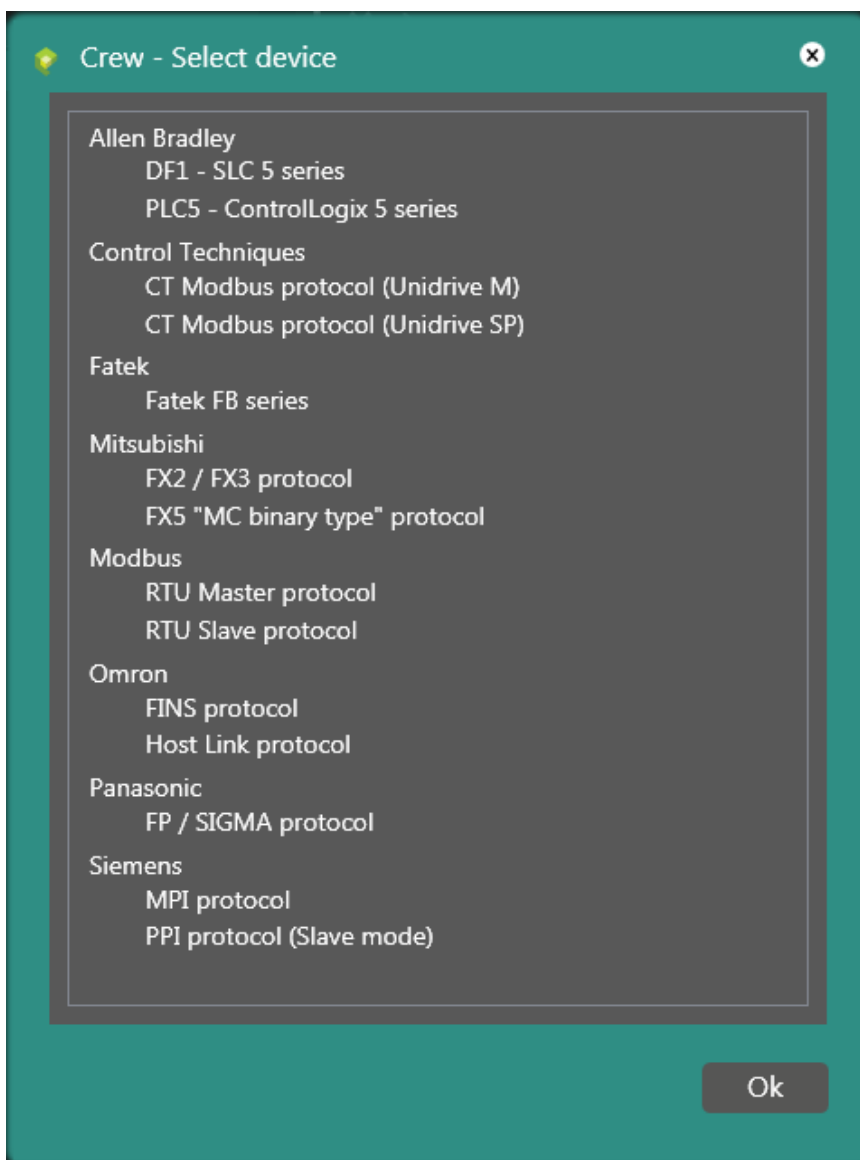
List of ProfiBUS Drivers :





## CREW Manual

# Serial Drivers



# CREW Manual

## Mitsubishi - FX2-3 series

### Communication cable and/or connection type description

To set up a connection with the PLC, you need a CPU with RS232 port or connect an adapter card of this model:

FX2N-232-BD

FX3G-232-BD

### Connection parameters (PANEL)

COM interchar timeout: the maximum amount of time that elapses between characters (inter-character) in a reply message.

### Connection parameters (DEVICE)

No information.

### Memory areas

Access to the memory areas is generic PLC standard, with the exception of some of the areas that particular notes apply to.

1. “Internal relay”, “States” area: DECIMAL base addressing in “Bit” format (M0, M1, M2, etc.) and access to Bit/Word/Dword. If you wish to use a type of Word or Dword, you will need to use addressing in multiples of 16 (MWO, MW16, MW32, MD48, etc.).
2. “Input”, “Output” area: OCTAL base addressing in “Bit” format (M0, M1, M2, etc.) and access to Bit / Word / Dword. If you wish to use a type of Word or Dword, you will need to use addressing in multiples of 16 (MWO, MW16, MW32, MD48, etc.).

# CREW Manual

## Error codes

The driver can report, in the system variable, the following standard error codes :

Nome	Codice
DRIVER ERROR	Unable to send the request message, possible problem with Serial board
PROTOCOL ERROR	Generic error of data receipt from the PLC
PROTOCOL TIMEOUT	Timeout error, there has been no response to a request for data
PROTOCOL OFFLINE	Device not connected because it does not respond to PING command
SOCKET ERROR	Error during creation of Ethernet socket, the device does not respond
ERROR	Value reported in the event of mishandling driver

# CREW Manual

## Mitsubishi - FX5 MC binary type protocol

### Communication cable and/or connection type description

To set up a connection with the PLC through this binary MC protocol, from the configuration software it is necessary (in the configuration of the RS485 port) to enable the following parameters:

#### Basic Settings

Communication Protocol Type  
Communication protocol type: MC protocol

#### Advanced settings

Data length: 8 bit  
Parity bit: None / Even / Odd  
Stop bit: 1 bit  
Baud rate: 4800 - 115200  
Sum check code: Added

#### Fixed Settings

Station Number: Set the desired station number  
Message pattern: Pattern 5  
Time-out period: 10ms.

By downloading the configuration into the PLC communication should run correctly with the operator panel.

#### Connection parameters (PANEL)

Serial parameters: selectable 4800-115200, (None - Even - Odd), 8, 1

COM interchar timeout: the maximum amount of time that elapses between characters (inter-character) in a reply message.

# CREW Manual

## Connection parameters (DEVICE)

### Mitsubishi Network

Station No.: Address of the PLC station

Network No.: 00h .. FFh

PC No.: 00h .. FFh

Request destination module IO No.: 0000h .. FFFFh

Request destination module station No.: 00h .. FFh

The list of “Mitsubishi network” parameters is used for the configuration of the connection to a Mitsubishi CPU network:

- When set with the default values, this makes it possible to connect to a single CPU to the specified address.
- Change the parameters to address various CPUs in a Mitsubishi-type sub-network (see the “Mitsubishi FX5 Series User Manual, MELSEC communication protocol” on page 25).

### Memory areas

No specific note.

# CREW Manual

## Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
DRIVER ERROR	Unable to send the request message, possible problem with Ethernet board
PROTOCOL ERROR	Generic error of data receipt from the PLC
PROTOCOL TIMEOUT	Timeout error, there has been no response to a request for data
PROTOCOL OFFLINE	Device is offline, there is no response from the device when the ethernet connection
ERROR	Value reported in the event of mishandling driver

# CREW Manual

## Modbus Master RTU

### Communication cable and/or connection type description

No description available.

### Connection parameters (PANEL)

Protocol timeout: the value of the maximum wait time before reporting a device response error.

Idle chars: wait time measured in “RS232 characters” before the panel sends a new request to the serial port.

COM interchar timeout: the maximum amount of intercharacter time in a reply message.

### Connection parameters (DEVICE)

Device address: address of the slave you need to connect to.

## Memory areas

The provided Modbus RTU access functions are as follows:

- FC03-06, for reading and writing a single FC03-06 element, for reading and writing a single Register (Word) element.
- FC03-16, for reading and writing multiple FC03-16 elements, for reading and writing multiple Register (Word) elements.
- FC03-16, for reading and writing multiple FC03-16 elements, for reading and writing multiple Long Register (DWord) elements.

# CREW Manual

- FC02, for reading multiple FC02 elements, for reading multiple Input Status (Bit) elements.
- FC04, for reading multiple FC04 elements, for reading multiple Input Status (Word) elements.
- FC01-05, for reading multiple consecutive Coils elements, and single writing (Bit).

## Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
DRIVER ERROR	Unable to send the request message, possible problem with Serial board
PROTOCOL ERROR	Generic error of data receipt from the PLC
PROTOCOL TIMEOUT	Timeout error, there has been no response to a request for data
PROTOCOL OFFLINE	Device not connected because it does not respond to PING command
ERROR	Value reported in the event of mishandling driver

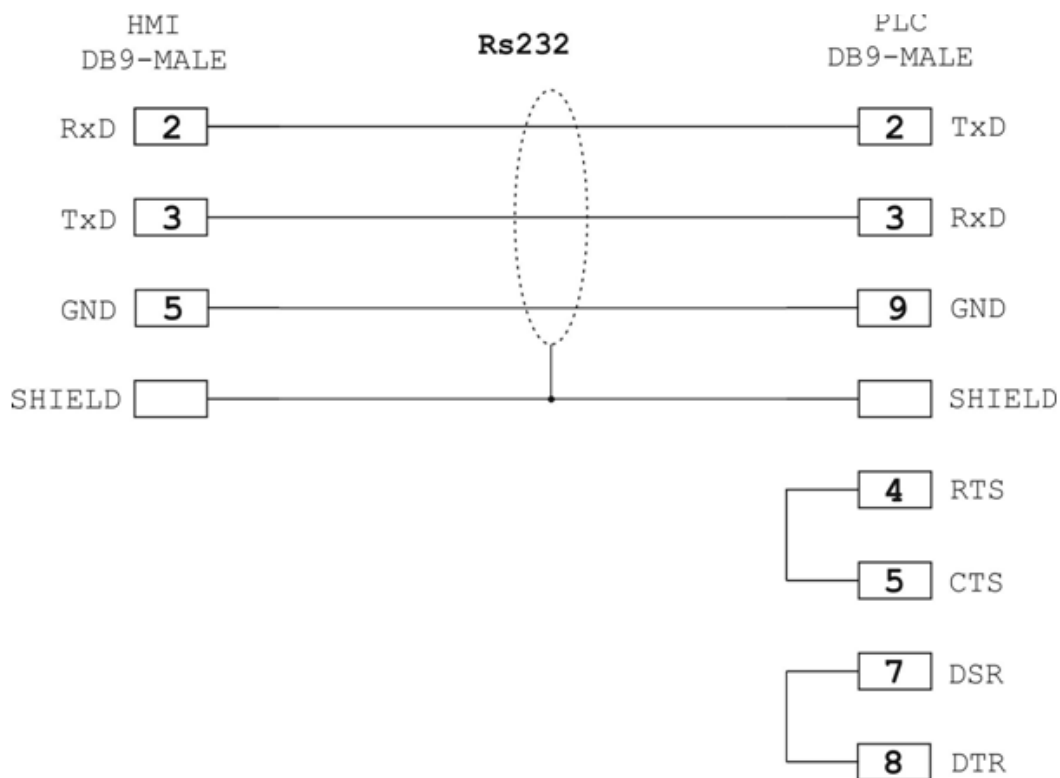


# CREW Manual

## Omron FINS protocol

Communication cable and/or connection type description.

Connection with Omron CS1 model on peripheral port.



### Connection parameters (PANEL)

No description available.

### Connection parameters (DEVICE)

Device address: address of the PLC entered from the programming software.

# CREW Manual

## Memory areas

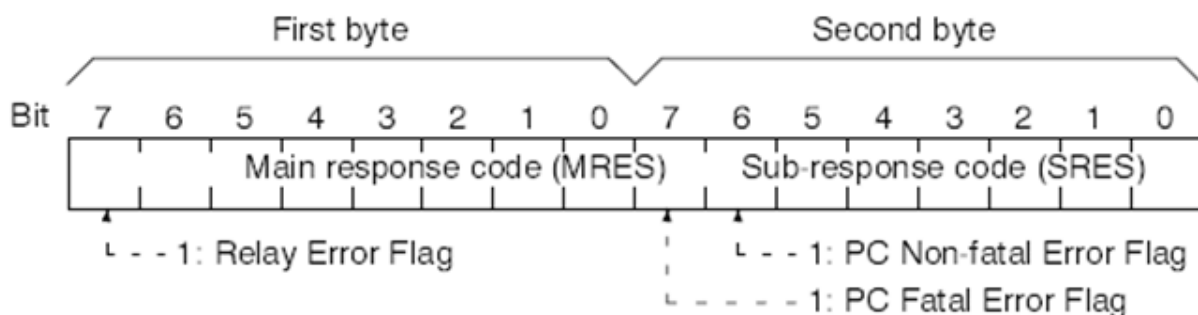
No particular features for the areas of the memory that are accessed through the Ethernet protocol. All of the areas are Word-based size, or some also Bit-based.

## Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
DRIVER ERROR	Unable to send the request message, possible problem with Serial board
PROTOCOL ERROR	Generic error of data receipt from the PLC
PROTOCOL TIMEOUT	Timeout error, there has been no response to a request for data
PLC CODE ERR: XXXX	See description on table below
ERROR	Value reported in the event of mishandling driver

For “PLC CODE ERR: XXXX” type of errors the XXXX values have a specific meaning, since the PLC replied with an error code that is explained in the image below.



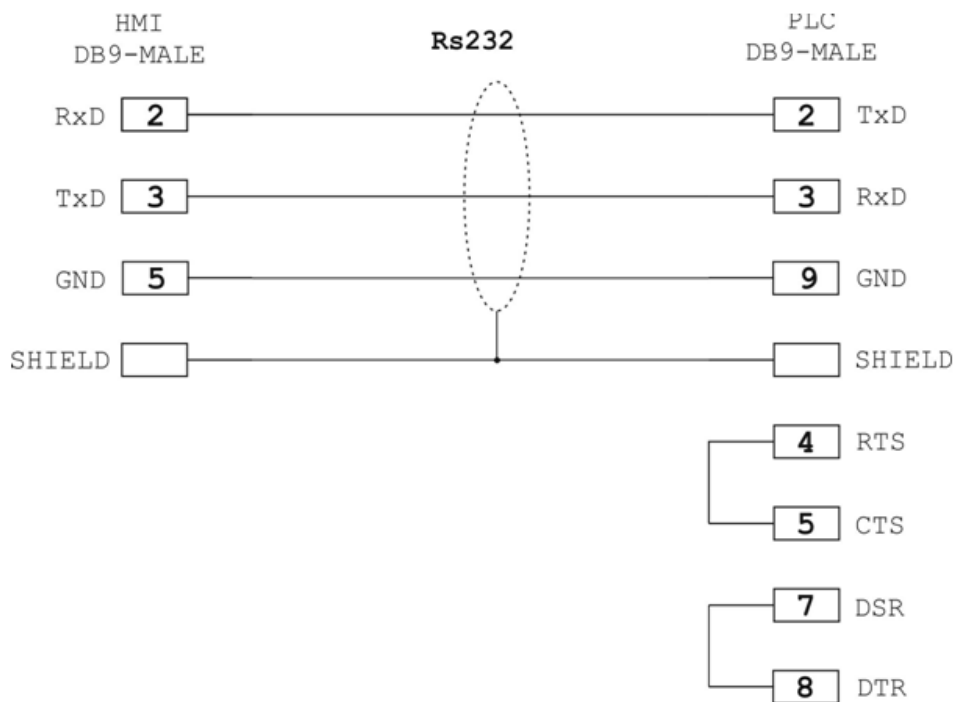
Note: The detailed description of the error codes can be viewed in the original PDF manual “Omron\_FINS\_W227E11.pdf”.

# CREW Manual

## Omron Host Link series

### Communication cable and/or connection type description

Connection with **Omron CS1** model.



### Connection parameters (PANEL)

The connection parameters are relative to the COM serial setting:

Baud Rate: 9600, 19200  
 Parity: Even  
 Data Bit: 7  
 Stop Bit: 2

# CREW Manual

## Connection parameters (DEVICE)

Device address: the address of the slave device.

## Memory areas

No particular features for the areas of the memory that are accessed through the serial protocol. All of the areas are Word-based size.

## Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
DRIVER ERROR	Unable to send the request message, possible problem with Serial board
PROTOCOL ERROR	Generic error of data receipt from the PLC
PROTOCOL TIMEOUT	Timeout error, there has been no response to a request for data
PLC CODE ERR: XXXX	See description on table below
ERROR	Value reported in the event of mishandling driver

# CREW Manual

For “PLC CODE ERR: XX” type of errors the XX value has a specific meaning, since the PLC replied with an error code that is explained in the table below.

Code (high)	Code (low)	Description
0	0	No error
0	1	Not executable in RUN mode
0	2	Not executable in MONITOR mode
0	3	Not executable with PROM mounted
0	4	Address out of range (Data overflow)
0	B	Not executable in PROGRAM mode
0	C	Not executable with PROM mounted
0	D	Not executable in LOCAL mode
1	0	Parity error
1	1	Frame error
1	2	Overrun
1	3	Checksum error
1	4	Formatting error (parameter length error)
1	5	Incorrect setting value
1	6	Instruction not found
1	8	Wrong frame length
1	9	Not executable
2	0	Wrong I / O table generation

# CREW Manual

## Siemens - PPI protocol Slave mode

### Communication cable and/or connection type description

The driver makes it possible to communicate with PLC Siemens S7-200 set in SLAVE mode, i.e. that do not have an active network (Token Pass) communication mode.

It is possible to set up the connection with multiple PLCs at the same time, but it is not compatible with the programming software (and relative Siemens adapter).

### Connection parameters (PANEL)

Terminal address: address of the operator panel.

Protocol timeout (msec.): maximum set amount of time before a communication with the PLC timeout is detected (no response).

### Connection parameters (DEVICE)

PLC address: the address of the PLC device.

Max retry numbers: the maximum number of data request sending retries before a communication error is reported on the panel page.

### Memory areas

No description.

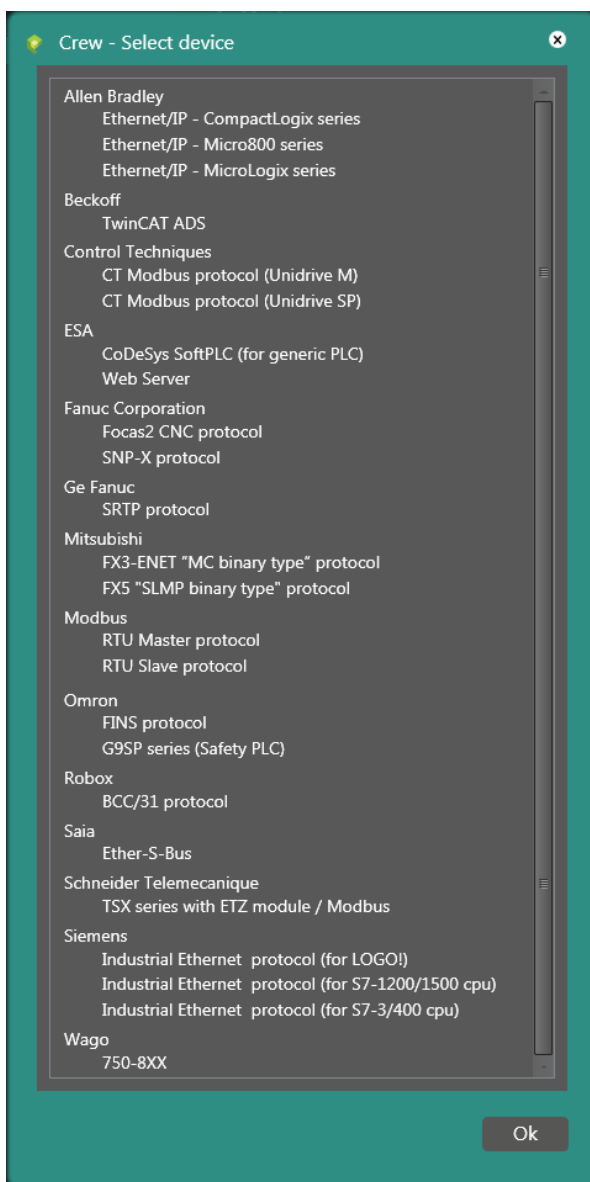
### Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
PROTOCOL ERROR	Generic error of data receipt from the PLC
PROTOCOL TIMEOUT	Timeout error, there has been no response to a request for data
PLC BUSY	PLC is not able to properly respond to a reading data request, because of resources missing

# CREW Manual

# Ethernet Drivers



# CREW Manual

## Allen Bradley - Ethernet-IP for ControlLogix

### Communication cable and/or connection type description

No description available.

### Connection parameters (PANEL)

No parameter.

### Connection parameters (DEVICE)

IP address: IP address of the device you need to connect to.

IP port: port number of the device you need to connect to.

### Memory areas

The communication protocol makes it possible to access reading and writing all of the basic tags contained in the “Controller Tags” folder of the Allen Bradley software. The available base formats are:

- BOOL, boolean data type.
- SINT, 8bit data type.
- INT, 16bit data type.
- DINT, 32bit data type.
- REAL, 32bit floating point type data.

The STRING type is the only composed type of data (LEN part and DATA part) that it is possible to read and write directly from the panel, in its original form (composed).

STRING-type data does not support special characters \$\$, \$', \$L, \$N, \$P, \$R, \$T. Therefore, using them can lead to anomalous behaviour of the string field.

All STRUCTURE-type data (structures composed of simple data) are accessible with the structure addressing method, using the coherent base type of the datum that you wish to read.



# CREW Manual

ADDRESSING FORMAT		
DESCRIPTION	ADDRESS TAG VALUE	NOTE
Simple addressing tag INT "Test" type	Test	
Reading of an array Tag DINT "MyVar" type of 32 items	MyVar	The same size declared in Allen Bradley software must be present in Crew software (Array dimension = 32)
Reading element 6 by an array SINT "myExample" Tag type 32 elements, using in Crew the same size declared in Allen Bradley software	MyExample[6]	It should be specified in square brackets the index of the desired item
Reading of an element of a user-defined structure. In the example we read the data "VAR" BOOL type that is present in the "Structure" structure	Structure.VAR	
In order to read and write an element of a COUNTER type data, refer to the addressing of a structure element.	MyCounter.PRE	The usable elements available for reading and writing are: PRE, ACC, CU, CD, DN, OV, AN.
In order to read and write an element of a given TIMER type, refer to the addressing of a structure element.	MyTimer.PRE	The usable elements available for reading and writing are: PRE, ACC, EN, TT, DN, FS, LS, OV, ER.

# CREW Manual

## Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
DRIVER ERROR	Unable to send the request message, possible problem with Ethernet board
PROTOCOL ERROR	Generic error of data receipt from the PLC
PROTOCOL TIMEOUT	Timeout error, there has been no response to a request for data
SOCKET ERROR	Error during creation of Ethernet socket, the device does not respond
DEVICE OFFLINE	Device not connected because it does not respond to PING command
TRANSMISSION ERROR	An error occurred while attempting to transmit from the ethernet port of the panel

# CREW Manual

## Control Techniques - CT Modbus -

### Communication cable and/or connection type description:

No description.

### Connection parameters (PANEL)

No parameter.

### Connection parameters (DEVICE)

IP address: IP address of the device you need to connect to.

IP port: port number of the device you need to connect to.

## Memory areas

### AREAS:

CT Parameter: to read and write the parameters of the device connected in local.

CT-Net Register: to read and write the parameters of a device connected to the network of the main device in CT-NET.

### ADDRESSING

Menu/Parameter: the main parameters for addressing the parameter of a local or networked device.

CT-Net Slot: slot number of the networked device that the CT-NET card is positioned in.

CT-Net Node: network address of the CT-NET device.

# CREW Manual

## Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
DRIVER ERROR	Unable to send the request message, possible problem with Ethernet board
PROTOCOL ERROR	Generic error of data receipt from the PLC
PROTOCOL TIMEOUT	Timeout error, there has been no response to a request for data
SOCKET ERROR	Error during creation of Ethernet socket, the device does not respond
DEVICE OFFLINE	Device not connected because it does not respond to PING command
TRANSMISSION ERROR	An error occurred while attempting to transmit from the ethernet port of the panel
ERROR	Value reported in the event of mishandling driver

# CREW Manual

## ESA - Codesys SoftPLC -

### Description of type of connection:

The CODESYS SoftPLC driver (generic) is a communication protocol that makes it possible to connect to a PLC that is not on the software list, but that supports the CODESYS-type of connection.

In this case it is crucial to know, also through PLC 3S software or similar, the type of connection set up between PLC and programming software (connection parameters).

The ESA driver is in fact configurable for the majority of the CODESYS connection parameters and supports models ARTI V2, GATEWAY V2, ARTI V3, GATEWAY V3.

### Connection parameters (PANEL)

No description available.

### Connection parameters (DEVICE)

The device's connection is set up through the correct configuration of the CODESYS connection parameters. It is possible to enable or disable a parameter through the relative "Download" box, where you can choose which parameters to keep active in communication.

### Memory areas

The PLC variables are imported (with specific ESA utility) from a specific file generated through the PLC programming software.

### Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
INTERCHAR ERROR	Connection error with CoDeSys server
PROTOCOL OFFLINE	Error during creation of Ethernet socket, the device does not respond

# CREW Manual

## Fanuc Robotics - Focas2 CNC protocol Tcp IP

### Communication cable and/or connection type description

No description available.

### Connection parameters (PANEL)

No description available.

### Connection parameters (DEVICE)

IP Address: the device's IP address.

IP Port: connection IP port (8193).

Connection timeout (sec.): the timeout of the connection with the device, managed by the Fanuc Focas library in seconds

Custom PMC unit selection. If enabled, all of the PMC tags use the “PMC Unit” parameter to identify which unit the datum is requested from. When disabled, the PMC is the default (1st Unit).

## Memory areas

Fanuc Robotics FOCAS2 libraries provide access to various types of data contained in the Fanuc CNC controllers. Currently “PMC”, “MACRO variables”, “P-CODE variables”, “ALARM text” and “MESSAGE text” memory areas are supported.

## PMC area

The PMC area is divided into various types contained in the controller (G, F, Y, X, A, R, T, K, C, D, M, N, E, Z) and can be read in “BYTE”, “WORD” and “DWORD” format. To address it, the following is necessary:

- **PMC Unit:** this identifies the PMC unit, it is only valid when the “Custom PMC unit selection” parameter is enabled.
- **Path No.:** the PATH number that variable is being read from (when there are multiple PATHs in the CNC controller).
- **Address:** variable address.

# CREW Manual

## MACRO variables / P-CODE variables

This area makes it possible to read and write the MACRO or P-CODE variables, and it differs from a classic variable as, in addition to a floating numeric value, it can also contain a value of nil which is highlighted as “DATA EMPTY”. The types in the area are, therefore, “FLOATING (64-BIT)” and “VALIDATION”.

To read the value contained in the variable use the “FLOATING (64-BIT)” type and the datum will be read in floating point (with DATA EMPTY it will show 0).

To read the validity status of the variable, use the “VALIDATION” type that returns the following values:

- 0, if the variable is not valid (DATA EMPTY).
- 1, if the variable contains a floating value.

To invalidate the variable (relative to the DATA EMPTY value) use the “VALIDATION” type and run writing (for example, from SET direct command) with a value of “1”.

Addressing requires the following parameters:

- Path No.: the PATH number that variable is being read from (when there are multiple PATHs in the CNC controller).
- Address: variable address.

## Messages (Alarm)/Messages (Operator)

This area makes it possible to read the alarm/message texts generated by PMC.

Addressing requires the following parameters:

- Path No.: the PATH number that variable is being read from (when there are multiple PATHs in the CNC controller).
- Address: index of text in queue.

## Program Name

This area makes it possible to read the name of the program currently running. The information that can be read is:

# CREW Manual

- Path String (Ansi), i.e. the full path of the program in execution, along with the program name.
- Name String (Ansi), i.e. only the name of the program in execution.
- Number (Byte), i.e. the number of the program in execution.

Addressing requires the following parameters:

- Path No.: the PATH number that variable is being read from (when there are multiple PATHs in the CNC controller).

## Sequence number

This area makes it possible to read the number of the sequence currently running.

- Path No.: the PATH number that variable is being read from (when there are multiple PATHs in the CNC controller).

## CNC Axis Position

This area makes it possible to read the information of an axis, with the following data:

- Value (Double), the current position of the axis.
- Unit (Byte), the enumerative value of the unit of measure (codes in the FOCAS manual).
- Name (String), name attributed to the requested axis (max 4 bytes).

To address the datum you need to enter the following parameters:

- Path No.: the PATH number that variable is being read from (when there are multiple PATHs in the CNC controller).
- Type: choose from “Absolute / Machine / Relative”.
- Axis No: the number of the requested axis.



# CREW Manual

## CNC Axis Servo

This area makes it possible to read the loading information (LoadMeter) for an axis, with the following data:

- Value (Double), the current position of the axis.
- Unit (Byte), the enumerative value of the unit of measure (codes in the FOCAS manual).
- Name (String), name attributed to the requested axis (max 4 bytes).

To address the datum you need to enter the following parameters:

- Path No.: the PATH number that variable is being read from (when there are multiple PATHs in the CNC controller).
- Type: choose from “Load meter / Load counter (% unit) / Load counter (Ampere)”.
- Axis No: the number of the requested axis.

## CNC Axis Spindle

This area makes it possible to read the loading information (LoadMeter) for a spindle, with the following data:

- Value (Double), the current position of the axis.
- Unit (Byte), the enumerative value of the unit of measure (codes in the FOCAS manual).
- Name (String), name attributed to the requested axis (max 4 bytes).

To address the datum you need to enter the following parameters:

- Path No.: the PATH number that variable is being read from (when there are multiple PATHs in the CNC controller).
- Type: choose from “Load meter / Motor speed”.
- Axis No: the number of the requested axis.

# CREW Manual

## Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
<b>DRIVER ERROR</b>	Generic error of the ethernet driver interface with the Fanuc robotics library
<b>PING ERROR</b>	Error is displayed if the device does not respond to the of the PING panel message (possible disconnected cable)

Or, if an error is reported by the Fanuc FOCAS library, the return numerical value is used along with a string code.

For example, if the library returns “Socket error (Ethernet version only)” the “EW\_SOCKET (-16)” message will appear.

# CREW Manual

## Fanuc Robotics SNP-X protocol - Tcp IP -

### Communication cable and/or connection type description:

The protocol is designed for FANUC ROBOTICS controllers, that control a series of robotic controls.

### Connection parameters (PANEL)

No description available.

### Connection parameters (DEVICE)

IP Address: the device's IP address.

IP Port: connection IP port (18245).

### Memory areas

No particular features for the areas of the memory that are accessed through the Ethernet protocol.

# CREW Manual

## Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
DRIVER ERROR	Unable to send the request message, possible problem with Ethernet board
PROTOCOL ERROR	Generic error of data receipt from the PLC
PROTOCOL TIMEOUT	Timeout error, there has been no response to a request for data
SOCKET ERROR	Error during creation of Ethernet socket, the device does not respond
TRANSMISSION ERROR	An error occurred while attempting to transmit the ethernet port of the panel
RECEIVE ERROR	Error during the initial connection with the device
ERROR	Value reported in the event of mishandling driver

# CREW Manual

GeFanuc - SRTP protocol -

**Communication cable and/or connection type description:**

No description.

**Connection parameters (PANEL)**

No parameter.

**Connection parameters (DEVICE)**

IP address: IP address of the device you need to connect to.

IP port: port number of the device you need to connect to.

**Memory areas**

No description.

# CREW Manual

## Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
DRIVER ERROR	Unable to send the request message, possible problem with Ethernet board
PROTOCOL ERROR	Generic error of data receipt from the PLC
PROTOCOL TIMEOUT	Timeout error, there has been no response to a request for data
SOCKET ERROR	Error during creation of Ethernet socket, the device does not respond
DEVICE OFFLINE	Device not connected because it does not respond to PING command
TRANSMISSION ERROR	An error occurred while attempting to transmit the ethernet port of the panel
ERROR	Value reported in the event of mishandling driver

# CREW Manual

## Mitsubishi MC protocol FX2 - 3 Tcp

### Communication cable and/or connection type description

To set up a connection with the PLC, through this binary MC protocol, you need a CPU with an ethernet port or connect an adapter card of this model:

FX3U-ENET-ADP  
FX3U-ENET-ADP

In both cases, from the configuration software, it is necessary to enable the “MC binary protocol” inside the configuration of the Ethernet port (in one of the available and selectable slots) with relative port 1025.

### Connection parameters (PANEL)

No description available.

### Connection parameters (DEVICE)

IP address: IP address of the device.

Port: TcpIP port you need to connect to (1025 default).

# CREW Manual

## Memory areas

Access to the memory areas is generic PLC standard, with the exception of some of the areas that particular notes apply to:

1. “Internal relay”, “States” area: DECIMAL base addressing in “Bit” format (M0, M1, M2, etc.) and access to Bit/Word/Dword. If you wish to use a type of Word or Dword, you will need to use addressing in multiples of 16 (MWO, MW16, MW32, MD48, etc.)
  
2. “Input”, “Output” area: OCTAL base addressing in “Bit” format (M0, M1, M2, etc.) and access to Bit/Word/Dword. If you wish to use a type of Word or Dword, you will need to use addressing in multiples of 16 (MWO, MW16, MW32, MD48, etc.). “Input”, “Output” area: OCTAL base addressing in “Bit” format (M0, M1, M2, etc.) and access to Bit/Word/Dword. If you wish to use a type of Word or Dword, you will need to use addressing in multiples of 16 (MWO, MW16, MW32, MD48, etc.)

## Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
DRIVER ERROR	Unable to send the request message, possible problem with Ethernet board
PROTOCOL ERROR	Generic error of data receipt from the PLC
PROTOCOL TIMEOUT	Timeout error, there has been no response to a request for data
PROTOCOL OFFLINE	Device is offline, there is no response from the device when the ethernet connection
ERROR	Value reported in the event of mishandling driver



# CREW Manual

## Mitsubishi - FX5 SLMP binary type protocol

### Communication cable and/or connection type description

To set up a connection with the PLC through this binary SLMP protocol, from the configuration software it is necessary (in the Ethernet port configuration) to enable the “SLMP binary protocol” (in one of the available and selectable slots) with relative port 1025.

By downloading the configuration into the PLC communication should run correctly with the operator panel.

### Connection parameters (PANEL)

No description available.

### Connection parameters (DEVICE)

IP Address: the device's IP address.

Port: TcpIP port you need to connect to (1025 default).

### Mitsubishi specific parameters

Destination network NO.: 00h .. FFh

Destination station NO.: 00h .. FFh

Destination module NO.: 0000h .. FFFFh

Destination multidrop station NO.: 00h .. FFh

The list of “Mitsubishi specific parameters” is used for the configuration of the connection to a Mitsubishi CPU network:

- When set with the default values, this makes it possible to connect to a single CPU to the specified IP address.
- Change the parameters to address various CPUs in a Mitsubishi-type sub-network (see the “Mitsubishi FX5 Series User Manual [SLMP]” on page 22).

# CREW Manual

## Memory areas

No specific note.

## Error codes

The driver can report, in the system variable, the following standard error codes:

Nome	Codice
DRIVER ERROR	Unable to send the request message, possible problem with Serial board
PROTOCOL ERROR	Generic error of data receipt from the PLC
PROTOCOL TIMEOUT	Timeout error, there has been no response to a request for data
PROTOCOL OFFLINE	Device not connected because it does not respond to PING command
SOCKET ERROR	Error during creation of Ethernet socket, the device does not respond
ERROR	Value reported in the event of mishandling driver

# CREW Manual

## Modbus Master RTU Tcp

### Communication cable and/or connection type description

No description available.

### Connection parameters (PANEL)

No description available.

### Connection parameters (DEVICE)

IP Address: the device's IP address.

IP Port: connection IP port (502).

Device address: address of the slave you need to connect to.

## Memory areas

The provided Modbus RTU access functions are as follows:

- FC03-06, for reading and writing a single Register element (Word).
- FC03-16, for reading and writing multiple Register elements (Word).
- FC03-16, for reading and writing multiple Long Register elements (Dword).
- FC02, for reading multiple Input Status (Bit) elements.
- FC04, for reading multiple Input Status (Word) elements.

# CREW Manual

- FC01-05, for reading multiple consecutive Coils elements, and single writing (Bit) .

## Error codes

The driver can report, in the system variable, the following standard error codes:

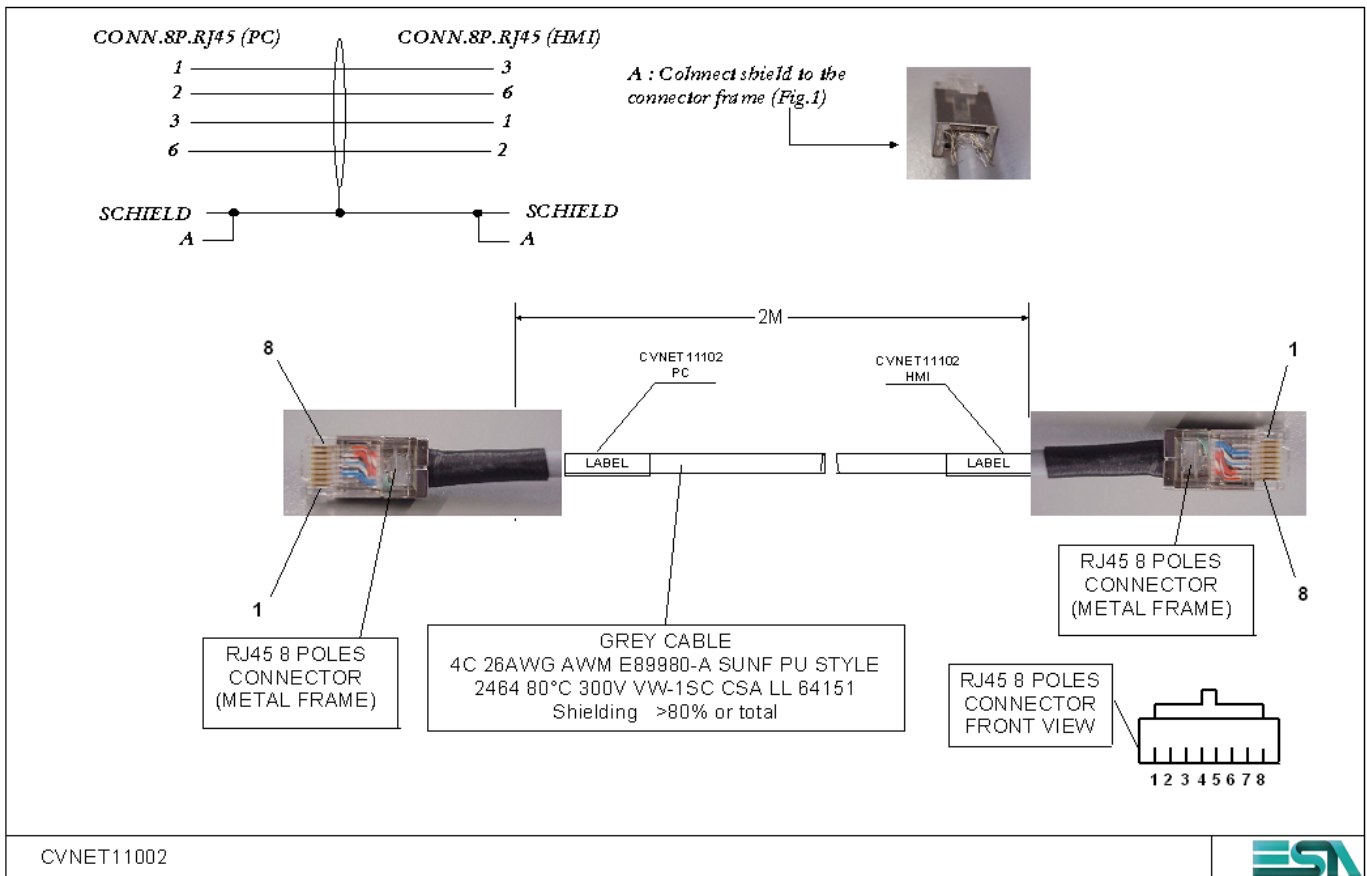
Name	Code
PROTOCOL ERROR	Generic error of data receipt from the PLC
PROTOCOL TIMEOUT	Timeout error, there has been no response to a request for data
SOCKET ERROR	Error during creation of Ethernet socket, the device does not respond
DEVICE OFFLINE	Device not connected because it does not respond to PING command
TRANSMISSION ERROR	An error occurred while attempting to transmit from the ethernet port of the panel
RECEIVE ERROR	Error during the initial connection with the device
ERROR	Value reported in the event of mishandling driver

# CREW Manual

## Modbus RTU Slave Tcp IP

Communication cable and/or connection type description:

### CVNET11002



Note: Only one device can be connected to the panel since there is one driver and only one slave, with a determined Modbus address (specified in the parameters).

### Connection parameters (PANEL)

No description available.

# CREW Manual

## Connection parameters (DEVICE)

IP Address: the device's IP address.

IP Port: connection IP port (502).

Device address: address of the slave (panel).

## Memory areas



Note: Generally the internal areas are divided into two distinct ones: COILS (Bit) and REGISTERS (Word).

The provided Modbus RTU access functions are as follows:

- FC03, for reading a single element of the Register area in Word format.
- FC06, for writing a single element of the Register area in Word format.
- FC10, for writing multiple consecutive elements of the Register area (max 32 Word).
- FC01, for reading multiple consecutive elements of the Coils area in BIT format (max 32 Bit).
- FC05, for writing a single element of the Coils area in BIT format.
- FC08, Modbus diagnostic message to which the same message is replicated.

## Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
DRIVER ERROR	Unable to send the request message, possible problem with Ethernet board
PROTOCOL ERROR	Generic error receiving message from Master
ERROR	Value reported in the event of mishandling driver

# CREW Manual

## Omron FINS protocol Tcp

### Communication cable and/or connection type description

The driver connects to a FINS network, it is therefore necessary to enter the right settings on the PLC so that it can communicate correctly.

Automatic client node allocation (HMI) by the PLC server must be enabled in the software options, so that you do not need to explicitly specify a panel node (which is supplied by the PLC).

The network it can communicate in is the LOCAL one (namely with NetworkAddress = 0). It is currently not possible to communicate in FINS Omron networks outside of the local one.

### Connection parameters (PANEL)

No description available.

### Connection parameters (DEVICE)

IP Address: the device's IP address.

IP Port: connection IP port (9600).

### Memory areas

No particular features for the areas of the memory that are accessed through the ethernet protocol, all of the areas have Word-base size, or some also Bit-base.

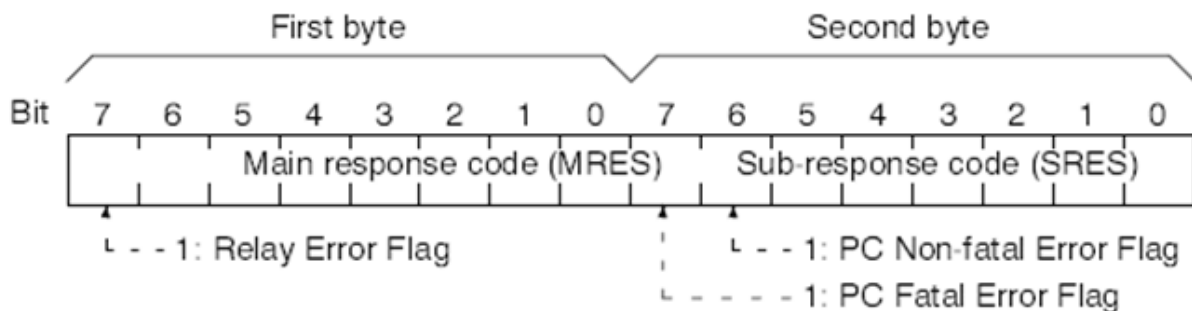
# CREW Manual

## Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
DRIVER ERROR	Unable to send the request message, possible problem with Serial board
PROTOCOL ERROR	Generic error of data receipt from the PLC
PROTOCOL TIMEOUT	Timeout error, there has been no response to a request for data
SOCKET ERROR	Error during creation of Ethernet socket, the device does not respond
PLC CODE ERR: XXXX	See description on table below
ERROR	Value reported in the event of mishandling driver

For “PLC CODE ERR: XXXX” type of errors the XXXX values have a specific meaning, since the PLC replied with an error code explained in the image below :



Note: The detailed description of the error codes can be viewed in the original PDF manual “Omron\_FINS\_W227E11.pdf”.



# CREW Manual

## SAIA - Ether S-BUS

### Communication cable and/or connection type description

No description available.

### Connection parameters (PANEL)

No description available.

### Connection parameters (DEVICE)

IP Address: the device's IP address.

Destination Unit: the logical address of the destination PLC.

Retries: number of retries of forwarding the message in case of connection error.

IP Port: IP SAIA port number. Default value of 5050 for all connections.

### Memory areas

#### Data area

Data Area	Field Name	Type	Range	
Flag	F	Bit	0..16383	
Input	I	Bit	0..8191	
Output	O	Bit	0..8191	
Register	R	Word	0..16383	
Timer	T	Word	0..1599	
Counter	C	Word	0..1599	
Text	Text	Byte	Text	Character
			0..8191	0..3071
Data Block	DB	Word	DB	DW
			0..8191	0..16383

# CREW Manual

## Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
<b>DRIVER ERROR</b>	Unable to send the request message, possible problem with Serial board
<b>PROTOCOL ERROR</b>	Generic error of data receipt from the PLC
<b>PROTOCOL TIMEOUT</b>	Timeout error, there has been no response to a request for data
<b>PROTOCOL OFFLINE</b>	Device not connected because it does not respond to PING command
<b>SOCKET ERROR</b>	Error during creation of Ethernet socket, the device does not respond
<b>PING FAIL</b>	The device does not respond to a standard ethernet PING request
<b>FORMAT DATA ERR</b>	The value (or string) inserted into the field when writing is not consistent with the rules of format allowed
<b>ERROR</b>	Value reported in the event of mishandling driver

# CREW Manual

Schneider Telemecanique TSX series with ETZ module (Tcp)

## Communication cable and/or connection type description

The device installs exactly the Modbus MASTER Tcp/IP protocol.

## Connection parameters (PANEL)

No description available.

## Connection parameters (DEVICE)

IP Address: the device's IP address.

IP Port: connection IP port (502).

Device address: address of the slave you need to connect to.

## Memory areas

The device uses the FC03-06 standard Modbus command as the only data memory, for reading and writing a single Register element (Word).

# CREW Manual

## Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
PROTOCOL ERROR	Generic error of data receipt from the PLC
PROTOCOL TIMEOUT	Timeout error, there has been no response to a request for data
SOCKET ERROR	Error during creation of Ethernet socket, the device does not respond
DEVICE OFFLINE	Device not connected because it does not respond to PING command
TRANSMISSION ERROR	An error occurred while attempting to transmit from the ethernet port of the panel
RECEIVE ERROR	Error during the initial connection with the device
ERROR	Value reported in the event of mishandling driver

# CREW Manual

## Siemens Industrial Ethernet for LOGO

### Communication cable and/or connection type description

No description available.

### Connection parameters (PANEL)

No description available.

### Connection parameters (DEVICE)

IP Address: the device's IP address

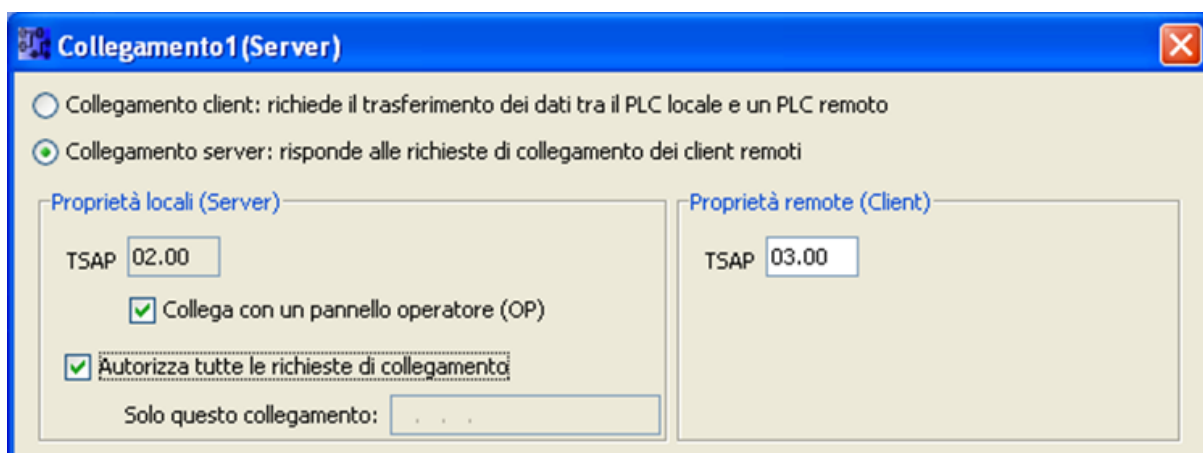
Source TSAP: source TSAP value.

Destination TSAP: destination TSAP value.

The IP Port number is set at 102.

It is possible to set the TSAP parameters from the LOGO configurator. Therefore it is necessary to use the same entered parameters to configure the connection towards PLC LOGO.

If the PLC is used as a server, the default setting of the device's TSAP is 0x0200. The source TSAP must be the same as the one set as TSAP through the configurator (Client). The software used for the configuration is LOGO Soft Comfort V7.



# CREW Manual

## Memory areas

Unlike devices in the same S7-200/300 category, the LOGO device has the same variations in the structure of the memory areas.

As for MERKER, INPUT and OUTPUT type areas, no access was made to their value through the <Byte.Bit> structure (for example M1.2) but directly through their bit (for example M1, M9, etc.).

In particular, the analogue values for the same areas are also available: Analog INPUT, Analog OUTPUT, Analog MERKER.

This data is mapped in Word in the same area as the VBs and after them.

They are mapped with the following offset in bytes:

Analog input: 926 Analog input: 926

Analog output: 944 Analog output: 944

Analog merker: 952 Analog merker: 952

# CREW Manual

## Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
DRIVER ERROR	Unable to send the request message, possible problem with Serial board
PROTOCOL ERROR	Generic error of data receipt from the PLC
PROTOCOL TIMEOUT	Timeout error, there has been no response to a request for data
PROTOCOL OFFLINE	Device not connected because it does not respond to PING command
SOCKET ERROR	Error during creation of Ethernet socket, the device does not respond
PING FAIL	The device does not respond to a standard ethernet PING request
FORMAT DATA ERR	The value (or string) inserted into the field when writing is not consistent with the rules of format allowed
ERROR	Value reported in the event of mishandling driver

# CREW Manual

## Siemens Industrial Ethernet

### Communication cable and/or connection type description

No description available.

### Connection parameters (PANEL)

No description available.

### Connection parameters (DEVICE)

IP Address: the device's IP address

Expansion Slot: the number of slots that the CPU is positioned in the rack.

The “Expansion slot” default value has the following values:

- For S7 300/400 the value is 2.
- For S7 1200 the value is 1.

### Memory areas

The protocol provides access to the main areas of the memory in Word or Bit base format.

The COUNTER and TIMER data is now in binary format and it is no longer necessary to specify BCD because the driver does the conversion.

COUNTER data counts a fixed range of 0 .. 999.

TIMER (or SIMATIC TIME) data counts according to the following rules:



# CREW Manual

- 0ms .. 9s990ms (with time base = 0, namely 1/100s base.)
- 0ms .. 1m30s990ms (with time base = 1, namely 1/10s base.)
- 0s .. 16m39s (with time base = 2, namely 1s base.)
- 0s .. 2h46m30s (with time base = 2, namely 10s base.)

There are two areas that operate differently from the standard of a simple TAG, namely:

- Timer area
- Data Block area (Simatic Time)

The following type of data is contained in these areas:

TimeBase = 1/100s.

TimeBase = 1/10s.

TimeBase = 1s.

TimeBase = 10s.

TimeBase = AUTO (1ms.)

## String Format

Representation / Setting of Timer, Data Block types (SimaticTime)

READ: fixed time base, LONG numerical format, representation 0..999000 (x10ms.)

WRITE: fixed time base (x10ms), LONG numerical format, admissible values 0..999

# CREW Manual

READ: fixed time base, LONG numerical format, representation 0..99900  
(x100ms.)

WRITE: fixed time base (x100ms), LONG numerical format, admissible values  
0..999

READ: fixed time base, LONG numerical format, representation 0..9990 (x1s.)

WRITE: fixed time base (x1s), LONG numerical format, admissible values 0..999

READ: fixed time base, LONG numerical format, representation 0..999 (x10s.)

WRITE: fixed time base (x10s), LONG numerical format, admissible values 0..999

READ: automatic time base (x1ms), LONG num. format, representation  
0..9990000

WRITE: automatic time base (x1ms), LONG num. format, admissible values  
0..9990000

Note: The driver automatically adapts the time base in write.

READ: automatic time base (x1ms), STRING format, repr. ##h##m##s###ms

WRITE: automatic time base (x1ms), STRING format, admissible values:

###ms (ex: 100ms - 450ms - 30ms)

##s###ms (ex: 4s100ms - 6s450ms - 15s30ms)

# CREW Manual

##m##s (ex: 2m4s - 1m40s - 15m30s)

##h##m (ex: 2m4s - 1m40s - 15m30s)

##h##m##s###ms (ex: 1m25s300ms - 3m1s250ms)

##### (ex: 100 - 4000 - 567000)

Only numerical characters and values 'm' 's' 'h' 'ms' are allowed. Spaces are not allowed, on the other hand, and the format must be coherent. If the time is not indicated (i.e. if there is only a numerical value), the datum is considered as milliseconds.

## Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
DRIVER ERROR	Unable to send the request message, possible problem with Serial board
PROTOCOL ERROR	Generic error of data receipt from the PLC
PROTOCOL TIMEOUT	Timeout error, there has been no response to a request for data
PROTOCOL OFFLINE	Device not connected because it does not respond to PING command
SOCKET ERROR	Error during creation of Ethernet socket, the device does not respond
PING FAIL	The device does not respond to a standard ethernet PING request
FORMAT DATA ERR	The value (or string) inserted into the field when writing is not consistent with the rules of format allowed
ERROR	Value reported in the event of mishandling driver

# CREW Manual

Wago - 750-8xx -

## Description of type of connection:

The communication protocol connects the PLC through CODESYS, customised according to Wago settings. It comes with the settings required to connect to the specific PLC.

## Connection parameters (PANEL)

No description available.

## Connection parameters (DEVICE)

The device's connection is set up through the correct configuration of the CODESYS connection parameters. It is possible to enable or disable a parameter through the relative "Download" box, where you can choose which parameters to keep active in communication.

## Memory areas

The PLC variables are imported (with specific ESA utility) from a specific file generated through the PLC programming software.

# CREW Manual

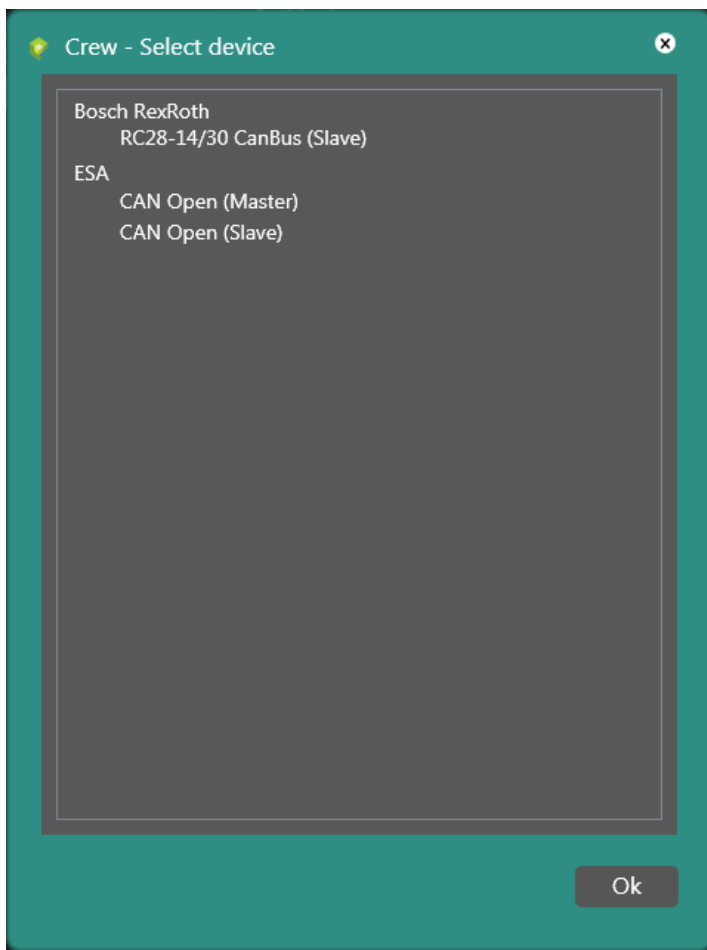
## Error codes

The driver can report, in the system variable, the following standard error codes:

Name	Code
INTERCHAR ERROR	Connection error with CoDeSys server
PROTOCOL OFFLINE	Error during creation of Ethernet socket, the device does not respond

## CREW Manual

# Drivers CAN Open



# CREW Manual

## ESA - CanOpen Master -

### General description

By standard, the CanOpen Master protocol supports up to 127 networked slave devices. A status area, an SDO-type parameter exchange area and four configurable PDO exchange areas are available for each device.

### Connection parameters (PANEL)

Baud Rate: CAN communication speed.

BootUP time (msec.): wait time for the panel before sending the “Enter Operational” broadcast command.

**Note:** If this value is 0 the command is not sent.

Sync time (msec.): cyclical time established for sending the SYNC command in CAN network, after which all PDO data configured as SYNC is sent in CAN network.

**Note:** if this value is 0 the command is disabled and the PDO type of data configured as SYNC is not sent.

Cyclic time (msec.): cyclical time established for sending all PDO data configured as CYCLIC in CAN network.

**Note:** If this value is 0 PDO type of data configured as CYCLIC is not sent.

Node Guarding (msec.): amount of time established to send the “Node status” command in CAN network, i.e.

Node Guarding system.

**Note:** If this value is 0, control of device state is disabled.

# CREW Manual

## Connection parameters (DEVICE)

Device address: the slave device address.

Process Data Object PDO1..4

Mode: Sync, the panel transmits the PDO every time the SYNC command is sent.

Cyclic, the panel transmits the PDO at every “Cyclic time” interval.

Event, the panel transmits the PDO with every datum variation.

Tx COB-ID: value of the PDO transmission “COB-ID” identifier (device side).

Rx COB-ID: value of the PDO reception “COB-ID” identifier (device side).

Service Data Object SDO

Tx COB-ID: value of the PDO transmission “COB-ID” identifier (device side).

Rx COB-ID: value of the PDO reception “COB-ID” identifier (device side).

Request timeout (msec.): the maximum amount of time established for the response to an SDO request.

Node Guarding

Mode: this establishes whether the Node Guarding system is enabled for the device.



# CREW Manual

Rx COB-ID: value of the PDO reception “COB-ID” identifier (device side).

Node life time (msec.): the maximum amount of time established for the response to a status request.

## Memory areas

Various memory areas are available for the CanOpen Master driver, some defined as status, others as commands, and others as actual memory areas.

Areas intended for command execution:

COMMAND	DESCRIPTION
START REMOTE NODE	Send to a remote node (X) the START command
START ALL NODES	As above, but Broadcast command (for all connected nodes)
STOP REMOTE NODE	Send to a remote node (X) the STOP command
STOP ALL NODES	As above, but Broadcast command (for all connected nodes)
ENTER PRE-OP NODE	Send to a remote node (X) the “Enter Pre-Operational” command
ENTER PRE-OP ALL	As above, but Broadcast command (for all connected nodes)
RESET NODE	Send to a remote node (X) a reset, and then the command “Enter Operational Mode”
RESET ALL NODES	As above, but Broadcast command (for all connected nodes)
RESET NODE COMM.	Send to a remote node (X) a reset command to the communication port
RESET ALL COMM.	As above, but Broadcast command (for all connected nodes)

# CREW Manual

Areas intended for device status

SINCE ESA CREW 1.7 VERSION	
STATUS	DESCRIPTION
<b>HMI</b>	Provide the panel status in the CAN network: 0 = Boot Up 4 = Stopped 5 = Operational 127 = Pre Operational
<b>HMI TX ERROR COUNTER</b>	Error counter during the messages transmission on CAN network
<b>HMI RX ERROR COUNTER</b>	Error counter during messages receiving on CAN network
<b>CanBus</b>	It provides the CAN bus status 0 = No Error 1 = Bus Warning 2 = Bus Off
<b>DEVICE (Node Guarding)</b>	It provides the status of the device (only if enabled "node guarding" function) : 4 = Stopped 5 = Operational 127 = Pre-Operational

# CREW Manual

Areas intended for data exchange

AREA	DESCRIPTION
SERVICE DATA OBJECT	Parameters data request of the device (functional in all operating CAN status)
PROCESS DATA OBJECT (FROM DEVICE)	Area dedicated to the synchronous / asynchronous cyclic data exchange from device (data are active only when the device is in OPERATIONAL mode)
PROCESS DATA OBJECT (TO DEVICE)	Area dedicated to the synchronous / asynchronous cyclic data exchange to device (data are active only when the device is in OPERATIONAL mode)

Error codes

The driver can report, in the system variable, the following standard error codes:

CODE	DESCRIPTION
SDO TIMEOUT	Communication error when reading an SDO parameter (response timeout).
SDO ERROR	Response error for an SDO parameter request (wrong message)
NODEGUARD ERROR	Device did not respond properly, in the project set time, to the node guarding request (device absent from the network).
RANGE ERROR	Request error PDO reading or writing (data not configured).
COMMAND ERROR	Requested command executing Error
RESET ERROR	Error during device reset
ENTER-OP ERROR	Device Enter Operational Mode command execution error

# CREW Manual

## ESA - CanOpen Slave -

### General description:

The Can Open Slave protocol is used when the operator panel needs to behave like a “Slave” device in a CAN network. There is a status area, a Bus error area, and sixteen configurable PDO exchange areas.

### Connection parameters (PANEL)

Baud Rate: CAN communication speed.

HMI address: operator panel address (slave).

Sync time (msec.): cyclical time established for sending the SYNC command in CAN network, after which all

PDO data configured as SYNC is sent in CAN network.

Note: If this value is 0 the command is disabled and the PDO type of data configured as SYNC **is not sent**)

Cyclic time (msec.): Cyclical time established for sending all PDO data configured as CYCLIC in CAN network.

Note: If this value is 0 PDO type of data configured as CYCLIC is not sent.

Heartbeat interval (msec.): Amount of time established for sending the operator panel “Node status” in CAN network (i.e. Heartbeat system).

**Note: If this value is 0, control of device state is disabled.**

### Process Data Object PDO1..16

Mode: Sync, the panel transmits the PDO every time the SYNC command is sent.

Cyclic, the panel transmits the PDO at every “Cyclic time” interval.

Event, the panel transmits the PDO with every datum variation.

# CREW Manual

Tx COB-ID: value of the PDO transmission “COB-ID” identifier (device side).

Rx COB-ID: value of the PDO reception “COB-ID” identifier (device side).

## Connection parameters (DEVICE)

None.

## Memory areas

Various memory areas are available for the CanOpen Slave driver, some defined as status, others as actual memory areas.

Areas intended for panel status:

STATUS	DESCRIPTION
<b>HMI</b>	Provide the panel status in the CAN network: 0 = BootUP 4 = Stopped 5 = Operational 127 = Pre-Operational
<b>HMI TX ERROR COUNTER</b>	Error counter during the messages transmission on CAN network
<b>HMI RX ERROR COUNTER</b>	Error counter during messages receiving on CAN network
<b>CanBus</b>	Provide the CAN network status: 0 = No Error 1 = Bus Warning 2 = Bus OFF

# CREW Manual

Areas intended for data exchange

AREA	DESCRIPTION
<b>PROCESS DATA OBJECT (FROM DEVICE)</b>	Area dedicated to the synchronous / asynchronous cyclic data exchange from device (data are active only when the device is in OPERATIONAL mode)
<b>PROCESS DATA OBJECT (TO DEVICE)</b>	Area dedicated to the synchronous / asynchronous cyclic data exchange to device (data are active only when the device is in OPERATIONAL mode)

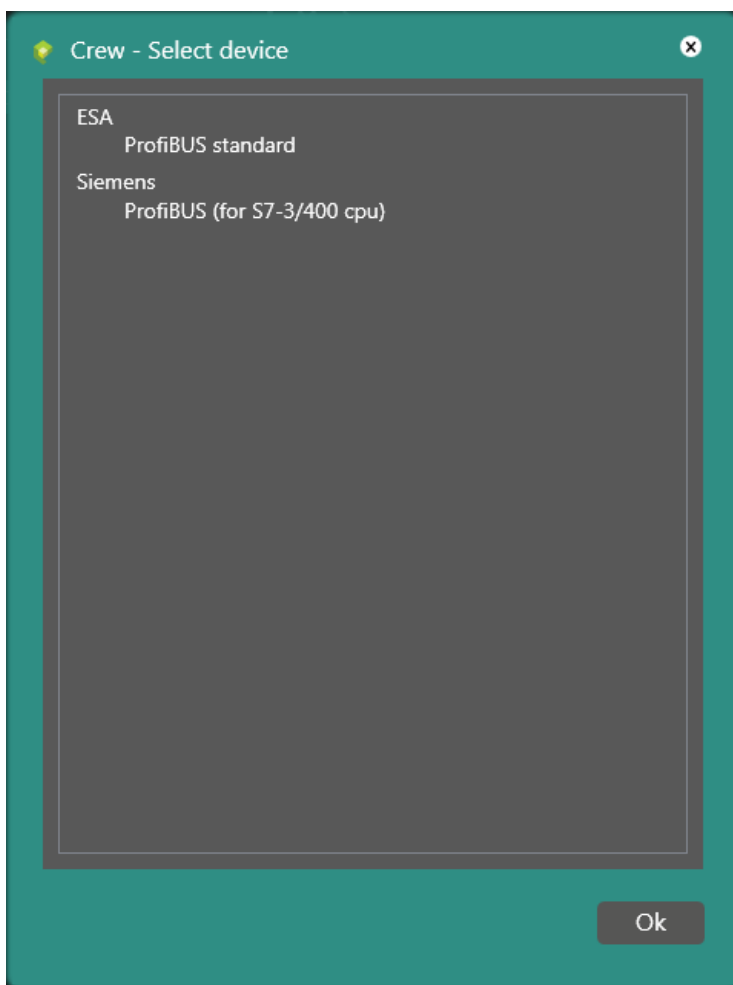
Error codes

The driver can report, in the system variable, the following standard error codes:

CODE	DESCRIPTION
<b>RANGE ERROR</b>	Request error PDO reading or writing (data not configured).
<b>WARNING ERROR</b>	There is a line error in the CAN line, below the 96 transmission / reception attempts
<b>BUSOFF ERROR</b>	There is a line error in the CAN line, blocking communication and exchange of data

## CREW Manual

# Drivers ProfiBUS



# CREW Manual

## ESA - ProfiBUS standard -

### General description

The ProfiBUS standard (slave) protocol is used to exchange two areas, respectively Input and Output with a size of 96 Word each, with only one device (Master).

### Connection parameters (PANEL)

Baud Rate: fixed 1.5 Mbps

HMI address: HMI address identified in the ProfiBUS network.

Exchange area: size of the areas of data (IN / OUT) exchanged with the Master ProfiBUS, in bytes.

Timeout (msec.): communication timeout with the Master device, that the error is reported within.

### Memory areas

AREA	DESCRIPTION
HMI data IN	Read-only area, input data to HMI panel
HMI data OUT	Reading and writing area, output data to Master device



# CREW Manual

## Error codes

The driver can report, in the system variable, the following standard error codes:

CODE	DESCRIPTION
ASIC FAULT	ASIC failed attempt initialization, driver not enabled.
ASIC IO ERROR	ASIC initializing error, driver not enabled.
COMM. ERROR	Timeout error when reading or writing data, possible inhibit communication (BUS halted).
ASIC ERROR	ASIC setup error during the open phase.
CONF. ERROR	Software configuration error (GSD file).
RANGE ERROR	Data required beyond the limit set in the panel configuration.

# CREW Manual

## SIEMENS - ProfiBUS -

### Connection parameters (PANEL)

The Siemens ProfiBUS (slave) protocol makes it possible to exchange all standard Siemens data with only one device (Master), through the use of a FB preventively loaded on the PLC (and configured for communication with the panel).

### Connection parameters (DEVICE)

Baud Rate: fixed 1.5 Mbps

HMI address: HMI address identified in the ProfiBUS network.

Exchange area: size of the areas of data (IN / OUT) exchanged with the Master ProfiBUS, in bytes.

Timeout (msec.): communication timeout with the Master device, that the error is reported within.

### Memory areas

The protocol provides access to the main areas of the memory in Word or Bit base format.

The COUNTER and TIMER data is now in binary format and it is no longer necessary to specify BCD because the driver does the conversion.

COUNTER data counts a fixed range of 0 ..999. TIMER (or SIMATIC TIME) data counts according to the following rules:

- 0ms .. 9s990ms (with time base = 0, namely 1/100s base.)

# CREW Manual

- 0ms .. 1m30s990ms (with time base = 1, namely 1/10s base.)
- 0s .. 16m39s (with time base = 2, namely 1s base.)
- 0s .. 2h46m30s (with time base = 2, namely 10s base.)

There are two areas that operate differently from the standard of a simple TAG, namely:

- Timer area
- Data Block area (Simatic Time)

The following type of data is contained in these areas:

1. TimeBase = 1/100s.
2. TimeBase = 1/10s.
3. TimeBase = 1s.
4. TimeBase = 10s.
5. TimeBase = AUTO (1ms.)
6. String Format

Representation / Setting of Timer, Data Block types (SimaticTime):

# CREW Manual

1. READ: fixed time base, LONG numerical format, representation 0..999000  
(x10ms.)

WRITE: fixed time base (x10ms), LONG numerical format, admissible values  
0..999

2. READ: fixed time base, LONG numerical format, representation 0..99900  
(x100ms.)

WRITE: fixed time base (x100ms), LONG numerical format, admissible values  
0..999

3. READ: fixed time base, LONG numerical format, representation 0..9990 (x1s.)

WRITE: fixed time base (x1s), LONG numerical format, admissible values 0..999

4. READ: fixed time base, LONG numerical format, representation 0..999 (x10s.)

WRITE: fixed time base (x10s), LONG numerical format, admissible values 0..999

5. READ: automatic time base (x1ms), LONG num. format, representation  
0..9990000

WRITE: automatic time base (x1ms), LONG num. format, admissible values  
0..9990000

Note: The driver automatically adapts the time base in write.

# CREW Manual

6. READ: automatic time base (x1ms), STRING format, repr. ##h##m##s###ms

WRITE: automatic time base (x1ms), STRING format, admissible values:

- ###ms (ex: 100ms - 450ms - 30ms)
- ##s###ms (ex: 4s100ms - 6s450ms - 15s30ms)
- ##m##s (ex: 2m4s - 1m40s - 15m30s)
- ##h##m (ex: 2m4s - 1m40s - 15m30s)
- ##h##m##s###ms (ex: 1m25s300ms - 3m1s250ms)
- ##### (ex: 100 - 4000 - 567000)

Only numerical characters and values 'm' 's' 'h' 'ms' are allowed. Spaces are not allowed, on the other hand, and the format must be coherent. If the time is not indicated (i.e. if there is only a numerical value), the datum is considered as **milliseconds**.

# CREW Manual

## Error codes

The driver can report, in the system variable, the following standard error codes:

CODE	DESCRIPTION
DRIVER ERROR	Unable to send the request message, possible problem with Ethernet board
PROTOCOL ERROR	Generic error of data receipt from the PLC
PROTOCOL TIMEOUT	Timeout error, there has been no response to a request for data
PROTOCOL OFFLINE	Device is offline, there is no response from the device when the ethernet connection
SOCKET ERROR	Error during creation of Ethernet socket, the device does not respond
PING FAIL	The device does not respond to a standard ethernet PING request
FORMAT DATA ERR	The value (or string) inserted into the field when writing is not consistent with the rules of format allowed
ERROR	Value reported in the event of mishandling driver

## CREW Manual

# Esaware: the products in the line



# CREW Manual





# CREW Manual

## HMI



EW100 is the new generation HMI based on a modern and powerful architecture that connects the view to application supervision and control.

# CREW Manual



Esaware offers three possible HMI variants:

- EW100AA, our standard HMI.
- EW100AB, our HMI powered by SoftPlc CODESYS and Ethercat.
- EW100AC, our HMI with SoftPlc CODESYS and with our I/O EW600.

# CREW Manual



The Esaware HMIs use the Windows Embedded Compact 7 operating system, the only solution offering transparent interconnectivity with any company system combined with the renowned reliability of embedded operating systems. Thanks to the wide screen, the display area is 40% larger than the traditional 4:3 screen. Plus, not only are LED widescreens long-lasting, they also offer considerable energy savings, as it is possible to adjust the brightness of the screen up to 100%.



# CREW Manual

	EW104A A-B-C	EW107A A-B-C	EW112A A-B-C	EW115A A-B-C
Display Size	4,3"	7"	12,1"	15,6"
Display Technology	TFT	TFT	TFT	TFT
Display Colors	262k	16M		
Display Backlight	LED			
Display Brightness (cd/m <sup>2</sup> )	400	600	400	300
Display Resolution (pixel)	480x272	800x480	1280x800	1366x768
Backlight life(hours)	50k			
Processor	ARM Cortex A8			
RAM	256 MB DDR3		512 MB DDR3	
Flash	3GB			
NVRAM	32K (SoftPLC)			
Serial Ports	SP1 RS232/485-MPI ; SP2 RS232/485-MPI ; CAN ; Profibus			
Ethernet	1x 10/100Mb (Ethercat only B/C version)	2x 10/100Mb (Ethercat only B/C version)		
USB 2.0	1 x USB Host +1 x USB Device (V. 2.0)		2 x USB Host + 1 x USB Device (V. 2.0)	
Slot	1 x SDHC/MMC			
Power Supply (Vdc)	12-32 (A version) / 1832 (B-C versions)			
Consumption (W)	4	7	15	19
Operating Temperature (C°)	- 10 ... + 50			
Storage Temperature (C°)	-20 ... + 65			
Humidity	<90%(non condensing)			
External dimensions (mm)	167x113	203x143	341x239	437x286
Cut-out dimensions (mm)	158,5 x 104,5	195,0 x 135,0	326,0 x 227,0	422,5 x 271,5
Weight (kg)	0,6	1,6	2,8	6
Certifications	CE / EN60068-2-6 / EN60068-2-27 / Humidity EN60068-2-30 / Atex Group II-cat.3 G/D / Gost / IP66			

CREW Manual

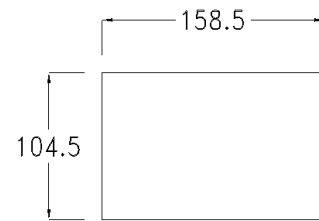
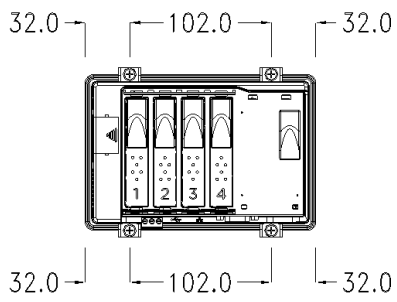
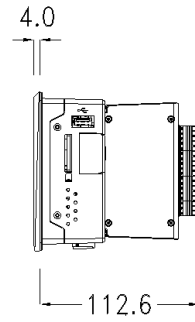
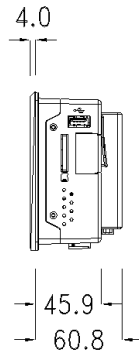
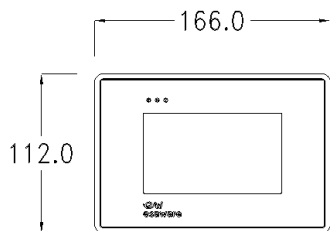
EW104XXXXX



EW MAN-MACHINE INTERFACE

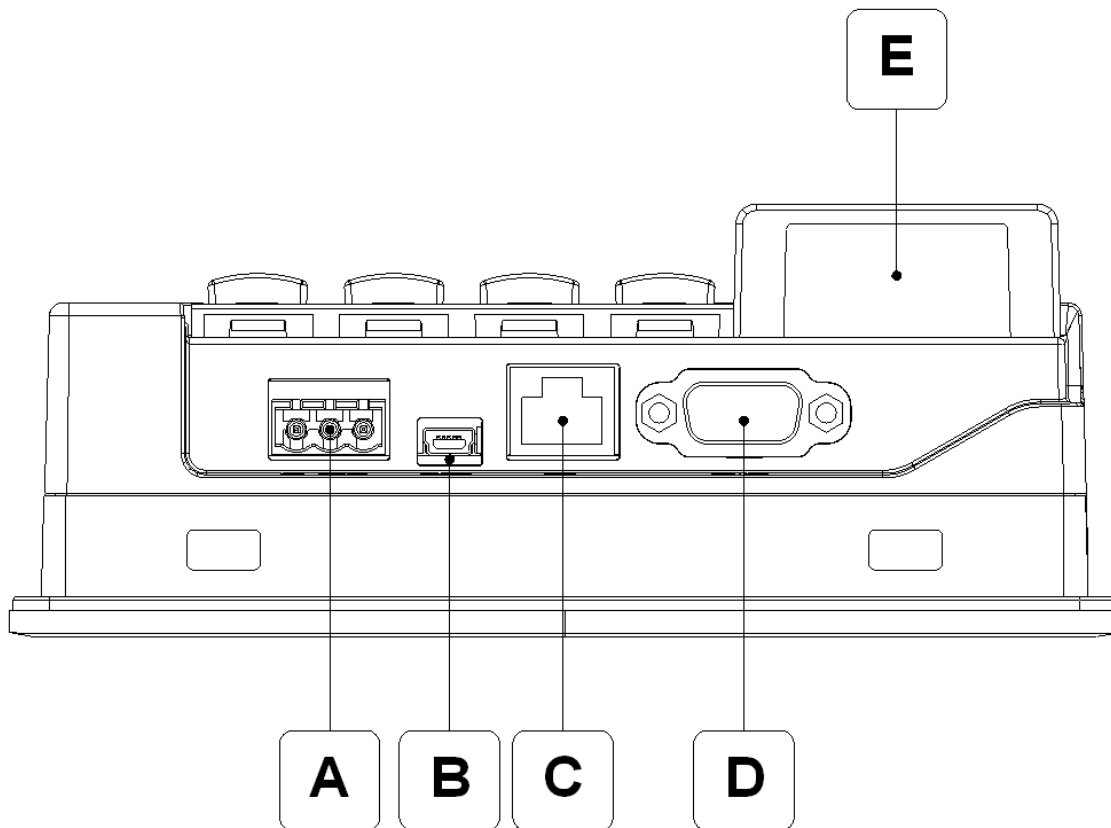
# CREW Manual

## Dimensions - Drilling



# CREW Manual

## Rear



**A: Power supply**

**B: Mini USB-B**

Serial port for project transfer

**C: ETH10/100**

Ethernet 10/100 Mbit - Rj45

**D: Serial port for communication with other devices**

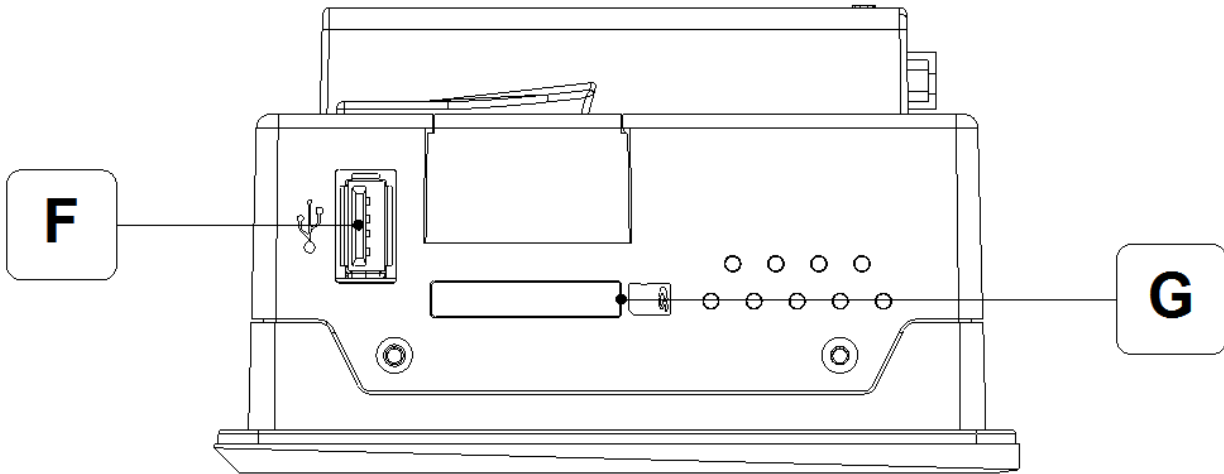
**E: RS232/485 (Optional)**

Serial port for communication with other devices

**E: CAN (Optional) CAN Serial port**

**E: PROFIBUS-DP (Optional) Serial port for communication in network**

# CREW Manual



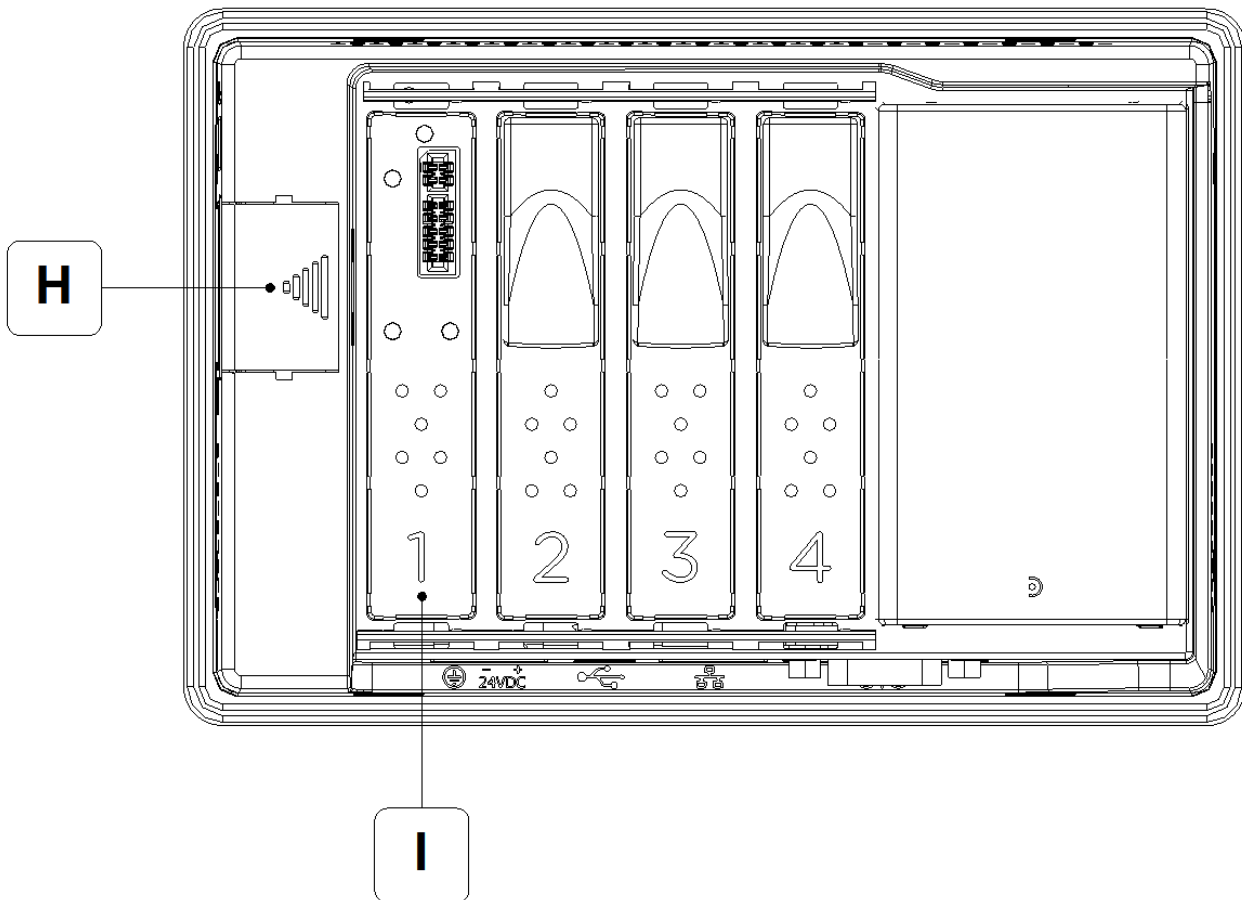
F: USB-A (USB port)

G: SD (Push-Push System)

Press in/press out



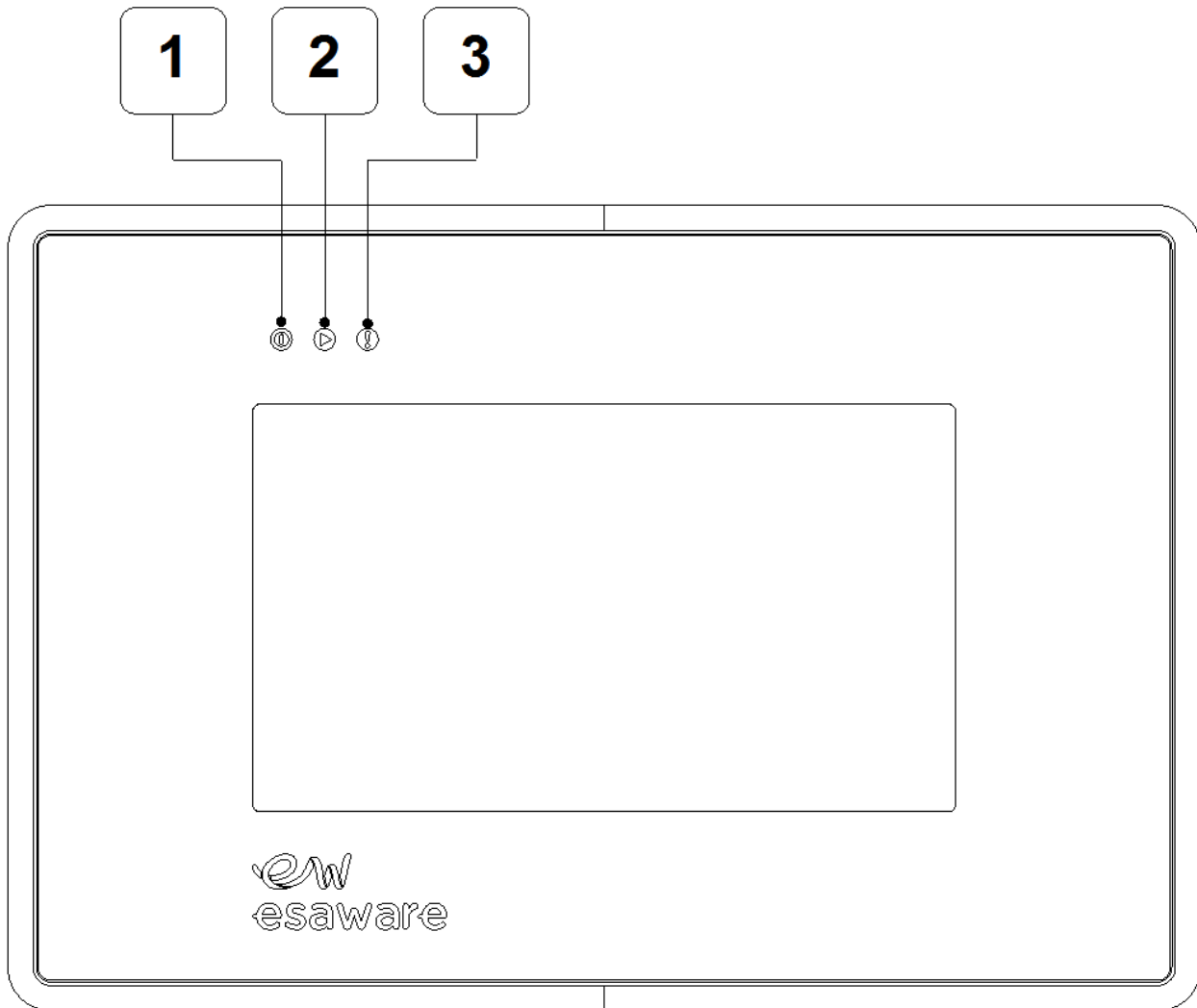
# CREW Manual



- H: Battery door
- I: Slot I/O (where expected)  
Input/Output module

# CREW Manual

Front



1: Power Green - Powered

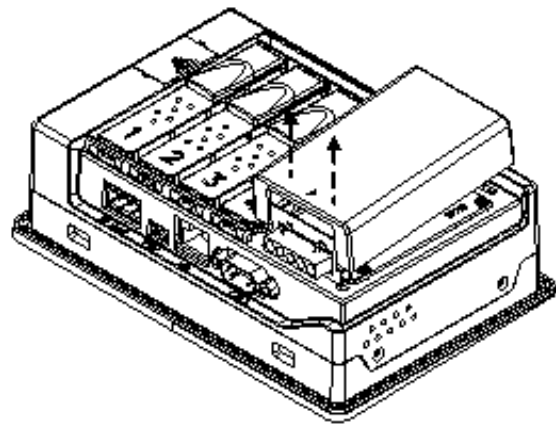
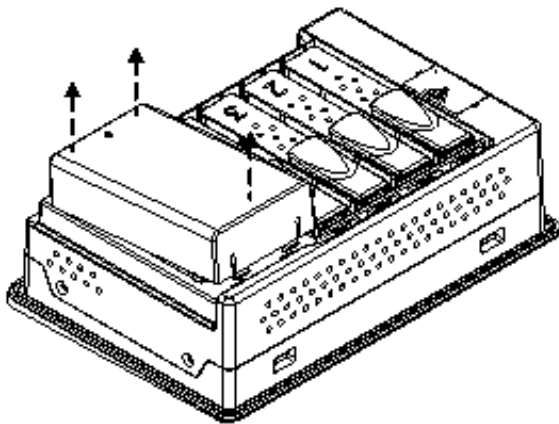
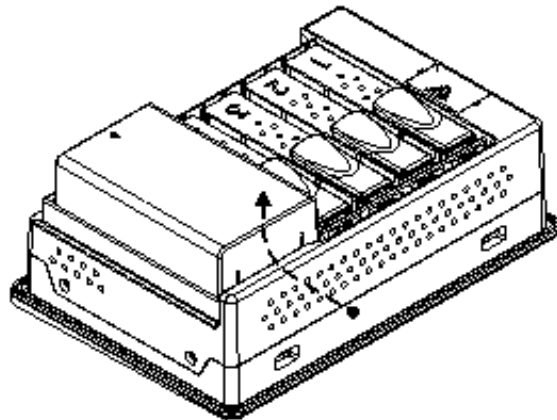
2: SoftPLC (where expected) Yellow - Stop / Green - Run

3: SoftPLC (where expected) Red - Error

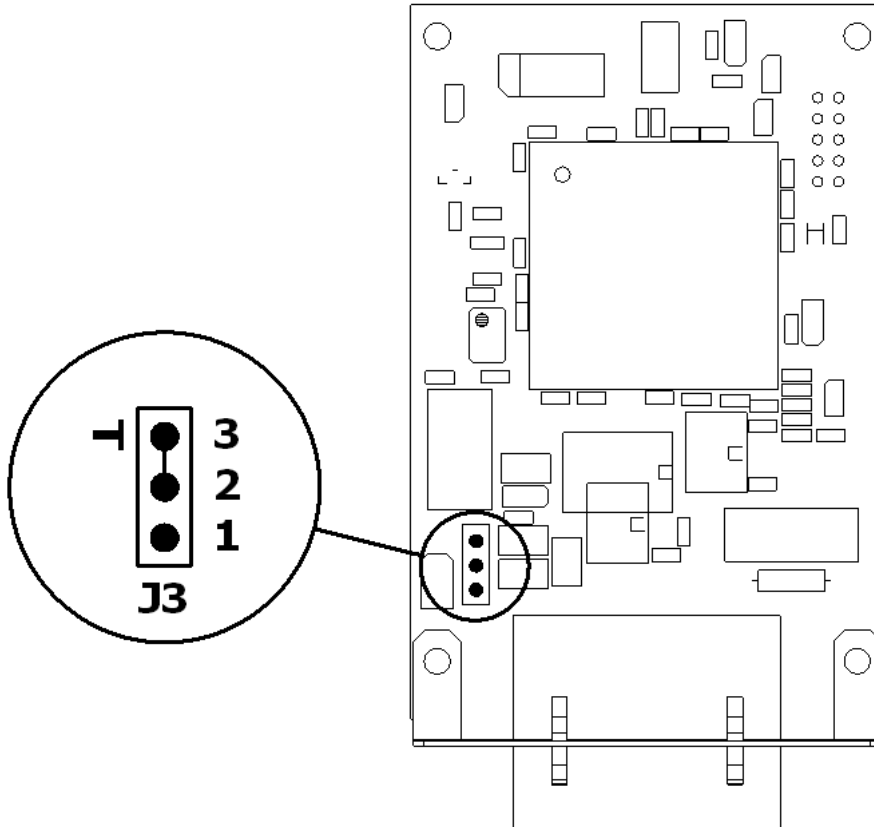
# CREW Manual

## CAN line termination

- 1: Switch off EW..
- 2: Take off the cover.



# CREW Manual



## Notes

J3 pin 1-2: Line open (default)

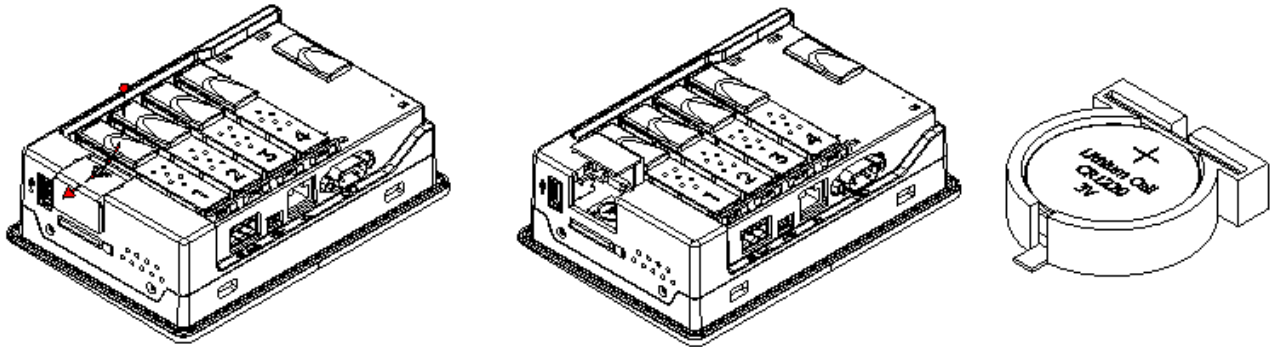
J3 pin 2-3 (T): Line terminated (120ohm)

3 Put the cover back on.

# CREW Manual

## Changing the battery

- 1: Switch off EW.
- 2: Open the battery door.



3. Changing the battery of the mother board.

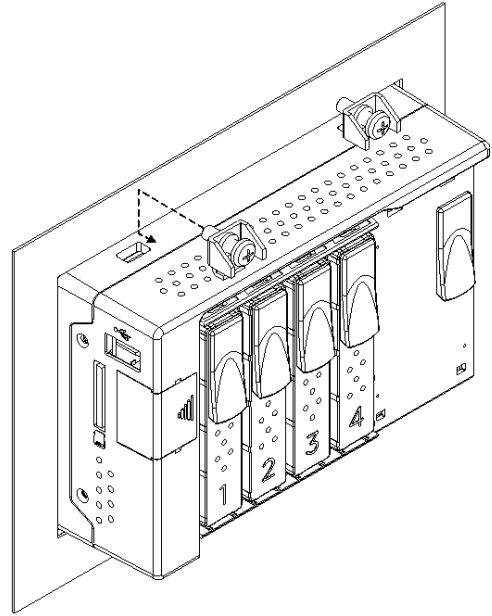
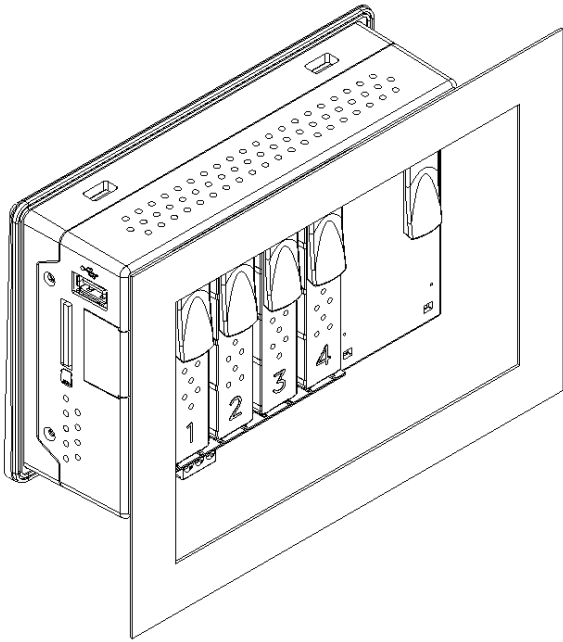


Attention: CR1220 3V Lithium battery- Put the new battery in observing type and polarity. Do not release the batteries into the environment.

4. Close the battery door.

# CREW Manual

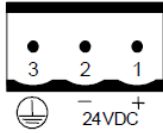
## EW Installation



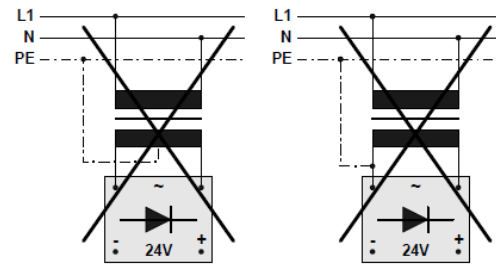
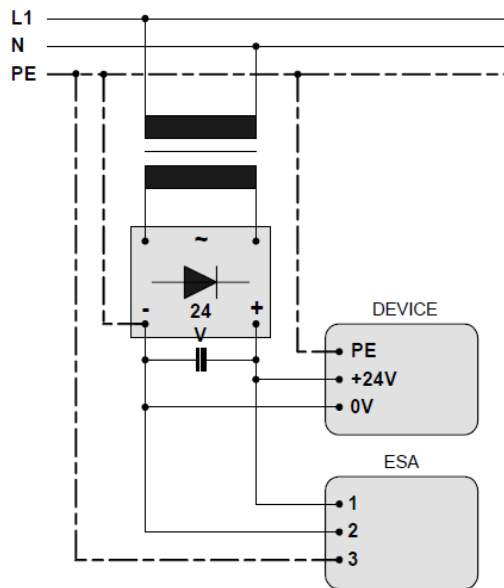
Tightening torque: 1.24 Nm (11 lbs. in.) +/- 5%

# CREW Manual

## Power supply



Power supply 3 pins connector		
1	+L 24 VDC	AWG12 - AWG30
2	M 0 V	
3	PE Protective ground	



**Warning:** These two configuration will seriously damage components.



### IMPORTANT:

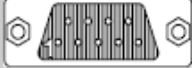
The mass of the devices connected to the serial and/or parallel communication ports must be absolutely at the same potential of the 0V power supply of the EW. The circulation of a current between 0V power supply and the mass of the communication ports could damage some components of the EW and the devices connected to it.

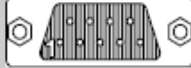
## Electromagnetic compatibility


Operating restrictions: the protection requirements are not guaranteed in residential areas.

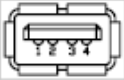
# CREW Manual


## Communication ports

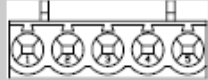
RS232/485	
	
Db 9 female connector	
1	Tx Rx485 +IN/OUT
2	RX1 IN
3	TX1 OUT
4	N.C.
5	Signal GND
6	Tx Rx485 -IN/OUT
7	RTS1 OUT
8	CTS1 IN
9	+ 5 VDC (reserved)

Profibus-DP	
	
Db 9 female connector	
1	Shield
2	N.C.
3	TXRX485+ Data B
4	Repeater-Control-signal RTS
5	Signal GND
6	P5V
7	N.C.
8	TXRX485- Data A
9	N.C.

ETH10/100	
	
RJ45 8 pin female connector	
1	TX+
2	TX-
3	RX+
4	N.C.
5	N.C.
6	RX-
7	N.C.
8	N.C.

USB-A	
	
4 pin male connector	
1	USBVDC (OUT)
2	USBD-
3	USBD+
4	Signal GND

Mini USB-B	
	
5 pin male connector	
1	USBVDC (IN)
2	USBD-
3	USBD+
4	N.C.
5	Signal GND

CAN	
	
5 pin female connector	
1	V -
2	CAN -
3	Shield
4	CAN +
5	N.C.

N.C. : Not connected.



Note

N.C.: Not connected.



CREW Manual

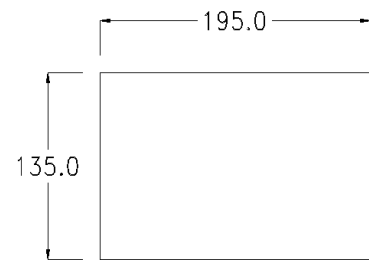
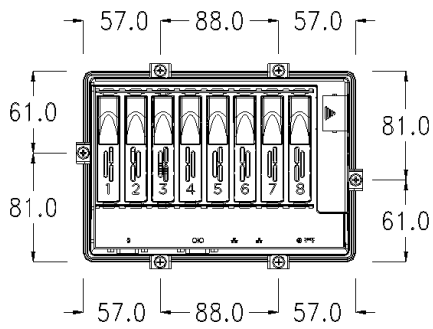
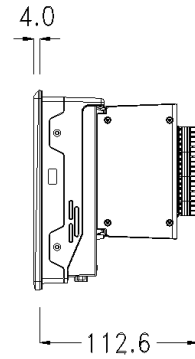
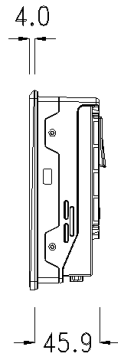
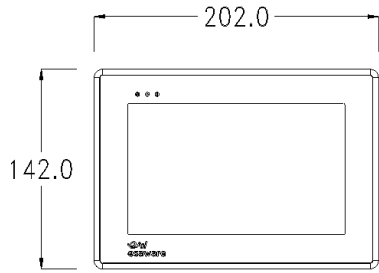
EW107xxxxx



EW MAN-MACHINE INTERFACE

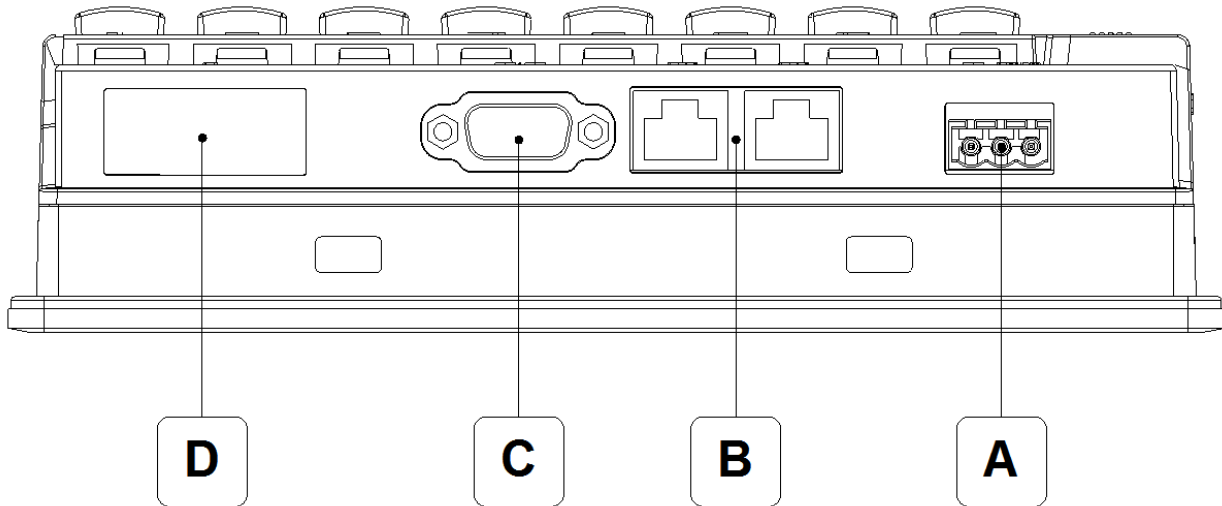
# CREW Manual

## Dimensions - Drilling



# CREW Manual

## Rear



A: Power supply

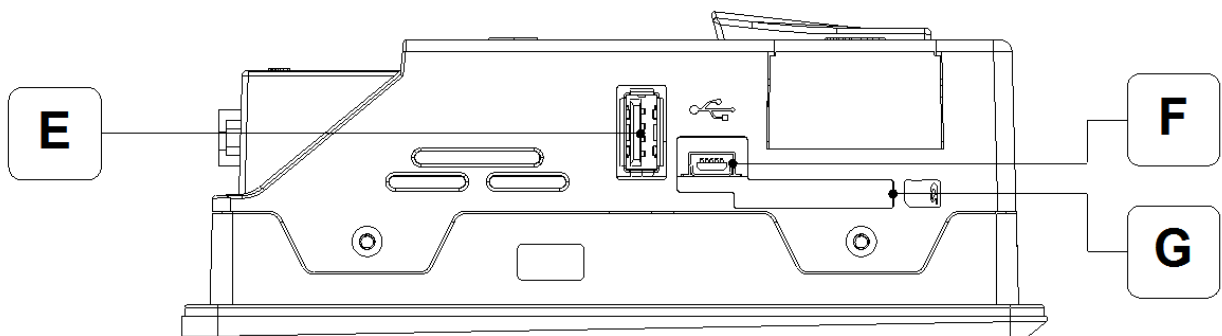
B: 2 x ETH10/100  
Ethernet 10/100 Mbit - Rj45

C: Serial port for communication with other devices

D: RS232/485 (Optional) Serial port for communication with other devices

D: CAN (Optional) CAN Serial port

D: PROFIBUS-DP (Optional) Serial port for communication in network

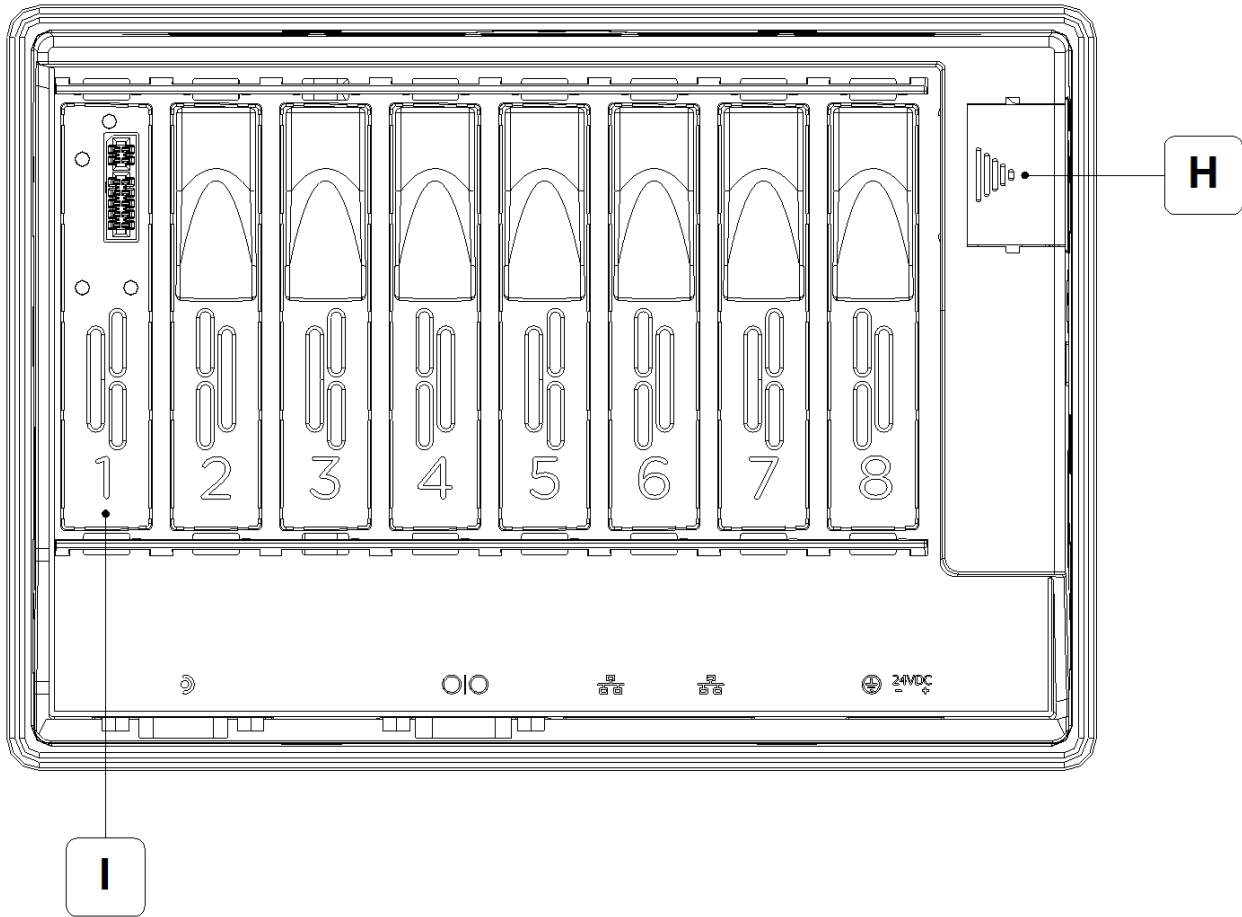


E: USB-A (USB port)

F: Mini USB-B Serial port for project transfer

G: SD (Push-Push System) Press in/Press out

# CREW Manual

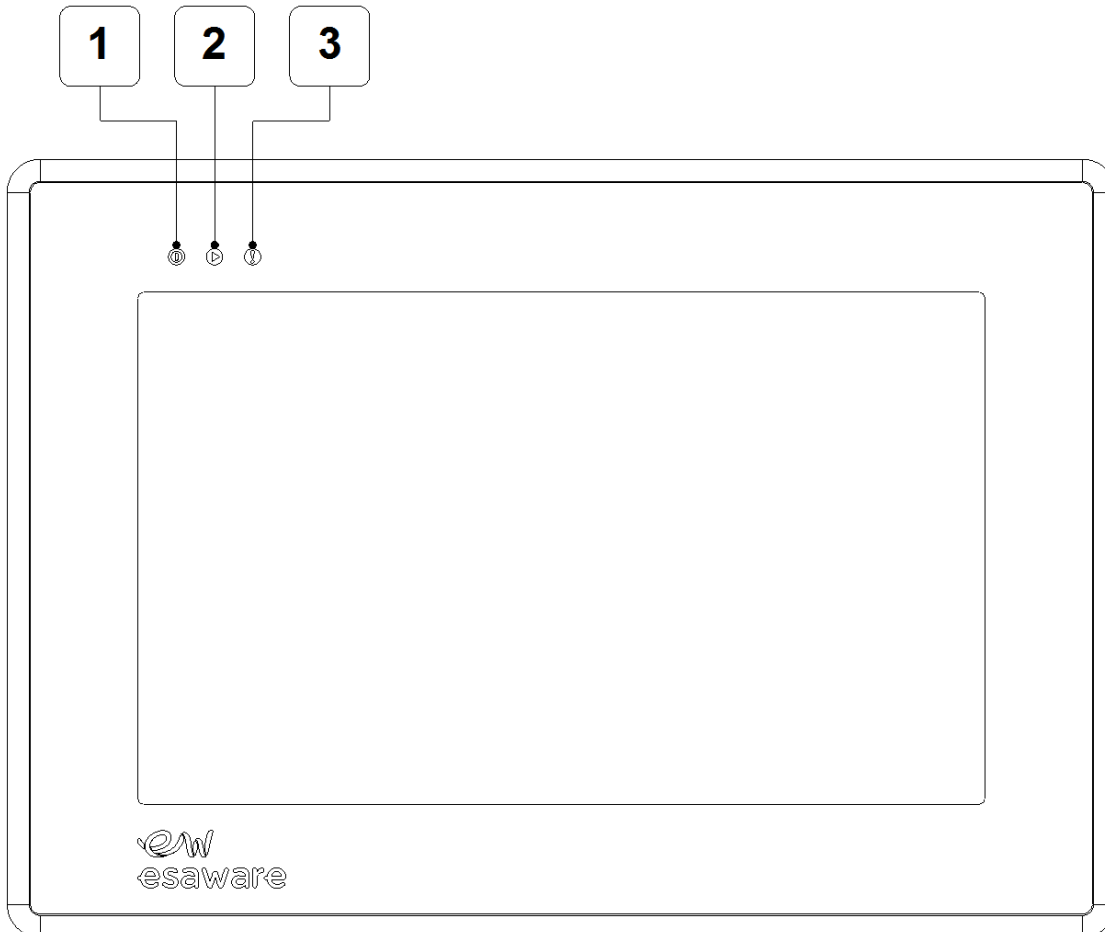


H: Battery door

I: Slot I/O (where expected) Input/Output module

# CREW Manual

Front



1: Power Green - Powered

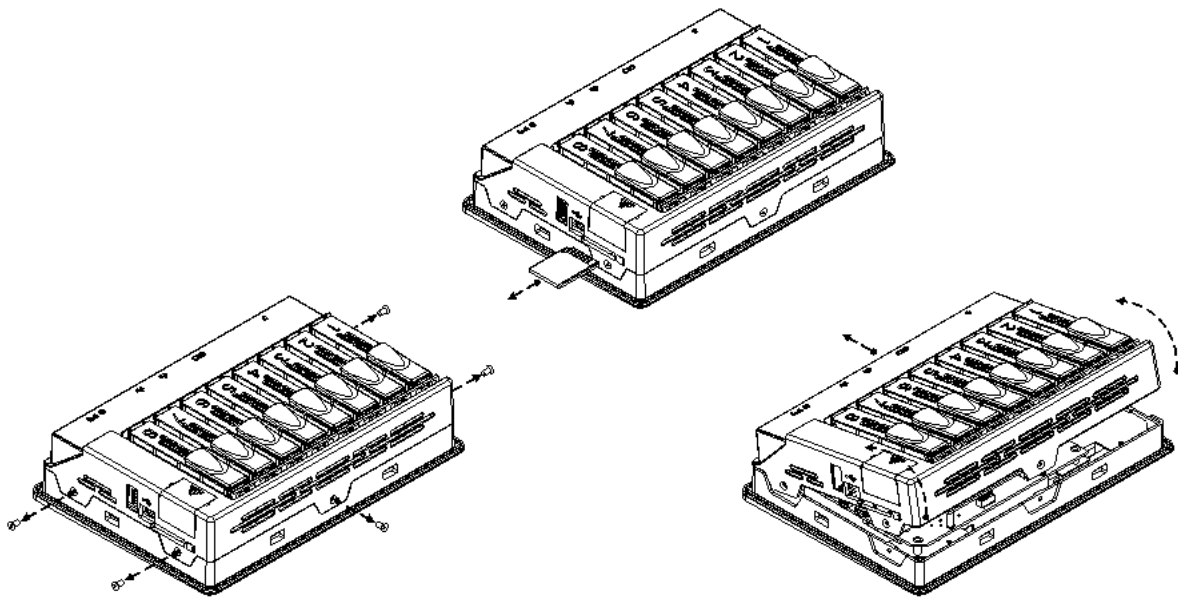
2: SoftPLC (where expected) Yellow - Stop / Green - Run

3: SoftPLC (where expected) Red - Error

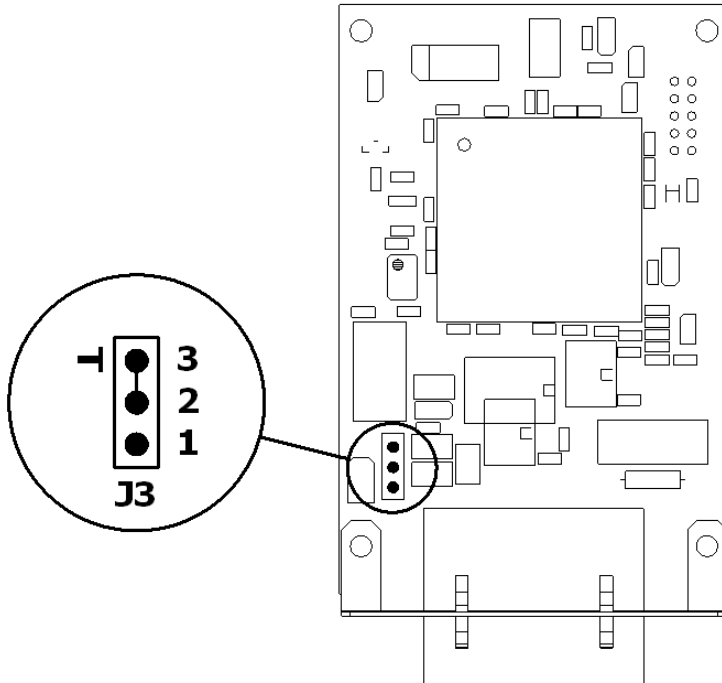
# CREW Manual

## CAN line termination

- 1: Switch off EW.
- 2: Remove the SD card (if any).



# CREW Manual



3: Take off the cover.



Note:

J3 pin 1-2: Line open (default)

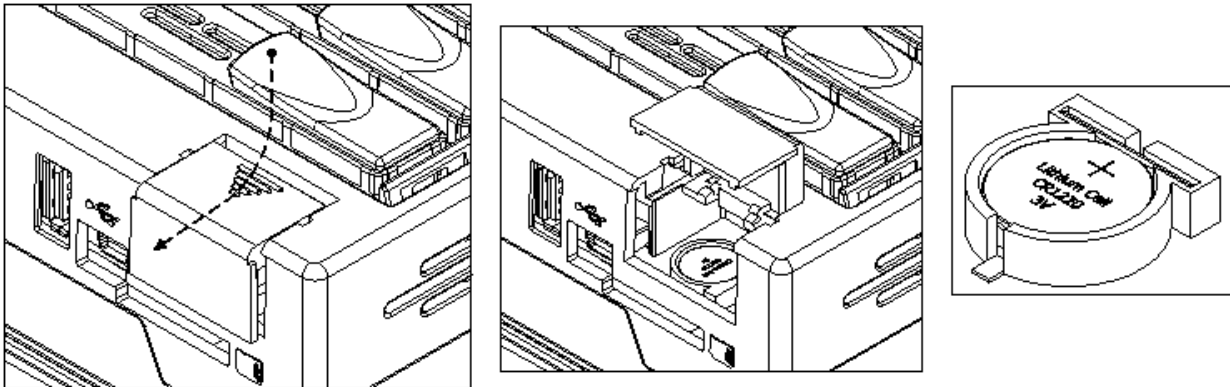
J3 pin 2-3 (T): Line terminated (120ohm)

4 Put the cover back on.

# CREW Manual

## Changing the battery

- 1: Switch off EW.
- 2: Open the battery door.



- 3: Changing the battery of the mother board.



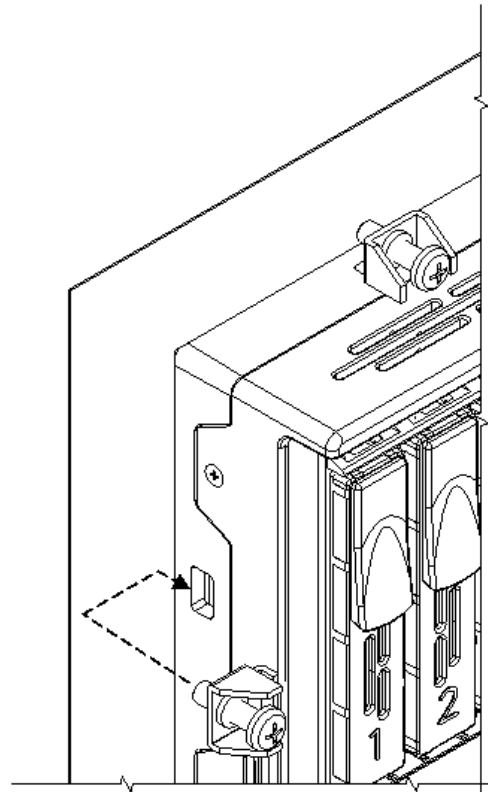
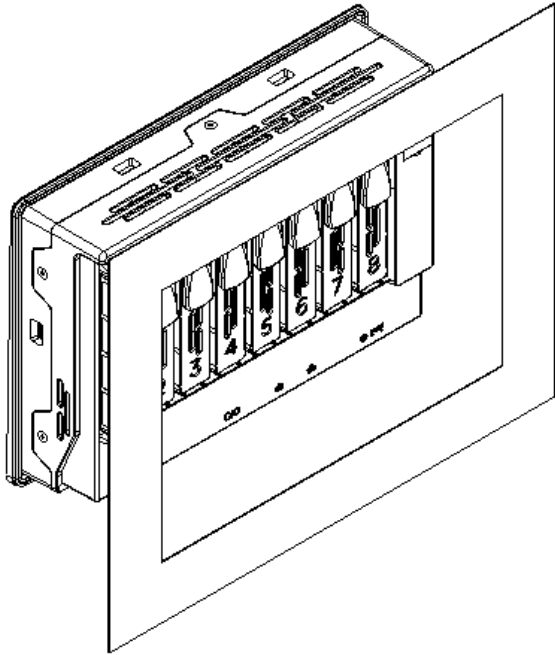
Attention: CR1220 3V Lithium battery- Put the new battery in observing type and polarity. Do not release the batteries into the environment.

4. Close the battery door.



# CREW Manual

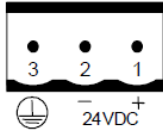
## EW Installation



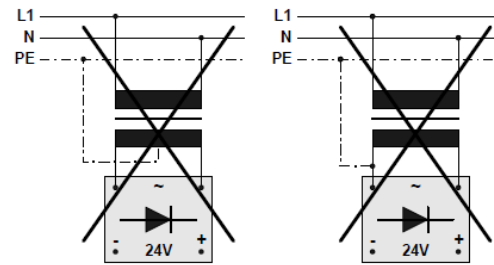
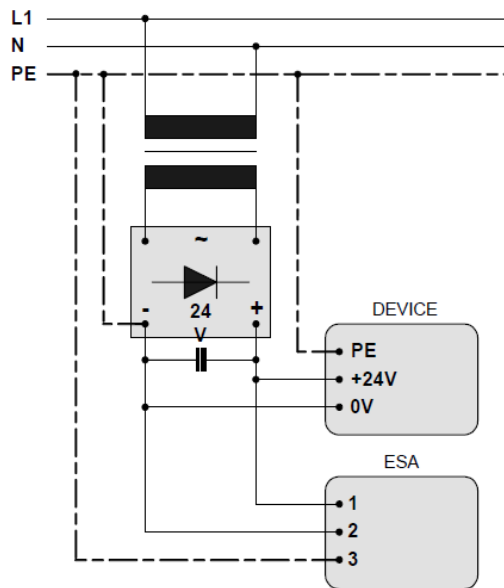
Tightening torque: 1.24 Nm (11 lbs. in.) +/- 5%

# CREW Manual

## Power supply



Power supply 3 pins connector		
1	+L 24 VDC	AWG12 - AWG30
2	M 0 V	
3	PE Protective ground	



**Warning:** These two configuration will seriously damage components.



### IMPORTANT:

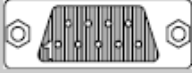
The mass of the devices connected to the serial and/or parallel communication ports must be absolutely at the same potential of the 0V power supply of the EW. The circulation of a current between 0V power supply and the mass of the communication ports could damage some components of the EW and the devices connected to it.

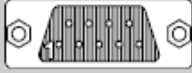
## Electromagnetic compatibility

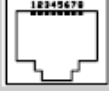
Operating restrictions: the protection requirements are not guaranteed in residential areas.

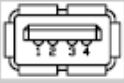
# CREW Manual


## Communication ports


RS232/485	
	
Db 9 female connector	
1	Tx Rx485 +IN/OUT
2	RX1 IN
3	TX1 OUT
4	N.C.
5	Signal GND
6	Tx Rx485 -IN/OUT
7	RTS1 OUT
8	CTS1 IN
9	+ 5 VDC (reserved)

Profibus-DP	
	
Db 9 female connector	
1	Shield
2	N.C.
3	TXRX485+ Data B
4	Repeater-Control-signal RTS
5	Signal GND
6	P5V
7	N.C.
8	TXRX485- Data A
9	N.C.

ETH10/100	
	
RJ45 8 pin female connector	
1	TX+
2	TX-
3	RX+
4	N.C.
5	N.C.
6	RX-
7	N.C.
8	N.C.

USB-A	
	
4 pin male connector	
1	USBVDC (OUT)
2	USB D-
3	USB D+
4	Signal GND

Mini USB-B	
	
5 pin male connector	
1	USBVDC (IN)
2	USB D-
3	USB D+
4	N.C.
5	Signal GND

CAN	
	
5 pin female connector	
1	V -
2	CAN -
3	Shield
4	CAN +
5	N.C.

N.C. : Not connected.



Note:

N.C.: Not connected.

CREW Manual

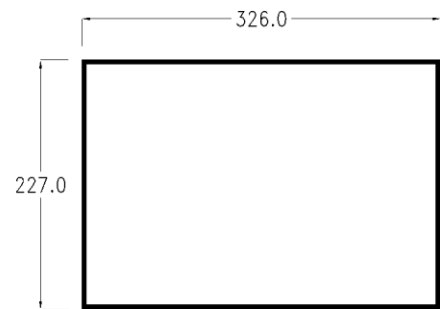
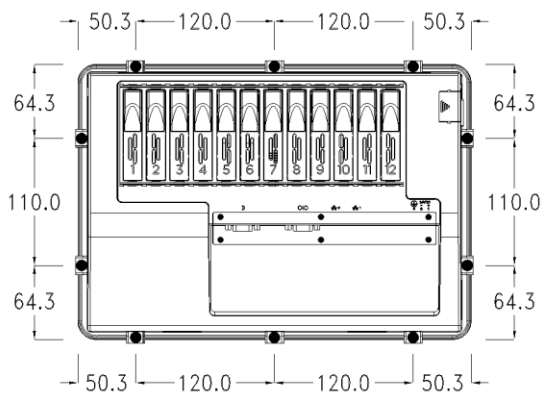
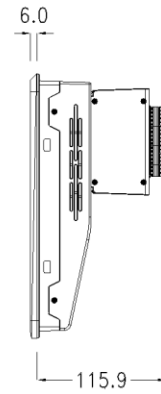
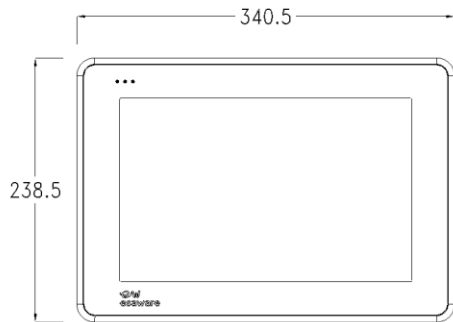
EW112xxxxx



EW MAN-MACHINE INTERFACE

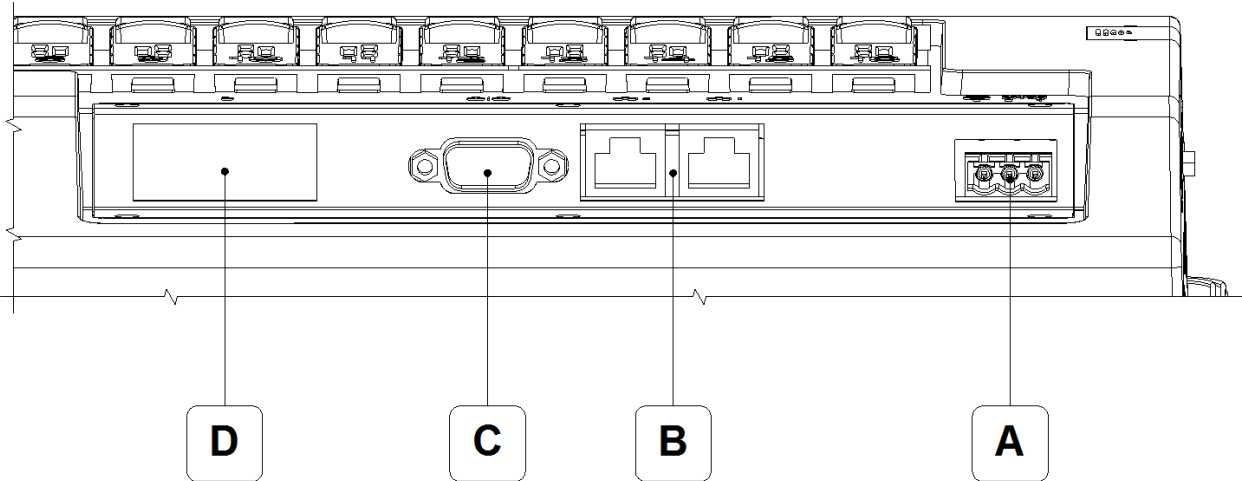
# CREW Manual

## Dimensions - Drilling



# CREW Manual

## Rear



A: Power supply

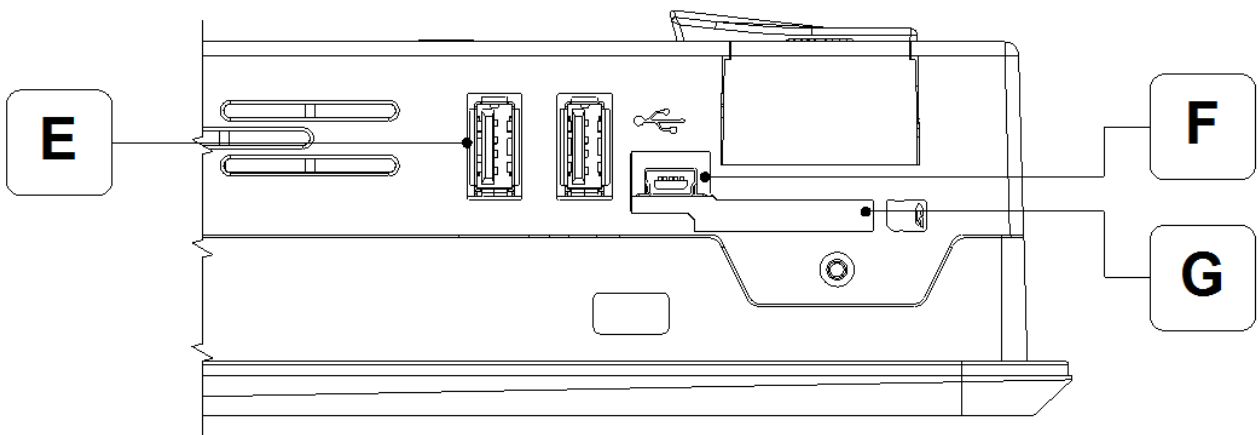
B: 2 x ETH10/100 Ethernet 10/100 Mbit - Rj45

C: Serial port for communication with other devices

D: RS232/485 (Optional) Serial port for communication with other devices

D: CAN (Optional) CAN Serial port

D: PROFIBUS-DP (Optional) Serial port for communication in network

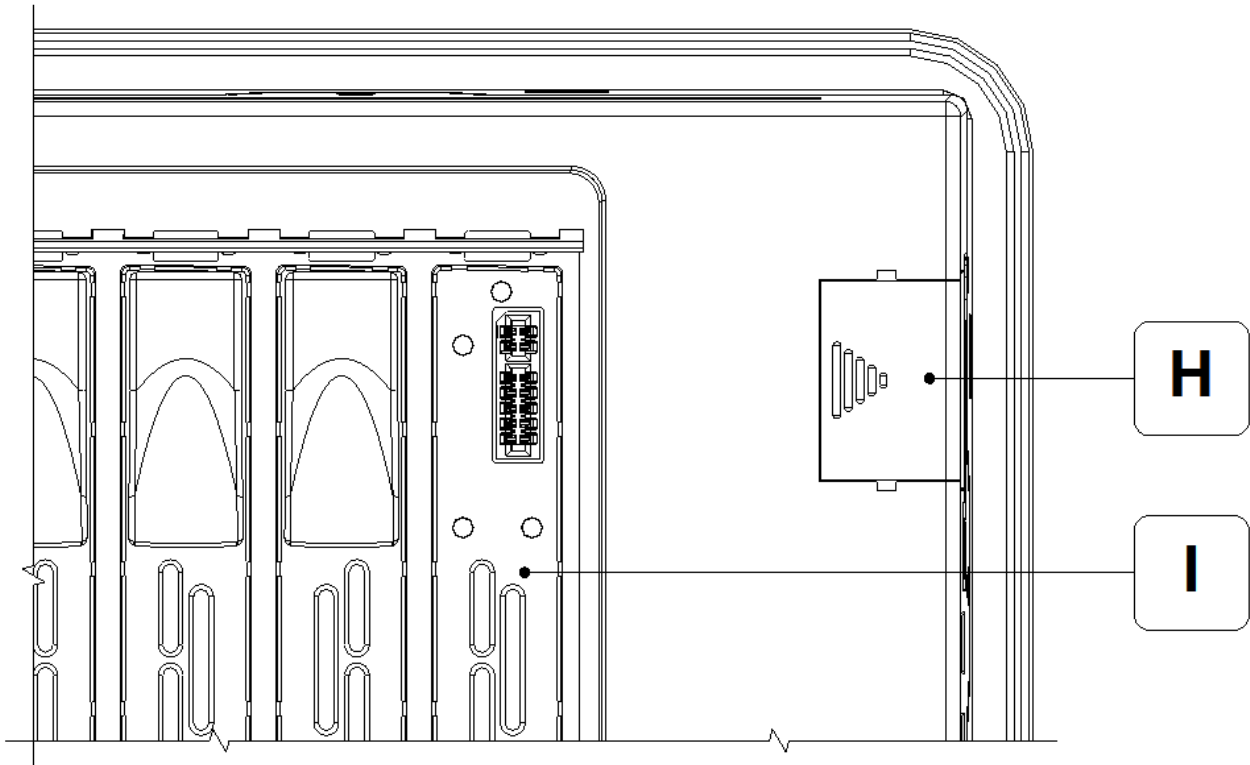


E: 2X USB-A (USB port)

F: Mini USB-B Serial port for project transfer

G: SD (Push-Push System) Press in/Press out

# CREW Manual

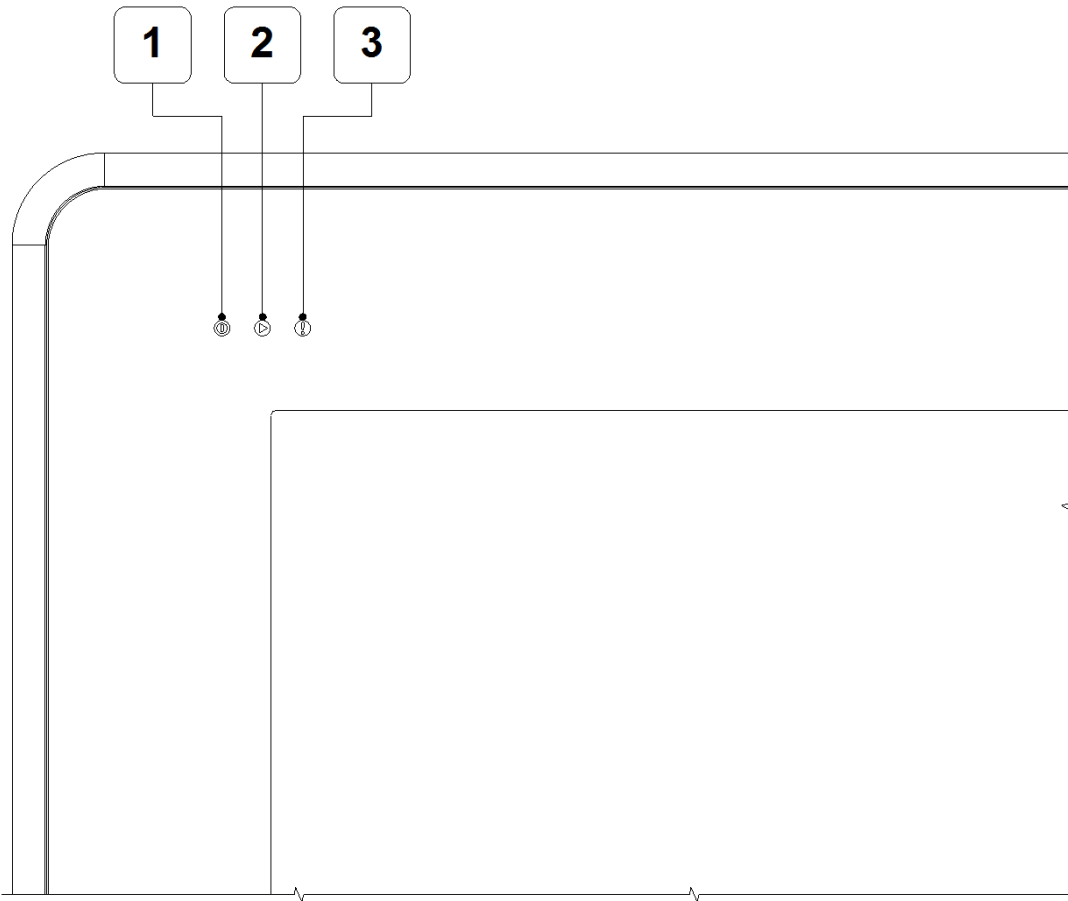


H: Battery door

I: Slot I/O (where expected) Input/Output module

# CREW Manual

Front



1: Power Green - Powered

2: SoftPLC (where expected) Yellow - Stop / Green - Run

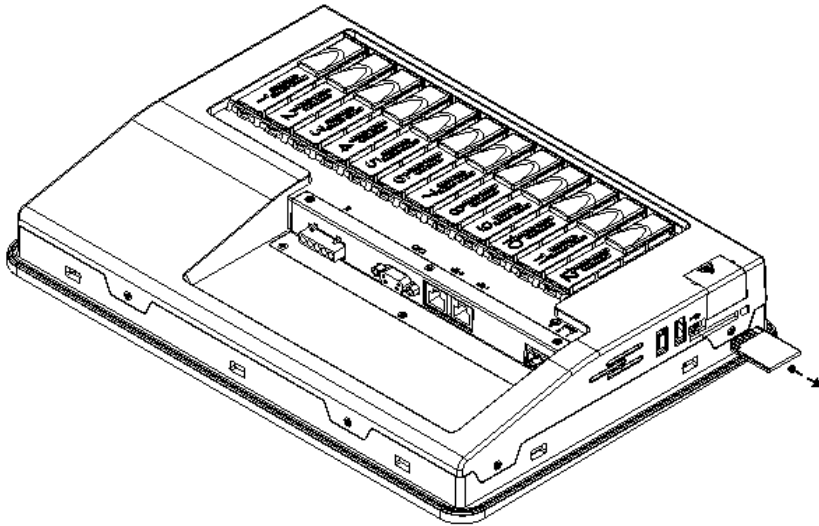
3: SoftPLC (where expected) Red - Error



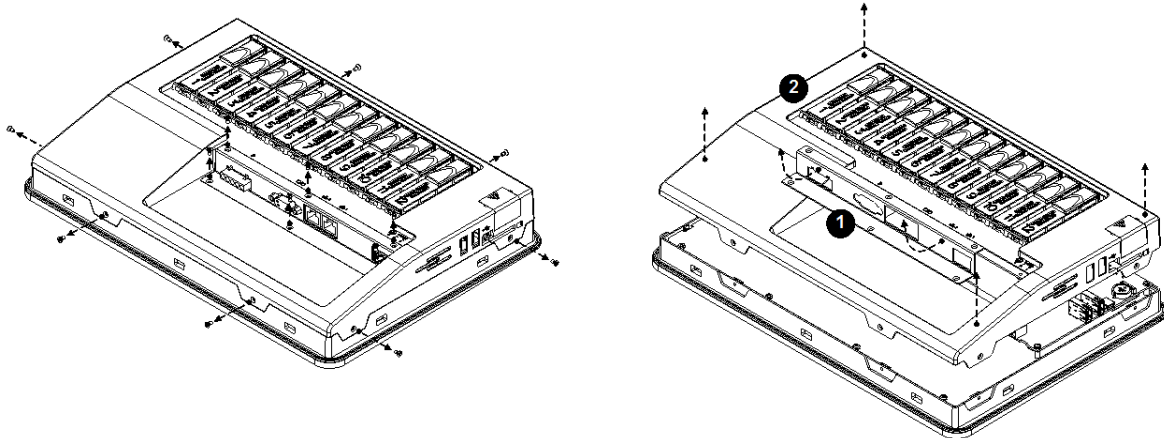
# CREW Manual

## CAN line termination

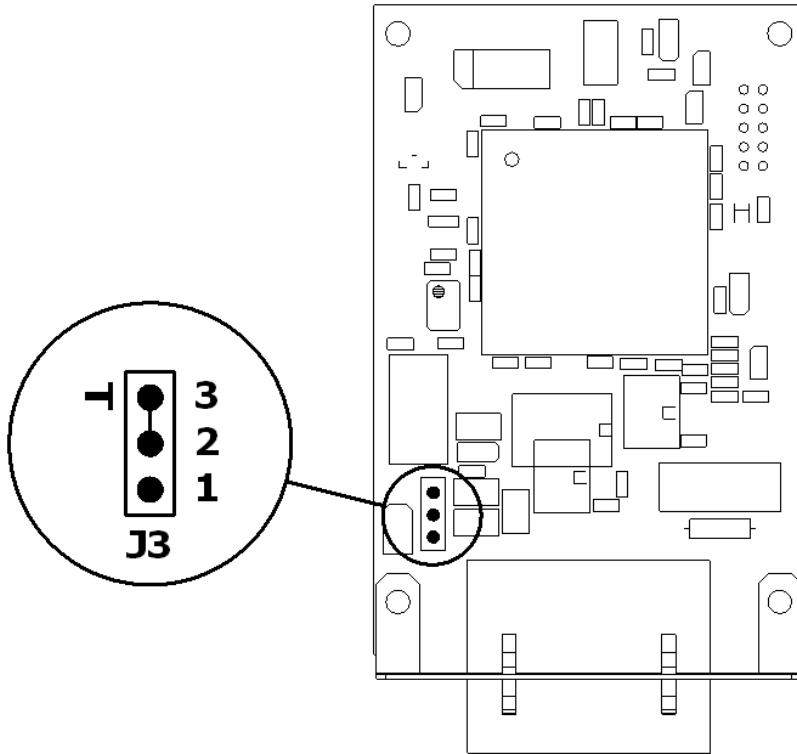
- 1: Switch off EW.
- 2: Remove the SD card (if any).



- 3: Take off the cover.



# CREW Manual



## Note:

J3 pin 1-2: Line open (default)

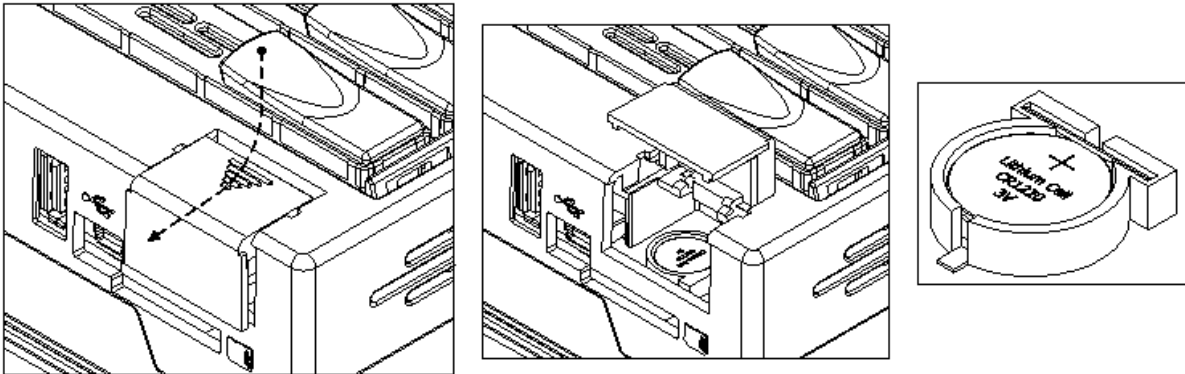
J3 pin 2-3 (T): Line terminated (120ohm)

4 Put the cover back on.

# CREW Manual

## Changing the battery

- 1: Switch off EW.
- 2: Open the battery door.



- 3: Changing the battery of the mother board.

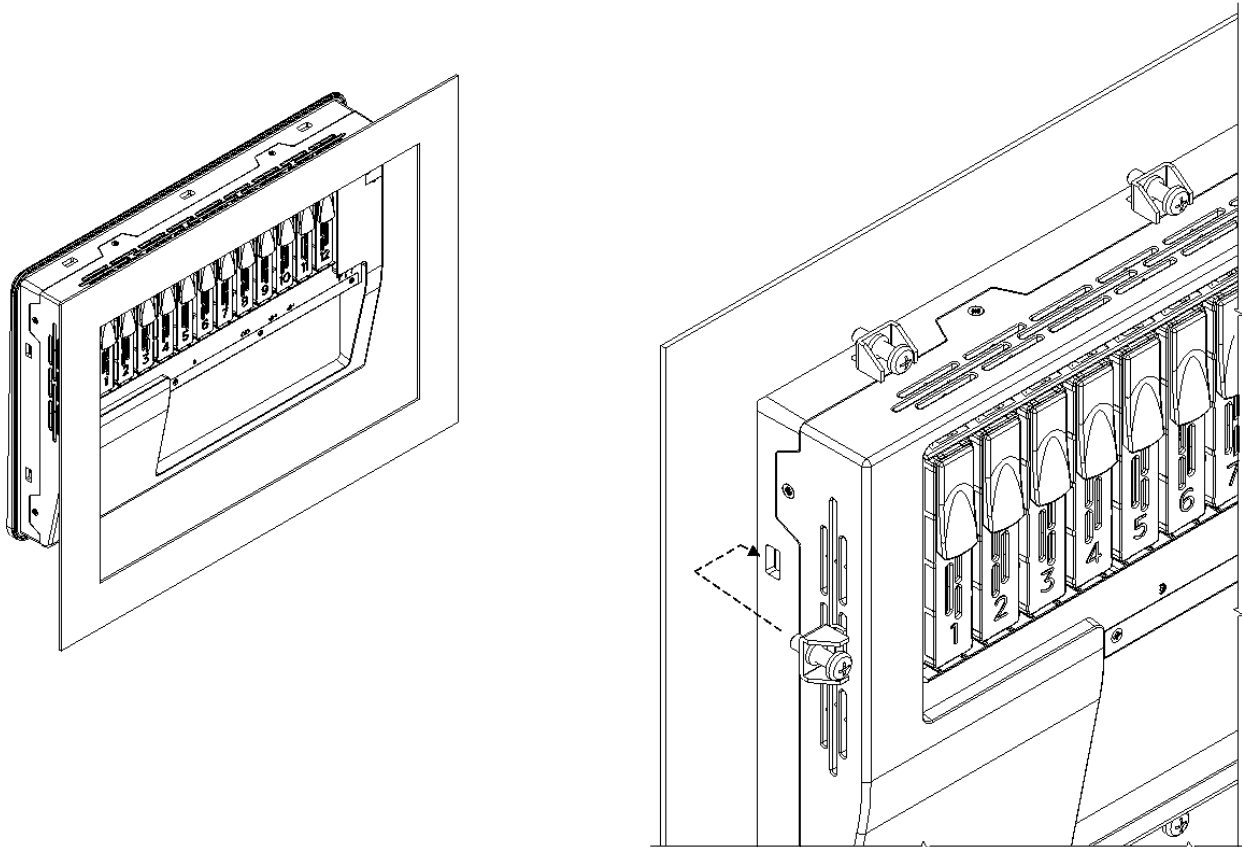


Attention: CR1220 3V Lithium battery- Put the new battery in observing type and polarity. Do not release the batteries into the environment.

- 4: Close the battery door.

# CREW Manual

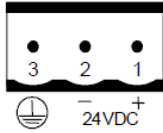
## EW Installation



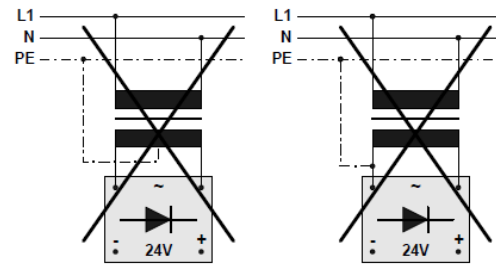
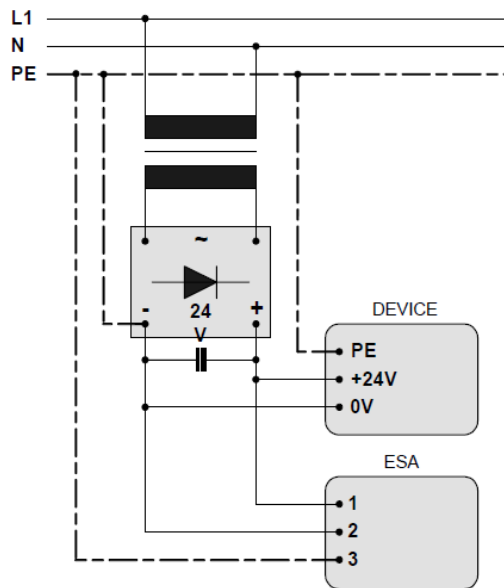
Tightening torque: 1.24 Nm (11 lbs. in.) +/- 5%

# CREW Manual

## Power supply



Power supply 3 pins connector		
1	+L 24 VDC	AWG12 - AWG30
2	M 0 V	
3	PE Protective ground	



**Warning:** These two configuration will seriously damage components.



### IMPORTANT :


The mass of the devices connected to the serial and/or parallel communication ports must be absolutely at the same potential of the 0V power supply of the EW. The circulation of a current between 0V power supply and the mass of the communication ports could damage some components of the EW and the devices connected to it.

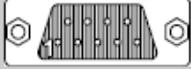
## Electromagnetic compatibility


Operating restrictions: the protection requirements are not guaranteed in residential areas.

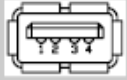
# CREW Manual


## Communication ports


RS232/485	
	
Db 9 female connector	
1	Tx Rx485 +IN/OUT
2	RX1 IN
3	TX1 OUT
4	N.C.
5	Signal GND
6	Tx Rx485 -IN/OUT
7	RTS1 OUT
8	CTS1 IN
9	+ 5 VDC (reserved)

Profibus-DP	
	
Db 9 female connector	
1	Shield
2	N.C.
3	TXRX485+ Data B
4	Repeater-Control-signal RTS
5	Signal GND
6	P5V
7	N.C.
8	TXRX485- Data A
9	N.C.

ETH10/100	
	
RJ45 8 pin female connector	
1	TX+
2	TX-
3	RX+
4	N.C.
5	N.C.
6	RX-
7	N.C.
8	N.C.

USB-A	
	
4 pin male connector	
1	USBVDC (OUT)
2	USB D-
3	USB D+
4	Signal GND

Mini USB-B	
	
5 pin male connector	
1	USBVDC (IN)
2	USB D-
3	USB D+
4	N.C.
5	Signal GND

CAN	
	
5 pin female connector	
1	V -
2	CAN -
3	Shield
4	CAN +
5	N.C.

N.C. : Not connected.



Note

N.C.: Not connected.

CREW Manual

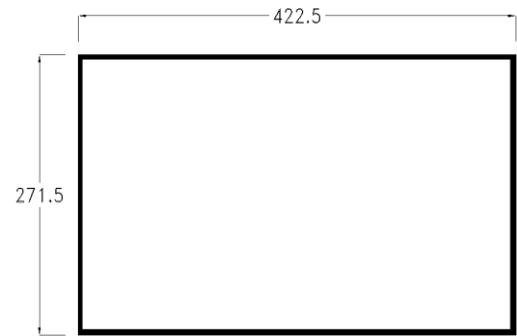
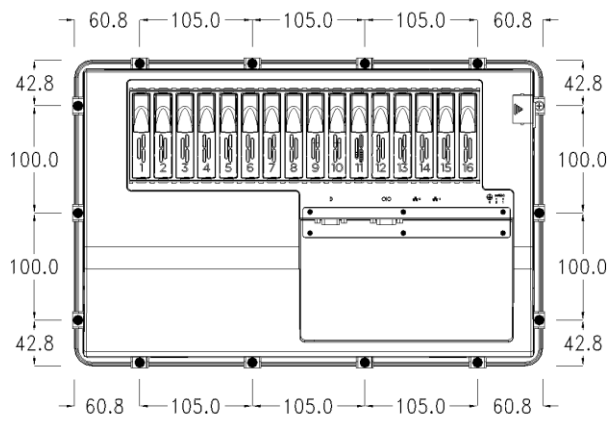
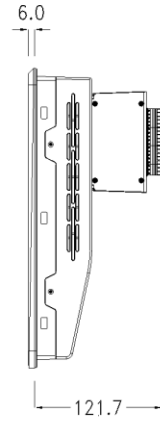
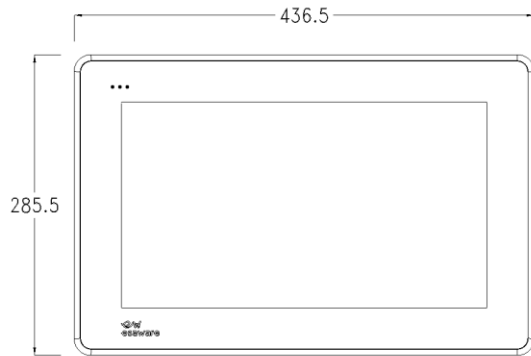
EW115xxxxx



EW MAN-MACHINE INTERFACE

# CREW Manual

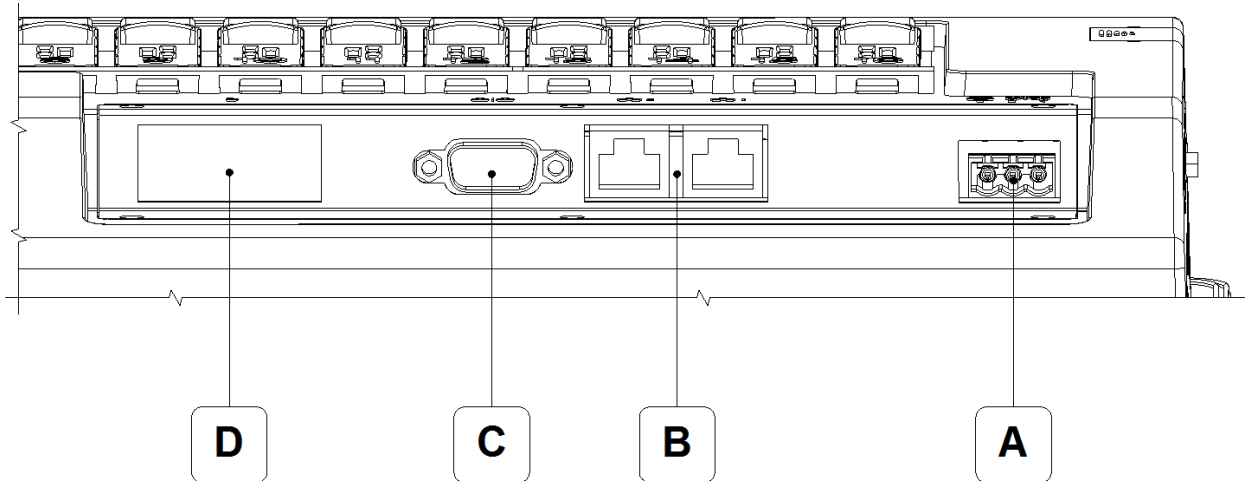
## Dimensions - Drilling





# CREW Manual

## Rear



A: Power supply

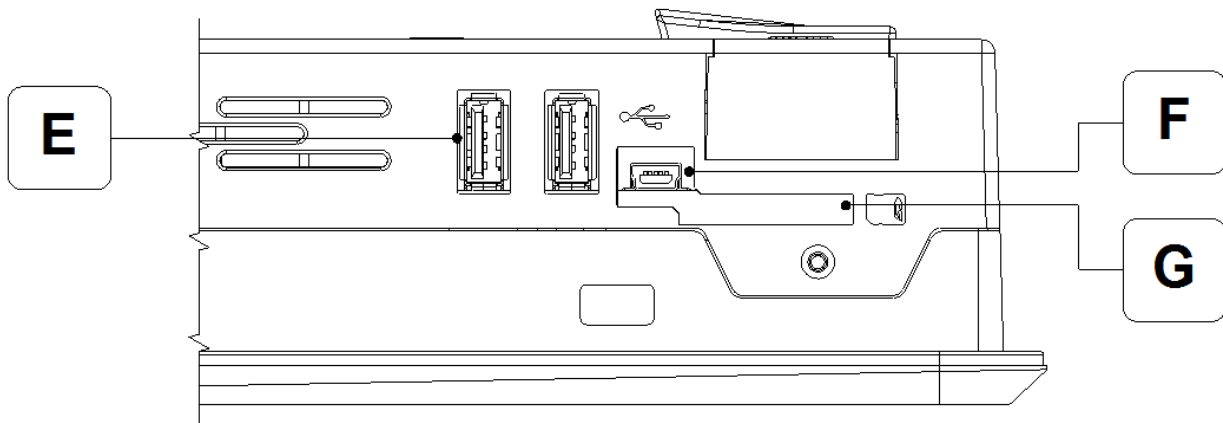
B: 2 x ETH10/100 Ethernet 10/100 Mbit - Rj45

C: Serial port for communication with other devices

D: RS232/485 (Optional) Serial port for communication with other devices

D: CAN (Optional) CAN Serial port

D: PROFIBUS-DP (Optional) Serial port for communication in network

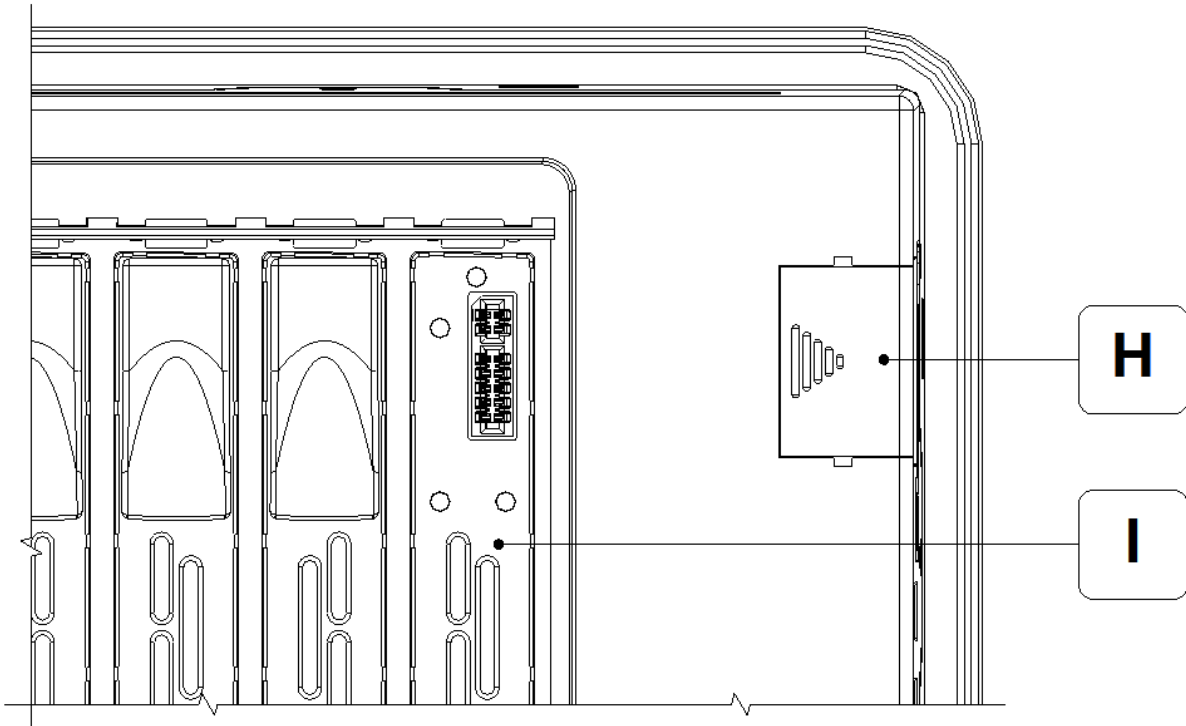


E: 2X USB-A (USB port)

F: Mini USB-B Serial port for project transfer

G: SD (Push-Push System) Press in/Press out

# CREW Manual

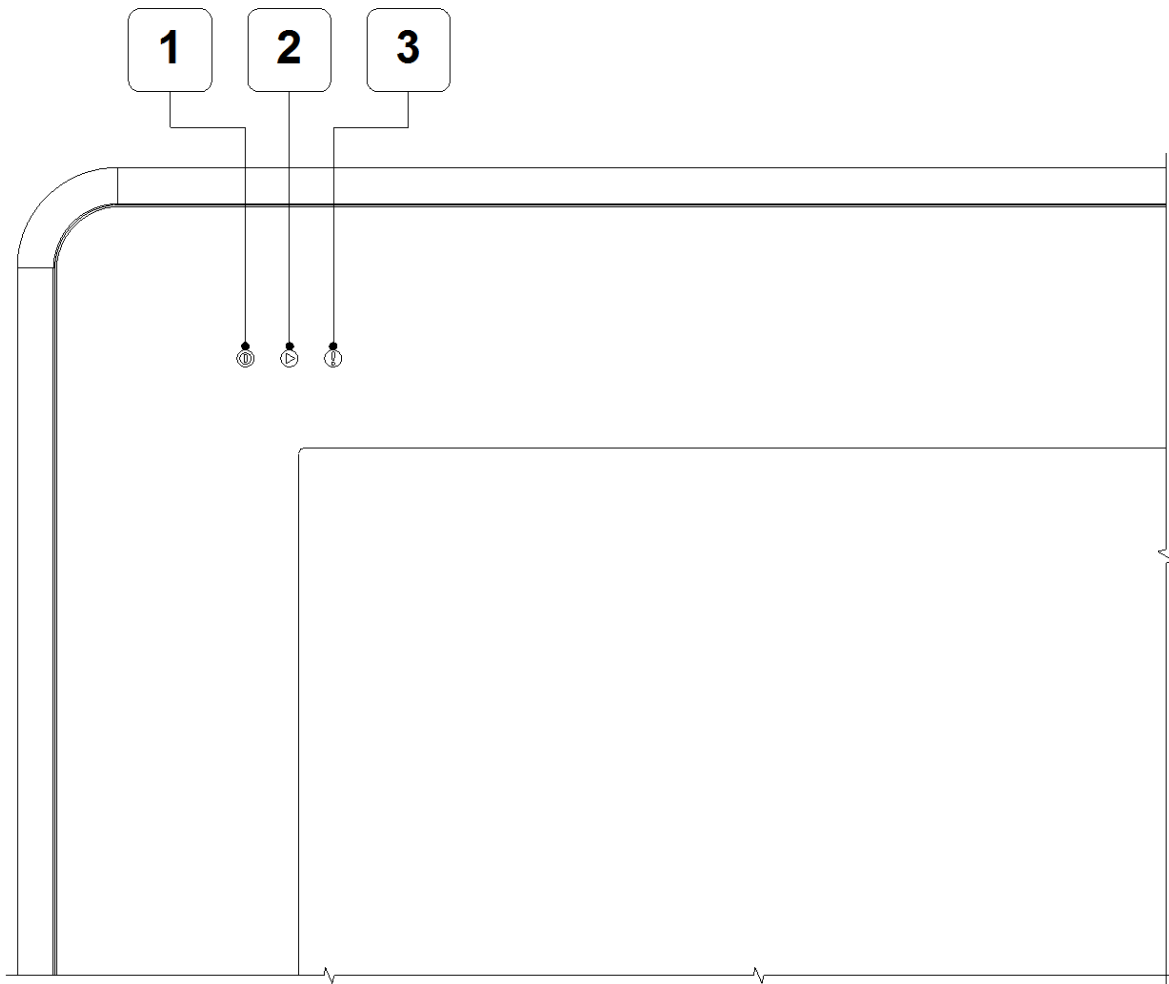


H: Battery door

I: Slot I/O (where expected) Input/Output module

# CREW Manual

Front



1: Power Green - Powered

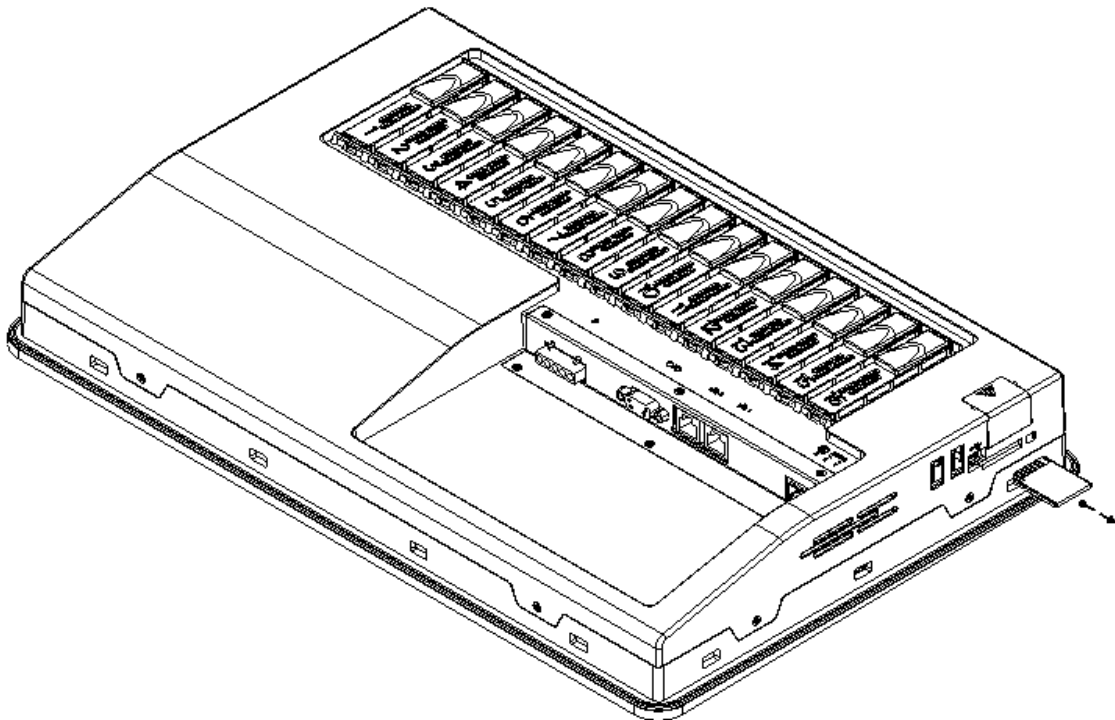
2: SoftPLC (where expected) Yellow - Stop / Green - Run

3: SoftPLC (where expected) Red - Error

# CREW Manual

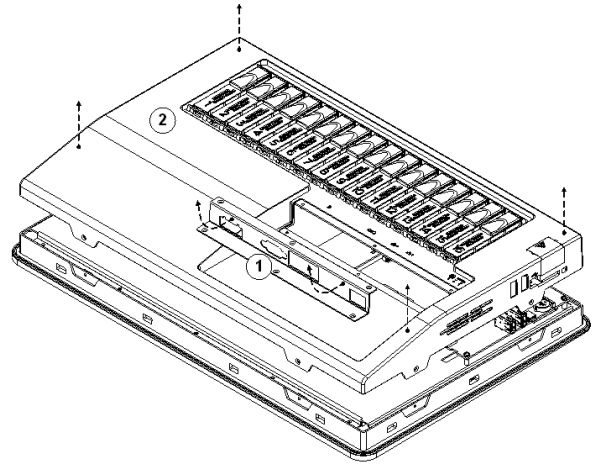
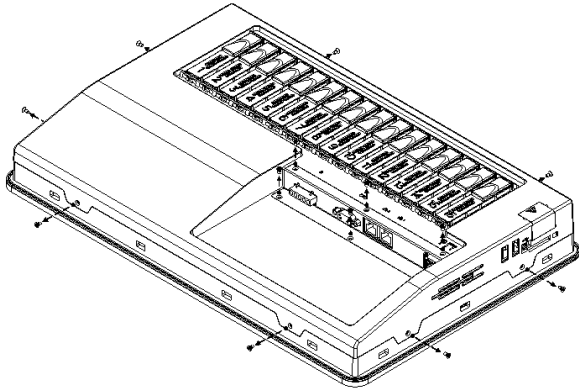
## CAN line termination

- 1: Switch off EW.
- 2: Remove the SD card (if any).

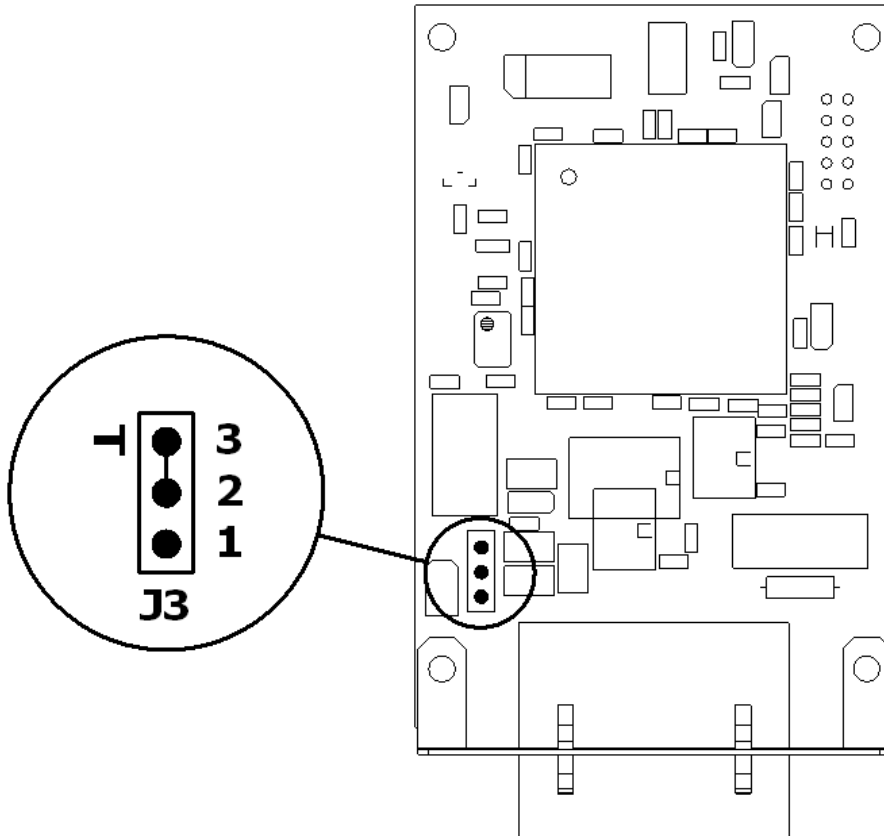


# CREW Manual

3: Take off the cover.



# CREW Manual



Note:

J3 pin 1-2: Line open (default)

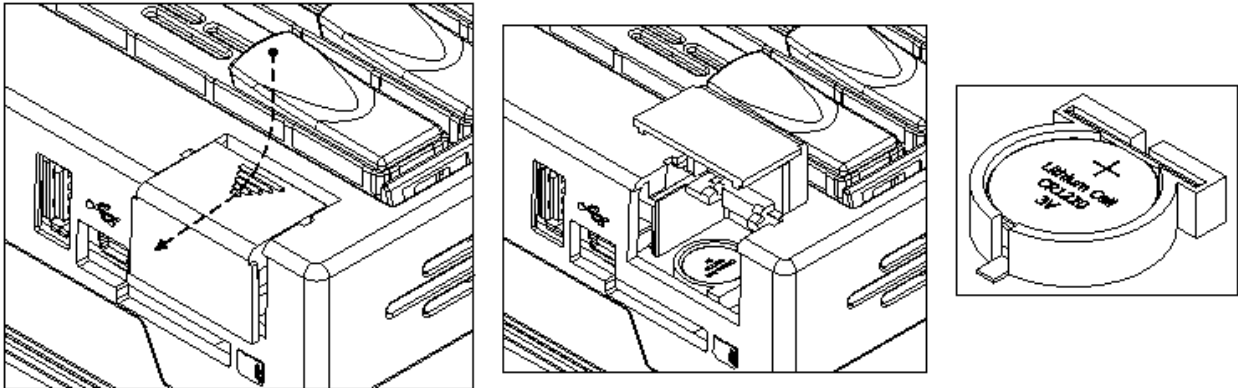
J3 pin 2-3 (T): Line terminated (120ohm)

4 Put the cover back on.

# CREW Manual

## Changing the battery

- 1: Switch off EW.
- 2: Open the battery door.



- 3: Changing the battery of the mother board.

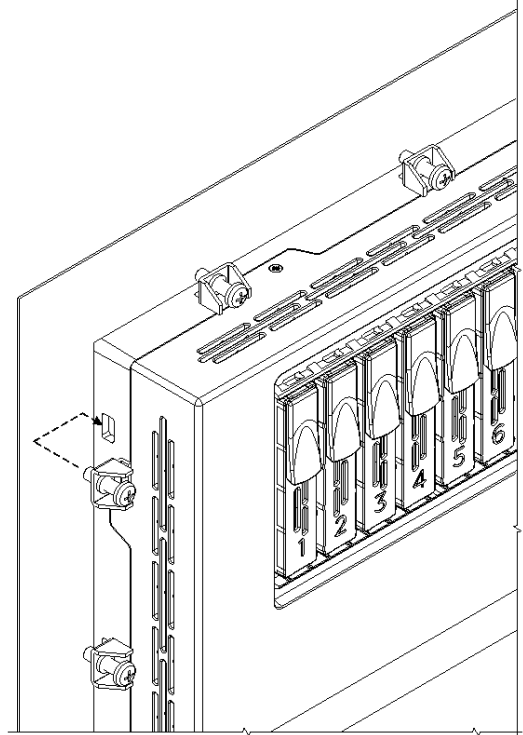
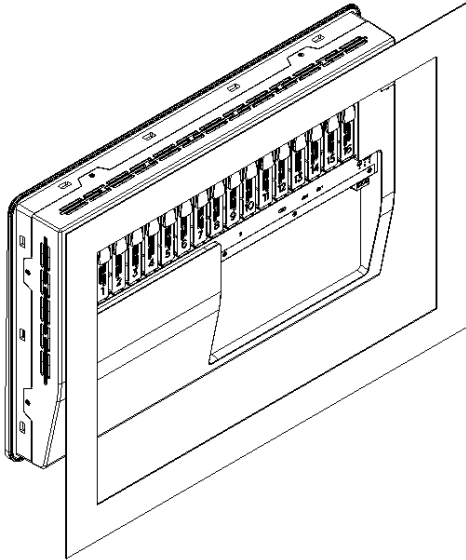


Attention: CR1220 3V Lithium battery- Put the new battery in observing type and polarity. Do not release the batteries into the environment

4. Close the battery door.

# CREW Manual

## EW Installation

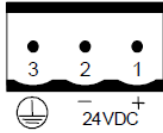


Tightening torque: 1.24 Nm (11 lbs. in.) +/- 5%

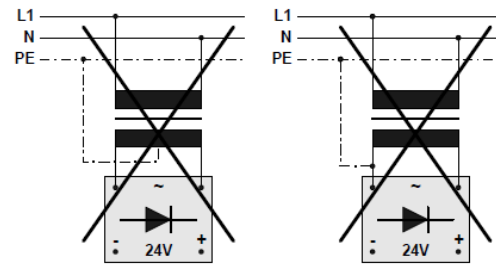
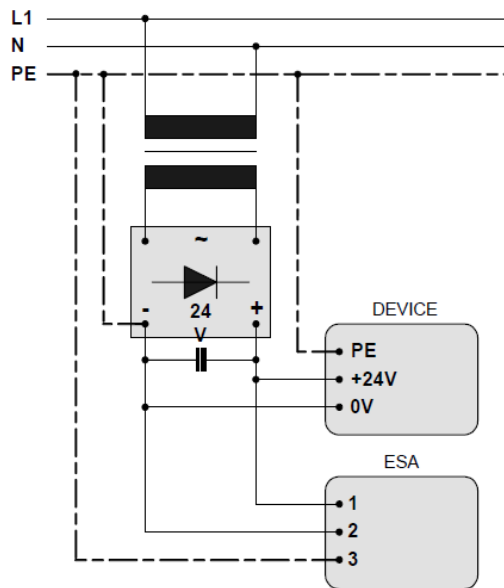


# CREW Manual

## Power supply



Power supply 3 pins connector		
1	+L 24 VDC	AWG12 - AWG30
2	M 0 V	
3	PE Protective ground	



**Warning:** These two configuration will seriously damage components.



### IMPORTANT:


The mass of the devices connected to the serial and/or parallel communication ports must be absolutely at the same potential of the 0V power supply of the EW. The circulation of a current between 0V power supply and the mass of the communication ports could damage some components of the EW and the devices connected to it.


## Electromagnetic compatibility

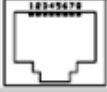
Operating restrictions: the protection requirements are not guaranteed in residential areas.

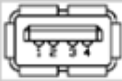
# CREW Manual


## Communication ports


RS232/485	
	
Db 9 female connector	
1	Tx Rx485 +IN/OUT
2	RX1 IN
3	TX1 OUT
4	N.C.
5	Signal GND
6	Tx Rx485 -IN/OUT
7	RTS1 OUT
8	CTS1 IN
9	+ 5 VDC (reserved)

Profibus-DP	
	
Db 9 female connector	
1	Shield
2	N.C.
3	TXRX485+ Data B
4	Repeater-Control-signal RTS
5	Signal GND
6	P5V
7	N.C.
8	TXRX485- Data A
9	N.C.

2 x ETH10/100	
	
RJ45 8 pin female connector	
1	TX+
2	TX-
3	RX+
4	N.C.
5	N.C.
6	RX-
7	N.C.
8	N.C.

2 x USB-A	
	
4 pin male connector	
1	USBVDC (OUT)
2	USBD-
3	USBD+
4	Signal GND

Mini USB-B	
	
5 pin male connector	
1	USBVDC (IN)
2	USBD-
3	USBD+
4	N.C.
5	Signal GND

CAN	
	
5 pin female connector	
1	V -
2	CAN -
3	Shield
4	CAN +
5	N.C.



Note:

N.C.: Not connected.

CREW Manual

EW6xxxxxx



EW INPUT/OUTPUT

# CREW Manual



## Description

The series of Esaware I/O modules completes the HMI EW100C range. Thanks to EW600s, it is now possible to command and control all your applications.

# CREW Manual



## Performance

The modularity of EW600 makes it possible to create various configurations. All of the I/Os are designed to guarantee excellent ergonomics and maximum ease of installation.

# CREW Manual



## Features

Thanks to simple spring clamps, the EW600 I/Os are equipped with a fast cabling system that can be easily set up by taking the connectors out of their seat.

Esaware offers the following types of I/O modules:

### Digital I/Os

Mixed input and output modules opto-isolated to prevent any module-damaging surges or overloads.

### Analogue I/Os

Mixed input and output modules, for managing the various types of analogue signals produced by the field and for regulating the different actuators so that you have control over your application. Thanks to the powerful signal processing unit, the analogue modules guarantee high control precision.

# CREW Manual

## **Fast Inputs**

Opto-isolated fast input module for the acquisition of signals up to 100 KHz.

## **Thermocouple Inputs**

Input module for thermocouples with integrated or external cold junction.

## **Thermistor Inputs**

Input module for thermistors.

## **Fast Outputs**

High speed output module to control signals up to 300 KHz.

## **PWM (Pulse with modulation) Output**

PWM output module to control signals up to 300 KHz.

# CREW Manual

## Technical specifications

### DIGITAL I/O

EW600B08B04 8 Digital Input + 4 Digital Output

Supply Voltage (Vdc)	24
Insulation	Optoisolated
Input Numbers	8
Input Type	PNP, NPN
Output Numbers	4
Output Type	PNP (300 mA/output)
Operating Temperature (°C)	-10 ... + 50 non condensing
Storage Temperature (°C)	-20 ... + 65
Humidity	<90% (non condensing)
External dimensions (W/H/D) (mm)	96 x 72 x 20
Protection Degree	IP 20
Certifications	CE / EN60068-2-6 / EN60068-2-27 / Humidity EN60068-2-30 / cULus (Certificate no. E189179) / LOGO EAC(Image) / Directive 94/9/EC Atex Group II - Category 3 G-D Zone 2/22(Mounted on EW100AC)



# CREW Manual

## ANALOG I/O

EW600A03A02 3 Analog Input + 2 Analog Output

Supply Voltage (Vdc)	24
Input Numbers	3
Input Type	0 / 5 V, 0 / 10 V, +10 / -10 V, 0 / 20 mA, 4 / 20 mA
Output Numbers	2
Output Type	0 / 5 V, 0 / 10 V, +10 / -10 V, 0 / 20 mA, 4 / 20 mA
Resolution	16 Bit
Operating Temperature (°C)	-10 ... + 50 non condensing
Storage Temperature (°C)	-20 ... + 65
Humidity	<90% (non condensing)
External dimensions (W/H/D) (mm)	96 x 72 x 20
Protection Degree	IP 20
Certifications	CE / EN60068-2-6 / EN60068-2-27 / Humidity EN60068-2-30 / cULus (Certificate no. E189179) / LOGO EAC(Imagine) / Direttiva 94/9/EC Atex Group II - Category 3 G-D Zone 2/22(Mounted on EW100AC)

## HIGH SPEED INPUT

EW600C02N00 2 High Speed Input

Supply Voltage (Vdc)	24
Input Numbers	2
Insulation	Optoisolated
Input Type	Incremental Pulse / Differential phase (4x) / Up/Down / Pulse + Direction (5-30 Vdc)
Frequency (KHz)	100
Operating Temperature (°C)	-10 ... + 50 non condensing
Storage Temperature (°C)	-20 ... + 65
Humidity	<90% (non condensing)
External dimensions (W/H/D) (mm)	96 x 72 x 20
Protection Degree	IP 20
Certifications	CE / EN60068-2-6 / EN60068-2-27 / Humidity EN60068-2-30 / cULus (Certificate no. E189179) / LOGO EAC(Image) / Directive 94/9/EC Atex Group II - Category 3 G-D Zone 2/22(Mounted on EW100AC)

# CREW Manual

## THERMOCOUPLE INPUT

EW600D06N00 6 Thermocouple Input

Input Numbers	6
Input Type	K / J / E / T / N / B / R / S
Resolution (°C)	+0,1° / - 0,1°
Cold Junction	Internal and External
Operating Temperature (°C)	-10 ... + 50 non condensing
Storage Temperature (°C)	-20 ... + 65
Humidity	<90% (non condensing)
External dimensions (W/H/D) (mm)	96 x 72 x 20
Protection Degree	IP 20
Certifications	CE / EN60068-2-6 / EN60068-2-27 / Humidity EN60068-2-30 / cULus (Certificate no. E189179) / LOGO EAC(Image) / Directive 94/9/EC Atex Group II - Category 3 G-D Zone 2/22(Mounted on EW100AC)

## THERMORESISTANCE INPUT

EW600E04N00 4 Resistance Thermometer Input

Input Numbers	4
Input Type	Pt100 / Pt200/Pt500/Pt1000 / Ni100/Ni1000
Resolution (°C)	+0,1° / - 0,1°
Operating Temperature (°C)	-10 ... + 50 non condensing
Storage Temperature (°C)	-20 ... + 65
Humidity	<90% (non condensing)
External dimensions (W/H/D) (mm)	96 x 72 x 20
Protection Degree	IP 20
Certifications	CE / EN60068-2-6 / EN60068-2-27 / Humidity EN60068-2-30 / cULus (Certificate no. E189179) / LOGO EAC(Image) / Directive 94/9/EC Atex Group II - Category 3 G-D Zone 2/22(Mounted on EW100AC)

# CREW Manual

## HIGH SPEED OUTPUT

EW600N00C04 4 High Speed Output

Supply Voltage (Vdc)	24
Output Numbers	4
Insulation	Optoisolated
Output Type	CW/CCW - Pulse+Direction 12 - 32Vdc push-pull
Output Current (mA)	5 - 10
Resolution (Hz - KHz)	200Hz - 300KHz
Operating Temperature (°C)	-10 ... + 50 non condensing
Storage Temperature (°C)	-20 ... + 65
Humidity	<90% (non condensing)
External dimensions (W/H/D) (mm)	96 x 72 x 20
Protection Degree	IP 20
Certifications	CE / EN60068-2-6 / EN60068-2-27 / Humidity EN60068-2-30 / cULus (Certificate no. E189179) / LOGO EAC(Image) / Directive 94/9/EC Atex Group II - Category 3 G-D Zone 2/22(Mounted on EW100AC)

## PULSE WITH MODULATION OUTPUT

EW600N00E04 4 PWM Output

Supply Voltage (Vdc)	24
Output Numbers	4
Insulation	Optoisolated
Output Type	PWM -12 - 32Vdc push-pull
Output Current (mA)	5 - 10
Resolution (Hz - KHz)	200Hz - 300KHz
Operating Temperature (°C)	-10 ... + 50 non condensing
Storage Temperature (°C)	-20 ... + 65
Humidity	<90% (non condensing)
External dimensions (W/H/D) (mm)	96 x 72 x 20
Protection Degree	IP 20
Certifications	CE / EN60068-2-6 / EN60068-2-27 / Humidity EN60068-2-30 / cULus (Certificate no. E189179) / LOGO EAC(Image) / Directive 94/9/EC Atex Group II - Category 3 G-D Zone 2/22(Mounted on EW100AC)

CREW Manual

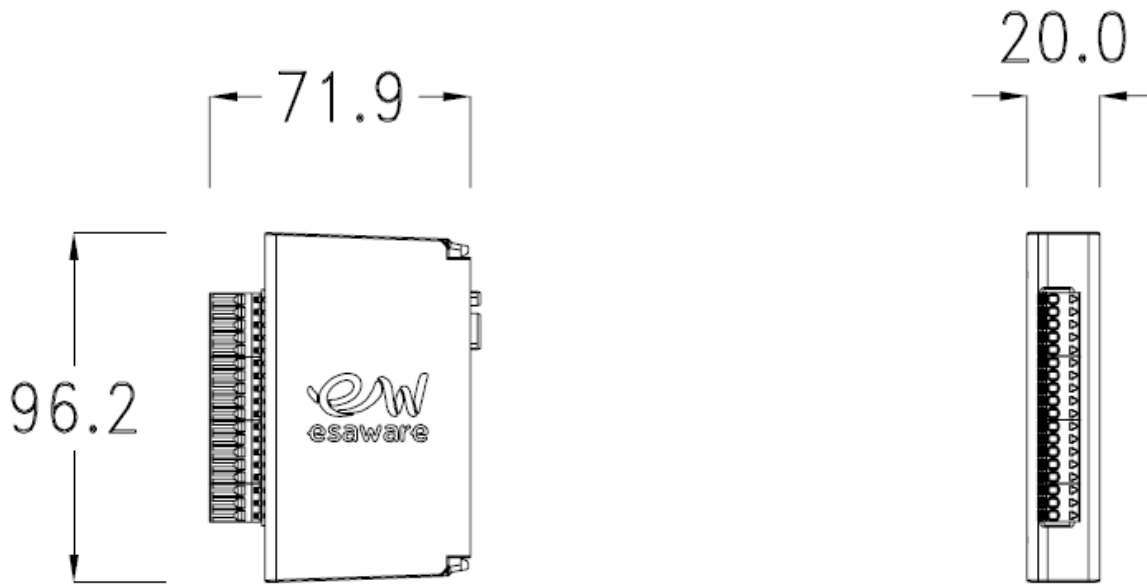
EW600Axxxxx



EW INPUTS / OUTPUTS

# CREW Manual

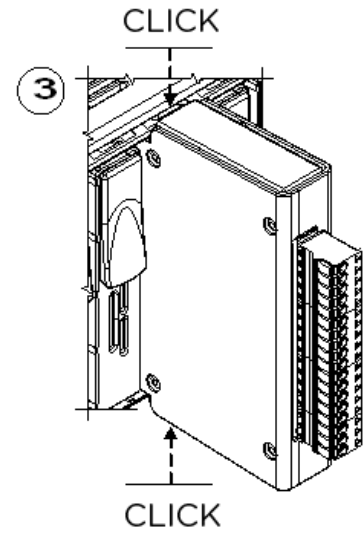
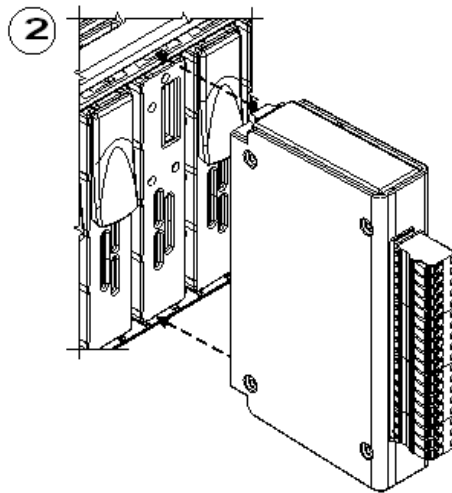
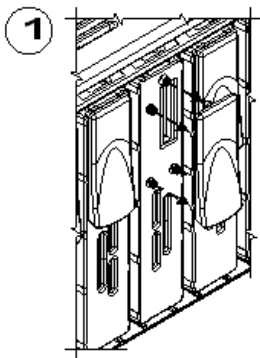
## Dimensions



# CREW Manual

## EW600 assembly on EW terminal

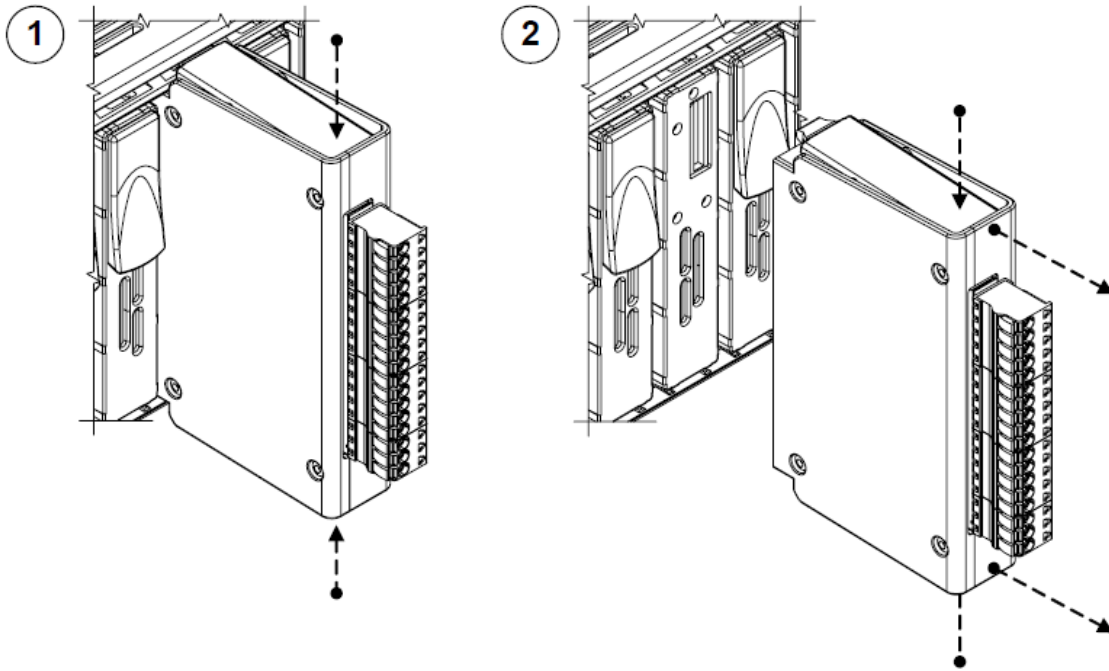
1. Switch off EW.



# CREW Manual

## EW600 removal from EW terminal

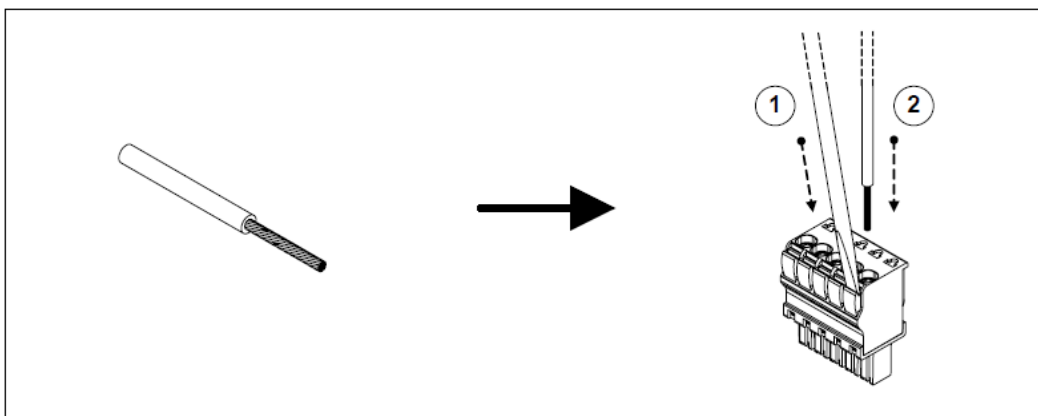
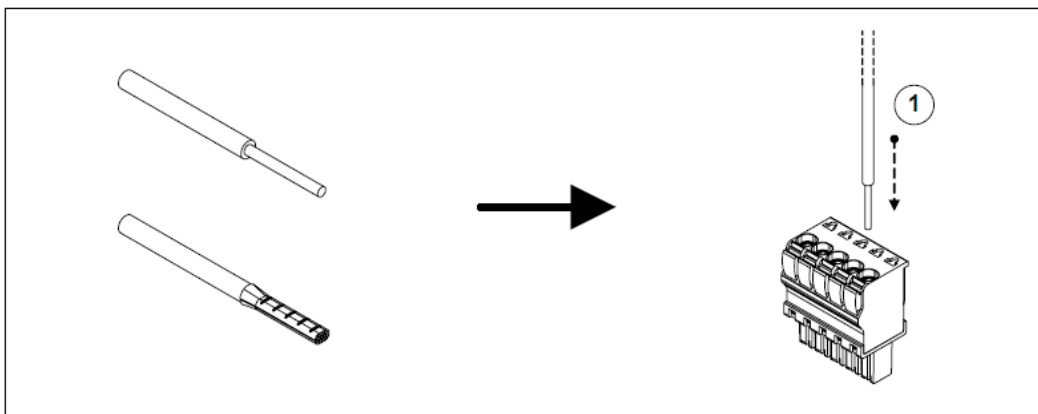
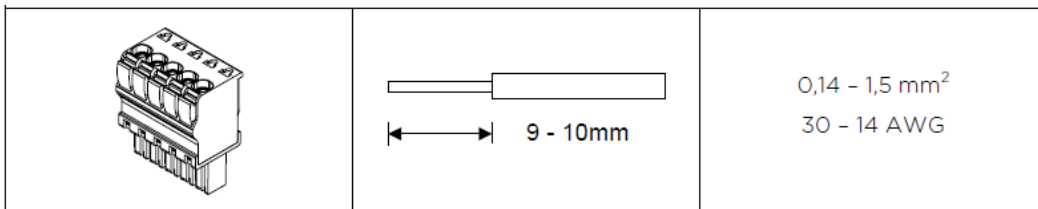
1. Switch off EW.



# CREW Manual

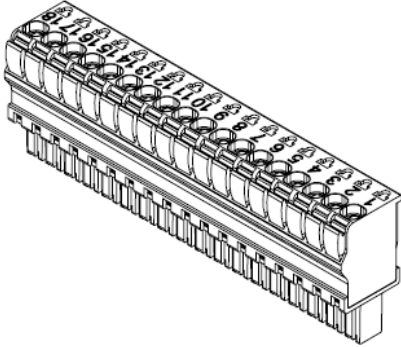
## Electrical connections

### Spring connector

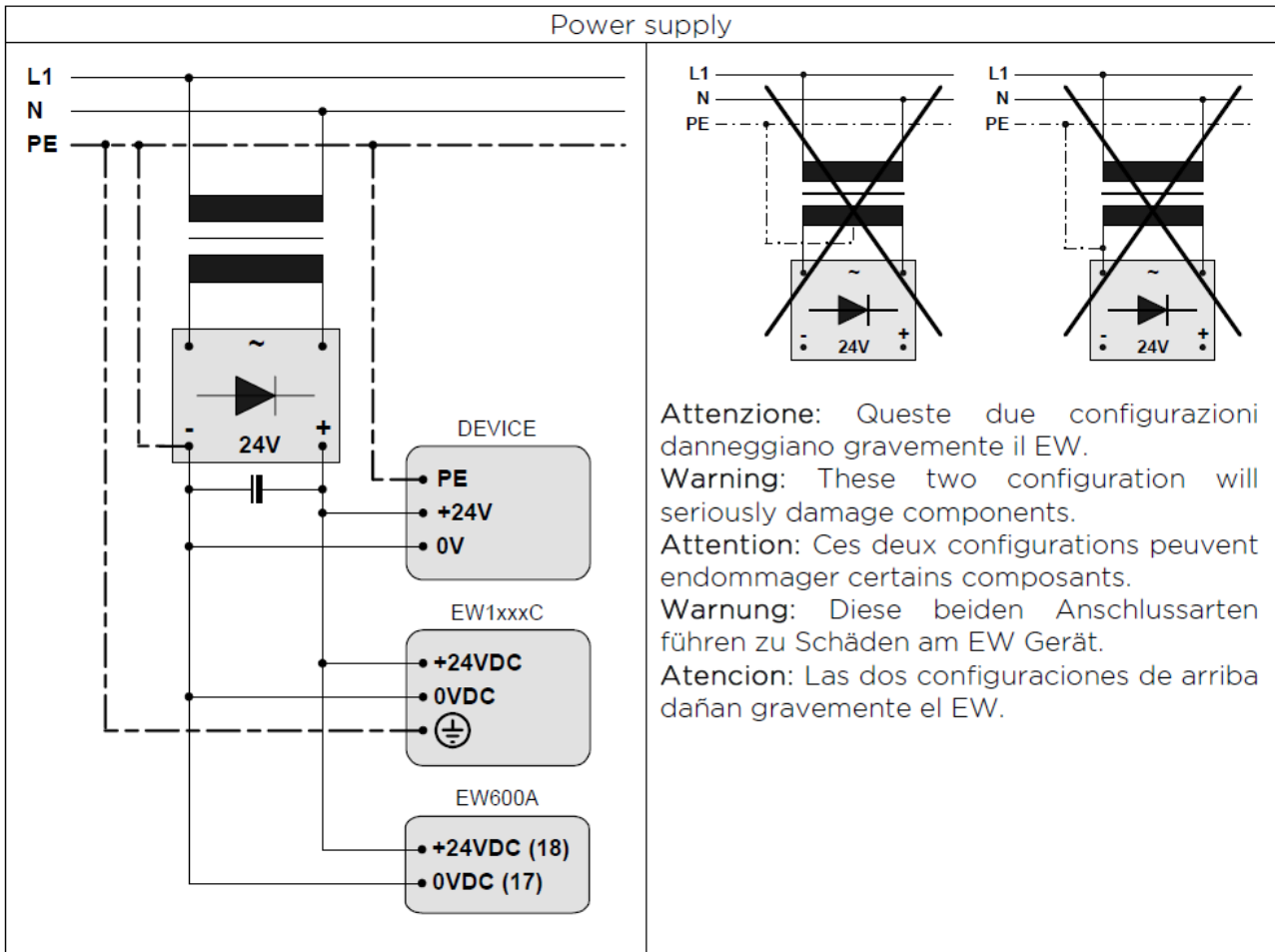




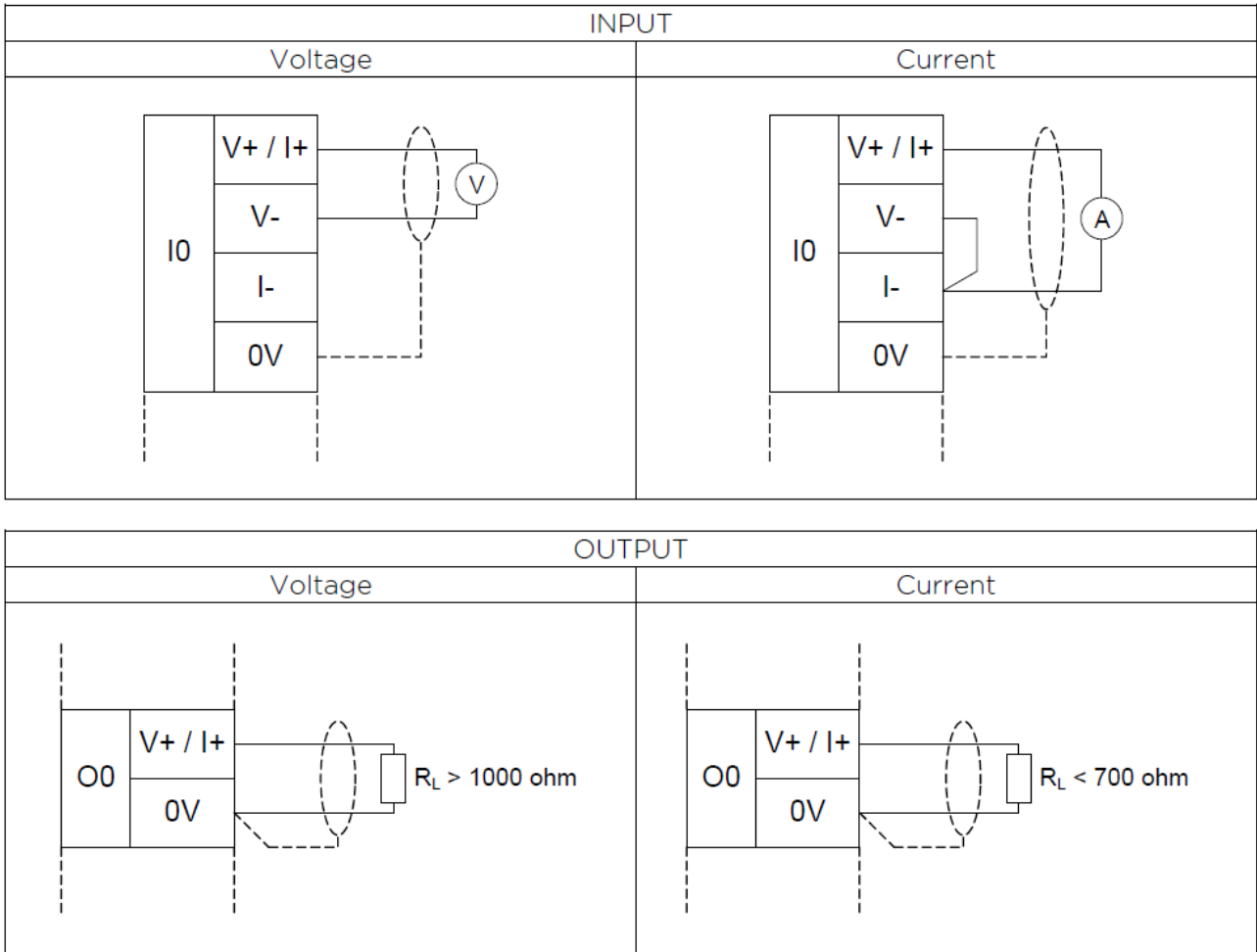
# CREW Manual

EW600A03A02 - I/O AN3IN AN2OUT - 16BIT 0... 5V / 0... 10V / -10... +10V / 0... 20mA / 4... 20mA (Software selectable)				
	1	I0	V+ / I+	
	2		V-	
	3		I-	
	4		OVDC	
	5	I1	V+ / I+	
	6		V-	
	7		I-	
	8	OVDC	I2	V+ / I+
	9	V-		
	10	I-		
	11	OVDC	O0	OVDC
	12	V+ / I+		
	13	OVDC	O1	OVDC
	14	V+ / I+		
	15	OVDC		OVDC
	16			V+ / I+
	17			OVDC
	18			+24VDC

# CREW Manual



# CREW Manual



CREW Manual

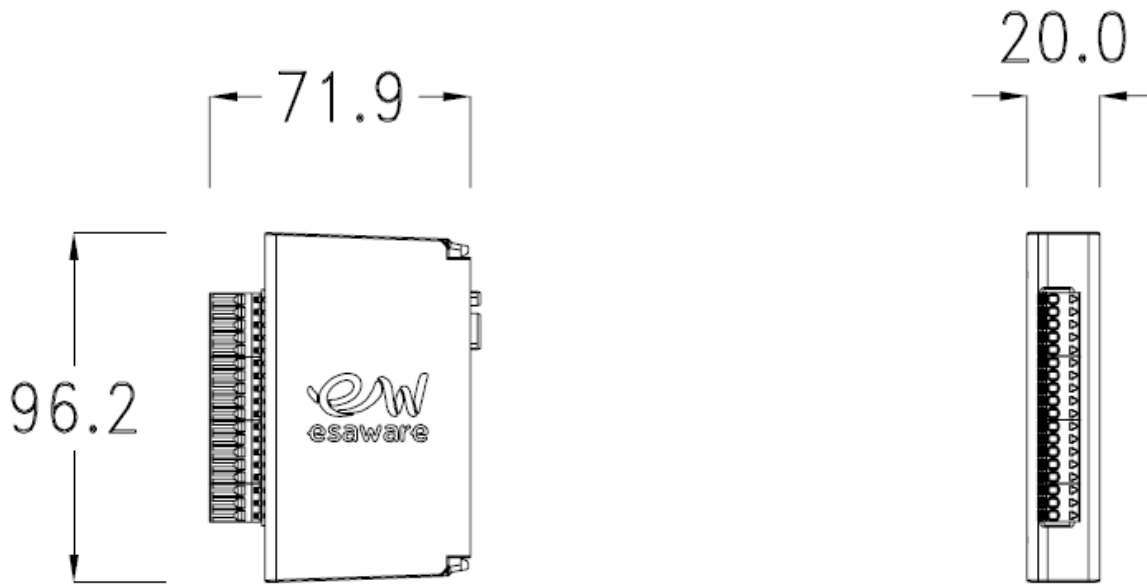
EW600Bxxxxx



EW INPUTS / OUTPUTS

# CREW Manual

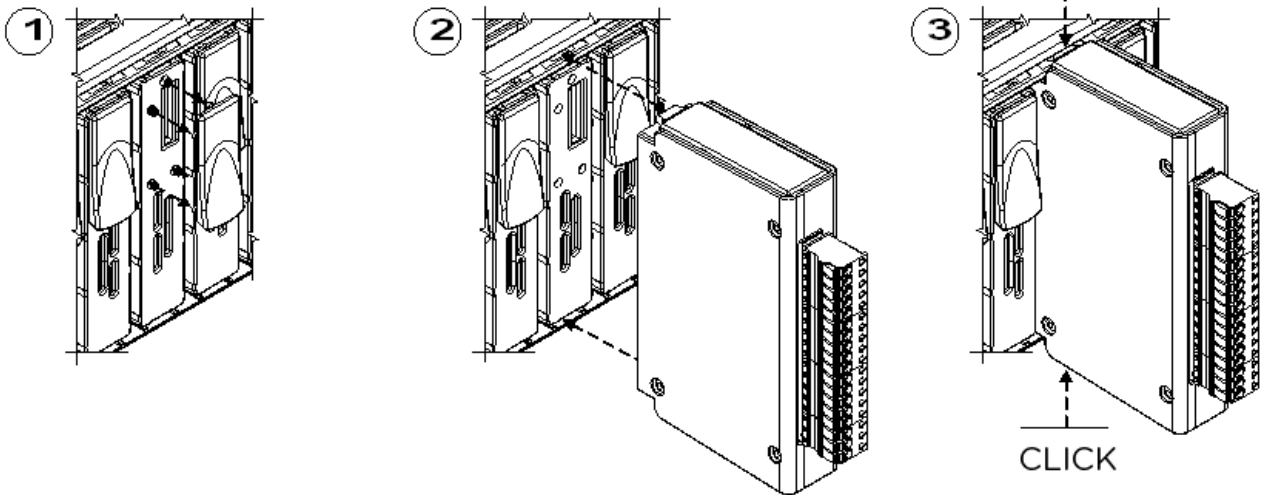
## Dimensions



# CREW Manual

## EW600 assembly on EW terminal

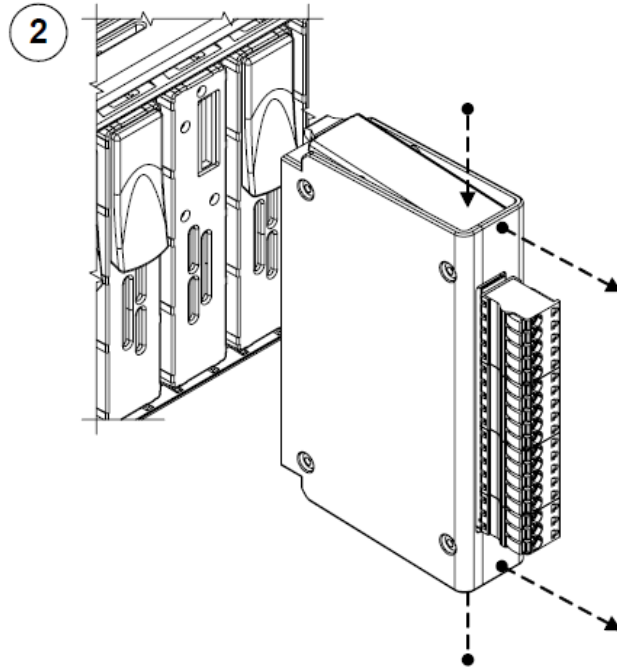
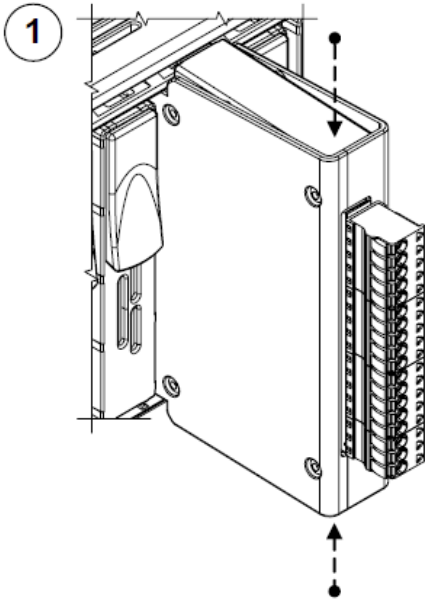
1. Switch off EW.



# CREW Manual

## EW600 removal from EW terminal

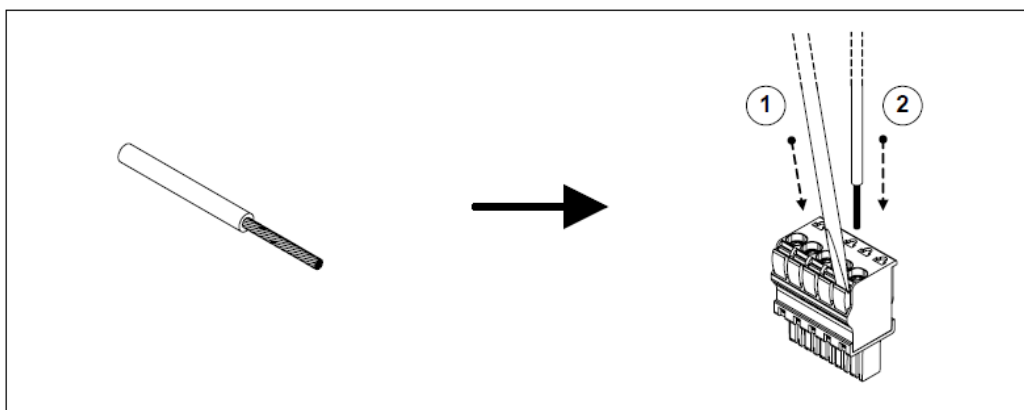
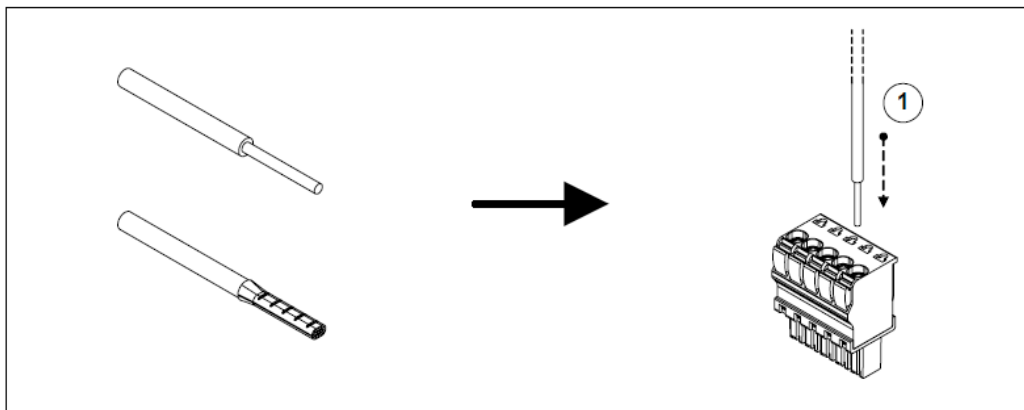
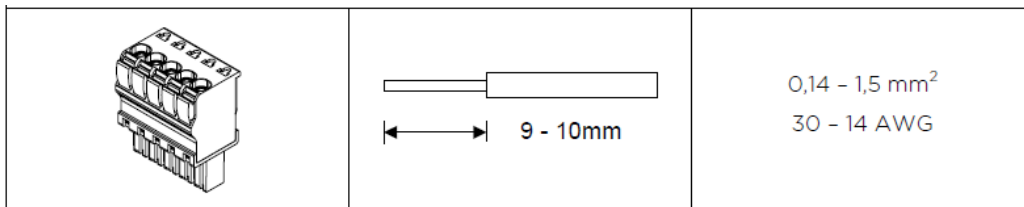
1. Switch off EW.



# CREW Manual

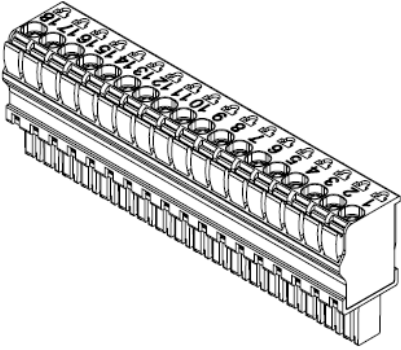
## Electrical connections

### Spring connector

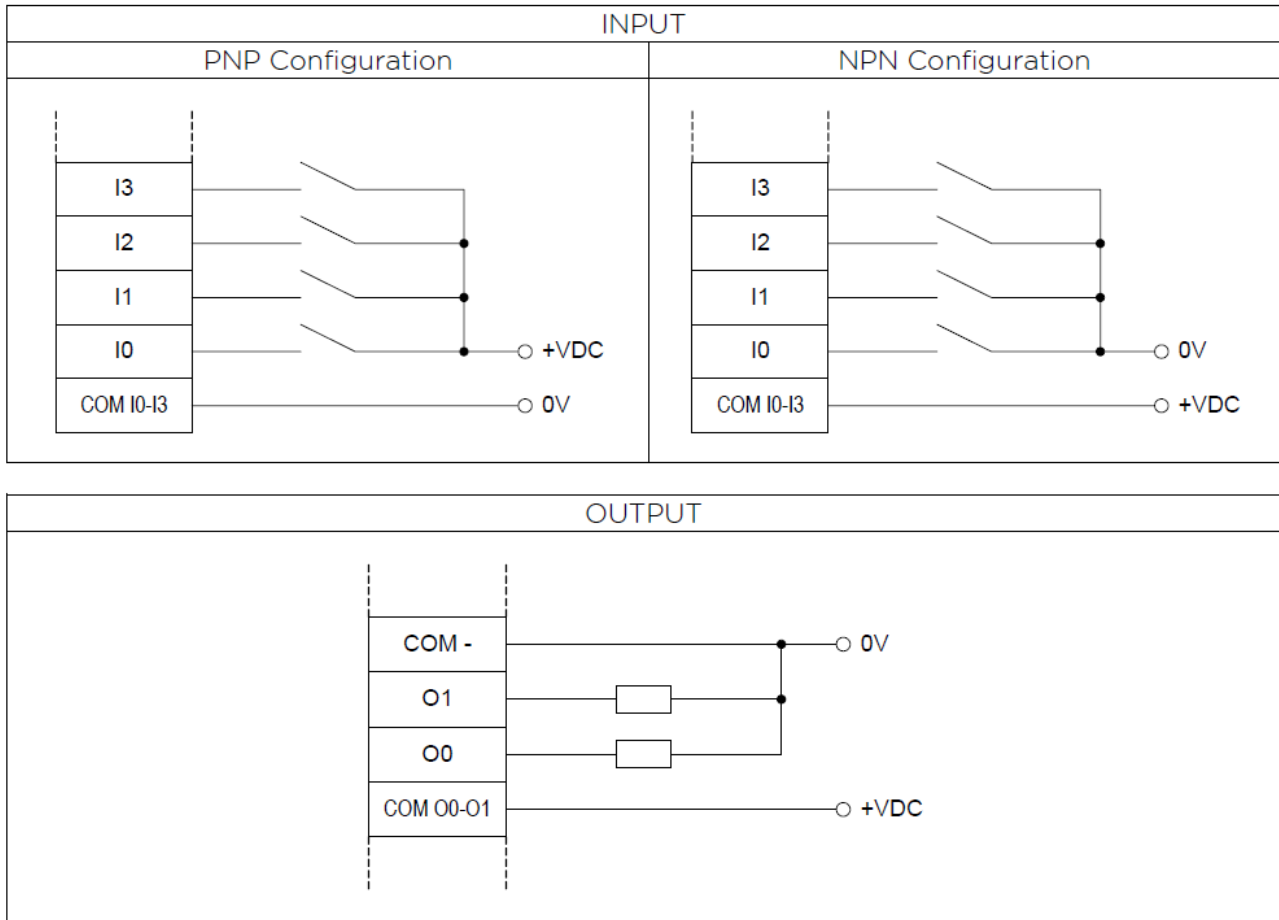




# CREW Manual

EW600B08B04 - I/O DIG8IN DIG4OUT - Opto-isolated		
	1	COM I0-I3
	2	I0
	3	I1
	4	I2
	5	I3
	6	COM I4-I7
	7	I4
	8	I5
	9	I6
	10	I7
	11	Not Connected
	12	COM O0-O1
	13	O0
	14	O1
	15	COM -
	16	O2
	17	O3
	18	COM O2-O3

# CREW Manual



CREW Manual

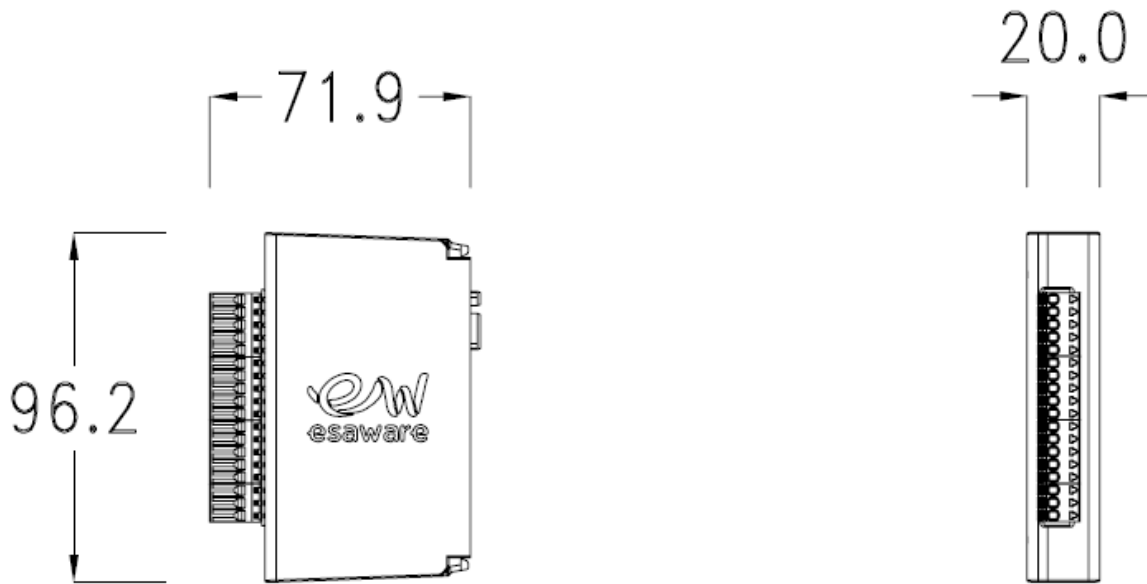
EW600Cxxxxx



EW INPUTS / OUTPUTS

# CREW Manual

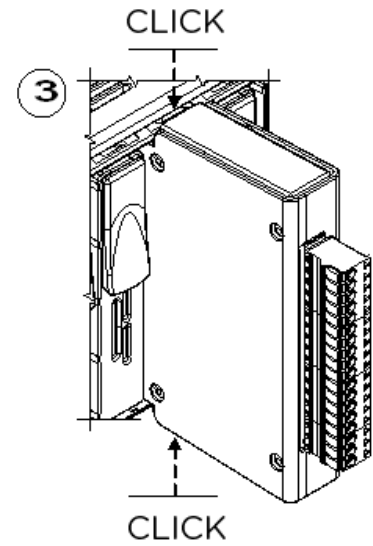
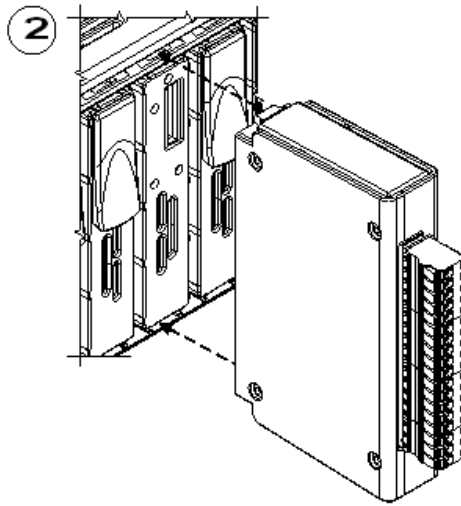
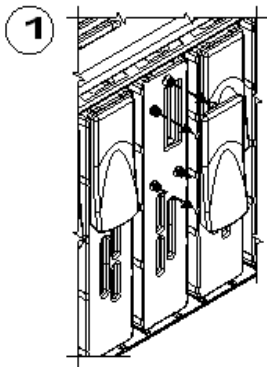
## Dimensions



# CREW Manual

## EW600 assembly on EW terminal

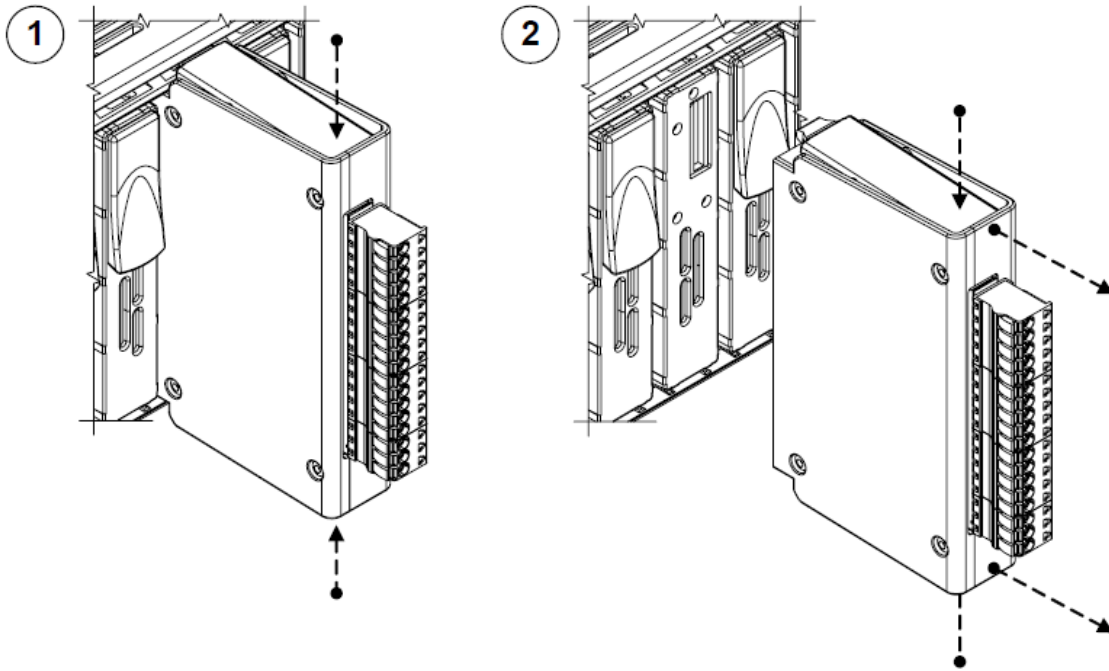
1. Switch off EW.



# CREW Manual

## EW600 removal from EW terminal

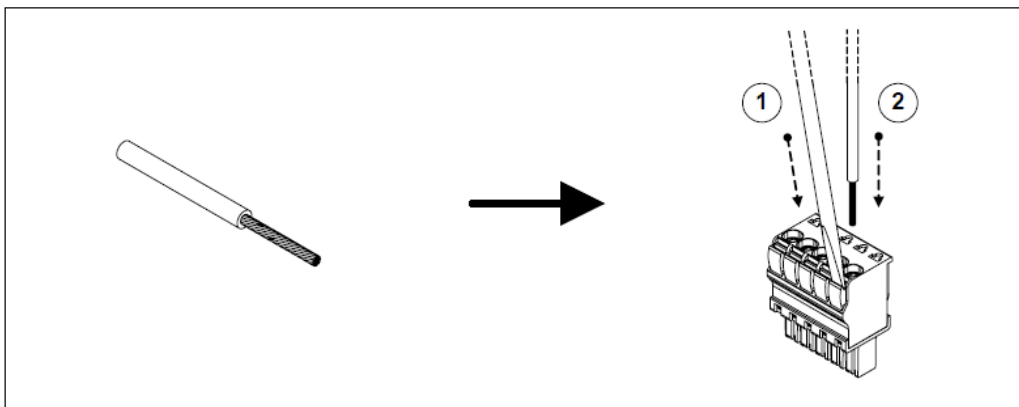
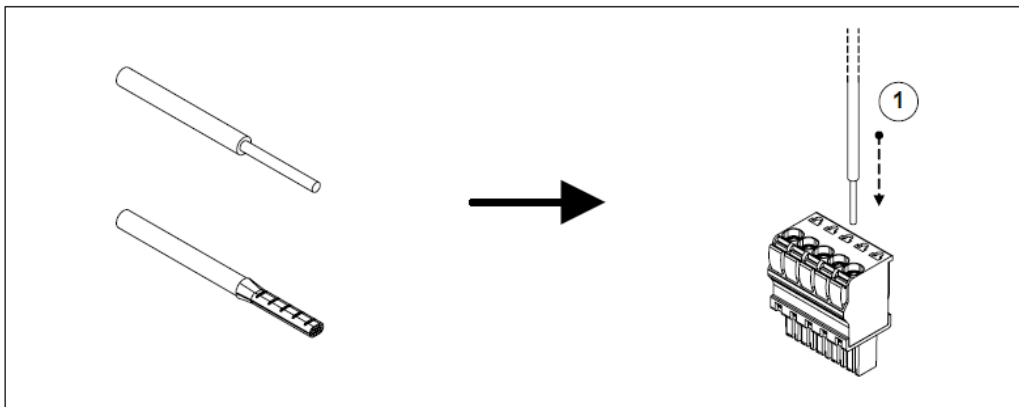
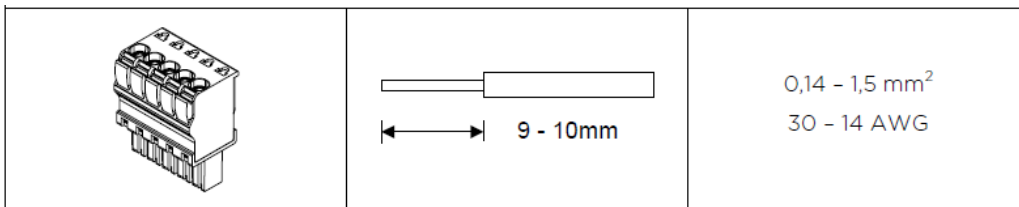
1. Switch off EW.



# CREW Manual

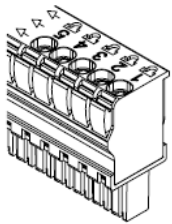
## Electrical connections

### Spring connector



# CREW Manual

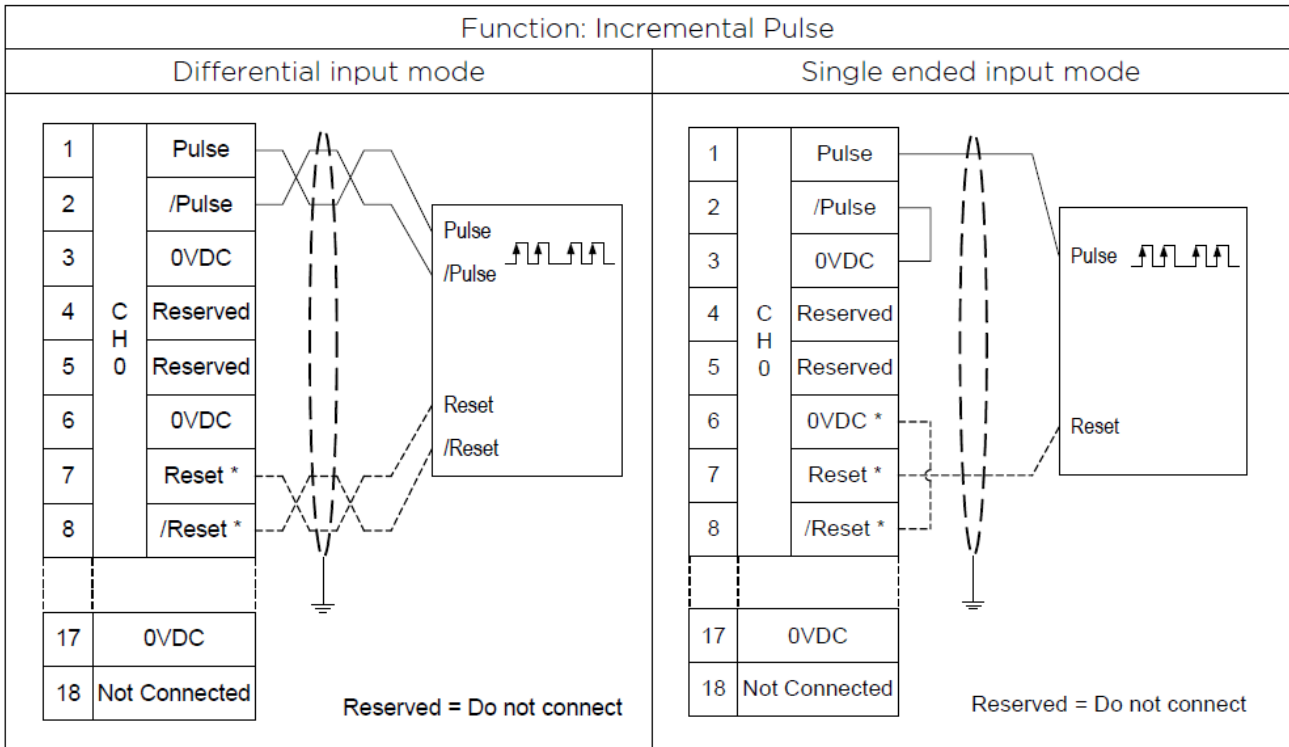
EW600C02N00 - I/O FAST INPUT 2IN - Opto-isolated Max. frequency 100KHz - Input voltage 5-30V						
Pin	Function (software selectable)				CH	
	Incremental pulse input	Differential phase input (4x)	Up/Down input	Pulse+Direction input		
1	Pulse	A	Up pulse	Pulse	0	
2	/Pulse	/A	/Up pulse	/Pulse		
3	OVDC	OVDC	OVDC	OVDC		
4	Reserved **	B	Down pulse	Direction		
5	Reserved **	/B	/Down pulse	/Direction		
6	OVDC	OVDC	OVDC	OVDC		
7	Reset *	Z *	Reset *	Reset *		
8	/Reset *	/Z *	/Reset *	/Reset *		
9	Pulse	A	Up pulse	Pulse	1	
10	/Pulse	/A	/Up pulse	/Pulse		
11	OVDC	OVDC	OVDC	OVDC		
12	Not Used	B	Down pulse	Direction		
13	Not Used	/B	/Down pulse	/Direction		
14	OVDC	OVDC	OVDC	OVDC		
15	Reset *	Z *	Reset *	Reset *		
16	/Reset *	/Z *	/Reset *	/Reset *		
17	OVDC					
18	Not Connected					



\* Software configurable    \*\* Reserved = Do not connect

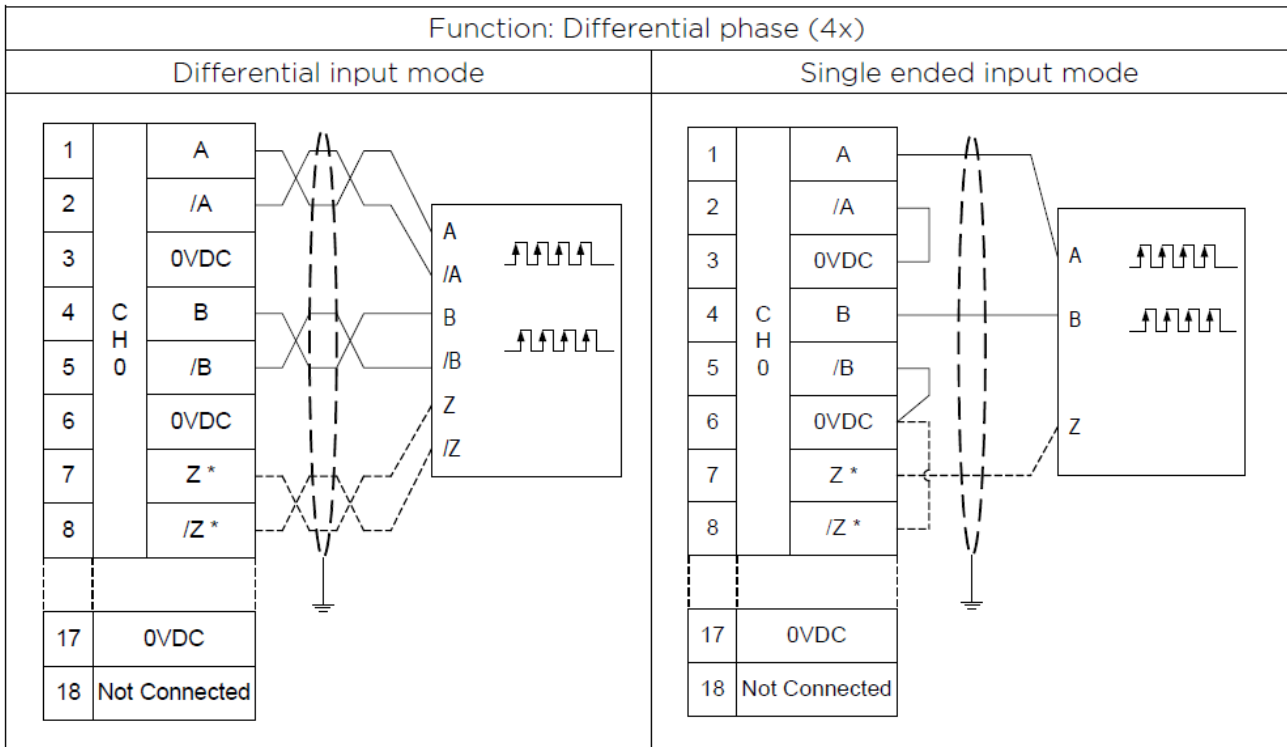


# CREW Manual



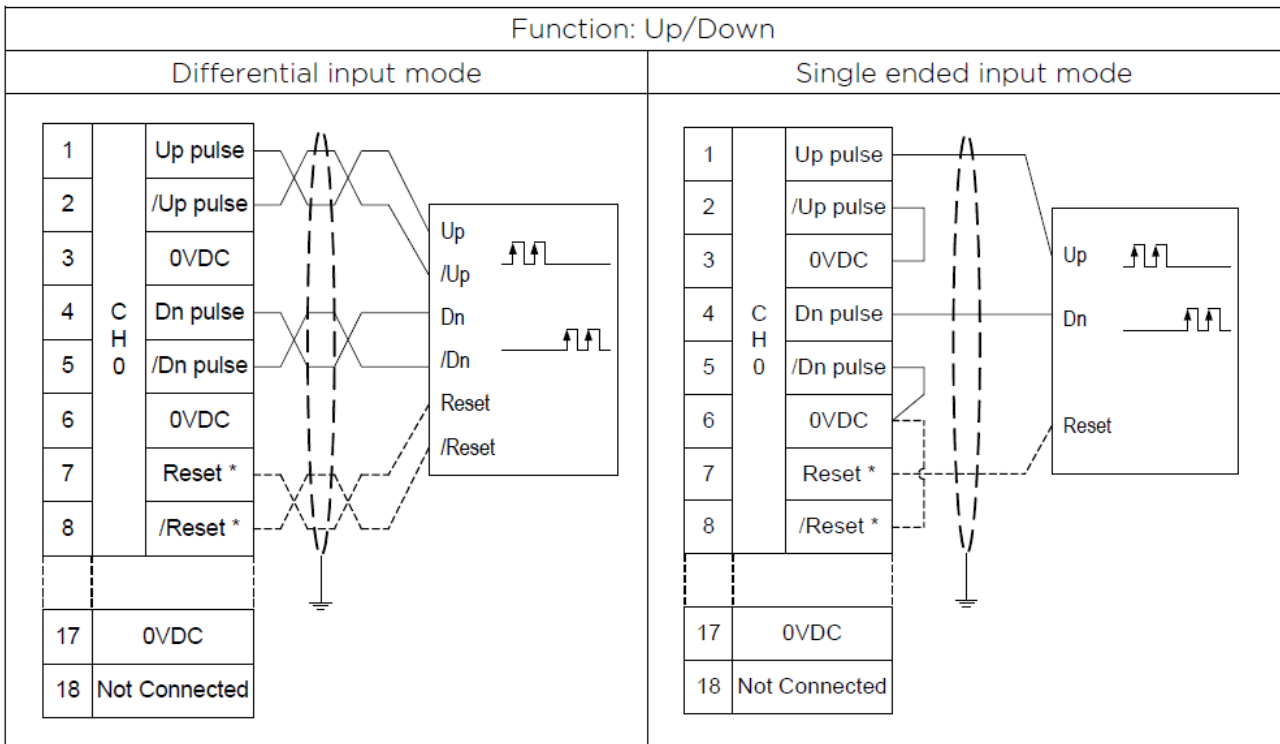
\* Connect only if the signal is configured by software

# CREW Manual



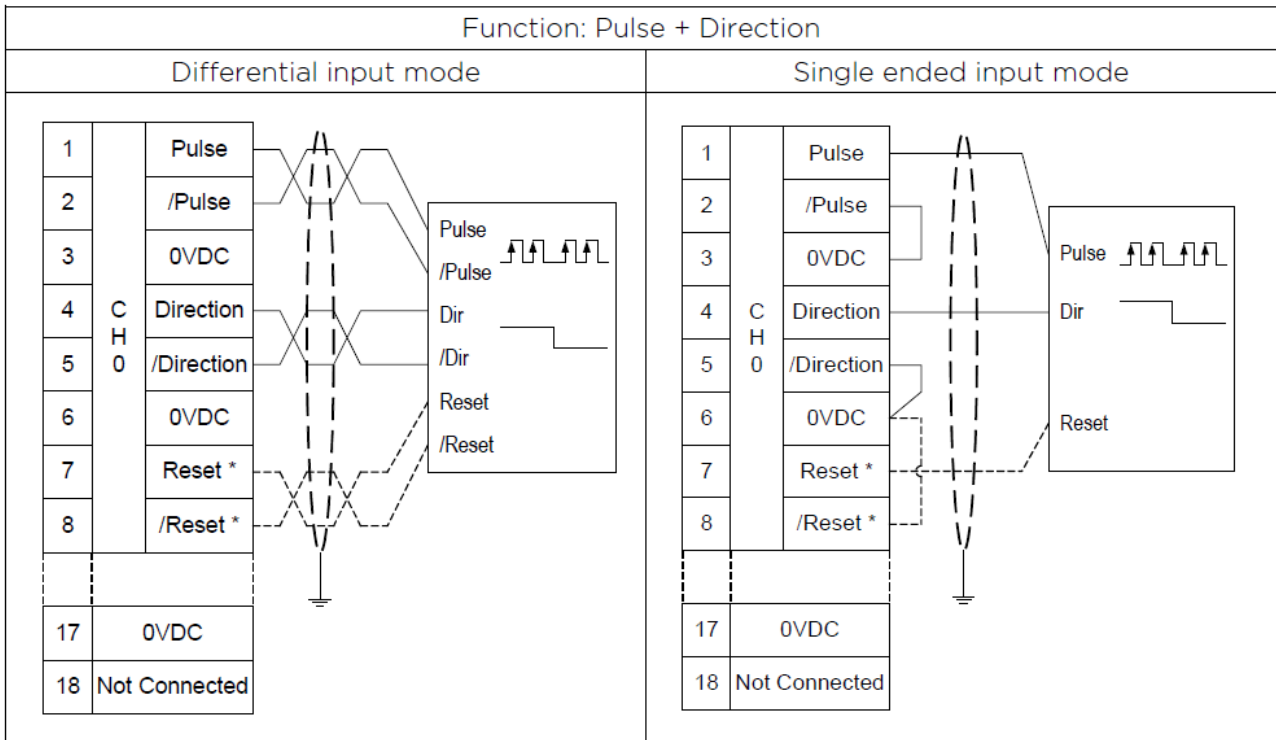
\* Connect only if the signal is configured by software

# CREW Manual



\* Connect only if the signal is configured by software

# CREW Manual



\* Connect only if the signal is configured by software

CREW Manual

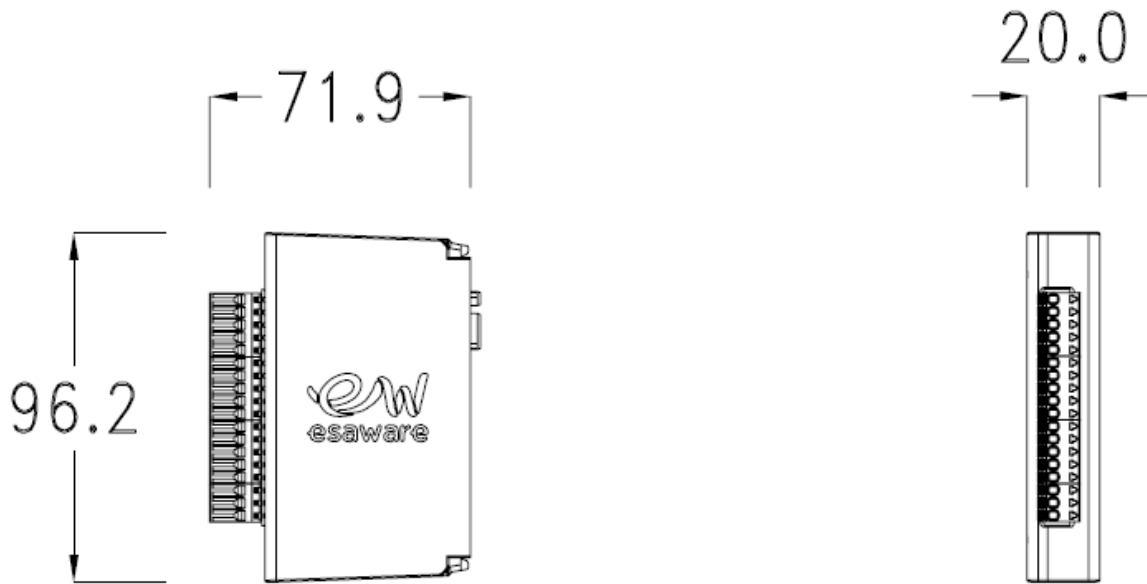
EW600Dxxxxx



EW INPUTS / OUTPUTS

# CREW Manual

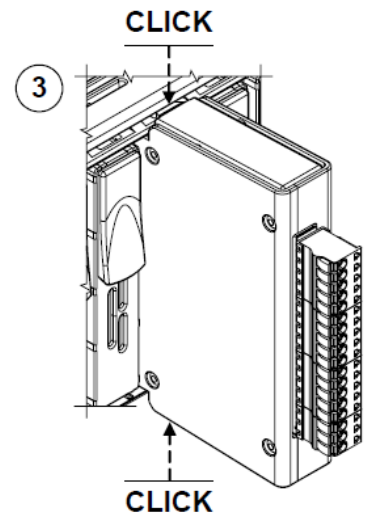
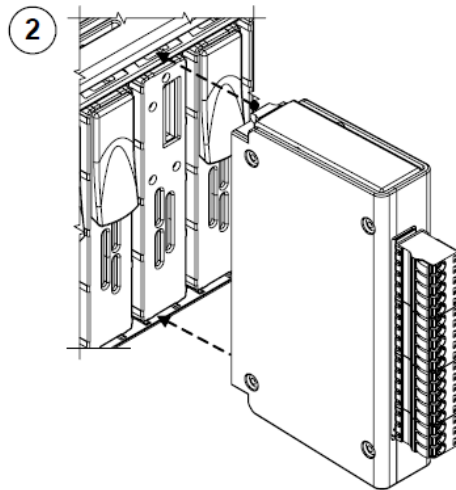
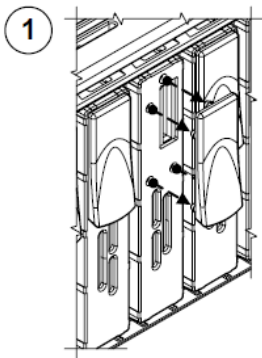
## Dimensions



# CREW Manual

## EW600 assembly on EW terminal

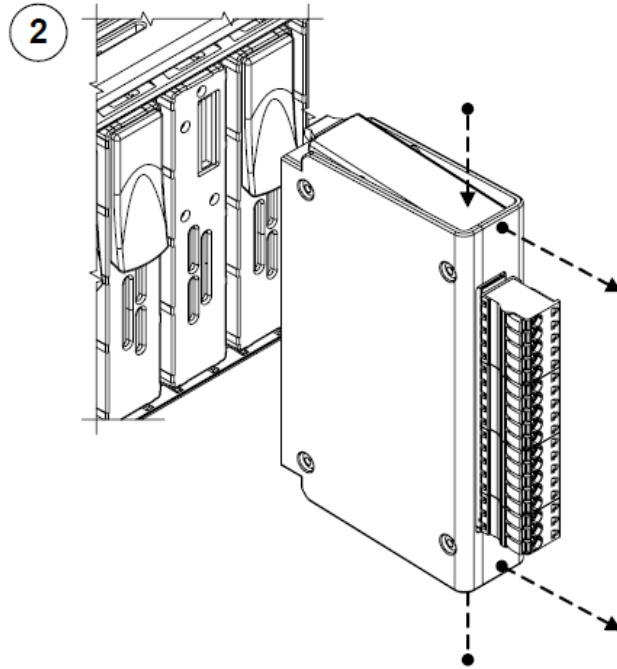
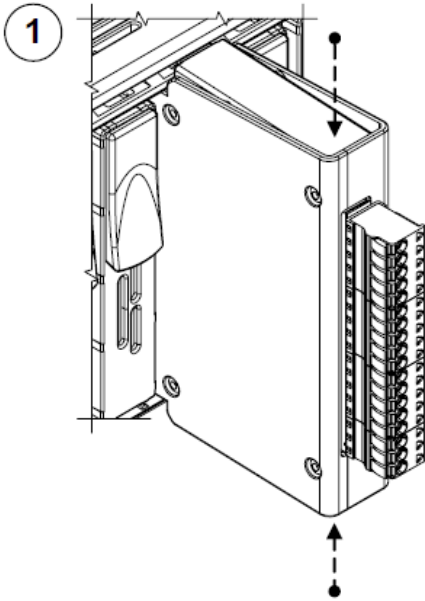
1. Switch off EW.



# CREW Manual

## EW600 removal from EW terminal

1. Switch off EW.

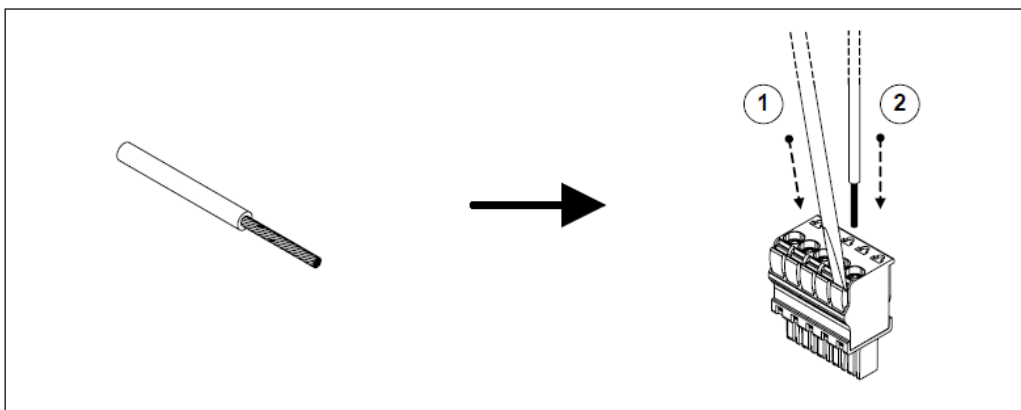
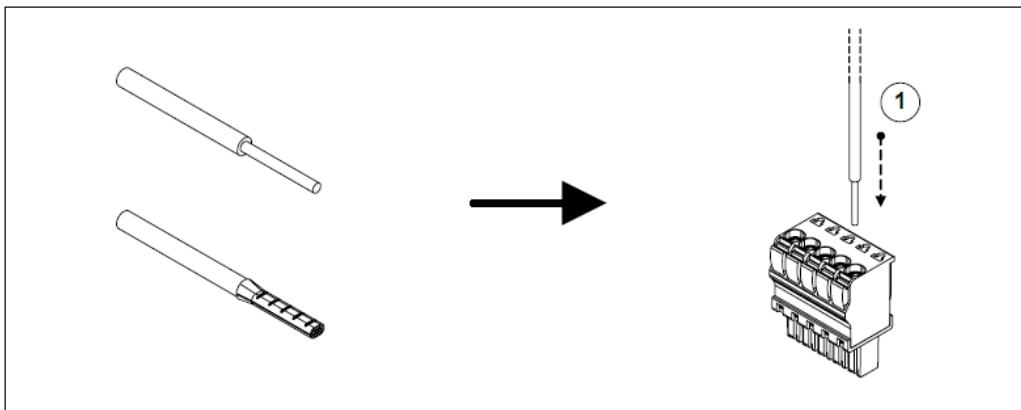
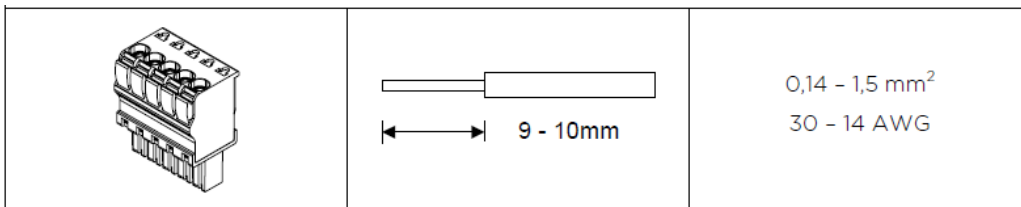




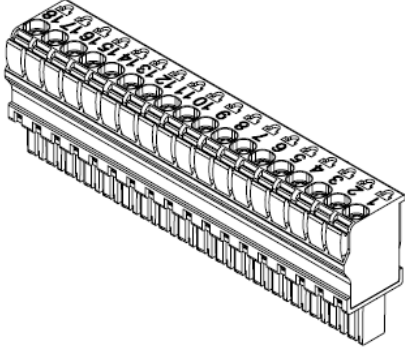
# CREW Manual

## Electrical connections

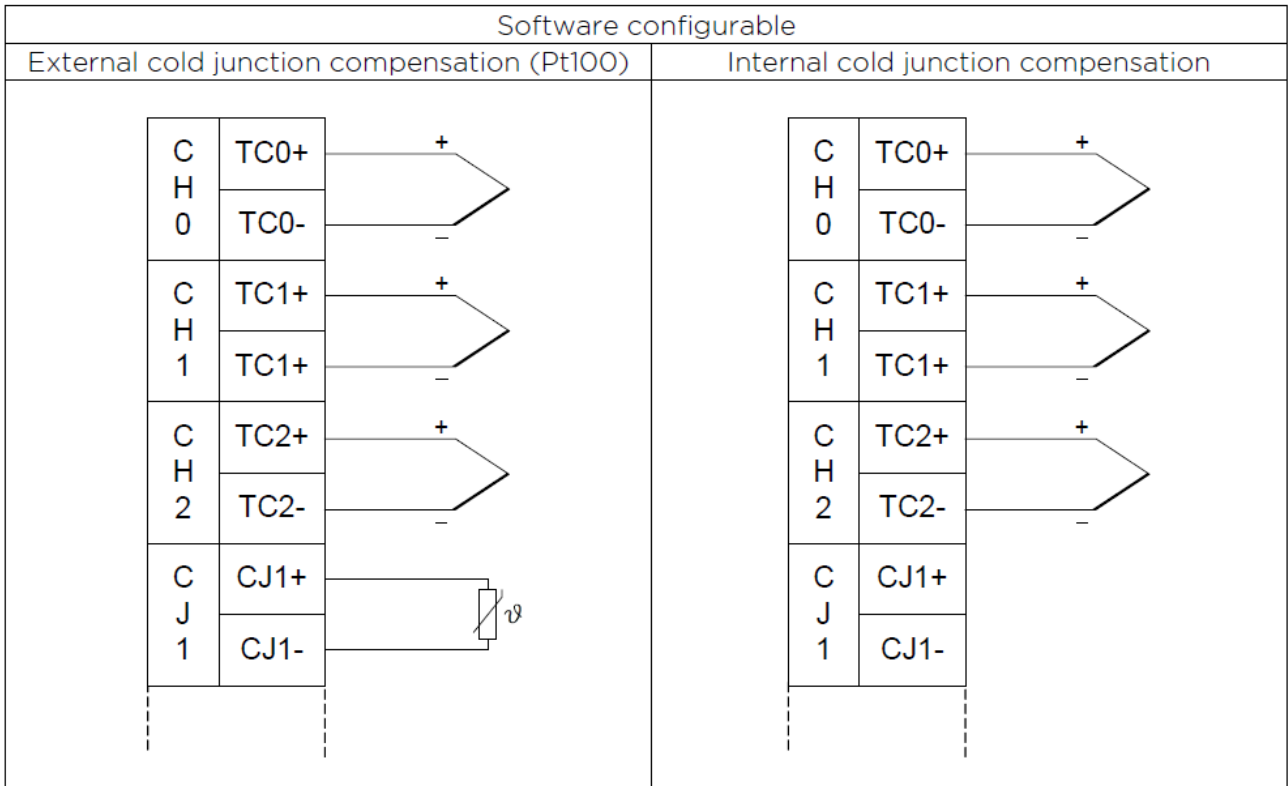
### Spring connector



# CREW Manual

EW600D06N00 - I/O THERMOCOUPLE 6IN Type K / J / E / T / N / B / R / S (software selectable)			
	1	CH0	TC0+
	2		TC0-
	3	CH1	TC1+
	4		TC1-
	5	CH2	TC2+
	6		TC2-
	7	CJ1	CJ1+
	8		CJ1-
	9	CH3	TC3+
	10		TC3-
	11	CH4	TC4+
	12		TC4-
	13	CH5	TC5+
	14		TC5-
	15	CJ2	CJ2+
	16		CJ2-
	17	Not Connected	
	18	Not Connected	

# CREW Manual



CREW Manual

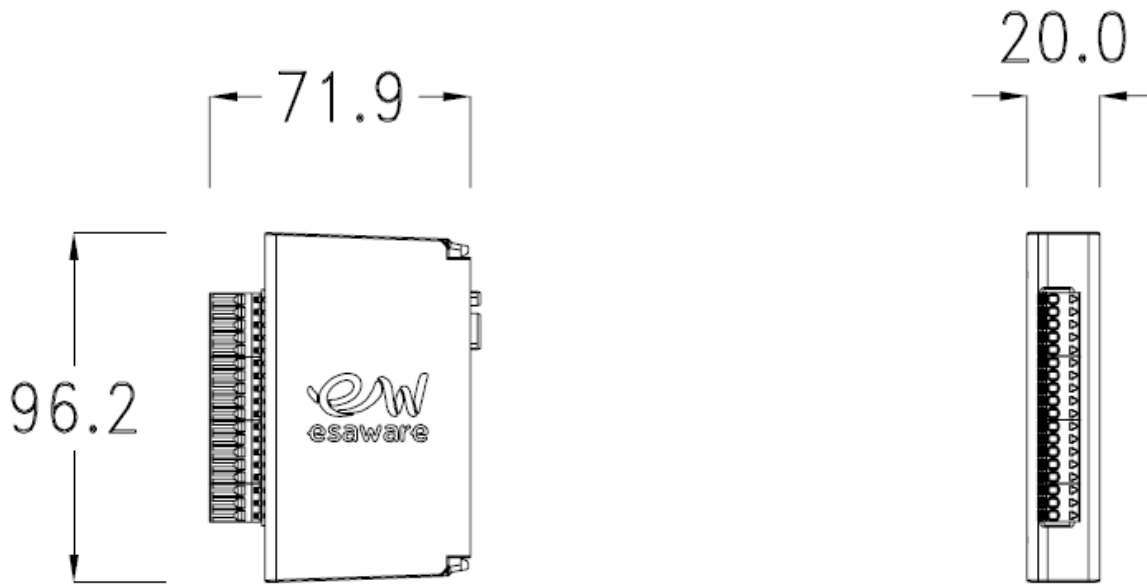
EW600Exxxx



EW INPUTS / OUTPUTS

# CREW Manual

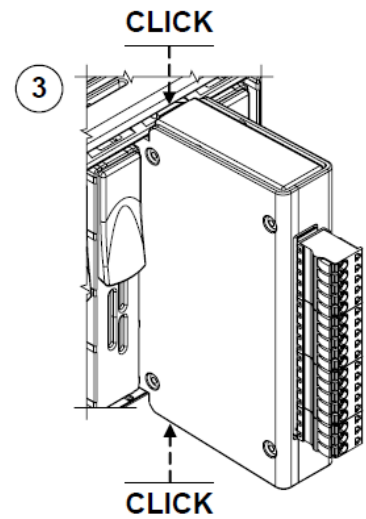
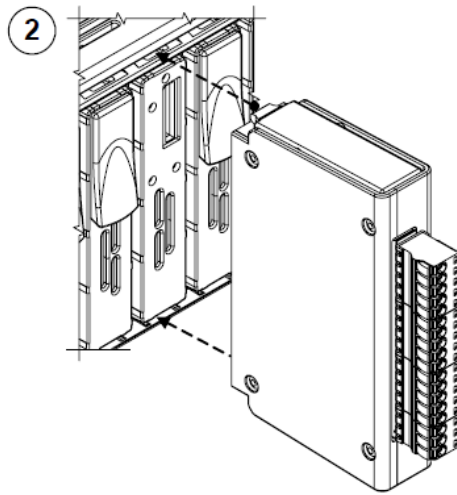
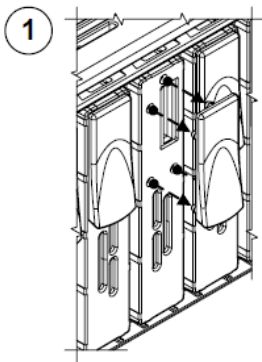
## Dimensions



# CREW Manual

## EW600 assembly on EW terminal

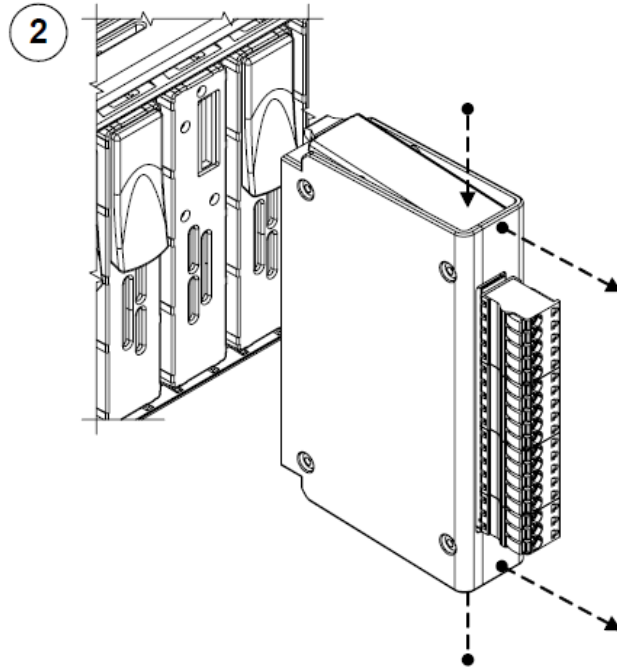
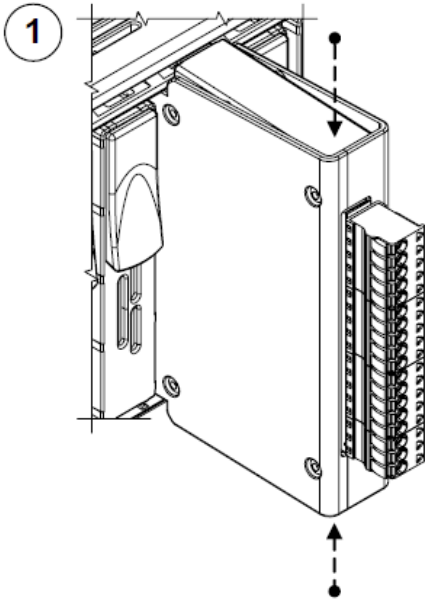
1. Switch off EW.



# CREW Manual

## EW600 removal from EW terminal

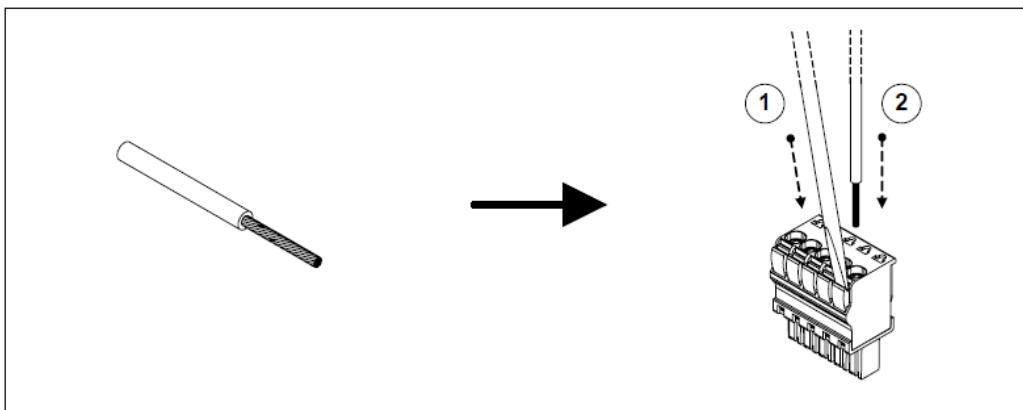
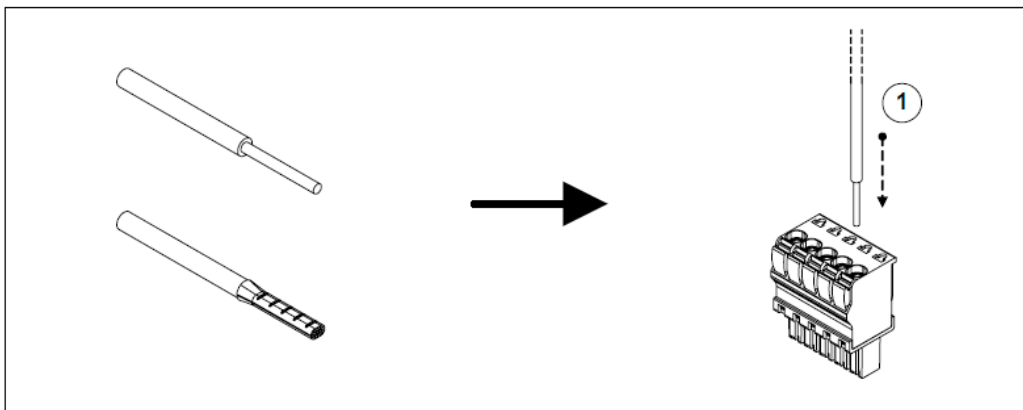
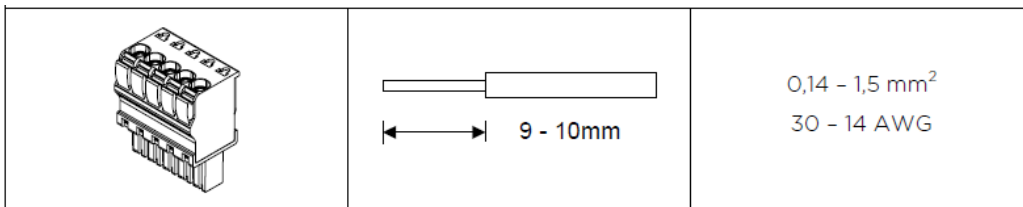
1. Switch off EW.



# CREW Manual

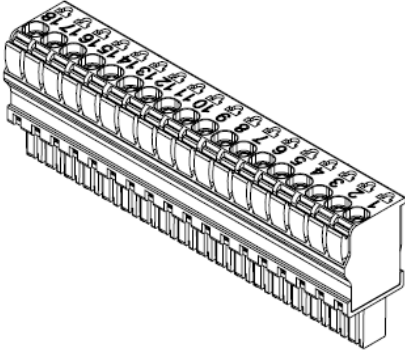
## Electrical connections

### Spring connector

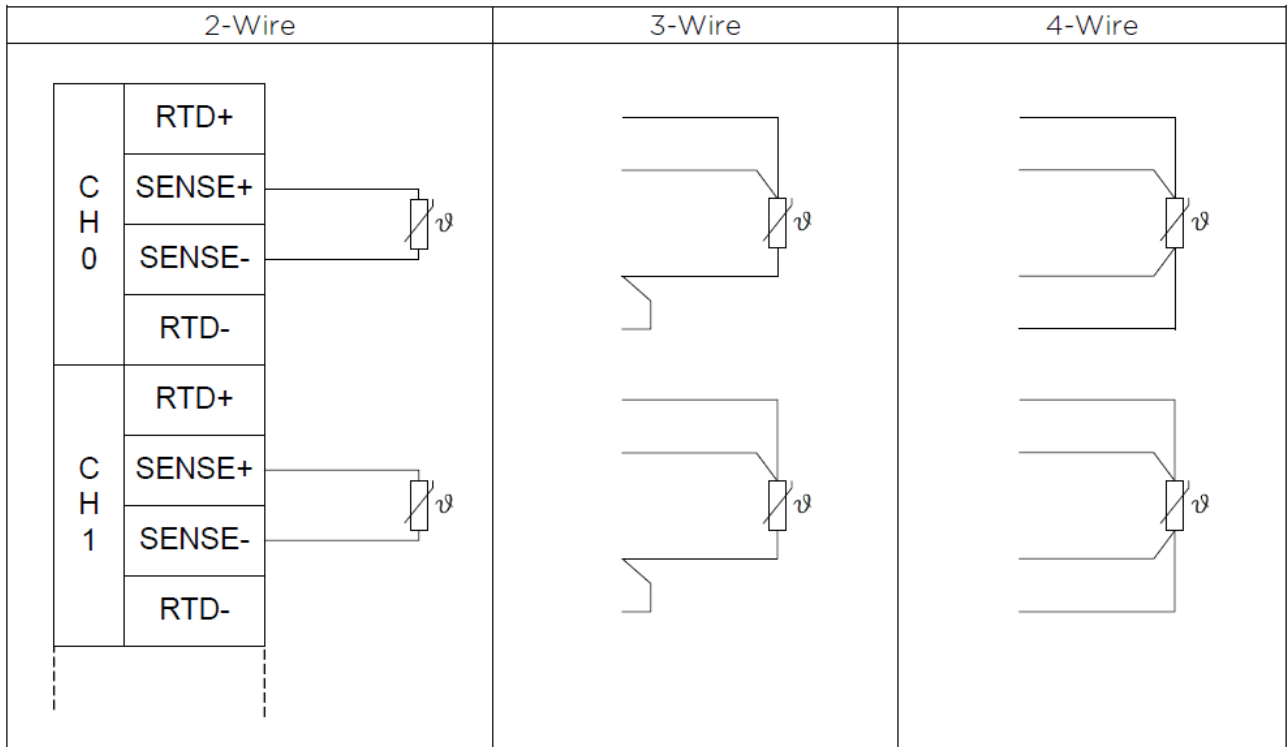




# CREW Manual

EW600E04N00 - I/O RESISTANCE THERMOMETER 4IN Pt100 / Pt1000 (software selectable)			
	1	CH0	RTD+
	2		SENSE+
	3		SENSE-
	4		RTD-
	5	CH1	RTD+
	6		SENSE+
	7		SENSE-
	8		RTD-
	9	CH2	RTD+
	10		SENSE+
	11		SENSE-
	12		RTD-
	13	CH3	RTD+
	14		SENSE+
	15		SENSE-
	16		RTD-
	17	Not Connected	
	18	Not Connected	

# CREW Manual



CREW Manual

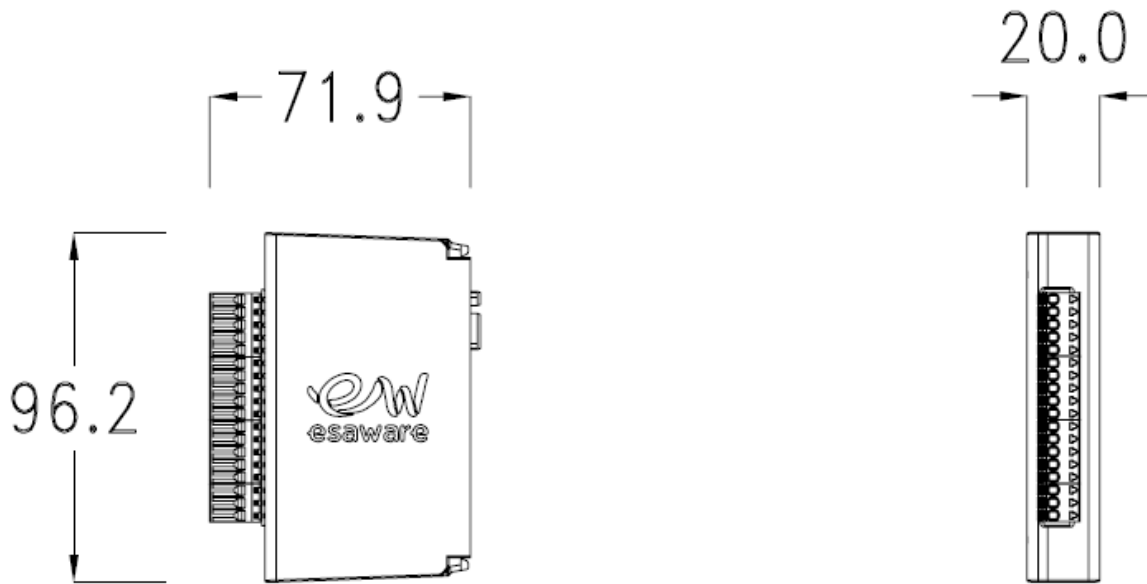
EW600N00Cxx



EW INPUTS / OUTPUTS

# CREW Manual

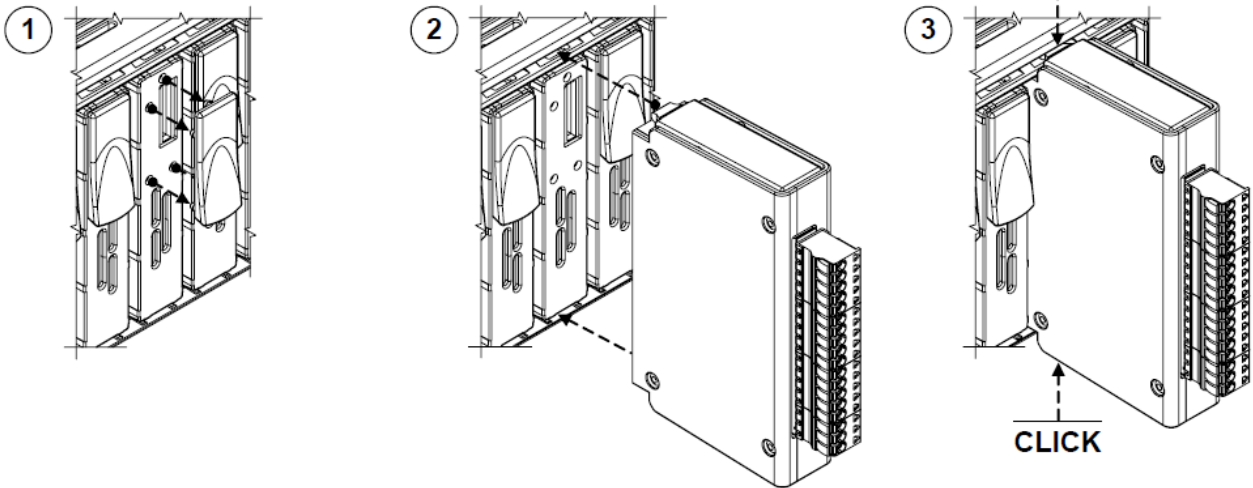
## Dimensions



# CREW Manual

## EW600 assembly on EW terminal

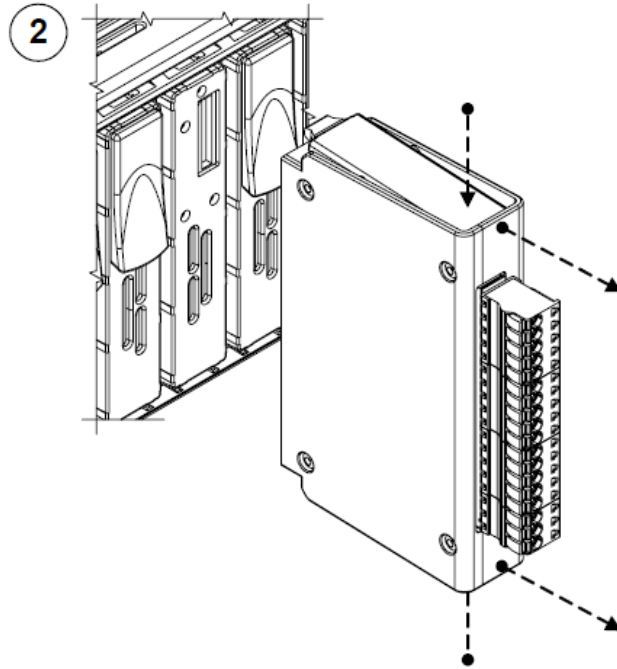
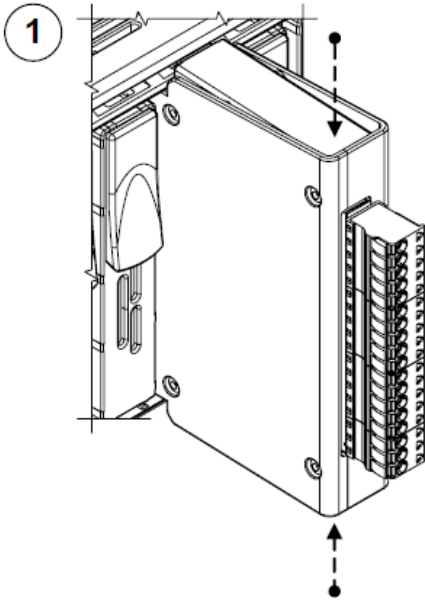
1. Switch off EW.



# CREW Manual

## EW600 removal from EW terminal

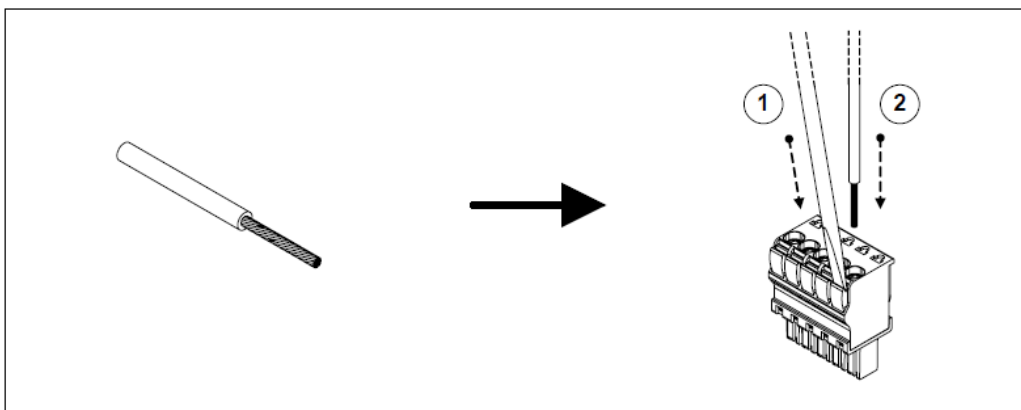
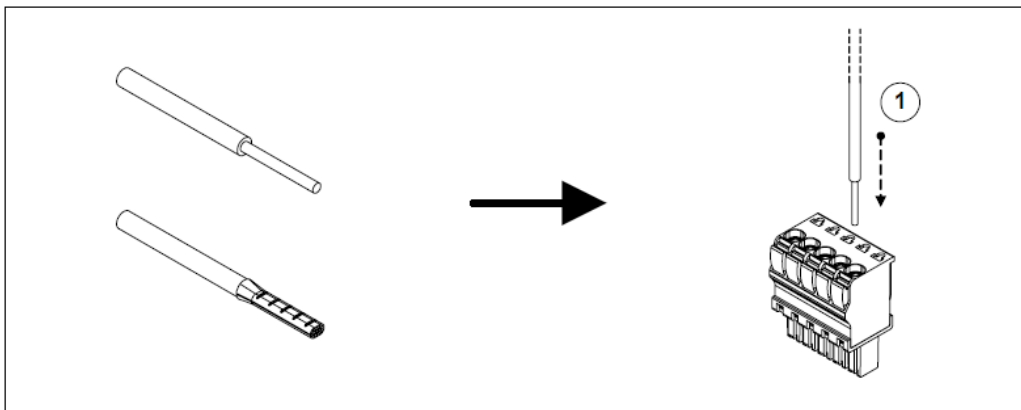
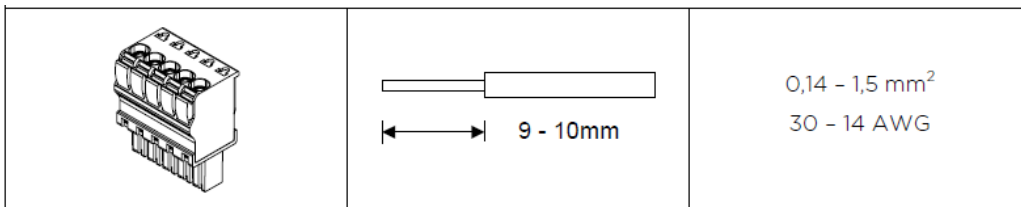
1. Switch off EW.



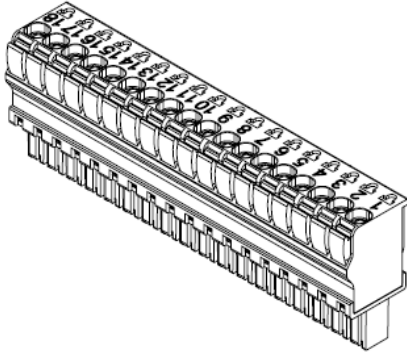
# CREW Manual

## Electrical connections

### Spring connector



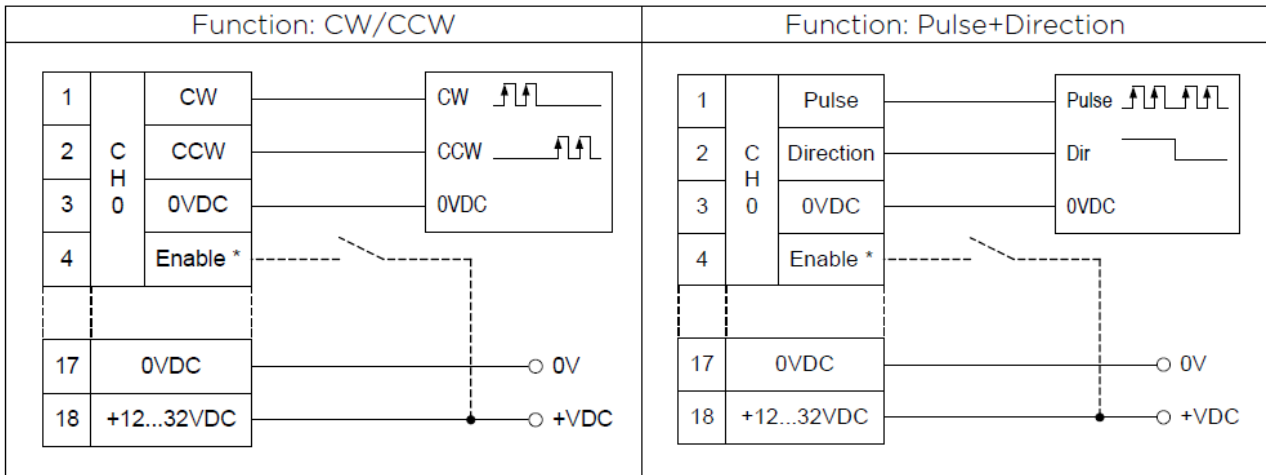
# CREW Manual

EW600N00C04 - I/O HIGH SPEED 4OUT - Push/Pull - Opto-isolated Frequency range: 200Hz...300KHz - Output current range: 5mA...10mA					
Pin	Function (software selectable)		I/O	CH	
	CW/CCW	Pulse+Direction			
1	CW	Pulse	O	0	
2	CCW	Direction	O		
3	OVDC	OVDC	-		
4	Enable *	Enable *	I	1	
5	CW	Pulse	O		
6	CCW	Direction	O		
7	OVDC	OVDC	-	2	
8	Enable *	Enable *	I		
9	CW	Pulse	O		
10	CCW	Direction	O	3	
11	OVDC	OVDC	-		
12	Enable *	Enable *	I		
13	CW	Pulse	O		
14	CCW	Direction	O		
15	OVDC	OVDC	-		
16	Enable *	Enable *	I		
17	OVDC				
18	+12...32VDC				

\* Software configurable



# CREW Manual



\* Connect only if the signal is configured by software

CREW Manual

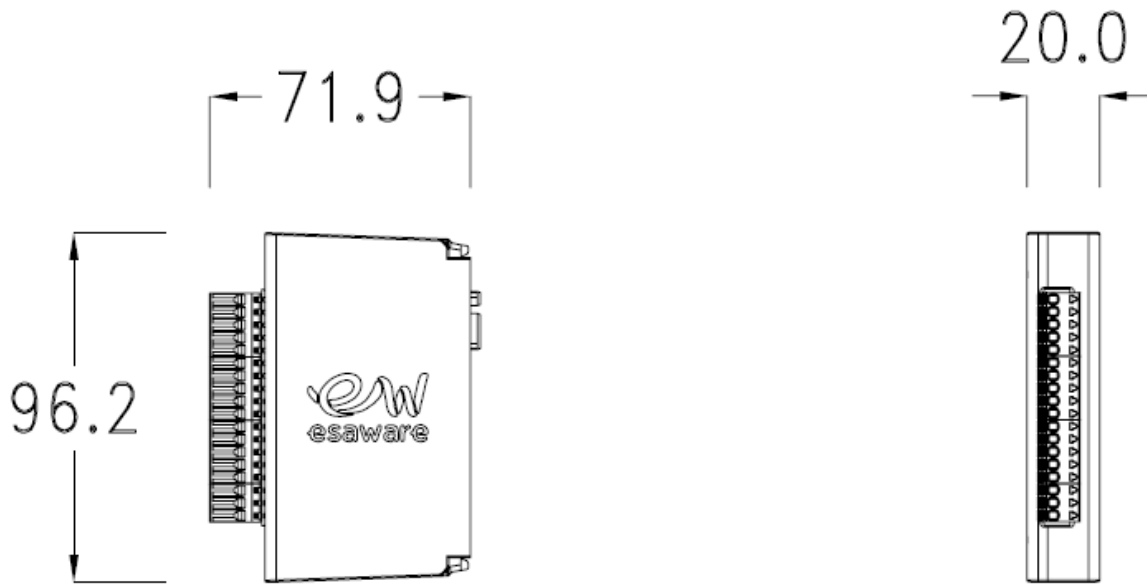
EW600N00Exx



EW INPUTS / OUTPUTS

# CREW Manual

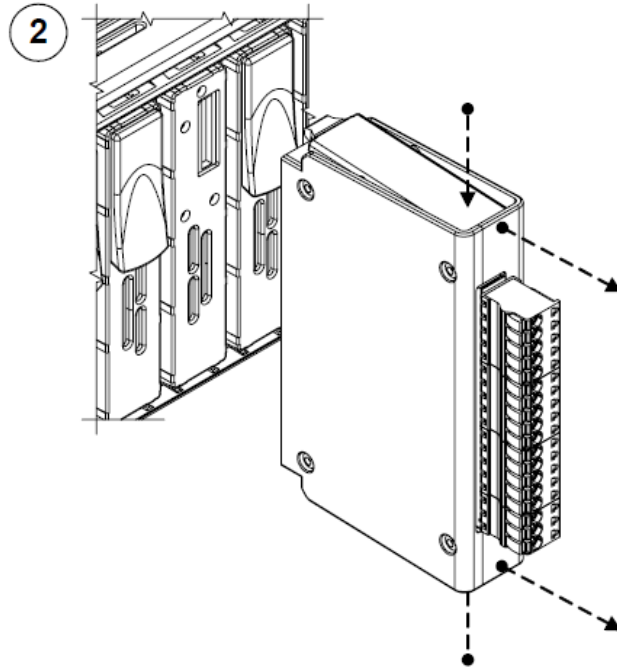
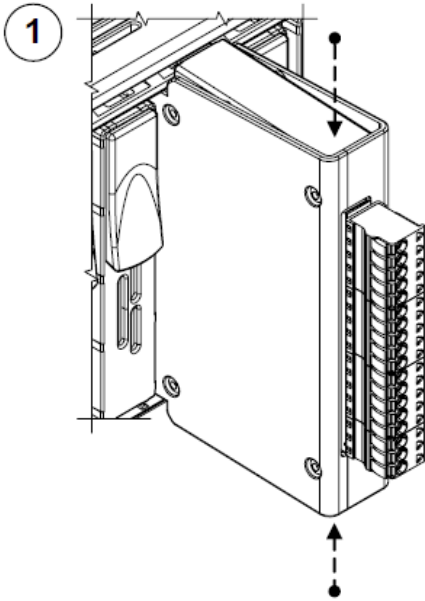
## Dimensions



# CREW Manual

## EW600 assembly on EW terminal

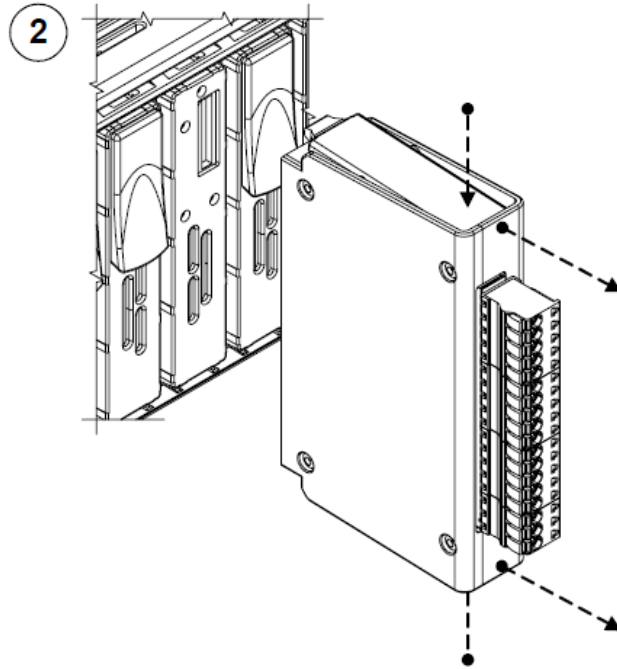
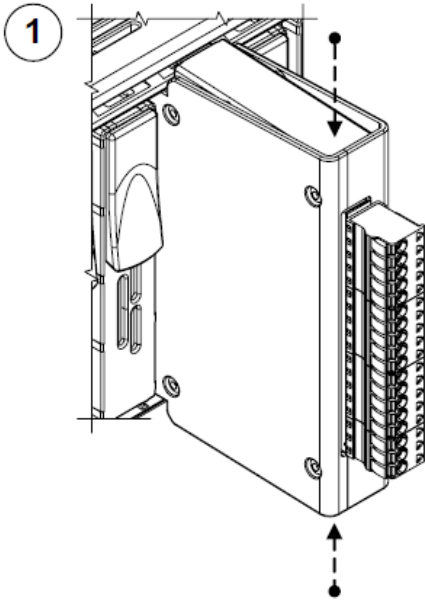
1. Switch off EW.



# CREW Manual

## EW600 removal from EW terminal

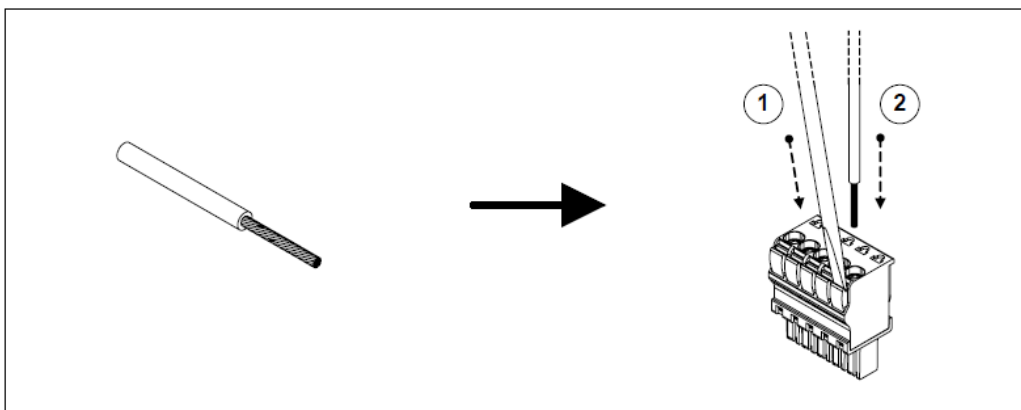
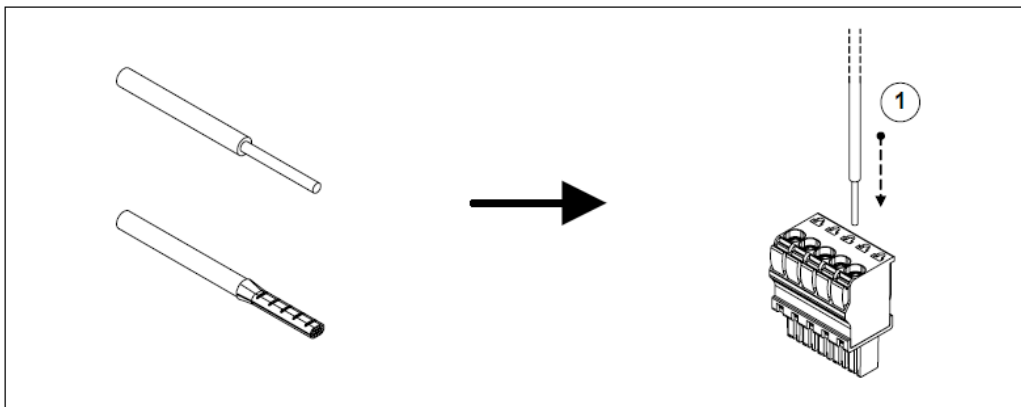
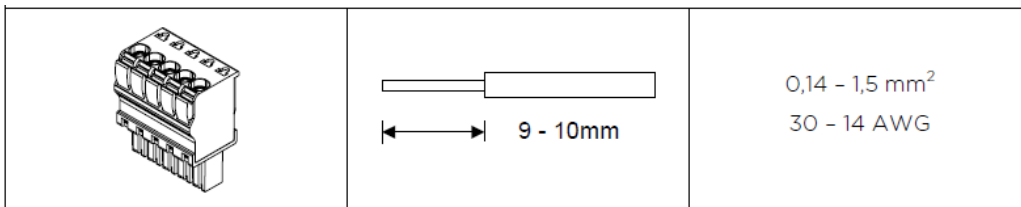
1. Switch off EW.



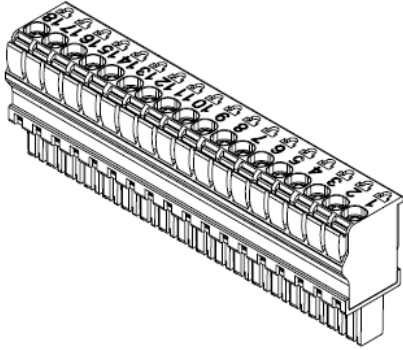
# CREW Manual

## Electrical connections

### Spring connector

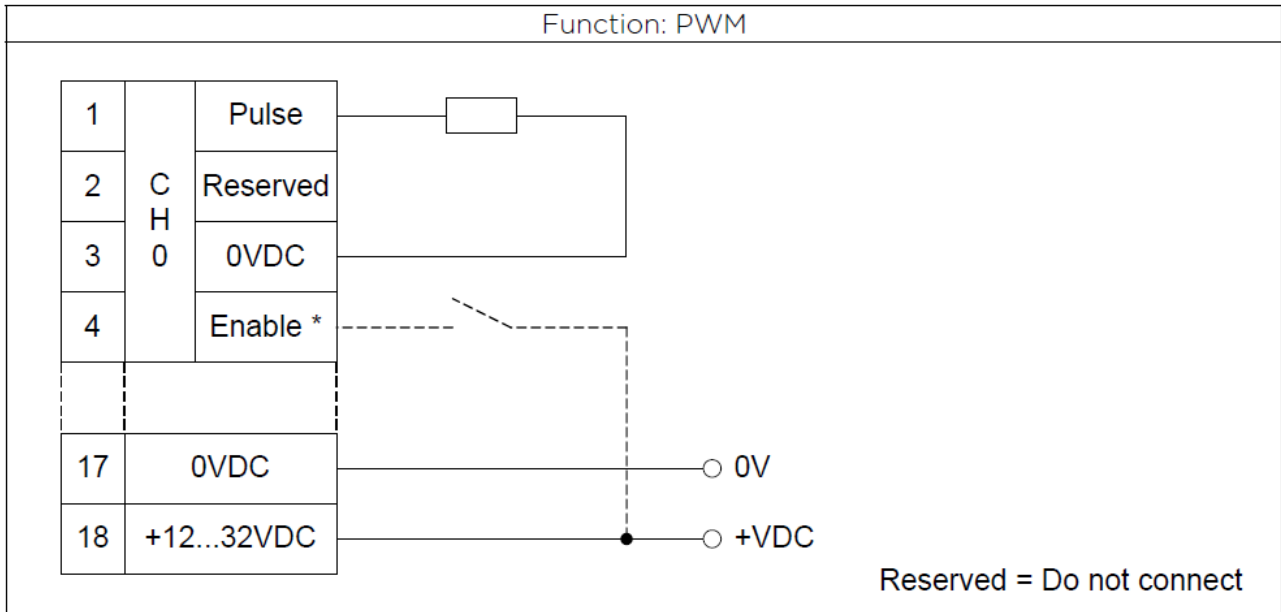


# CREW Manual

EW600N00E04 - I/O PWM 4OUT - Push/Pull - Opto-isolated Frequency range: 200Hz...300KHz - Output current range: 5mA...10mA				
Pin	Function	I/O	CH	
	PWM			
1	Pulse	O	0	
2	Reserved / Do NOT connect	-		
3	OVDC	-		
4	Enable *	I	1	
5	Pulse	O		
6	Reserved / Do NOT connect	-		
7	OVDC	-	2	
8	Enable *	I		
9	Pulse	O		
10	Reserved / Do NOT connect	-	3	
11	OVDC	-		
12	Enable *	I		
13	Pulse	O		
14	Reserved / Do NOT connect	-		
15	OVDC	-		
16	Enable *	I		
17	OVDC			
18	+12...32VDC			

\* Software configurable

# CREW Manual



\* Connect only if the signal is configured by software



## CREW Manual

# Service Page



To access the service page include a button in the project (exit runtime).

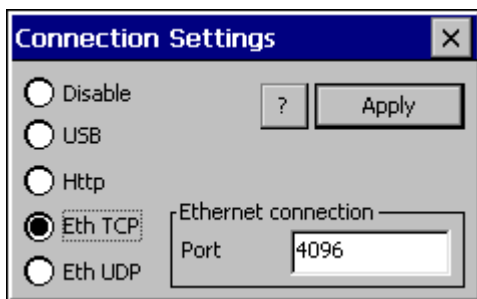
Alternatively it is possible to access the service page by enabling Runtime exit from the terminal by pressing the two corners on the top right and on the bottom left (or top left and bottom right) of any page in the project (see [“Panel”](#) section). This method is helpful when, for example, a specific button as not yet been provided in the project.

# CREW Manual

The service page contains the following options:

- Start ESA Application: this runs the project runtime
- Downloader Configuration: this opens the download configuration.
- Control Panel: this opens the control panel (see "[Control Panel](#)" section).

Click "Downloader Configuration" to configure the connection settings:

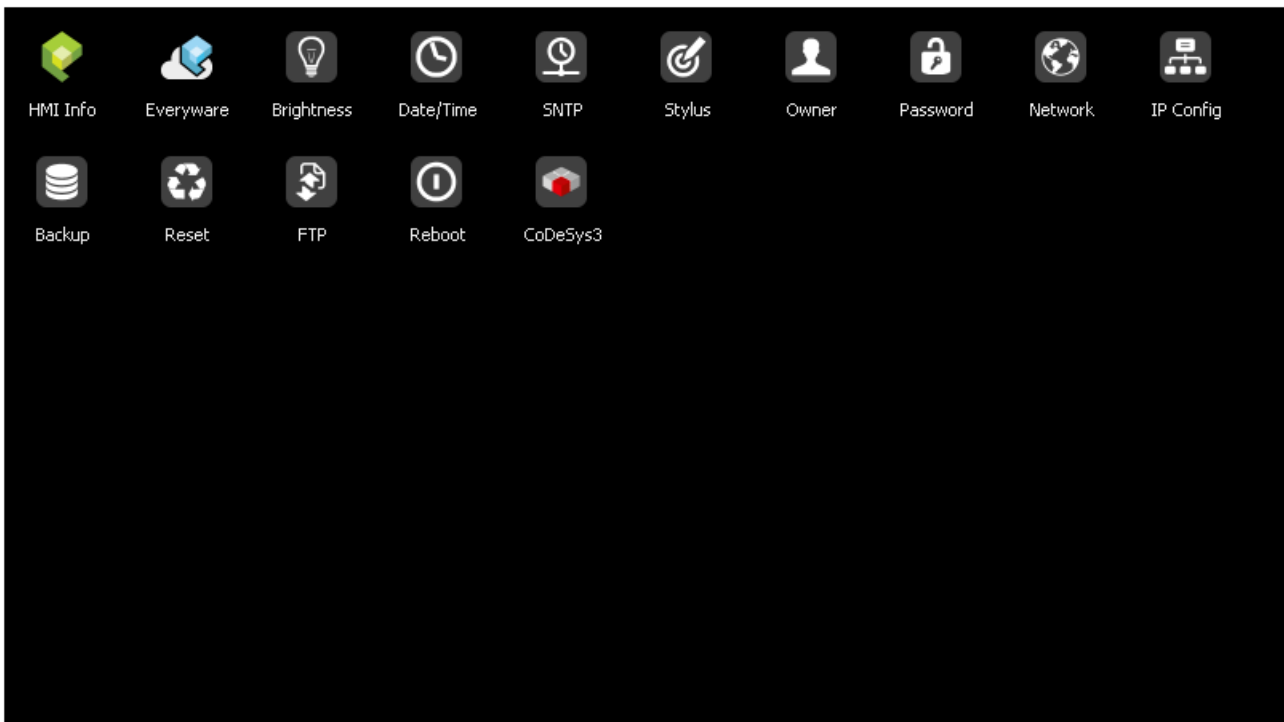


- Disable: this disables the connection with the terminal.
- USB: this enables the USB connection with the terminal.
- http: this enables the ethernet connection with the terminal through an http protocol.
- Eth TCP: this enables the ethernet connection with the terminal and allows configuring the port (slower but more precise).
- Eth UDP: this enables the ethernet connection with the terminal and allows configuring the port (faster but less precise).

For more information on the project "Download", refer to the "[Project download](#)" section.

# CREW Manual

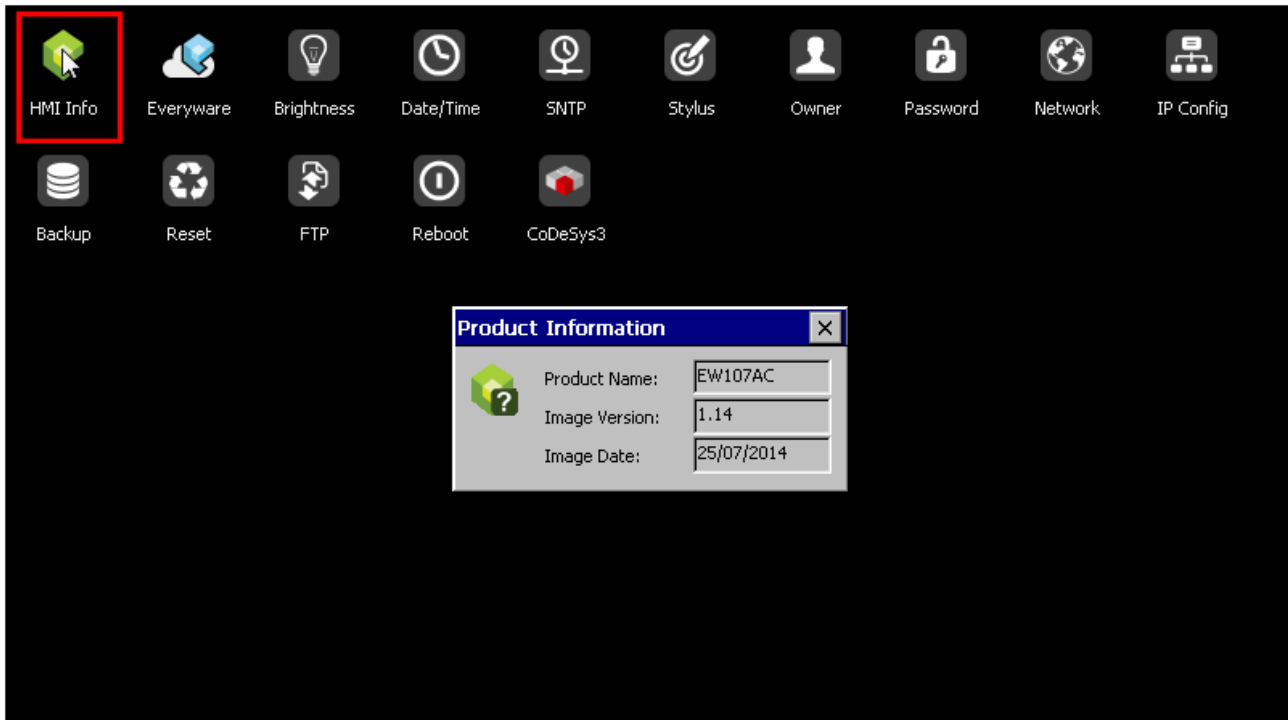
# Control Panel



Click on each of these icons to access terminal configuration.

# CREW Manual

HMI Info:

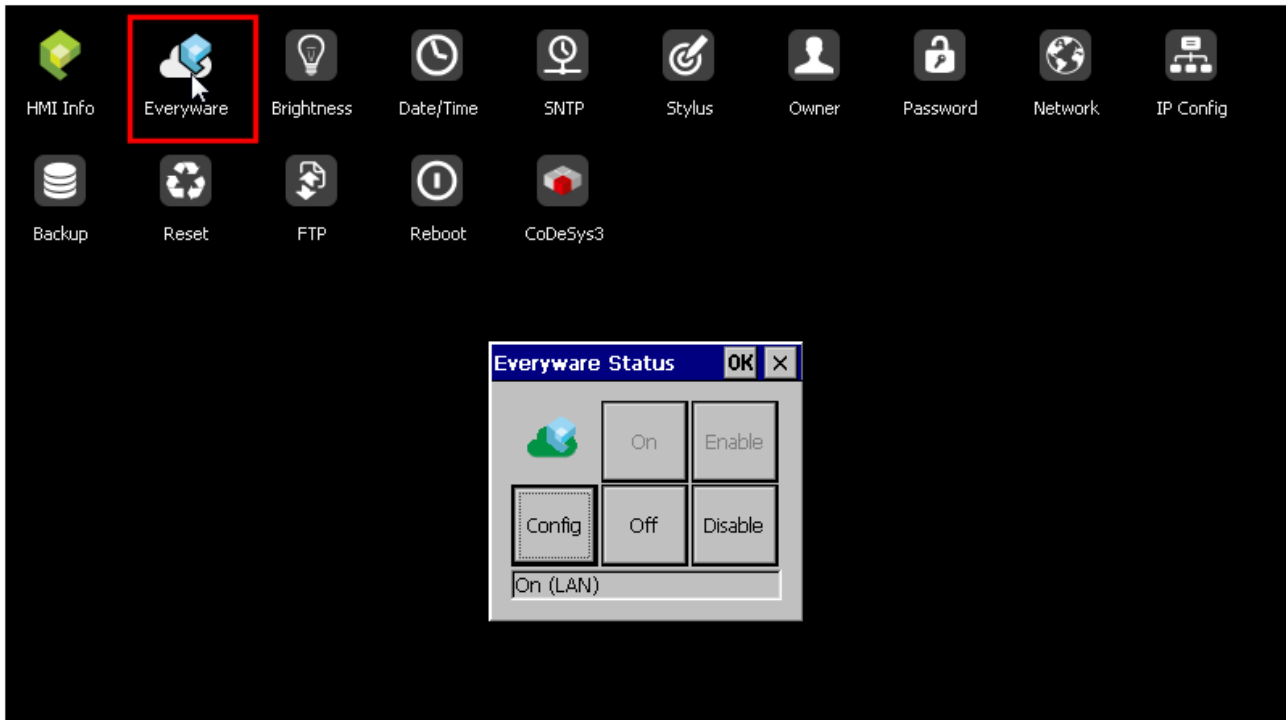


Panel information is displayed, such as: terminal model, revision of the Windows CE image and the image date.

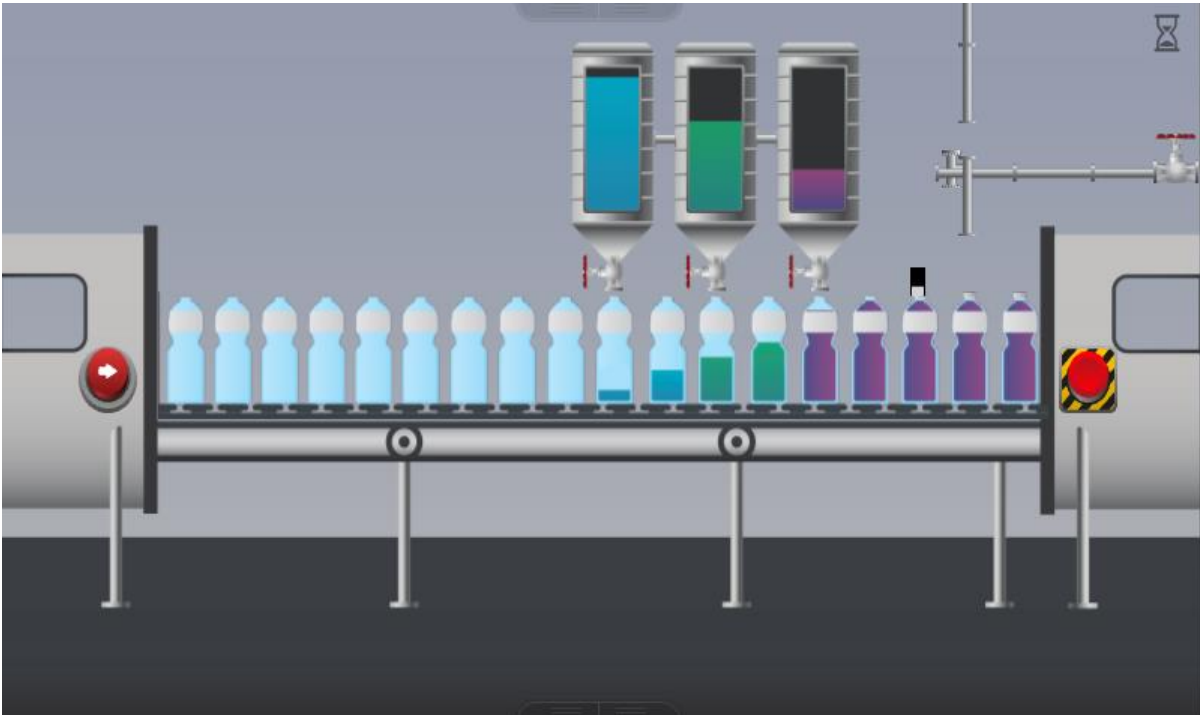
# CREW Manual

Everyware:

Activate the Everyware application on the terminal (see "[Everyware](#)" section).




# CREW Manual



**DataLog View**

Time	Data1	Data2	Data3	Data4
04:35:09 19-02-2015	14	36	63	77
04:35:10 19-02-2015	38	89	41	9
04:35:11 19-02-2015	38	89	41	9
04:35:12 19-02-2015	28	22	62	96
04:35:13 19-02-2015	64	59	68	48
04:35:14 19-02-2015	64	59	68	48
04:35:15 19-02-2015	42	36	86	12
04:35:16 19-02-2015	55	43	41	0
04:35:17 19-02-2015	95	62	95	10
04:35:18 19-02-2015	95	62	95	10
04:35:19 19-02-2015	59	34	44	98
04:35:20 19-02-2015	15	37	95	17
04:35:21 19-02-2015	15	37	95	17
04:35:22 19-02-2015	59	84	27	87
04:35:23 19-02-2015	88	18	92	95
04:35:24 19-02-2015	77	34	51	33
04:35:25 19-02-2015	77	34	51	33



**Data 1**  
83

**Data 2**  
44

**Data 3**  
49

**Data 4**  
12



# CREW Manual



**DataLog View**

Time	Data1	Data2	Data3	Data4
04:37:25 19-02-2015	69	33	28	3
04:37:26 19-02-2015	69	33	28	3
04:37:27 19-02-2015	36	45	1	81
04:37:28 19-02-2015	33	1	74	95
04:37:29 19-02-2015	33	1	74	95
04:37:30 19-02-2015	34	56	45	54
04:37:31 19-02-2015	55	86	6	29
04:37:32 19-02-2015	55	86	6	29
04:37:33 19-02-2015	48	79	70	37
04:37:34 19-02-2015	26	40	80	81
04:37:35 19-02-2015	50	99	24	71
04:37:36 19-02-2015	50	99	24	71
04:37:37 19-02-2015	64	19	7	97
04:37:38 19-02-2015	50	86	77	2
04:37:39 19-02-2015	50	86	77	2
04:37:40 19-02-2015	82	19	78	63
04:37:41 19-02-2015	66	58	97	78

Data 1: 35  
 Data 2: 29  
 Data 3: 69  
 Data 4: 53



**DEFAULT PASSWORD: "blueocean"**

**DataLog view**

Time	Data1	Data2	Data3	Data4
04:39:21 19-02-2015	68	66	90	60
04:39:22 19-02-2015	99		69	8
04:39:23 19-02-2015	99			8
04:39:24 19-02-2015	35			94
04:39:25 19-02-2015	42			10
04:39:26 19-02-2015	42			10

Data 1: 44  
 Data 2: 66

Quit Runtime

\*\*\*\*\*

Cancel

1 2 3 4 5 6 7 8 9 0

TAB - + / \* ( [ ) { } % ^ & #

ALT < > . : , ; ? ! ' " = \ \$ £ HOME END

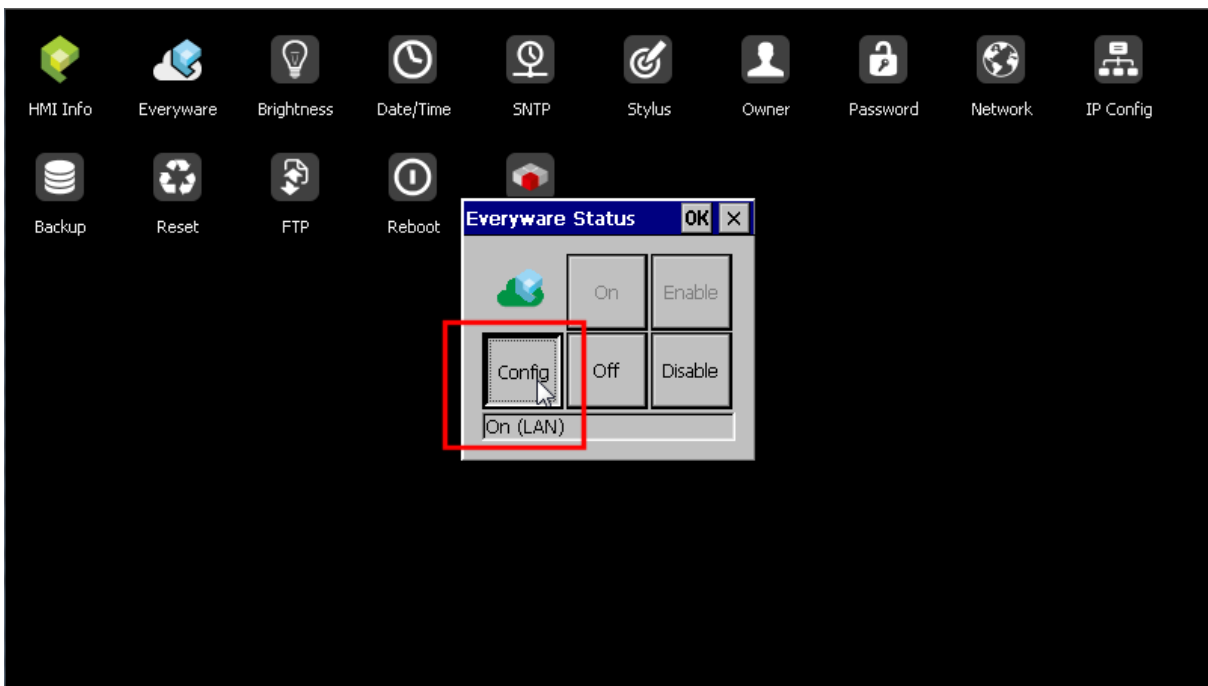
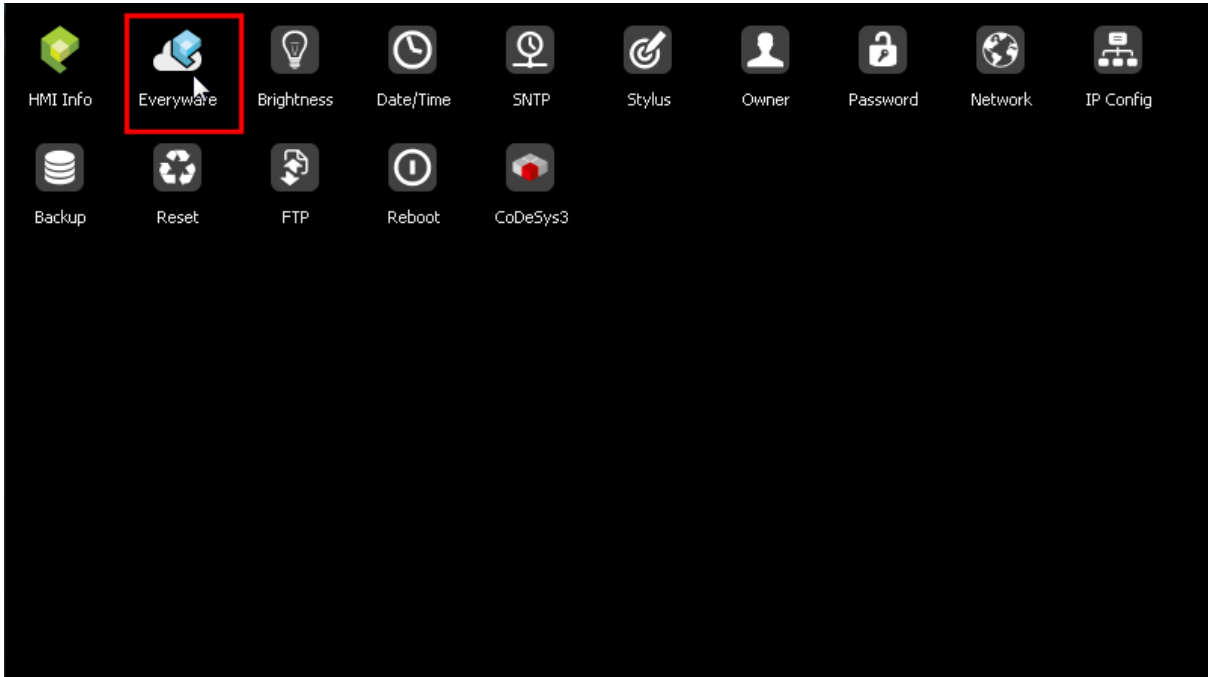
ABC [ ] DEL

# CREW Manual

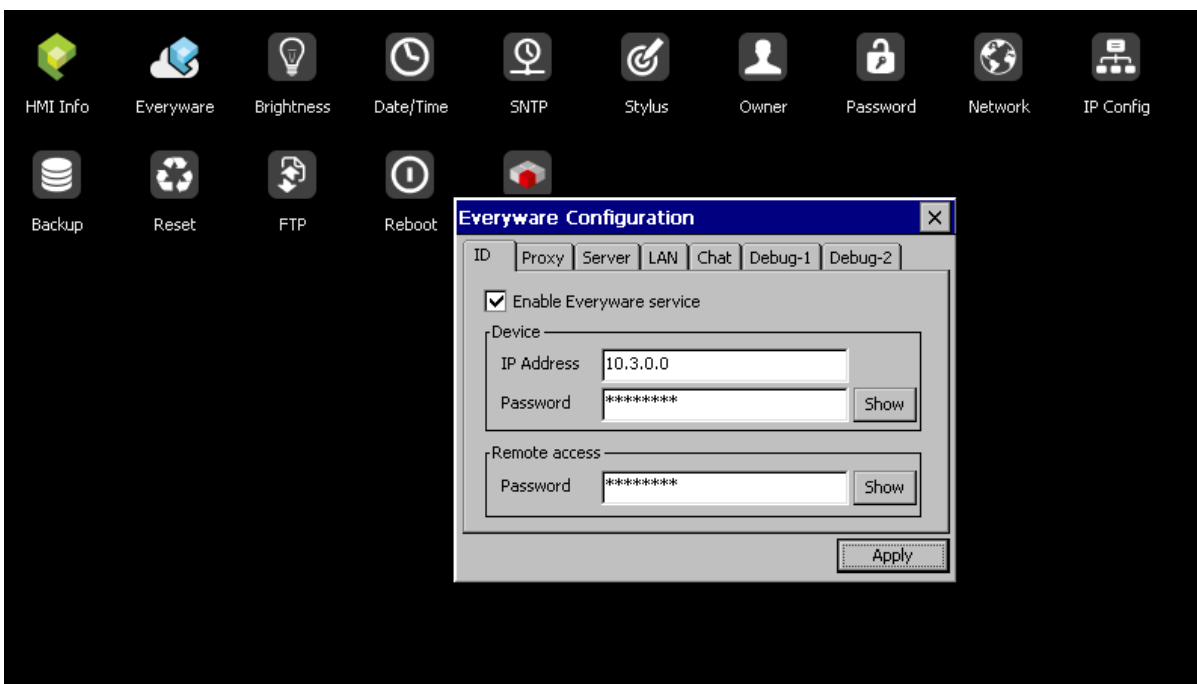




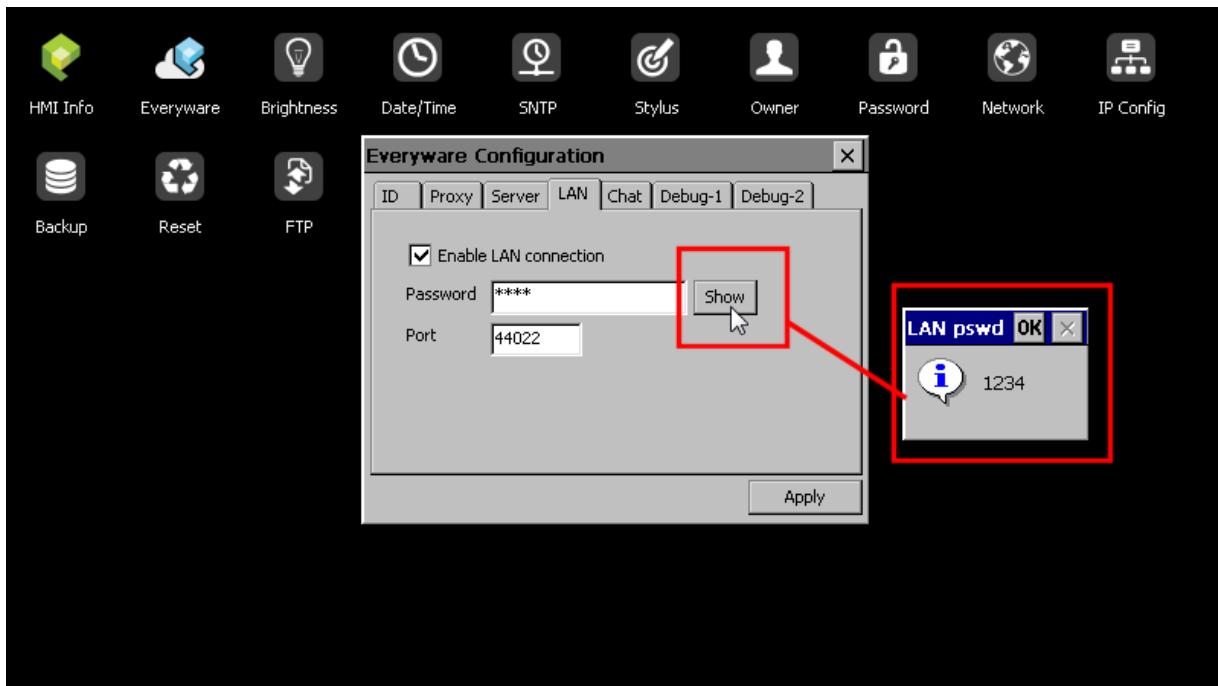
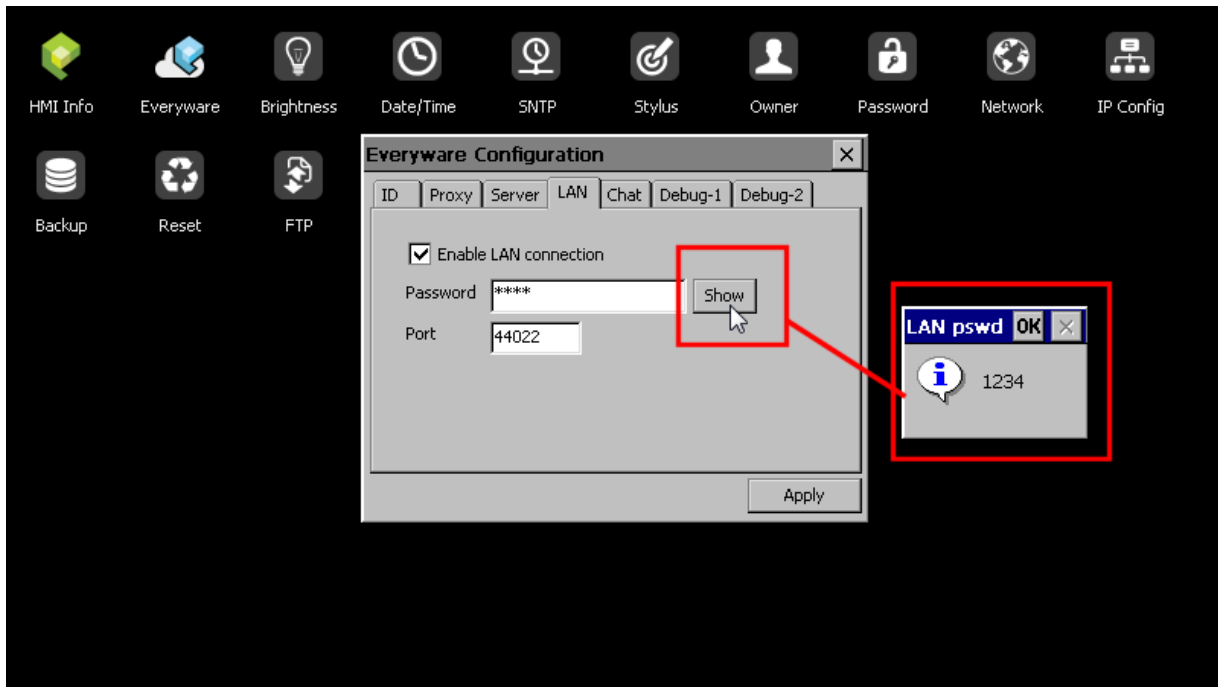
# CREW Manual



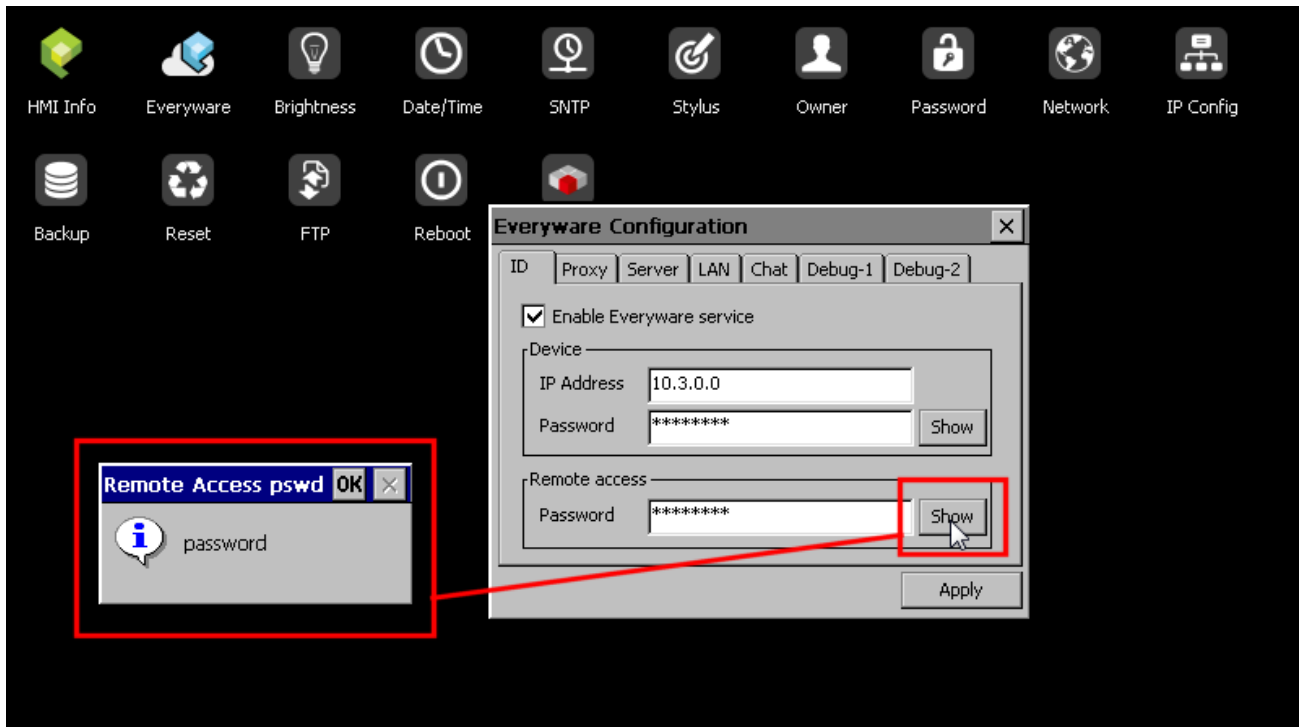
# CREW Manual



# CREW Manual

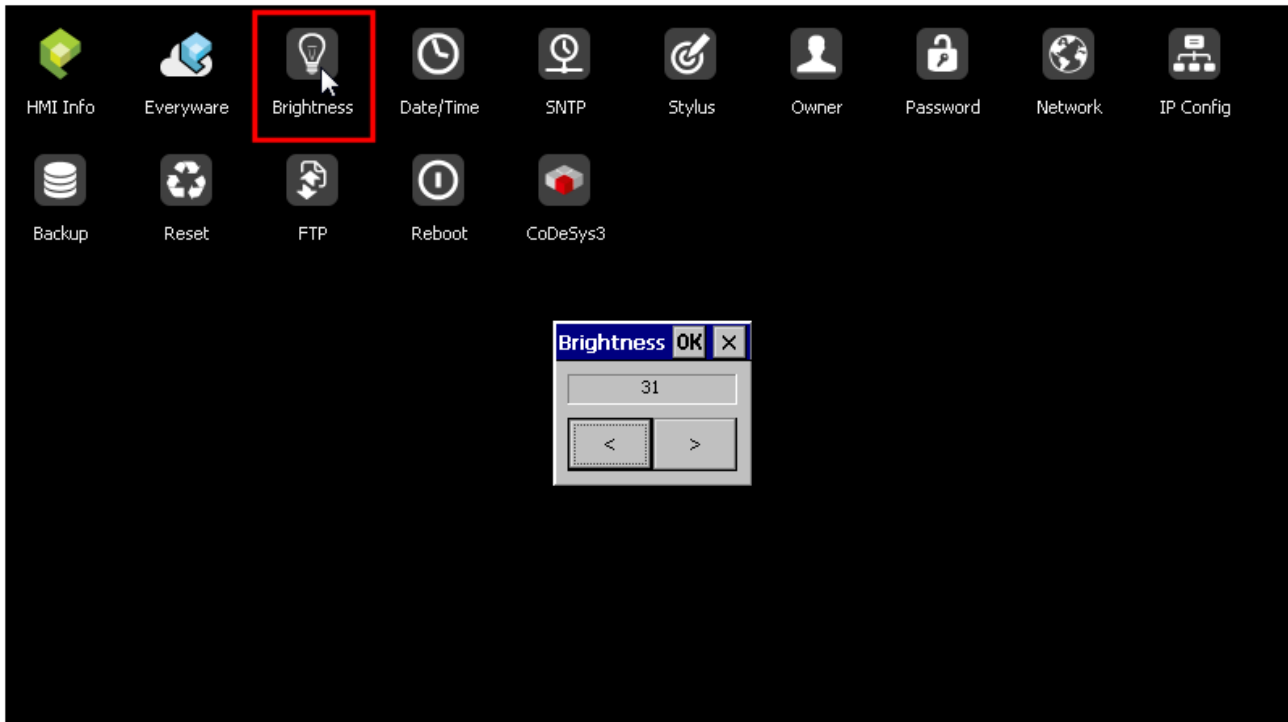


# CREW Manual



# CREW Manual

Brightness:



Use the relative keys to configure the brightness of the screen.

# CREW Manual

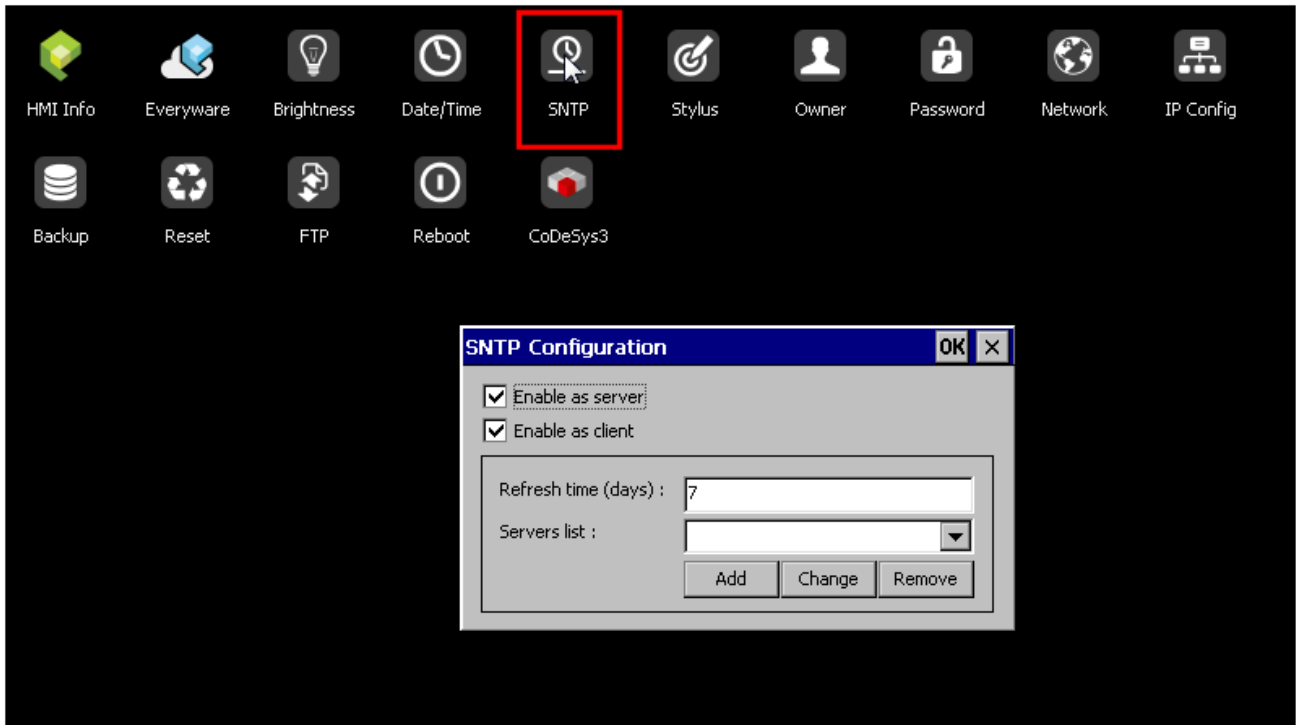
## Date / Time:



This changes date, hour and time zone. By enabling the “automatically adjust clock for daylight saving” check, the time automatically updates to BST or GMT.

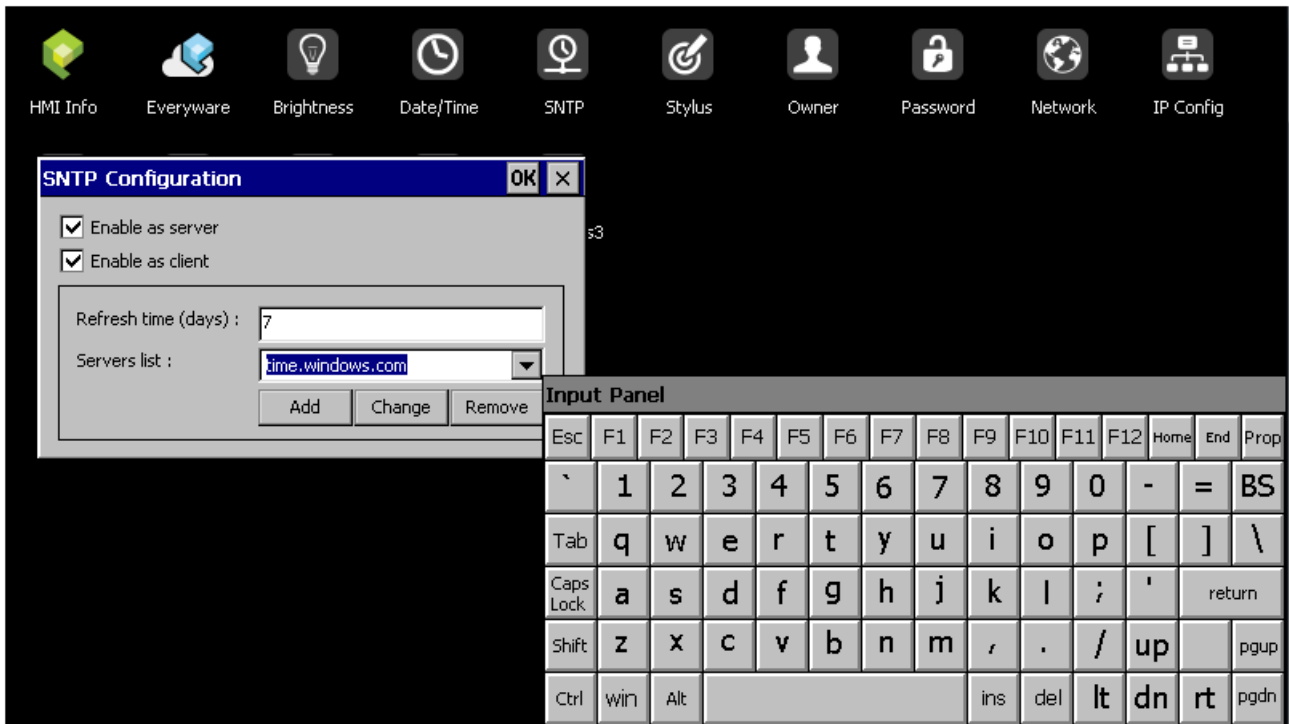
# CREW Manual

SNTP:



This updates date and time by extracting it from a remote server.

# CREW Manual





# CREW Manual

Stylus:



This is used to repeat, if necessary, the touch screen calibration procedure.

# CREW Manual

Carefully press and briefly hold stylus on the center of the target.  
Repeat as the target moves around the screen.  
Press the Esc key to cancel.



Carefully press and briefly hold stylus on the center of the target.  
Repeat as the target moves around the screen.  
Press the Esc key to cancel.



# CREW Manual

Carefully press and briefly hold stylus on the center of the target.  
Repeat as the target moves around the screen.  
Press the Esc key to cancel.



Carefully press and briefly hold stylus on the center of the target.  
Repeat as the target moves around the screen.  
Press the Esc key to cancel.



# CREW Manual



Carefully press and briefly hold stylus on the center of the target.  
Repeat as the target moves around the screen.  
Press the Esc key to cancel.



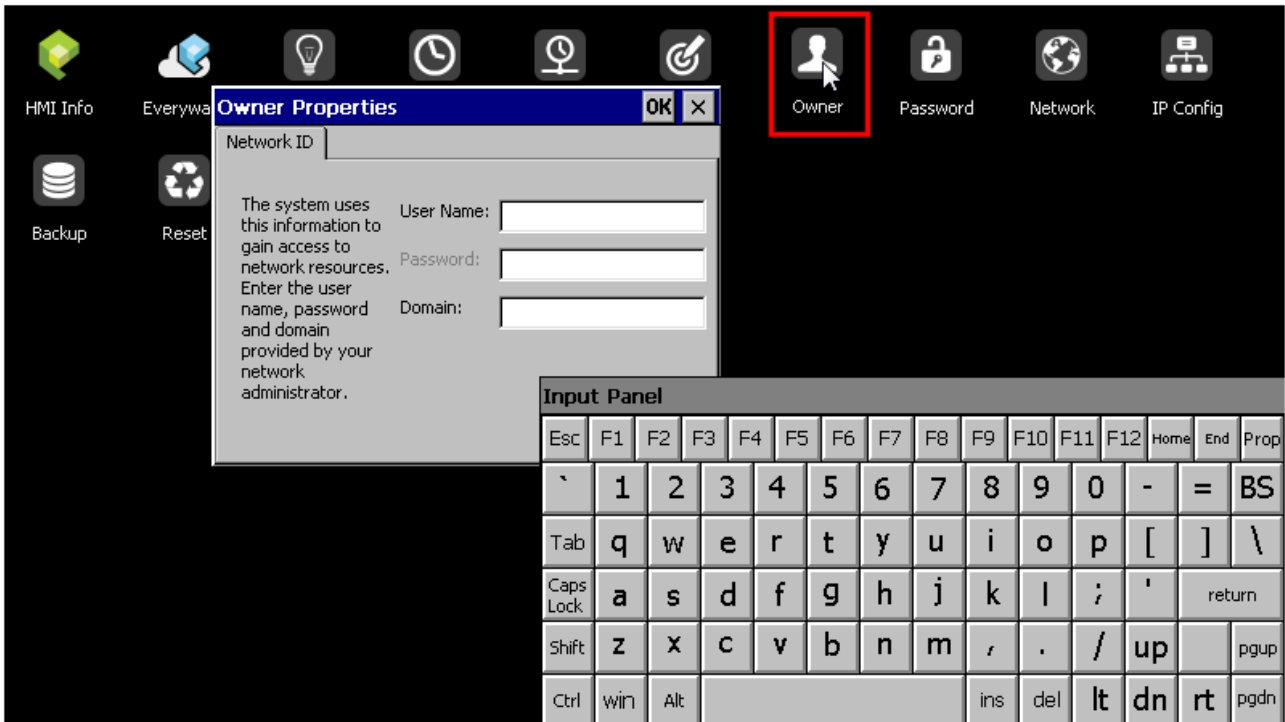
New calibration settings have been measured.  
Press the Enter key to accept the new settings.  
Press the Esc key to keep the old settings.

# CREW Manual



# CREW Manual

Owner:



This information is used by Windows CE to access the network resources.

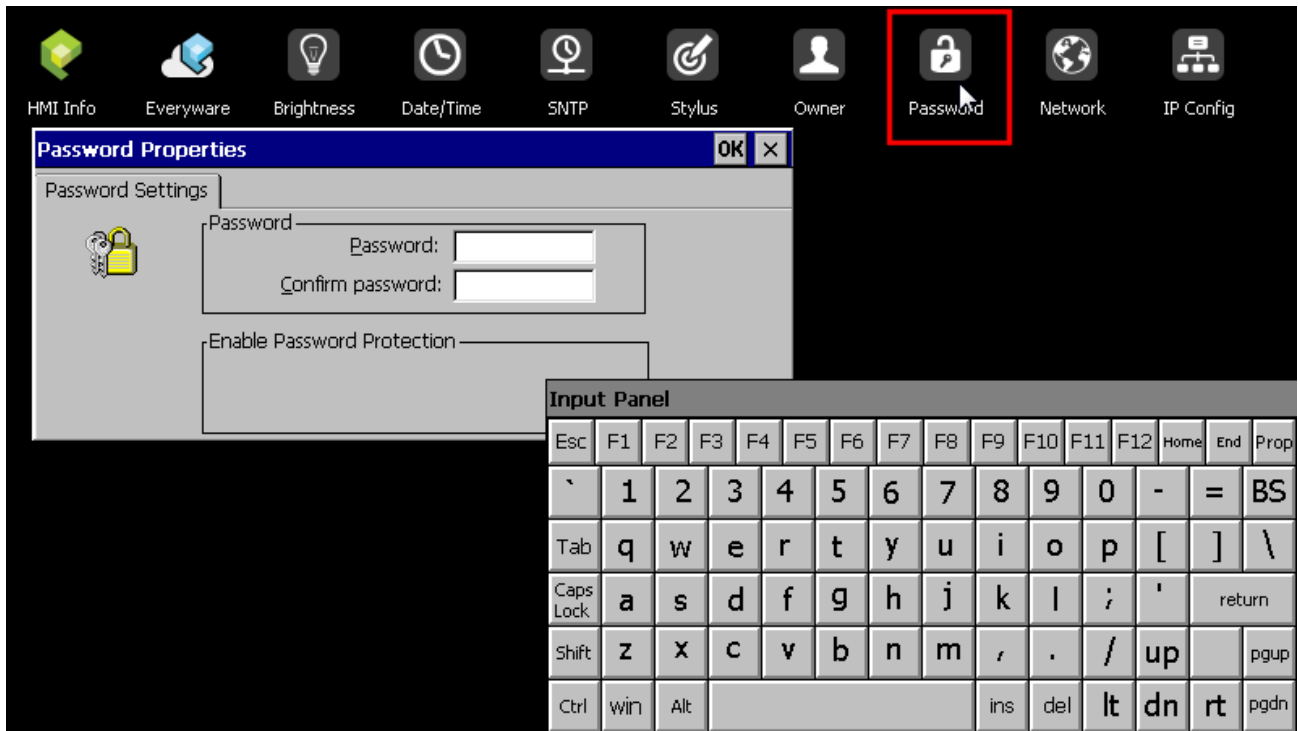
Username: enter the user name to access the network.

Password: enter the password to access the network.

Domain: enter the domain to access the network.

# CREW Manual

Password:



This is used to enter the password to access the terminal. The password is requested, for example, when using the “Remote Desktop” application (see "[Everyware](#)" section, "Remote Desktop" paragraph) or the "Online Tools" (see "[Online Tools](#)" section).



Attention: If you lose the access password assigned to the terminal it is necessary to send the terminal to our headquarters for full restoration.

# CREW Manual

## Network:



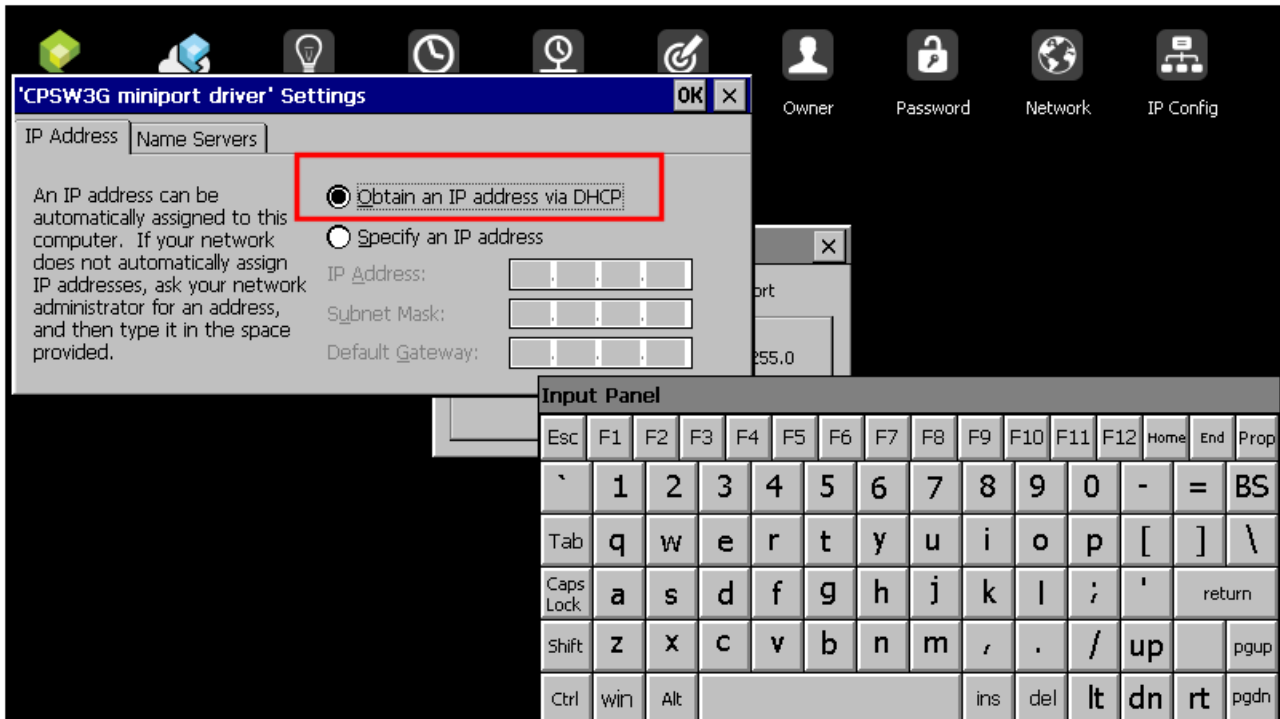
To view and set the IP address on the terminal (see example in the "[Everyware](#)" section, "Network Settings" paragraph).

Select the active Ethernet port ("Ethernet 1" in the example), to open the box where you can assign the terminal's IP address to that Ethernet port. It can be done by choosing from two options:



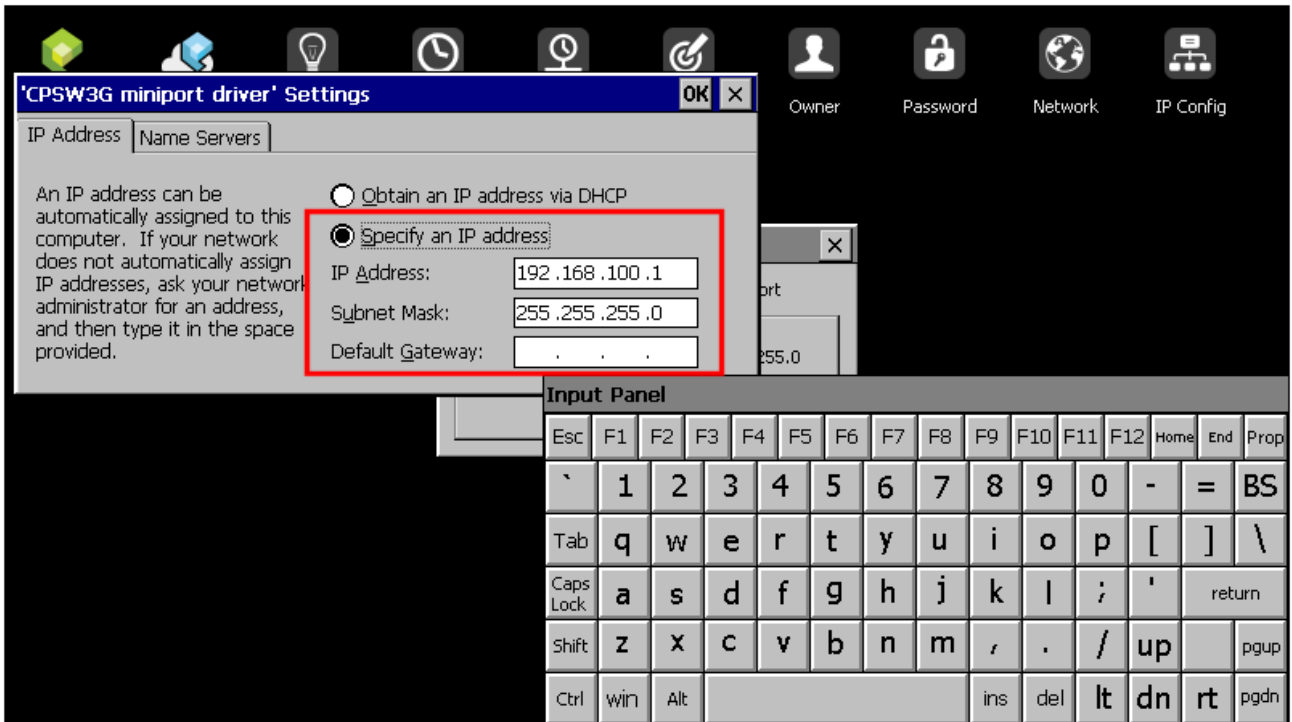
# CREW Manual

1) Obtain an IP address via DHCP: to automatically obtain an IP address (ensure that the DHCP server is enabled on the network).



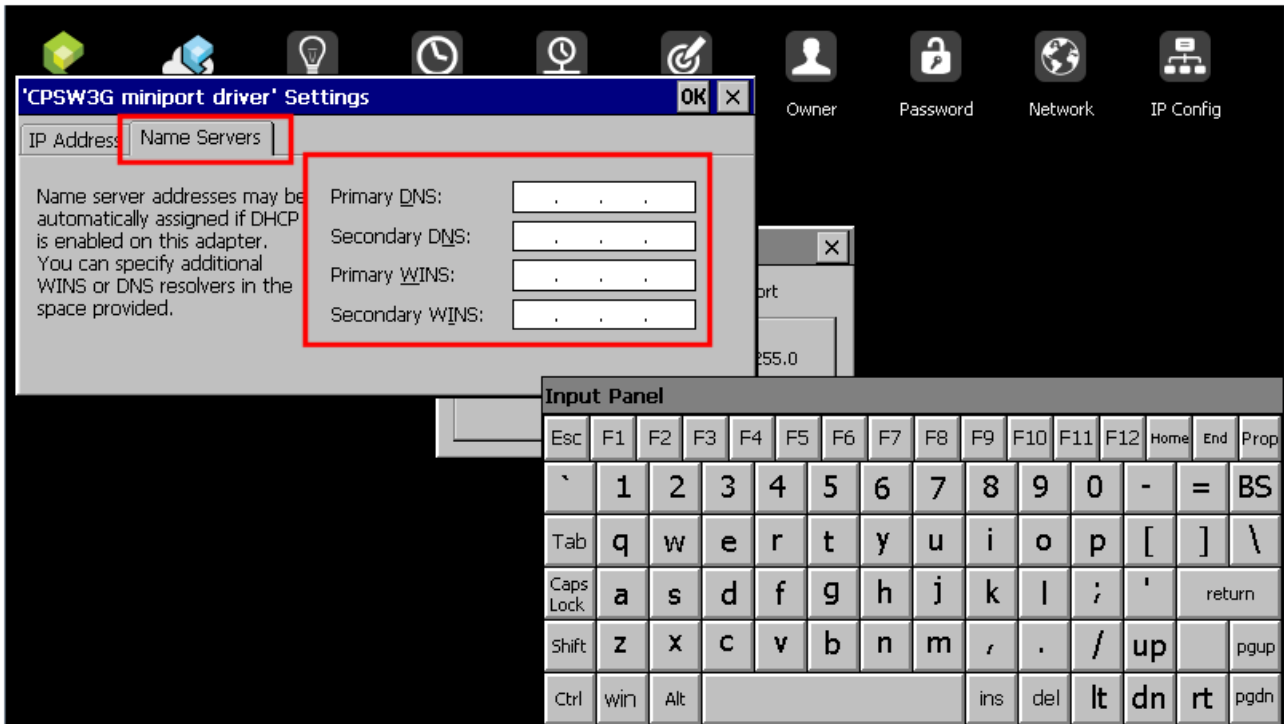
# CREW Manual

- Specify an IP address: it is necessary to enter the parameters manually (IP Address, Subnet Mask, Default Gateway).



# CREW Manual

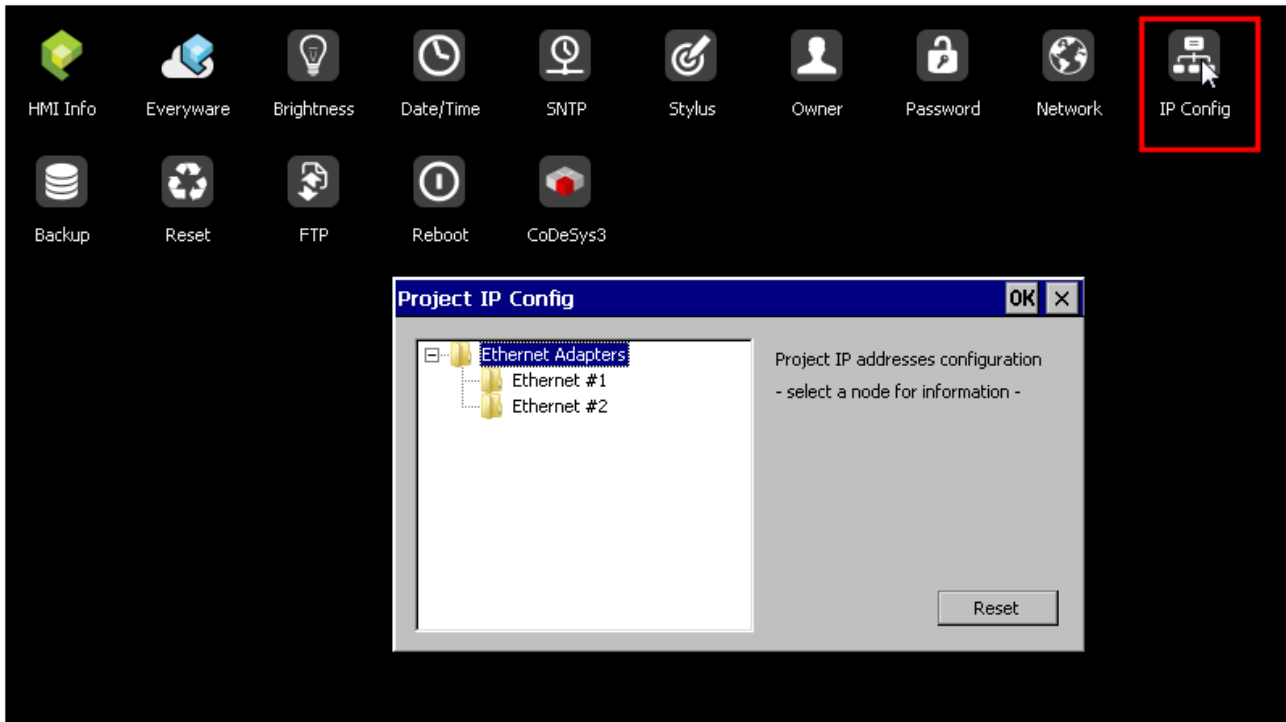
Name Servers: if necessary, the parameters relating to the relative DNS or AL WINS must be entered.



Contact the network administrator if you do not know the aforementioned data.

# CREW Manual

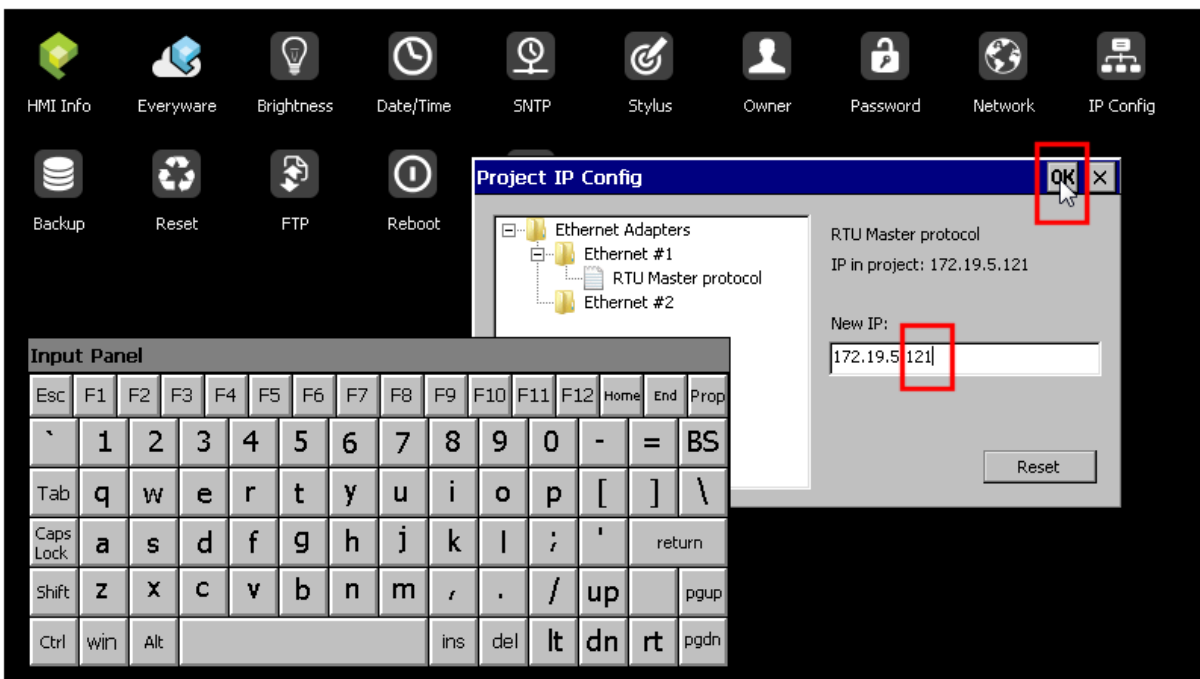
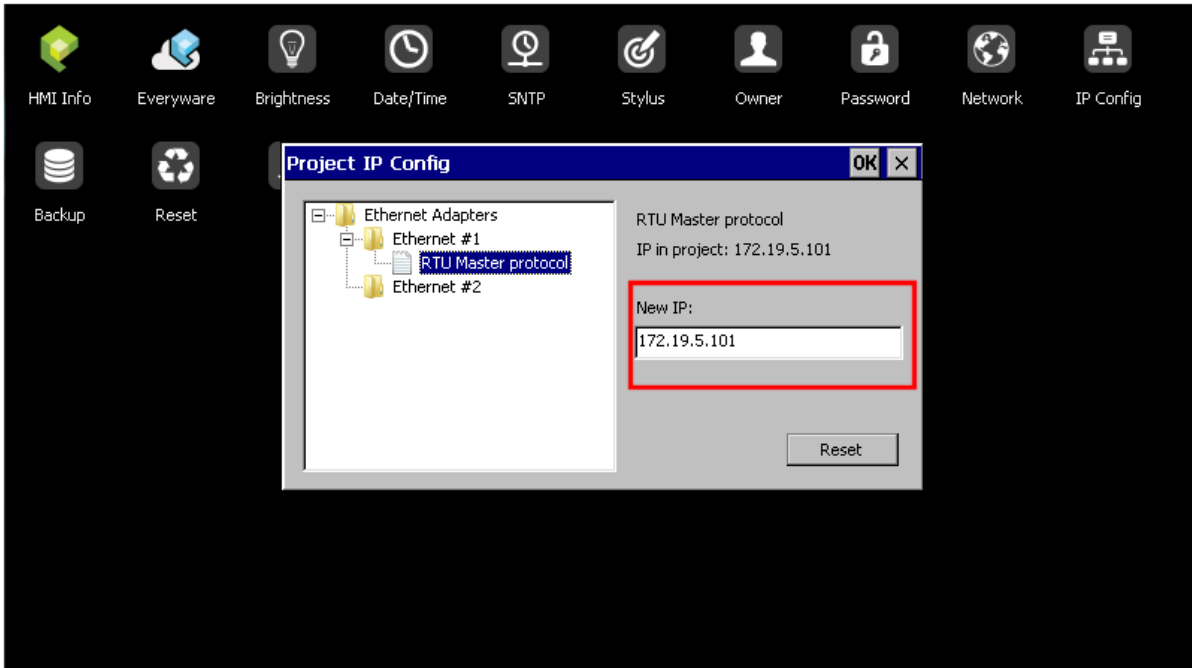
## IP Config:



To change the IP addresses (terminal and PLC) used in the project, without having to change and transfer the project itself.

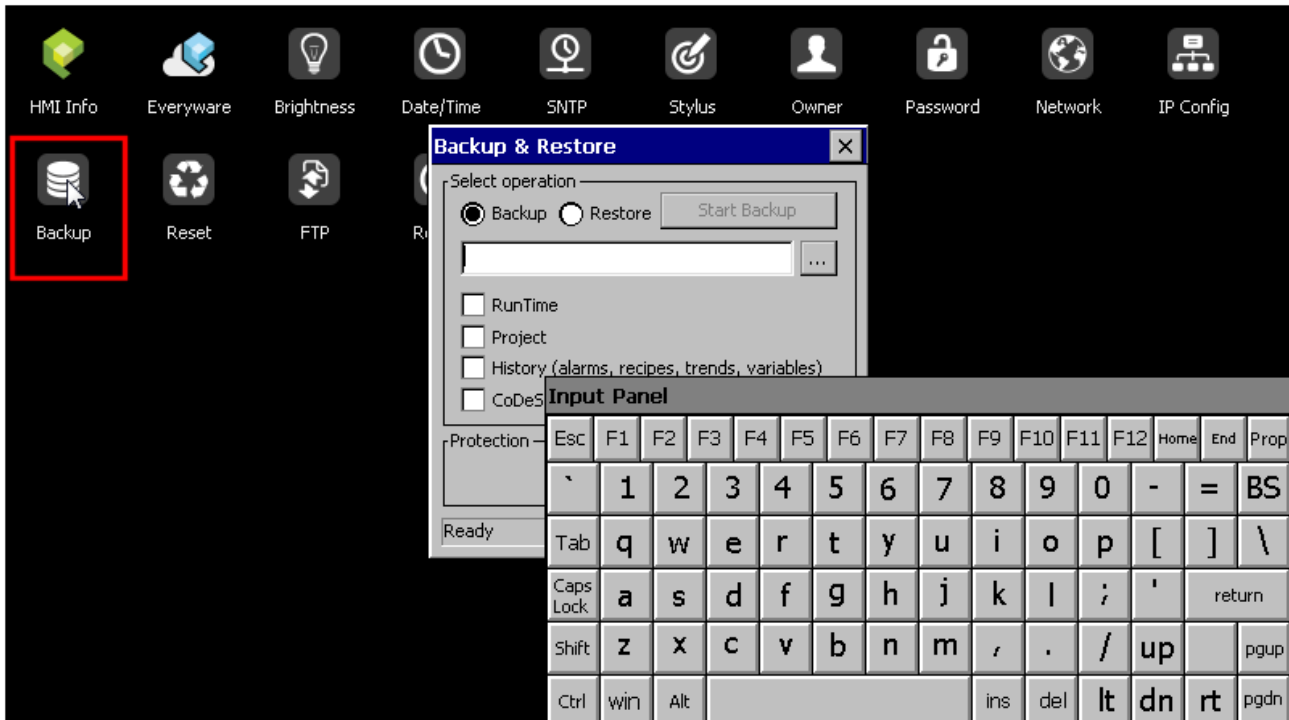
For example, if there is a "RTU Master" PLC entered in the project on the Ethernet 1 port, by clicking on the "IP Config" icon you will note that the IP address that was entered in the project appears on the Ethernet 1 port. This can easily be changed without having to transfer the project to the terminal.

# CREW Manual



# CREW Manual

## Backup:



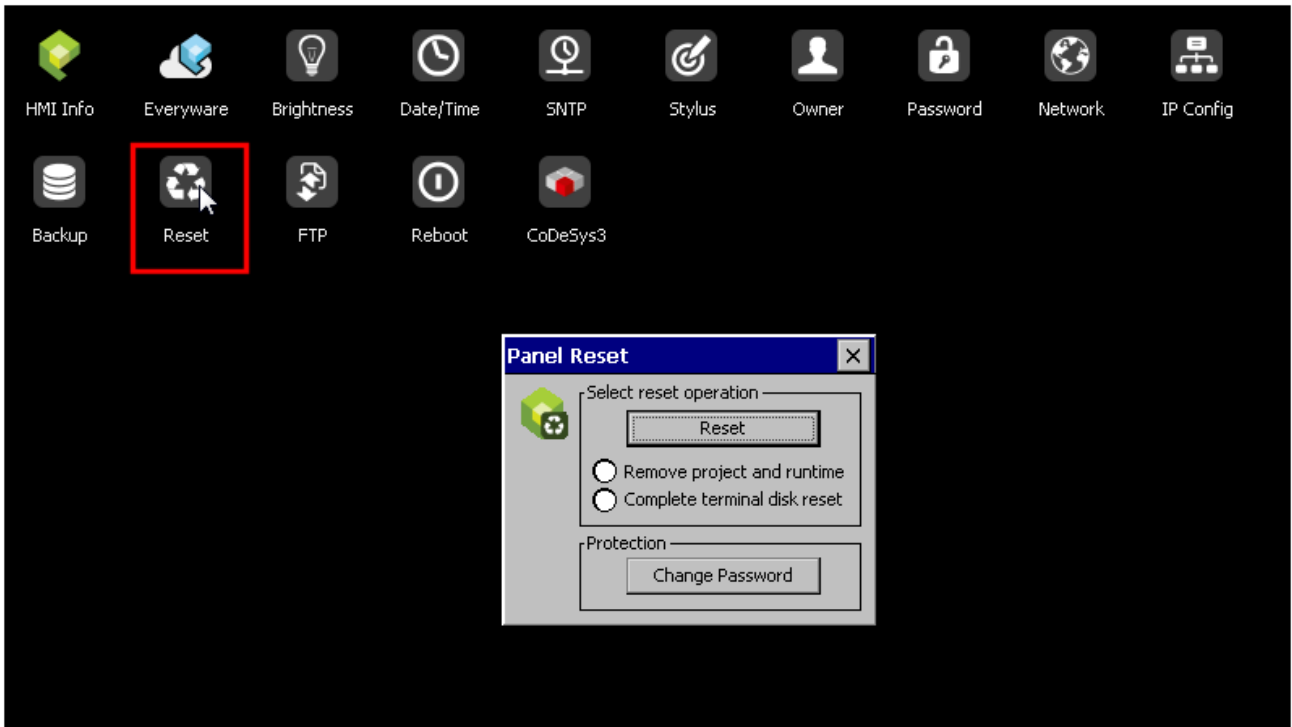
To make a Backup copy of chosen parts (Runtime, Project, Logs or CODESYS project).

It is essential to tick at least one of the components to be exported and choose a path where to save the file.

Then, it is possible to Restore the files, and this can be done for all exported components, or only for the ticked ones (see "[Backup and Restore - Example](#)" section).

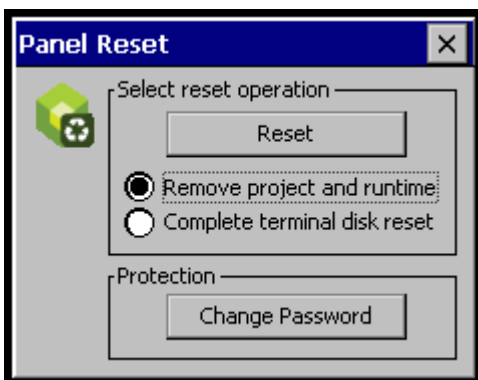
# CREW Manual

Reset:



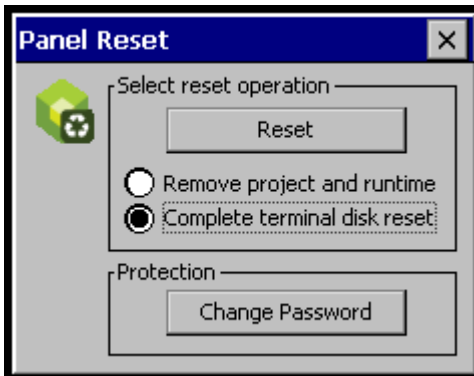
To Restore (clear) the terminal by choosing from the following two options:

- Removing only the project and Runtime :



# CREW Manual

- Clearing the entire terminal: the full contents of the terminal will be deleted, except for the files that are essential to its operation.



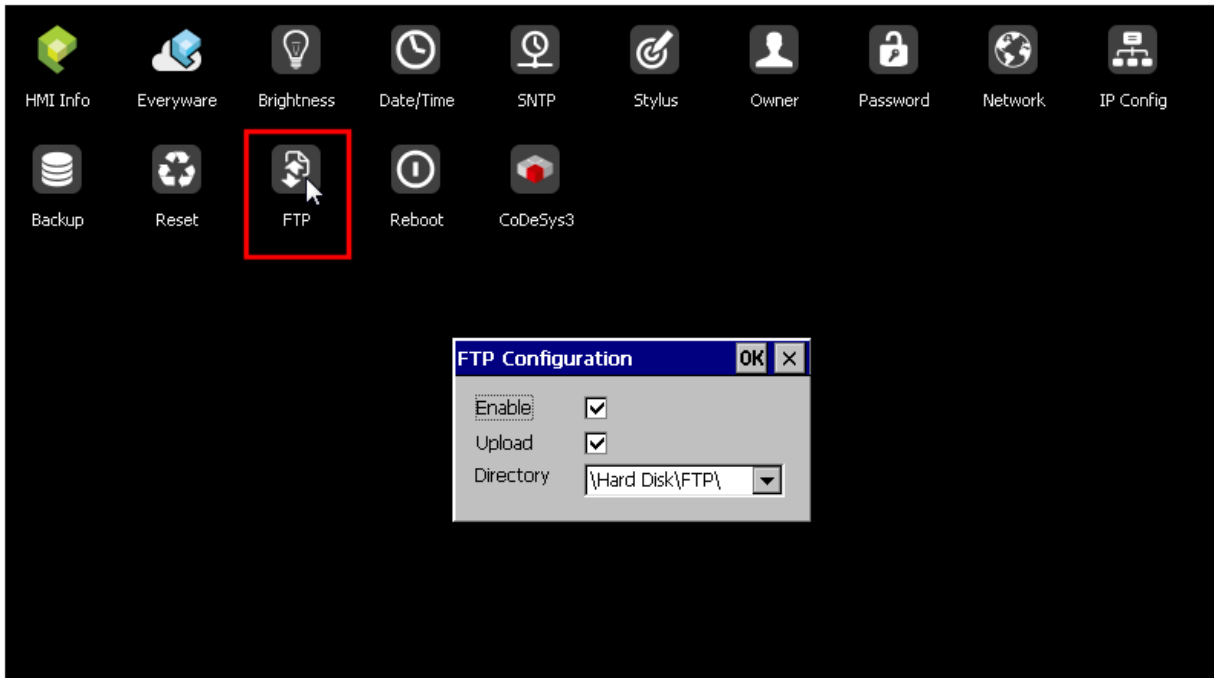
It is also possible to define or change the protection password.





# CREW Manual

## FTP:

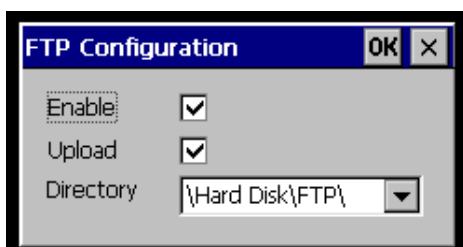


To enable the sharing service of a “FTP” (Files Transfer Protocol) folder.

The user has the possibility of enabling or disabling the “FTP Server” service of the panel from any other device connected to the network.

This function is very useful when it is necessary to write, cancel or modify data on the terminal easily from a remote access.

Select the “Enable” option to enable the “FTP” folder sharing service in the “Hard Disk” directory. On the other hand, select the “Upload” option, to enable the “writing / modify” mode of the shared folders.



# CREW Manual

Reboot:

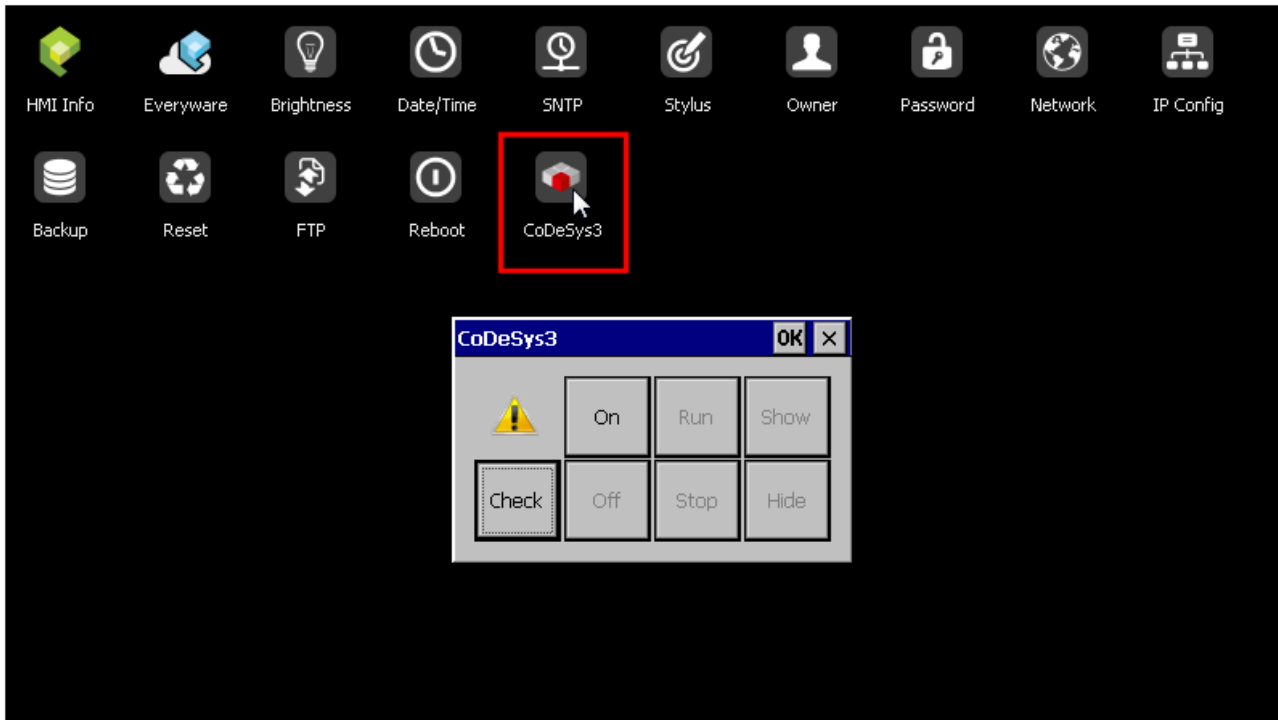


To restart the terminal in the two following ways:

- Normal restart: the terminal starts back up again normally.
- Restart to transfer CE image: the terminal is restarted by positioning yourself so as to transfer the terminal's CE image (see "[Upload panel image](#)" section).

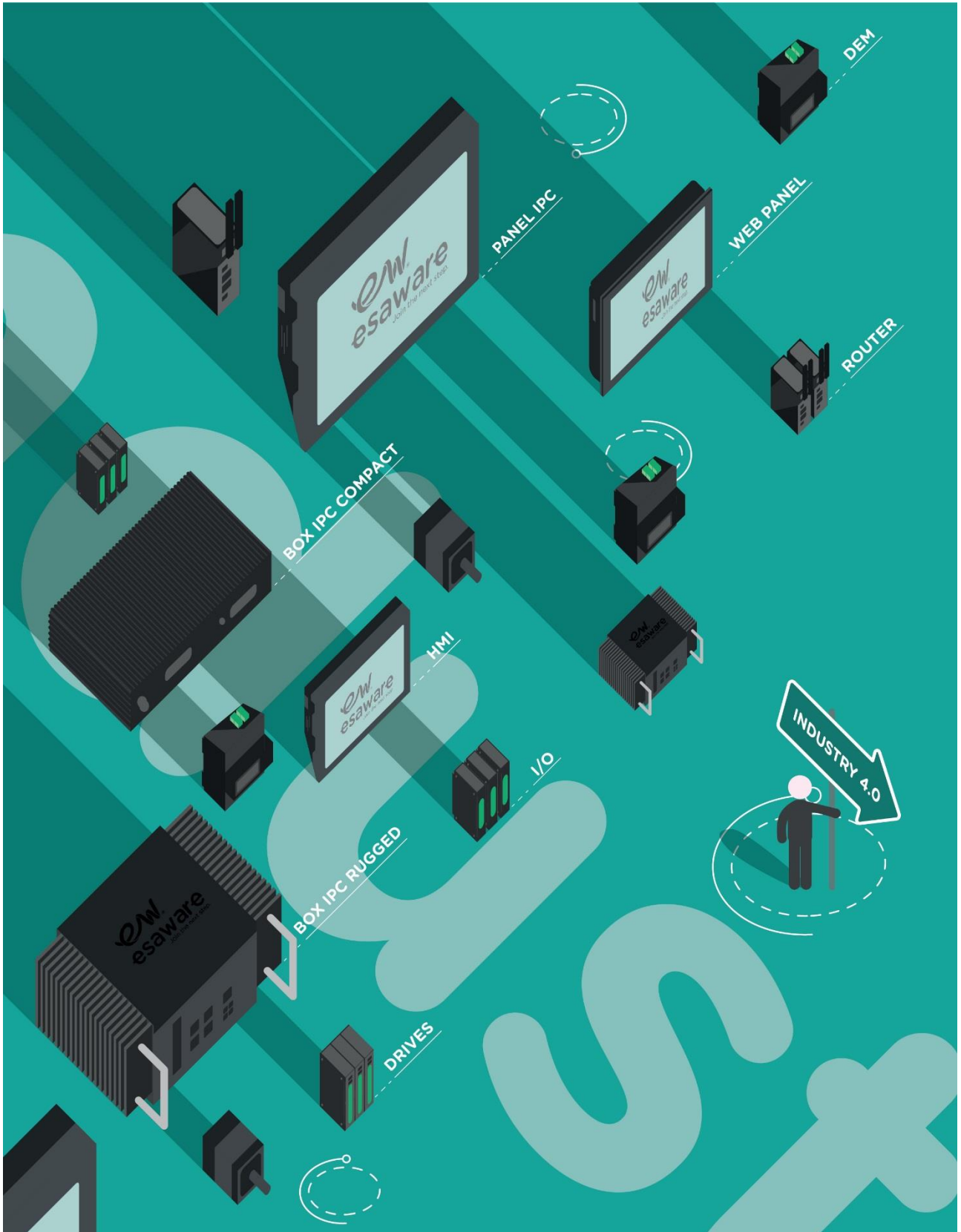
# CREW Manual

## CODESYS3:



To use CODESYS Runtime on the terminal (see the example in section "[Example of Crew+CODESYS project](#)", paragraph "Crew Section").

# CREW Manual



# CREW Manual



At ESA Automation we pride ourselves in designing, manufacturing and supplying the most technically advanced automation solutions available worldwide. We invest heavily in research and development to maintain our record of intuitive, high speed, reliable and sustainable products.

By utilising the latest hardware and software innovations ESA Automation's engineers ensure our products and solutions are future proof and by implementing state of the art smart technology they guarantee ease of use.

Our mission is to create solutions not just products.  
Our job is to simplify yours.

[www.esa-automation.com](http://www.esa-automation.com)  
[info@esa-automation.com](mailto:info@esa-automation.com)