

USER MANUAL
CRYSTAL XE

Evolution of this document

Date	Author	Rev	Description
19/11/2024	FC	18	Description of the recipe tool bar and add the amber recipe converter.
07/10/2024	FC	17	Explanation of security filtering modes.
08/08/2024	FC	16	Clock time and scheduler section updated.
01/08/2024	FC	15	New batch features, update options, add direct connect.
12/09/2023	FC	14	ASCII server: new event OnReceiveStr.
21/10/2022	FC	13	Script: new instructions: With, Record and Case..of
05/09/2022	FC	12	Automatic report generation.

Table of contents

1. ABOUT THIS MANUAL.....	8
2. INSTALLATION.....	8
2.1. INSTALLING CRYSTAL XE	8
2.2. LICENSE KEY	9
2.3. GENERAL INFORMATION	9
2.3.1. <i>Multiple instances</i>	9
2.3.2. <i>Clock, timestamp and real time engine</i>	10
3. BACKUP	11
4. FILE DESCRIPTIONS	15
4.1. FILES IN PROGRAM FOLDER.....	16
4.2. FILES IN PROJECT DIRECTORY	20
5. INTERFACE OVERVIEW	27
5.1. ICONS OVERVIEW	27
5.1.1. <i>User icon</i>	27
5.1.2. <i>Simulator icon</i>	28
5.1.3. <i>Alarm icon</i>	29
5.2. EVENT HISTORY	29
5.2.1. <i>Log viewer</i>	30
6. MENU BAR FEATURES	30
6.1. FILE MENU	30
6.1.1. <i>File/Open</i>	31
6.1.2. <i>File / Open recent:</i>	31
6.1.3. <i>File/Save:</i>	32
6.1.4. <i>Backup:</i>	32
6.1.5. <i>Close all forms</i>	32
6.1.6. <i>Browse project and program directories</i>	32
6.1.7. <i>Exit</i>	32
6.2. VIEW MENU	32
6.2.1. <i>Console</i>	33
6.2.2. <i>Memory</i>	33
6.2.3. <i>Splash window</i>	34
6.2.4. <i>Diagnostic</i>	34
6.2.5. <i>GSM inspector</i>	34
6.2.6. <i>Cloud inspector</i>	35
6.2.7. <i>Script explorer</i>	37
6.2.8. <i>Data explorer</i>	39
6.3. RECIPE MENU.....	39
6.4. TOOLS MENU	40
6.4.1. <i>Test tools</i>	40
6.4.2. <i>UDP Client/Server test</i>	43
6.4.3. <i>AIM IP setup</i>	43
6.4.4. <i>Log Viewer</i>	44
6.4.5. <i>CSV viewer</i>	45
6.4.6. <i>Black box viewer</i>	46
6.4.7. <i>Search</i>	46
6.4.8. <i>Check all scripts</i>	47
6.5. EDITORS MENU.....	47
6.6. SETUP MENU	47
6.7. WINDOW MENU.....	48
6.8. HELP MENU.....	48
6.8.1. <i>Operator manual</i>	48
6.8.2. <i>Script manual</i>	48
6.8.3. <i>Contact</i>	50
6.8.4. <i>Crystal XE versions</i>	50

6.8.5.	Check for update	51
7.	MAIN TAB.....	52
7.1.	AUTOMATION	52
7.2.	USER VIEW	52
7.2.1.	Example of equipment representation.....	53
7.2.2.	Interacting with the equipment.....	54
7.3.	SYSTEM SECURITY	56
8.	CHAMBER TABS	57
8.1.	USER VIEW	58
8.2.	EQUIPMENT VIEW	58
8.2.1.	Equipment list view	59
8.2.2.	Equipment setup	61
8.2.3.	Example of Detail view window (temperature).....	64
8.2.4.	Sub equipment inspector	67
8.3.	CHAMBER SECURITY.....	67
8.3.1.	Securities equipment view	68
8.3.2.	Tree view	69
8.3.3.	Alarm levels.....	70
8.3.4.	Configuring the security agents	70
8.4.	RECIPES	79
8.4.1.	Creating a new recipe	80
8.4.2.	Editing an existing recipe.....	81
8.4.3.	Executing a recipe.....	81
8.5.	CHARTS.....	84
8.5.1.	General.....	85
8.5.2.	Curves and data acquisition.....	90
8.5.3.	Chart options.....	98
8.6.	RECORDER	104
8.6.1.	Recording data	104
8.6.2.	Recorder templates.....	105
8.7.	DEVICES.....	108
8.8.	STATISTICS	110
9.	HARDWARE CONFIGURATION	112
9.1.	RESTORING DEFAULT SETUP VALUES	113
9.2.	MAIN ITEM.....	113
9.2.1.	Project information	113
9.2.2.	Global variables	114
9.2.3.	Simulation mode.....	117
9.3.	ARCHITECTURE ITEM	118
9.3.1.	Chambers	118
9.3.2.	Automation	120
9.3.3.	Chamber palette	120
9.4.	CONFIGURING CHAMBERS.....	121
9.4.1.	Parameters	121
9.4.2.	Recipes	122
9.4.3.	Platens used for platen automation.....	122
9.5.	CONFIGURING DEVICES.....	124
9.5.1.	Devices views	125
9.5.2.	Device palette.....	127
9.5.3.	Configuring communication ports.....	130
9.5.4.	Preconfigured devices	131
9.5.5.	Device models (Protocols)	132
9.5.6.	Connecting devices to communication ports.....	132
9.5.7.	Device settings	133
9.6.	CONFIGURING EQUIPMENT.....	133
9.6.1.	Equipment palette.....	134
9.6.2.	Adding equipment 	135
9.6.3.	Group of equipment (folders) 	138

9.6.4.	<i>Configuring sub equipment</i>	139
10.	OPTIONS	142
10.1.	GENERAL OPTIONS.....	143
10.2.	APPEARANCE.....	151
10.3.	STARTUP.....	154
10.4.	CHAMBERS	156
10.5.	EMAIL.....	157
10.6.	SMS	160
10.7.	CLOUD/MOBILE.....	162
10.8.	DATA LOGGER OPTIONS	163
10.9.	SERVERS.....	163
10.9.1.	<i>Modbus servers</i>	164
10.9.2.	<i>ASCII Servers</i>	166
10.9.1.	<i>Direct connect</i>	174
10.9.2.	<i>UDP Server</i>	175
10.10.	ADVANCED.....	176
10.10.1.	<i>Black box (background circular recorder)</i>	176
11.	USERS	176
11.1.	MANAGING USERS	179
11.1.1.	<i>Creating a new user</i>	179
11.1.2.	<i>Managing groups</i>	180
12.	RECIPES	183
12.1.	HOW DOES THE RECIPE WORK?	183
12.2.	RECIPE EDITOR	185
12.3.	OVERVIEW OF THE RECIPE EDITOR.....	186
12.4.	WORK WITH THE TREE VIEW	186
12.5.	WORK WITH THE GRID VIEW	188
12.6.	THE MAP VIEW	190
12.7.	HOW TO ADD A TOP-LEVEL ELEMENT.....	192
12.8.	ADD AN EMPTY LAYER.....	193
12.9.	ADD A LAYER WITH EVENTS	194
12.10.	ADD ACTIONS TO A LAYER.....	194
12.11.	USING TAG OR PROPERTIES (LIKE GLOBAL VARIABLES).....	201
12.12.	USING OF AN EXTERNAL SCRIPT LIBRARY.....	203
12.13.	USE OF LOCAL VARIABLES.....	205
12.14.	LOOPS	206
12.15.	CONDITIONAL LAYERS	207
12.16.	MODIFY A VALUE WITH REPERCUSSIONS IN THE OTHER LAYERS	212
12.17.	OFFSETS	212
12.18.	SEARCH AND REPLACE	214
12.19.	WORK IN THICKNESS	215
12.20.	SCRIPT TAB.....	217
12.21.	CHART VIEW.....	217
12.22.	SCRIPT RECIPE EDITOR.....	218
12.23.	CONVERT RECIPE CSV TO RCP.....	219
12.24.	EXPORT OLD “CRYSTAL EYES” RECIPES TO CSV FILES	219
12.25.	CONVERT CSV FILES FROM CRYSTAL XE.....	220
12.26.	CONVERT OLD CRYSTAL EYES RECIPES TO CRYSTAL XE RECIPES	223
12.27.	CONVERT AMBER RECIPES TO CRYSTAL XE SCRIPT	224
12.27.1.	<i>Preliminary</i>	224
12.27.2.	<i>Using the Amber recipe converter</i>	224
12.27.3.	<i>Conversion table</i>	226
12.27.4.	<i>Limitations</i>	228
12.28.	USE A DIALOG BOX (FORM) IN A RECIPE.....	229
12.28.1.	<i>Create the dialog box</i>	229
12.28.2.	<i>Create the Recipe</i>	230
12.28.3.	<i>Run the recipe</i>	231
12.29.	AUTOMATIC REPORT GENERATION	231
12.29.1.	<i>Default automatic report</i>	232

12.29.2.	Customize the automatic report	233
13.	FORMS AND OBJECTS.....	234
13.1.	INTRODUCTION	234
13.2.	OBJECT LIBRARY	235
13.3.	SETTINGS	237
13.4.	FORMS	237
13.4.1.	CREATING FORMS.....	239
13.5.	FORM PROPERTIES AND EVENTS	247
13.5.1.	Basic properties and OnCreate event.....	247
13.5.2.	Advanced properties and events.....	259
13.6.	OBJECT PROPERTIES AND EVENTS	261
13.6.1.	Basic settings.....	262
13.6.2.	Advanced settings.....	266
13.7.	TYPES OF OBJECT	269
13.7.1.	Basic tab (list of equipment).....	269
13.7.2.	Usual tab.....	271
13.7.3.	ON/OFF tab.....	297
13.7.4.	Windows tab.....	305
13.7.5.	OTHERS.....	319
14.	PLATEN AUTOMATION	342
14.1.	EDITING BATCH USING THE BATCH EDITOR	344
14.2.	EDITING BATCH USING THE SCRIPT EDITOR	350
14.3.	EXECUTING A BATCH	352
14.4.	MODIFY PATHS INDIVIDUALLY AT ANYTIME	356
14.5.	MOVING A PLATEN FOR A SIMPLE MOVEMENT	357
15.	DIRECT CONNECT	359
15.1.	INTRODUCTION	359
15.2.	EXAMPLE WITH THE RIBER SOFT CLIENT APPLICATION	360
15.3.	PROTOCOL	364
16.	DATA LOGGER.....	366
16.1.	INTRODUCTION	366
16.2.	PARAMETERS.....	366
16.3.	DATA SENT TO THE DATA LOGGER.....	367
16.4.	PROTOCOL.....	368
16.5.	DATA FLOW STRUCTURE.....	368
16.6.	EXAMPLE OF DATA EXCHANGED WITH A SERVER.....	369
17.	DYNAMIC DATA EXCHANGE (DDE).....	370
17.1.	INTRODUCTION	370
17.2.	DDE EXCHANGES WITH EXCEL.....	371
17.2.1.	Display live data from Crystal XE: Hot DDE Link.....	371
17.2.2.	From Crystal XE request data to Excel (Using script / DDE_Request).....	372
17.2.3.	From Crystal XE send data to Excel (Using script / DDE_Poke).....	372
17.2.4.	From Crystal XE execute a method in Excel.....	372
17.2.5.	From Excel write data to Crystal XE (Using macro / DDEPoke).....	374
17.2.6.	From Excel read data in Crystal XE (Using macro / DDERequest).....	374
17.3.	DDE EXCHANGES WITH WORD	375
17.3.1.	From Crystal XE send data to Word.....	375
17.3.2.	From Crystal XE execute a method in Word.....	375
17.4.	FROM EXCEL OR WORD, SEND DATA TO CRYSTAL XE (USING MACRO / DDEEXECUTE).....	376
17.5.	DDE EXCHANGE WITH LIBREOFFICE CALC.....	377
17.5.1.	Display live data from Crystal XE (Hot DDE Link).....	377
17.6.	MORE INFORMATION ON DDE	377
18.	OBJECT LINKING AND EMBEDDING (OLE).....	377
19.	FAQ	378
19.1.	DISPLAY PROBLEM IN CHARTS	378

19.2.	USING OF NATIONAL INSTRUMENTS CARDS.....	378
19.2.1.	<i>Installation procedure</i>	380
19.3.	USING CRYSTAL WITH ARDUINO	381
20.	ANNEXE A - WRITE A RECIPE BY EXAMPLE	382
20.1.	OPERATING BASICS	385
20.2.	MORE INFORMATION ABOUT RECIPE EDITION	391
20.3.	LOOP EDITION	400
20.4.	IN-DEPTH INFORMATION ABOUT LOOP USAGE	402
20.5.	RECIPE EDITION: FINALIZATION	407
20.6.	IN DEPTH INFORMATION ABOUT RECIPE EDITION.....	410
20.7.	RECIPE EDITION: ADDITIONAL VIEWINGS	426
21.	LICENSE	433

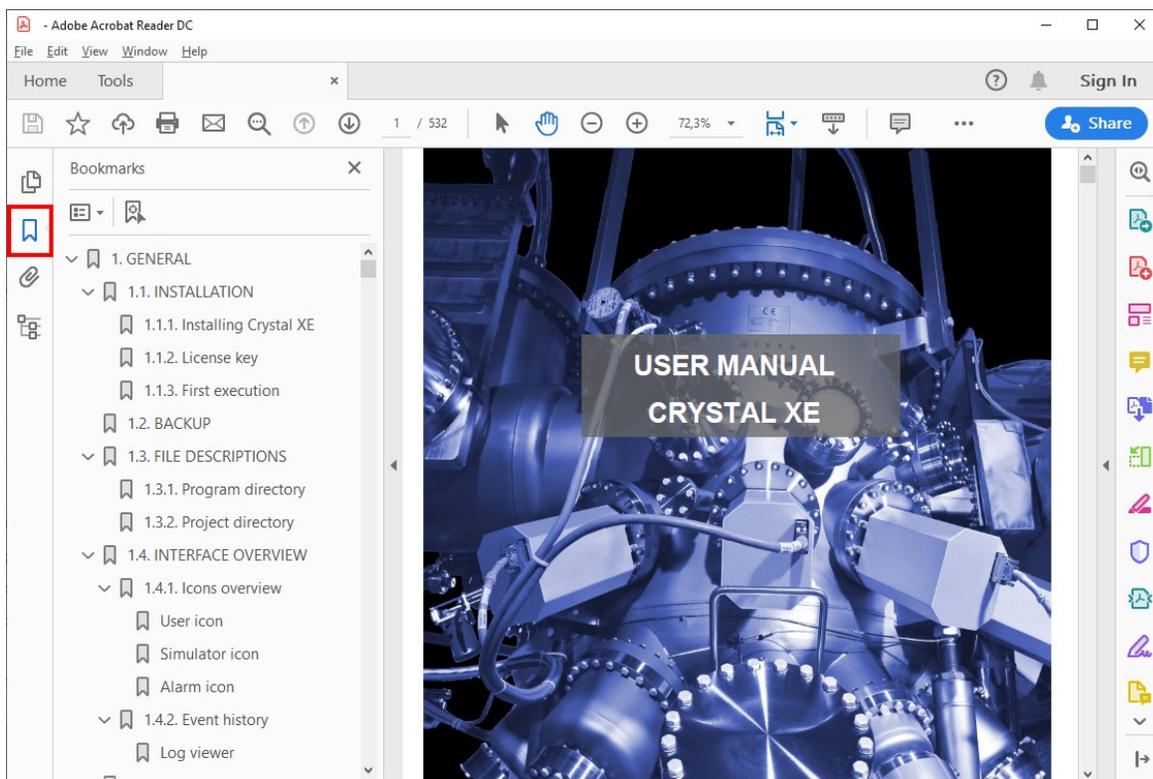
1. ABOUT THIS MANUAL



We attach importance to the environmental impact and for this reason we do not recommend the printing of this manual.

In addition, reading is made easier with the search and navigation tools offered by PDF file readers such as Adobe Acrobat Reader.

In Adobe Acrobat Reader, to display the navigation pane, press the following button (in a red rectangle):



2. INSTALLATION

2.1. Installing Crystal XE

If *Crystal XE* is preinstalled on your computer, the default directories are the following:

Program directory: `c:\riber\CrystalXE`

Project directory: `d:\riber\<Project name>`

If it is not installed, execute the setup program and follow the installation instructions (select the demo *Files*) in the dialog box to complete the installation.

① In *Crystal XE*, a project corresponds to the directory name in which you can find the data *Files* specifying your application (configuration *File*, synoptic, security *Files* etc.).

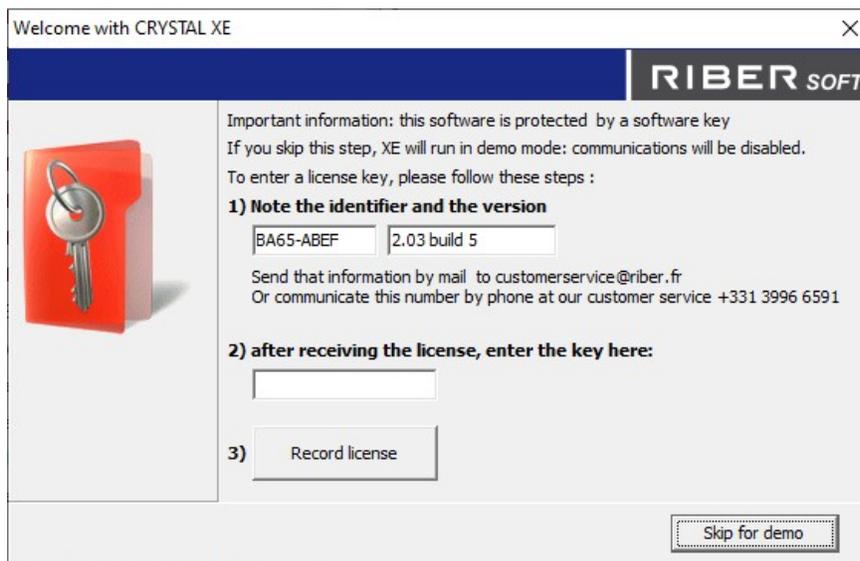
A *Crystal XE* shortcut icon should be available on your desktop. If not, create the shortcut from the 'CrystalXE.exe' *File* located in the directory: *C:\riber\CrystalXE*.

2.2. License key

A license key is required to use the full functionality of *Crystal XE*.

You will be asked for a software license key when running the software.

Note down your software information and send it to RIBER customer service to receive your license key.



If you have no license, you can test the software in demonstration mode.

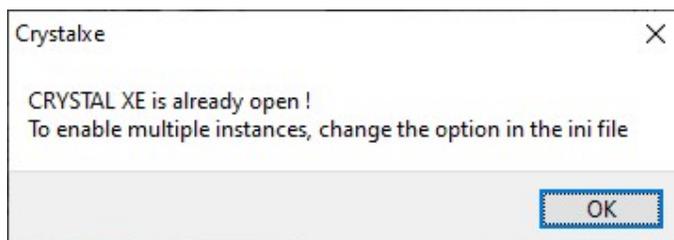
In demonstration mode, communication with the devices is not possible, however, all the other functionalities are operational.

2.3. General information

2.3.1. Multiple instances

By default, it is not possible to open *Crystal XE* several times.

If you try to open the application several times the following message will appear:



In some cases, it may be necessary to run the application more than once simultaneously.

In this case, here is the procedure to follow:

- o Make a copy of the program folder and all the files it contains to another location (example: c:\riber\CystalXE2).
- o Open the CrystalXE.ini file in this new folder using a text editor such as NotePad.
- o In the [Config] section, change the line AllowMultipleInstance = 0 to AllowMultipleInstance = 1.
- o Do the same in the original program folder.

Why duplicate the program files?

Note that the name of the last project used is saved in the crystalxe.ini file in the program folder. Thus, in order to be able to run CrystalXE several times with different projects, the program folders must be separated.

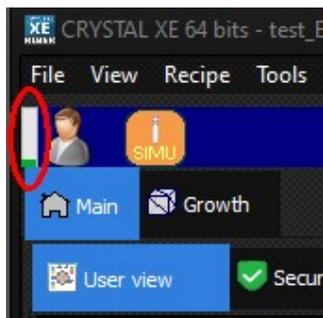
2.3.2. Clock, timestamp and real time engine

In SCADA software, time management, stability and reliability are critical. For this reason, every effort has been made during the development of Crystal XE to offer the application maximum performance so as not to affect the results obtained. We attach the utmost importance to ensuring that critical times are not affected, which is what makes Crystal XE so reliable and trustworthy.

CPU Usage

You can constantly monitor CPU activity via the progress bar at the top left of Crystal XE's main screen. The color of this progress bar changes to yellow and then to red as activation increases.

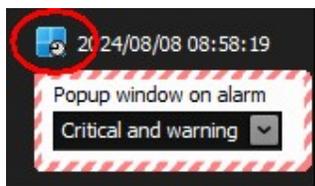
When the situation becomes critical, an alert will be generated.



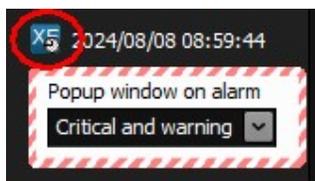
Excessive activity may indicate the presence of a virus or a scan in progress for indexing or virus scanning.

Time stamped data and clock

All time stamped data uses the PC clock but not only that, the recipe scheduler also uses the PC clock to calculate the time of each layer. So if the time changes during a recipe, the duration of the current layer could be affected. To avoid this, Crystal uses its own internal clock during critical moments. You can tell which clock Crystal is using by looking at the bottom right of the screen.



Crystal XE uses the Windows clock



Crystal XE uses the internal clock

Since the user can update the clock or when connected to the Internet, the operating system itself updates the PC clock using the Internet time server.

When Crystal is running but inactive (no recipe in progress...), the Windows clock is used. In this case, if the system clock is updated, this will only affect graphics and recorders (CSV data files).

Therefore, when the date and time changes, this information is recorded in the historical log so that the information can be retrieved.

When you are connected to the Internet, we recommend that you disable the connection to the Internet time server.

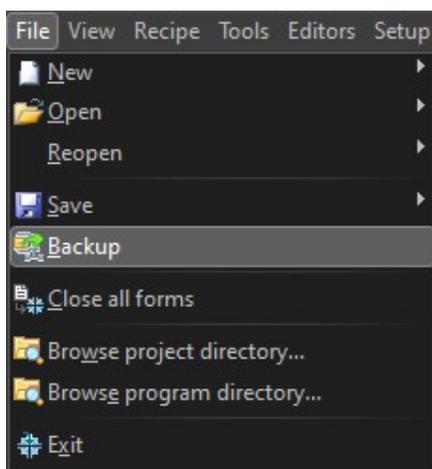
Please note that communication with devices and data servers is not affected by a time change, as they use a thread independent of the system clock. This thread is not affected by CPU activity either, as it is a priority task. In this way, the times used in recipes will remain reliable with a resolution of around 10ms.

3. BACKUP

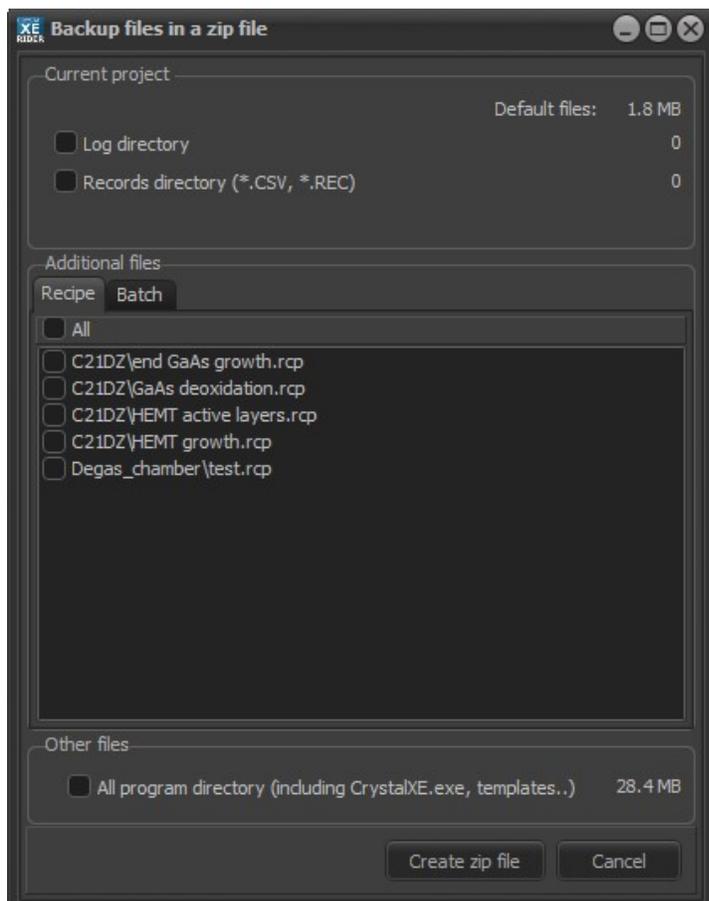
This option allows you to save all the files necessary for the project in a single compressed file.

By adding the program directory option, this allows you to create a clone of the project with the Crystal XE application in a single zip file.

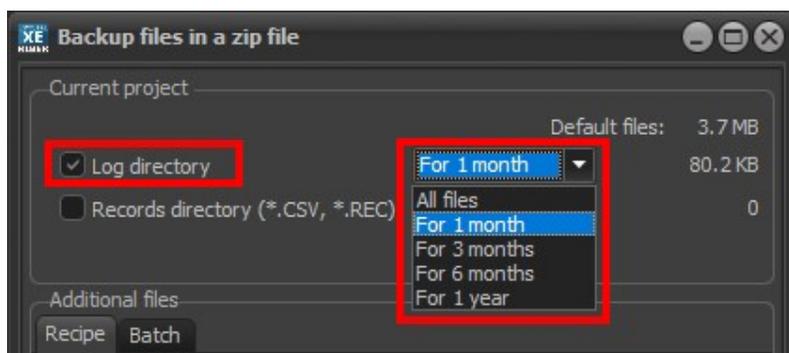
From the main menu, select **File / Backup**



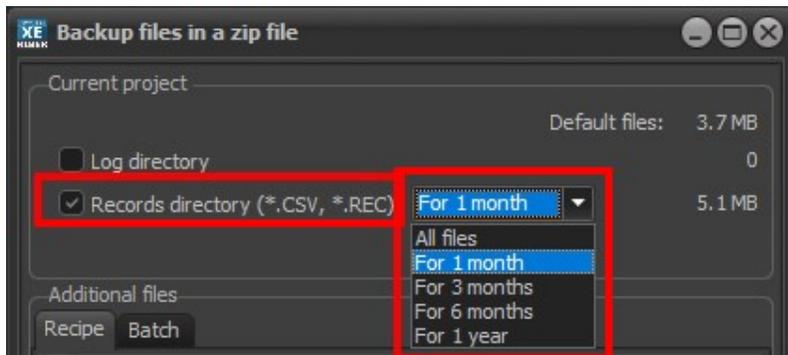
The following window opens:



By clicking on the log check box, a list box appears allowing you to choose the duration of the log files.

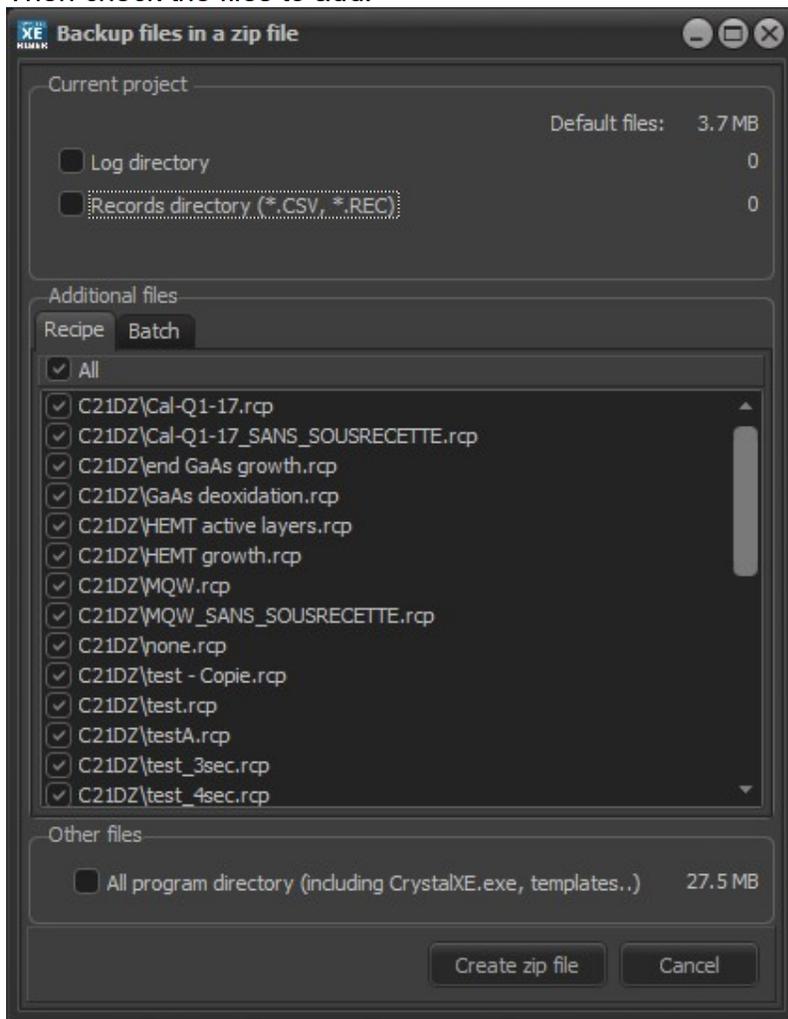


Same for the Records directory

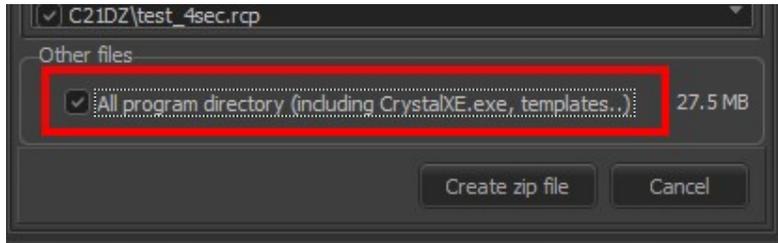


 If you select *Records directory box*, large files might be generated.

Then check the files to add:

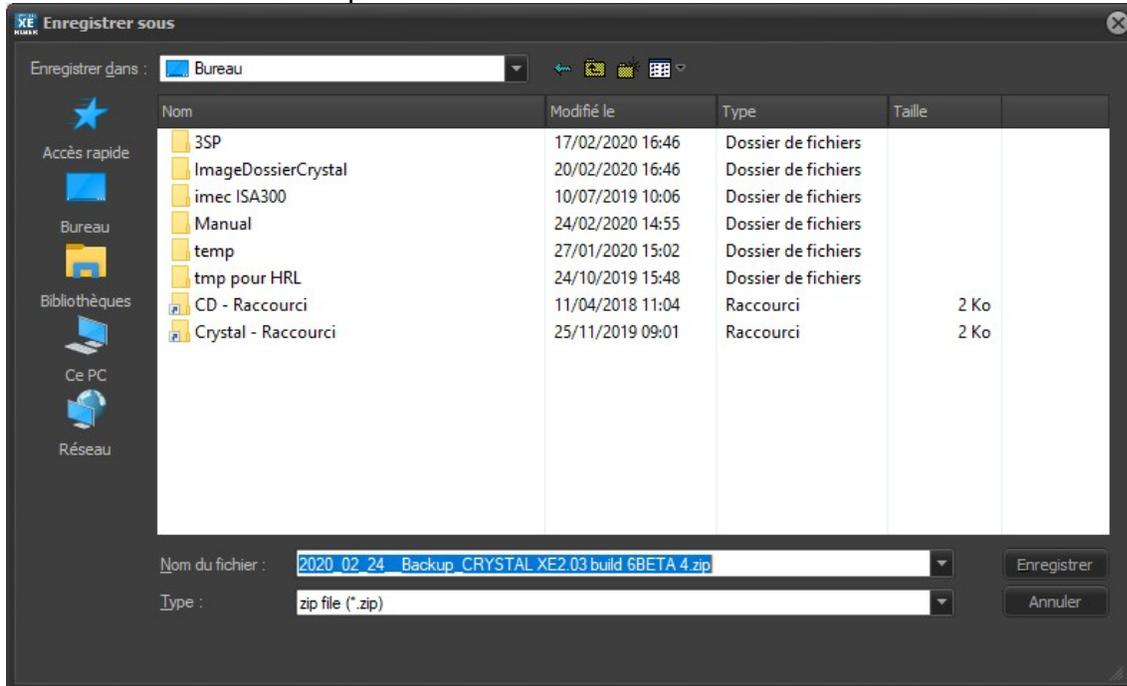


To add also the program directory, check the box in the Other files section.

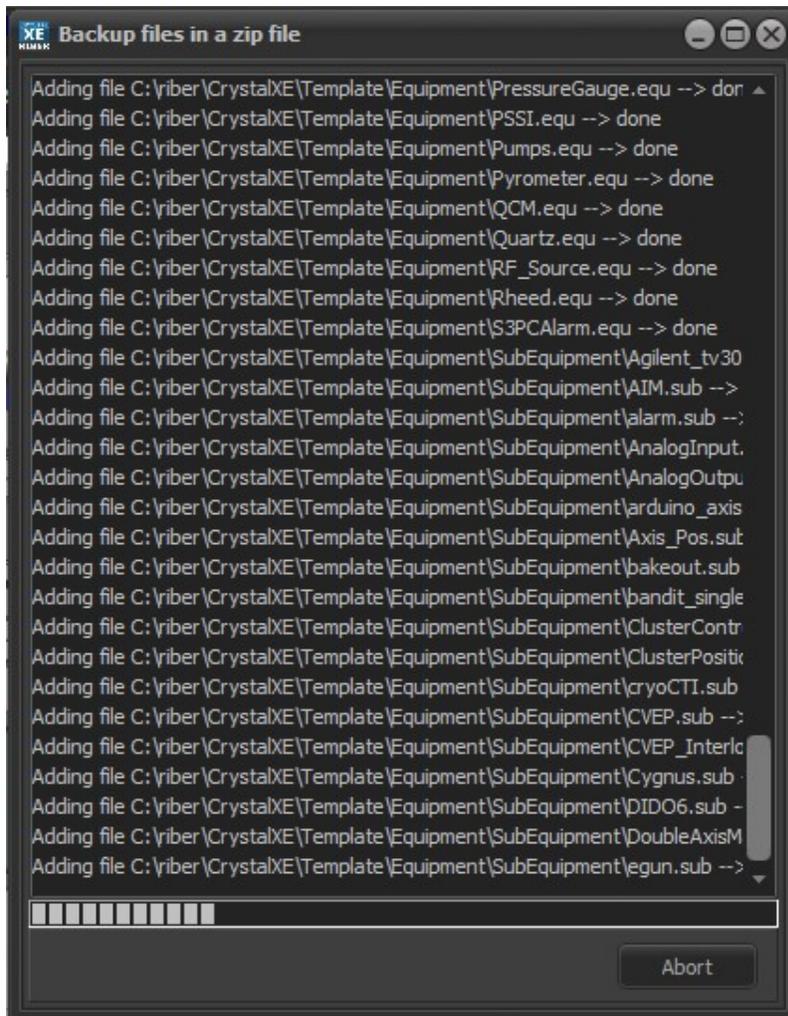


And Press “Create zip file”

Give a name to the backup file:



And press Save



If the program check box has been checked, the zip file will contains two directories:



To open the project, extract this zip fil into a new directory.

Go in the program directory and click on CrystalXE.exe to launch Crystal.

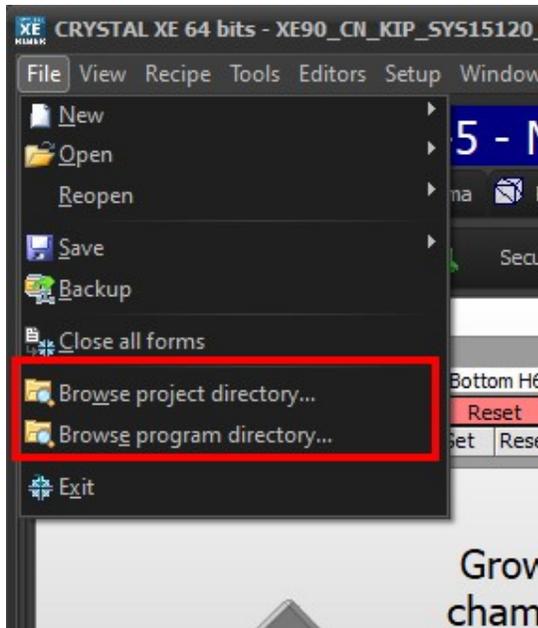
When opening Crystal XE, you will be asked to select the project folder. Choose the folder that has just been created.

4. FILE DESCRIPTIONS

Crystal XE files are distributed in two directories.

- Crystal XE program files are located by default in the directory c:\riber\CrystalXE.
- Crystal XE project files

To browse these folders, from the main menu, click File and then "Browse project directory .." or "Browse program directory .."

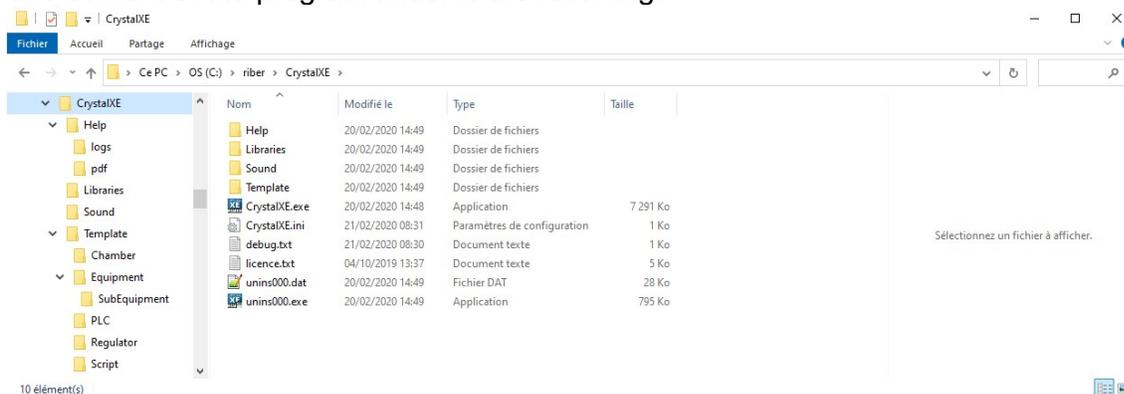


File menu that shows the "Browse" items.

4.1. Files in program folder

The program folder contains all files needed for the application. When installing a new version of Crystal XE, this directory is renamed with the current date and time. So it is possible to go back to the previous version by opening the file CrystalXE.exe in the old directory. For that reason, it is not recommended to save the data files in the Program directory because they will be lost during a software update.

The content of the program folder is the following:



Content of program folder

- **CrystalXE.exe:** this is the main executable file for Crystal XE. Simplicity makes a program reliable. For this reason, Crystal XE does not require other files like framework or other external modules. It possibly uses two DLLs which are in the library folder when using email or when using the cloud. This feature enhances the reliability of the software. This also makes it easy to install by simply copying the program folder to another PC.
- **CrystalXE.ini:** contains some information like the current project to load at startup, this file is a text file that it is readable with any text editor but we don't recommend to modify it.
- **licence.txt:** it is a text file that contains the license that was approved during software installation.
- **Unins000.dat and Unins000.exe:** these files are used to uninstall Crystal XE. They are executed by windows when uninstalling the software.

Help folder:

This folder contains:

- o The Crystal XE help file which notably contains help on scripts.
- o The Crystal XE help operator which contains an help on equipment, sub-equipment and devices.

Libraries folder:

 libeay32.dll	21/12/2019 12:01	Application exten...	2 240 KB
 OpenSSL License.txt	31/08/2016 13:14	TXT File	7 KB
 ReadMe.txt	21/12/2019 12:01	TXT File	3 KB
 ssleay32.dll	21/12/2019 12:01	Application exten...	378 KB

Content of libraries directory

Theses files are needed to access to the cloud or to send data to the data logger, or to send email.

libeay32.dll contains encryption functions which allow for coded communications over networks. This file is opensource and is used in many opensource programs to help with SSL communication

OpenSSL is a software library for applications that secure communications over computer networks against eavesdropping or need to identify the party at the other end. It is widely used by Internet servers, including the majority of HTTPS websites. OpenSSL contains an open-source implementation of the SSL and TLS protocols. The core library, written in the C programming language, implements basic cryptographic functions and provides various utility functions. Wrappers allowing the use of the OpenSSL library in a variety of computer languages are available.

The OpenSSL Software Foundation (OSF) represents the OpenSSL project in most legal capacities including contributor license agreements, managing donations, and so on. OpenSSL Software Services (OSS) also represents the OpenSSL project, for Support Contracts.

ssleay32.dll is a module associated with The OpenSSL Toolkit from The OpenSSL Project, <http://www.openssl.org/>.

The other files are text files that you can read, such as license files.

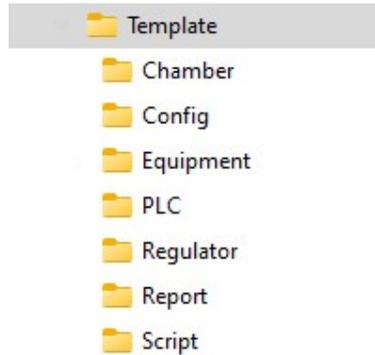
Sound folder:

This folder contains sound files that can be used in the securities to make a sound when an alarm occurred.

Template folder:

This folder is very important. It contains all the files needed to communicate with the devices, all the files to manage equipment and also script that are used for the ramp or other processes.

The template folder contains other sub folder which are the following:



Content of template folder

Chamber folder:

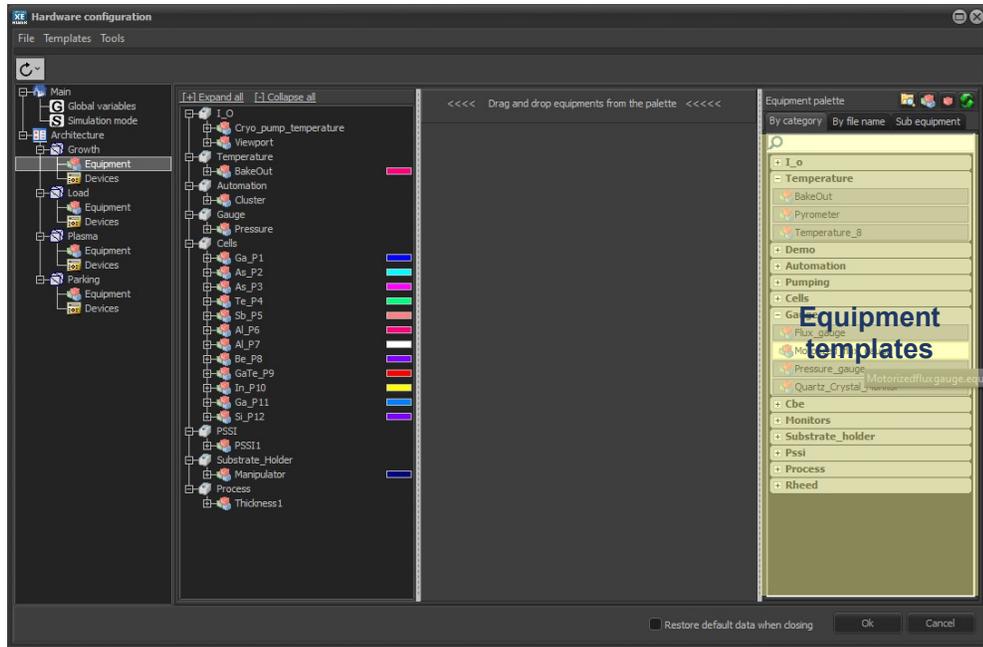
This folder contains chamber models that can be used in the hardware configuration. By default, there is one template in this folder: C21_DZ.xfg. The default extension for this kind of file is .xfg

Config folder:

This folder contains system models that can be used in the hardware configuration.

Equipment folder:

The equipment models are saved here. These models are displayed in hardware configuration, to the right of the window, when you click on equipment, in the equipment palette.



Equipment templates in hardware configuration.

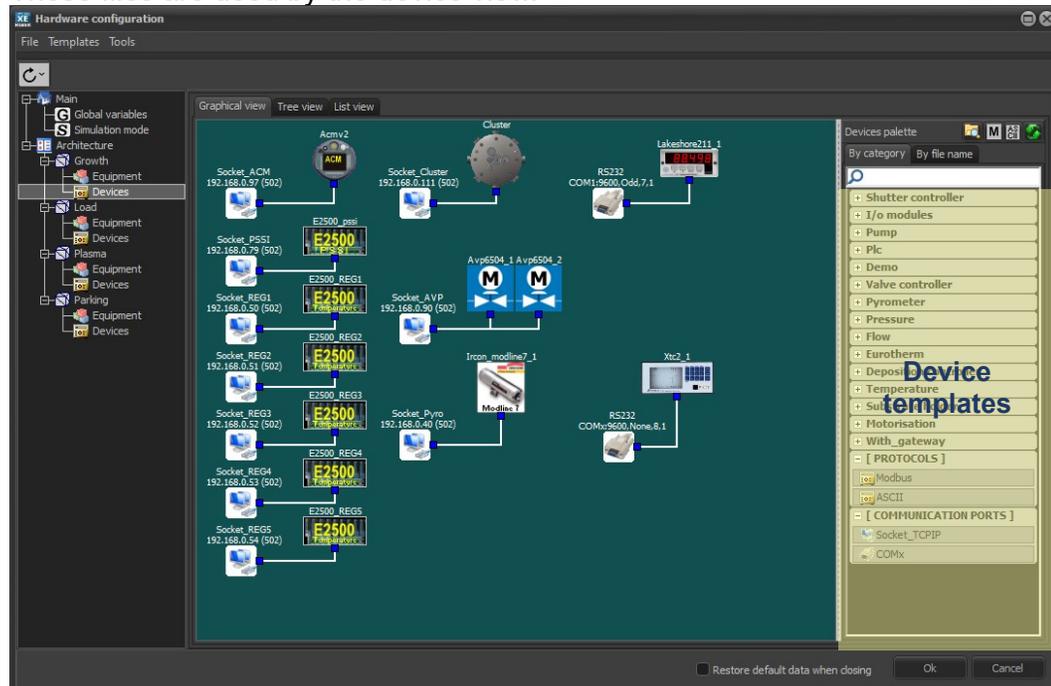
This folder contains also a directory named subequipment. These files are used by the equipment and are also templates. The extension of sub equipment is .Sub.

PLC folder:

This folder contains files that are used by hardware configuration. These files contain the list of transfer automation alarms.

Regulator folder:

This folder contains files that are used by hardware configuration (device files). These files are used by the device view.

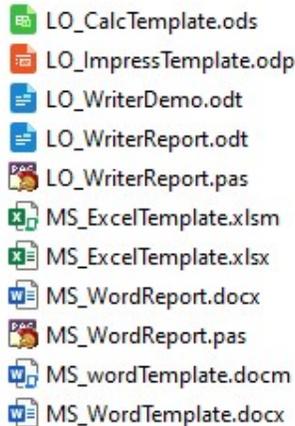


Regulator templates in hardware configuration

Two files are used for each device. One is the device driver (.rgr), and the other is an image file (.bmp) that is displayed in the device view.

Report sub folder:

This folder contains reports templates that can be used in recipes.



We supply two types of templates for Libre office (LO) and Microsoft office (MS).

Script sub folder:

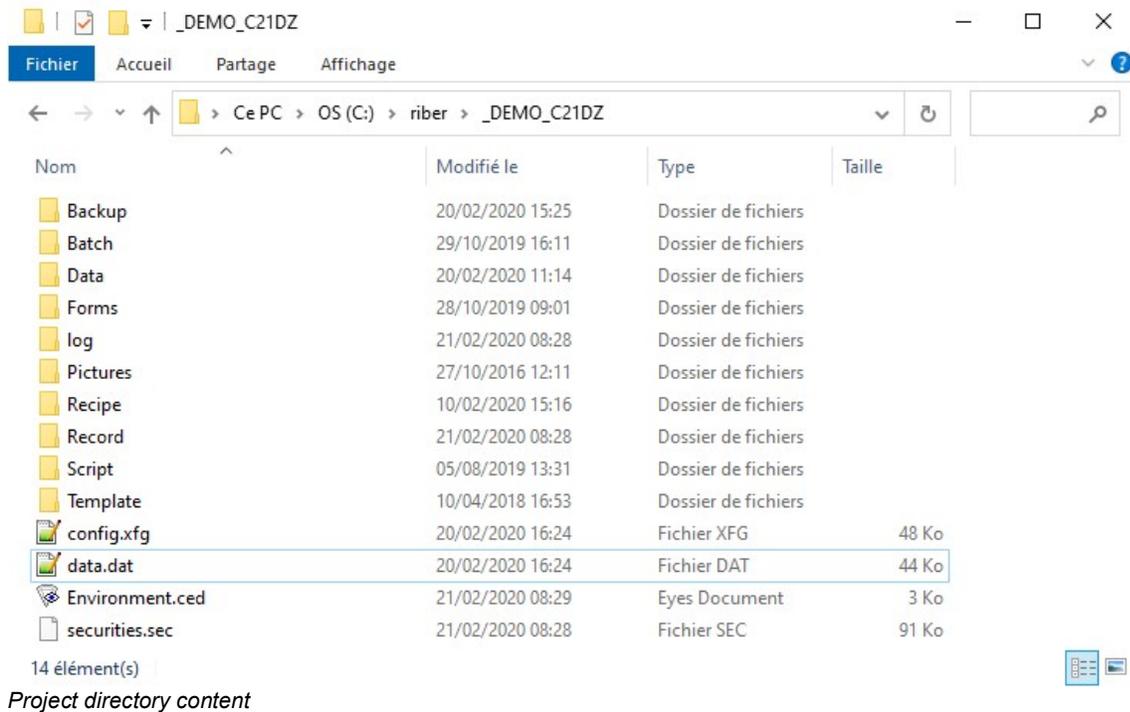
This folder contains script files (.pas) that are used in equipment, sub-equipment or devices.

Some script files can also be used by the user such as const_colors which contains constant colors.

4.2. Files in project directory

The project directory contains all files necessary for the project to supervise the Riber system. Generally, this directory is located in the disk d:\riber\

It must not be in the program directory because when updating the program, the directory path will be lost.



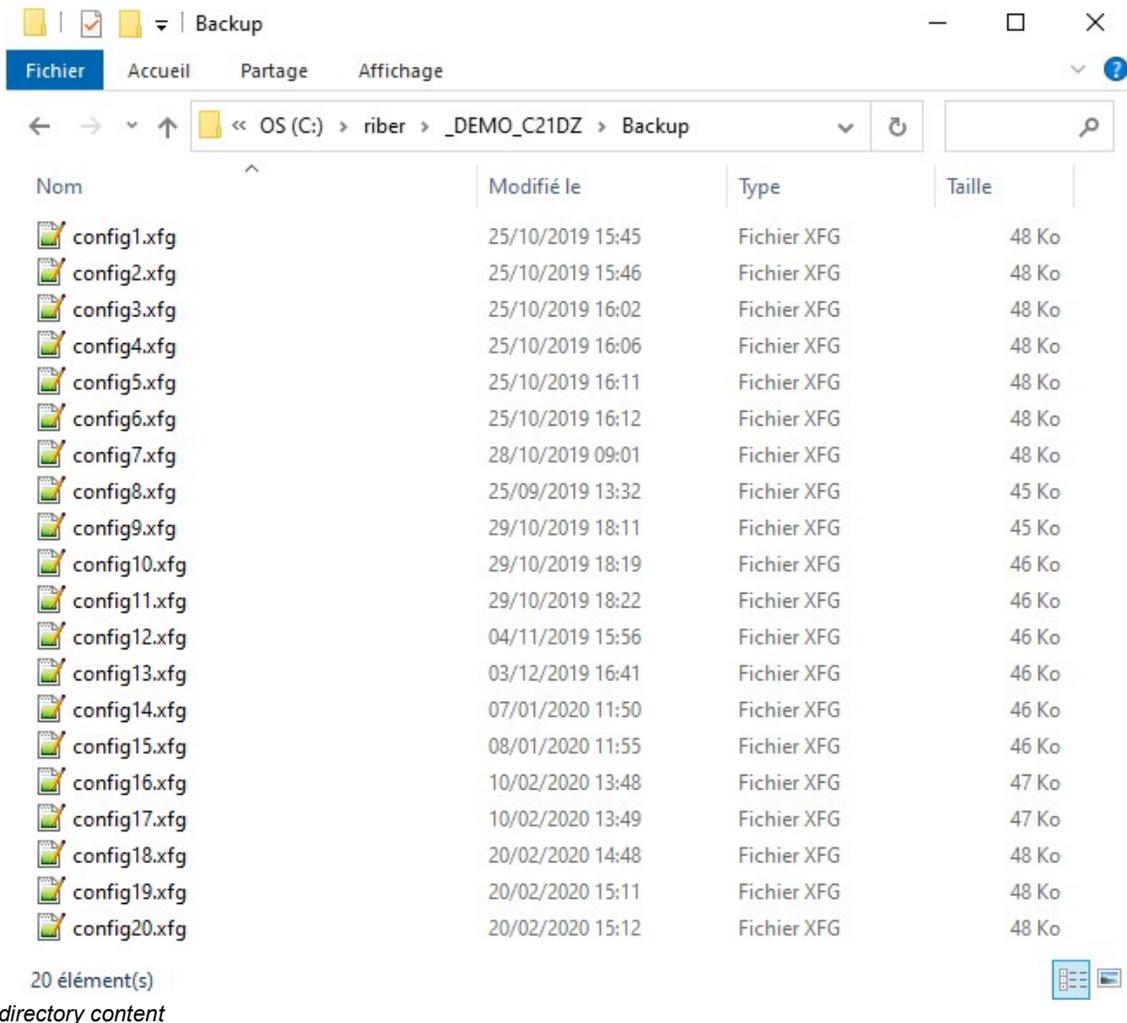
config.xfg: This XML file contains all the information of the hardware configuration as well as the setup/options. It is not recommended to modify this file yourself.

data.dat: This XML file contains all the data that the user can modify like the minimum setpoints, maximum setpoints etc.. all the data that can be change in the setup of all equipment and sub equipment, and also the setup of all devices.

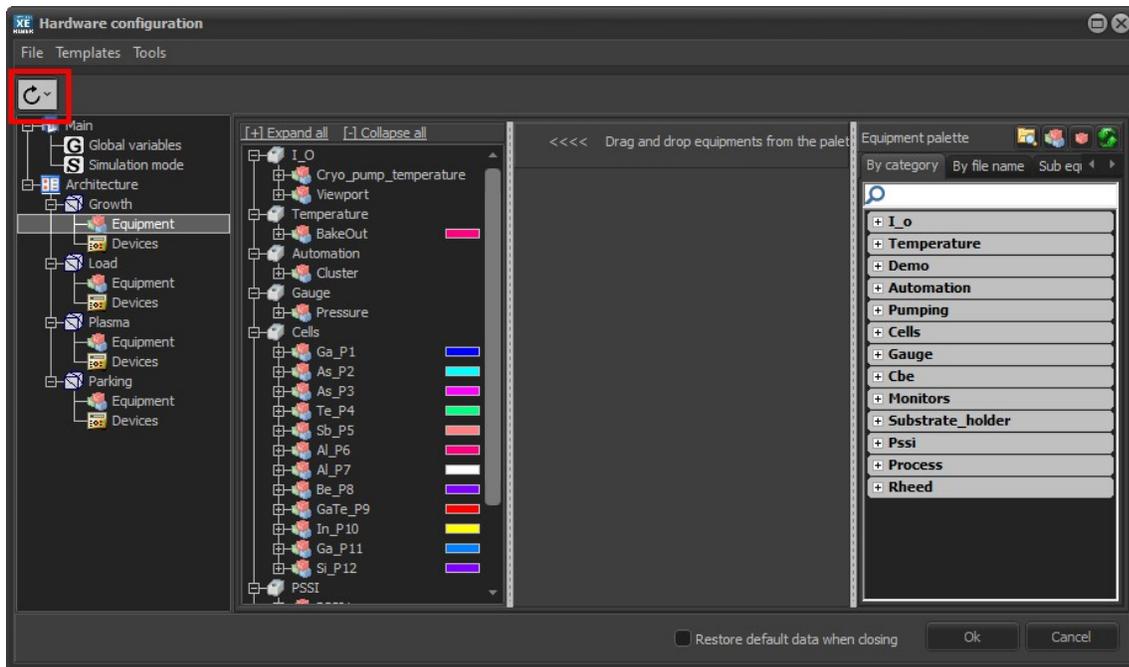
Environment.ced: This XML file contains the configuration of the charts tab, the equipment tab (order of the equipment, if there are hidden or visible etc..)

Securities.sec: This XML file contains all the information of the securities tab for each chamber.

Backup directory:

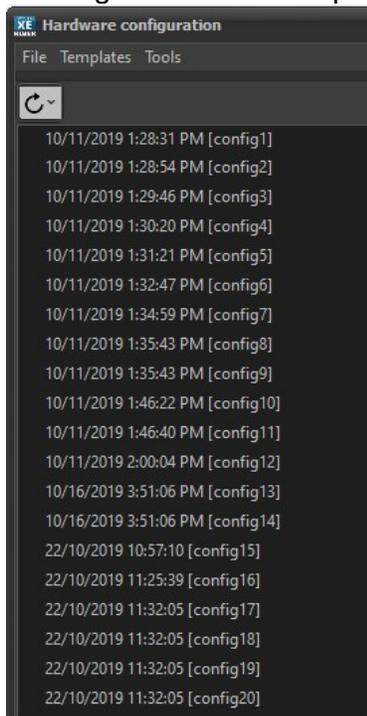


The backup directory contains the last 20 changes in the hardware configuration. You can go back to a previous configuration by using the following button in the hardware configuration:



Button to restore previous configurations

Clicking this button will open a context menu with a maximum of the last 20 configurations:



List of previous configurations

You can click on one of them to open an older configuration.
For more details, see the hardware configuration section in this manual.

Batch directory

This directory contains all the batch files uses by the transportation system.

The batch files are in XML format.

Data directory

Some temporary files are saved in this directory.

- **.prf files:** are profile files for ramping.
 - **Recipe_conversion_table.ini:** contains necessary data to convert old recipe files.
- This directory can also contain user data files such as growth rate data used for thick recipes or the content of graphics, etc.

Forms directory

In this directory, two kind of files can be found:

- **.sysform** files: these files are used by user view forms like the main user view whose filename is mainuserview.sysform
- **.form** files: these files are additional forms created by users or specially developed for the project.

Log directory

This directory contains several types of log files.

- **<date>.log** : contains the historical log which is displayed at the bottom of the screen (example of file: 2020_01_07.log).
- **Recipe log files** whose name contains the date and time and name of the chamber. These files contain the historical log which is displayed in the recipe tab during the execution of a recipe.
- **<date_time>_Batch.log:** These files contain the historical log which is displayed in the automation tab during the execution of a batch.
- **<date_time>_Batch.ini:** Contains the main information of a batch like the start time, the number of platens and for each platen the path with date and time information.

Pictures directory

This directory is reserved for storing images that could be used in the forms.

Recipe directory

Contains several sub directories, one per process chamber.

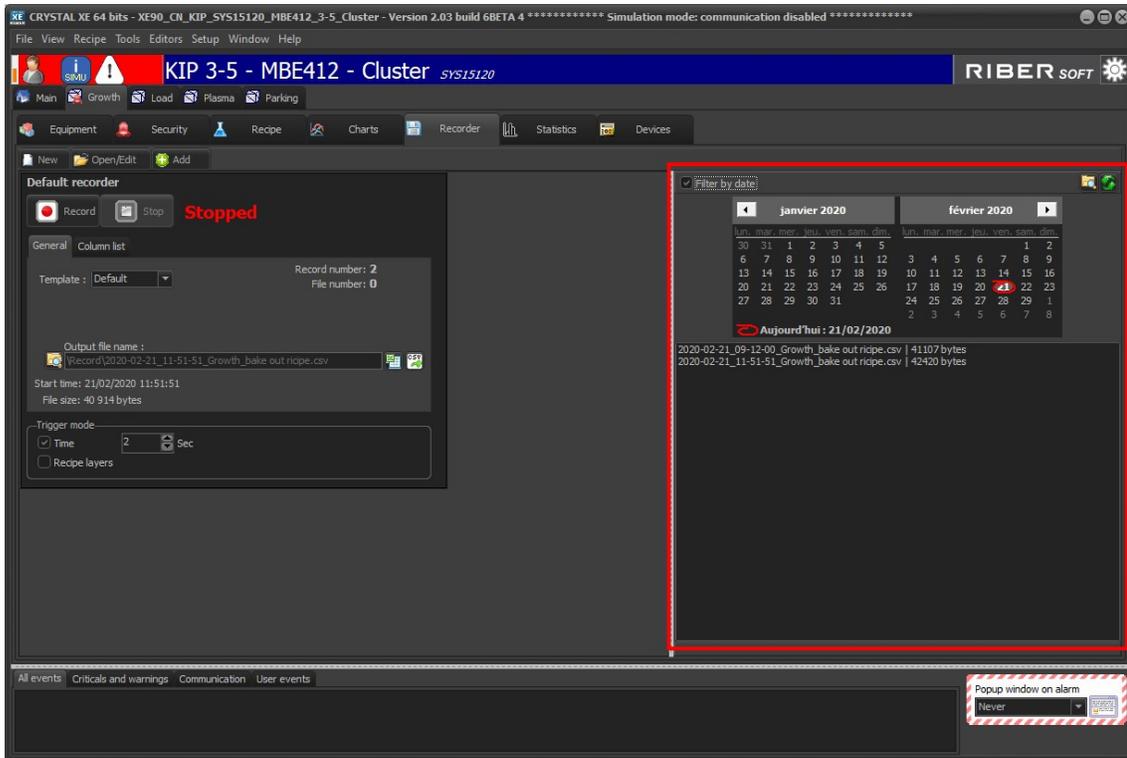
On each sub directory, several files can be found:

- **.rcp** files: XML files created with the recipe editor.
- **.pas** files: Pascal files created with the script editor.

Record directory

Contains all the CSV files recorded by the recorders of each chambers.

This directory can also be browsed in the recorder tab of each room, on the right side of the window.



Browse the CSV files from the Crystal interface

Script directory

This directory contains users script files that can be used in other scripts. These scripts are generally use as libraries of functions and / or libraries of constants.

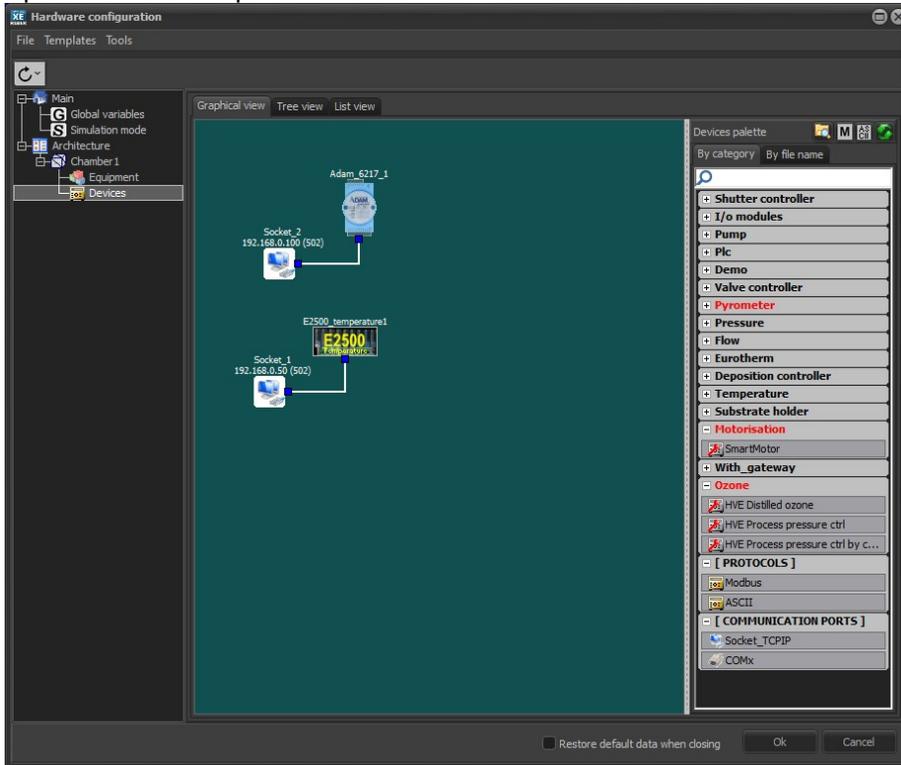
Template directory

The template files that are in the project directory are similar to those in the program directory, but they are specific to the projects.

If the same template file exists in both the program folder and in the project folder, the one in the project folder will be loaded (priority for project files)

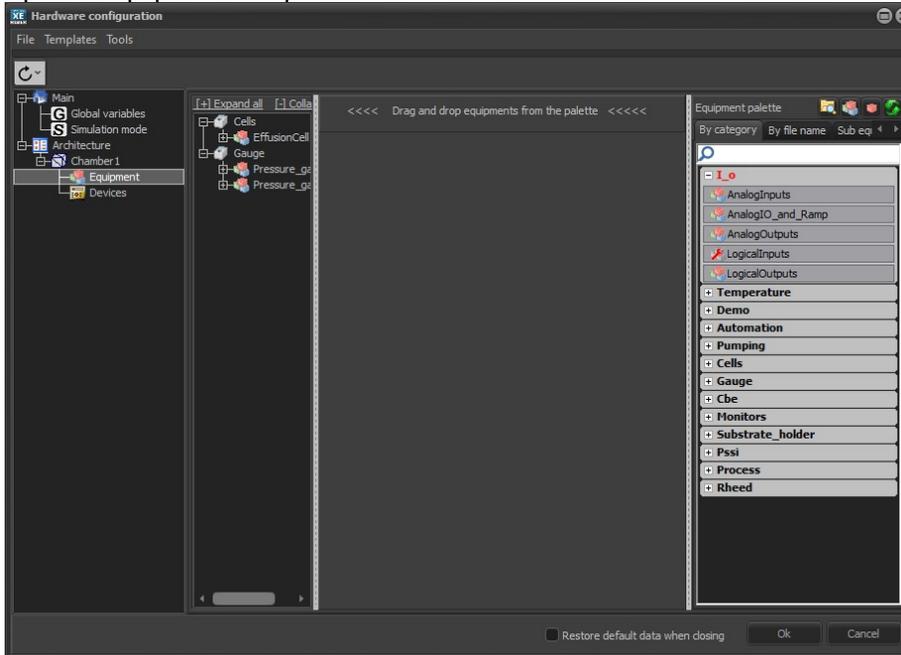
The files in this folder appear in red in hardware config.

Special device templates:



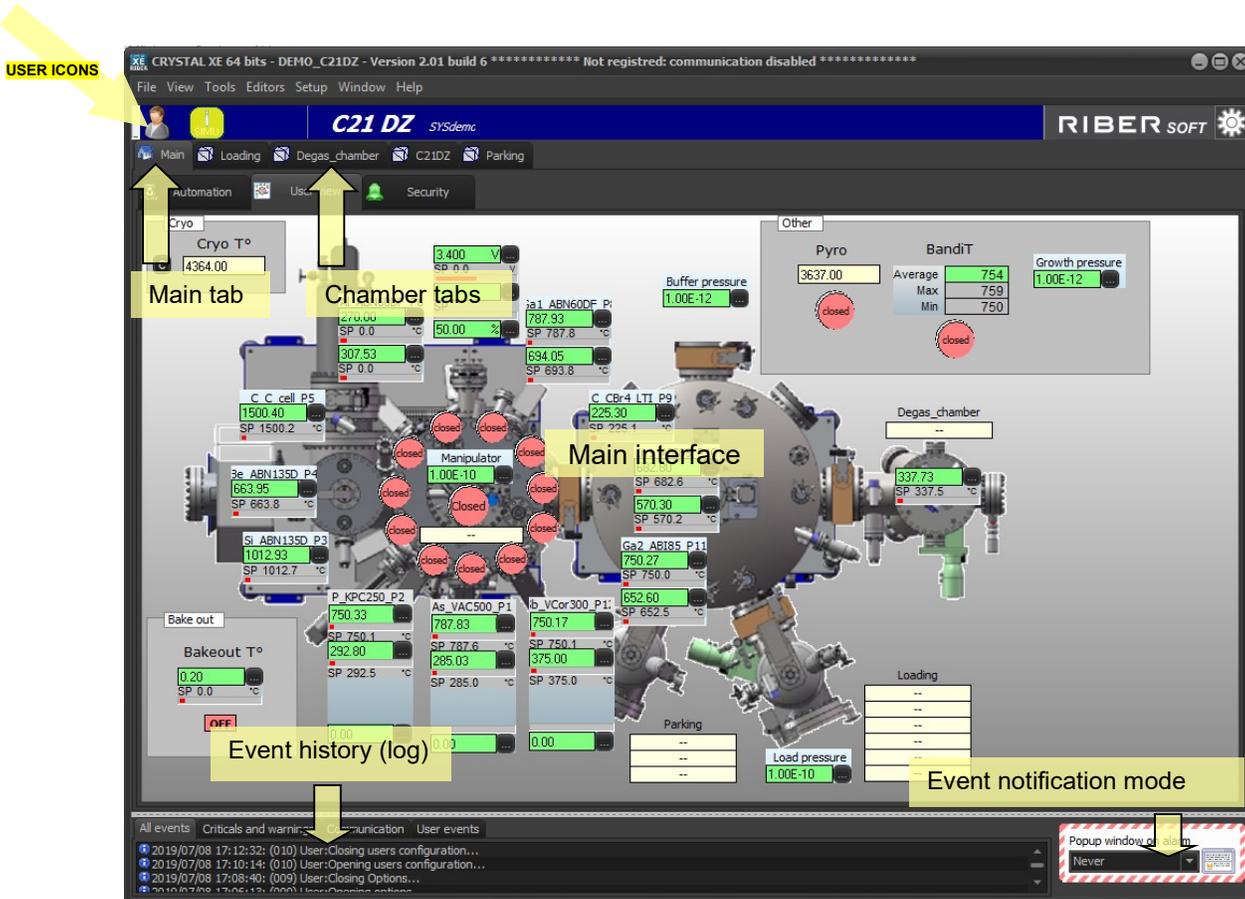
In red are the device templates located in the project directory.

Special equipment templates:



In red are the equipment templates located in the project directory.

5. INTERFACE OVERVIEW



5.1. Icons overview

5.1.1. User icon

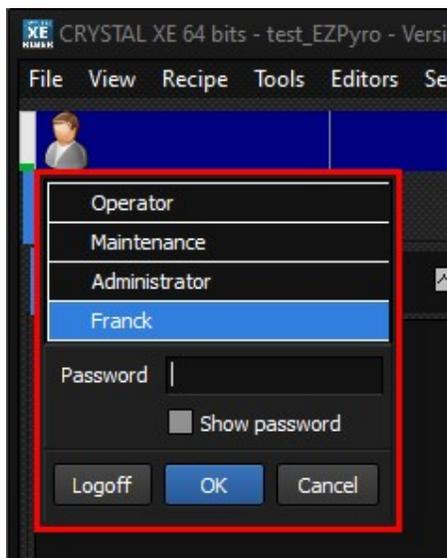
Click on the following icon to display the list of users and log in (if user restrictions are enabled):



The following icon indicates that you are logged in or that user restrictions are disabled:

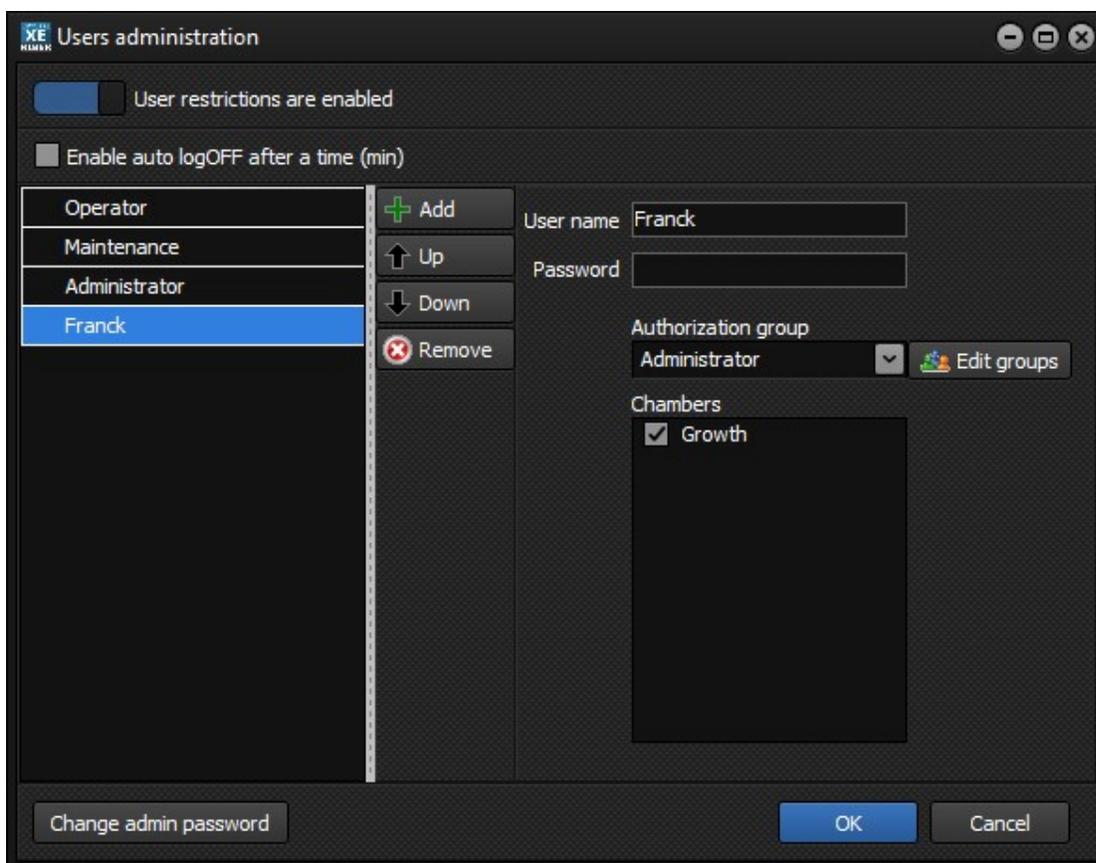


Click on the icon to login:



Select the user and enter the relative password.

You can open the *Users administration* window by right-clicking on the user icon and selecting *Manage users*.



See section about users for more information

5.1.2. Simulator icon

The following icon indicates that *Crystal XE* is operating in simulation mode.



In simulation mode, all communications with peripheral devices are disabled and some values are simulated.

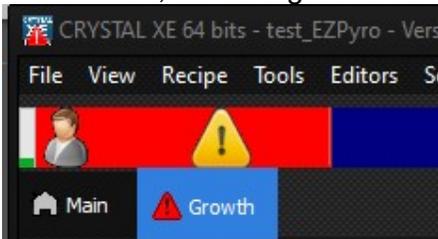
You can switch from *Simulation mode* to *Connected mode* by checking the corresponding box in the *Project configuration* window (the current user's rights must allow this change). On the menu bar, select *Setup > Hardware configuration* to open the *Project configuration* window.

5.1.3. Alarm icon

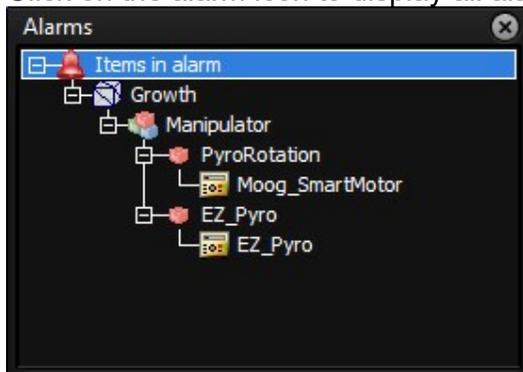
The alarm icon indicates that at least one alarm is activated throughout the program.



In that case, the background color becomes red:

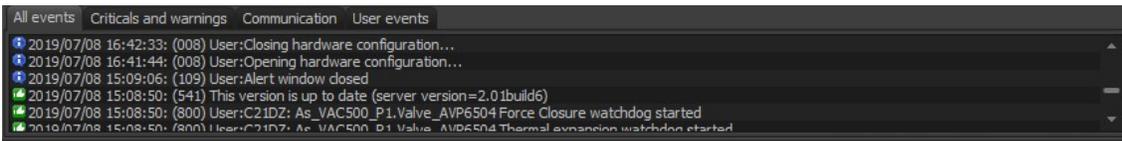


Click on the alarm icon to display all alarms in a pop-up window.

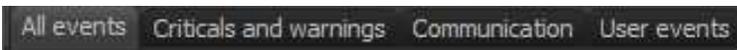


5.2. Event history

The history box, at the bottom of the interface, displays all events (logs). Events are as daily text files (with '.log' extension) to your program directory > *Log*.



You can choose to display only a specific type of events by using the filter tabs:



- Critical and warnings: security agents related events (alarms)
- Communication: connection to the devices

- User events: actions performed by users

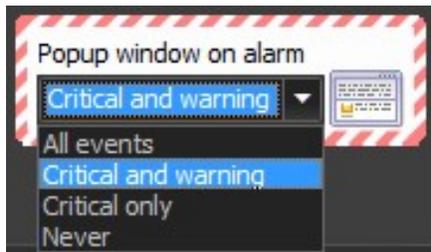
To delete the history of events, right-click on the history zone and select Clear all logs.

5.2.1. Log viewer

Refer to the Menu “Tools / log viewer” for more details.

Event notifications

Use the combo boxes located in the bottom right-hand corner of the interface to define when users should be notified by a pop-up window or by an email alert (available only if the email option is enabled).



Use the corresponding drop-down list to select your choice:

All events: enables Email alerts/pop-up windows for all events (critical and warning alarms, user events, communication alarms)

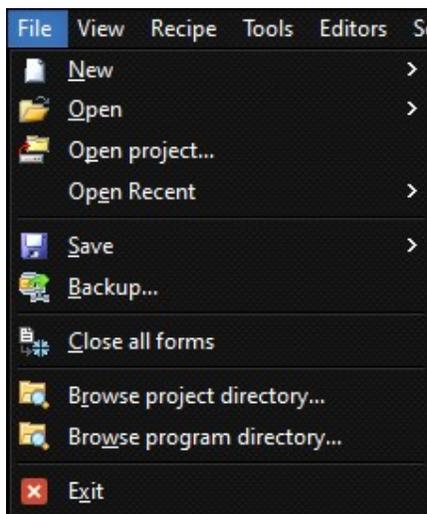
Critical and warnings: enables Email alerts/pop-up windows for critical level or warning level alarms.

Critical only: enables Email alerts/pop-up windows for critical level alarms only

Never: disables all Email alerts/pop-up windows

6. MENU BAR FEATURES

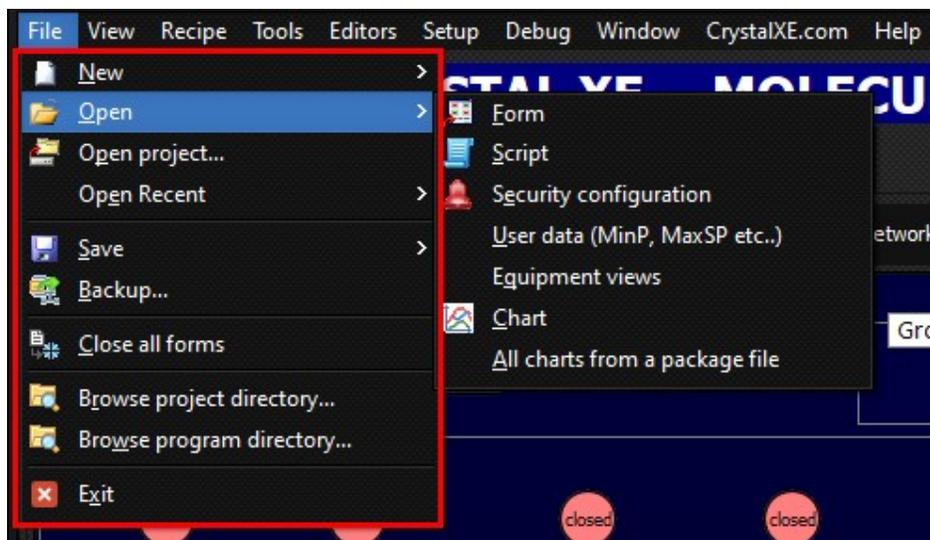
6.1. File menu



On the menu bar, the *File* menu provides access to file related features such as opening new *Files*, saving your configuration, etc.

File/New: Allows you to create a form, a script or a new project.

6.1.1. File/Open



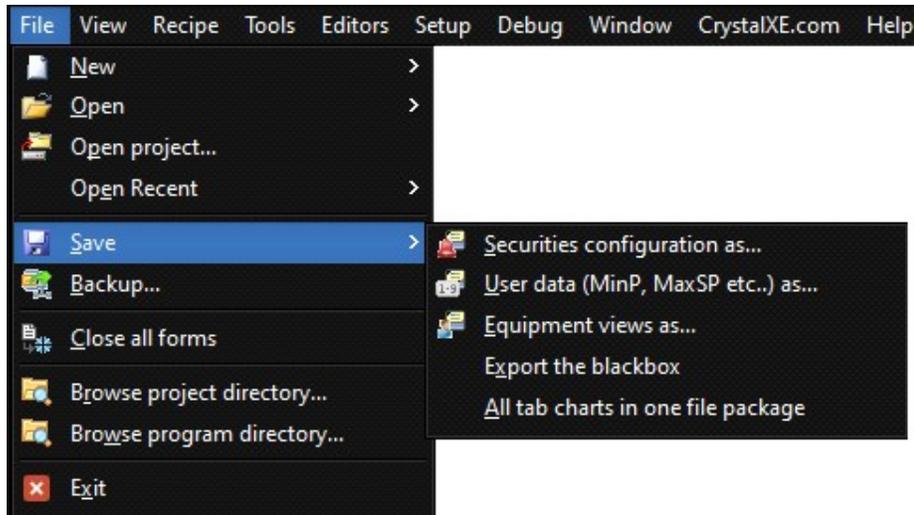
Allow you to open:

- A form to display it.
- A script file to edit it.
- Load security configuration: will replace the current security configuration.
- User data: load a previously saved data file (for more details, see the chapter “File descriptions”)
- Equipment views: load a .ced file (for more details, see the chapter “File descriptions”)
- Chart: load a chart file.
- All charts will load a packed.cpk file which contains all the charts of the chart tab of each chamber.
- A Project: you must specify the directory where is located all the project files (and especially the .xfg file).

6.1.2. File / Open recent:

This menu displays the previously loaded files and projects. A line separates file names and project names.

6.1.3. File/Save:



Allows to save:

- The current **securities configuration** in a .sec file
- The current **user data** in a .dat file
- The current **Equipment view** in a .ced file.
- Export the **black box**: this option is only useful if you have enabled the black box in the option. The black box contains all the communication frames of all the device for a time. This can be useful to debug communication problems.
- **All charts**: save the content of all the charts of all the chart tab in a .cpk file that can be loaded later from the menu File/Load all charts.

6.1.4. Backup:

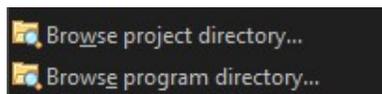
Allows you to create a zip file that contain all or some projects files. For more details, see the **BACKUP** chapter.

6.1.5. Close all forms

Will close all floating windows. Chart opens by the contextual menu “Add in a new chart” are not closed. Same for the forms open from the user view that display details view of equipment or sub equipment, these are not closed.

6.1.6. Browse project and program directories

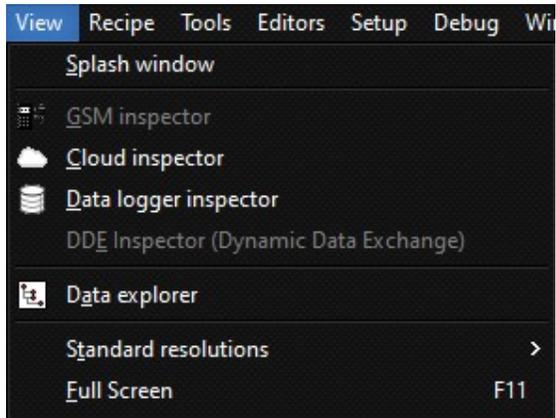
- On the *File* menu, you can access easily to *Crystal XE* program and project directories by clicking on the following items:



6.1.7. Exit

Close the application (after confirmation)

6.2. View menu

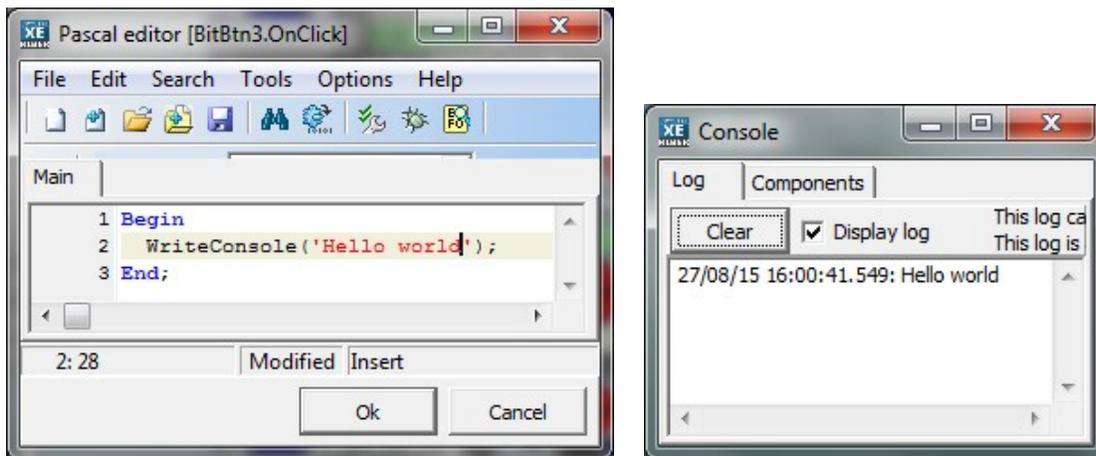


The *View* menu features mainly displays information about the system (scripts, data, connection etc.).

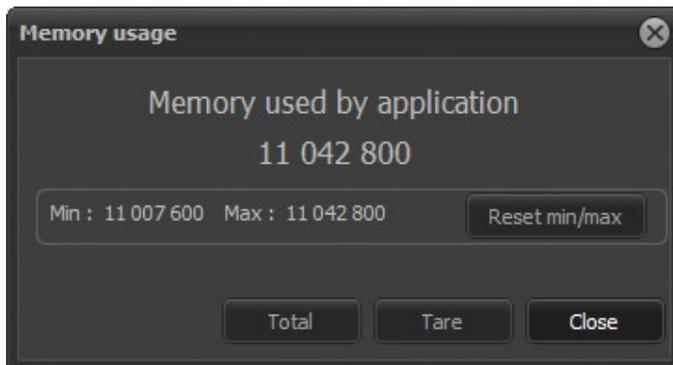
6.2.1. Console

The console is a pop-up window mainly used to debug or give users information about executing scripts.

The console appears automatically when a function *WriteConsole()* is used in a script:



6.2.2. Memory

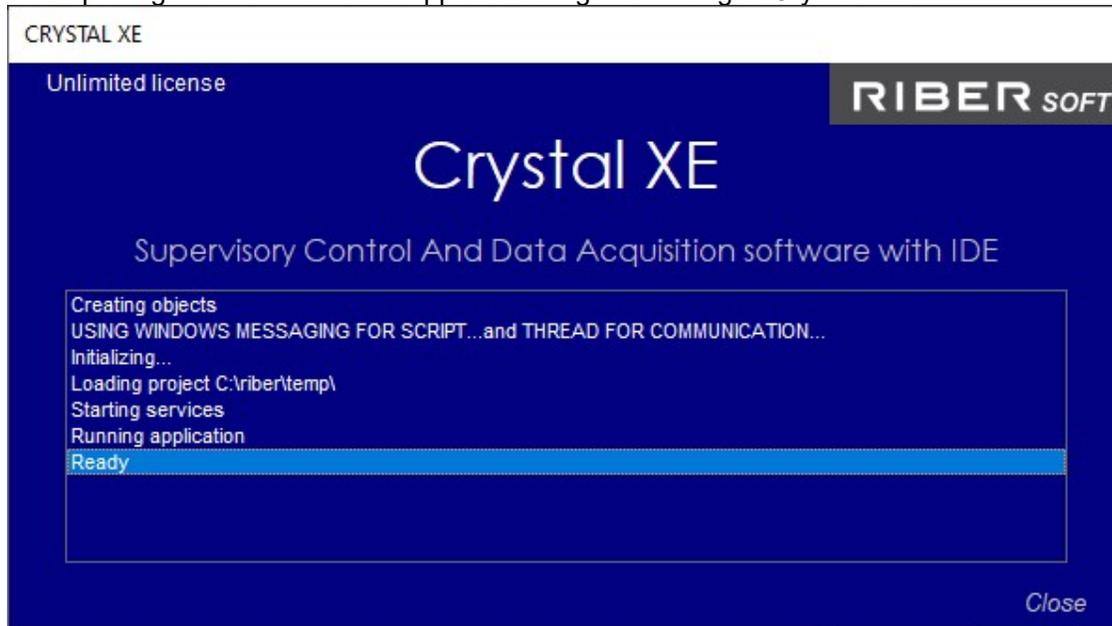


This information display the memory occupation of Crystal XE.

Crystal XE consumes very little RAM memory. This information is useful only in the case where a lot of resources are used in order to follow the evolution. This information is mainly used by Crystal developers to monitor that all tasks do not consume memory unnecessarily.

6.2.3. Splash window

This opens again the window that appears during the loading of Crystal XE.



6.2.4. Diagnostic

Click on the Diagnostic item to display information about your system in a pop-up window including used memory, Email and SMS service, number of equipment in your project, etc.

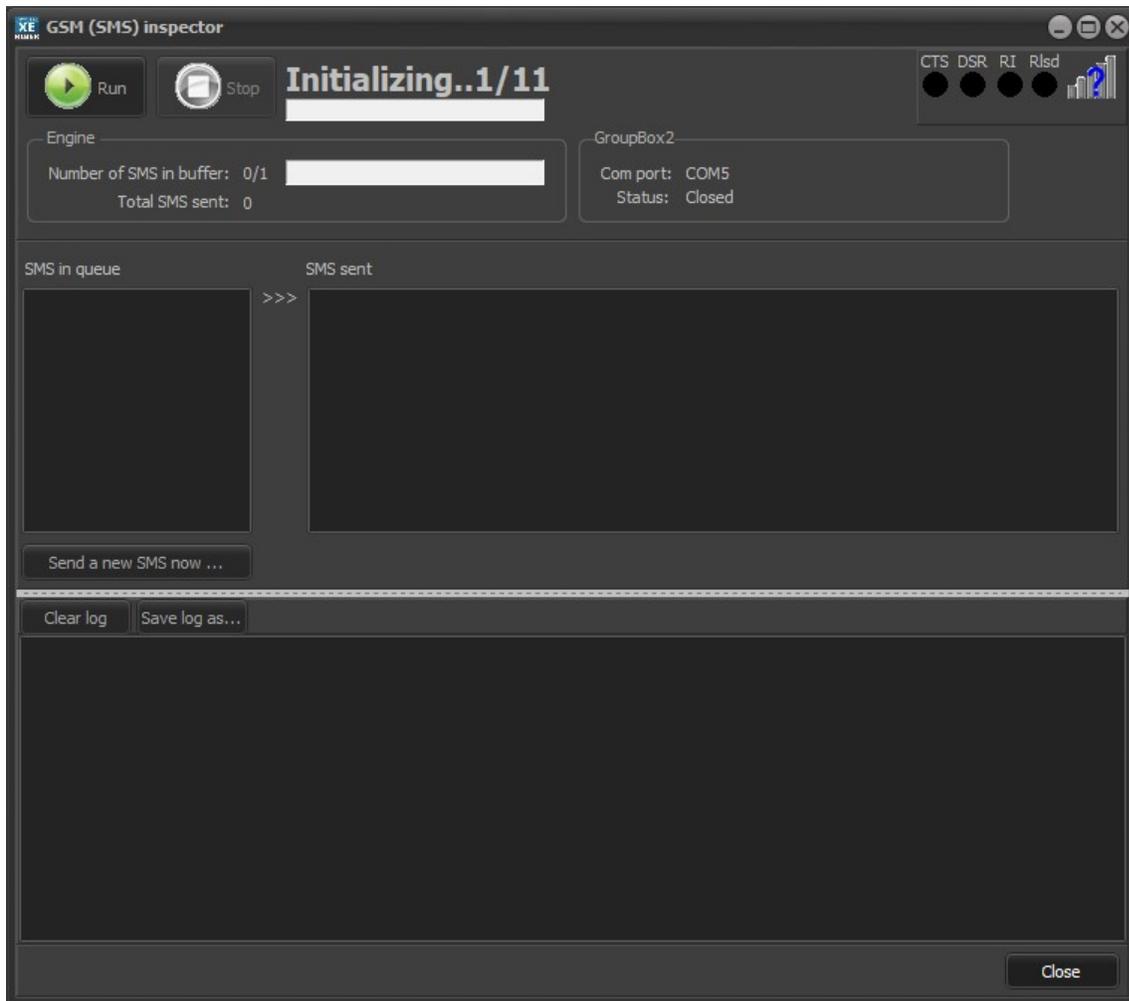
Full screen

Select *Full screen* or press the F11 key to switch from normal screen to full screen mode and conversely.

In full screen mode, press the [ALT] key to show/hide the menu bar.

6.2.5. GSM inspector

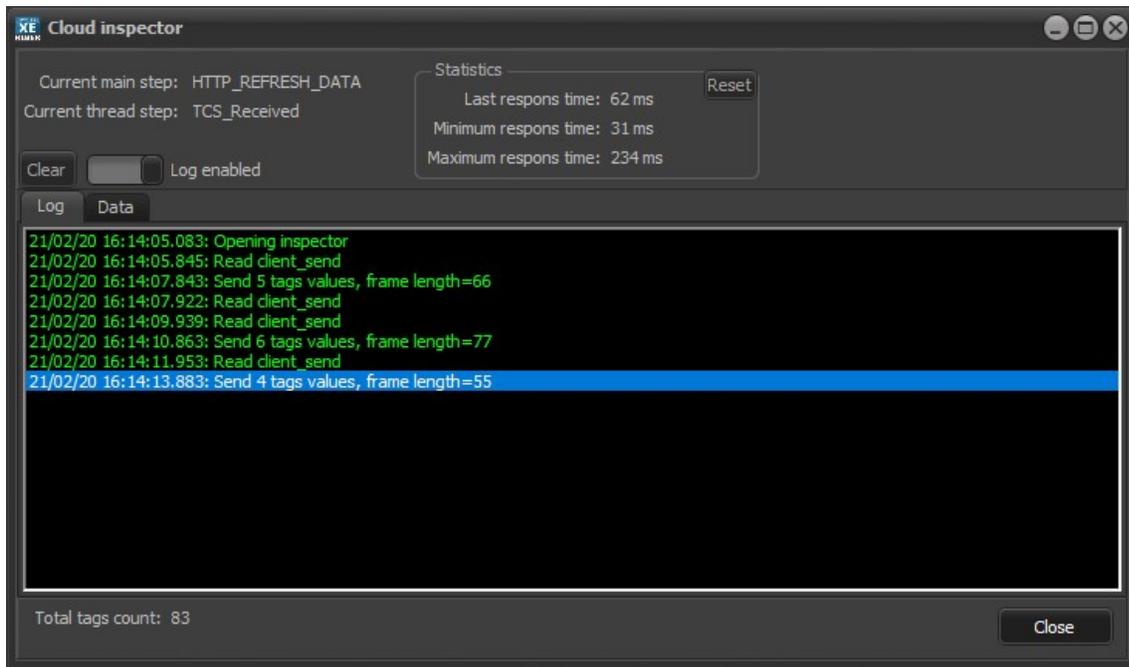
This opens this GSM manager window:



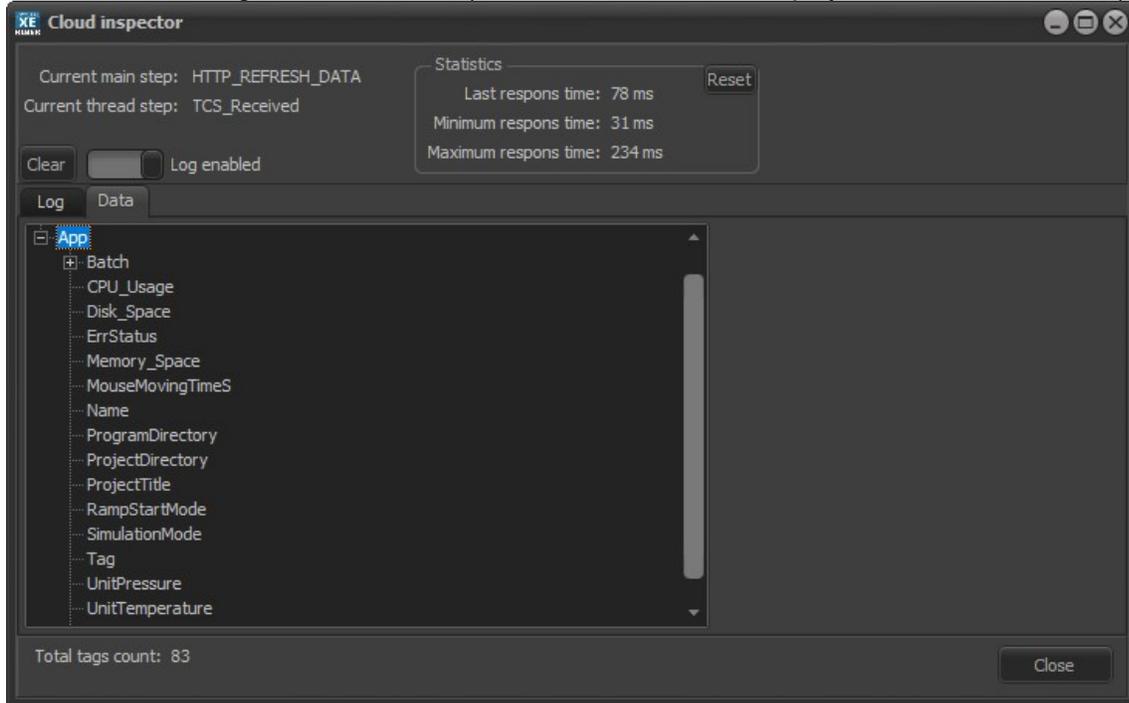
This is useful if you want to test the sending of text messages (SMS) and also to check the communication with the GSM modem.

6.2.6. Cloud inspector

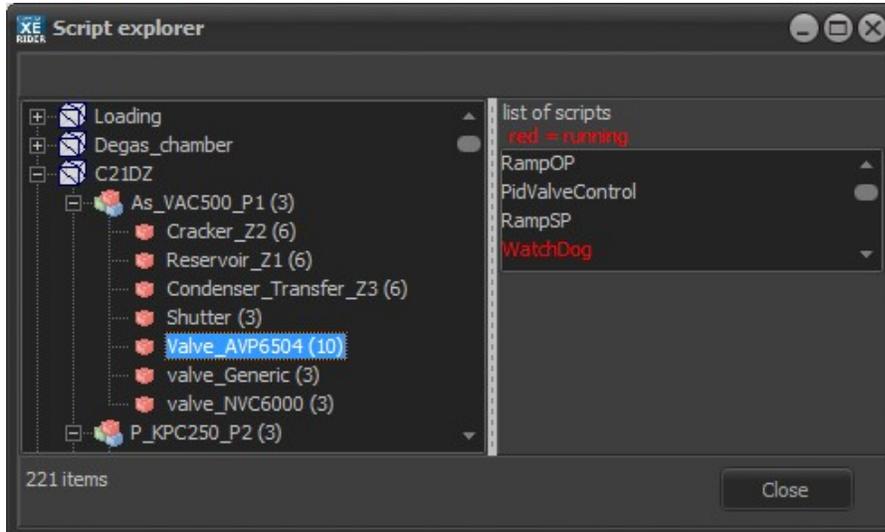
This item opens the following window:



In this window, it is possible to display the communication activity between Crystal XE and the cloud server. This is useful to check the communication with the cloud server. It is also possible to display the data that is exchanged with the cloud (click on the data tab to display the data in a tree view)



6.2.7. Script explorer



All the scripts in your system are displayed in a tree structure on the left side of the window. For each equipment and sub equipment, the number of scripts is displayed in brackets.

Select an item in the tree structure to display the scripts list in the box.

When a script is currently running, it appears in red.

The scripts edited within forms and objects are listed at the bottom of the tree structure:

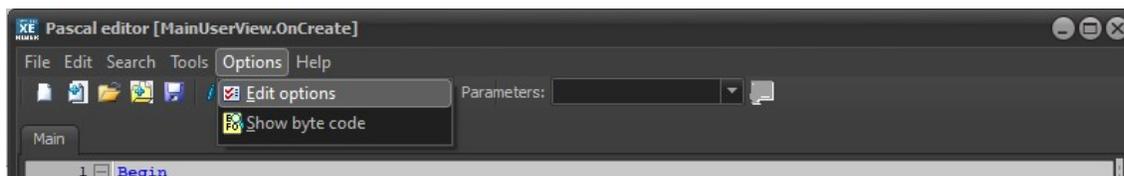
App > Forms > YourForm (>Object):



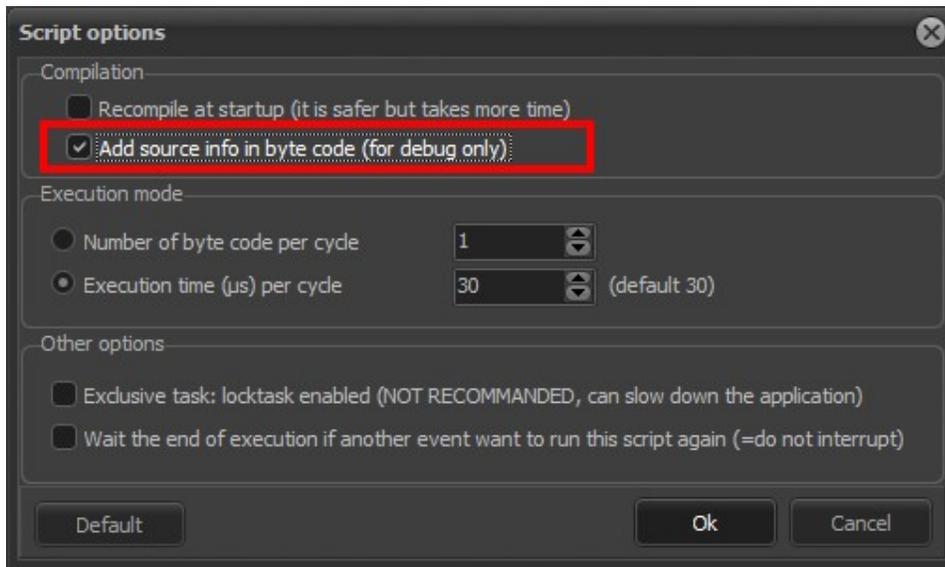
If you want an edited script to appear in the *Script explorer*, you must enable it from the Pascal editor (= script editor).

Remark: The Pascal editor opens when you create a new script or when you edit an existing script. Scripts are either integrated in a visual component event, or as a script recipe, or as a .pas file outside a recipe or on the event of a tag of a equipment or device.

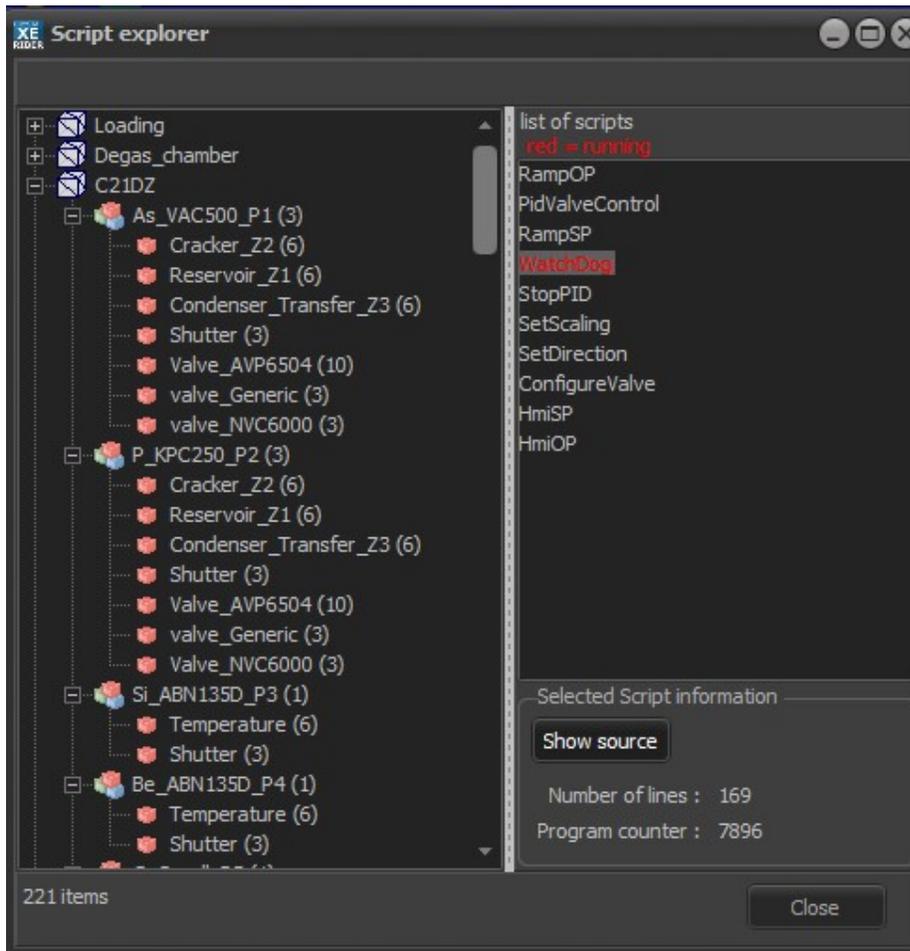
To do so, on the menu bar of the Pascal editor, select *Options > Edit options*:



Then in the pop-up window, check the following box:



Select a script in the box to display the script information:



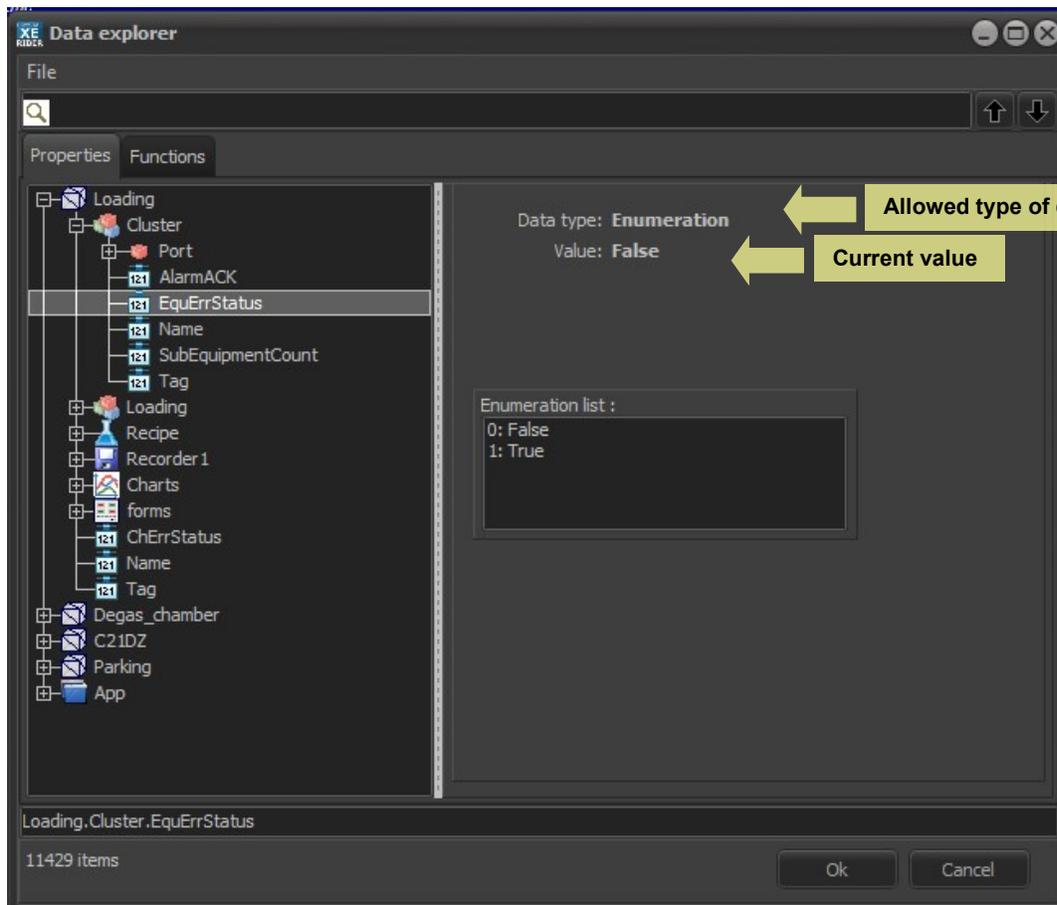
Double-click on a script or click on the Show source button to open the Script debug tool allowing you to monitor the execution of the script.

The debug tool can be automatically opened when a script is executed using the functions: *ShowSource(0)* (recommended) and *ShowSource(1)* (advanced).

```
Begin  
ShowSource(0);
```

6.2.8. Data explorer

The *Data explorer* lists, in a tree structure, all data of your system. In the tree structure, expand the lists and select a property. The information of the selected property appears on the right side of the window.



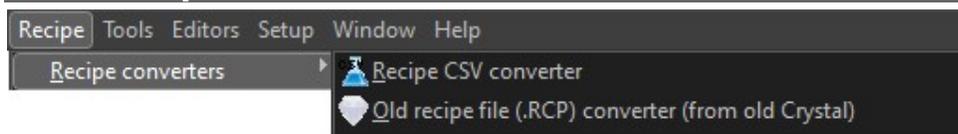
Depending on the property, different types of data can be stored:

Numeric value: measured value, output power value, etc.

Enumeration: checked/ unchecked, on/ off, etc.

Texts (strings): equipment name

6.3. Recipe menu



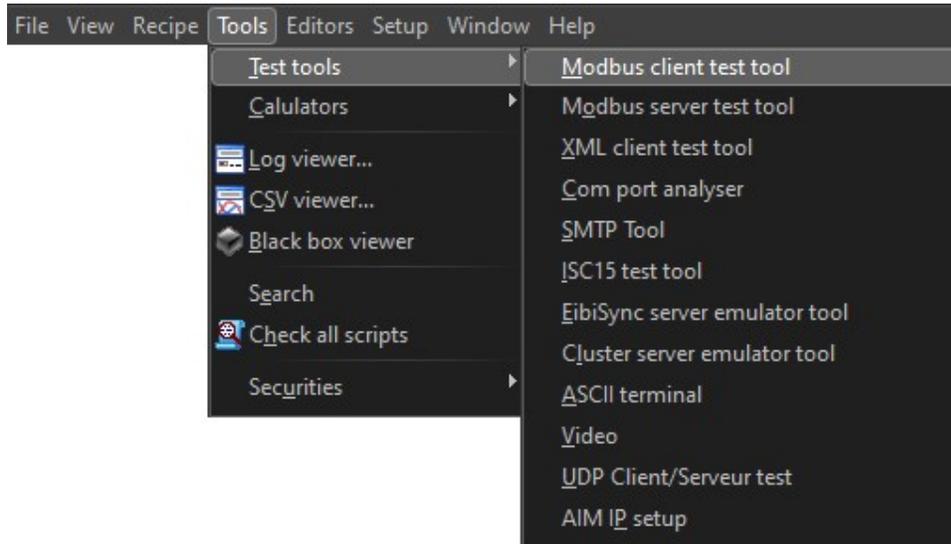
From this menu you can convert recipe in CSV format to rcp files and also convert old crystal recipe files (From Crystal Eyes) to Crystal XE recipe files.

⇒ For more details see the section **Recipe converters in the RECIPES chapters.**

6.4. Tools menu

The *Tools* menu offers several debugging features allowing you, for example, to test the connection and behavior of devices.

6.4.1. Test tools

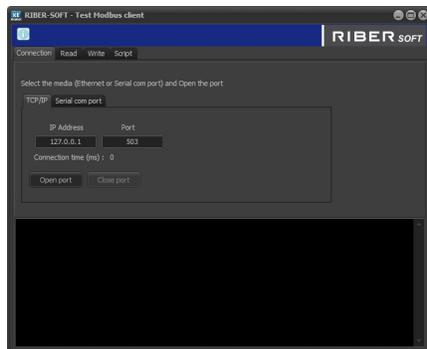


▪ Modbus client

The Modbus client test tool allows you to test the connection to a Modbus device either using TCP/ IP or serial communication mode.

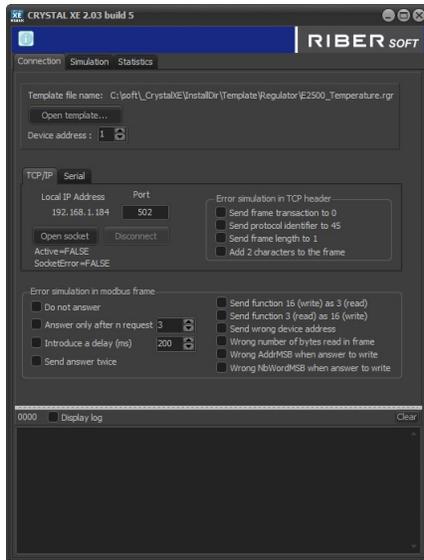


The communication port must not be opened when performing the test. If you want to test on a serial port that is already open in Crystal, you must first deactivate it from the Device tab.



▪ Modbus server

With the Modbus server test tool, you can select a template and emulate a device (regulators, etc.) in order to debug. Device tags appear in a table and you can simulate different values.

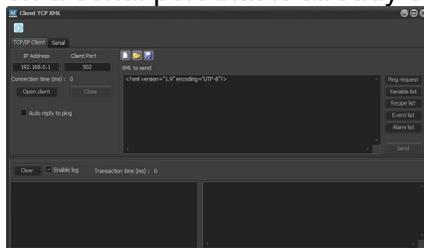


■ XML client

This test tool allows you to test the connection to an XML server either using TCP/ IP or the serial communication mode. You can send standard frames such as “Ping request”, “Variable list request”, “Recipe list request”, “Event list request”, “Alarm list request” or any other personalized message.



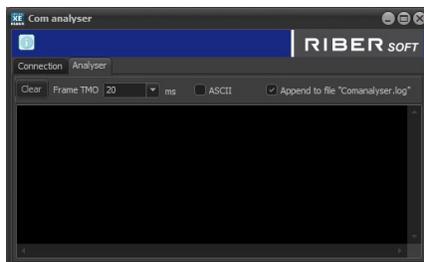
The communication port must not be opened when performing the test. If you want to test on a serial port that is already open in Crystal, you must first deactivate it from the Device tab.



■ Com port analyzer

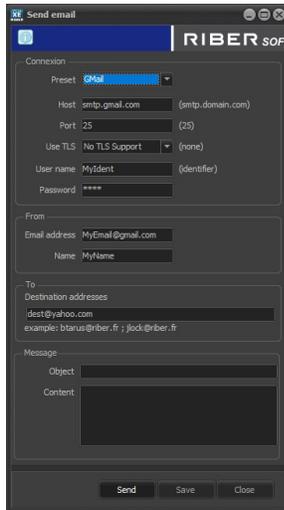
Using appropriate cable, the *Com port analyzer* allows you to analyze two serial ports simultaneously.

The tool displays frames that are sent by *Crystal XE* to the device (PORT A) as well as frames that are sent by the device to *Crystal XE* (PORT B) at the same time.



■ SMTP tool

The *SMTP tool* allows users to directly send an Email to one or several addresses using any mail server and test email delivery.



- **ISC15 tool**

The *ISC15 test tool* allows you to emulate ISC15 former *RIBER* shutter controller.



- **EibiSync server emulator**

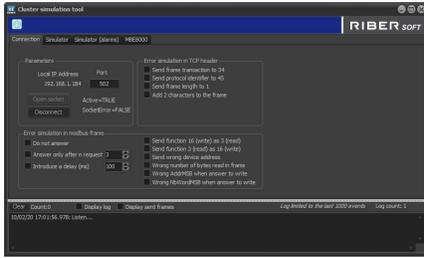
Use *EibiSync server* tool to emulate a device using EI BiSync protocol either using TCP/ IP or serial communication mode.



The communication port must not be opened when performing the test. If you want to test on a serial port that is already open in Crystal, you must first deactivate it from the Device tab.

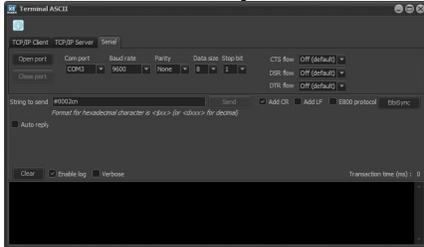
- **Cluster server emulator**

You can use this tool to emulate a cluster server.



■ ASCII Terminal

ASCII tool allows you to send ASCII frames to any serial device and displays the response.

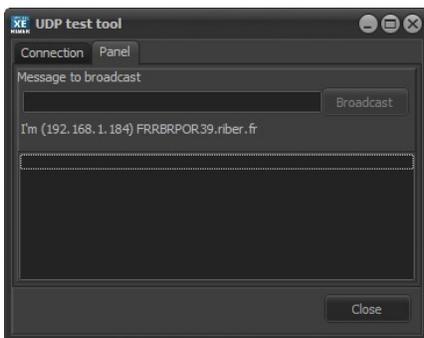


■ Video

Use the *Video tool* to test video connection and display.

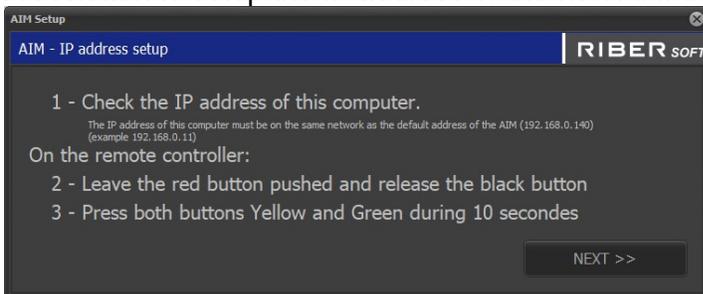
6.4.2. UDP Client/Server test

This tool can be used to send broadcast messages over the local area network using the UDP protocol. The received data is displayed in a memo for analysis.



6.4.3. AIM IP setup

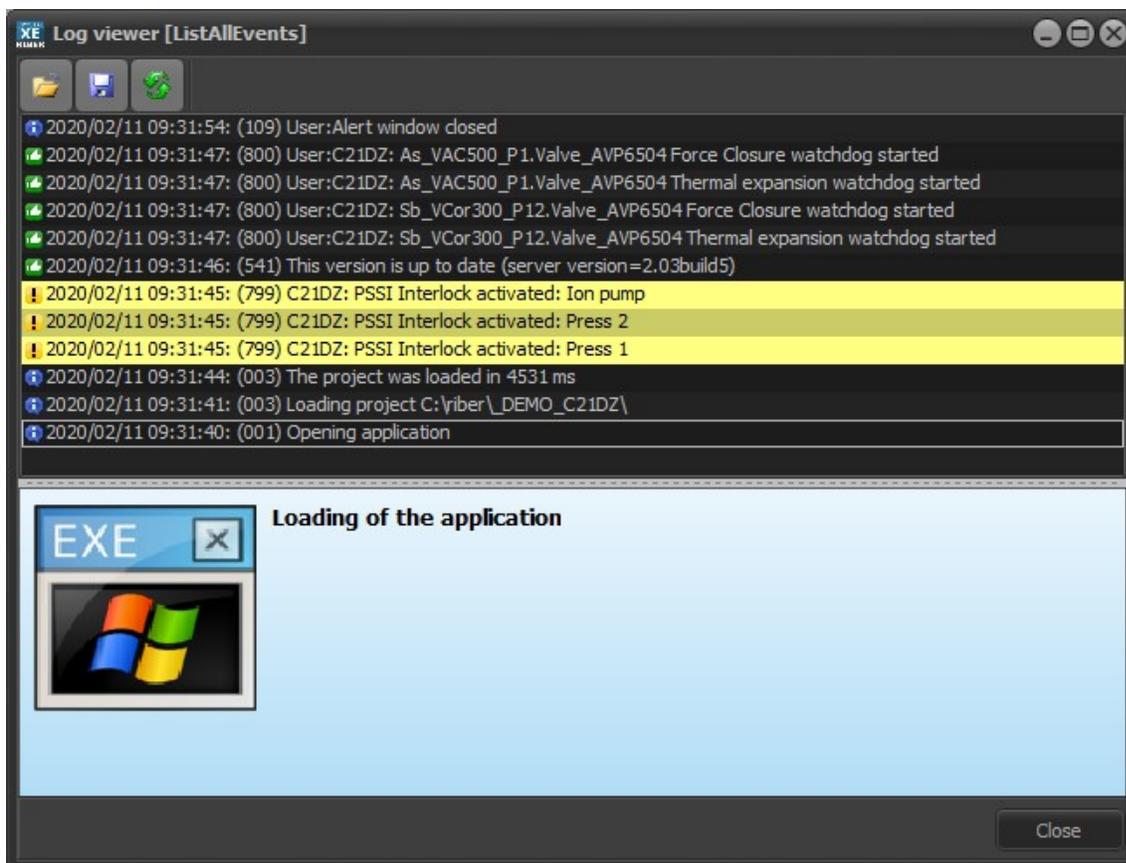
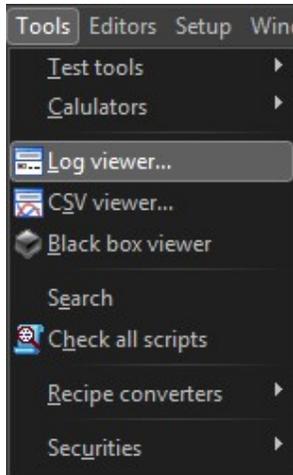
A assistant to setup the IP address or the new AIM device.



6.4.4. Log Viewer

In the history log (at the bottom of the main Crystal XE screen), double-click on any of the events or right-click on it and select *Open in a new window* to open the *Log viewer*.

On the menu bar, select *Tools > Log viewer* to access the *Log viewer* window.



The *Log viewer* displays log details. When you open the *Log viewer* window, logs are displayed as in the history box (logs that have been cleared in the history box do not appear in the *Log viewer*).

You can save the logs under a specific name by clicking on the following icon:

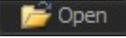


You can display any log file by clicking on the file icon  and selecting a file to open.

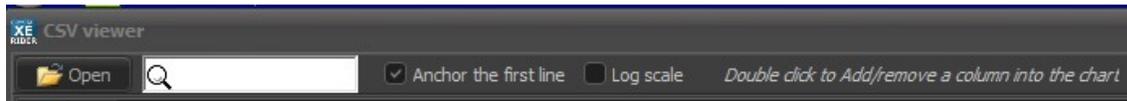
6.4.5. CSV viewer

The *CSV viewer* allows you to display data recorded by the system recorder saved as *CSV Files*, represented in table and chart.

The recorded data is displayed in a table. Each row corresponds to a period of time (or a layer) according to the defined trigger mode. Each column corresponds to a property whose values have been recorded.

Open a CSV file by clicking on the Open button located in the top left-help corner of the window: 

By default the first line of the CSV file is always displayed (title of the columns). To hide the first line, unselect Anchor the first line:

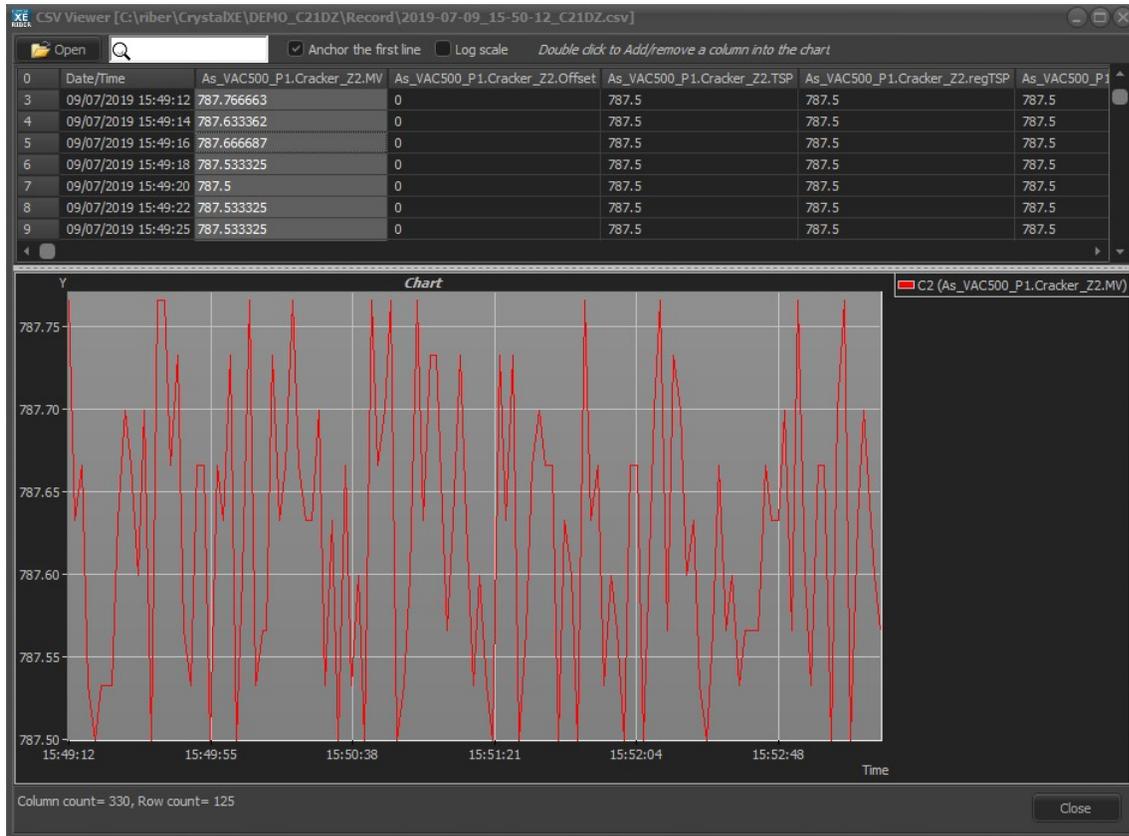


You can view one or several properties values on a chart in the lower part of the window. Each property corresponds to a curve on the chart.

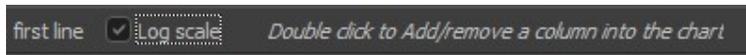
To display or hide a curve, double-click on the property column in the table.

When a curve is displayed, the column is highlighted in yellow.

The curves legend appears on the left side of the chart. Double-click on a curve in the legend to hide/display the curve.



By default, the chart has linear scale. Check the Log scale box to use logarithmic scale:



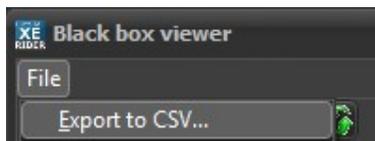
If you open a record file when the recorder is still recording, only data that has been recorded before opening the CSV viewer will be displayed.

6.4.6. Black box viewer

The *Black box* is the *Crystal XE* background circular recorder. The *Black box viewer* displays files saved by the *Black box*.



The *Black box* option must be enabled to record files. To enable the *Black box* go to *Setup > Options > General* and check the box:

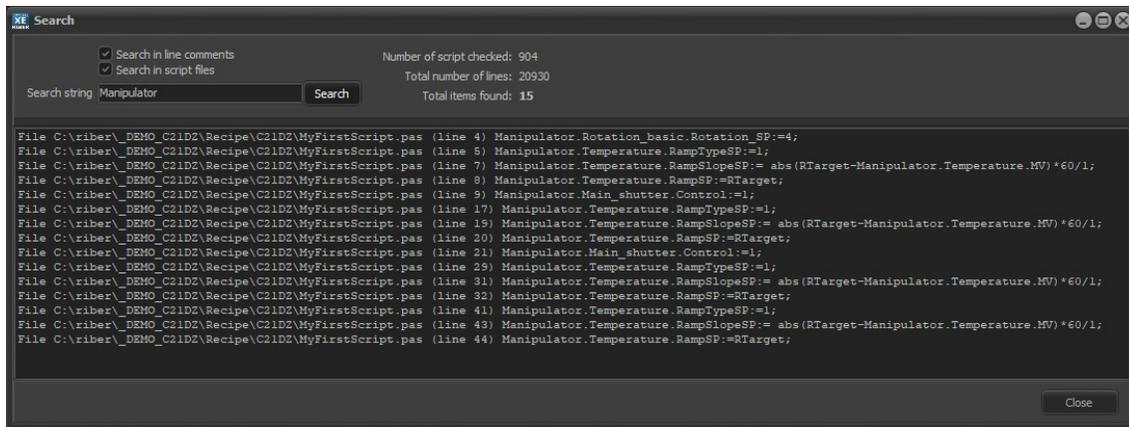


You may need appropriate user right to perform this action.

6.4.7. Search

Do a string search in *Crystal XE*.

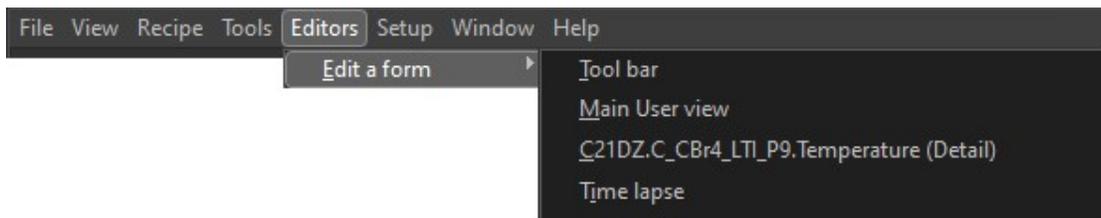
This functionality is useful if you want to know if a tag is used, in which script and when it is used.



6.4.8. Check all scripts

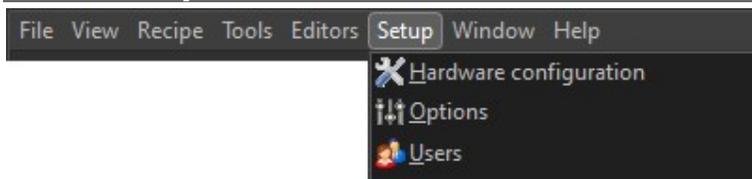
The script verification tool analyzes all the scripts in the project and indicates if there is an error. This tool does not check files on the disk unless they are used by another script. Only scripts already loaded are checked.

6.5. Editors menu

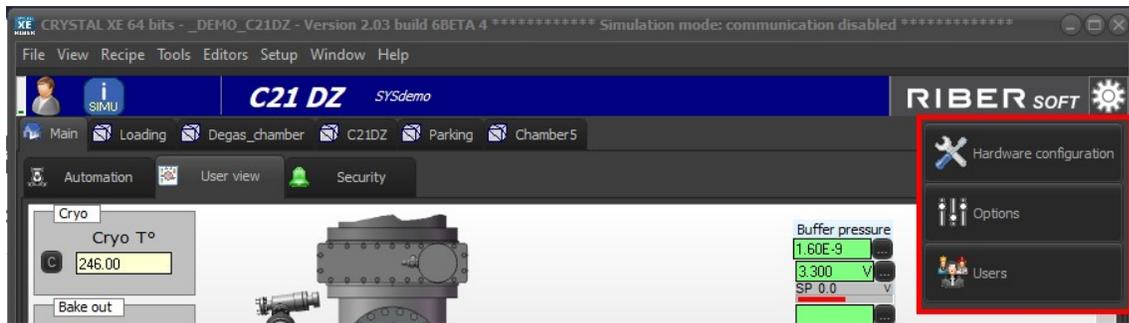


From this menu, you can switch a form to design mode to modify its content. This is useful when in some cases the right-click context menu for editing the form does not appear.

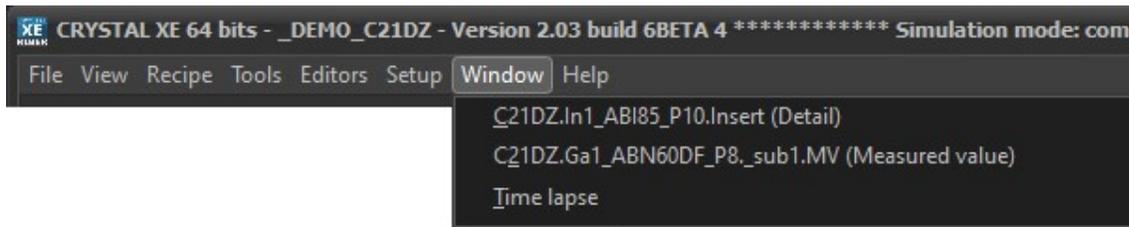
6.6. Setup menu



The setup menu allows to access to the hardware configuration, the options and the users. This is another way to access to the menu located on top right of the application:

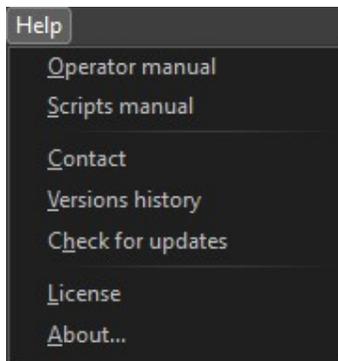


6.7. Window menu



This menu depends on the number of windows that are open. The name of each window that is opened appears here. By clicking on one of the names, if the window was minimized, it is restored. This is useful for finding a lost window.

6.8. Help menu



6.8.1. Operator manual

Open the operator manual.
This is the online help which is also accessible by the HELP button in each of the equipment.

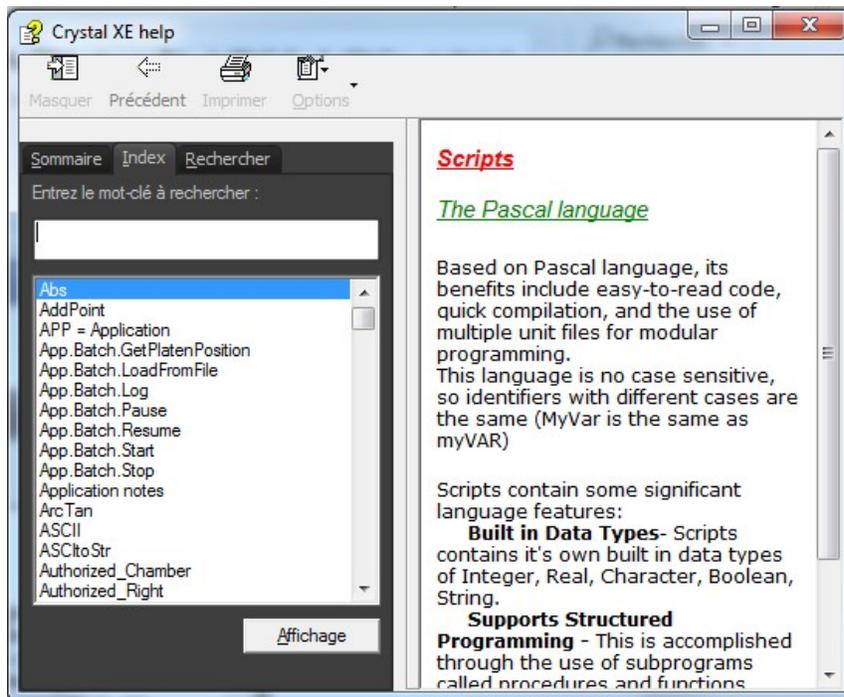
6.8.2. Script manual

This is the online help which is also accessible by the F1 function button in the script editor.

On the Help menu, click on the Scripts manual to display Crystal XE help window that mainly provides information about Crystal XE scripts.

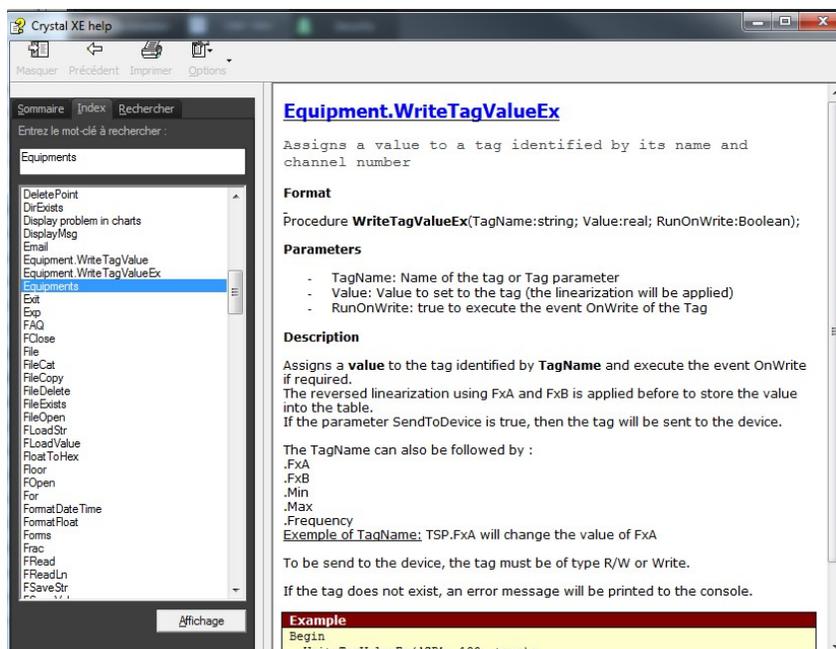
On the tree structure, expand the lists and selected the desired item to display information on the right side of the window.

Expand the Script list and select The Pascal language to find general information about Pascal Language. Click on the items to navigate through the sections:



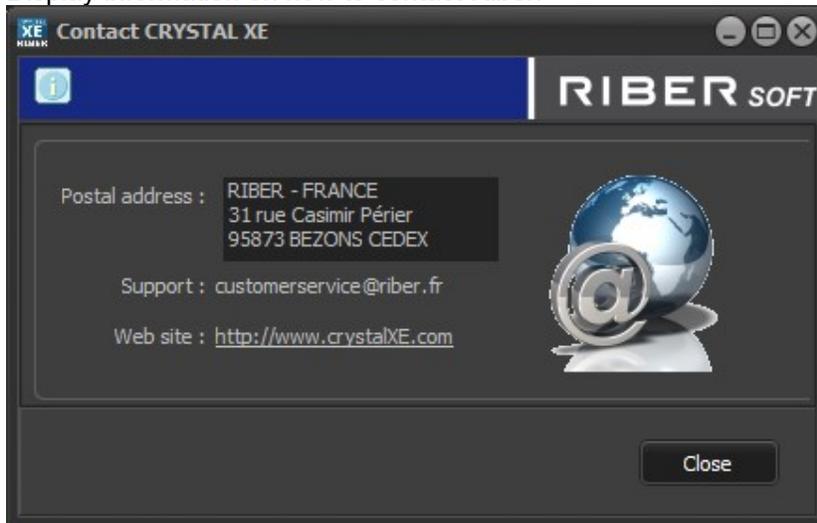
The functions used in Crystal XE, are listed in a tree structure. When editing scripts, you can refer to the information provided.

Expand Functions reference, then expand the groups and select a function to display the information.



6.8.3. Contact

Display information on how to contact Riber.

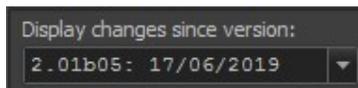


The best way to contact our customer service is to use email.

6.8.4. Crystal XE versions

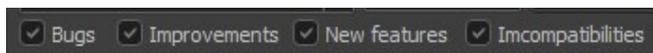
Select *Versions history* to view new features and corrections made to *Crystal XE* since the first version:

Select your version from the drop-down list:



The history displays the modifications that have been made to the selected version.

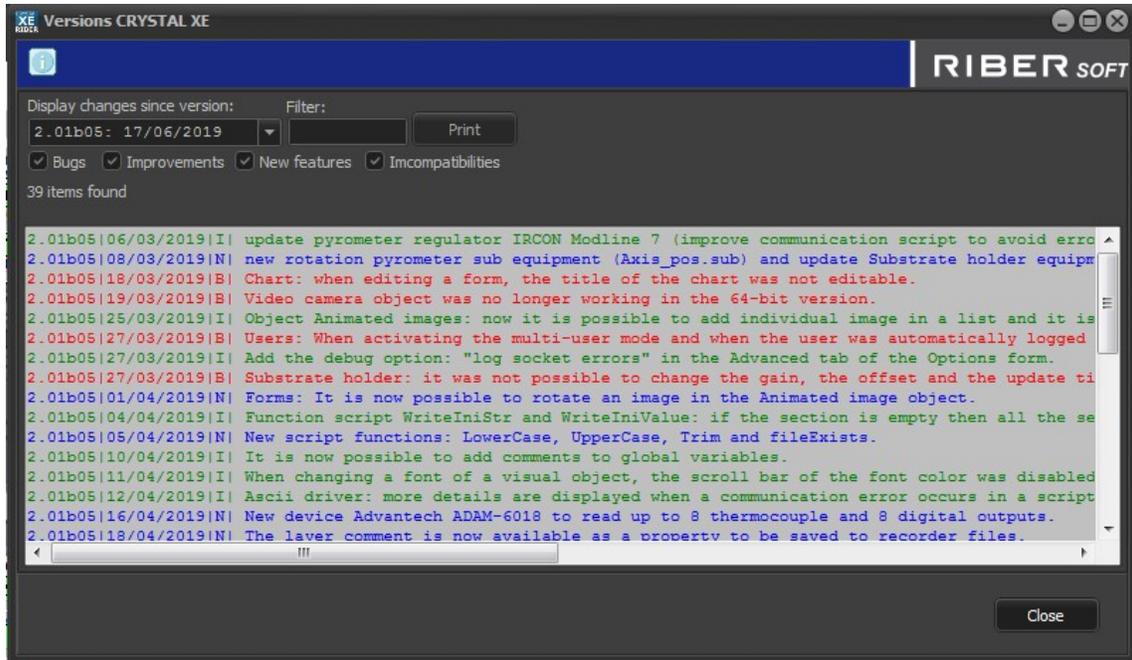
Check/uncheck the boxes to filter modifications made in Crystal XE: fixed bugs (red), improved features (green), new features (blue) or incompatibilities with older versions (pink):



You can also filter modifications made to a specific feature of the software by entering text in the text box:

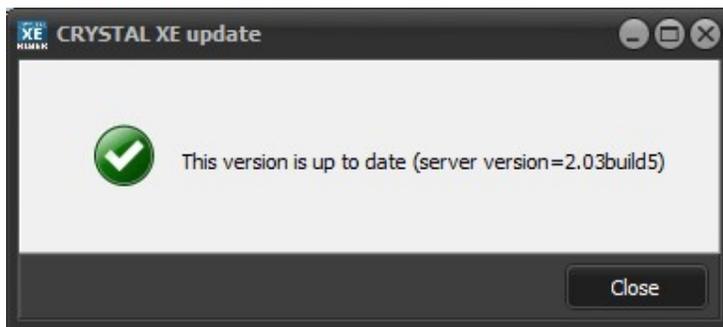


Example: viewing recipe related modifications:

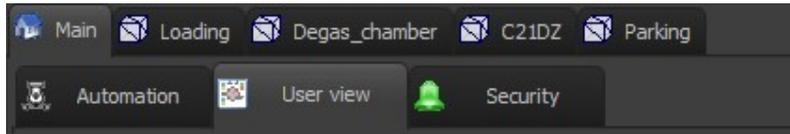


6.8.5. Check for update

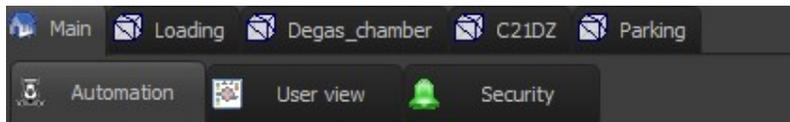
This functionality required a Internet connection.



7. MAIN TAB



7.1. AUTOMATION

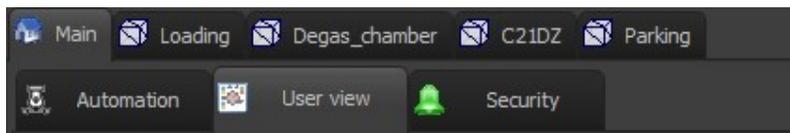


The *Automation tab* allows you to create, edit and execute production batches.

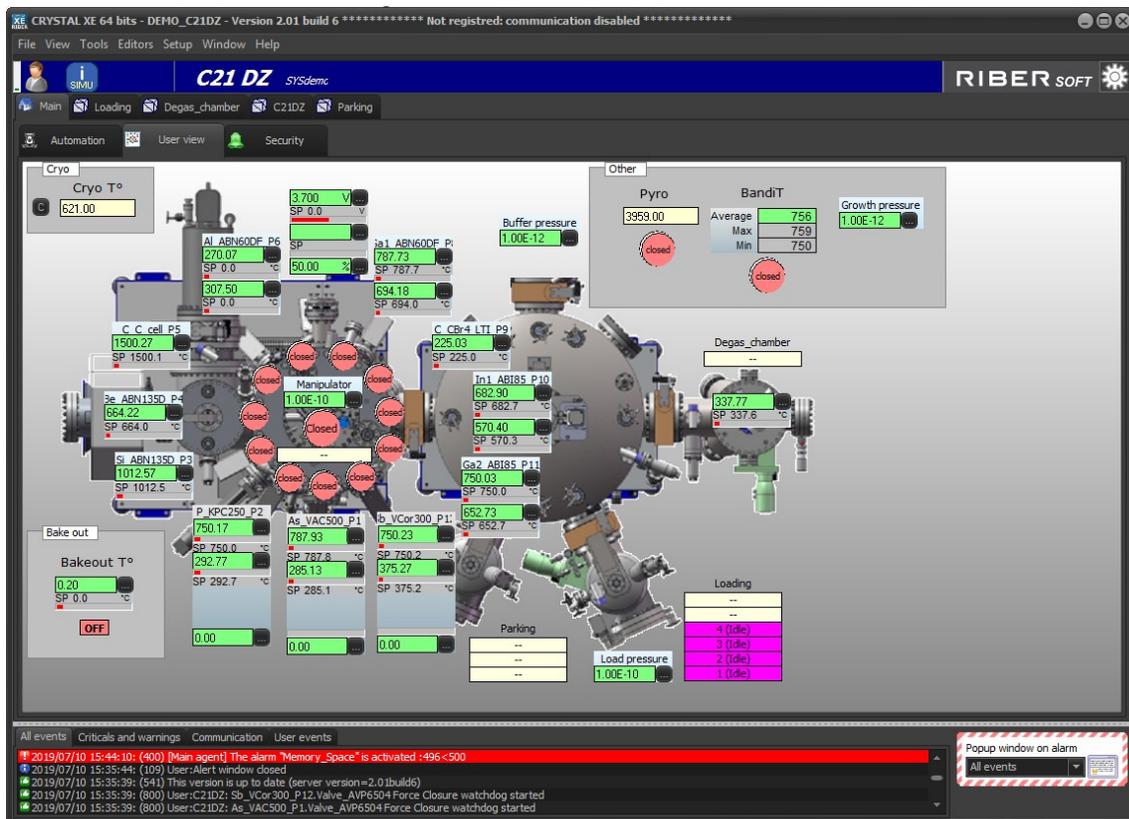
A batch is used to organize every operation performed by the different parts of the epitaxy system, either in parallel or in sequence. A 'batch file' contains the description of the platens' movements in the system.

→ For more details about the creation and execution of batches, please refer to the chapter [Platens automation](#) in this manual.

7.2. USER VIEW



The *User view* interface is a visual representation of the MBE system allowing users to monitor and control the functioning equipment.



The *User view* interface depends on your configuration.

The *User view* is a form (synoptic view) composed of several objects representing your MBE system: equipment, chambers, shutters (confirmation buttons), etc.

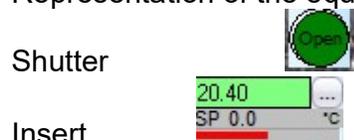
You can customize the *User view* interface using the *Tool palette*: right-click the interface background and select *Edit form* to open the *Tool palette*.

→ For more details about the edition of forms, please refer to the chapter [Forms and objects](#) in this manual.

7.2.1. Example of equipment representation



Representation of the equipment *Ga2_ABI85_P11* and its three pieces of sub equipment:





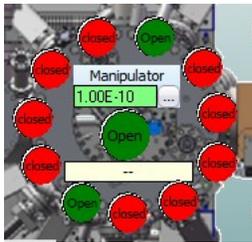
The interface allows you to perform the following actions:

- Opening/closing shutters 
- following the measured value 
- viewing the set point value 
- following the output power bar graph 
- opening the Detail view window 

→ For more details about equipment objects and shutters (confirmation buttons), please refer to the chapter **Forms and objects** in this manual.

7.2.2. Interacting with the equipment

Opening/closing shutters



You can open/close a shutter by double-clicking on the desired shutter icon. Green icon indicates that the shutter is open while red icon indicates that it is closed. As any confirmation button, when you first click on the shutter icon, a progress bar appears over the icon:



Double-click before the time (represented in yellow) is up.

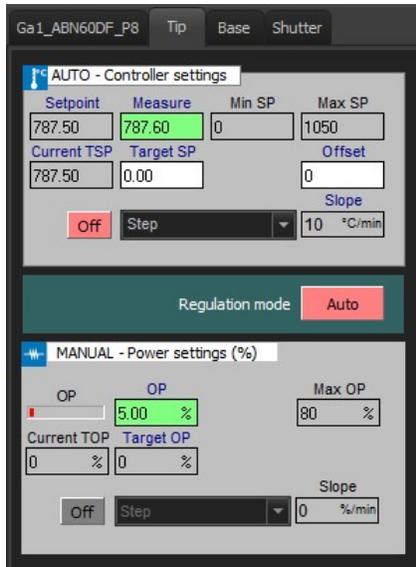
Opening the Detail view window



Click on the following button , from the desired sub equipment's item, to open the *Detail view window*.

A *detail view* window is associated with each piece of sub equipment. The *Detail view* allows users to control the values of the sub equipment.

Example: Detail view of Insert sub equipment associated with Ga2_ABI85_P11 equipment

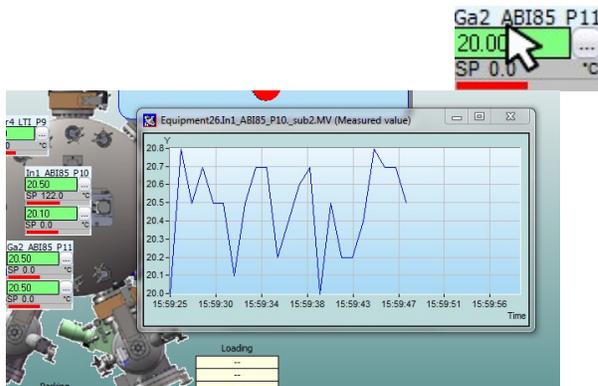
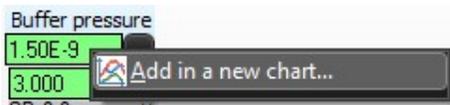


The available settings depend on the type of sub equipment.

→ For more details about the Detail view, please refer to the section [Example of Detail view window](#) from the chapter [Chambers > Equipment view](#).

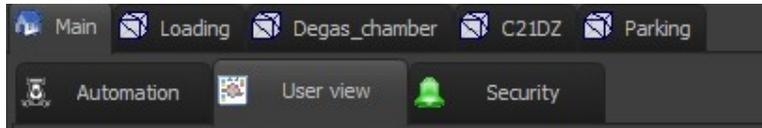
Add in a new chart

Right-click on a parameter's value and select *Follow in a chart* to open the corresponding chart in a pop-up window.



To save this chart, right-click on any part of the chart window, select *File > Save as* and then select the desired file format.

7.3. SYSTEM SECURITY

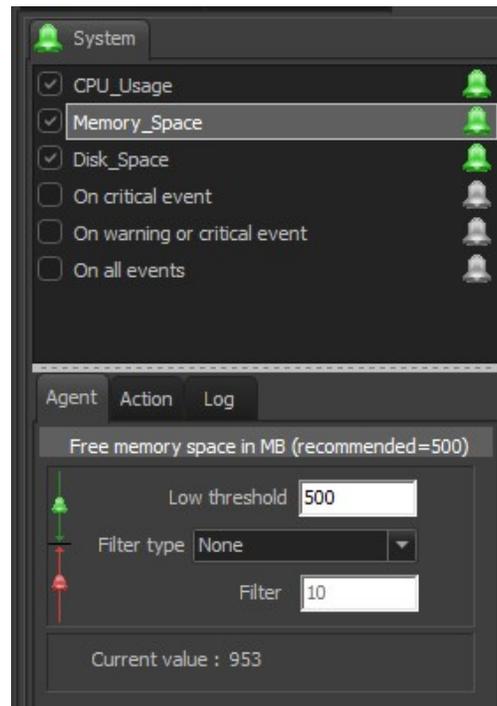


The *Security* tab allows you to enable or disable the system security agents and configure their alarms.

1. Select a security agent



2. Set the alarm



Select the desired security agent in the upper panel. You can set the alarm for each selected security agent using the lower panel.

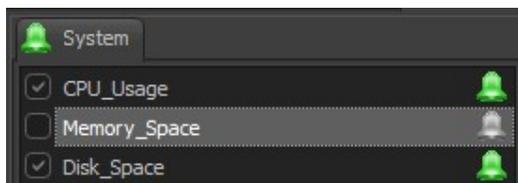
For example, after selecting the *Memory Space* security agent, you can set a 500 MB low threshold value from the *Agent* tab.

An alarm will be activated as soon as the amount of memory available drops under 500 MB.

The *Action* tab allows you to configure alarm notifications (sound, email, SMS, etc.).

The *Log* tab is history of the last 50 events related to selected

You can enable/disable security agents by checking/unchecking the box:



→ For more details about the configuration of security agents and alarms, please refer to the chapter [Chambers](#) > section [Chamber security](#) in this manual.

8. CHAMBER TABS

In Crystal, a chamber is a sub system.

For example, a chamber can be a growth chamber, a load/unload, a preparation chamber, a flip/flop etc... each sub system that is connected to a cluster port is called a chamber.



On the right side of the *Main* tab, the chamber tabs allow you to display chamber features. The number of visible chambers depends on your configuration.

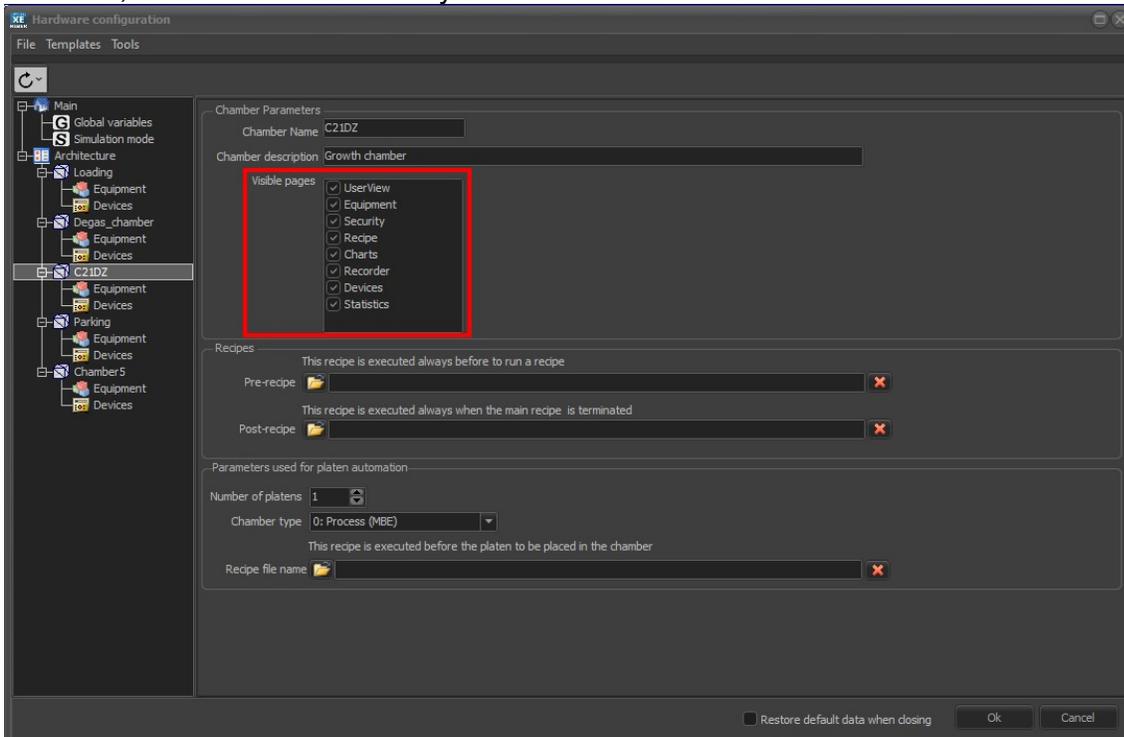
Click on the desired chamber to display the chamber tabs.

The chambers tab are the following:



Depending on the configuration of the chamber concerned, some tabs may not be visible. For example, if no device is defined for this chamber, the device tab will not be displayed.

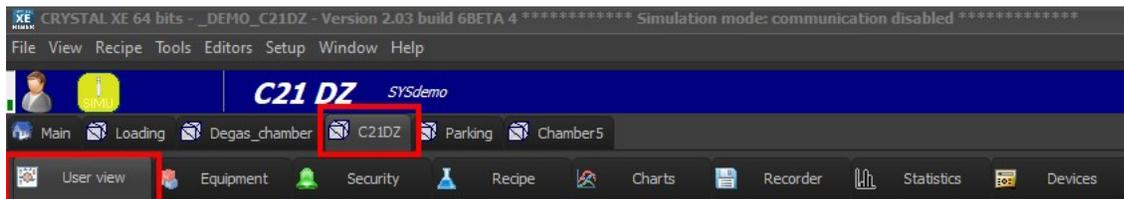
You can also show or hide each tab. To do this, go in the hardware configuration, click on the chamber, and check the tab that you want to see:



- **Userview:** by default, this tab is not displayed. You can create a special view for this chamber only.

- **Equipment:** display the list of all equipment and a detail view.
- **Security:** the security tab allows you to configure all securities in Crystal, equipment by equipment and sub-equipment by sub-equipment.
- **Recipe:** this tab is used to create and edit recipes, browse the recipe files and execute a recipe.
- **Charts:** displays the mains data in several charts in function of time.
- **Recorder:** Each of the 3 recorders save data in a separate file. The file format is CSV (text file). The default acquisition period is 2 seconds. It can be modified but cannot be less than one second.
- **Devices:** This view allows to see if there is no communication issues with devices. It is also possible to open an analyzer for each device to help to solve a communication problem.
- **Statistics:** for maintenance purposes, the statistics tab is used to find out how long a cell has been used, how many times a shutter has been used, how many turns a manipulator has performed, etc.

8.1. USER VIEW



The user view of a chamber can be used to make a second user view, after the main user view.

→ For more information, see the chapter [MAIN TAB / Userview](#)

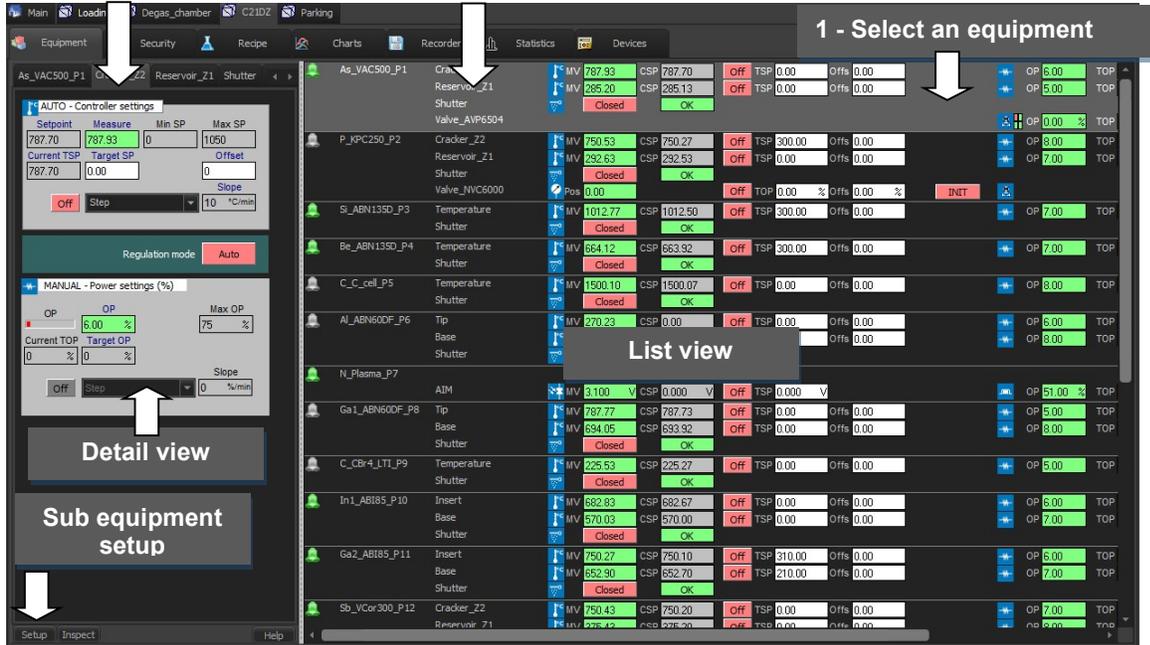
→ You can also see the chapter [Forms and objects](#) in this manual.

8.2. EQUIPMENT VIEW



2 - Select sub equipment

Or click directly on a sub equipment here



Example: C21DZ chamber's Equipment view

The *Equipment* interface displays all equipment in a list view (right) depending on your configuration.

You can directly modify some of the equipment's parameters from the list view (closing/opening shutters, modifying target set point values, etc.).

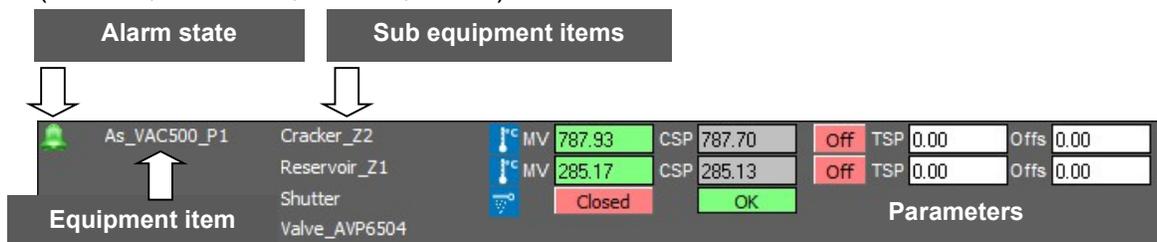
Click on the desired equipment from the list view to display its associated *Detail view* window (left).

8.2.1. Equipment list view

Equipment item

Each line displays one piece of equipment and all its associated sub equipment. For each piece of sub equipment, you can find its parameters listed horizontally.

Example: List view of *As_VAC500_P1* equipment and its four pieces of sub equipment (*Cracker*, *Reservoir*, *Shutter*, *Valve*).



Select a piece of equipment to display its corresponding *Detail views* on the left side of the interface.

Alarm icon

The alarm icon (bell), located on the left side of each piece of equipment, indicates the state of the equipment alarm:



Grey bell: all sub equipment alarms are disabled



Green bell: at least one of the sub equipment alarms is enabled



Red bell: at least one of the sub equipment alarms is activated

Alarms occur (are activated) according to the configuration of their associated security agent.

→ For more details about the configuration of security agents and alarms, please refer to the section **Chamber security** in this chapter.

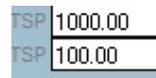
Interact with the equipment

From the list view, you can directly modify some of the sub equipment parameters.

Example:



opening /closing shutters



modifying the values



stopping a running ramp

The editable fields and available features depend on the type of sub equipment.

Customizing the list view

- **Changing the equipment order**

Drag the equipment item up or down to modify the order.

- **Hiding/displaying equipment**

Click on the equipment item, hold down the left mouse button and drag it up or down to display the trash icon:



Drag and drop the desired equipment item into the trash icon to hide the equipment.

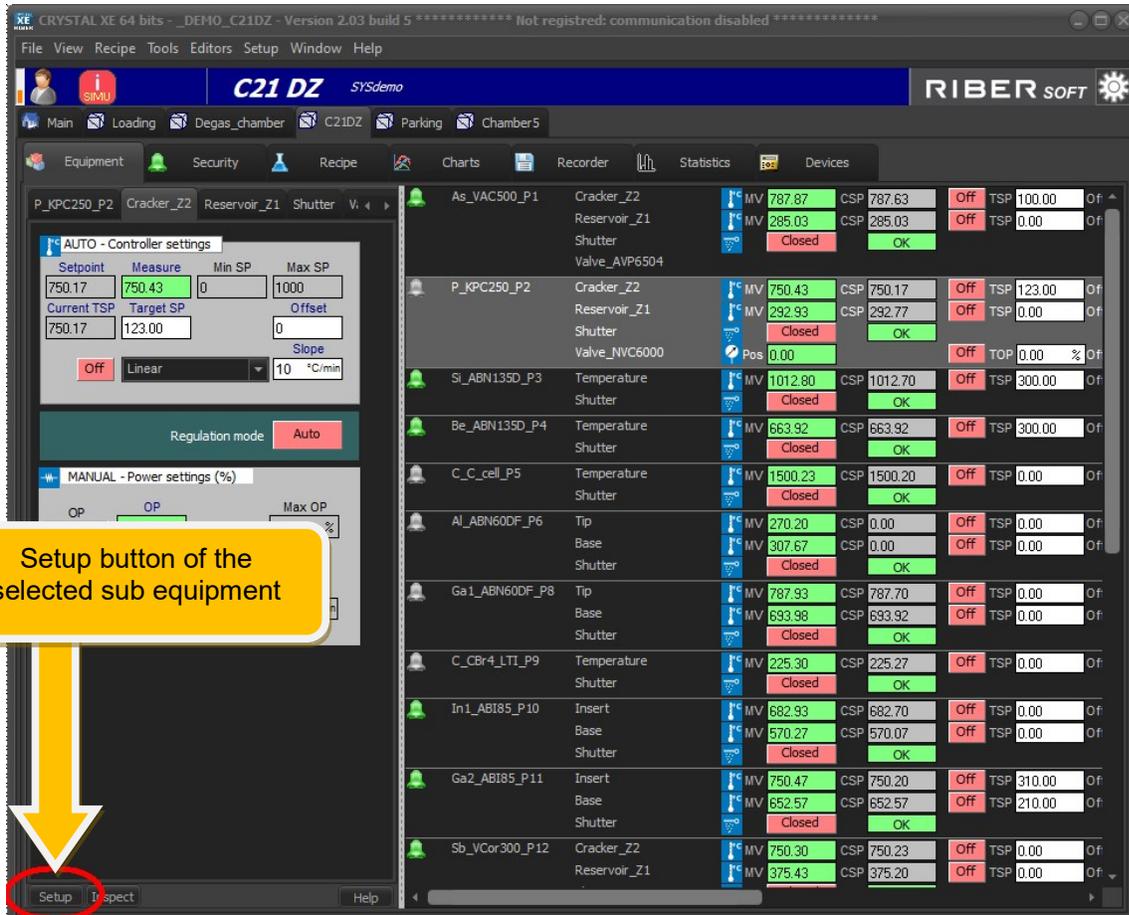
You can display all hidden equipment by right-clicking on any part of the interface and selecting *View all hidden equipment*.

To display only specific equipment, right-click on any part of the interface, move the cursor over *Show specific* equipment and select the equipment to be displayed.

The equipment reappears at its previous order.

8.2.2. Equipment setup

Some sub equipment parameters, such as maximum and minimum set point values, are not directly editable.



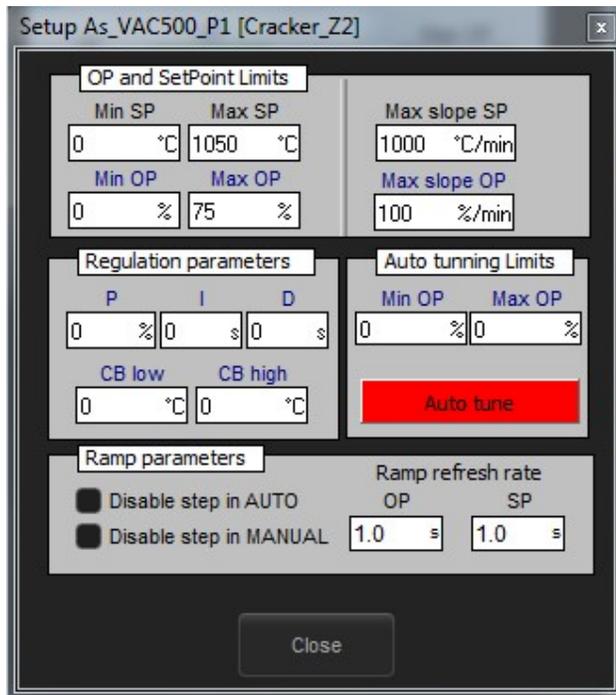
Setup button of the selected sub equipment

To configure the sub equipment setup values:

- Select the equipment in the list view
- Select the sub equipment in the Detail view:

Click on the Setup button **Setup**, located in the bottom left-hand corner of the window, to open the Setup pop-up window:

Modify the setup parameters:



The setup parameters depend on the sub equipment type.



The appropriate user right (*Setup equipment*) is needed to access that window.

8.2.3. Example of Detail view window (temperature)

The *Detail view* window displays the parameters of sub equipment and allows users to follow and edit their values (when editable).



Users need the appropriate user right (Control equipment) to edit the sub equipment values.

Each type of sub equipment displays a specific *Detail view* window.

Click on the desired equipment in the list view to display the *Detail view*.

Example:

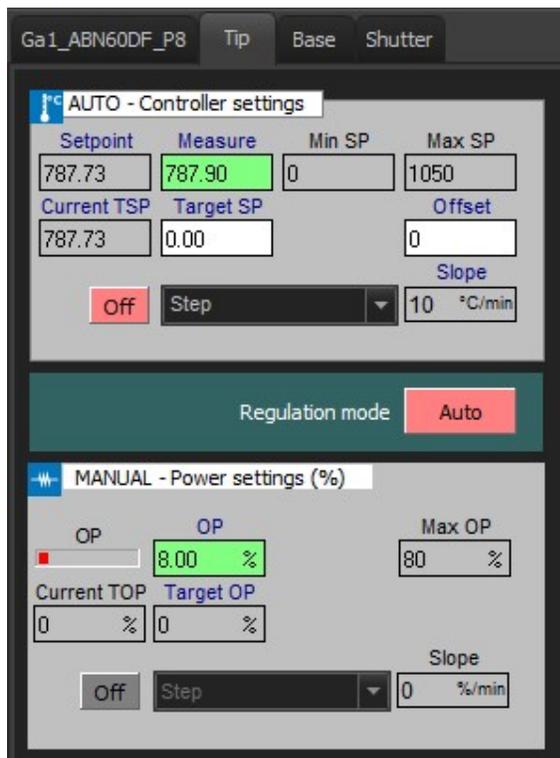
Detail view of *Ga1_ABN60DF_P8* equipment with its three associated pieces of sub equipment (broken into three tabs):

Tip

Base

Shutter

Click on the sub equipment tab to display the parameters.

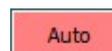


Automatic mode and manual mode

As for some types of equipment, you can switch the regulation mode from Automatic to *Manual* by double-clicking on the *Regulation mode* button:



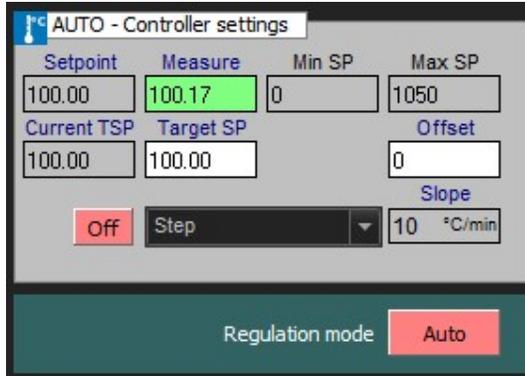
When you first click on the button, a progress bar appears. Click again before the time (yellow) is up.





If a linear ramp is running (**On**), you cannot switch the regulation mode.
 Stop the linear ramp (**Off**) before switching the regulation mode.

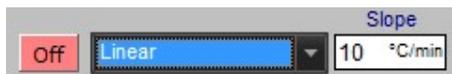
The Automatic mode allows you to specify a target set point value (°C) to be reached. Click on the Target SP field to type the desired value.



The Manual mode allows you to specify a constant power set point (%) to reach the required temperature. Click on the Target OP field to type desired value.



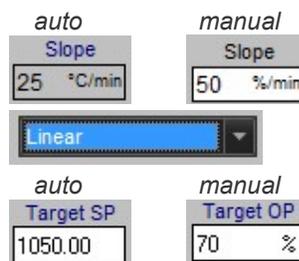
Step and linear ramp



Step: select Step from the drop-down list to directly set a target set point value (in Automatic mode) or a target output power value (in Manual mode).

Linear ramp: select linear from the drop-down menu to modify the temperature by programming a rise using a linear ramp until reaching the target set point value (in Automatic mode) or the target output power value (in Manual mode):

- set the ramp slope
- select 'Linear' from the drop-down list
- set a target value



The ramp activity button is turned on (green) when a linear ramp is running



To stop a running linear ramp, turn off the ramp activity button  by double-clicking on it.

When you first click on the button, a progress bar appears .

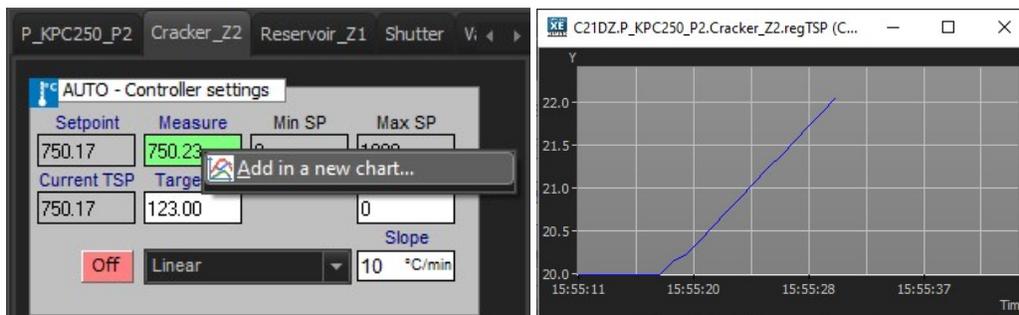
Click again before the time (yellow) is up).

Other ramp profiles can be defined from the Options window. (Setup > Options).

→ For more details about ramp profiles, please refer to the chapter Setup > sections Options > **General options** in this manual.

Add in a new chart

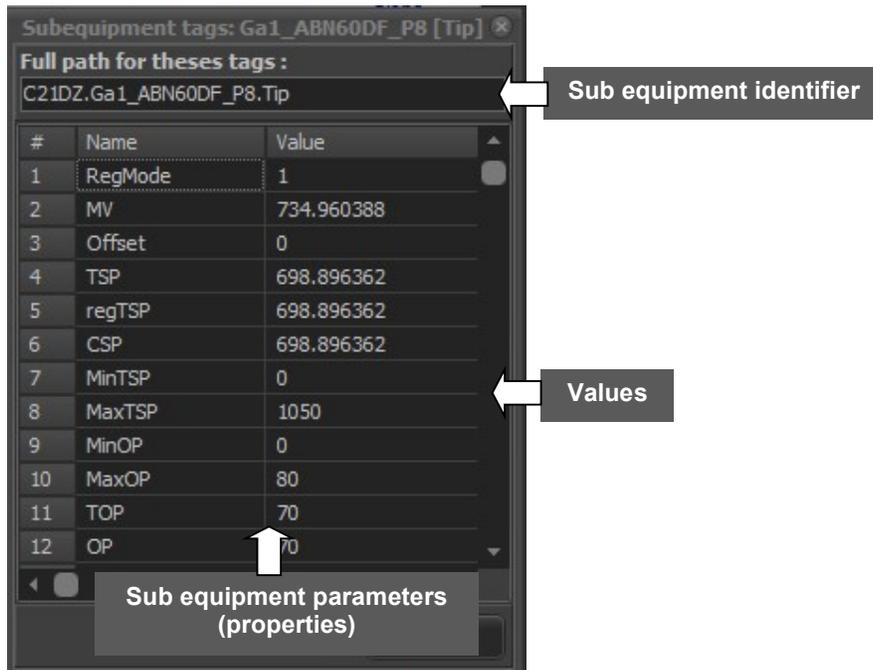
Right-click on a parameter's value and select *Follow in a chart* to open the corresponding chart in a pop-up window.



To save this chart, right-click on any part of the chart window, select *File > Save as* and then select the desired file format.

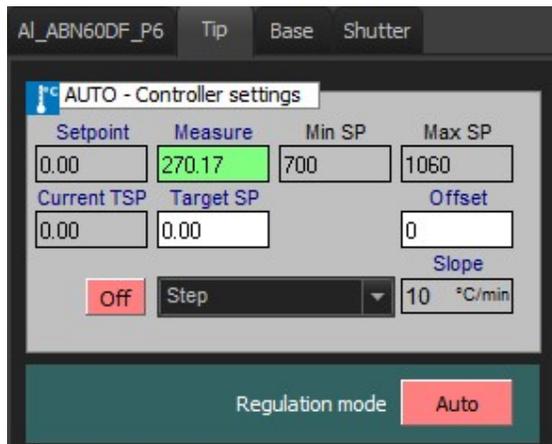
8.2.4. Sub equipment inspector

The sub equipment inspector is a pop-up window that lists all the sub equipment parameters and their current values in a table. The values can be followed in real time:



Example: inspector of *Tip* sub equipment associated with *AI_ABN60F_P6* equipment

To open the inspector pop-up window:
select the equipment in the list view
select the sub equipment in the Detail view



click on the Inspect button, located in the bottom left-hand corner of the window, to open the inspector pop-up window:



8.3. CHAMBER SECURITY



The alarms management interface allows you to:
 configure security agents for each type of sub equipment,
 set an alarm for each security agent,
 choose the action to be performed when the alarm is activated.

If one of the chamber's alarms is activated, a red bell icon will appear over the chamber's tab and the *Security* tab alarm icon will turn red as follows:

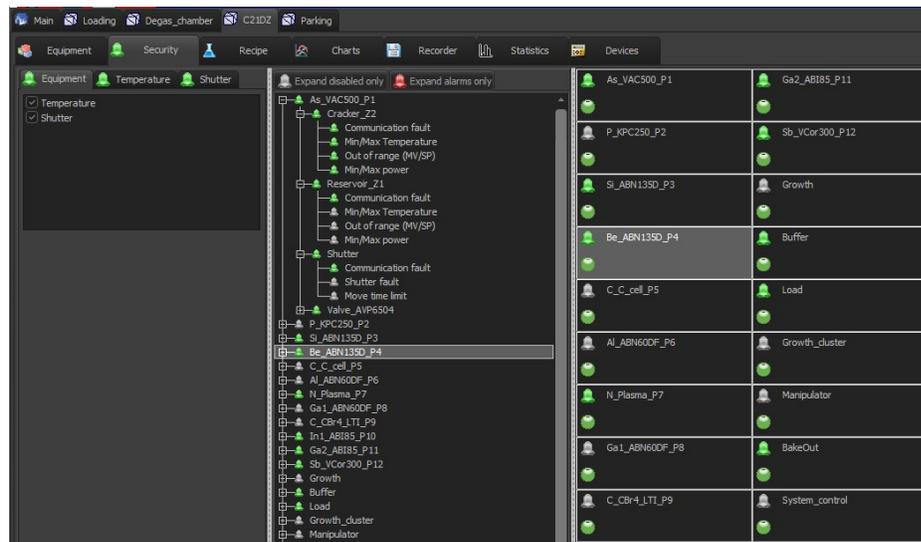


Bell icons indicate the state of the alarm as follows:

-  Grey bell: the alarm is disabled
-  Green bell: the alarm is enabled
-  Red bell: the alarm is activated (has occurred)

The security interface is divided into three parts as follows:

- the equipment view (right),
- the tree view (center),
- the security agents management panel (left).



Security agents panel

Tree view

Securities equipment view

8.3.1. Securities equipment view

The securities equipment view displays all the functioning equipment, depending on your configuration.

Click on the desired equipment to highlight the equipment line in the tree view and manage its security agents on the left side of the interface.

The bell icon indicates the state of the equipment's alarm (enabled, activated or disabled). The equipment alarm is activated if at least one of its security agents' alarms has occurred (security alarms of its sub equipment).



The following circle icon  indicates that the equipment alarm has occurred. The digit indicates the number of times the alarm of the equipment occurred.

For example, the following icon  indicates that the equipment's alarm occurred four times, whether or not an alarm is still occurring.

Click on this icon to acknowledge the activation of the alarms. Once all alarms have been acknowledged, the icon will turn green .

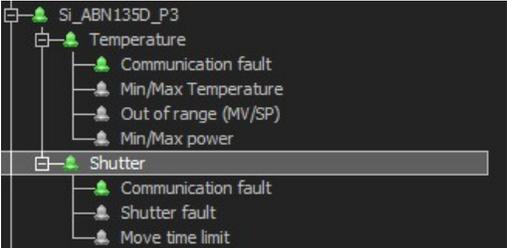
 You cannot acknowledge an alarm if this alarm is still activated.

8.3.2. Tree view

All the equipment and sub equipment alarms are displayed in a tree view as follows:

Equipment > sub equipment > security agents

Example: tree structure of the Si_ABN135D_P3 equipment



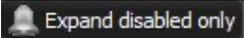
Si_ABN135D_P3 equipment associated with two pieces of sub equipment:

- Temperature
- Shutter

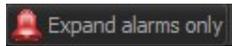
Temperature sub equipment associated with four security agents:

- Communication fault
- Minimum and maximum temperature
- Out of range
- Minimum and maximum power

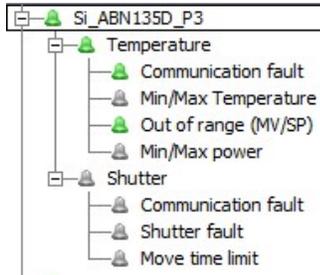
In the tree structure, double-click (or click on the + / - icons) to expand/collapse a list. You can choose to expand only the security agents whose alarms are disabled by clicking on the Expand disabled only tab:



You can also choose to expand the equipment level alarms only to have a quick view on the alarms' state, by clicking on the *Expand* alarms only tab:



Enabling/Disabling alarms



In the tree structure, right-click on the desired security agent and select *Enable* to enable the alarm (green icon) .

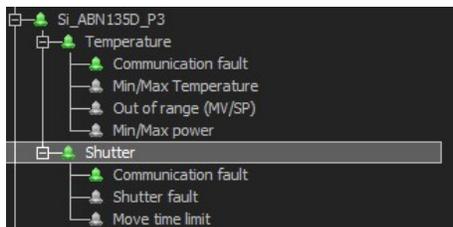
Right-click and select *Disable* to disable the alarm (grey icon) .

You can disable or enable alarms at the security agent level or at the sub equipment level in the tree structure.



You cannot disable alarm at the equipment level.

8.3.3. Alarm levels



Example: *tree structure of the Si_ABN135D_P3 equipment*

The alarms can be activated, enabled or disabled at three different levels:

Equipment level: you cannot enable/disable alarm at the equipment level. The equipment alarm will occur (be activated) if one of its sub equipment alarm is activated.

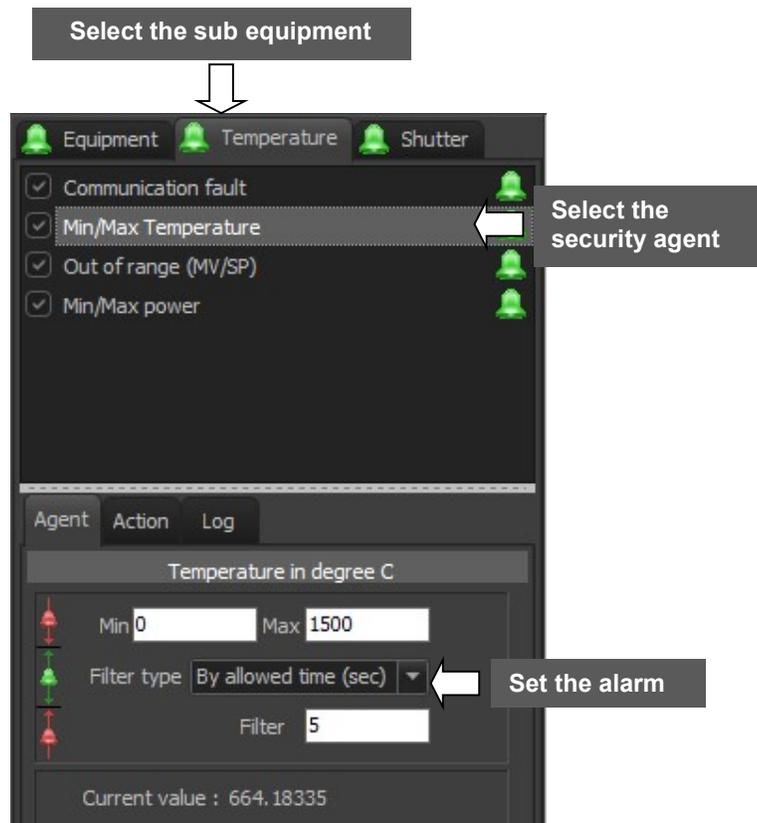
Sub equipment level: you can enable/disable alarms at the sub equipment level. When the sub equipment alarm is disable, all its security agents' alarms are disabled as well. The sub equipment alarm will be activated is at least one of its security agent alarm is activated.

Security agent level: you can enable/disable alarm alarms at the security agent level. If the alarm at the sub equipment level has been disabled, you must enable it prior to enable an alarm at the security agent level. The security agent alarm occurs according to its configuration (threshold values).

8.3.4. Configuring the security agents

Select a piece of equipment whether on the equipment view or on the tree view to display its security agents in the security management panel (on the left side of the interface).

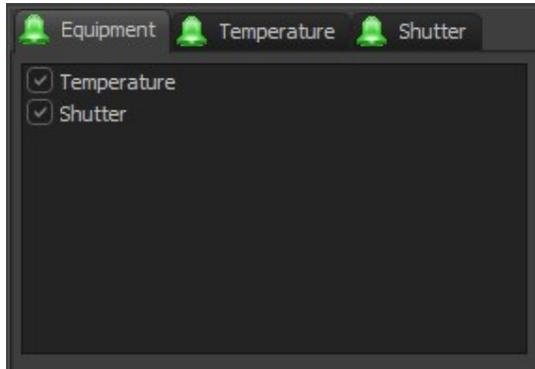
You can configure security agents for each piece of sub equipment that are divided into different tabs.



Enabling/disabling alarms

Sub equipment level

The *Equipment* tab displays all the pieces of sub equipment associated with the selected equipment:



Example: Si_ABN135D_P3 equipment

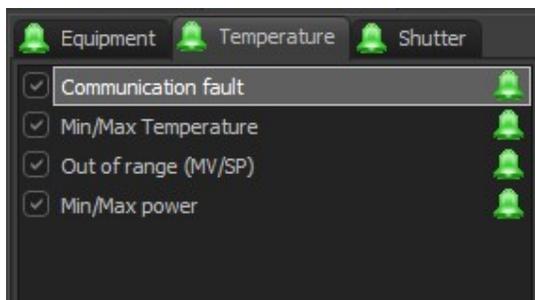
To enable/disable a sub equipment alarm, check/uncheck the corresponding box. When you disable a sub equipment alarm, all its security agents' alarms are disabled as well.

Security agent level

Select a piece of sub equipment from the available tabs to display the sub equipment's security agents.

In the following example, the Temperature sub equipment is associated with four security agents:

- Communication fault
- Minimum and maximum temperature
- Out of range
- Minimum and maximum power



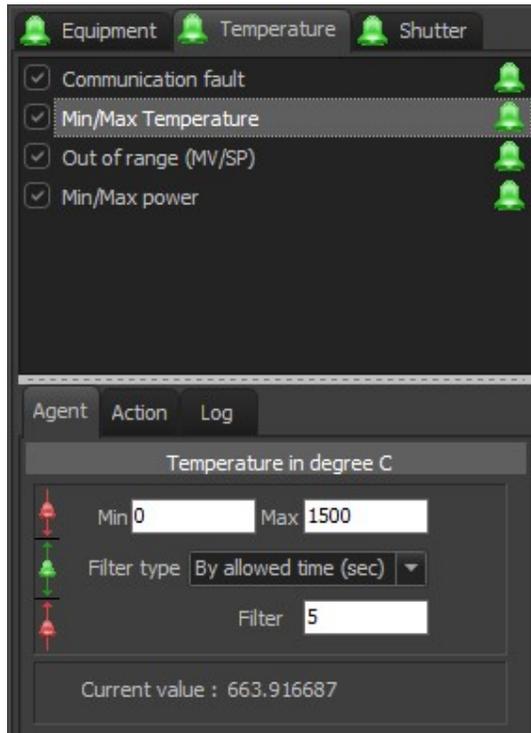
To enable/disable a security agent alarm, check/uncheck the security agent box.



The sub equipment level alarm must be enabled to allow you to enable its different security agents' alarms.

Settings alarms

From the sub equipment tab, select the desired security agent to display its setting in the lower part of the panel:

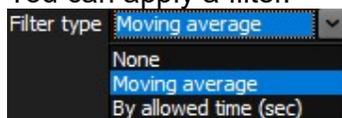


The available settings depend on the type of the selected security agent.

Agent

The *Agent* tab allows you to set threshold values and configure the activation of the alarm.

You can apply a filter.



Moving average filter

The moving average is more relevant for values other than binary ones, so for example for a shutter it's of limited interest, but for a temperature or a pressure value it's more appropriate.

The calculation is as follows:

$$\text{NewValue} = ((\text{PreviousValue} * \text{FilterValue}) + \text{CurrentMeasuredValue}) / (1 + \text{FilterValue})$$

The focus is on testing the result of the calculation, rather than the measured value.

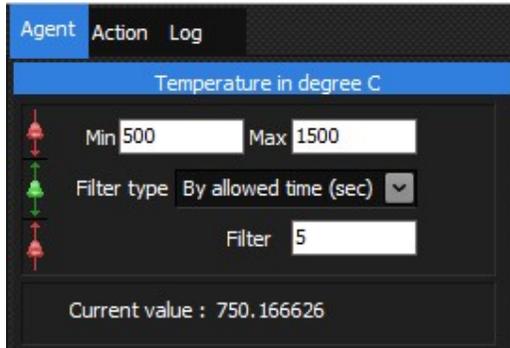
This smoothes the curve to avoid triggering the alarm on a measurement peak.

The larger the filter, the more effective the smoothing, but the longer the alarm triggering time.

By allowed time

In this mode, the filter value represents a time in seconds. The measured value must exceed the threshold throughout the set time. This is called hysteresis.

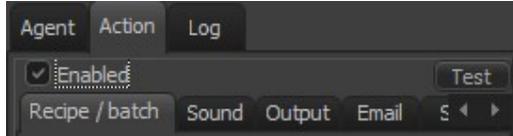
For example, after selecting *Min/Max Temperature* security agent, you can specify minimum and maximum temperature values:



In this example, if the temperature value falls below 500°C or exceeds 1500°C continuously for 5 seconds (filter), the alarm will be activated. 🚨.

Action

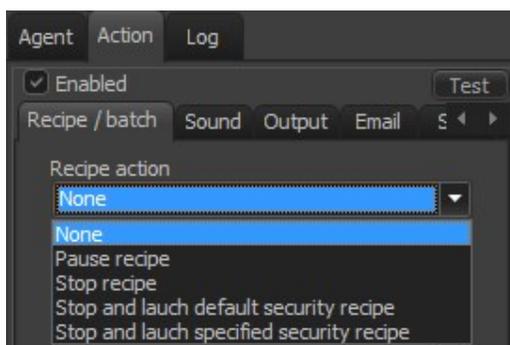
On the *Action* tab, check *Enabled* to display the available actions divided into different tabs:



You can define actions to be performed whenever the alarm is activated using these different tabs.

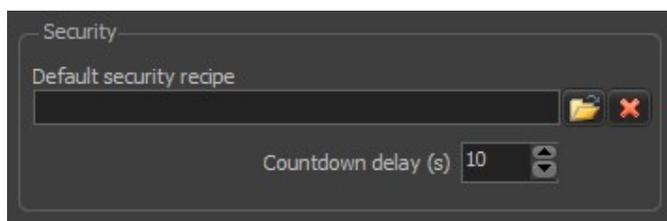
Recipe and batch

From the drop-down list, select the action to be applied to the **recipe** when the alarm occurs:

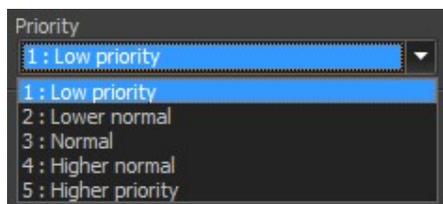


- **Security recipe**

The **default recipe** and countdown can be specified for the chamber by clicking on the *Setup* menu > *Options* > *Chambers*:



Define the security recipe **priority** from the drop-down list:



 For the security recipe to be executed, the priority level of the security recipe must be higher than the executing recipe. If the priority level of the security recipe is equivalent of lower than the recipe running within the chamber, the security recipe will not execute.

▪ Batches

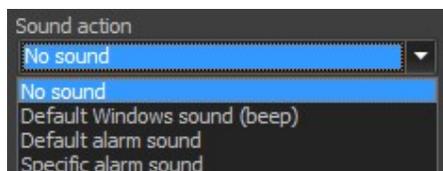
From the drop-down list, select the action to be applied to the batches when the alarm occurs:



Batches can either be stopped or paused when the alarm is activated.

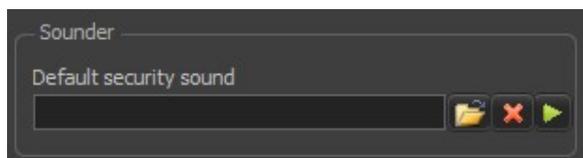
Sound

When the alarm is activated, you can be notified by a sound. Select the desired option from the drop-down list:

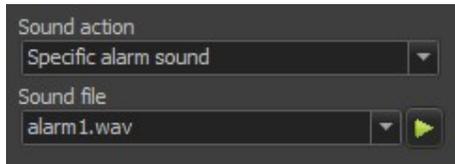


The default alarm sound can be defined from the *Options* menu (**WAV format only**):

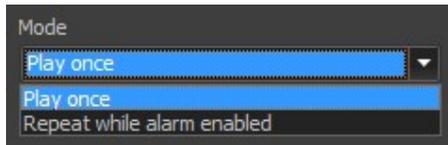
Go to *Setup* > *Options* > *General*



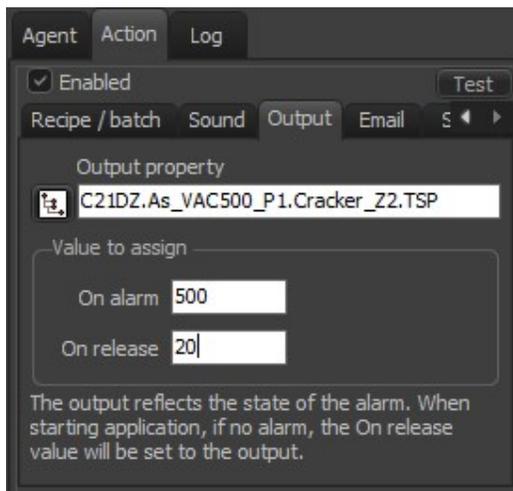
Select Specific alarm sound to select a sound file (**WAV format only**)



Select the repeat option from the drop-down list:

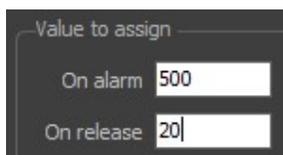


Output



On the *Output* tab, you can set values to be assigned automatically to the defined property if the alarm occurs.

Select a property from the Data explorer by clicking on the following icon:  Enter the values to be assigned when the alarm is activated and/or when the alarm terminates:



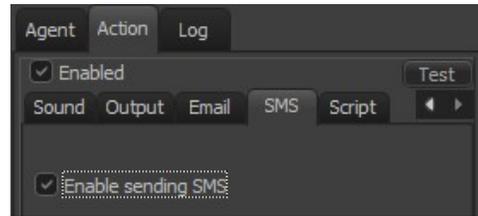
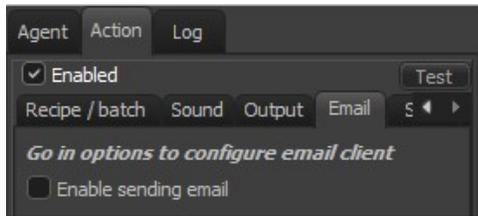
Email and SMS

If the alarm occurs, you can be notified an Email and/or SMS alert.



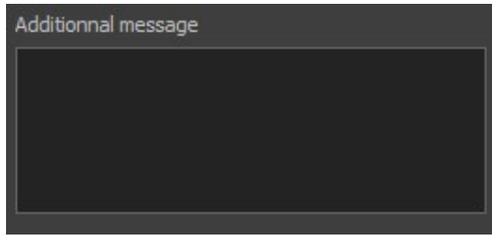
Email and SMS notifications must be enabled to configure these options. To enable notifications, go to *Set up > Options* and then *Emails* or *SMS* tabs.

Enable the Email and SMS notifications by checking the box on the corresponding tab:



 After enabling notifications from the *Options*, if the checkbox is not active, please select another security agent and then select again the security agent to activate the modification.

You can type a customized message into the text box:

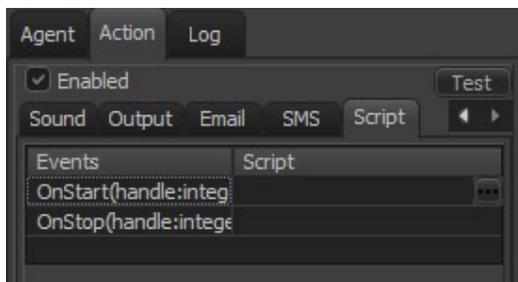


Appropriate user right is needed to perform the above actions (Options).

→ For more details about the Options please refer to the chapter Setup > section **Options** in this manual.

Script

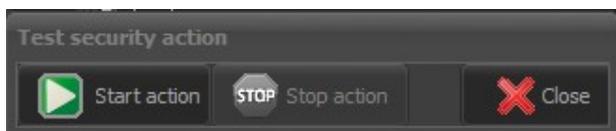
On the *Script* tab, you can specify scripts to be executed when the alarm is activated (*OnStart* event) and/or when the alarm terminates (*OnStop* event).



Select the event in the table and click on the following icon  to edit the script using Pascal editor.

Testing the actions

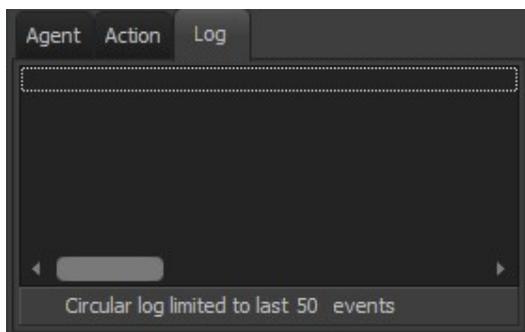
Click on the *Test* button , located in the top right-hand corner of the box, to test all actions that have been set simultaneously:



Testing actions allows you to verify the execution of scripts, the delivery of Email or SMS, the sound, etc. before the activation of the alarm.

Log

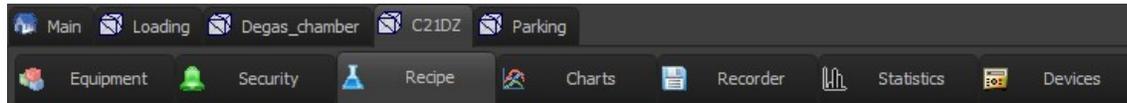
The *Log* tab displays all alarm related events.



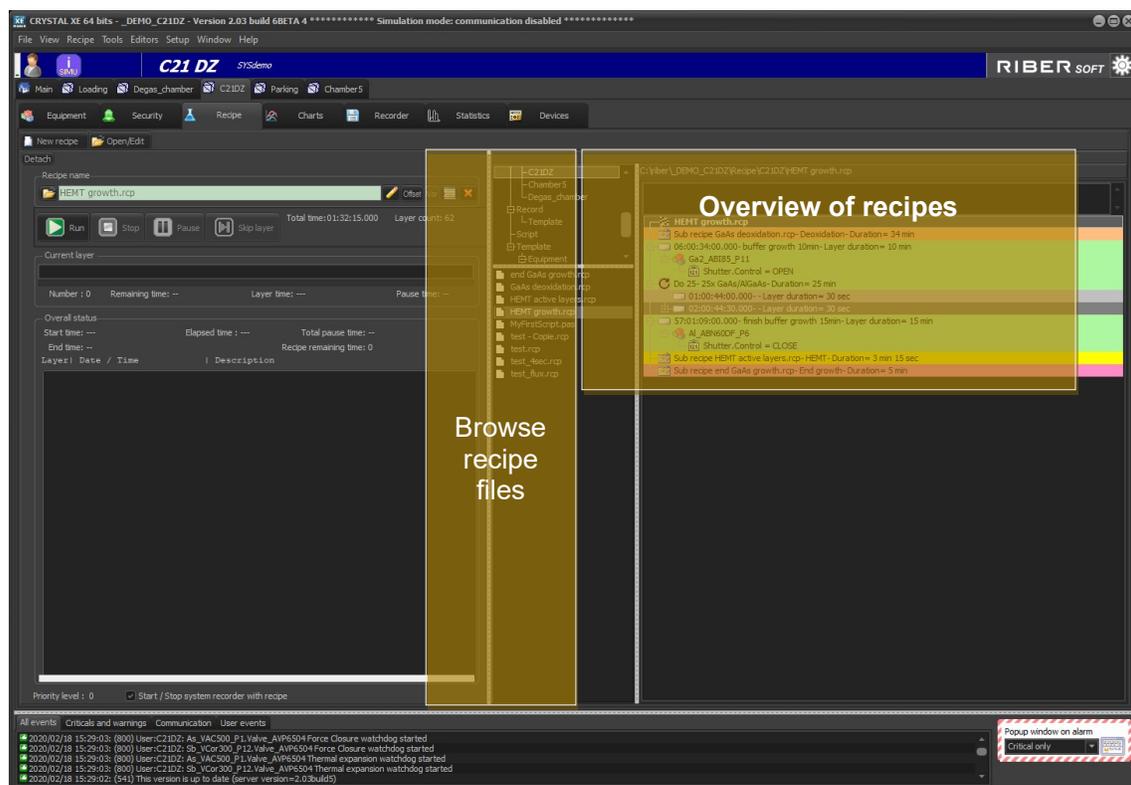
The events are saved automatically as *.txt* files to the *Log* folder, in your project directory.

You can open your project directory by clicking on the *File* menu (menu bar) and selecting *Browse project directory*.

8.4. RECIPES



The *Recipes* tab, allows users to create, edit or execute recipes. It also allows you to browse all the recipe files and display an overview.



In *Crystal XE*, recipes are made of a succession of layers. In these layers, every set point stays frozen for a time duration predefined for this specific layer.

All the actions defined for a layer are executed at the beginning of the layer's timer (temperature, opening the shutter, opening valve, etc.). The ramp duration can be defined over the layer's duration or can be customized.

⇒ For more details, refer to the **RECIPE EDITORS** section later in this manual.

① Also, the annex **Recipes by example** introduces you to the creation of new recipes step by step.

8.4.1. Creating a new recipe

You can edit a new recipe using either:

- the **recipe editor** (graphical interface),
- or
- the **script editor** (Pascal language).

Recipe editor

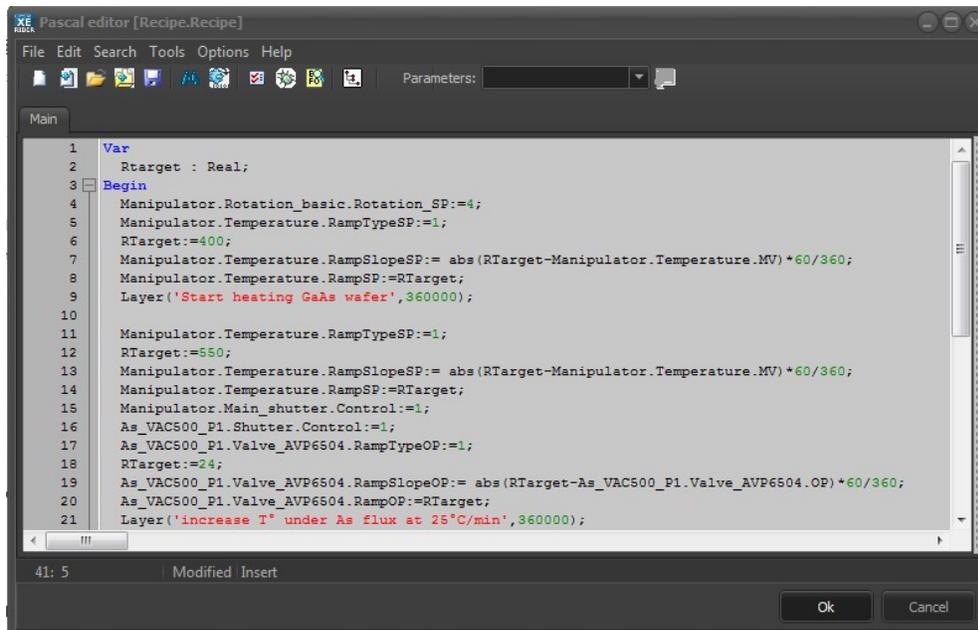
The graphical editor is the simplest and fastest method to write a recipe. On the other hand, these functionalities are more limited although the possibilities are endless.

The screenshot shows the 'Main recipe editor [GaAs deoxidation.rcp]' interface. It features a tree view on the left with steps like 'Start heating GaAs wafer', 'increase T° under As flux at 25°C/min', 'go to degas T° under As flux', 'wafer degas at 615°C real T°', and 'go to growth T°'. The center panel shows the 'Properties' for the selected step, including 'Layer duration' (00:06:00.000) and 'Comment' ('Start heating GaAs wafer'). The right panel displays a 'SUBSTRATE' layer stack visualization with colored blocks representing different steps. At the bottom, a 'Detach' table provides a detailed view of the recipe steps.

#	Layer	Description	Duration	Thickness	Material	1	2	3	4	5
01	00:00:00.000	Start heating GaAs wafer	6 m			Rotation_basic	Manipulator	Main_shutter	Shutter	As_VAC500_P1
02	00:06:00.000	increase T° under As flux at 25°C/min	6 m		As	Rotation_SP	Temperature	Control	Control	Valve_AVP6504
03	00:12:00.000	go to degas T° under As flux	13 m		As					
04	00:25:00.000	wafer degas at 615°C real T°	5 m		As					
05	00:30:00.000	go to growth T°	4 m		As					

Script editor

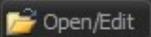
Scripts require programming knowledge.



The saved recipe files can be accessed from the *Recipe* folder, in your project directory.
 To open your project directory, clicking on the *File* menu and select *Browse project directory*.

⇒ **For details on how to write a recipe, see the "RECIPE EDITOR" section later in this manual.**

8.4.2. Editing an existing recipe

Click on *Open/Edit*  to edit a recipe file.

You can also open the recipe file by clicking on the following file icon



Once the recipe is loaded, click on the pencil icon to open the recipe editor.



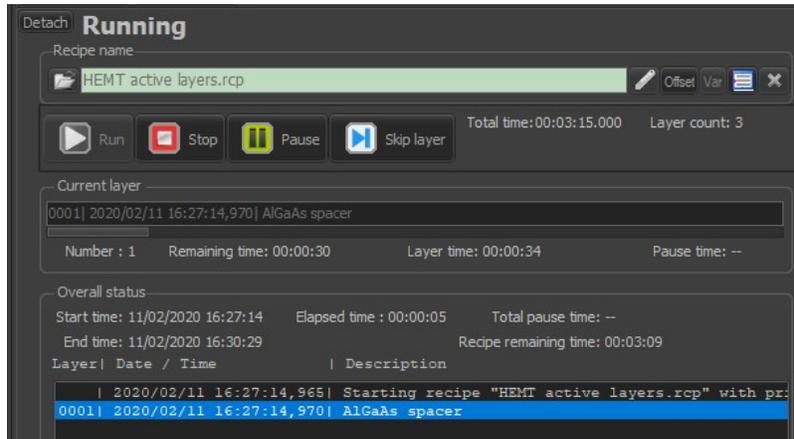
You can edit either *.pas* files (edited with the script editor) or *.rcp* files (edited with the main recipe editor).



Clicking on the following icon allows you to clear the file name input field but it does not delete the recipe file.

8.4.3. Executing a recipe

You can supervise and control the execution of the recipe using the Recipe inspector:



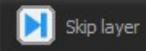
Click on the following icon  in the Recipe name field and select the desired recipe file.

Click on the Run button  and select “Run now” to start running now or “Run at” to choose the date and time for starting the recipe.

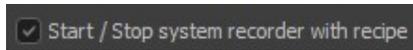
Overview of the dialog box for choosing when to start the recipe:



When the recipe is running, the Run button  is greyed.

Click on Skip layer  to abort the ongoing layer and continue the execution of the next one.

If the following box is checked, the *Data recorder* will start recording as soon as the recipe starts executing and will stop when the recipe terminates.



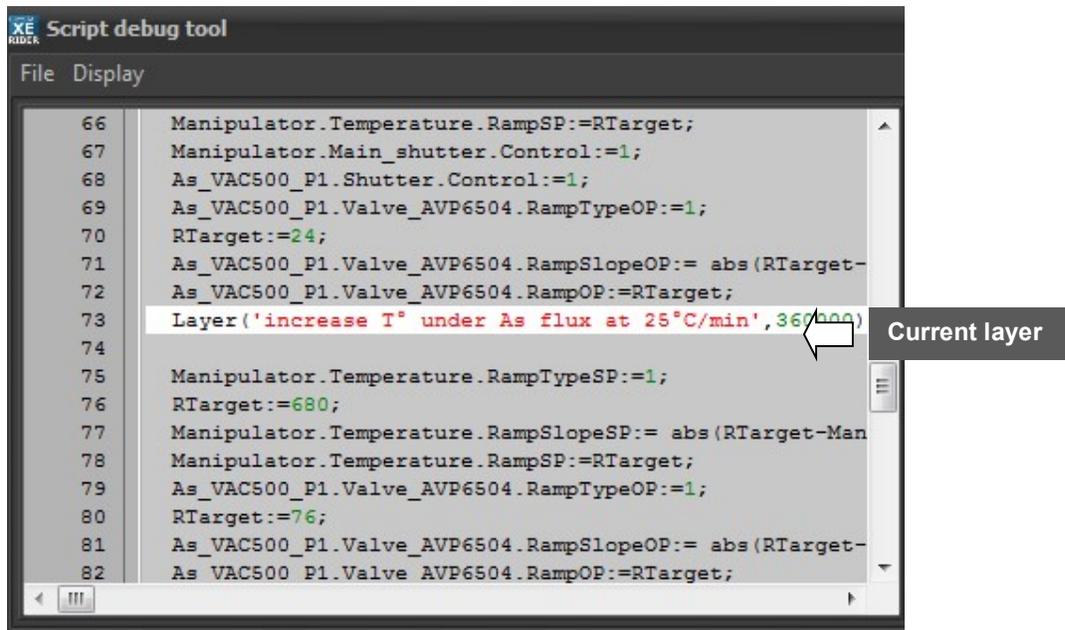
→ For more details about the system recorder, please refer to the chapter Chamber > section **Recorder** in this chapter.

You can open the recipe inspector in a separate window by clicking on the *Detach*  button, located on in the top right-hand corner of the inspector. Close the pop-up window to attach it back.

Script debug tool

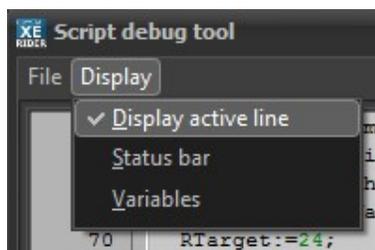
You can monitor the execution of the recipe by clicking on the following icon  and open the *Script debug tool*.

The script of the recipe is displayed in a pop-up window:

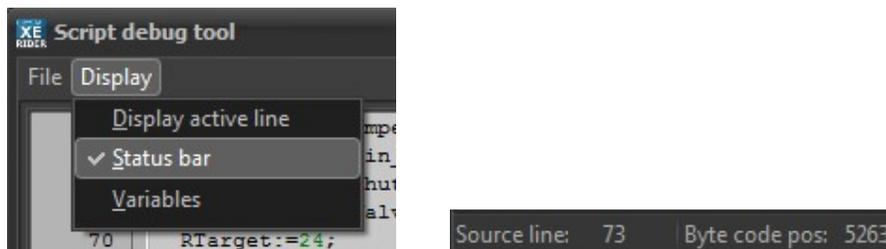


By default the layer that is currently executing is highlighted and you can follow the execution of the script.

- On the *Display* menu, if you uncheck *Display active line*, the executing layer will not be highlighted:

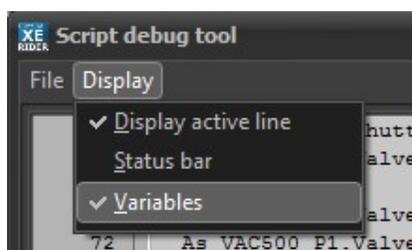


Select Status bar to display information about the executing source and compiled code at the bottom of the window:

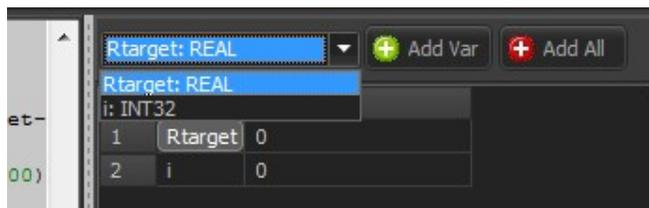


Displaying the number of the executing source line and byte code (compiled code) position

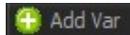
Select Variables to display the list of global variables declared in the script (that can be referred to throughout the script). Variables declared within a procedure (local variables) are not displayed.



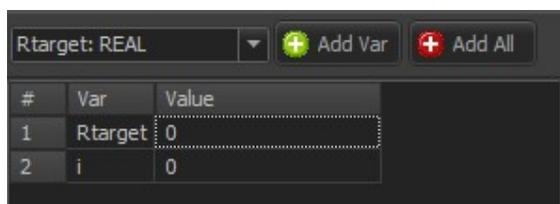
The variables are listed in a drop-down menu on the left side of the window.



Add the variables to the table below to display their values.

Select a variable from the drop-down list and click on *Add var*  to add a specific variable only.

Click on *Add All*  to add all variable at once to the table.

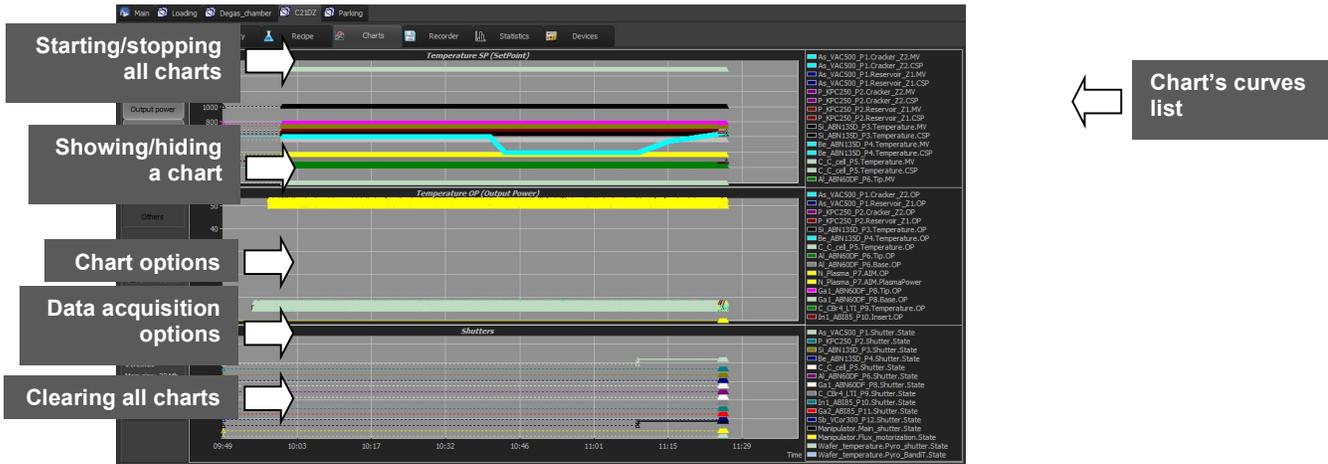


You can close or open the *Script debug* window at anytime before, during or after the recipe execution.

8.5. CHARTS



The *Charts* tab allows users to easily follow the chamber data using chart view. Charts are generated automatically depending on the equipment and sub equipment that have been set within the chamber.



8.5.1. General

Starting/Stopping charts

Starting all charts

- Use the buttons located in the top left hand corner of the interface to start or stop all charts:

Click on Run  to start the charts.

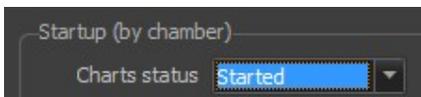
When all charts are running the button is turned grey .

Click on Stop  to stop running charts.

When all charts are stopped, the button is turned grey .

The chamber's charts can start automatically at the launch of the program.

On the menu bar, select *Setup > Options > Startup* and then select *Started* from the drop-down list:

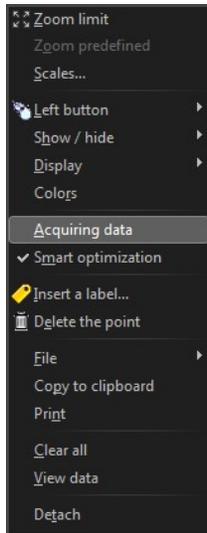


 Appropriate user right might be needed to perform the above action (*Options*).

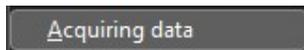
→ For further information, please refer to the chapter *Setup > section Options* in this manual.

Starting one chart

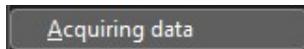
To start a specific chart only, right-click the chart background and select *Acquiring Data*:



When the chart is acquiring data, the following icon appears:



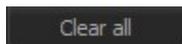
Unselect Acquiring data to stop the data acquisition:



Or click on the Stop button  Stop to stop all charts.

Clearing charts

To clear all charts' data, click on the *Clear all* button, located on the left side of your interface:



To clear only one specific chart, right-click the chart background and select *Clear all*. All data of the chart is cleared.

Displaying charts

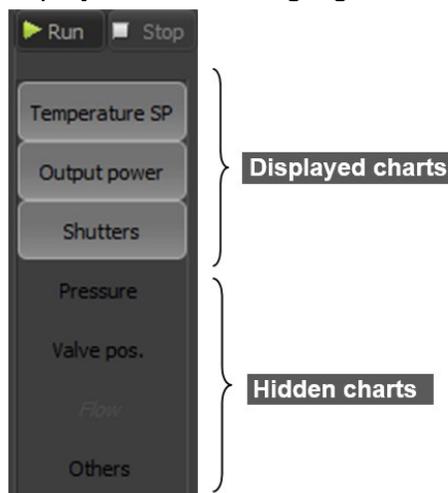
Seven different charts can be displayed for a chamber, regardless of your project configuration. The chart's data is represented by curves on the chart depending on the equipment and sub equipment set within the chamber.

Example: C21DZ (growth chamber):

Charts	Examples of sub equipment (growth chamber)	Examples of data (growth chamber)
Temperature set point (SP)	<ul style="list-style-type: none"> ▪ Temperature ▪ Cracker ▪ Reservoir 	Measured value (MV) Current set point (CSP)
Temperature output power	<ul style="list-style-type: none"> ▪ Temperature ▪ Cracker ▪ Reservoir 	Output power (OP)
Shutters	Shutter	State (1:on, 0:off)
Pressure	Pressure	Measured value (MV)
Valve position	Valve	Output power (OP) Target output power (tOP)
Flow	Mass flow controller (MFC)	Flow of gases
Others	<ul style="list-style-type: none"> ▪ AIM ▪ Pyrometer ▪ Cryopump 	Measured value (MV)

For a given chamber, some charts may have no data to display. With the demonstration project, for example, on the Parking chamber, only the Pressure chart has data to acquire and display.

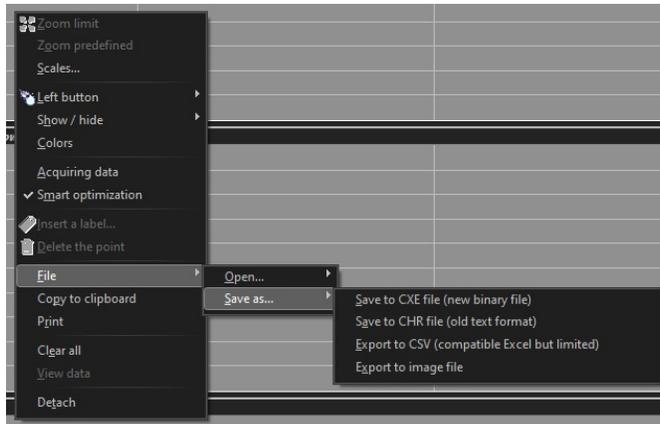
Click on a chart, from the pane located on the left side of the interface, to display/hide a chart. Displayed chart are highlighted :



Saving charts

Charts are not saved automatically onto your hard disk.

To save a chart, right-click the chart background and select *File > Save As* and then select the desired file format: CXE, CHR, CSV (text) or BMP (image *File*).

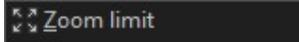


- CXE : is the latest chart format. This file format is the fastest.
- CHR: is kept only for reasons of compatibility with older versions.
- CSV: can be read by Excel but some information are lost.
- BMP: is an image file (bitmap).

Zooming in or out on chart

Zoom limit

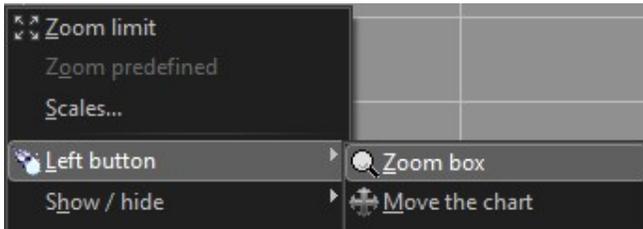
Right-click a chart background to display the pop-up menu. Select *Zoom limit* to automatically zoom in or out in order to display all data plotted on all curves from the beginning of the acquisition:



You can also double-click on a chart to perform a zoom limit.

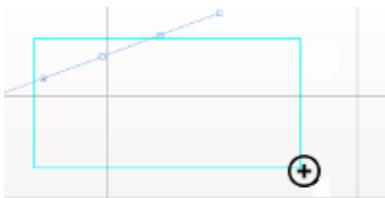
Left mouse button option

You can modify the option of the left mouse button. Right-click a chart background, select Left button and then select the desired option:



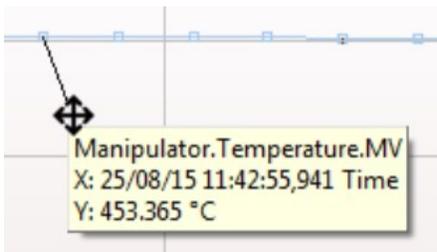
- **Zoom box (default):** focus on a specific part of the chart.

Click outside the points you want to focus on (zoom in), hold down the left mouse button and create a rectangle selection outside these points:



- **Move the chart:** display point information and maximize

Move the mouse over a curve point. A link (straight line) appears between the point and the cursor. Hold down the left mouse button and drag the cursor to pull the link. Then release the mouse to maximize and move over the chart. The more you pull the more it is maximized.



Mouse wheel

When the mouse cursor is moving over a graph, scroll the mouse wheel to zoom in or out.



Since all charts are using the same X scale (time), zooming in or out automatically affects all the charts.

8.5.2. Curves and data acquisition

The data acquired by the chamber's charts, depends on the equipment and sub equipment set within that chamber.

The acquired data is plotted by points on the chart and linked by lines (curves). You can also choose to hide points or lines (refer to the part Curve display in this section).

Displaying/hiding curves

The data acquired by each chart is represented by curves that are listed in the chart legend, located on the right side of the chart:



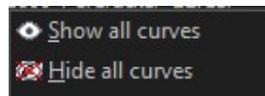
Example: Pressure chart's curves (growth chamber)

You can hide/display a curve by directly clicking on it in the legend:



The hidden curves appear in grey and are marked by a strikethrough line.

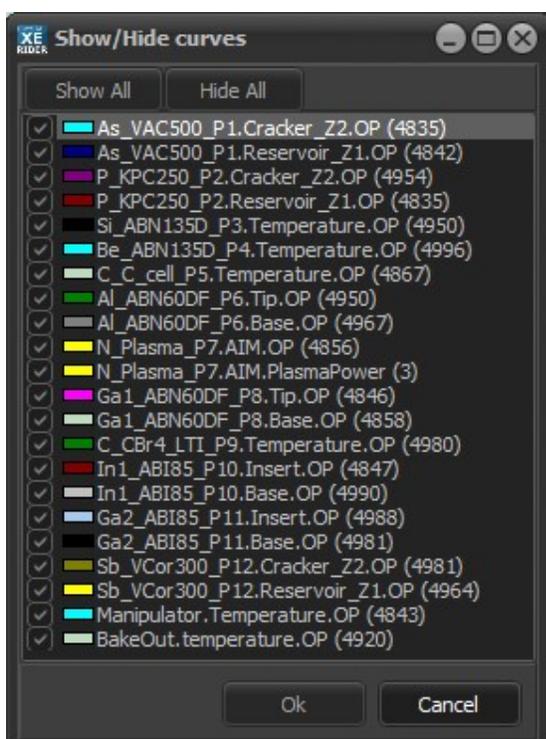
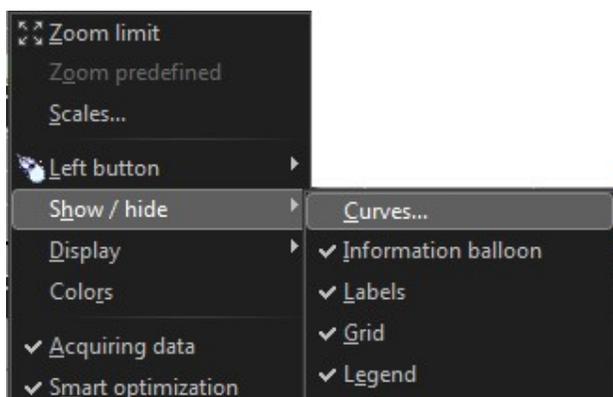
For each chart, you can hide/display all curves by right-clicking on the chart legend and selecting the corresponding item:



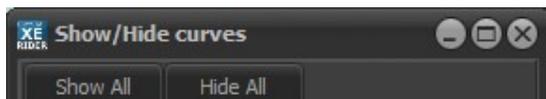
If your window is not maximized and/or if many charts are displayed, all curves may not appear in the chart legend (off-screen). To display the full legend list, unselect all the other charts and/or maximize the window.

You can also display the chart legend in a pop-up window and view all curves.

Right-click the chart background, select *Show/ hide* and then select *Curves* to open the pop-up window:



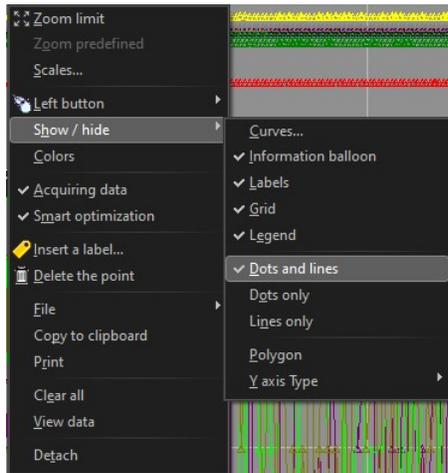
Using the pop-up window, display/hide curves by checking/unchecking a curve box. To hide or show all curves using the following buttons:



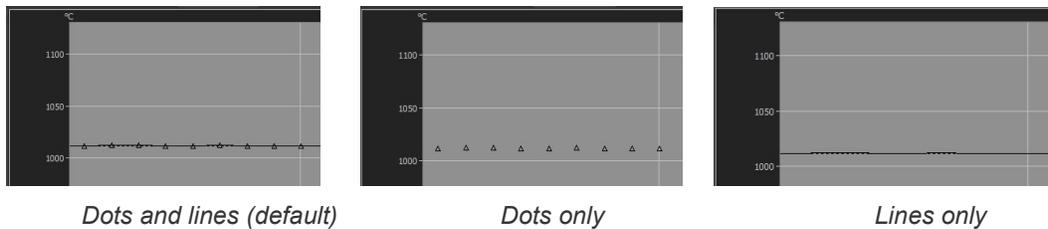
① Hiding curves do not stop the data acquisition. If you save the chart to CSV or CHR file, hidden curve's data will be saved as well.

Display options (Show / Hide)

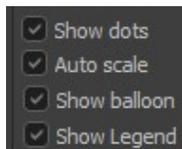
Right-click a chart background and move the mouse over *Show / Hide* to open the drop-down and select the desired option:



By default the acquired data is plotted by points linked by lines (line charts). You can choose to display both points and lines (default) or only dots or lines.



You can also show/hide dots by checking/unchecking the box in the panel located on the left side of your interface:

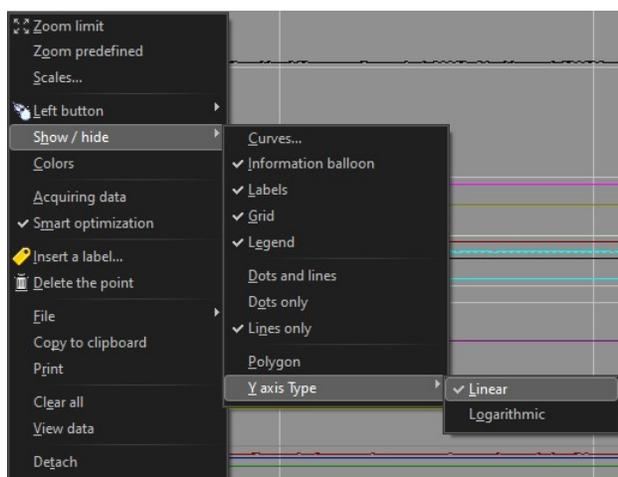


Select Polygon to display a filled polygon:



Polygon

Move the mouse over Y axis type and then select an item from the drop-down menu to define a linear (default) or logarithmic scale:

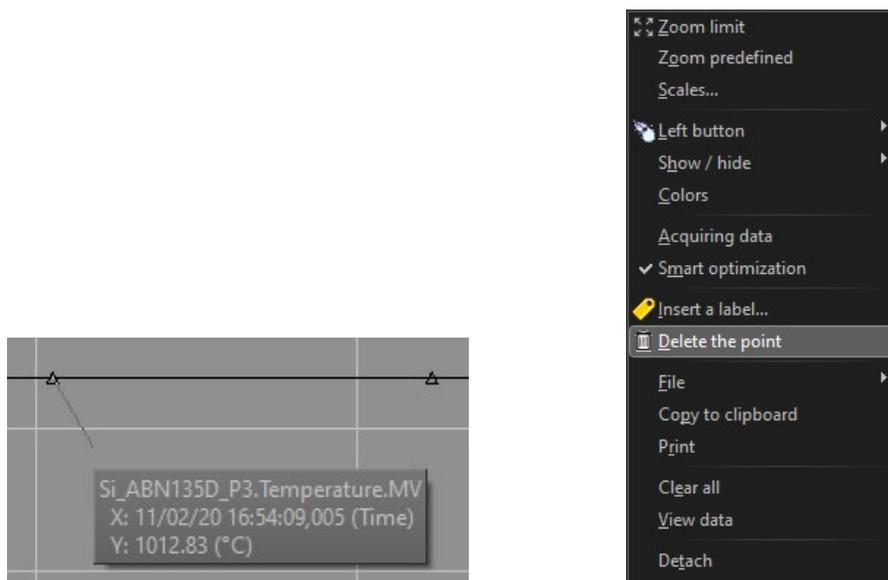


A logarithmic scale is a nonlinear scale usually used with large range of values.

Deleting points

You may need to delete a point (data) that has been plotted on a curve.

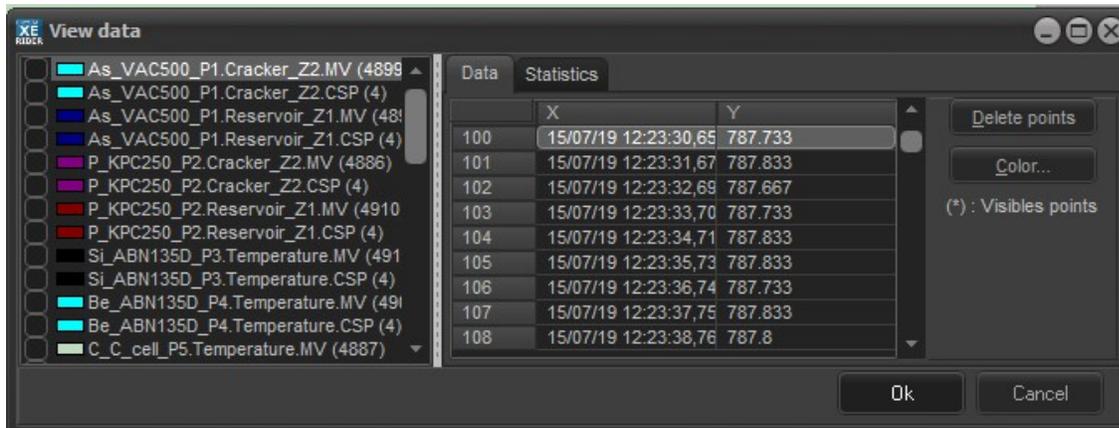
To delete a point, mouse the mouse over the point on the chart, right-click and select *Delete the point*:



You can also delete a point from the data view window (refer to the next part).

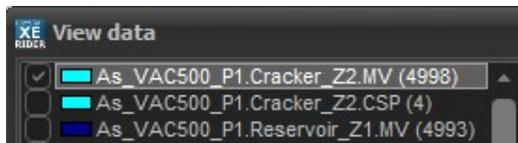
Viewing data and statistics

Right click a chart background and select *View Data* to open the pop-up window:



Example: Data view window of the Valve position chart (growth chamber)

First, on the left side of the window, select one or several curves from the list:

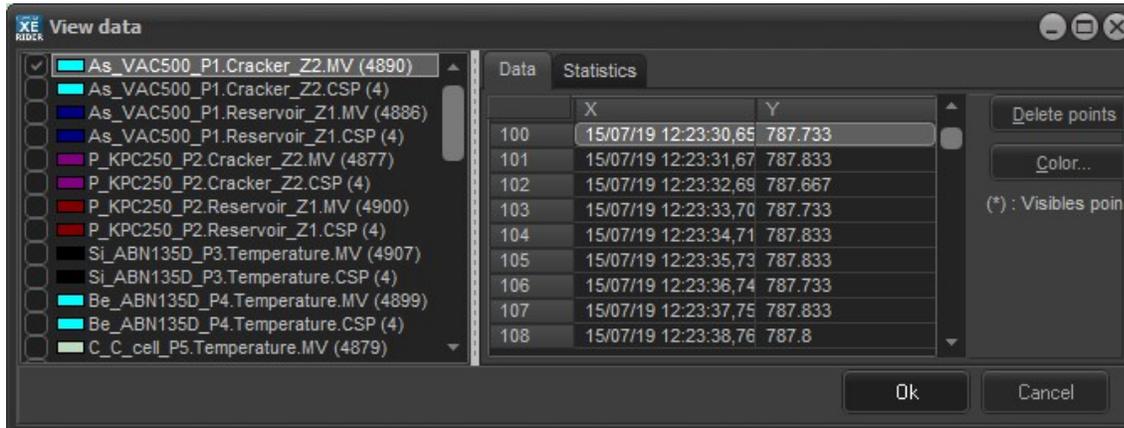


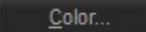
On the *Data* tab, points (data) of the selected curve(s) are displayed in a table (X and Y values for each point).



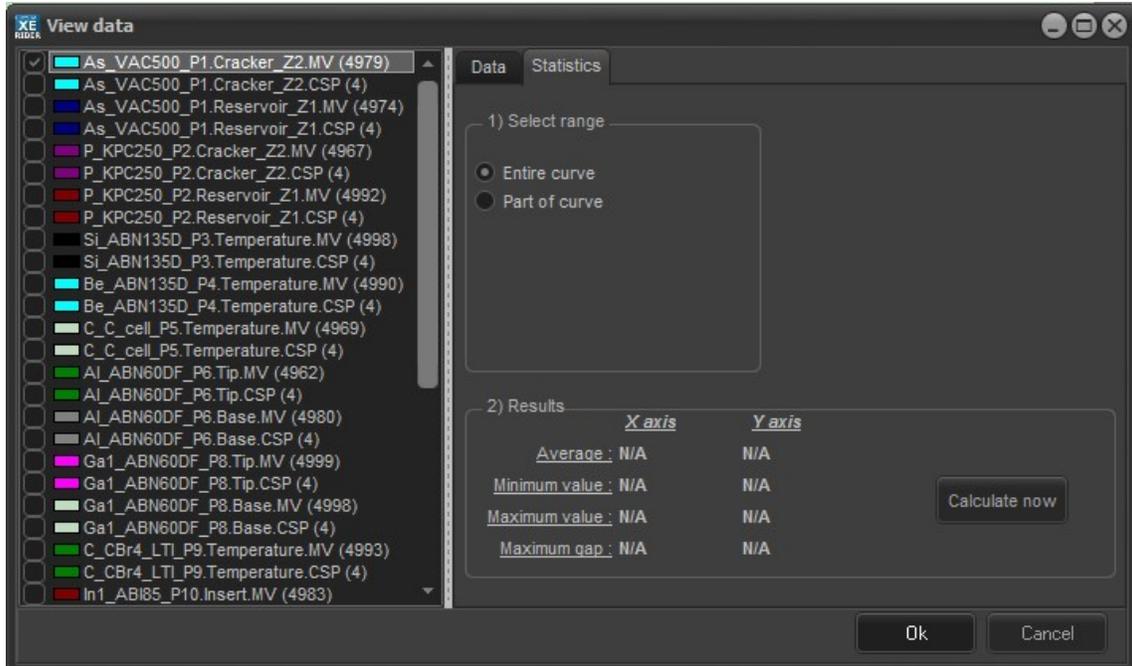
If *Smart optimization* option is enabled, the curve may not display all points (having the same value). To disable *Smart optimization*, right-click the chart background and then unselect *Smart optimization*. For further information, please refer to the part [Smart optimization](#) in this section.

To delete a point, select the point line in the table, and then click on Delete points:



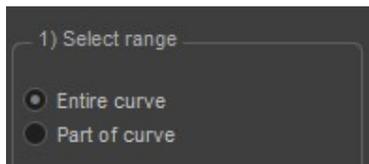
You can modify the color of the selected curve (highlighted in the curves list). Click on the Color button and then adjust the curve color: 

- You can calculate statistics for the selected curve(s) on the *Statistics* tab:

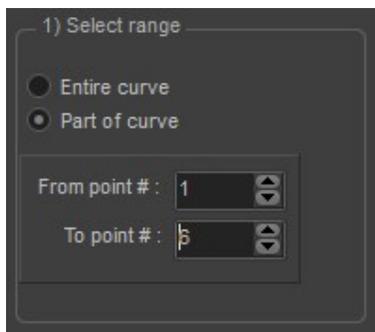


Select range

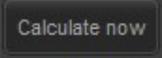
Select Entire curve to display statistics based on all points.

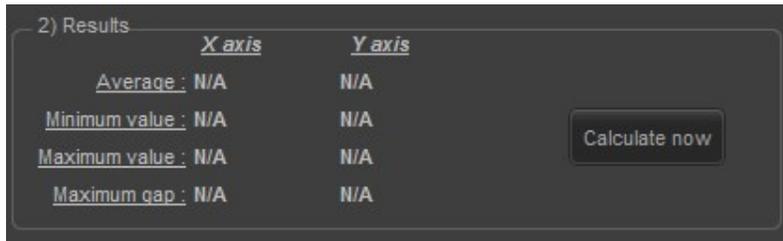


Select Part of curve to define a range of points using the spinner.



Results

Click on *Calculate now*  to display statistics of the selected curve(s).

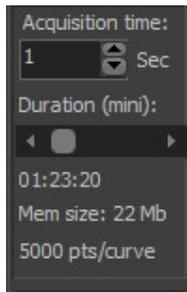


	<u>X axis</u>	<u>Y axis</u>
Average :	N/A	N/A
Minimum value :	N/A	N/A
Maximum value :	N/A	N/A
Maximum gap :	N/A	N/A

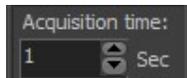
Calculate now

Data acquisition options

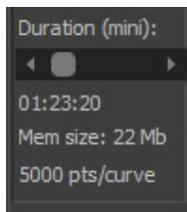
Modify the data acquisition options using the panel located on the left side of your interface:



The acquisition time is the frequency at which a point (data) is plotted on charts. By default, for each curve, a point is plotted every second. You can adjust the acquisition time using the spinner:



Once they are started, charts acquire data over defined time duration. You can adjust this duration using scrollbar button:

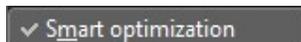


The information bellow indicates memory size the duration would require and the number of points that can be plotted over the duration (depending on the defined acquisition time and whether or not *Smart optimization* option is enabled).

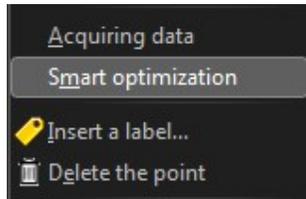
Smart optimization

When *Smart optimization* is selected, when three consecutive points having the same value are plotted on a curve, the second point is deleted automatically.

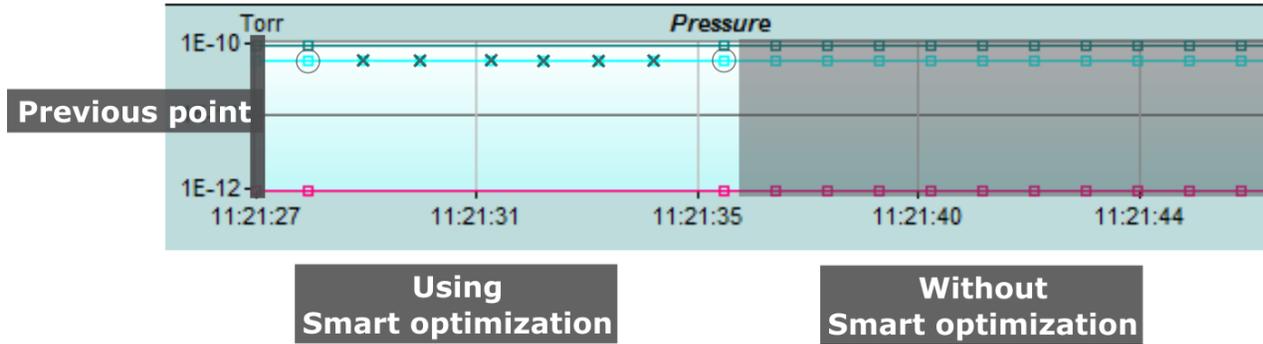
Smart optimization option is enabled by default for all charts:



To disable *Smart optimization* for a chart, right-click the chart background and unselect it:



Example: acquisition time set at 1 second



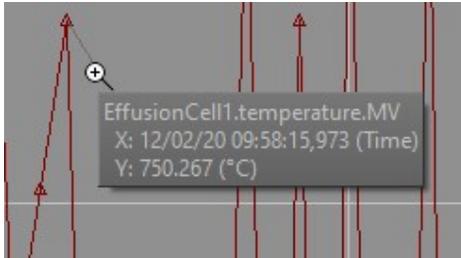
Data		Statistics
X	Y	
26/08/15 11:21:28,092	6.30957E-11	← Using Smart optimization
26/08/15 11:21:36,211	6.30957E-11	
26/08/15 11:21:37,225	6.30957E-11	← Without Smart optimization
26/08/15 11:21:38,240	6.30957E-11	
26/08/15 11:21:39,252	6.30957E-11	
26/08/15 11:21:40,265	6.30957E-11	

Corresponding data view

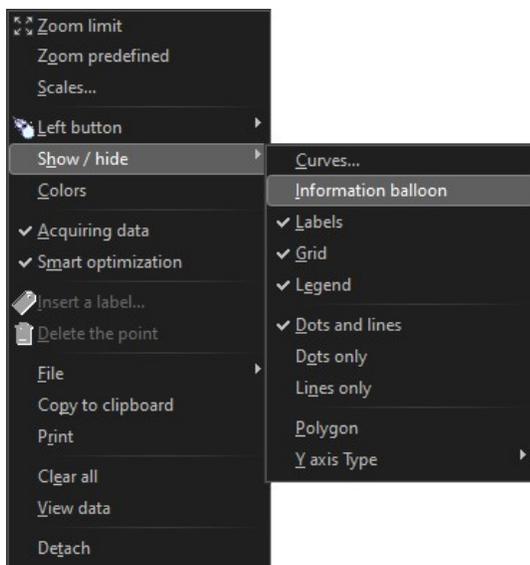
8.5.3. Chart options

Information balloon

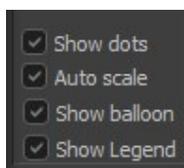
By default when you move the cursor over a point, a link (straight line) appears between the point and the cursor and point information is displayed in a tooltip (information balloon):



To hide the tooltip on a chart, right-click the chart background, select *Show/hide* and then unselect Information balloon from the drop-down menu:

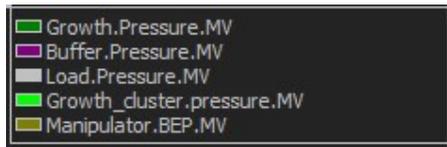


You can hide/display the information balloon on all charts by checking/unchecking Show X/Y in the panel located on the left side of the interface:

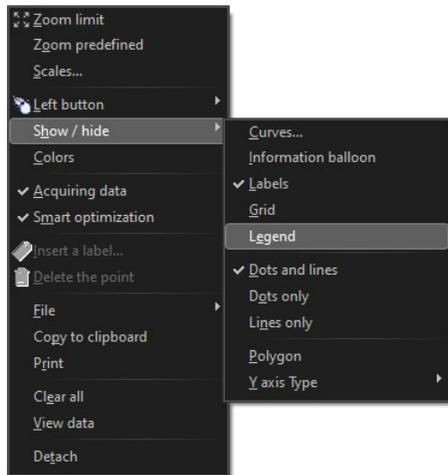


Displaying/hiding legend

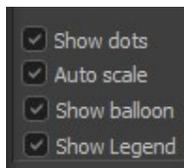
By default a legend is displayed on the left side of all charts:



To hide a chart legend, right-click the chart background, select *Show/hide* and then unselect *Legend* from the drop-down menu:



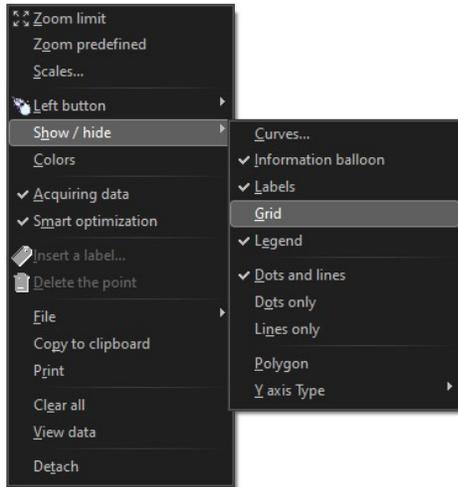
To hide/display the legend of all charts, uncheck/check the box in the panel located on the left side of the interface:



Displaying/hiding the grid

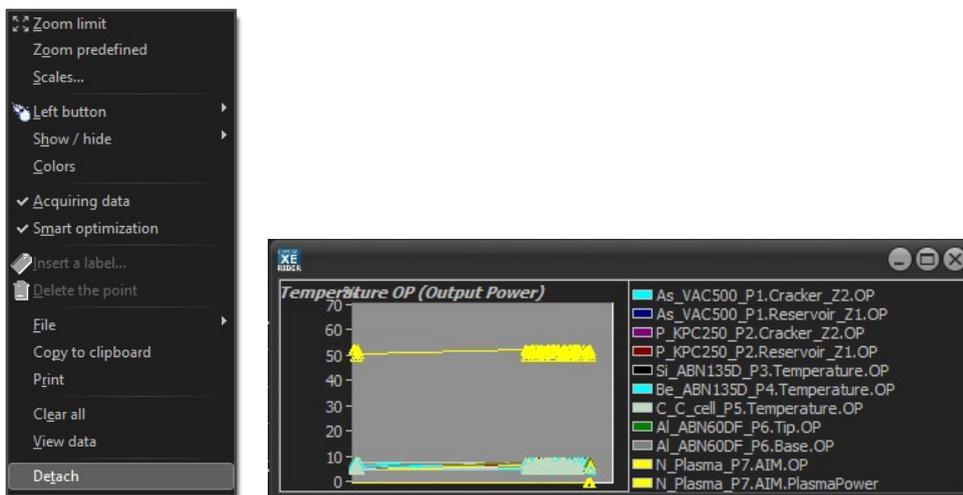
By default the grid chart is displayed in the background.

To hide the grid, right-click a chart background, select *Show/ hide* and then unselect *Grid* from the drop-down list:



Detach the chart

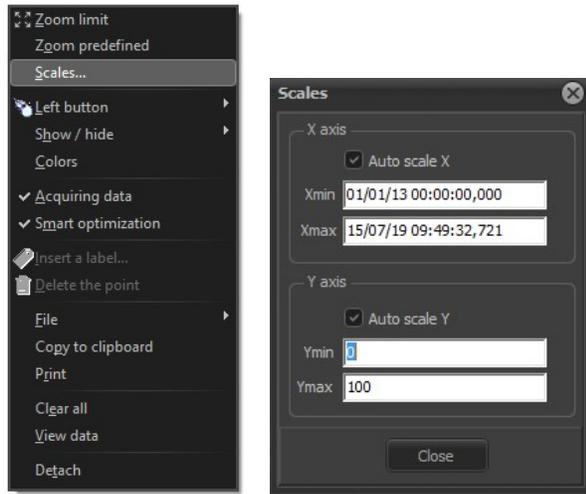
Right-click a chart background and select *Detach* to detach the chart from the main interface and open it in a separate window:



Close the window (click on the cross) to attach it back to the main interface.

Scales ranges and Auto scale

Right-click on a chart and select *Scales* from the drop-down menu to open the scales pop-up window:

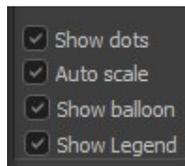


By default *Auto scale* option is enabled for all charts and the last points of the running curves are always displayed automatically. Each time a curve reaches the end of a chart window the X axis is recalculated. Uncheck the box to disable *Auto scale* option.

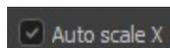


Since all charts share the same X axis, if *Auto scale* is enabled for at least one of the charts that are displayed, the X axis will still be recalculated.

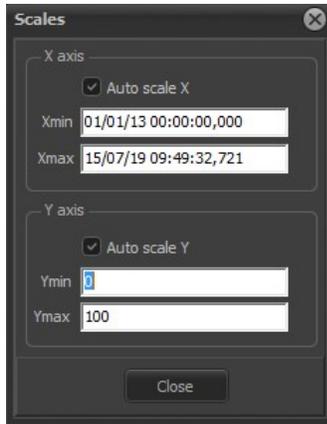
You can disable the *Auto Scale* option for all charts by unchecking the box in the panel located on the left side of your interface:



You can set the ranges of minimum and maximum values for each axis. First disable the *Auto scale* option (uncheck it):

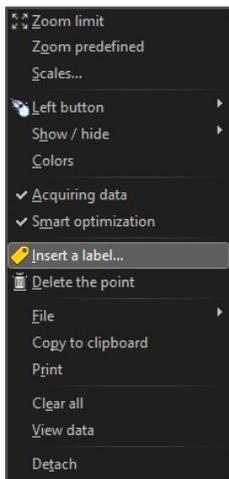


Then enter the desired value in the corresponding boxes:

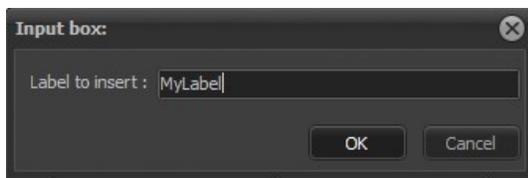


Labels

To add a label to a point, move the mouse over the desired point, right-click and select Insert a label:



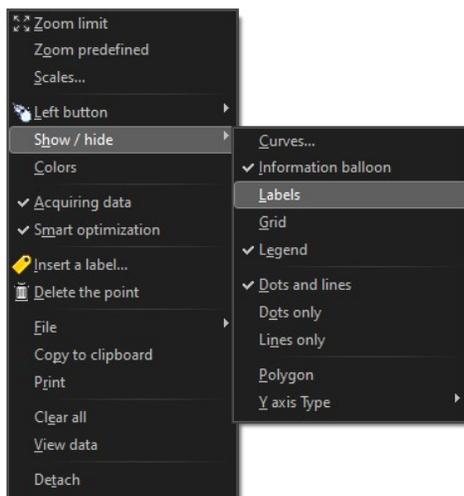
Give the label a name:



The label appears over the point:



To hide the chart labels, right-click on the chart, select *Show/hide* and then unselect *Labels* from the drop-down menu:



① All the chart labels are hidden.

To delete a label, move the mouse over the labeled point, right-click and select *Modify a label*. Then clear the field and leave it empty as follows:

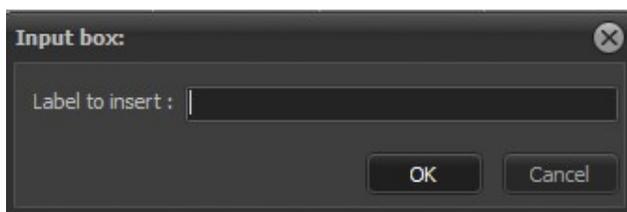
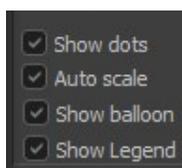


Chart options panel

The following box, located on the left side of the interface, allows you to quickly modify some options for all charts:



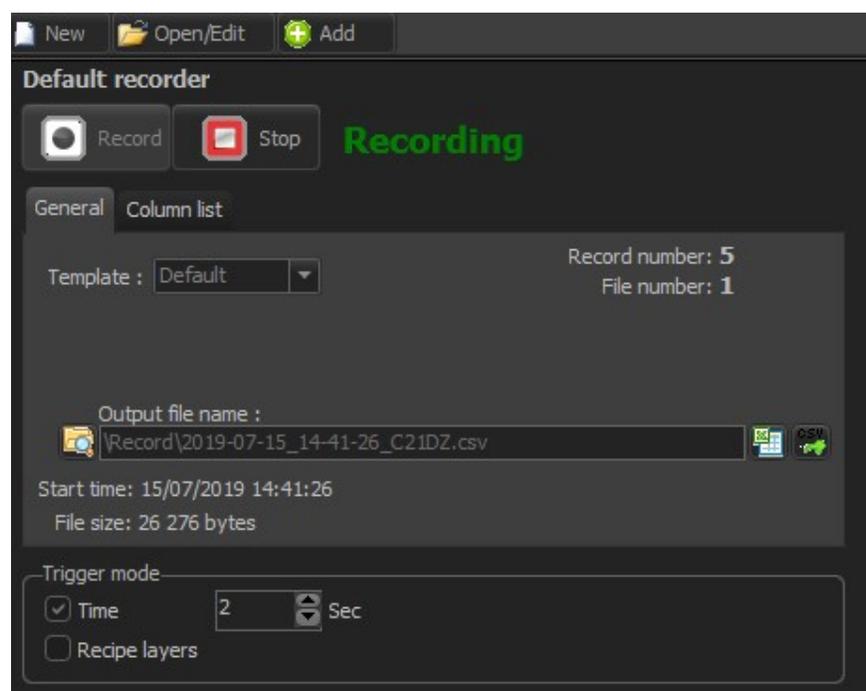
Check/uncheck a box to enable/disable the corresponding option:

- **Show dots:** displaying the chart points
- **Auto scale:** recalculating automatically the X axis when a curve reaches the end of a chart window.
- **Show X/Y:** showing information balloon over a point
- **Show Legend:** showing the chart legend

8.6. RECORDER



The system recorder allows you to save the main data onto the hard disk (default). You can also create your own recorder template and specify the data you want to record.



Recorder window

8.6.1. Recording data

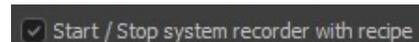
Click on the Record button to start recording data.



A grey Record button indicates that the system is currently recording.



Checking the following box, located at the bottom of the *Recipe* inspector (*Recipe* tab) enables the recorder to start automatically as soon as a recipe starts executing within the chamber:



The recorder stops when the recipe terminates.

The recorded data is saved automatically as '.csv' file (text format) located in:

Project directory > Record

The file names indicate the date, time and chamber within which data has been recorded.

If the maximum number of lines in the file is reached, the data will be saved to a new date and stamped file.

→ For more details about the recorder options, please refer to the section chapter Setup > section **Options** in this manual.

Click on the CSV icon , to copy the file onto your hard disk under a specific name. You can save the data during or after recording.

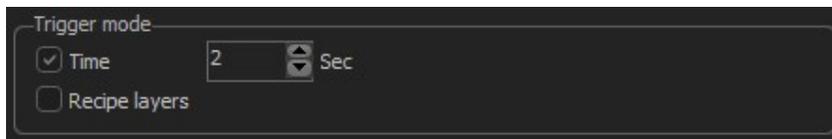
You can view the recorder data at any time by clicking on the spreadsheet icon . The recorded data is displayed in your spreadsheet program.

You can also view the CSV file using *Crystal XE CSV viewer*. On the menu bar, select *Tools > CSV viewer* to open the pop-up window.

→ For more details about the CSV viewer, please refer to chapter General > **Menu bar features** in this chapter.

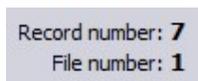
In the CSV file, each line (record) corresponds to either a period of time or a layer.

Select the desired parameter in the *Trigger mode* box:



- **Time:** by default a new record (line) is added every two seconds. Adjust the duration using the spinner.
- **Recipe layers:** a new record (line) is added each time a new layer is executed.

The current number of records (lines) and the number of created files are indicated in the recorder window:



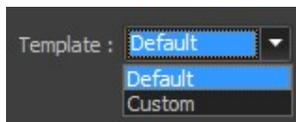
8.6.2. Recorder templates

On the *Template* drop-down menu, you can select either:

- default template

or

- custom template



The default template records all the main data (depending on the equipment set within the chamber) while a custom template allows you to choose only specific data to be recorded.



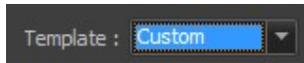
Custom template is recommended in order to reduce the size of the generated file. Custom templates are saved as '.rec' file located in your project directory as follows:

Project directory > Record > Templates

You can also click on the file icon  to open the template directory.

Opening a custom record template

To open a custom template, select *Custom* from the drop-down list:



Then click on the file icon  to select a custom template file:



Four different recorders can be running simultaneously.

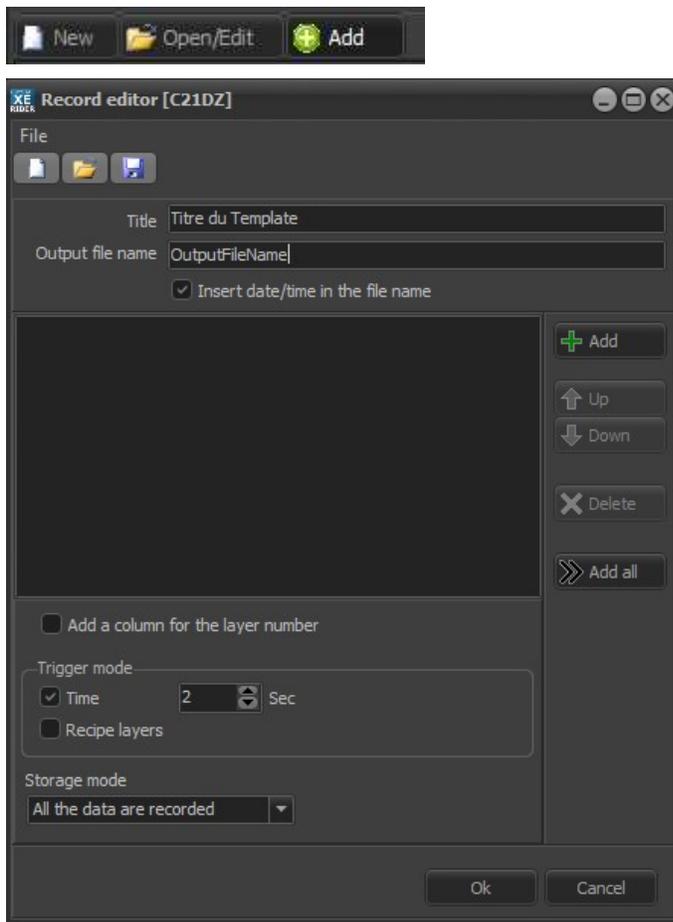
To add a recorder (custom template), click on *Add* and select a template file:



Creating a custom template

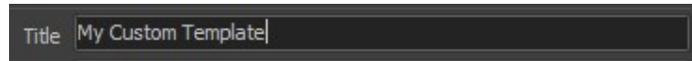
The custom template must be created before recording.

Click on *New* to open the *Record editor* pop-up window and create a new customized template:



Custom template editor

Give the template a title. The title only appears in the recorder window.

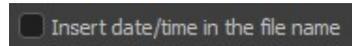


▪ **File name options:**

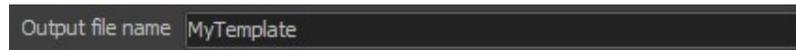
By default the file name is date and time stamped as follows:

'year-month-day_chamber.csv'

You can uncheck the following box so that the date and time do not appear in the file name (not recommended).



You can customize the default name using the following text box:

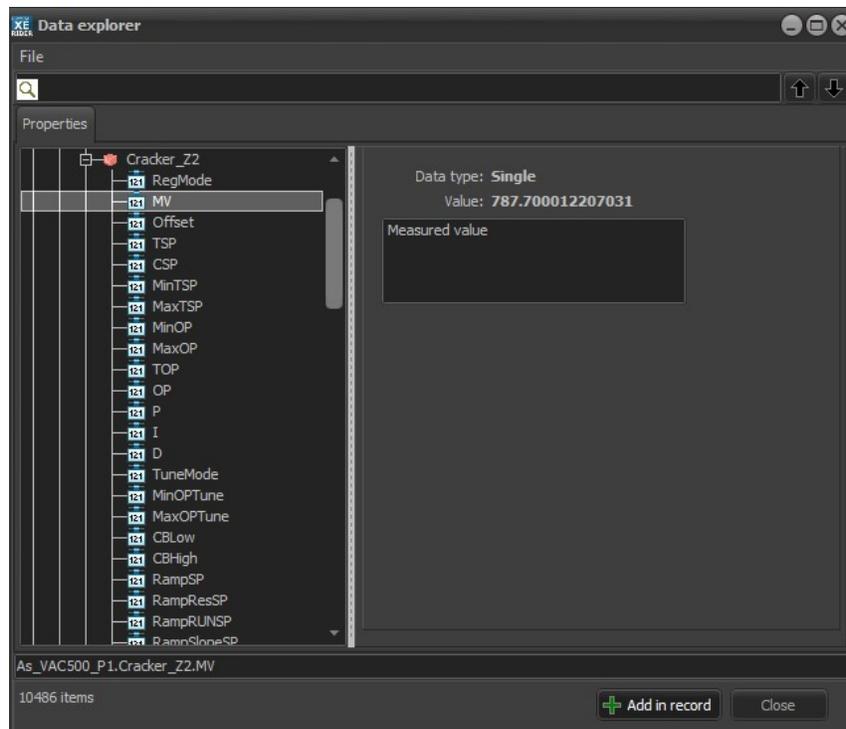


The output file name will be added to the default file name as follows:

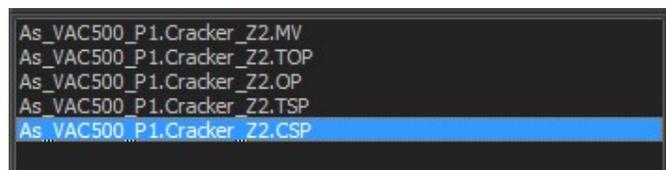
'year-month-day_chamber_MyTemplate.csv'

▪ **Recorded data:**

Click on Add  Add to open the Data explorer and select data to be recorded:



To select properties, double-click on it or click on Add in record  Add in record. The properties are listed in the following box:



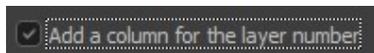
Each property corresponds to a column in the CSV file.

You can modify the order of the recorded properties (columns) using the *Up* and *Down* buttons:

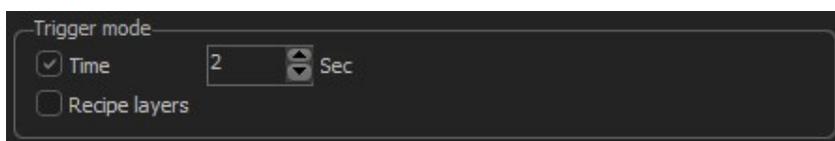


Click on delete to remove a property: 

Check the box to add a column with layer number:

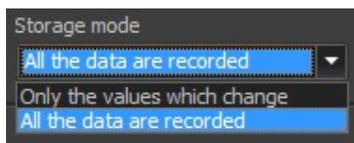


- Use the following box to define the trigger mode by default each time you open the custom recorder:



The trigger mode can be modified later, from the recorder window.

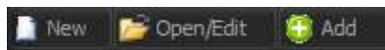
- **Storage mode:**



By default, all values are saved to the CSV file.

To optimize the file size, modify the option so that only data expressing a change in value is recorded. In this case, if the value of a property does not change from one line (record) to another, the cell is not filled until.

- To **edit an existing custom template** file, click on *Open/Edit* to select the template file you want to edit.



If the template has been opened, you can also click on the pencil icon .



If the custom recorder is recording, you can only modify the template using the *Open/edit* button. The modified file will open automatically the next time the recorder starts recording.

8.7. DEVICES



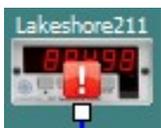
The *Devices* tab is a visual representation of the chamber electronic devices that can communicate with *Crystal XE* and allows you to easily detect a communication or a connection problem.

The connections between the communication ports and the regulators are represented by a wire.

The displayed devices depend on your project configuration. With the demonstration project, the *Devices* tab is only available for the growth chamber (*C21DZ*).

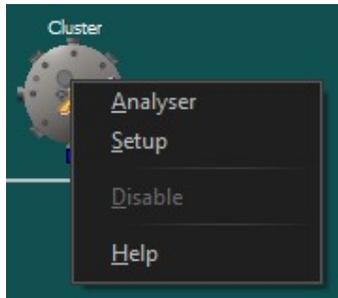
→ For more details about the devices configuration, please refer to the chapter [Hardware configuration](#).

If the connection has failed, the following alert icon will appear over the device icon as follows:



The absence of icon indicates that the connection has succeeded.

Right-click on a device to display the available options:



- **Analyser**: analyzing the connection to the device
- **Setup**: configuring the device (Setup)
- **Disable/Enable**: enabling/disabling the device.

When the device is disabled, the following icon appears over the device:



In simulation mode, all devices are disabled by default.

① Appropriate user rights are needed to perform the above actions.

8.8. STATISTICS

CRYSTAL XE 64 bits - XE52_FR_ALMAE_SYS14930_R100 - Version 2.03 build 6BETA 4 ***** Simulation mode: communication disabled *****

File View Recipe Tools Editors Setup Window Help

R100 ALMAE-SYS14930 RIBER SOFT

Main Growth Prep FEL2 FEL1

Equipment Security Recipe Charts Recorder Statistics Devices

#	Equipment	Sub equipment	Tag	Warning level	Reset date	Value
1	In1_THERMOCELL_P1		HeatingDuration	0	30/12/1899	1m 20d
2		shutter	counter	0	30/12/1899	445
3	Ga1_ABI500_P2		HeatingDuration	0	30/12/1899	1m 20d
4		shutter	counter	0	30/12/1899	414
5	Al1_S100_DZ_P3		HeatingDuration	0	30/12/1899	1m 20d
6		shutter	counter	0	30/12/1899	365
7	In2_ABI500_P4		HeatingDuration	0	30/12/1899	1m 20d
8		shutter	counter	0	30/12/1899	413
9	Be_S63_P5		HeatingDuration	0	30/12/1899	1m 20d
10		shutter	counter	0	30/12/1899	414
11	In3_THERMOCELL_P6		HeatingDuration	0	30/12/1899	1m 20d
12		shutter	counter	0	30/12/1899	422
13	Ga2_ABI500_P7		HeatingDuration	0	30/12/1899	1m 20d
14		shutter	counter	0	30/12/1899	415
15	Al2_S100_DZ_P8		HeatingDuration	0	30/12/1899	1m 20d
16		shutter	counter	0	30/12/1899	361
17	Si_S63_P9		HeatingDuration	0	30/12/1899	1m 20d
18		shutter	counter	0	30/12/1899	412
19	In4_ABI500_P10		HeatingDuration	0	30/12/1899	1m 20d
20		shutter	counter	0	30/12/1899	57
21	Ash3PH3_Inj_P11	shutter	counter	0	30/12/1899	1
22	Cluster1	Control	MovingCounter	0	30/12/1899	0
23	Manipulator		HeatingDuration	0	30/12/1899	<1h
24		Shutter	counter	0	30/12/1899	1
25		Rotation	counter	0	30/12/1899	0
26		PyroShutter	counter	0	30/12/1899	1

All events Criticals and warnings Communication User events

- 2020/02/20 15:30:08: (799) Growth: PSSI Interlock activated: Ion pump
- 2020/02/20 15:30:08: (799) Growth: PSSI Interlock activated: Press 2
- 2020/02/20 15:30:08: (799) Growth: PSSI Interlock activated: Press 1
- 2020/02/20 15:30:08: (003) The project was loaded in 3844 ms
- 2020/02/20 15:30:05: (003) Loading project c:\yiber_app\XE52_FR_ALMAE_SYS14930_R100\

Popup window on alarm
Never

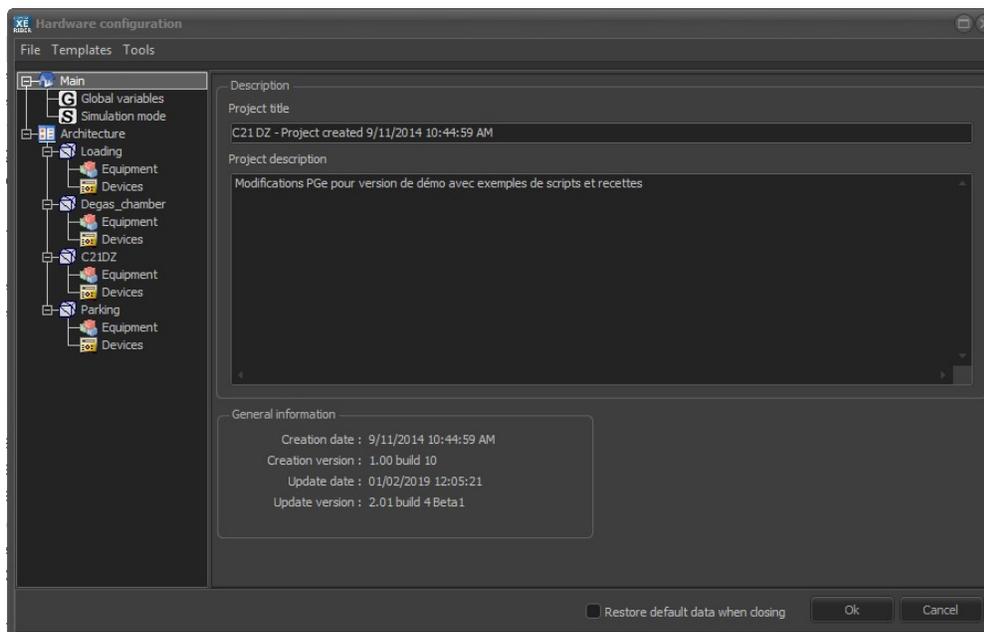
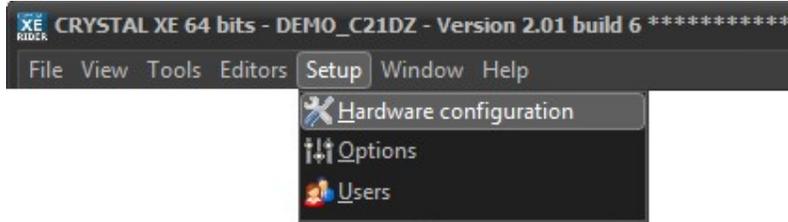
The statistics view is for maintenance purposes. The statistics tab is used to find out how long a cell has been used, how many times a shutter has been used, how many turns a manipulator has performed, etc.

It is possible to reset these counters by double click on it. This action required a password.

9. HARDWARE CONFIGURATION

Click on the following icon  located in the top right-hand corner of the main interface and click on the *Hardware configuration* button 

You can also click on the *Setup* menu and then the *Hardware configuration* submenu to open the *Project configuration* window.



Example of project configuration

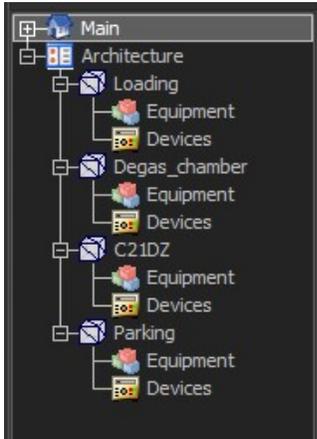
The *Project configuration* window enables you to configure the chambers, equipment and devices of the project.



You cannot modify the hardware configuration when a recipe is running.

All modifications of configuration are saved to the file '*config.xfg*' (XML file) located in the root of the project directory.

Navigate through the window panels using the tree structure on the left side of the window:



9.1. Restoring default setup values

Restore default data when closing

Checking the following box, located at the bottom of the window, restores the default setup values of sub equipment such as minimum and maximum set points, input range, etc.

→ For more details about equipment setup values, please refer to the corresponding chapter [Chambers](#) > section [Equipment view this manual](#).

9.2. Main item

9.2.1. Project information

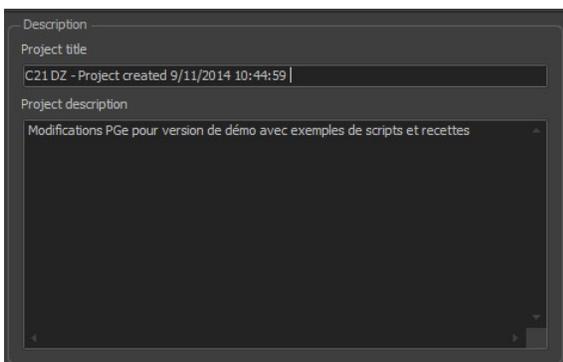
When *Crystal XE* is executed for the first time, the program asks if you want to load the demonstration project. The demonstration project (*DEMO_C21 DZ*) is already pre-configured with several chambers, devices and equipment.

In *Crystal XE*, the project corresponds to the directory located by default in: D:\riber\

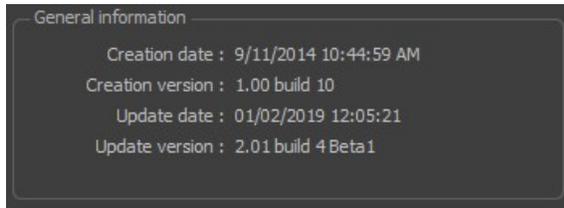


If you need to modify the name of the project directory, you must exit *Crystal XE* first and then rename it.

You can type the title and the description of your project in the corresponding boxes:



The title and description of your project is only for information purpose and do not affect the project.

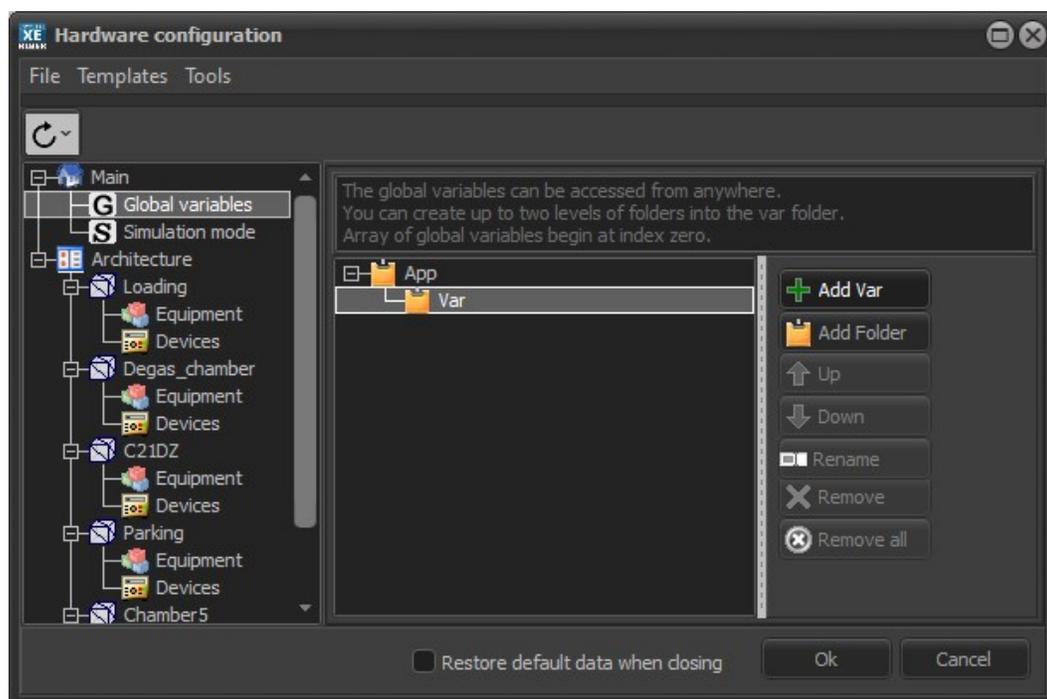


General information about the project and software version

9.2.2. Global variables

A variable is a storage location that contains value. The stored value can be accessed using the variable name (identifier). Each variable can store a defined type of data.

A **global variable** is a variable that is accessible throughout *Crystal XE* program.



Click on the Add Var button to **create a new variable** (You must click on the Var item to enable this button)



Give the variable a **name**:



- A **variable** name cannot start with a number and must start with a letter or the underscore character ('_').
- Variable names can only contain alpha-numeric characters (numbers or letters) and underscore characters.
- The name cannot contain white spaces (blanks).
- Variable names are not case sensitive.
- Different variables must have different names.

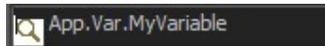
Created variables are listed in the *Data explorer* as follows:

App > Var > MyVariable



Throughout the program, you can access a variable using the following **full path identifier**:
App.Var.MyVariable

The path to access any variable is indicated in the top left-hand corner of the *Data explorer*:



Enter a comment:

You are free to enter a text giving information on the variable.

Select the **type of data** the variable can store from the drop-down list:



Each variable is assigned a data type that determines the possible values it can store:

BYTE: positive number between 0 and 255 (8 bits)

CHAR: a single character

WORD: positive number between 0 and 65535 (16 bits)

BOOLEAN: 0 or 1 (false or true)

INTEGER: negative and positive integers (16 bits)

INT32: negative and positive integers (32 bits)

INT64: negative and positive integers (64 bits)

REAL: 15 significant digits between 1E-308 and 1E+308

STRING: strings of text (words and sentences)

- Array size

You can create a variable dimension array.

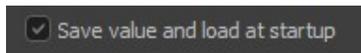


In that case the use of the variable is different. You must specify the index of the array such as: `MyVariable[12] := 345;`

An array of variables can only be used in a script, not in a property of a visual object such as an indicator.

Save value and load at startup

Check the following box to save the last value stored in the variable when exiting Crystal XE and load it at the launch of the program:



The data is stored in the data.dat file into the project directory.

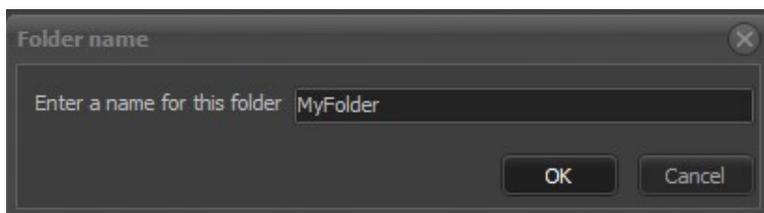
If the box is unchecked, the last value is not saved when exiting *Crystal XE* and the variable will store the default value at start-up.

Assign a **default value** to the variable at the launch of the program:



If you do not assign any value to the variable, the default value will be 0.

You can create **folders** to organize your variables: click on Add Folder  Add Folder (or right-click the tree structure background and select Add Folder) and give the folder a name:



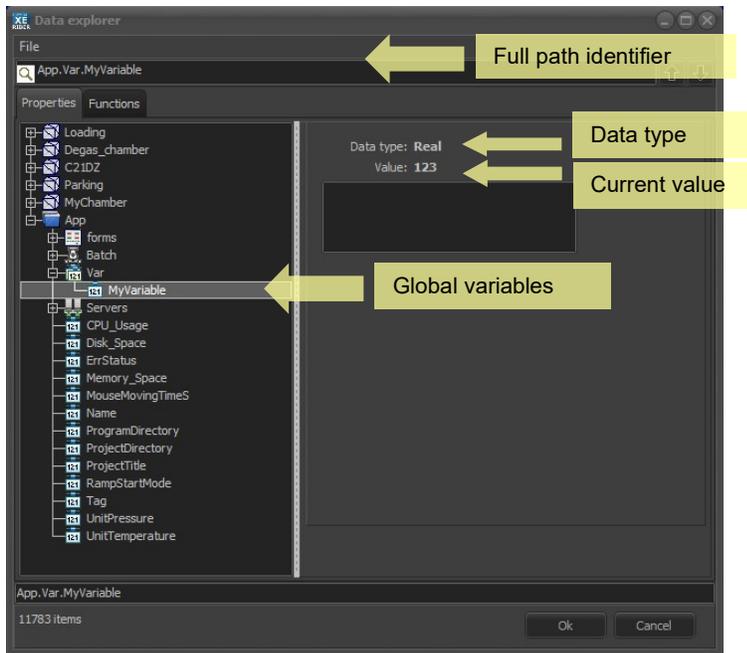
If a variable is located in a folder, the variable full path identifier will change accordingly:

`App.Var.MyFolder.MyVariable`

Move a variable to a folder using drag and drop.



To add a new variable to a folder, select the folder in the tree structure and click on the Add Var button (or right-click on it and select Add var).
 You can modify the list order using the Up and Down buttons. The list order will be the same as in the Data explorer:



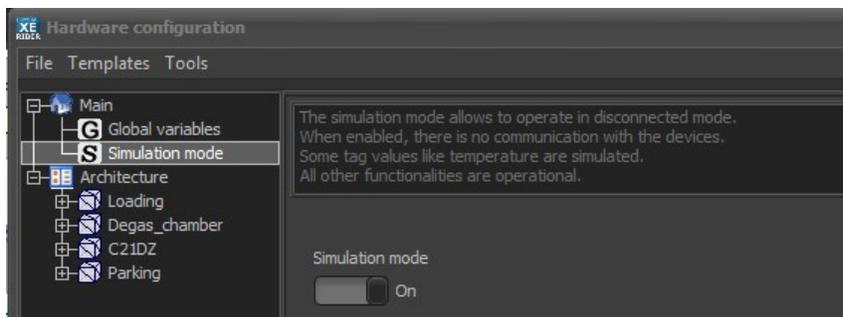
- You can **rename** a variable by:
- clicking on the *Rename* button,
 - right-clicking and selecting *Rename*,
 - pressing F2 key.



Renaming a variable may cause errors if the variable name has been already used in scripts or forms throughout the program.
 The name of the variable is used to access the value it contains.

9.2.3. Simulation mode

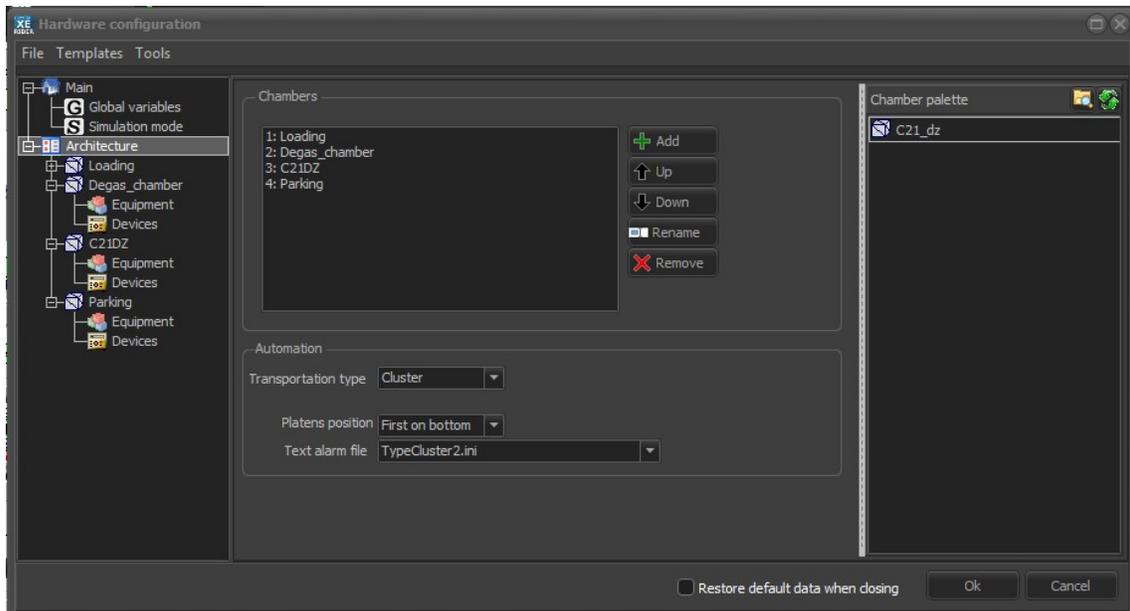
Check/uncheck the following box to switch from *Connected mode* to *Simulation mode*, and conversely:



In simulation mode, all communications with peripheral devices are disabled and some values are simulated.

ASCII servers, modbus servers, cloud and data logger remain functional.

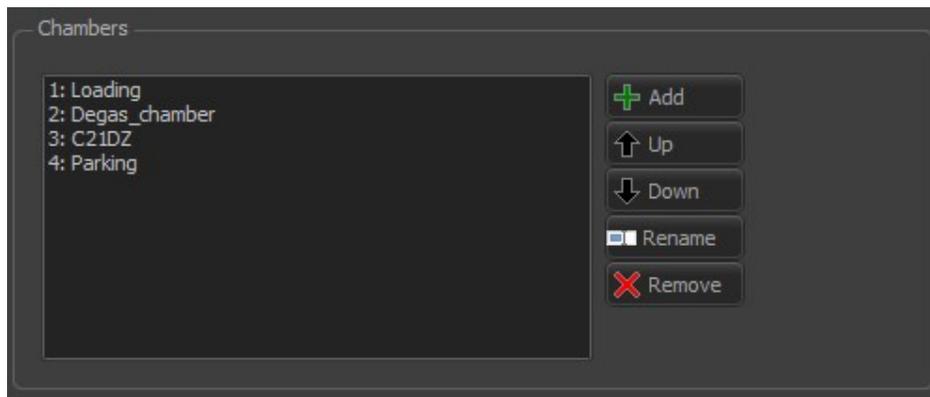
9.3. Architecture item



9.3.1. Chambers

We call chamber, a subsystem that can receive one or more platens through the programmable logic controller (PLC)

All chambers added to your system are listed in the following box:

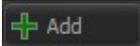


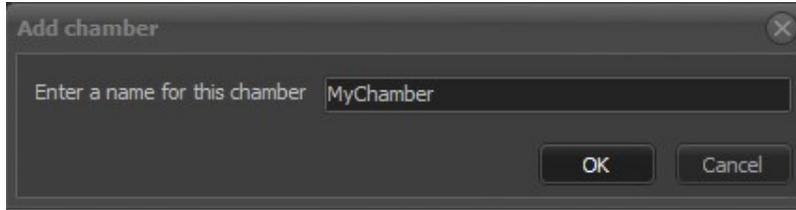
In the demonstration project, four chambers are added to the system by default: Loading, Degas, growth (C21DZ) and Parking chambers. Each default chamber is equipped with a set of preconfigured equipment and devices.





Renaming a chamber may cause errors if the chamber name has already been used in one or several scripts or forms throughout the program.

To add a new chamber, click on the Add button  and enter the chamber name:



You can also right-click on *Architecture* or on one of the chambers in the tree structure and select *Add new chamber*.

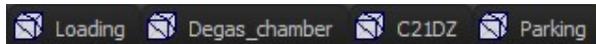


The new chamber added to your system is empty and no equipment and devices are set by default.

Use the *Up* and *Down* buttons to modify the chamber order:



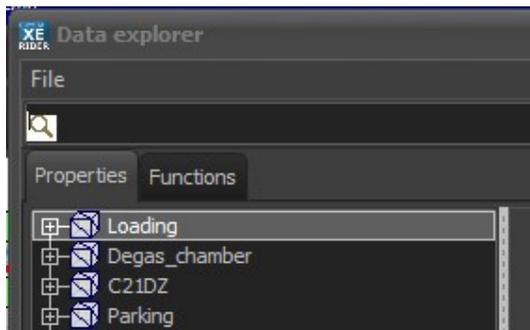
The list order corresponds to the order of:
the chamber tabs on your interface,



the chambers representation during batches (Automation tab),

Loading	Degas_chamber	C21DZ	Parking
--	--	--	--
--		--	--

The list in the Data explorer:



IMPORTANT: for Pick and place type transfer machines, the following order must be observed:

- 1) Load1
- 2) Prep
- 3) Growth

4) Load2

9.3.2. Automation

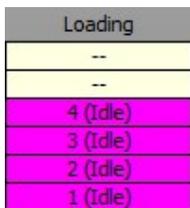
Transportation type:

If your MBE system is equipped with a transfer PLC (Cluster or MBE8000, MBE49 or 6000), you must define which transfer system is used by selecting it from the drop-down list:



Platens position:

The position of the platens can be "First down" or "First up". This information is only used in user views when graphically displaying the position of the plantens.



Example of view

Text alarm file:

Depending on the type of controller, you must specify which file contains information about the text message that is displayed in the history log when an alarm has occurred. This file contains a correspondence between the alarm numbers which is sent by the PLC and the text of the alarm. These files are provided in the template\PLC sub directory of the program directory (by default in c:\riber\crystalxe\template\PLC). The extension of these files is .INI

9.3.3. Chamber palette

The *Chamber palette*, located on the right side of the window, is a library of chambers preconfigured with a set of equipment and devices. By default you can find one preconfigured chamber: C21_dz (growth chamber).



Preconfigured chambers correspond to XML files with '.xfg' extension, located in your program directory as follows:

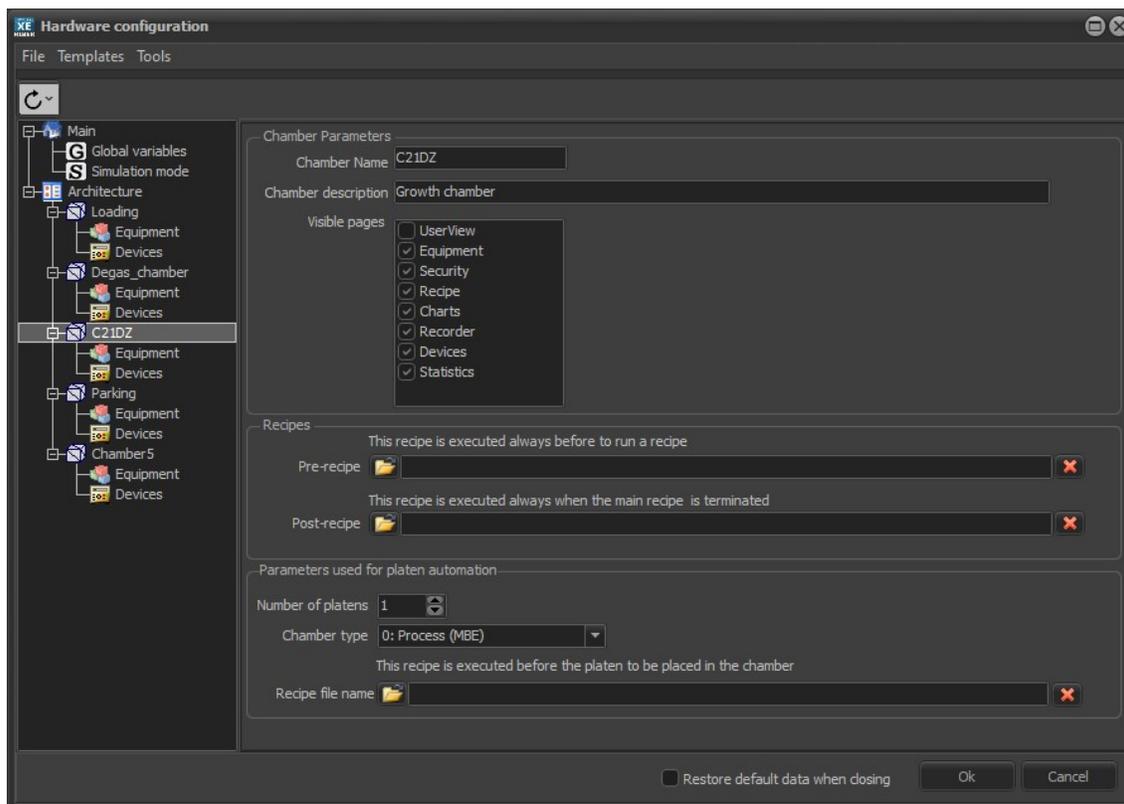
Program directory > Template > Chamber > C21_DZ.xfg

The files can also be accessed by clicking on the file icon . If a new file has been added to the template folder, click on the following icon to refresh the list: .

9.4. Configuring chambers

9.4.1. Parameters

Select the desired chamber in the tree structure to display its parameters



Chamber name:



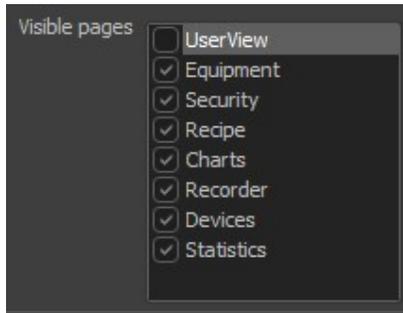
Renaming a chamber may cause errors if the chamber name has already been used in one or several scripts or forms throughout the program.

Chamber description:

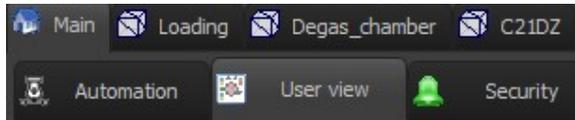
This is a commentary that explains what this chamber is. You are free to enter the text you want.

Visible pages:

The chamber **visible pages** correspond to the chamber tabs on your interface. Check/uncheck the corresponding boxes to hide/display a tab:



By default the *UserView* does not appear in the chamber tabs. Check *UserView* to add a *User view* (form) to the chamber. The new *User view* is empty by default.

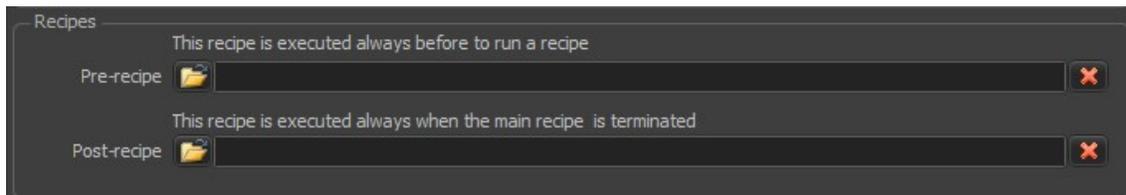


→ For more details about forms, please refer to the chapter [Forms and objects](#) in this manual.

9.4.2. Recipes

A **pre-recipe** is a recipe that is executed before the main recipe. You can specify a pre-recipe to be systematically executed prior to any recipe executed within the chamber.

The **post-recipe** will systematically be executed at the end of any recipe executed within the chamber.

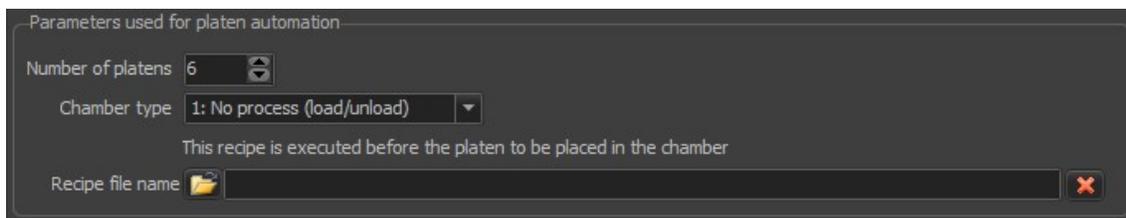


Click on the *File* icon  to select a recipe *File*.

To clear a field and define no pre or post recipe, click on the following button: .

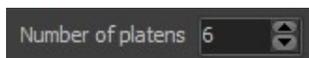
9.4.3. Platens used for platen automation

The following box only appears if your system is equipped with transportation PLC.

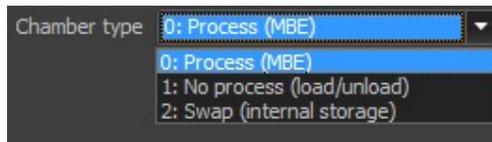


Number of platens:

Set the maximum number of platens that can be loaded into the chamber:



Chamber type:



The type of chamber will only impact the edition of batches:

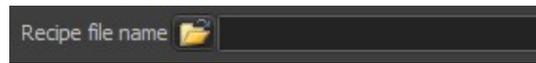
Process (MBE): a batch cannot start or end with a process chamber (such as growth chamber). In batch mode, at least one recipe must be specified with a process chamber.

No process: a batch must start with a no process chamber (such as loading or parking chambers). Recipe cannot be executed within a no process chamber.

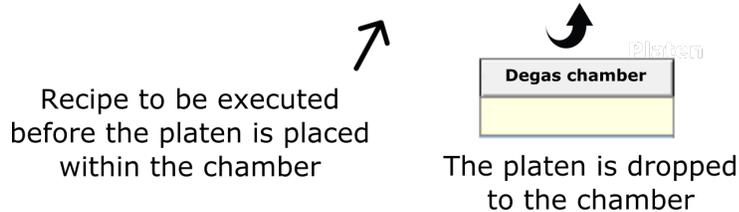
Swap: a swap chamber allows you to swap platens that need to be dropped to the same chamber.

If needed, you can modify the above properties from the batch editor (*Editor options*).

You can specify a recipe to be systematically executed before a platen is placed within the chamber.



Example: C21DZ chamber



9.5. Configuring devices

Select *Devices* in the tree structure to configure the chamber devices:



In *Crystal XE*, a device corresponds to a template file with *.rgr* extension, located in your program directory as follows:

Program directory > Template > Regulator > 'device.rgr'

In this folder, you can also find image files which are used to graphically represent the templates.

You can also click on the file icon to access the *File*: .

If you need to modify any of the device files, create the same directory path in your project directory as follows:

Project directory > Template > Regulator > 'device.rgr'

If you modify files located in the program directory directly, you may lose your modifications if *Crystal XE* is updated. *Crystal XE* will use the template *File* located in the project directory first.

If you need to create a new device based on an existing template, the file name must be different (advanced users).

9.5.1. Devices views

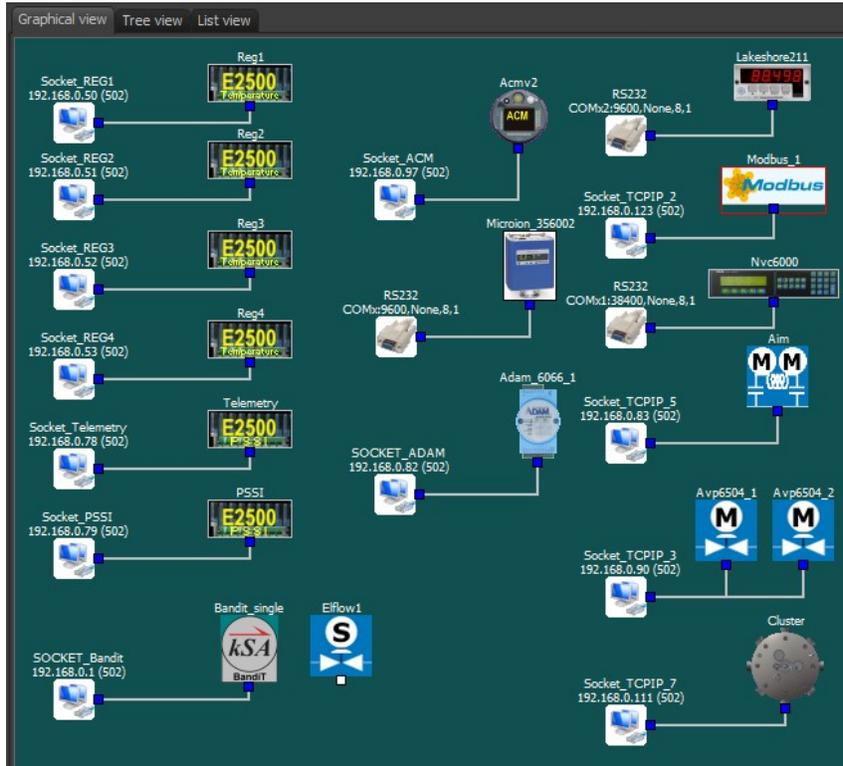
The device configuration interface is divided into three views (tabs):

Graphic view

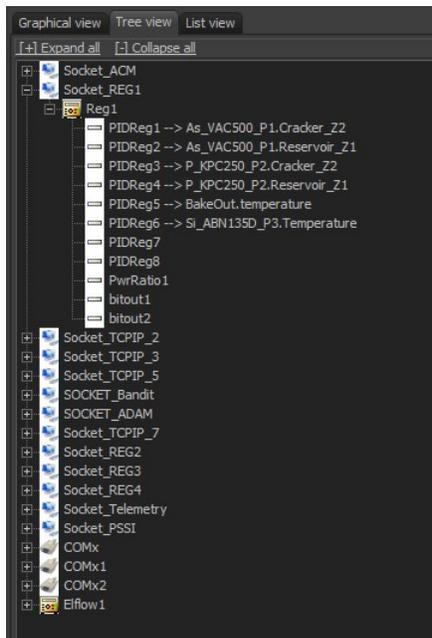
Tree view

List view

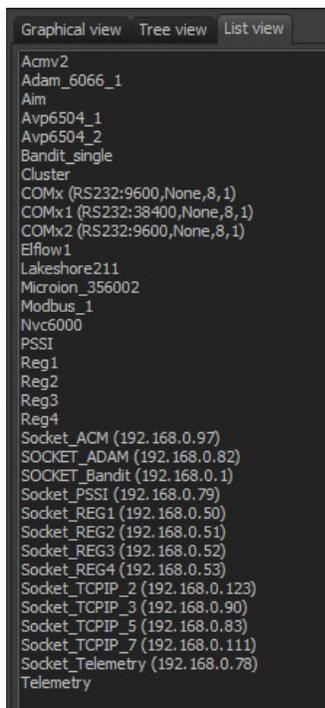
The **graphic view** allows you to add devices (communication ports and regulators) to the chamber, connect regulators to communication ports and configure the devices.



The **tree view** displays the ports and sockets and all devices connected to them. For each device, expand the list to display all the device's channels. For each channel, the sub equipment connected to it, if any, is displayed on the left (next to the arrow: - ->).

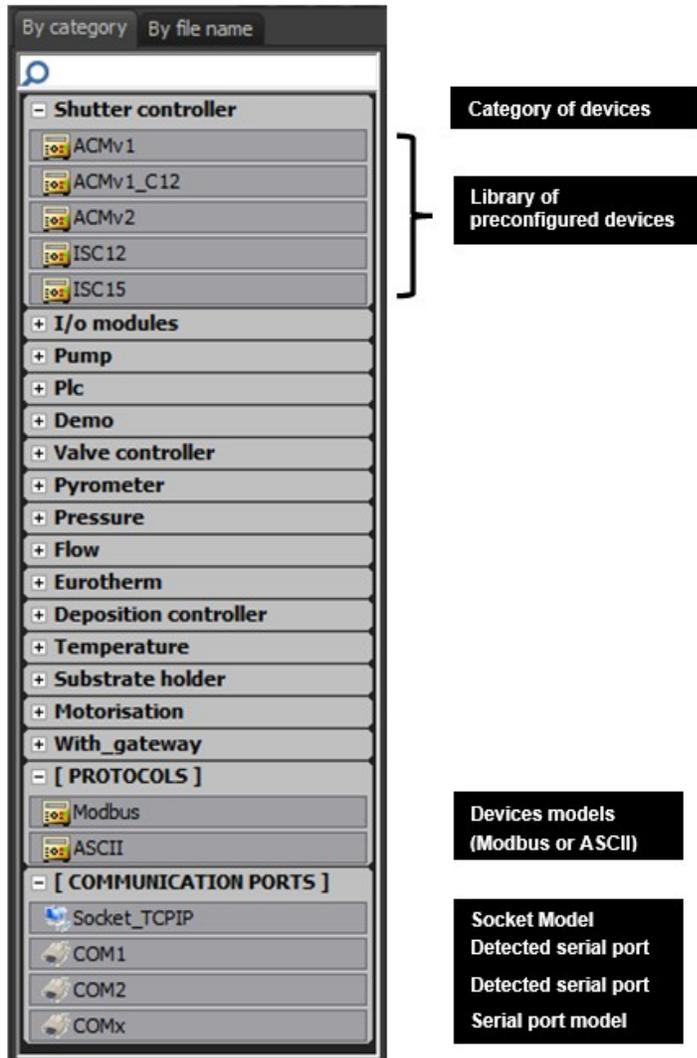


The **list view** lists all the chamber's devices (in alphabetical order) and allows you to configure, rename or remove them. The list view also allows you to select all devices at once.

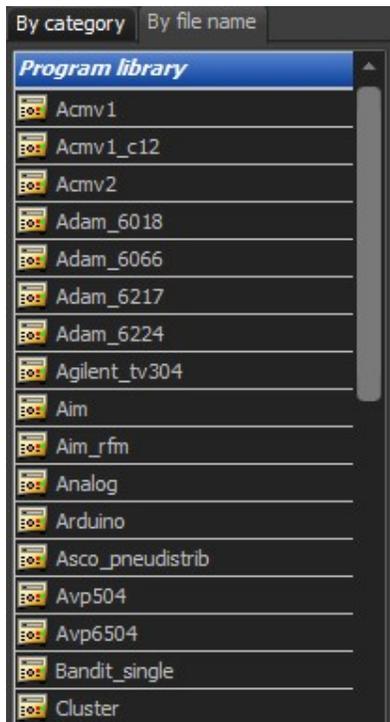


9.5.2. Device palette

The *Device palette*, located on the right side of the window, is the library of available devices. The *Device palette* is divided into two filters: category and file name.



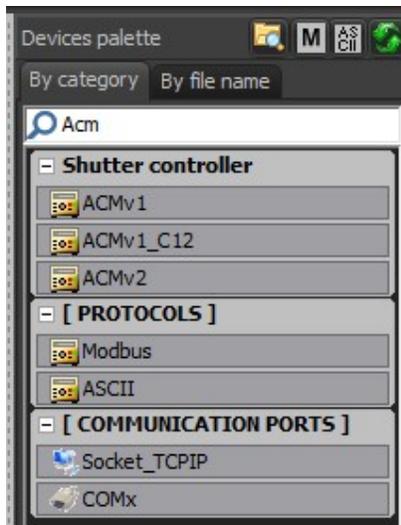
Devices by category



Devices by file name

Search field:

Start by typing the device name to find all the devices containing the letters you enter.



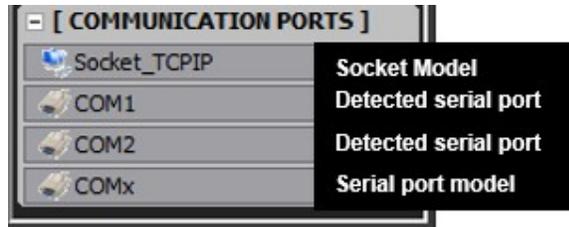
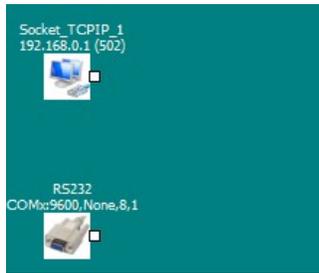
Example with "Acm"

The search is not case sensitive.

How to add a device to the current configuration:

First drag and drop the serial port or socket to the graphic view.

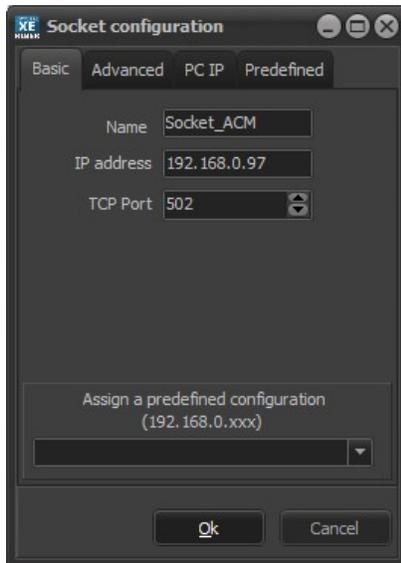
Since the anchor allowing you to create a link is located on the right side, it is recommended to place communication ports on the left side of the graphic view:



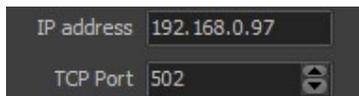
9.5.3. Configuring communication ports

Right-click on the serial port or socket and select *Edit* or double-click on it.

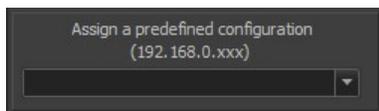
Socket



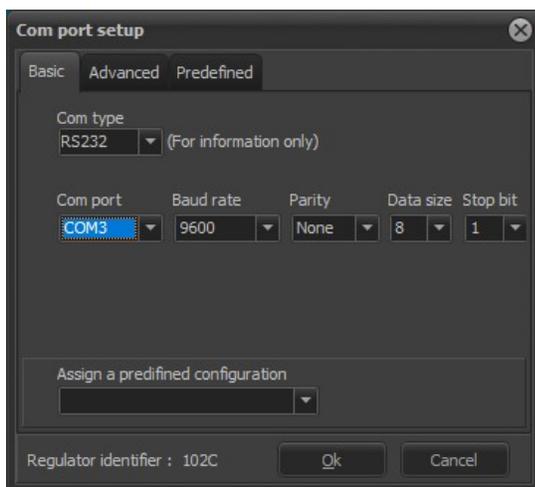
Determine the IP address and the port of *Crystal XE* computer:



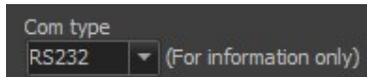
You can also select a predefined configuration from the drop-down list:



Serial port

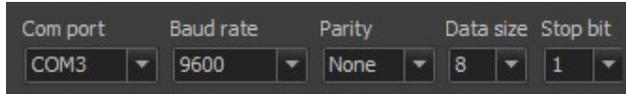


Select the port type from the drop-down list (does not affect the configuration):



The communication type is only text and has not effect on the operation of the system.

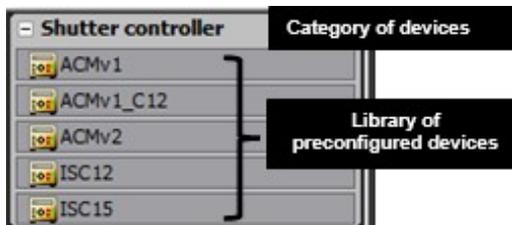
Select a communication port and configure the port:



The same configuration will be needed with all connected devices.

9.5.4. Preconfigured devices

The demonstration project provides a library of preconfigured regulators divided into several categories. Each device uses either Modbus or ASCII protocol.

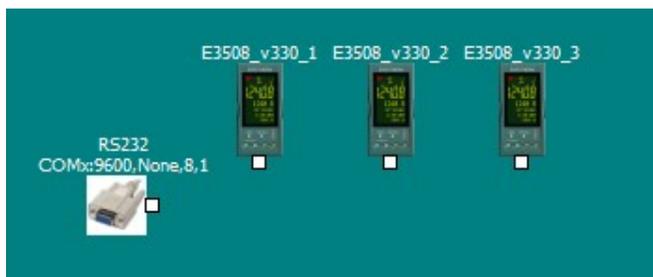


Drag and drop the device onto the graphic view. It is recommended to place the object at the top right of the port connected to it.

Example:



You can connect several devices to one serial port (for RS485 or RS422):



To remove a device:

To remove a device, right-click on it and select Delete.

9.5.5. Device models (Protocols)

If the device you want to configure is not provided by the library, you can create a new device by adding one of the two device models, depending on the protocol it uses (Modbus or ASCII):

Example with the Modbus protocol:



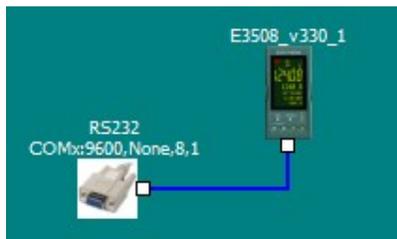
The red rectangle indicates that the module is not in the program template directory and will not be updated with the next release of Crystal XE.

9.5.6. Connecting devices to communication ports

To connect a device to a serial port or socket, move the mouse over the white anchor point until a cross cursor appears:



Then drag the cursor between the port and the device in order to create a link.



To delete a link, right-click one of the two anchors and select *Remove this link*.

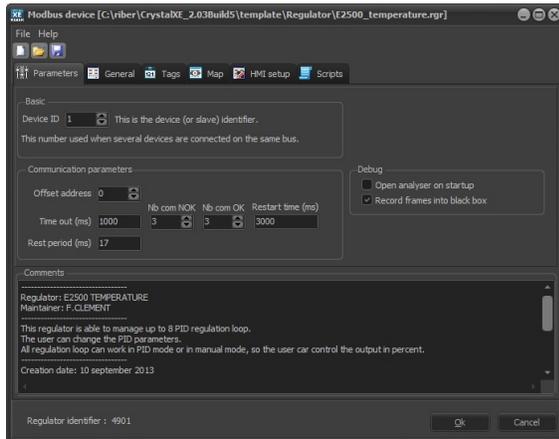
If several devices are connected to a port, you must create each link one after the other:



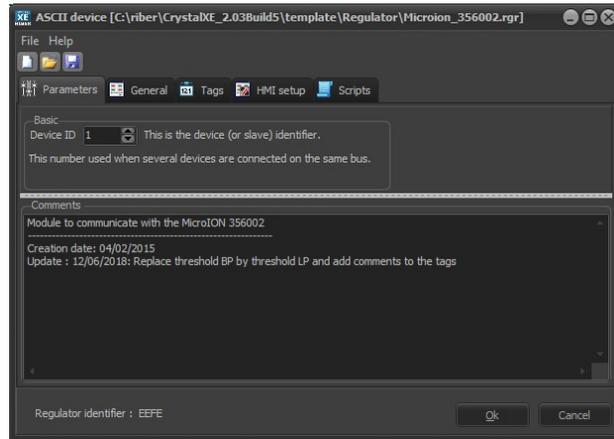
When several links are created, right-click on the anchor of the port to delete all links at once.

9.5.7. Device settings

Right-click on the device and select *Edit* or double-click on it to configure the device:

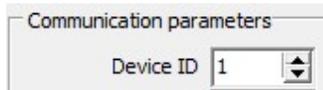


Modbus device



ASCII device

Determine the **device ID**:



The device ID identifies the device on the bus.

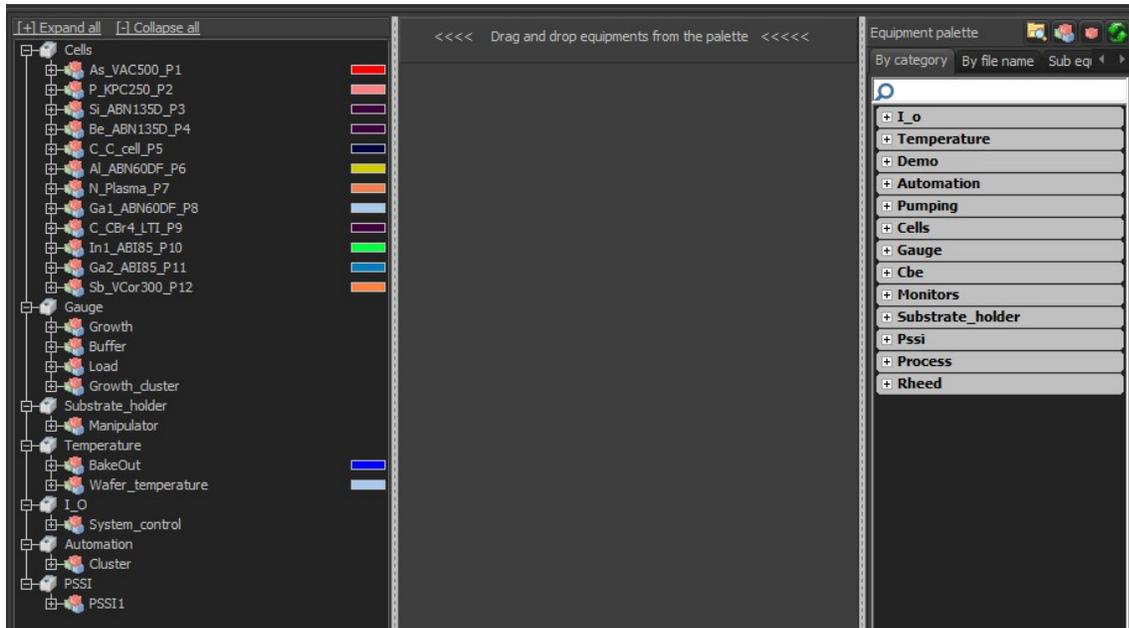
Modifying the offset address (Available for Modbus device only) will shift all addresses in the Modbus table. It is recommended to keep it at 0 (default).



9.6. Configuring equipment

Select *Equipment* in the tree structure to configure the chamber equipment:





Crystal XE provides a library of preconfigured equipment and sub equipment.

Equipment corresponds to template files, with *‘.equ’* extension, located in your program directory as follows:

Program directory > Template > Equipment > ‘equipment.equ’

Each equipment is composed of one or several sub equipment.

Sub equipment corresponds to template files, with *‘.sub’* extension, located as follows:

Program directory > Template > Equipment > SubEquipment > ‘subequipment.sub’

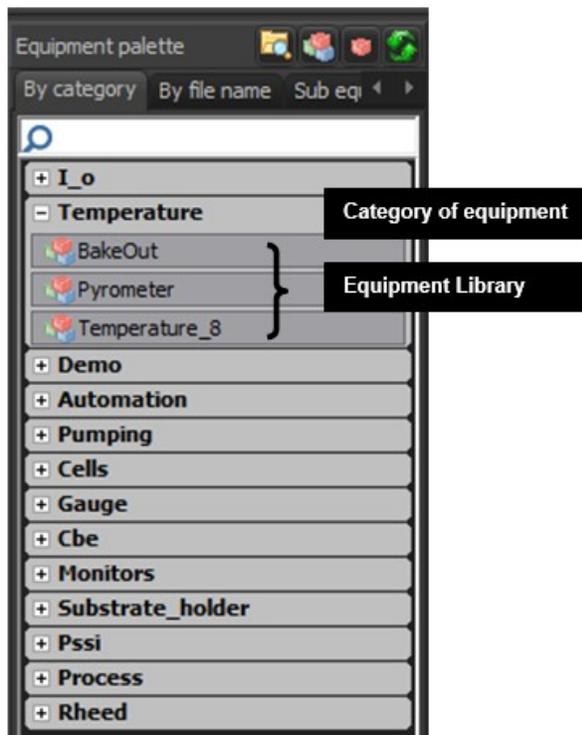
9.6.1. Equipment palette

The *Equipment palette*, located on the left side of the window, is a library of equipment provided by *Crystal XE*. The *Equipment palette* is divided into three tabs:

filter by category,

filter by File name (in alphabetical order),

sub equipment (for reference only).



Equipment by category



Equipment by file name

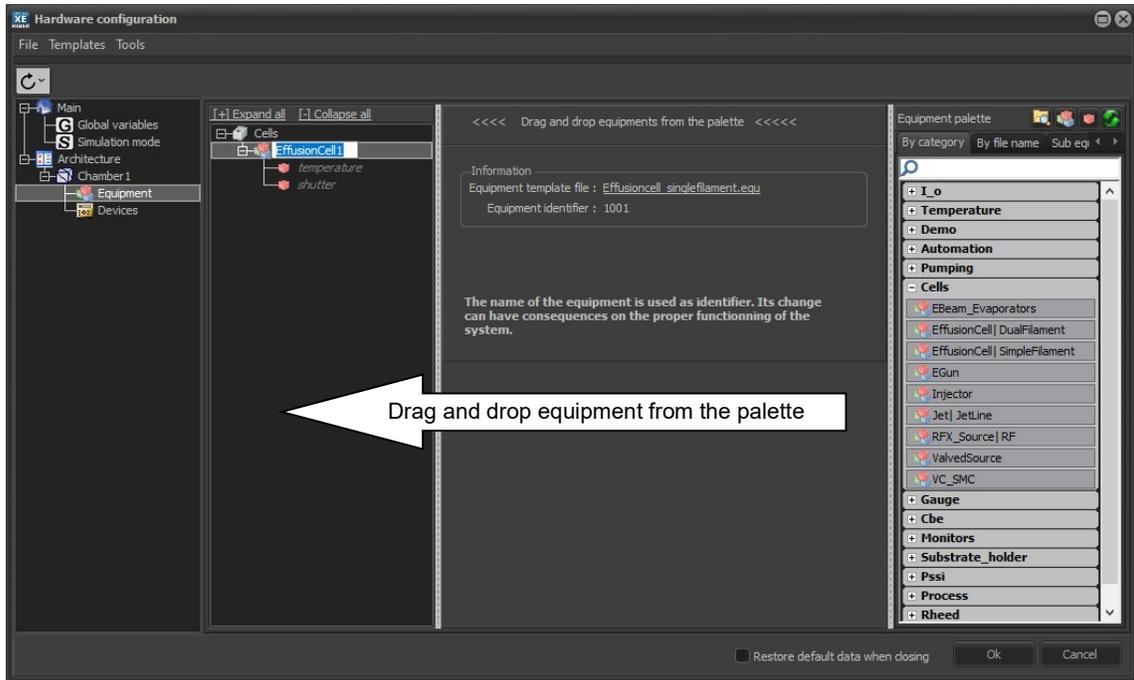
Click on the *File* icon  gives you access to the template files directory.

If a new equipment file has been added, click on the following icon  to refresh the list.

The equipment editor  and sub equipment editor  allow you to create new equipment and sub equipment files (advanced users).

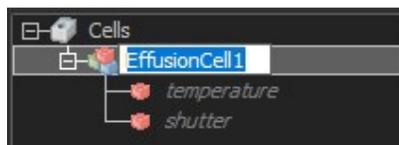
9.6.2. Adding equipment

Drag and drop equipment from the Equipment palette to the yellow box:



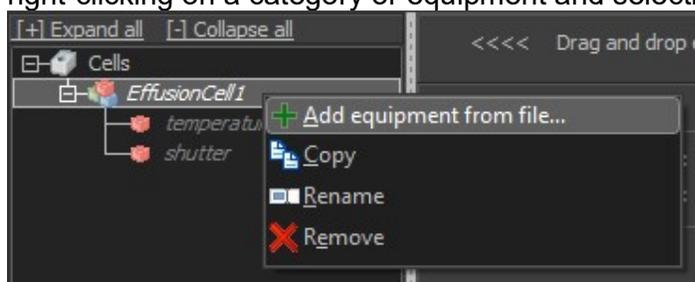
The added equipment appears in a tree structure as follows:

Category of equipment > equipment > sub equipment



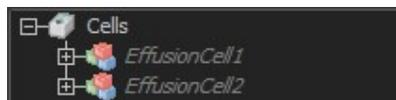
Give a name to the new equipment. The name must respect the name convention (no special characters, no spaces, must not start with a number)

After equipment has been added, you can also add more equipment to a **category** (group) by right-clicking on a category or equipment and selecting Add equipment from *File*.



When you first add a piece of equipment, it is allocated a number, so that you can add several same pieces of equipment:

Example:



Same equipment cannot be assigned the same name within the same chamber. The name of the equipment identifies it throughout the program. However same equipment from different chambers can have the same name, the chamber being used when identifying the equipment:

Example:

Degas_chamber.Cluster

Loading.Cluster

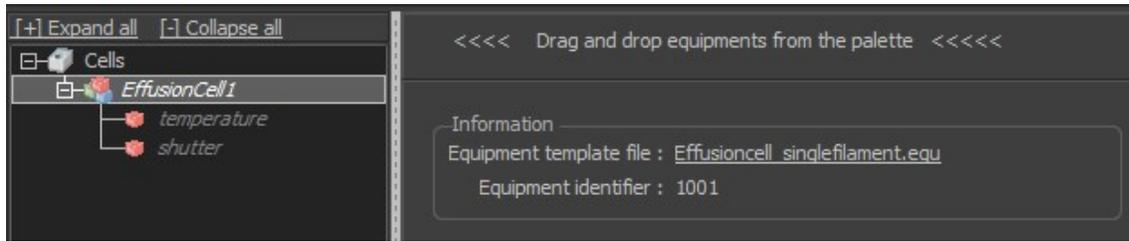
To **rename** equipment, right-click on it and select Rename or select it and press the F2 key.



Renaming equipment may cause errors if the equipment name has been used in one or several script or in forms throughout the program.

To **delete equipment**, right-click on it and select Remove or press the [Delete] key.

Select the equipment to display its information:



Equipment template file

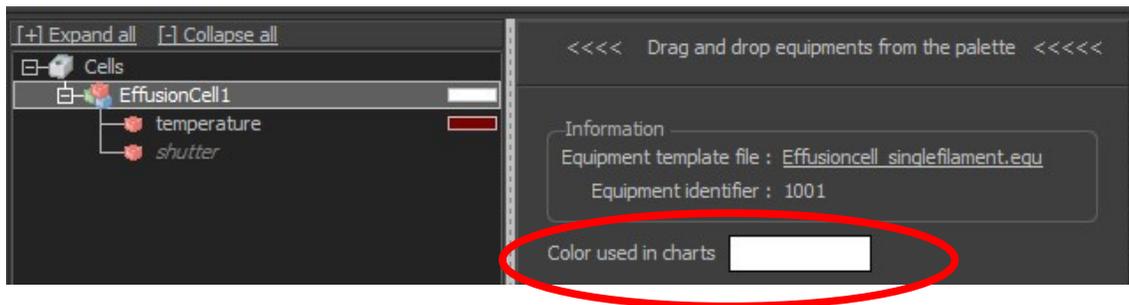
Click on the link to modify the equipment.



Modifying an equipment template will affect all equipment.

Define the **color** associated with the equipment.

Click on the color box to change the color:



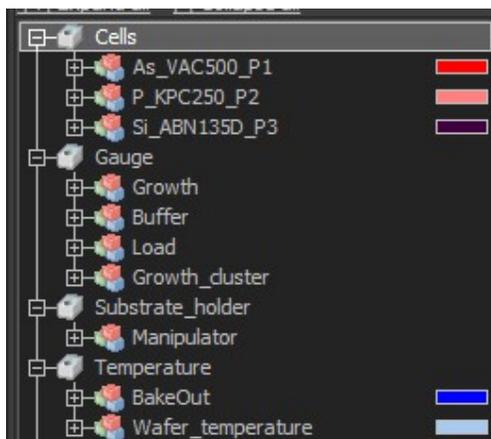
That color will be used in graphics with the equipment curve.



The color option only appears if at least one piece of sub equipment has been configured (= linked to a device channel).

9.6.3. Group of equipment (folders)

Equipment is automatically divided into groups (folders), displayed in the tree structure:



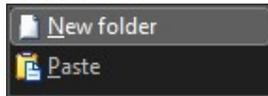
The groups correspond to the equipment categories in the *Equipment palette*.

To **rename** a group, right-click on it and select Rename. Renaming groups does not affect their equipment and sub equipment. Even you rename one of the default groups, when adding

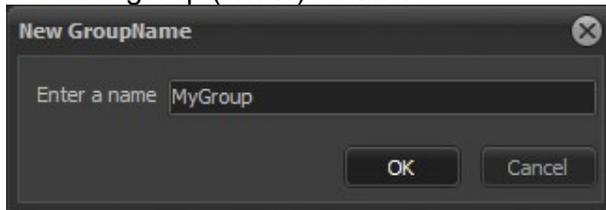
equipment from the Equipment palette, the default group will be also added in the tree structure.

To **remove** a group, right-click on it and select Delete. Removing a group removes all the group pieces of equipment and sub equipment.

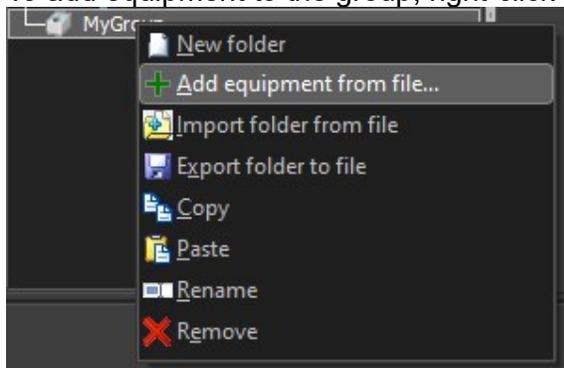
You can **add** new groups by right-clicking the tree structure background and selecting New folder.



Give the group (folder) a name:



To add equipment to the group, right-click on it and select Add equipment from file.

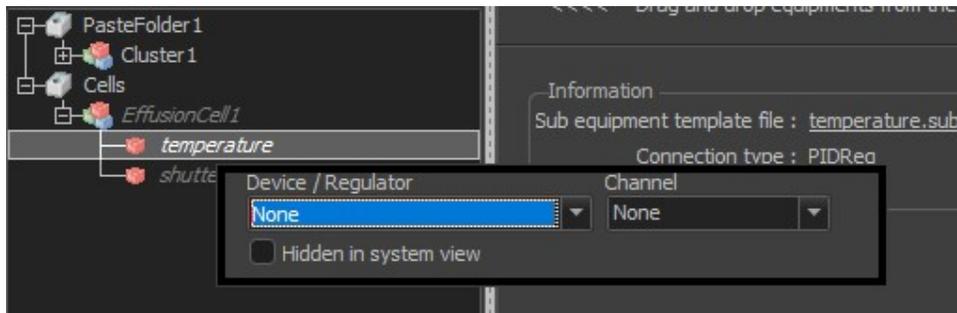


You can copy and paste a folder and its set of equipment and sub equipment either within the chamber or to another chamber. If you copy same pieces of equipment to the same chamber, a different number will be assigned to them.

9.6.4. Configuring sub equipment

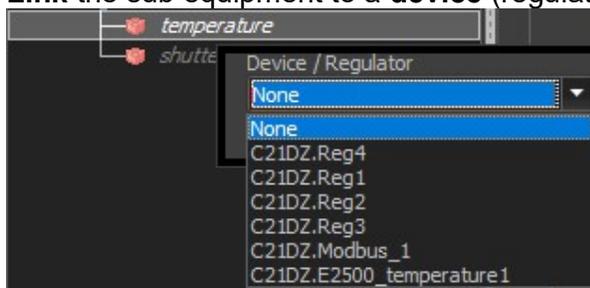
The equipment is added with a defined set of compatible sub equipment.

Click on the sub equipment in the tree structure to configure it:



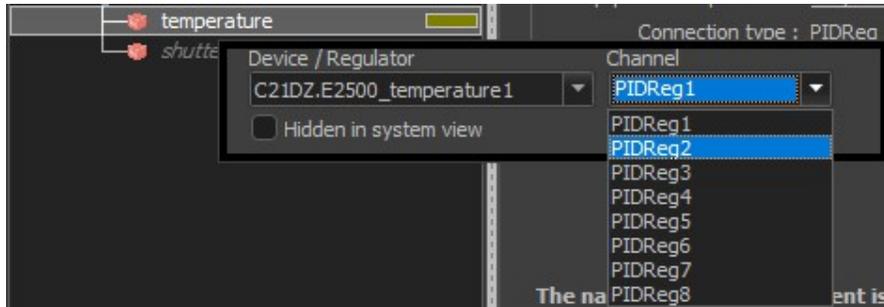
A window appears allowing you to link to a device and a channel.
If the link is already made, this window will not appear below but in the central part of the screen.

Link the sub equipment to a **device** (regulator) by selecting it from the drop-down list:



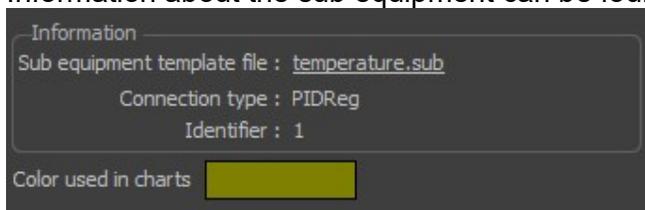
The list displays all devices that are compatible with the sub equipment, within and without the chamber. When a device is set within a different chamber, the name of the chamber appears before the device name. Several pieces of sub equipment can be connected to the same device through its different channels.

Select the desired regulator channel from the drop-down list:



The list displays available channels only. In the above example, channels 1 to 3 are already used. For each channel, only one piece of sub equipment can be connected.

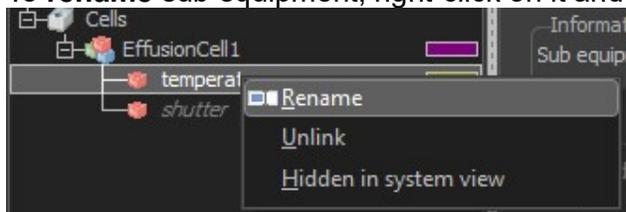
Information about the sub equipment can be found in the following box:



Color used in charts

Define the **color** associated with the sub equipment. Click on the color box to adjust the color. That color will be used in charts with the sub equipment curve.

To **rename** sub equipment, right-click on it and select Rename.



Renaming equipment may cause errors if the equipment name has been used in one or several scripts or in forms throughout the program.

To identify sub equipment, the names of its associated equipment and chamber are used as well.

Example: C21DZ.Wafer_temperature. Pyrometer

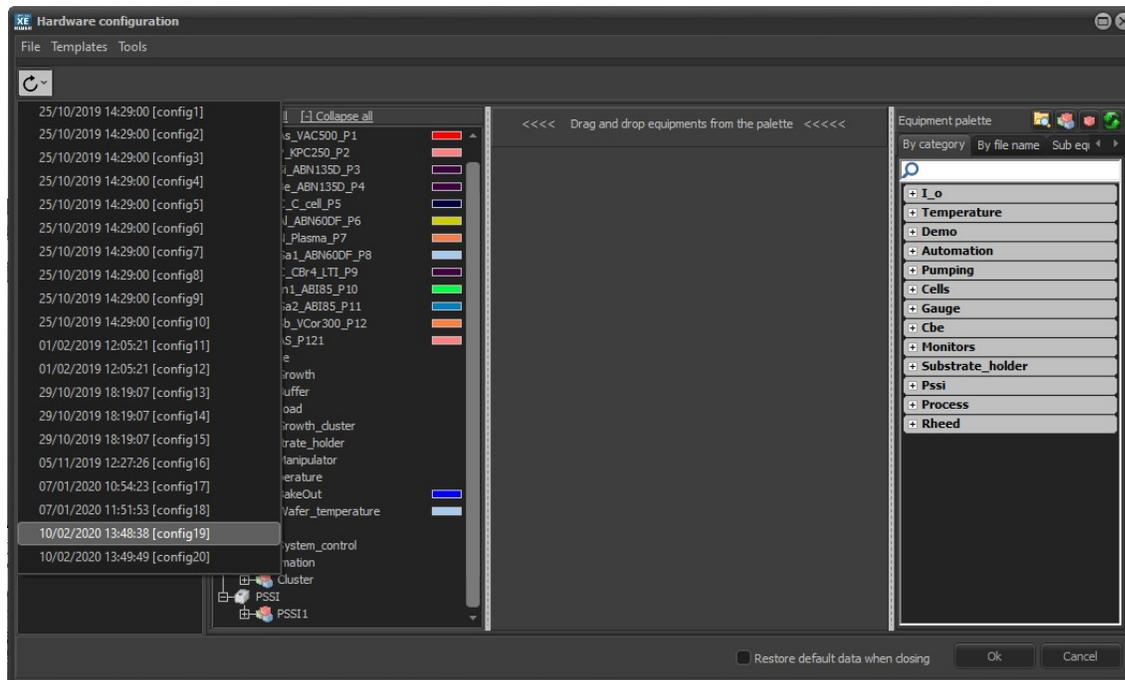
If the name of its associated chamber or equipment is modified, it modifies the identifier of the sub equipment.

End of hardware configuration

Once all the modifications to the hardware configuration have been completed, click on the OK button.

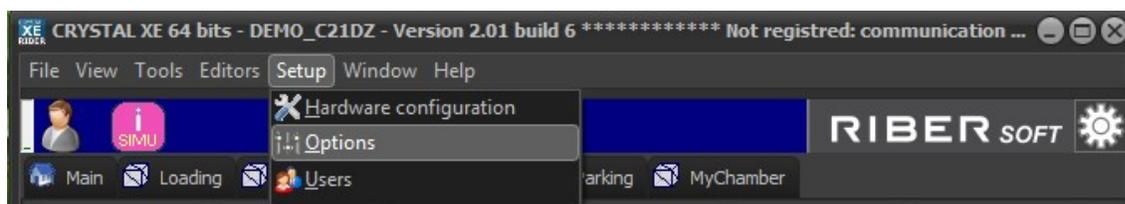
Configuration file:

The last configuration is saved in the config.xfg file which is in the project folder. It is possible to go back to the last 10 modifications. Older configuration files are saved in the backup subdirectory of the project directory. From the hardware configuration editor it is possible to revert to a previous configuration. To do this, click on the button located in the toolbar.



Click on a choice and view the content directly in the editor. You can repeat the operation several times. The modification will not be taken into account until you click on the OK button.

10. OPTIONS



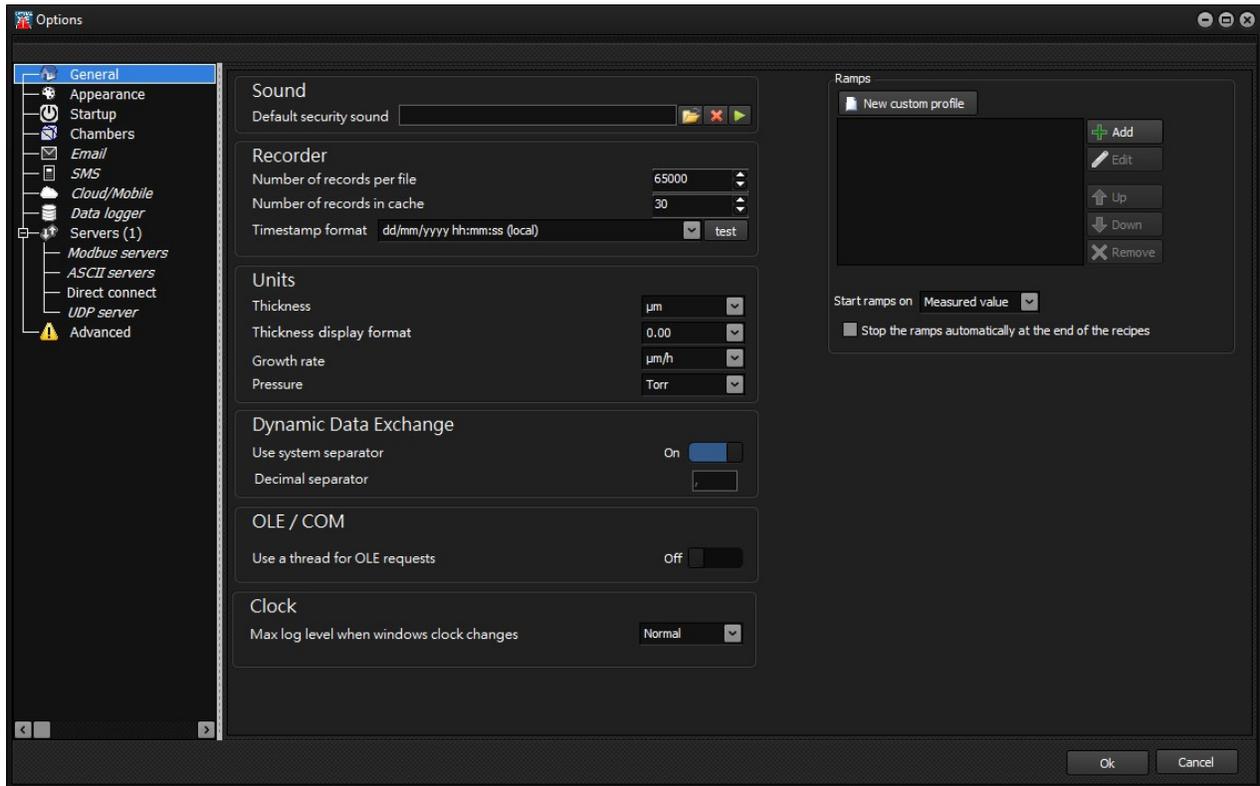
Click on *Options* from the *Setup* drop-down menu.

You can also access the *Options* sub menu by clicking on the following icon: .

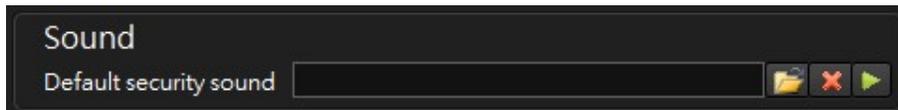
The *Options* window allows you to configure *Crystal XE* software. You can configure the software even if a recipe is currently executing.

All modifications of configuration are saved to the file 'config.xfg' (XML *File*) located in your project directory.

10.1. General options



Sound



Click on the “open file” icon  to select the desired audio file. The provided audio files can be found in your program directory (*Sound*).

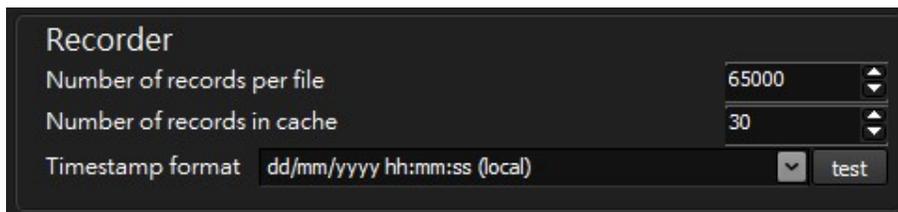


WAV (‘.wav’) audio files only can be used.

Of course, the computer must be equipped with a sound card and speakers.

You can test the sound by clicking on the “play” button

Recorder



When the recorder is running, data is saved in a time-stamped CSV file in the directory <project directory>\Recorder\<year>\<month> until the recorder is stopped.

For example, the full file name might be

D:\riber\XE000_MyProjectDir\Record\2024\07\2024-07-31_14-43-49_REC-Growth.csv

If the file reaches the maximum number of lines allowed, the recorder saves the data in a new time-stamped file.

A line corresponds either to a certain period of time, or to a recipe layer, depending on the trigger mode defined (in the Recorder tab). For example, if you define a trigger time of two seconds, a new line (record) is created every two seconds in the record file.

→ For more details about the system recorder, please refer to the chapter Chamber > section **Recorder** in this manual.

Set the maximum number of lines of the record files:



Number of records per file 65000

Set the number of records in cache to be saved to your file:



Number of records in cache 30

The recorder is always acquiring data (kept in memory cache), even when it is not recording. You can define the number of records (lines) the recorder should keep in memory cache and save to your CSV file.

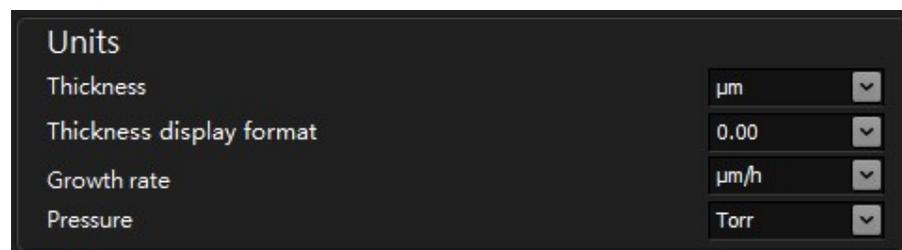
For example, by setting 30, when the recorder starts recording (saving data to the record file), the 30 lines in memory cache are added first to the file. Meaning that, with a time trigger of 2 seconds, the 30 first records of the file correspond to data acquired within the previous 1 minute period (before recording).

Timestamp format

Several timestamp formats can be selected.

You can test the selected timestamp by pushing the button “test”

Units

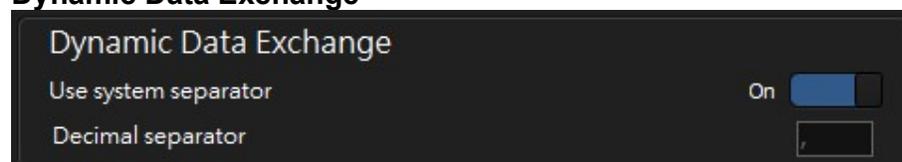


Units

Thickness	μm
Thickness display format	0.00
Growth rate	$\mu\text{m}/\text{h}$
Pressure	Torr

These units are used in the recipe editor when working with layers in thickness.

Dynamic Data Exchange



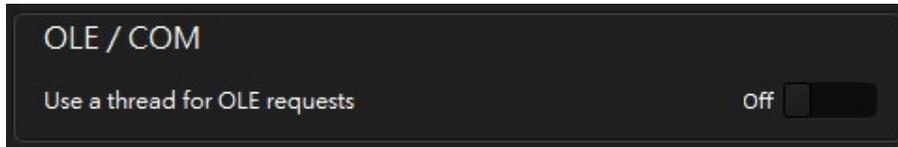
Dynamic Data Exchange

Use system separator	On
Decimal separator	,

Dynamic Data Exchange (or DDE) enables data to be exchanged in real time with another application.

For more information, refer to the section: Dynamic Data Exchange (DDE)

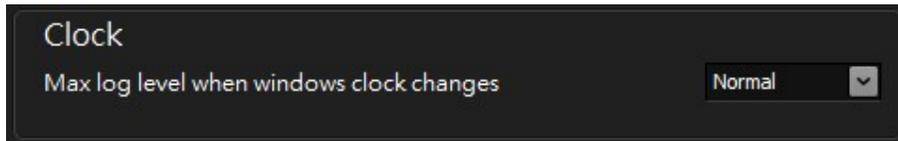
OLE / COM



OLE/COM is used to exchange data with MSOffice and Libre Office.

- For more information, refer to the sections:
- Dynamic Data Exchange (DDE)
 - Automatic report generation

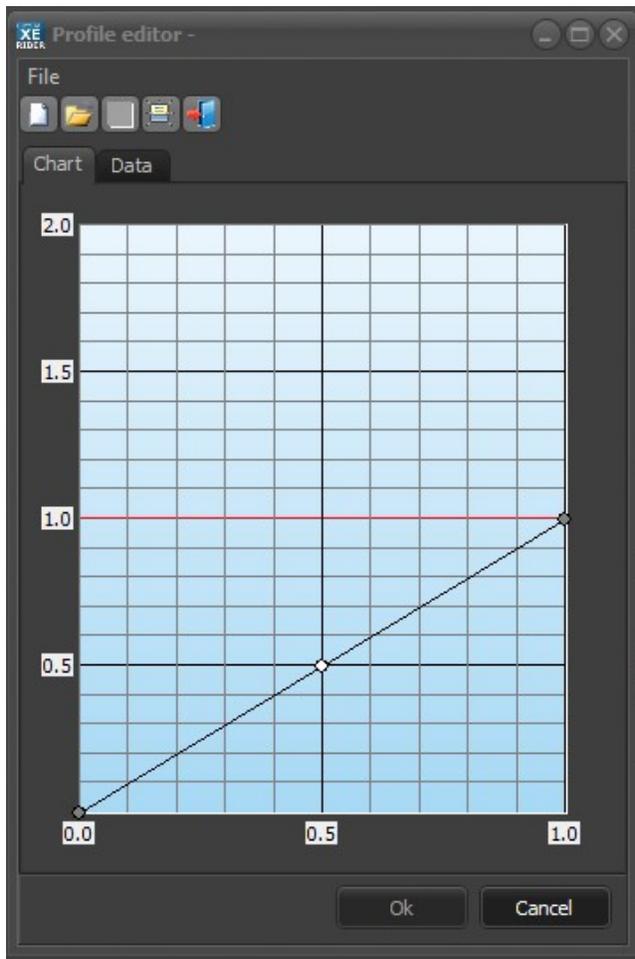
Clock



This option define the level of the event when the clock change (by the user or by the automatic update)

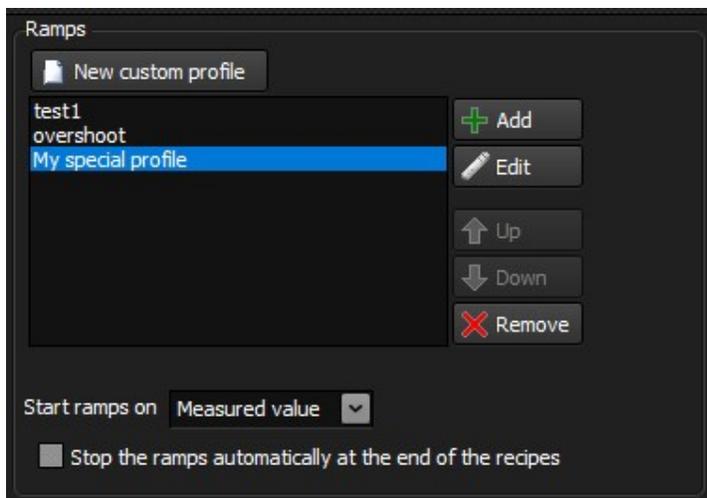
Ramps

By default, a ramp allows you to increase or decrease a value linearly over time (set point, output power, etc.).



Default ramp representation (Linear ramp)

You can also define the list of all the profiles you need, using the pane below:

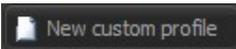


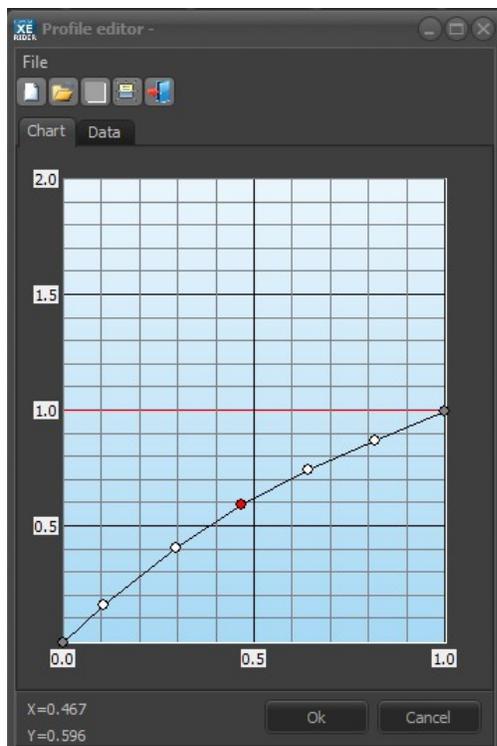
Start ramps on, let you the choice between:

- Measured value
- Current setpoint.

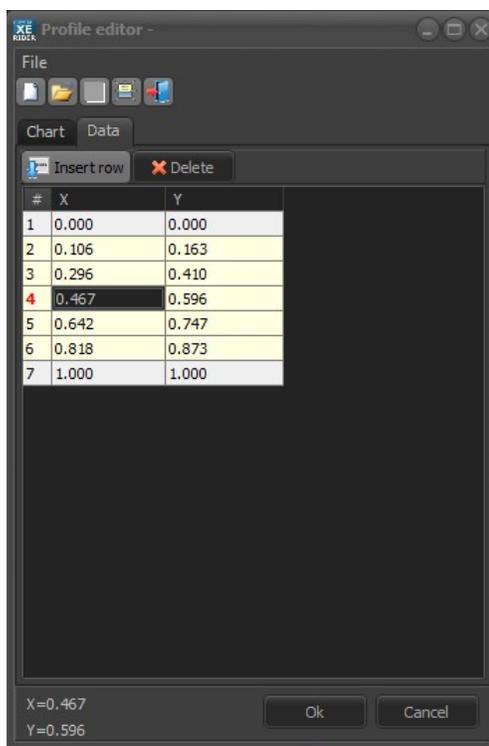
It is also possible to choose the behavior of ramps at the end of recipe execution. This only concerns ramps that have been launched in recipes.

Create a new profile:

To create a new profile, click on New profile  to open the profile editor.



Example of profile file: Chart



#	X	Y
1	0.000	0.000
2	0.106	0.163
3	0.296	0.410
4	0.467	0.596
5	0.642	0.747
6	0.818	0.873
7	1.000	1.000

The status bar at the bottom shows X=0.467 and Y=0.596.

Example of profile file: Data

The profile editor is divided into two tabs:

Chart tab: the data is represented visually in line chart

Data tab: the data is represented numerically in table

On the *Chart* tab, the profile is represented by a line with time on X axis. The Y axis depends on the value to be ramped (temperature, output power).

The range of values on both the X and the Y scales is 0 to 1.

Two points are automatically plotted and cannot be modified (gray points):

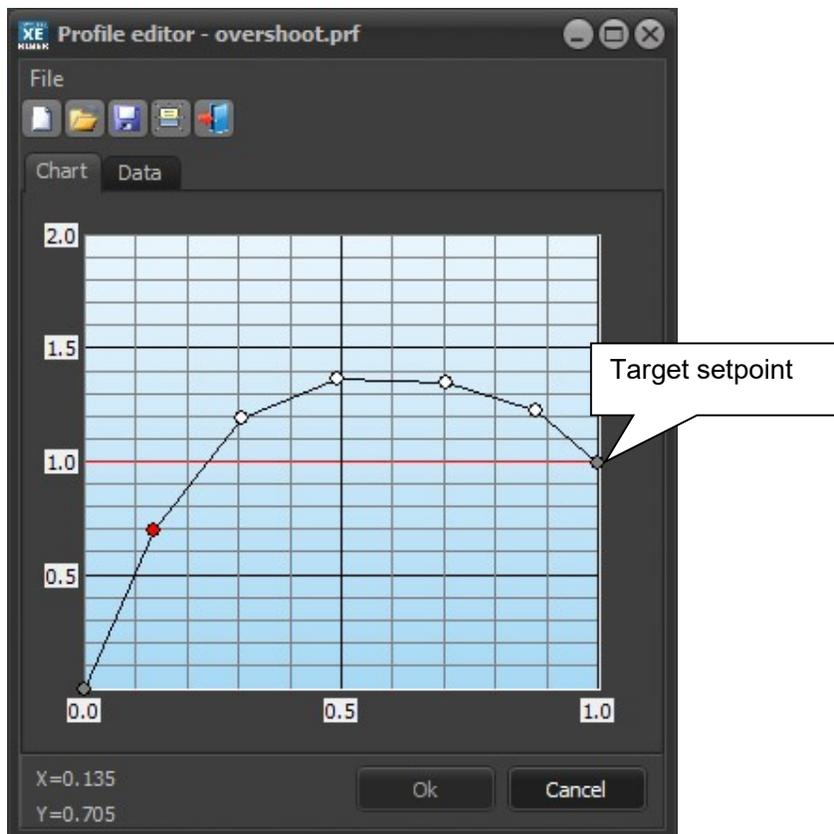
first point: (0.0, 0.0) (measured value)

last point: (1.0, 1.0) (target set point)

Another point is plotted by default in the middle of the curve. At least the curve must have one more point besides the first and the last points.

The ramp rate (slope) will determine the increase or decrease in value over a defined period of time. ($^{\circ}\text{C}/\text{min}$, $\%/ \text{min}$). The ramp rate is to be defined when setting the ramp (equipment *Detail view* window).

→ For more details about the control of equipment, please refer to the chapter Chamber > section **Equipment view** in this manual.



The first point on the chart (0.0, 0.0) corresponds to the value from which the coordinates are measured. In *Crystal XE*, it corresponds usually to the current measured value (MV). The last point corresponds to the value target set point value (TSP).

White points correspond to the points that have been added to the curve. The red point is the point being selected.

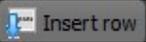
To plot a point on the chart, double-click on the desired coordinate or right-click on it and select *Add a new point*. The point coordinate appears in the left-hand corner of the window:

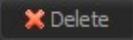
X=0.478
Y=0.669

You can modify a point by dragging it with the mouse. To delete a point, right-click on it and select *Delete* this point.

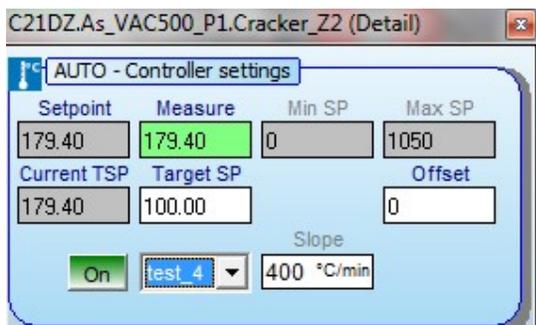
On the *Data* tab, all coordinates are displayed in a table:

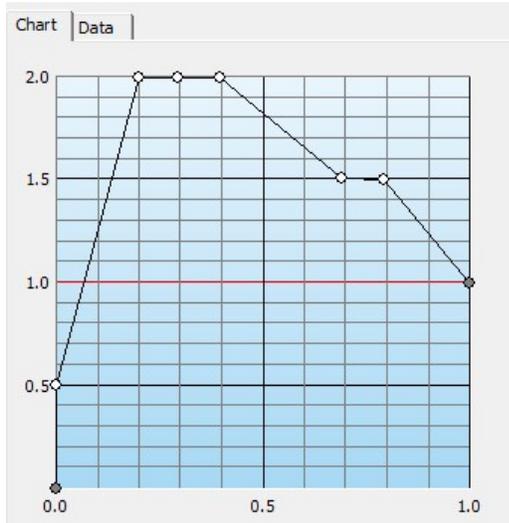
#	X	Y
1	0.000	0.000
2	0.099	0.114
3	0.296	0.355
4	0.478	0.669
5	0.646	0.711
6	0.814	0.855
7	1.000	1.000

To add a new point, click one of the cells, and click on *Insert row* . Then enter the point's ordinate and abscissa.

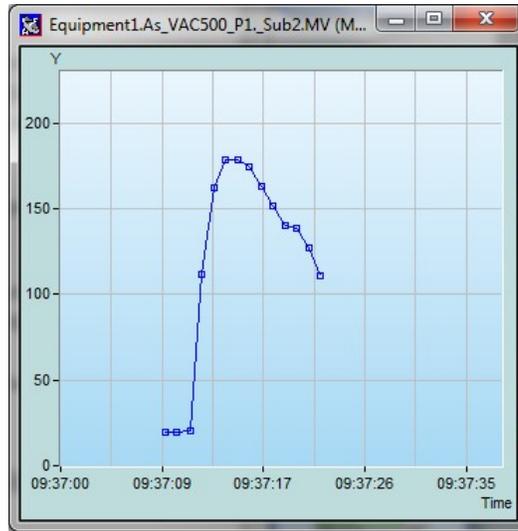
To delete a point, click on the *Delete* button .

Example:





Profile example (test_4)



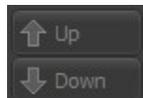
Cracker temperature (MV) following in chart

When created, the new profile is automatically added to the list. The profile is saved as a '.prf' file.

To add a profile file to the list, click on the Add button. 

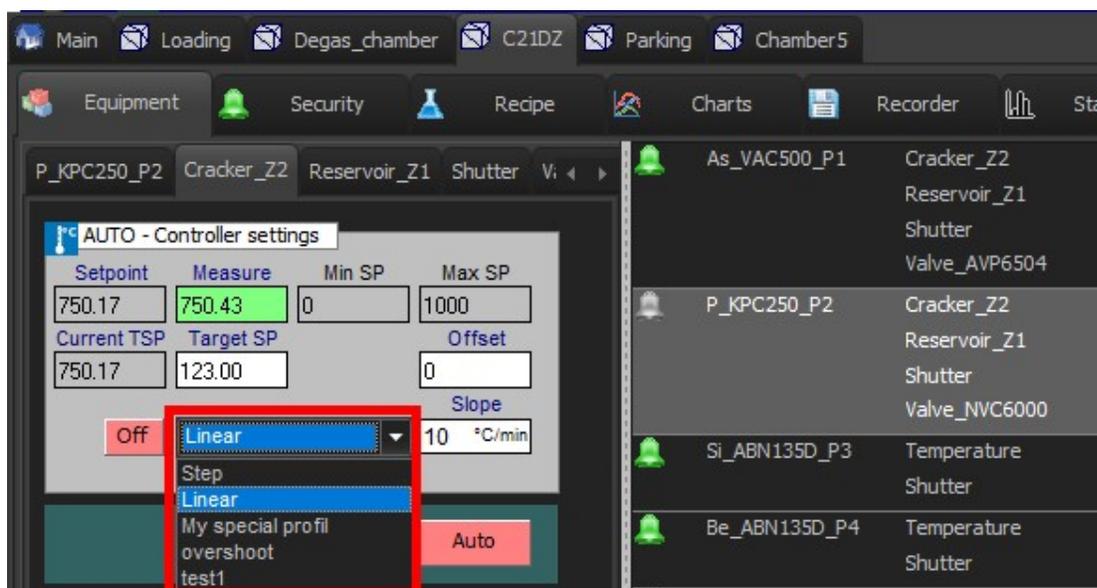
To edit a profile file select it in the list and click on Edit  or double-click on it.

USE THE UP AND DOWN BUTTONS TO MODIFY THE LIST ORDER:



Clicking the Remove button  only removes the file from the list but do not delete the profile file.

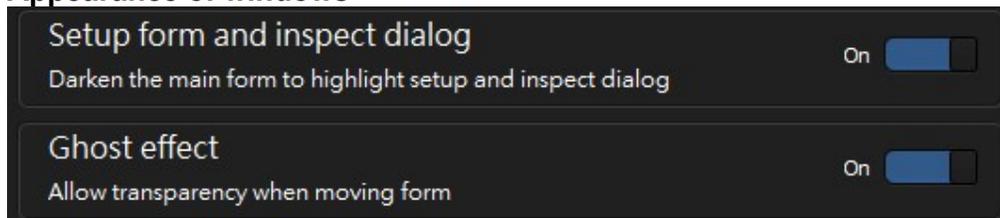
Then, in a temperature controller equipment view, the profiles added in the list appear in the drop-down list when selecting the ramp:



The profile will appear the in the same order in the drop-down list.

10.2. Appearance

Appearance of windows



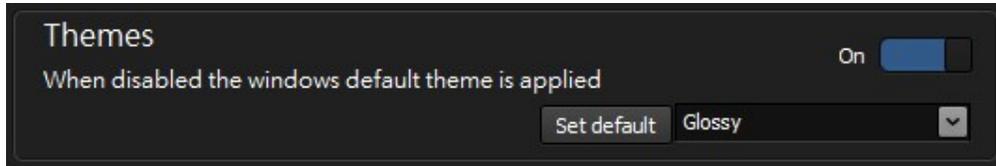
Darken the main form when opening setup and inspect dialog:

A “modal” form is a form (window) that, when open, prevent you to access the main interface until it is closed. By default when such a window is opened, the main interface is darkened.



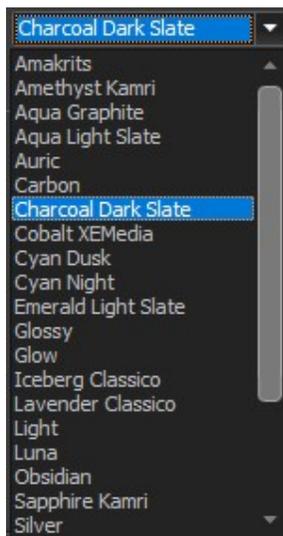
When one of the **Setup** menu modal forms is opened, the main interface is not darkened.

Ghost effect: Toggle the switch to enable/disable Crystal XE windows to become transparent when being moved:

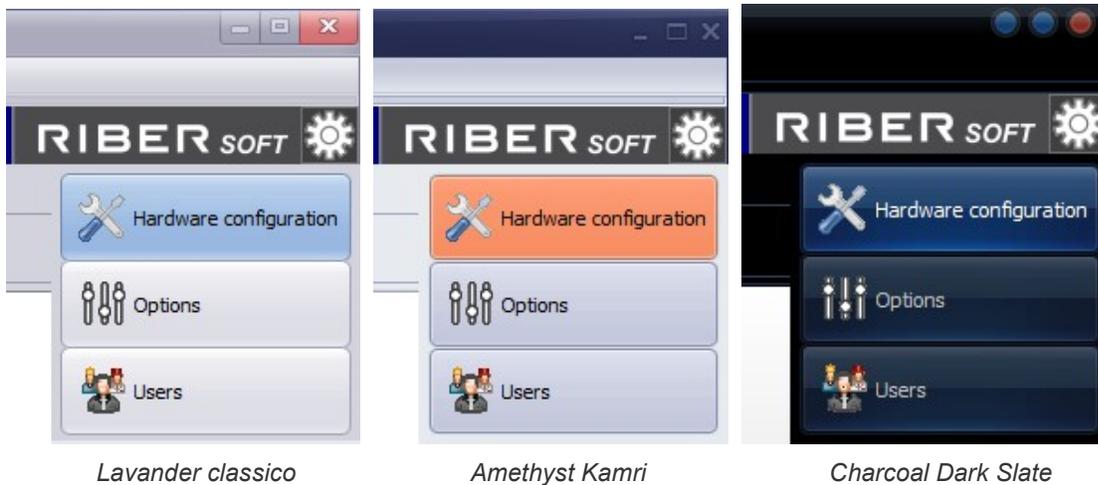


Turn on the switch to enable themes.

Drop the list down to select one of the themes:

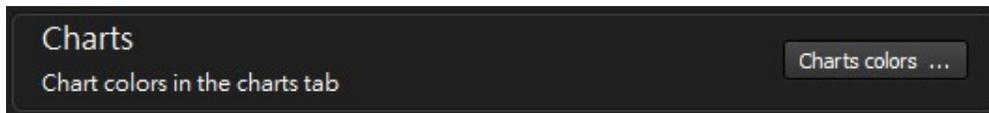


Examples of theme:



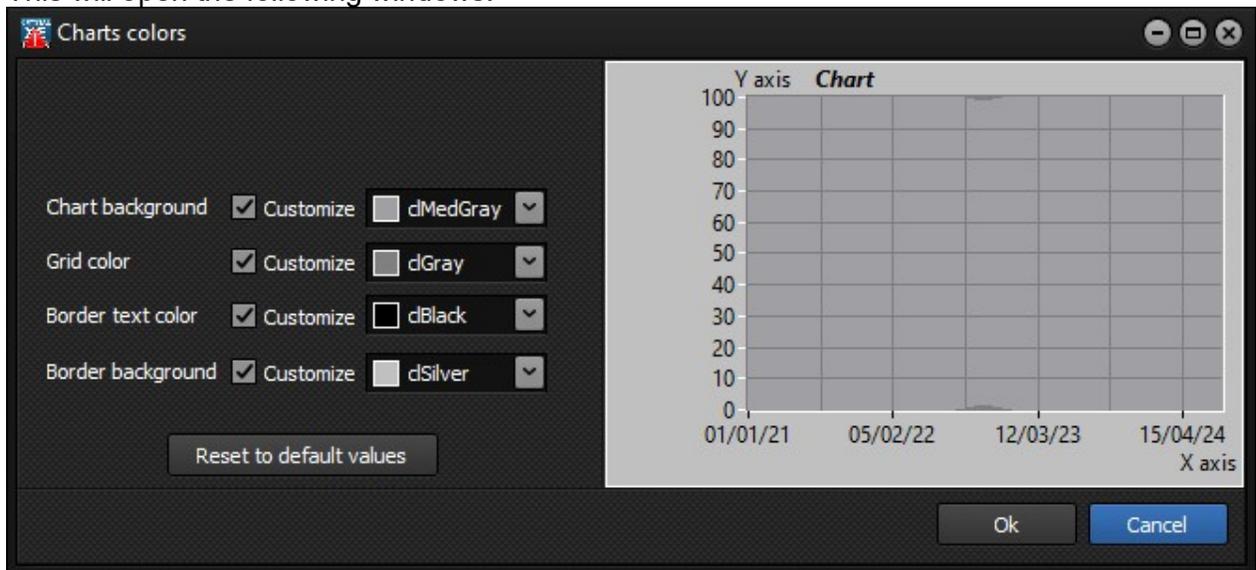
Please note that you'll need to restart Crystal XE for the new theme to take full effect.

Charts

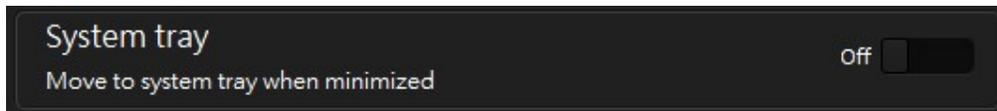


Click on the Charts colors to edit the default colors used in the charts.

This will open the following windows:



System tray



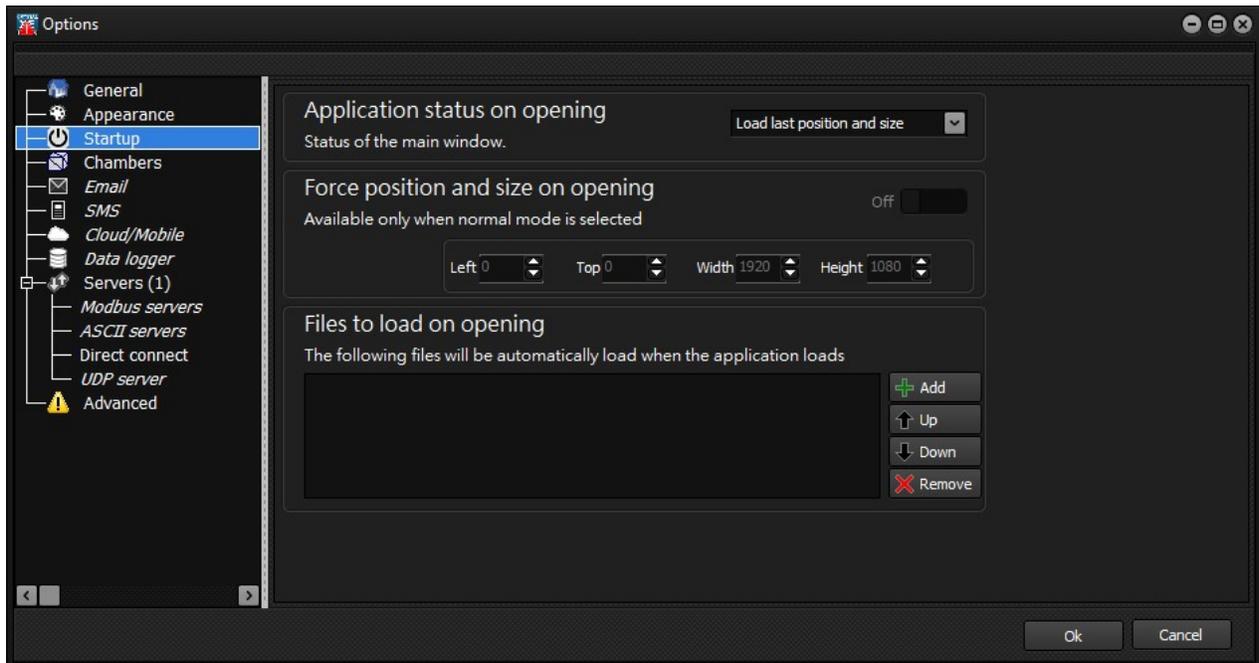
Toggle this switch to display Crystal XE icon in the Windows notification area (system tray) when the program window is minimized.

To restore Crystal XE, double-click on the system tray icon (located on the bottom right of the task bar) to restore the window:

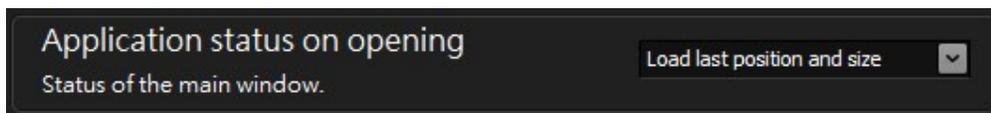


10.3. Startup

On the *Startup* tab, you can modify the default settings at the launch of the software.



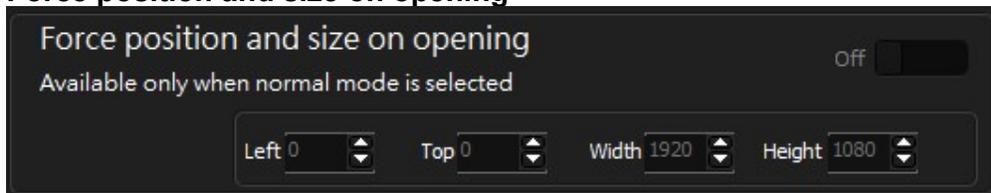
Application status on opening



Select the default state of the window at the launch of the software between this choices:

- Load last position and size (position and size at Crystal XE's last closing)
- Minimize (iconize)
- Maximize (full screen)
- Normal: this choice allows to enter the position and size in the next box.

Force position and size on opening

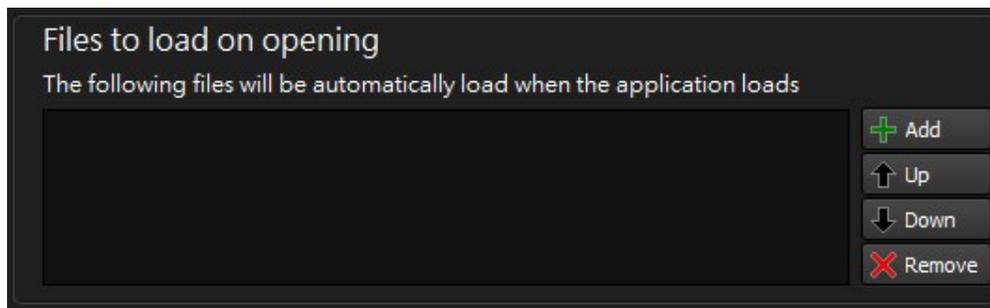


Only available when the status "Normal" is selected

To force the window to open in specific size and position, check the box and adjust the spinners.

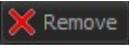
Files to load on opening

Specify forms to be opened automatically at the launch of the software.



Click on the *Add* button  to select the desired form files.

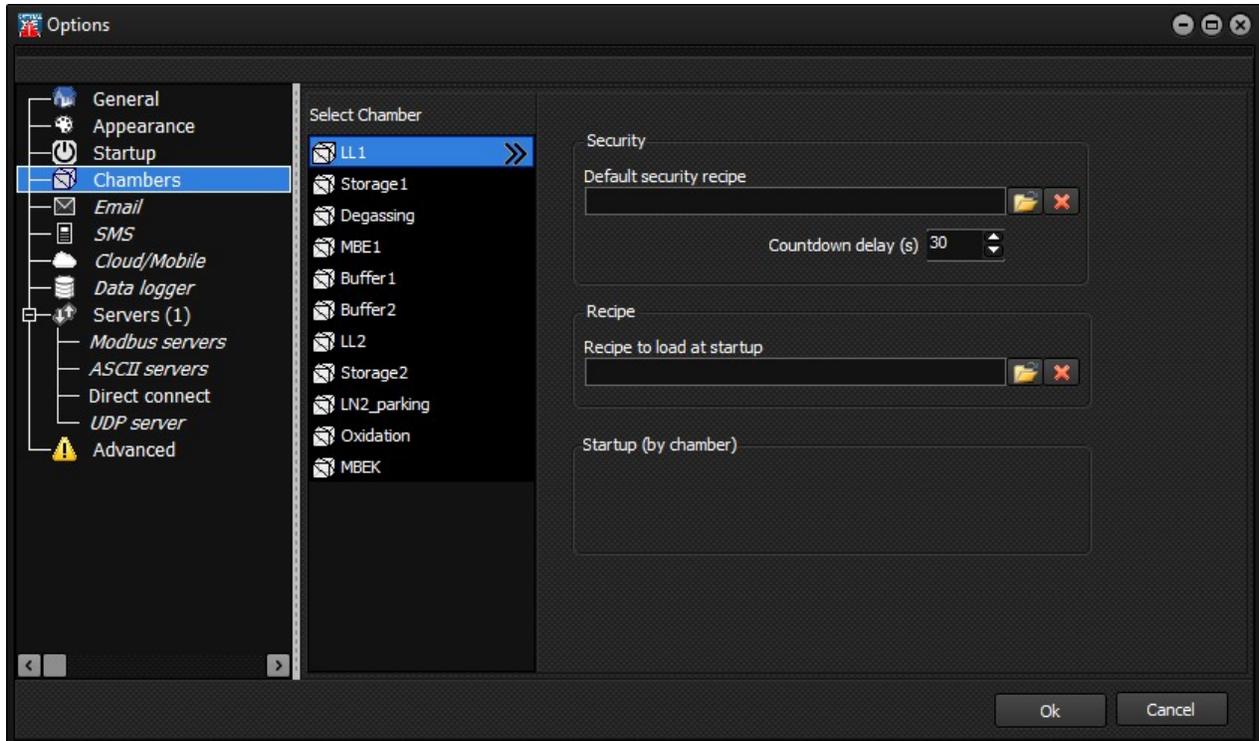


The *Remove* button  only removes the selected file from the list but do not delete the form file.

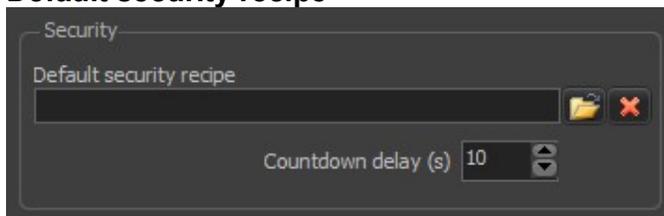
10.4. Chambers

For each chamber, you can specify a default security recipe and countdown and allow the charts and recorders to start automatically at the launch of the software.

Select a chamber in the central list



Default security recipe

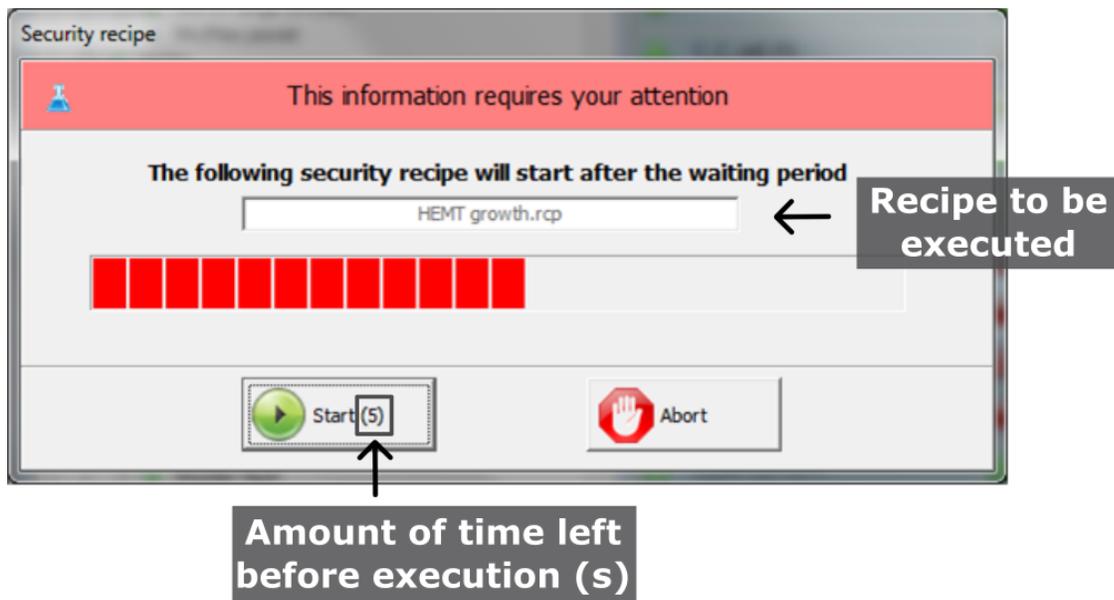


For each security agent, you can set a security recipe to be executed automatically when the security agent's alarm occurs (*Security* tab). You can set either default security recipe or specify one.

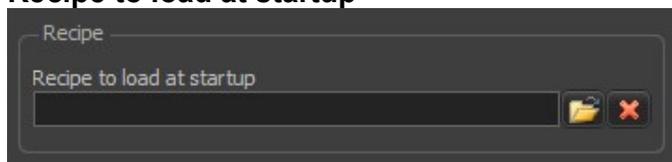
You can also define a time period during which a window will open with a progress bar and allow the user not to launch the recipe.

Countdown: determine the period of time after which the security recipe starts executing. During the countdown, a bar graph appears and you can stop the execution of the recipe or execute it immediately.



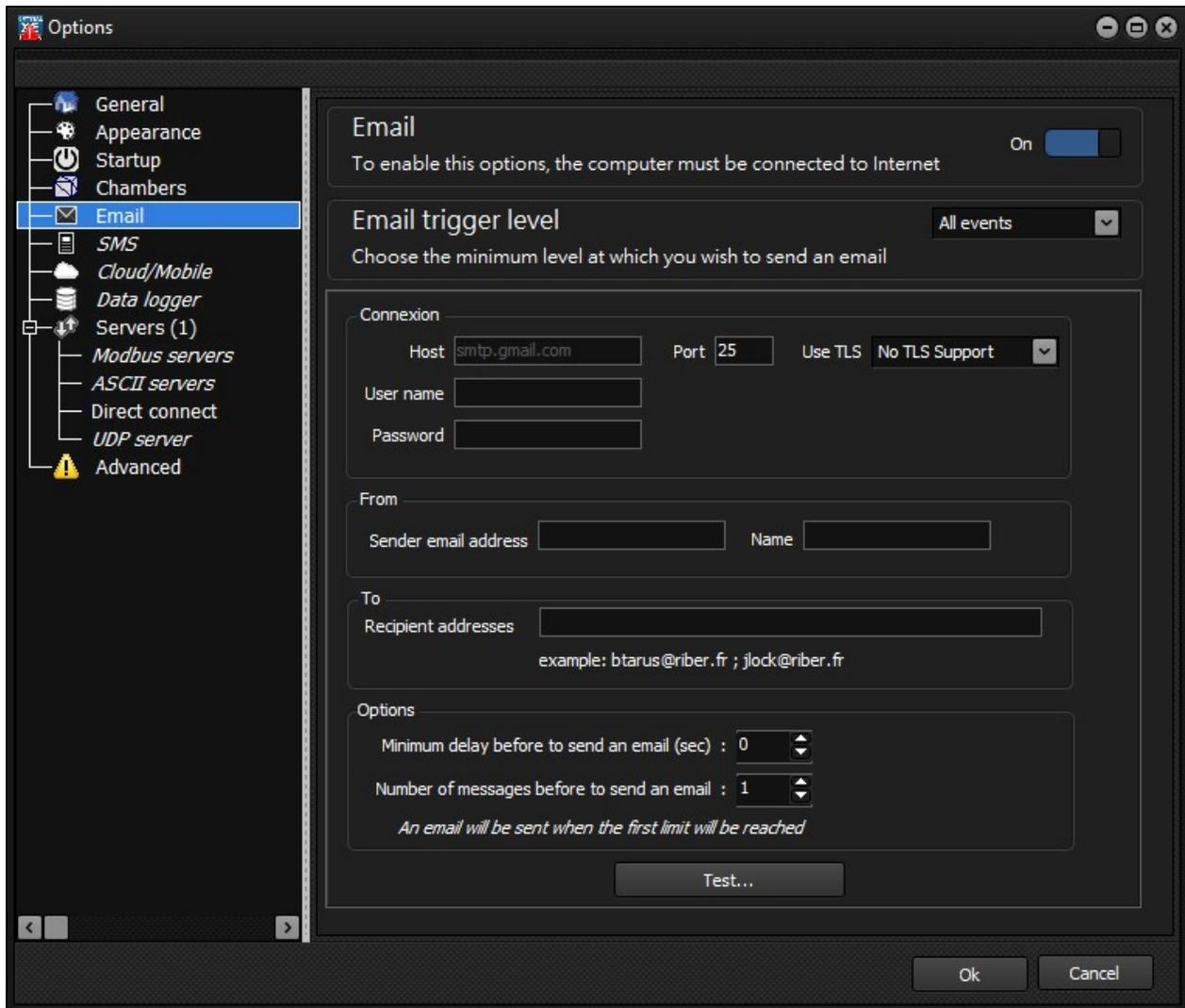


Recipe to load at startup



Specify the default security recipe to be executed in the selected chamber by clicking on the file icon  and selecting the desired recipe file:

10.5. Email



In order to send an e-mail, the computer must be connected to a local area network with an accessible e-mail server or an Internet connection if you are using an external e-mail server (like gmail).

Turn on the switch and you must fill in all the fields to use the email: host, Port, Use TLS, User name etc...

- ⇒ Refer to the application note “AN008_XtalXE_Use Gmail to send email” to send email using a Gmail account.

Recipient addresses

Enter the destination addresses in the text box:

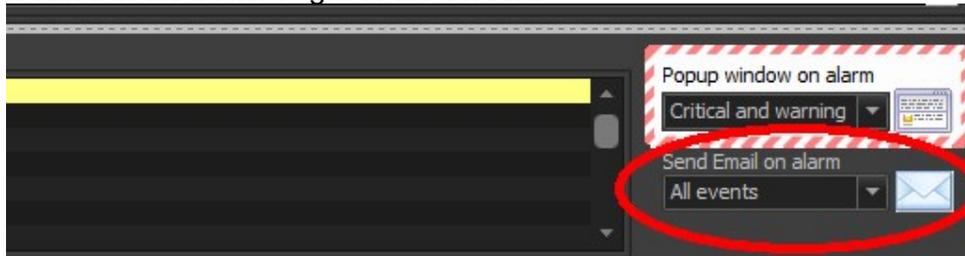


You can define several addresses separated by semi-colons:

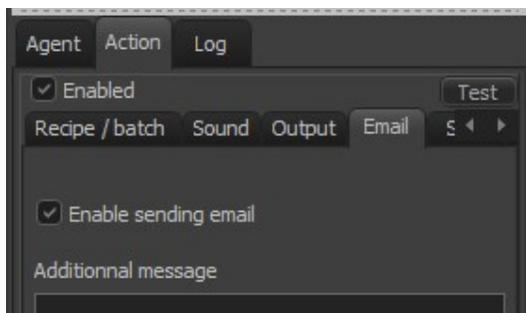


Example of several addresses.

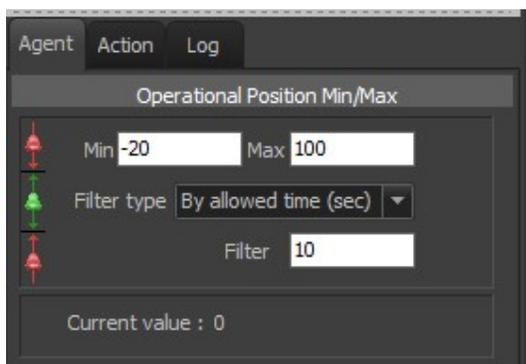
To enable sending of email when a general event occurs, select the appropriate item in the list located in the bottom right of the screen:



 To receive a notification when a specific alarm occurs, you must enable each type of notification in the security settings (*Security tab*) for each specific security agent (in each chamber):

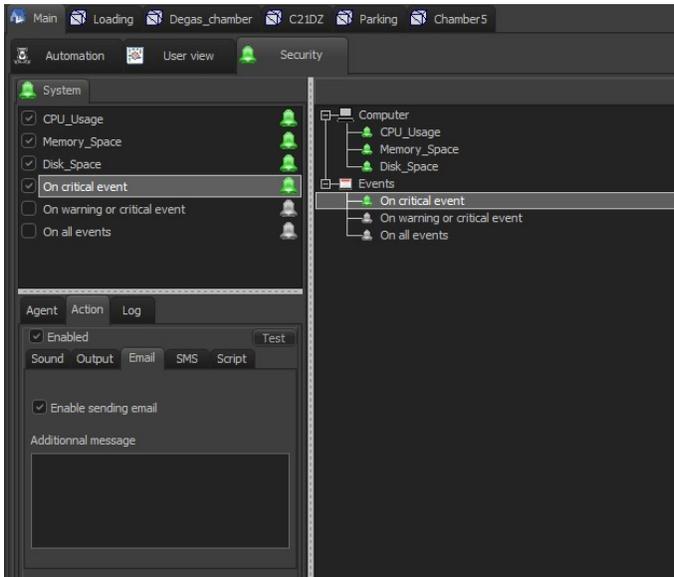


 *Email* and/or *SMS* notifications are sent according to the defined threshold values (*Security tab*):

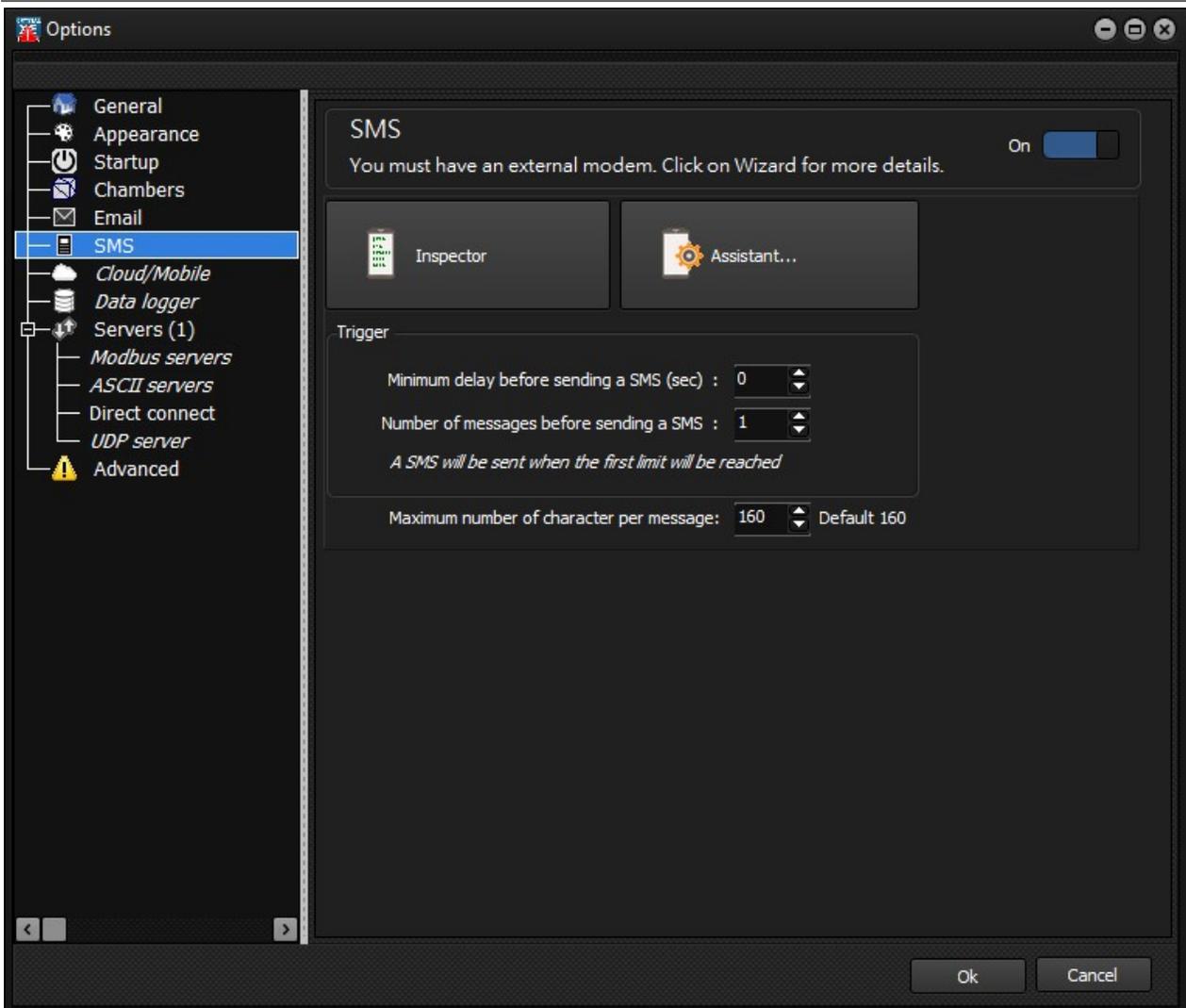


→ For more detail about security agents and alarms configuration, please refer to the chapter *Chamber > Chamber security* in this manual.

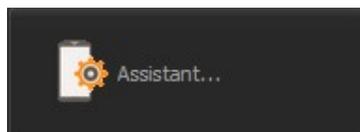
You can also enable the sending of an email when any critical event occurred. Go in the *Main Tab* and the *Security tab*, check the choice “On critical event”, enable the action box and also the box to enable sending email.



10.6. SMS

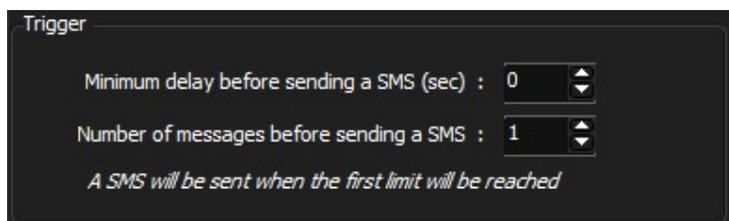


For *Crystal XE* to send SMS, a specific modem, provided by *RIBER*, must be connected. To connect the modem, click on *Assistant* and follow the installation instructions:



Trigger

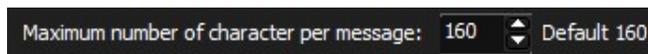
You can set the options of SMS notifications in the following pane:



SMS options

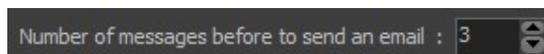
By default if notifications have been enabled for a security agent, an email or SMS notification will be sent each time the security agent alarm occurs.

You can set a maximum number of character per message:



For example, setting 120 seconds minimum delay prevents the system from sending you more than one SMS every 2 minutes. If you receive a SMS, it will report all alarms that occurred within the specified delay.

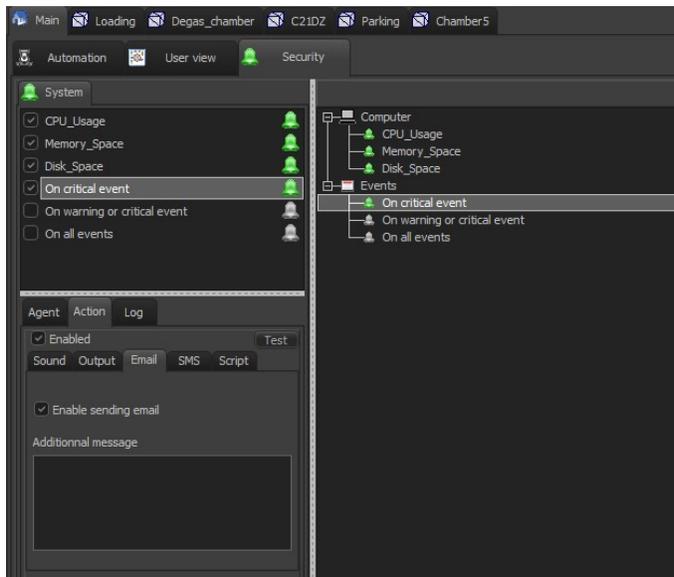
You can set a number of warning messages after which a SMS notification will be sent:



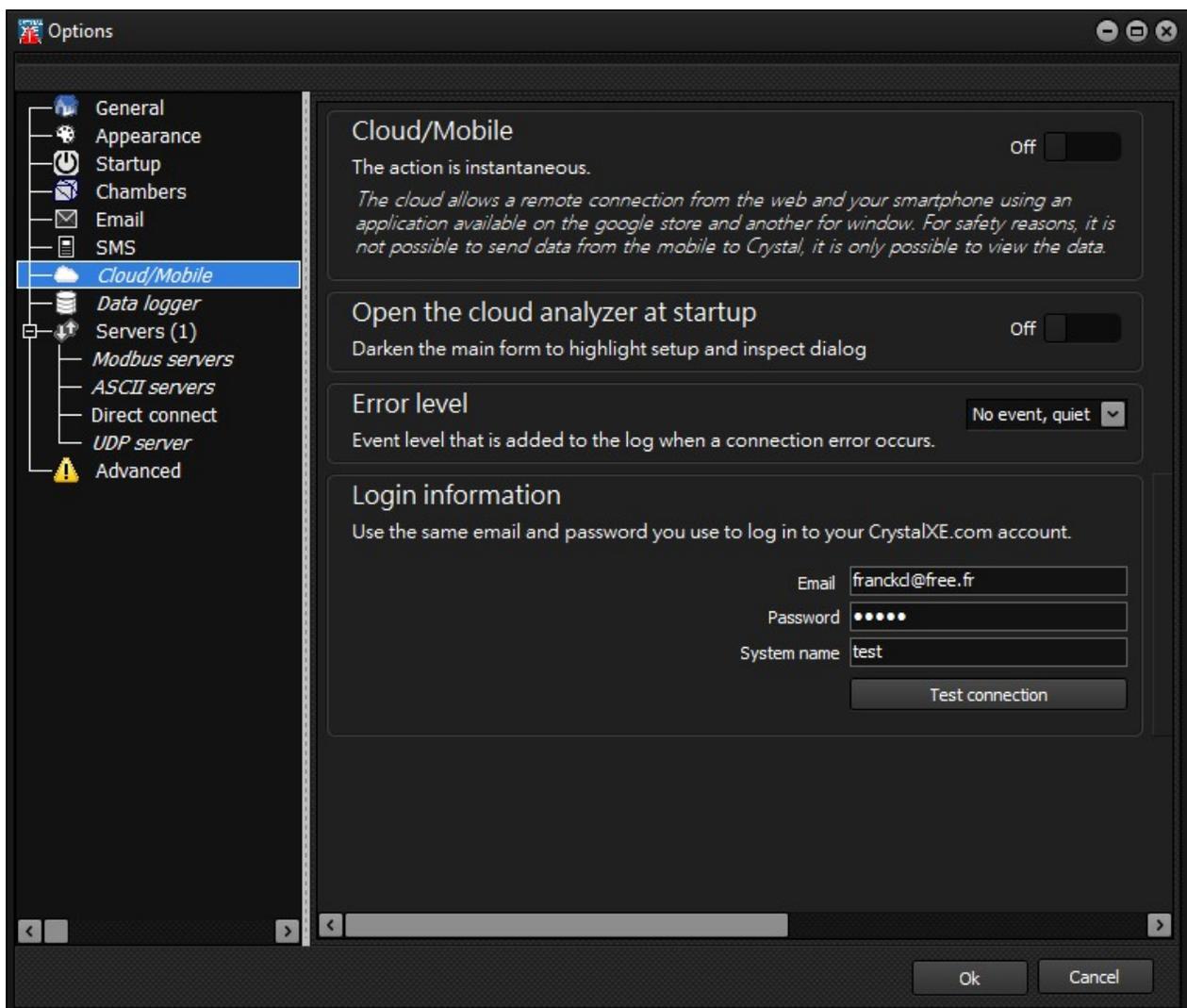
For example, by setting 3, a SMS notification will be sent after that the security agent alarm has occurred 3 times. That SMS will report all the 3 alarms.

If one of these thresholds is reached, a SMS notification will be sent. In the above examples, when an alarm occurs, a SMS notification will be sent either after two minutes or if two more alarms have occurred.

You can also enable the sending of a SMS when any critical event occurred. Go in the Main Tab and the Security tab, check the choice “On critical event”, enable the action box and also the box to enable sending SMS.

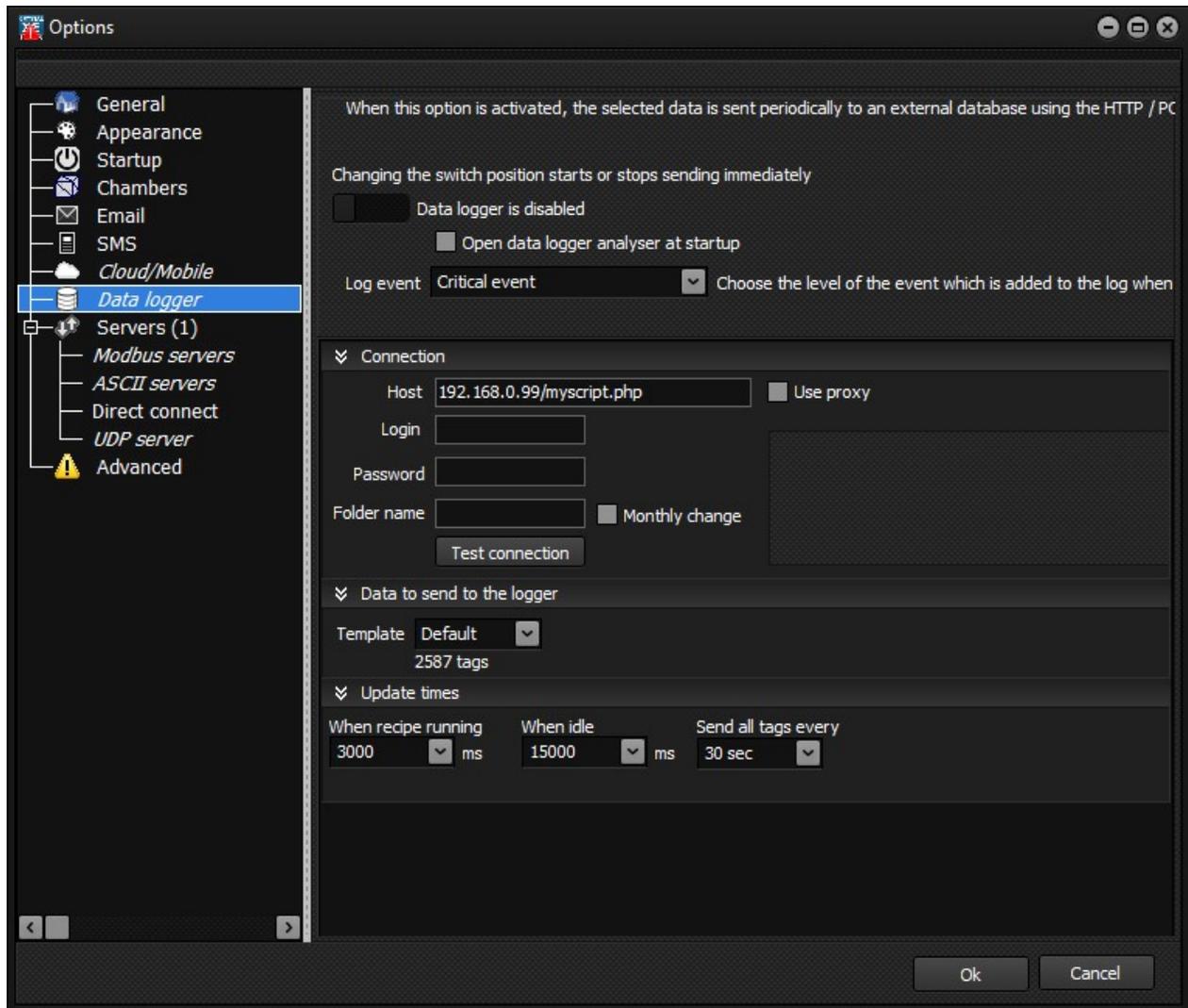


10.7. Cloud/Mobile



There is a special manual for the cloud and the mobile app. Please refer to this manual for more details.

10.8. Data logger options



For more information, please refer to the “Data logger” section later in this document.

10.9. Servers

Crystal XE allows you to create Modbus or ASCII servers to exchange data with other software or devices (computers or PLCs).



Modbus protocol abilities are needed to setup the server or modify any of these settings.

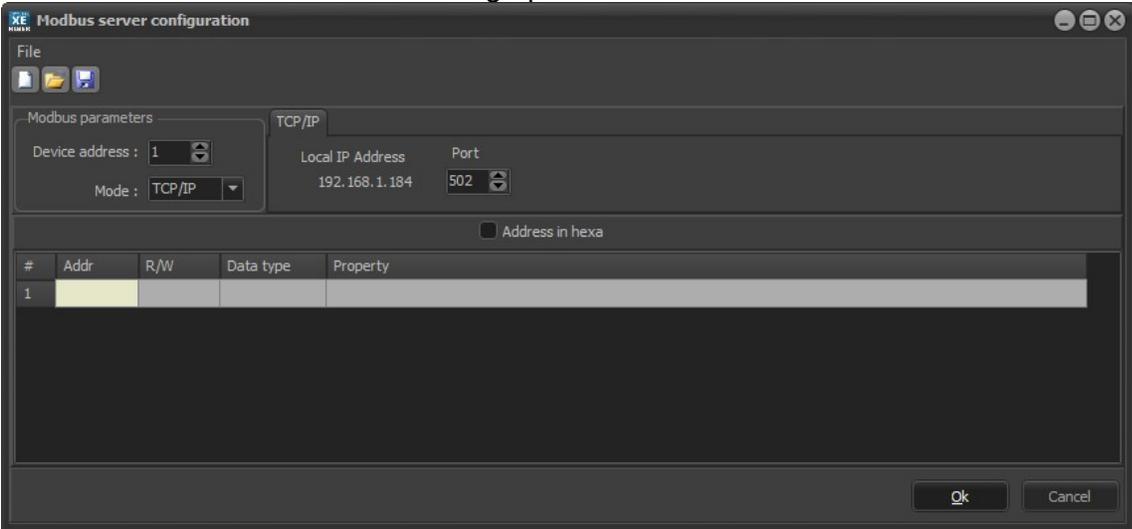
10.9.1. Modbus servers

You can define as many Modbus servers as you wish. Each server must be configured to operate either in TCP / IP or in serial link.

Each server is located on a line in the table below. Click on a line to bring up the server control buttons.

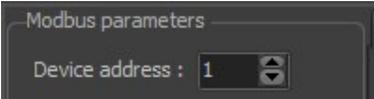


Select the Setting button  to change parameters

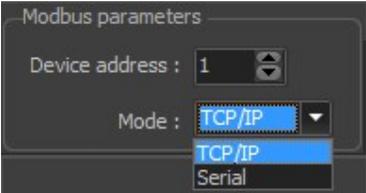


Modbus settings

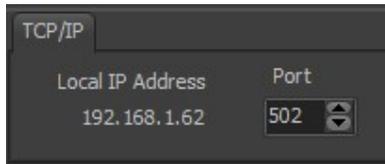
Specify *Crystal XE* device address:



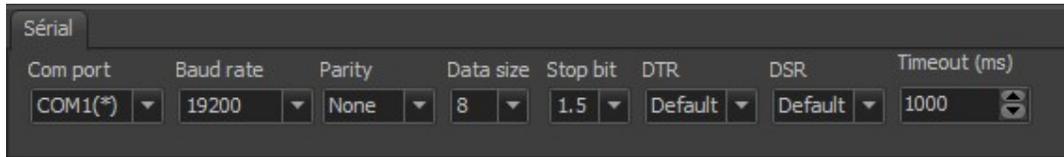
You can set either TCP/IP or serial communication mode:



Set the corresponding settings:



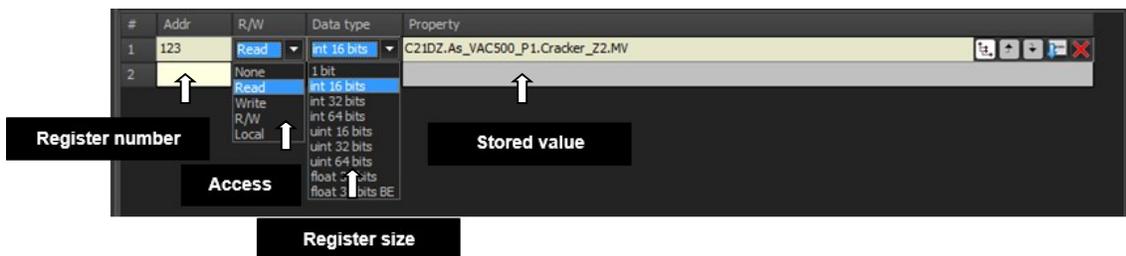
TCP/IP settings



Serial communication settings

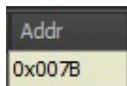
Register map

The data accessible by other software or devices is stored in a register table



Assign each register an arbitrary address between 0 and 65535 (register number). Each data is identified by its address. Each address is allocated an area of memory whose size is defined according to the selected data type (1bit, 16 bits, 32bits, etc.).

Check the following box to show the address in hexadecimal: Address in hexadecimal



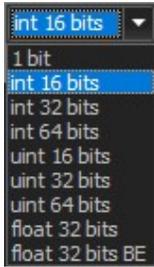
Specify the data access:



Local type is not used here, it is only for comparison reason with the modbus client.

Select the data type:

- 1 bit (0/1)
- Integer 16, 32 or 64 bits (int)
- Unsigned integer 16, 32 or 64 bits (unit)
- Float 32 bits or 32 big BE (Big endian)



The property column corresponds to the data to be read and/or written.

Click on the cell to select a property in the *Data explorer*.

In this example, the client (software or device) is allowed to read the measured value of the *As_VAC500_P1* equipment's cracker (C21DZ chamber):

#	Addr	R/W	Data type	Property
1	0x007B	Read	int 16 bits	C21DZ.As_VAC500_P1.Cracker_Z2.MV
2				

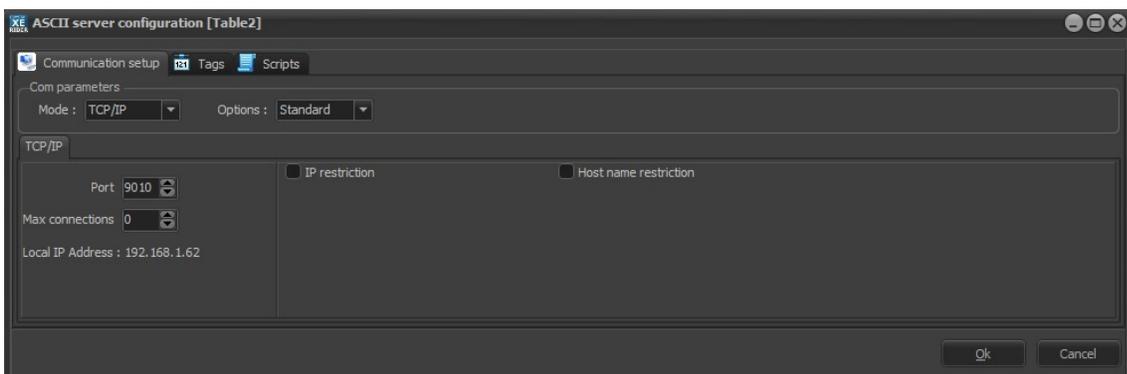
10.9.2. ASCII Servers

You can define as many ASCII servers as you wish. Each server must be configured to operate either in TCP / IP or in serial link.

Each server is located on a line in the table below. Click on a line to bring up the server control buttons.

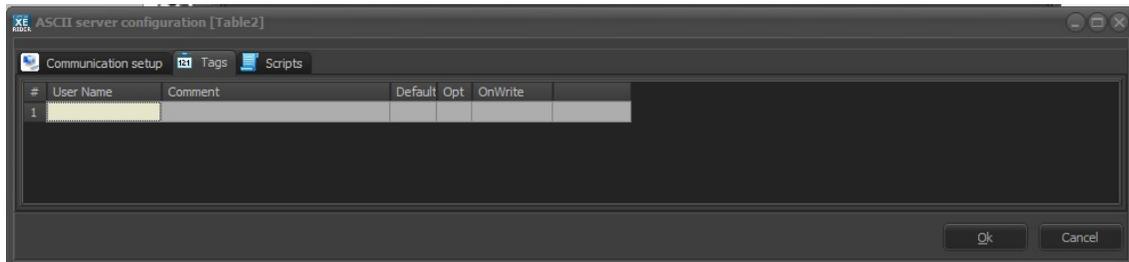


Select the Setting button  :
Communication setup



ASCII server settings

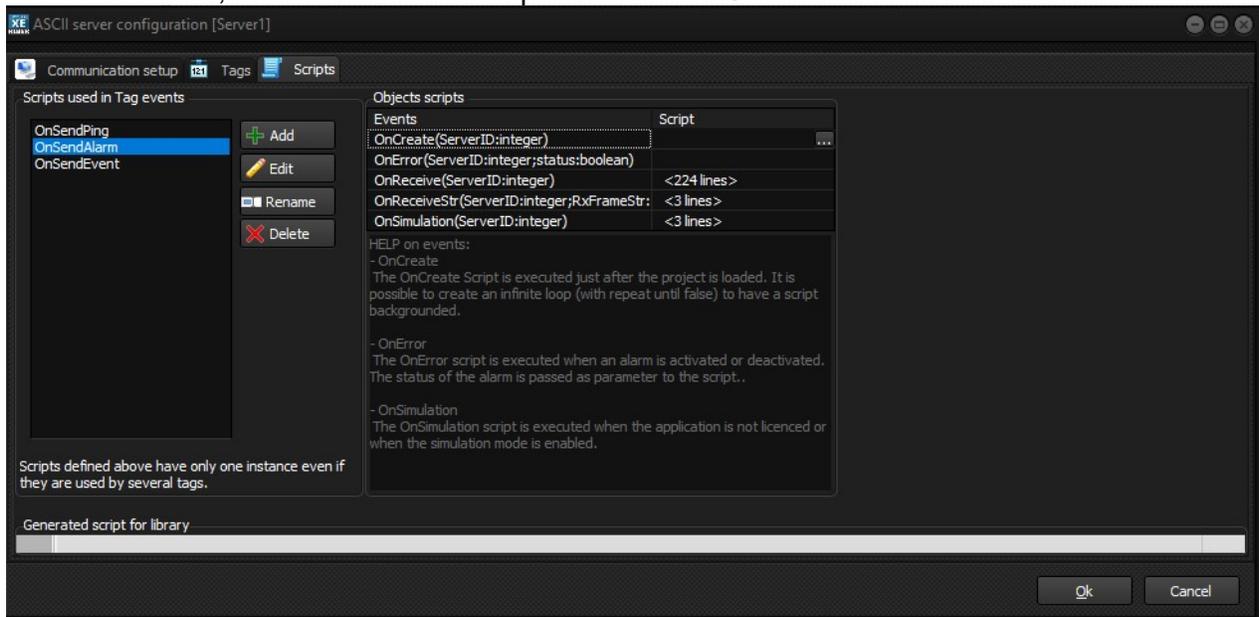
Tags



For each tag, give a name, a comment (optional) the default value (optional), Com options and which script to execute when the tag is changed.

Script

In the list bellow, enumerate all the scripts used in the OnWrite column of the server.

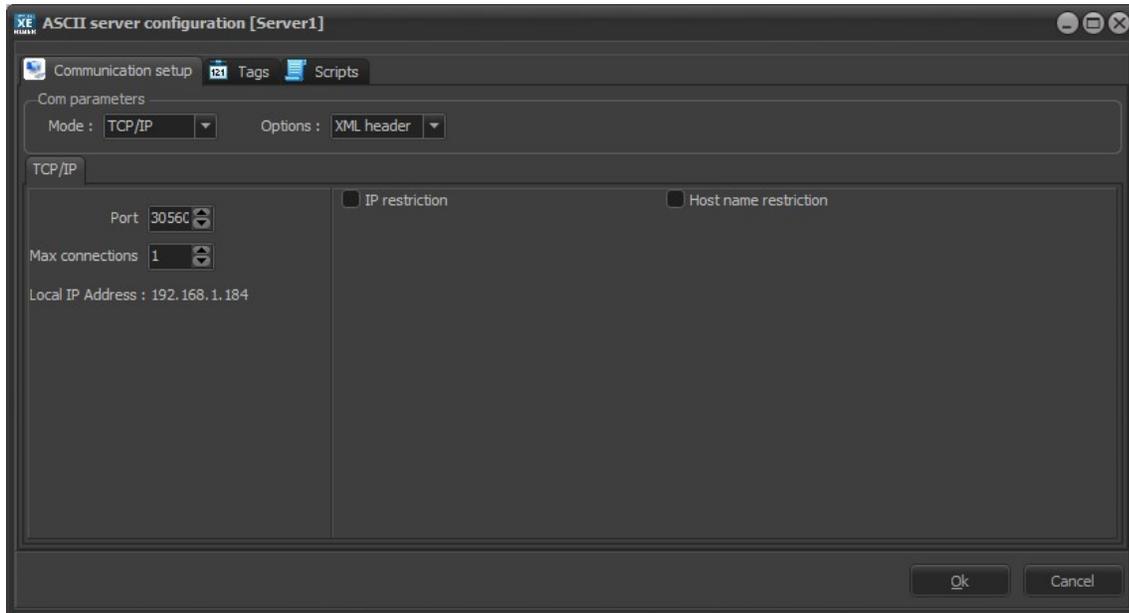


Others scripts:

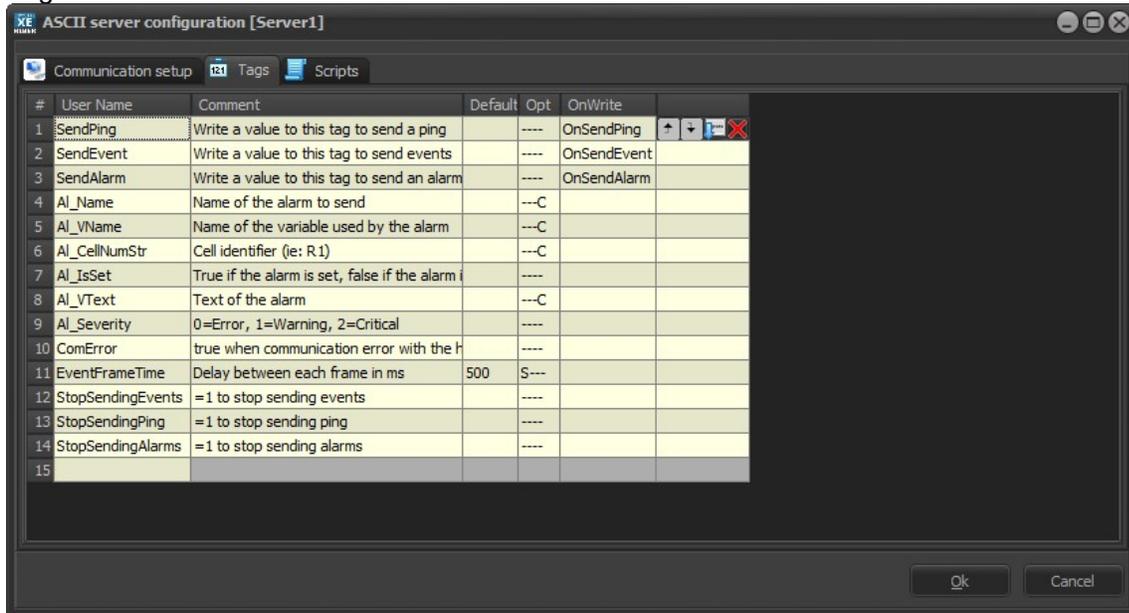
- **OnCreate:** This script is executed only when loading the project. It is possible to write a background script by using an infinite loop like “repeat (...) until false”. In that case, it is recommended to add the instruction “sleep(300)” in the loop.
- **OnError:** this script is executed after a communication error or after the error has been recovered.
- **OnReceive:** This script is executed after receiving a message from the host, and is preferred for XML transactions because it allows long messages to be received with function XML_ReceiveBuf.
- **OnReceiveStr:** As for the OnReceiveStr event, but the string received is passed in the RxFrameStr parameter. However, the string received is limited to 242 characters.
- **OnSimulation:** This script is only executed when Crystal is running in simulation mode.

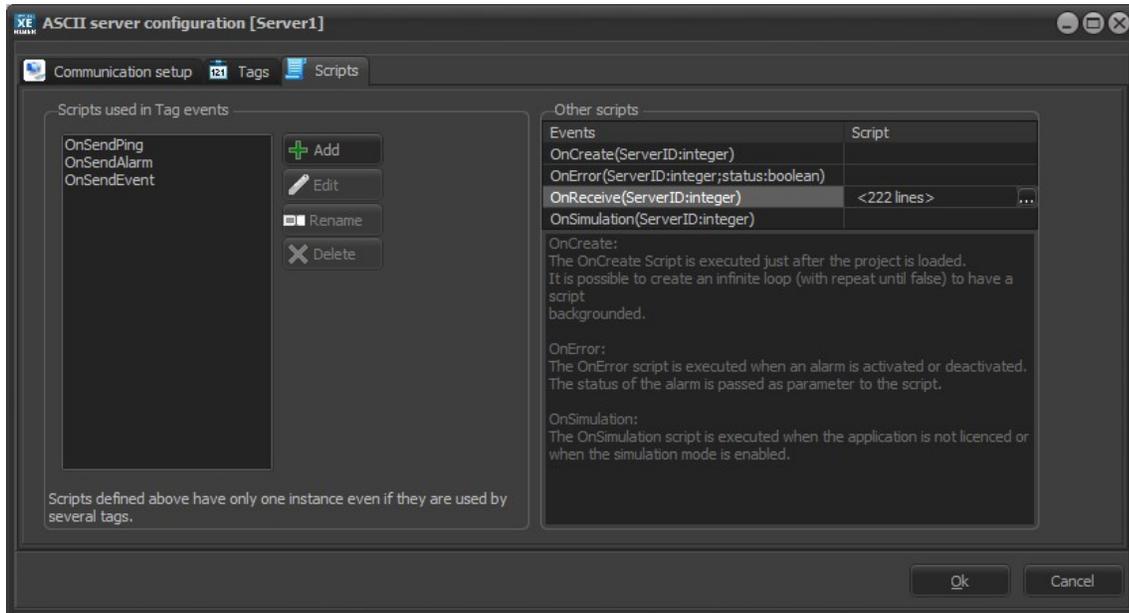
Example of TCP/IP - XML Server using the ASCII server:

Communication settings:

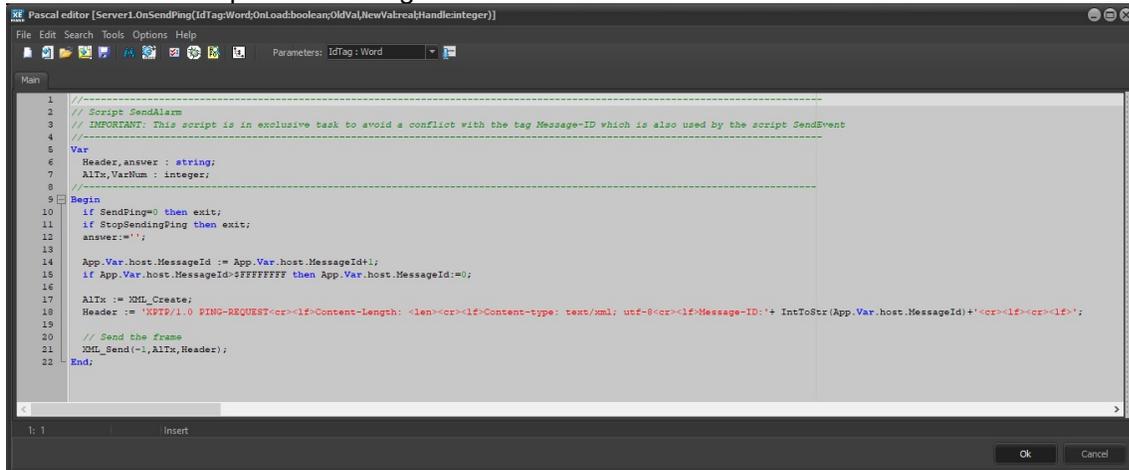


Tags:





Content of the script OnSendPing



Content of the script OnSendAlarm

```

Pascal editor [Server1.OnSendAlarm(IdTag:Word;OnLoad:boolean;OldVal/NewVal:real;Handle:integer)]
File Edit Search Tools Options Help
Parameters: IdTag: Word
Main
1 //-----
2 // Script SendAlarm
3 // IMPORTANT: This script is in exclusive task to avoid a conflict with the tag Message-ID which is also used by the script SendEvent
4 //-----
5 Var
6 S_AlarmName,Header,Severity,IsSet : string;
7 AlTx,VarNum : integer;
8 //-----
9 Procedure AddVarToAlarm(VarName,SValue:string);
10 Begin
11 XML_Write(AlTx,'Alarm/+V', 'name', VarName);
12 XML_Write(AlTx,'Alarm/#'+IntToStr(VarNum), '', SValue);
13 VarNum := VarNum+1;
14 End;
15 //-----
16 Begin
17 if StopSendingalarms or (SendAlarm=0) then EXIT;
18 // Increment the messageID
19 App.Var.host.MessageId := App.Var.host.MessageId+1;
20 if App.Var.host.MessageId>:FFFFFFF then App.Var.host.MessageId:=0;
21 AlTx := XML_Create;
22 if Al_Severity=0 then Severity := 'error';
23 if Al_Severity=1 then Severity := 'warning';
24 if Al_Severity=2 then Severity := 'critical';
25 if Al_IsSet then IsSet := 'true'; else IsSet:='false';
26 Header := 'MTP/1.0 DATA-NOTIFY<cr><lf>Content-Length: <len><cr><lf>Content-type: text/xml; utf-8<cr><lf>Message-ID: '+ IntToStr(App.Var.host.MessageId)+'<cr><lf><lf>';
27 XML_Write(AlTx,'Alarm', 'name', Al_Name);
28 XML_Write(AlTx,'Alarm', 'time', FormatDateTime('YYYY-MM-DD"THH:MM:SS', GetNow));
29 XML_Write(AlTx,'Alarm', 'severity', Severity);
30 XML_Write(AlTx,'Alarm', 'isSet', IsSet);
31 VarNum := 0;
32 if length(Al_VName)>0 then AddVarToAlarm(Al_VName, Al_Vcent);
33 if length(Al_CellNumStr)>0 then AddVarToAlarm('Cell', Al_CellNumStr);
34 // Send the frame
35 XML_Send(-1, AlTx, Header); // There is no answer to a NOTIFY message
36 SendAlarm:=0;
37 End;
38
39
40
41
42
43
44

```

Content of script onSendEvent

```

uses stdConst, common, events, alarmsS3PC;
//-----
// Script SendEvent
// IMPORTANT: This script is in exclusive task to avoid a conflict with the tag Message-ID which is also used by the
script SendAlarm
//-----
//-----Var
i : integer;
//-----Function
FrameDelay:boolean;
Begin
if not(ComError) and (EventFrameTime>0) then sleep(EventFrameTime);
result := not(ComError);
end;
//-----Procedure
AddVarToEvent(VarName, SValue:string);
Begin
XML_Write(EvTx, 'Event/+V', 'name', VarName);
XML_Write(EvTx, 'Event/#'+IntToStr(VarNum), '', SValue);
VarNum := VarNum+1;
End;
//-----Procedure
SendRecipeStatus;
Begin
CreateEvent('RecipeStatus');
AddVarToEvent('RecipeRunning', GetValueProp('App.Var.Recipe.Running'));
XML_Send(-1, EvTx, EventHeader); // Send the frame now, there is no answer to a NOTIFY message
End;
//-----//
GetCellStatusStr
// Called by SendCellEvents
//-----Function
GetCellStatusStr(CellNum:integer):string;
Var
Status : String;
n : integer;
Begin
n := App.Var.Recipe.CellStatus[CellNum];
Status:='Unknown';
if n=1 then Status:='Stopped';
if n=2 then Status:='Ramping to standby';
if n=3 then Status:='Standby';

```

```

    if n=4 then Status:='Heating to process';
    if n=5 then Status:='Processing';
    if n=6 then Status:='Switching Quartz';
    if n=7 then Status:='Alarm';
    result := status;
end;
//-----Function
GetCellRecipeSelect(CellNum:integer):string;
Var
    n : integer;
    Status : String;
Begin
    n := App.Var.Recipe.RecipeSelect[CellNum];
    Status:='none';
    if n=1 then status:='Standby';
    if n=2 then status:='Process';
    if n=3 then status:='Cool Down';
    result := status;
end;
//-----Procedure
SendCellEvents(CellNum:integer);
Var
    S,Racine : String;
    n : integer;
Begin
    S := GetCellIdent(CellNum); // example: S = 'R1'
    CreateEvent('StatusCell'+S);
    Racine := 'System.Row'+S;
    AddVarToEvent('CellRecipeStatus',      GetCellStatusStr(CellNum));
    AddVarToEvent('CellRecipeSelect',      GetCellRecipeSelect(CellNum));
    AddVarToEvent('RecipeID',              GetValueProp('App.Var.Recipe.RecipeID'));
    AddVarToEvent('RecipeName',            App.Forms.MainUserView.LabelRecipeName.Caption);
    AddVarToEvent('DMC',                    App.Var.AMAT.DMC);
    AddVarToEvent('WebMeter',               FormatFloat('0.#####',App.Var.AMAT.Rollmeter));
    AddVarToEvent('CellTempTop',           GetValueProp(Racine+'.tip_temp.MV_Temp'));
    AddVarToEvent('CellTempBot',           GetValueProp(Racine+'.base_temp.MV_Temp'));
    AddVarToEvent('CellPowTop',            GetValueProp(Racine+'.tip_temp.MV_Pow'));
    AddVarToEvent('CellPowBot',            GetValueProp(Racine+'.base_temp.MV_Pow'));
    AddVarToEvent('CellCurrentTop',         GetValueProp(Racine+'.tip_temp.MV_Curr'));
    AddVarToEvent('CellCurrentBot',         GetValueProp(Racine+'.base_temp.MV_Curr'));
    AddVarToEvent('CellRegulTypeTop',       GetValueProp(Racine+'.tip_temp.RegMode'));
    AddVarToEvent('CellRegulTypeBot',       GetValueProp(Racine+'.base_temp.RegMode'));
    AddVarToEvent('CellCurrentSetPointTop', GetValueProp(Racine+'.tip_temp.CSP'));
    AddVarToEvent('CellCurrentSetPointBot', GetValueProp(Racine+'.base_temp.CSP'));
    AddVarToEvent('CellQuartzRate',         GetValueProp(Racine+'.Cygnus.CY_Rate'));
    AddVarToEvent('CellQuartzShutterStatus', GetValueProp(Racine+'.Cygnus.CY_Shutter'));
    AddVarToEvent('CellCurrentQuartzSelected', GetValueProp(Racine+'.Cygnus.CY_Position'));
    AddVarToEvent('CellQuartzRemainingLife', GetValueProp(Racine+'.Cygnus.CY_Remaining_life'));

    XML_Send(-1,EvTx,EventHeader); // Send the frame now, there is no answer to a NOTIFY message
end;
//-----Procedure
SendLightColumn;
Begin
    CreateEvent('StatusLightColumn');
    AddVarToEvent('RedLight',      GetValueProp('App.Var.LightColumn.RedSatus'));
    AddVarToEvent('YellowLight',   GetValueProp('App.Var.LightColumn.YellowStatus'));
    AddVarToEvent('GreenLight',    GetValueProp('App.Var.LightColumn.GreenStatus'));
    AddVarToEvent('BlueLight',     GetValueProp('App.Var.LightColumn.BlueStatus'));
    AddVarToEvent('WhiteLight',    GetValueProp('App.Var.LightColumn.WhiteStatus'));
    XML_Send(-1,EvTx,EventHeader); // Send the frame now, there is no answer to a NOTIFY message
end;
//-----
Begin
    if StopSendingEvents or (SendEvent=0) then EXIT;

    if FrameDelay then SendRecipeStatus;
    for i:=0 to NB_MAX_CELLS-1 do
    Begin
        if FrameDelay then SendCellEvents(i);
    end;
    if FrameDelay then SendLightColumn;
    SendEvent:=0;
End;

```

Content of the OnReceive script:

Uses common, StdConst, AlarmS3PC, alarms, events;

```

//-----//
Server Script
// The host computer will send request to Crystal XE, this script manages these requests.
// Creation RIBER: July 2018
//-----Var
idTx, idRx, NodeNum : integer;
S : string;
//-----//
AddOneVar
// Called by SendVariablesList
// Add a variable to the XML file identified by idTx
//-----Procedure
AddOneVar(SName, SType, STransient, Description:string);
Var Path : String;
Begin
XML_Write(idTx, 'Variable-SD/'+Variable, 'name', SName);
Path := 'Variable-SD/'+IntToStr(NodeNum);
XML_Write(idTx, Path, 'type', SType);
XML_Write(idTx, Path, 'transient', STransient); // Transient ?
if length(Description)>0 then XML_Write(idTx, Path, '', Description); // Transient ?
NodeNum := NodeNum + 1;
end;
//-----//
SendVariablesList
// - Called by the main in respons of the message "Variable-SD-Query"
//-----Procedure
SendVariablesList;
Begin
idTx := XML_Create;
NodeNum := 0;

// used by alarms
AddOneVar('RecipeRunning', 'bool', 'false', 'General Recipe status: running or stopped, true when at least one cell recip
is running');
AddOneVar('RecipeID', 'integer', 'false', 'Current Recipe ID (available only when recipe is running)');
AddOneVar('RecipeName', 'string', 'false', 'Name of the current recipe (available only when recipe is running)');
AddOneVar('WebMeter', 'float', 'false', '= Roll Meter');
AddOneVar('Cell', 'string', 'false', 'Cell identifier used for alarm events');
AddOneVar('CellRecipeStatus', 'string', 'false', 'Status of the recipe');
AddOneVar('CellRecipeSelect', 'string', 'false', 'Recipe mode');
AddOneVar('CellTempTop', 'float', 'false', 'Cell Temperature top in degree C');
AddOneVar('CellTempBot', 'float', 'false', 'Cell Temperature bottom in degree C');
AddOneVar('CellPowTop', 'float', 'false', 'Cell power top in watt');
AddOneVar('CellPowBot', 'float', 'false', 'Cell power bottom in watt');
AddOneVar('CellCurrentTop', 'float', 'false', 'Cell power top in ampere');
AddOneVar('CellCurrentBot', 'float', 'false', 'Cell power bottom in ampere');
AddOneVar('CellRegulTypeTop', 'integer', 'false', 'Cell regulation type top: 0:voltage ; 1:Current ; 2:Power ; 3:Rati
; 4:Temperature ; 5:Rate');
AddOneVar('CellRegulTypeBot', 'integer', 'false', 'Cell regulation type bottom: 0:voltage ; 1:Current ; 2:Power
3:Ratio ; 4:Temperature ; 5:Rate');
AddOneVar('CellCurrentSetPointTop', 'float', 'false', 'Cell Current setpoint top with unit depending of CellRegulTyp
top');
AddOneVar('CellCurrentSetPointBot', 'float', 'false', 'Cell Current setpoint bottom with unit depending c
CellRegulType bottom');
AddOneVar('CellQuartzRate', 'float', 'false', 'Quartz rate in Angstrom per second');
AddOneVar('CellQuartzShutterStatus', 'integer', 'false', '0:closed ; 1: Open');
AddOneVar('CellCurrentQuartzSelected', 'integer', 'false', '0: Unknown ; 1-12 : Quartz Number');
AddOneVar('CellQuartzRemainingLife', 'float', 'false', 'Cell quartz remaining life in percent');
AddOneVar('DMC', 'string', 'false', 'Data matrix code');

AddOneVar('RedLight', 'integer', 'false', 'Status of the red light of the light column, 0:off ; 1:continuous ; 2
flashing');
AddOneVar('YellowLight', 'integer', 'false', 'Status of the yellow light of the light column, 0:off ; 1:continuous ; 2
flashing');
AddOneVar('GreenLight', 'integer', 'false', 'Status of the green light of the light column, 0:off ; 1:continuous ; 2
flashing');
AddOneVar('BlueLight', 'integer', 'false', 'Status of the blue light of the light column, 0:off ; 1:continuous ; 2
flashing');
AddOneVar('WhiteLight', 'integer', 'false', 'Status of the white light of the light column, 0:off ; 1:continuous ; 2
flashing');

XML_Answer(idTx, '@');
End;
//-----Procedure
SendCommandList; // no command is processed by XE but we need to send an answer
Begin
idTx := XML_Create;
XML_Write(idTx, 'Command-SD', '', '');
NodeNum := 0;

```

```

XML_Answer(idTx,'@');
end;
//-----//
SendRecipesList
// - Called by the main in respons of the message "RecipeList-Request"
//-----Procedure
SendRecipesList;
Var
EOL : boolean; // End Of List
RecipeNum,RecipeID : integer;
Section,RecipeTitle : String;
Begin
idTx := XML_Create;
EOL := false;
RecipeNum := 0;
for RecipeID:=1 to 100 do
Begin
Section := 'Recipe#'+IntToStr(RecipeID);
RecipeTitle := ReadIniStr(INI_FNAME, Section, 'LabelRecipeName','');
if Length(RecipeTitle)>0 then
Begin
XML_Write(idTx,'RecipeList-Response/'+Recipe','',Section+':'+RecipeTitle);
End;
end;
XML_Answer(idTx,'@');
End;
//-----//
SendPingResponse
// - Called by the main in respons of the message "PING-REQUEST"
//-----Procedure
SendPingResponse;
Begin
idTx := XML_Create;
MessageType := 'PING-RESPONSE';
XML_Answer(idTx,'@');
End;
//-----//
Debug_PropList
// for debug purpose only
//-----Procedure
Debug_PropList; // output properties to the console
Begin
Writeconsole('Prolog: '+Prolog);
Writeconsole('ContentLength: '+IntToStr(ContentLength));
Writeconsole('MessageIDRx: '+IntToStr(MessageIdRx));
Writeconsole('MessageIDTx: '+IntToStr(MessageIdTx));
Writeconsole('TransactionID: '+IntToStr(TransactionId));
End;
//-----//
AddOneAlarm
// - Called by SendAlarmList only just bellow
// Example of XML
// <Alarm-SD>
// <Alarm name="ComLostS3PC1"> <Description> Communication lost with S3PC1 </Description> <Variable name="VarName"/>
</Alarm>
// <Alarm name="ComLostAMAT"> <Description> Communication lost with AMAT</Description> </Alarm>
// </Alarm-SD>
//-----Procedure
AddOneAlarm(AName,SDescription,VarName:String);
Var Path : String;
Begin
XML_Write(AlTx,'Alarm-SD/'+Alarm','name',AName);
Path := 'Alarm-SD/#'+IntToStr(NodeNumAl);
XML_Write(AlTx,Path+'/Description','',SDescription);
if length(VarName)>0 then XML_Write(AlTx,Path+'/Variable','name',VarName);
NodeNumAl := NodeNumAl + 1;
End;
//-----//
SendAlarmsList
// - Called by the main in respons of the message "Alarm-SD-Query"
//-----Procedure
SendAlarmsList;
Var
bit : integer;
AlarmName,VarName : String;
Begin
InitS3PC_TabText_And_Level;
AlTx := XML_Create;
NodeNumAl := 0;

```

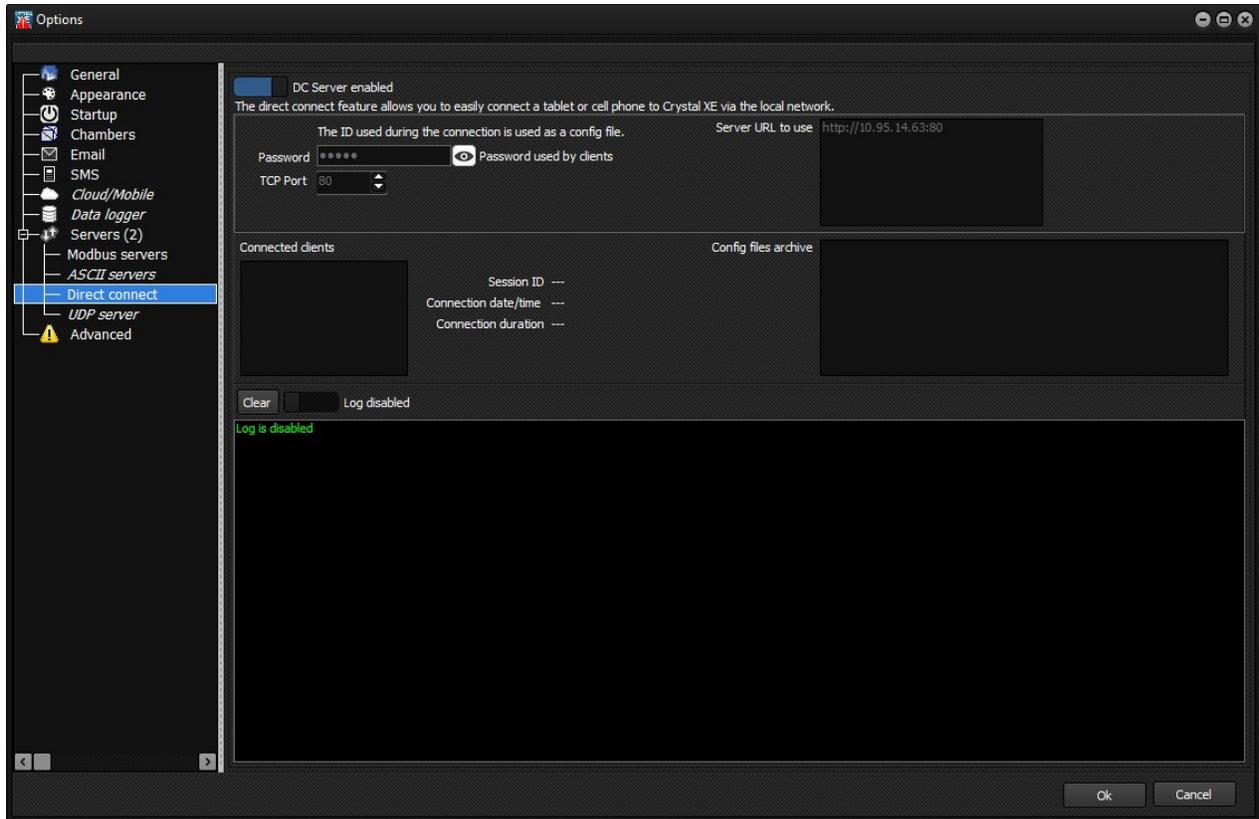
```

//AddOneAlarm('MainAlarm','The main alarm is activated, see Crystal software for more information','');
AddOneAlarm('ComLostS3PC1','Communication lost with the S3PC1','');
AddOneAlarm('ComLostAMAT','Communication lost with the AMAT','');
for bit:=0 to 63 do
Begin
  AlarmName := TabAlarmName[bit];
  if Length(AlarmName)>0 then
  Begin
    if TabCellOneOnly[bit] then VarName:=''; else VarName:='Cell';
    AddOneAlarm(AlarmName,TabAlarmText[bit],VarName);
  End;
end;
XML_Answer(AltX,'@');
End;
//-----//
SendEventsList
// Called by the main in respons of the message "Event-SD-Query"
//-----Procedure
SendEventsList;
Begin
  CreateEventList;
  XML_Answer(EvTx,'@');
End;
//-----// MAIN
entry point of the script
//-----Begin
ComError := 0;
idRx := XML_Create;
if XML_ReceiveBuf(idRx) then
Begin
  if compareText('DATA-REQUEST',MessageType)=0 then
  Begin
    S := XML_Read(idRx, '', '', '?');
    if (CompareText('Variable-SD-Query',S)=0) then SendVariablesList;
    if (CompareText('Alarm-SD-Query',S)=0) then SendAlarmsList;
    if (CompareText('Event-SD-Query',S)=0) then SendEventsList;
    if (CompareText('Command-SD-Query',S)=0) then SendCommandList;
    if (CompareText('RecipeList-Request',S)=0) then SendRecipesList;
  end;
  if compareText('PING-REQUEST',MessageType)=0 then SendPingResponse;
  if compareText('PING-RESPONSE',MessageType)=0 then SendPing := 0;

  // increment the messageId
  MessageIdTx:=MessageIdTx+1;
  if MessageIdTx>$FFFFFFFF then MessageIdTx:=1;
End;
End;

```

10.9.1. Direct connect



This option activates the direct connection mode. This mode enables another application to receive Crystal XE data on a regular basis. The XE mobile application (under Android and Windows) is perfectly suited to this protocol. It is also possible to use this protocol to activate remote chamber on another Crystal XE in the network.

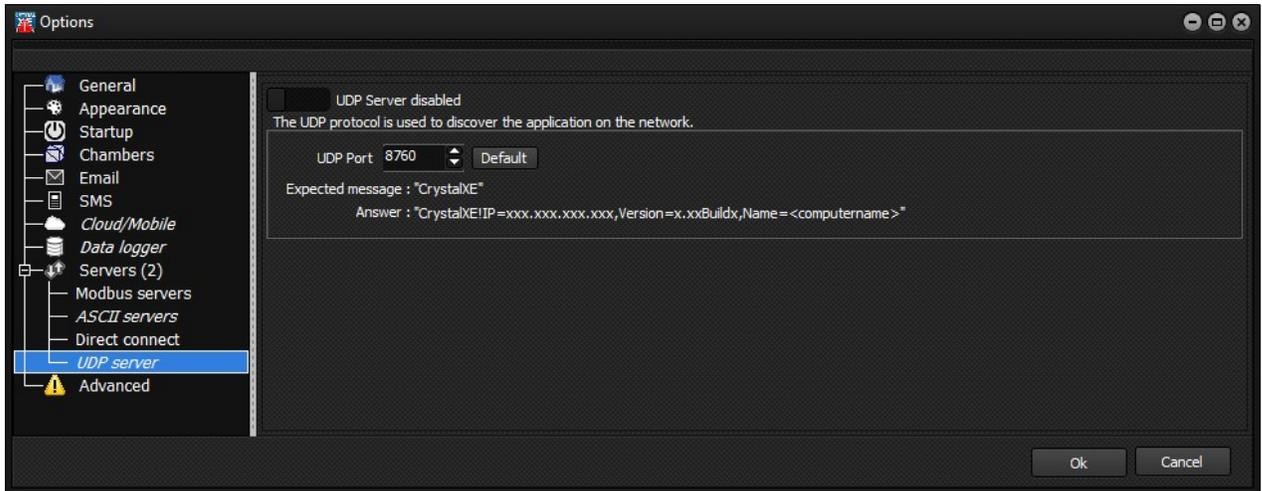
Crystal XE uses the protocol HTTP over TCP/IP.

It uses the POST method.

The protocol is the same as that specified for sending data to the cloud.

➔ For more information, refer to the section “Direct connect protocol”

10.9.2. UDP Server

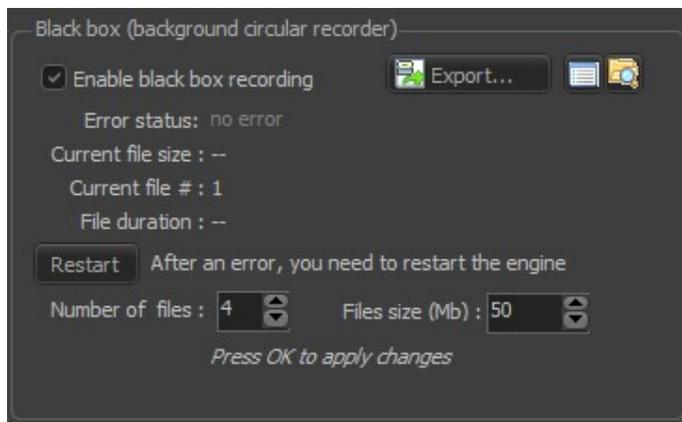


This option identifies Crystal XE on a network. By activating this option, Crystal XE will respond to a message sent in broadcast with the identifier “CrystalXE”. The response will contain the PC's IP address, Crystal XE version and computer name.

10.10. Advanced

These options are only used for debug purpose.

10.10.1. Black box (background circular recorder)

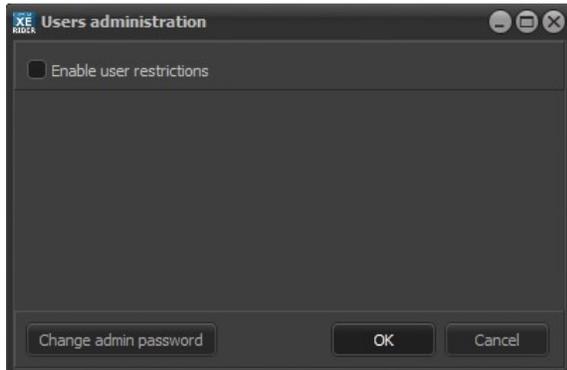


The black box allows you to record communication data between the devices connected to Crystal XE.

11. USERS



Drop the *Setup* menu down or click on the following icon  and click on *Users* to open the *User administration* window.



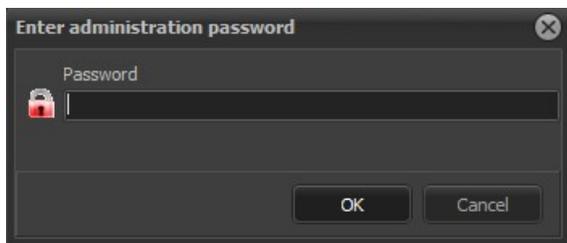
User restrictions are disabled by default

You can also open the *Users administration* window by right-clicking on the *User* icon, located in the top left-hand corner of the main interface, and selecting *Manage Users*.



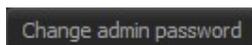
All modifications of configuration are saved to the file '*config.xfg*' (XML file) located in your project directory.

You systematically need to enter the admin password to open the *Users administration* window.



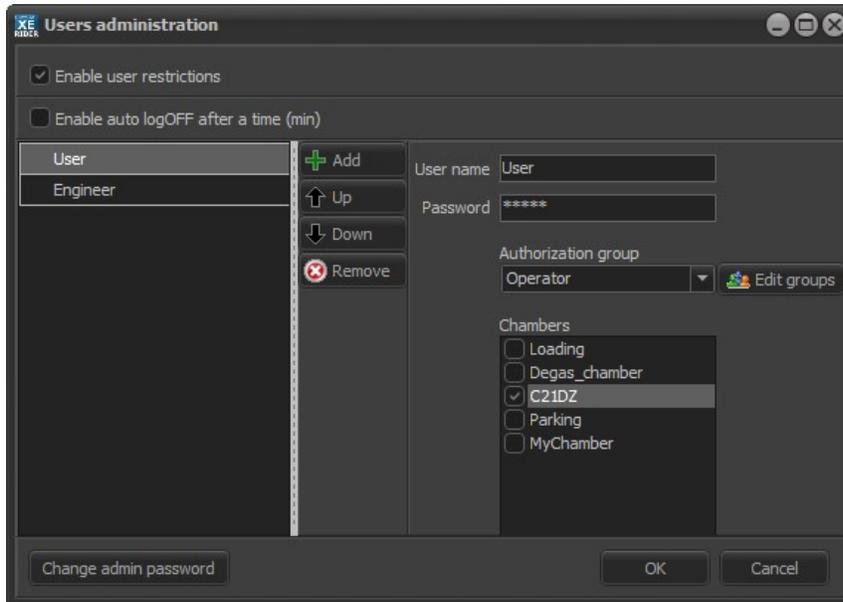
The default password is 'admin'.

You can change the password by clicking on the *Change admin password* button, in the bottom left-hand corner of the window.



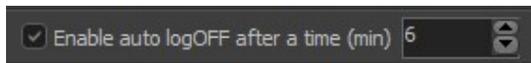
Only the admin password can give you access to the *Users management* window. Check the *Enable user restrictions* box to allow user logins and restrictions. When the box is unchecked, all user restrictions are disabled and users cannot log in.





User administration window

By default, the system automatically log the user off after 6 minutes of inactivity. You can adjust the inactivity period using the following spinner:

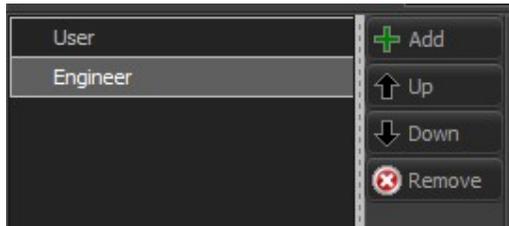


To disable automatic log off, uncheck the box as follows:



11.1. Managing users

The box on the left side of the window lists all users.

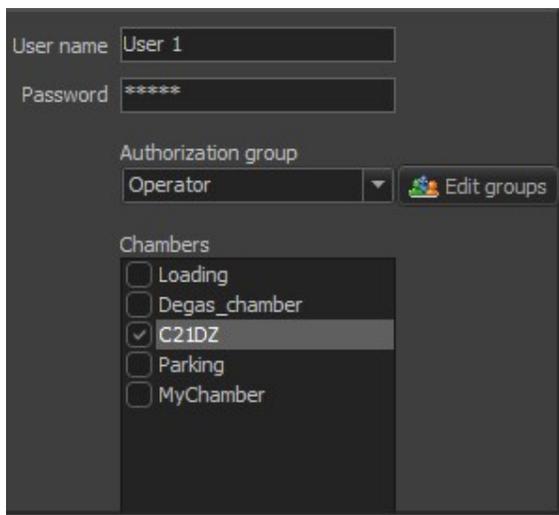


By default two users are created: an operator and an administrator.

Use the following buttons to add a new user, change the list order or delete a user.



Select a user to display its settings on the right side of the window:

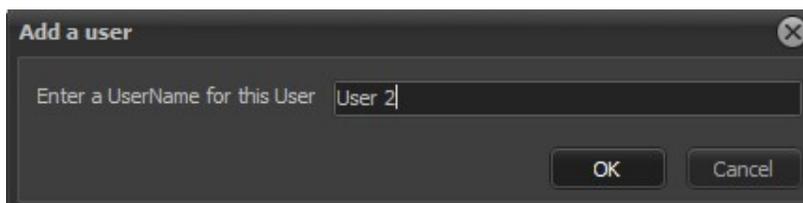


You can modify existing user name, password, the group they are assigned to and the chambers within which they can perform tasks (according to their group permissions).

11.1.1. Creating a new user

Click on the *Add* button to add a new user: 

Enter the **user name** in the dialog box:



Assign a **password** to the user

User name

Password

From the drop-down list below, select the group the user should be assigned to:

Authorization group

Admin

Operator

Each user group has a defined set of tasks their members are allowed or forbidden to perform. By default two user groups are available: administrator and operator. You can modify group restrictions or create your own user groups by clicking on *Edit groups*:



In the Chambers box, select the chamber within which the user permissions apply:

Chambers

Loading

Degas_chamber

C21DZ

Parking

MyChamber



If the chamber is unchecked, the user will not be able to perform any tasks within that chamber.

11.1.2. Managing groups

Click on the *Edit group* button to open the *Group authorization editor*.

The **Group authorization editor** dialog box shows a list of groups on the left (Admin, Operator) and a list of permissions on the right. The **Admin** group is selected, and its permissions are listed with checkboxes. All permissions are currently checked.

Group	Permission	Status
Admin	All	Checked
	Hardware configuration	Checked
	Options	Checked
	User rights	Checked
	File New	Checked
	File open	Checked
	Control equipment	Checked
	Setup equipment	Checked
	Disable com & sockets	Checked
	Securities On/Off	Checked
	Securities change	Checked
	Edit forms	Checked
	Edit script	Checked
	Edit templates	Checked
	Recipe control	Checked
	Batch control	Checked
	Recorder control	Checked
	Acknowledge alarms	Checked
	Exit application	Checked

In the *Group authorization* editor, you can assign permissions to groups, modify group names and add or delete groups.

Select one of the groups in the left box to display its permissions on the right side of the window.

Two groups are already created: *Admin* and operator. By default the administrator group (*Admin*) grants all privileges.

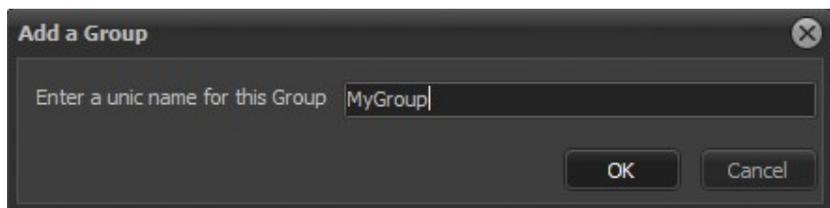
You can modify any existing group permissions by checking/unchecking the corresponding boxes.

Use the following button to add a new group, modify the list order or delete a group:

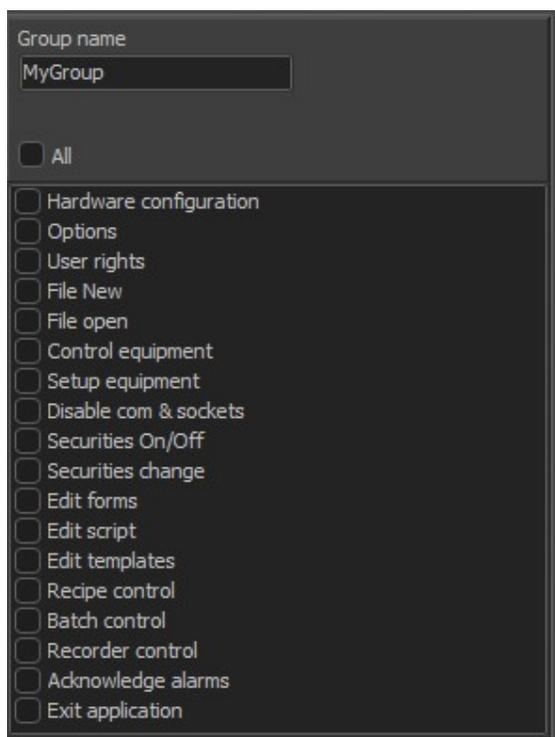


To create a new group, click on the *Add* button: 

Enter the group name in the dialog box:



To assign permissions to a group, check the corresponding boxes:

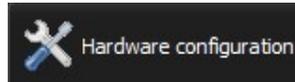


When the box is unchecked the user is not allowed to perform that task in any chamber.

Group permissions

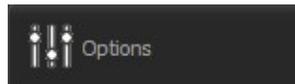
Hardware configuration

Allow/forbid users to access the Project configuration window and configure the chambers, equipment and devices.



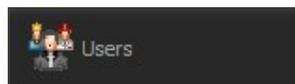
Options

Allow/forbid users to access the Options window and modify *Crystal XE* set-up.



User rights

Allow/forbid users to access the User management window.



At least one user must be allowed to modify user rights.

File New

Allow/forbid users to create new files (forms, scripts, projects).

File open

Allow/forbid users to open files (forms, scripts, security *Files*, projects, etc.)

Control equipment

Allow/forbid users to modify the equipment values (such as set points, etc.).

Setup equipment

Allow/forbid users to modify the sub equipment setup values (such as minimum and maximum set points etc.).

Disable com & sockets

Allow/forbid users to enable/disable communication ports (Devices tab).

Securities On/Off

Allow/forbid users to enable/disable security agents (alarms).

Securities change

Allow/forbid users to modify the security agents' threshold values (alarms).

Edit forms

Allow/forbid users to edit forms (access the *Tool palette*).

Edit script

Allow/forbid users to edit scripts in the Pascal editor.

Edit templates

Allow/forbid users to edit equipment and device templates (In the Project configuration window).

Recipe control

Allow/forbid users to control the execution of recipes.

Batch control

Allow/forbid users to control the execution of batches.

Recorder control

Allow/forbid users to control recorders and edit recorder templates.

Acknowledge alarms

Allow/forbid users to acknowledge alarms that occurred.

Exit application

Allow/forbid users to exit *Crystal XE* program.

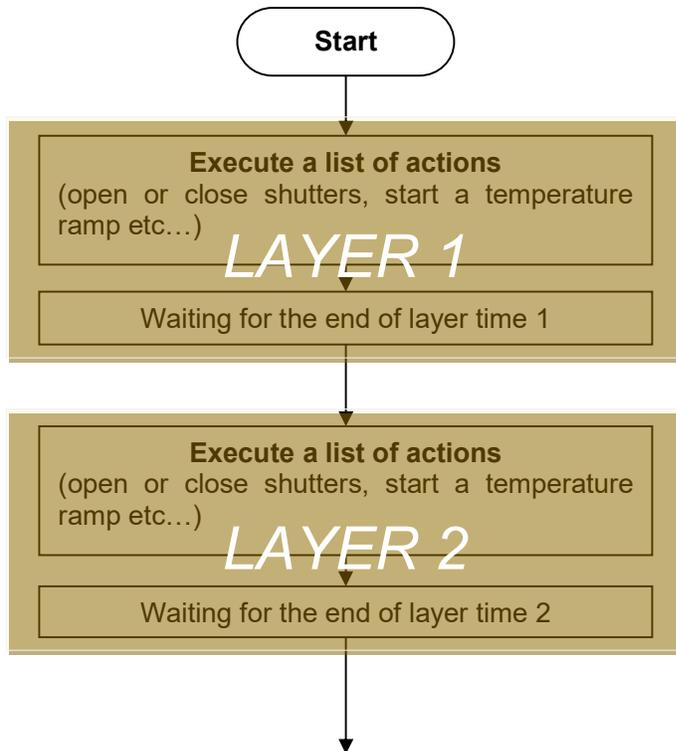
12. RECIPES

12.1. How does the recipe work?

In MBE, a recipe is based on a succession of layers.

A layer is determined by a list of actions to be performed and a time to wait.

The waiting time can be entered by the user or can be calculated according to the thickness of the material to be grown.



In MBE, the recipe engine is essential for epitaxies to be repeatable. The accuracy of the recipes is therefore essential. Crystal XE fully complies with these basic requirements.



Realtime: the heart of Crystal XE

Crystal uses an absolute and not a relative time base. This means that a drift due to a time-consuming stain will have no impact on the time of the layers. In addition, the process of executing the tasks performed at each layer uses all the capacities of the processor to reduce time as much as possible. Everything has been thought of to optimize times such as priority management throughout the management of communications with the devices in order to guarantee maximum repeatability with time drifts of less than 9ms.



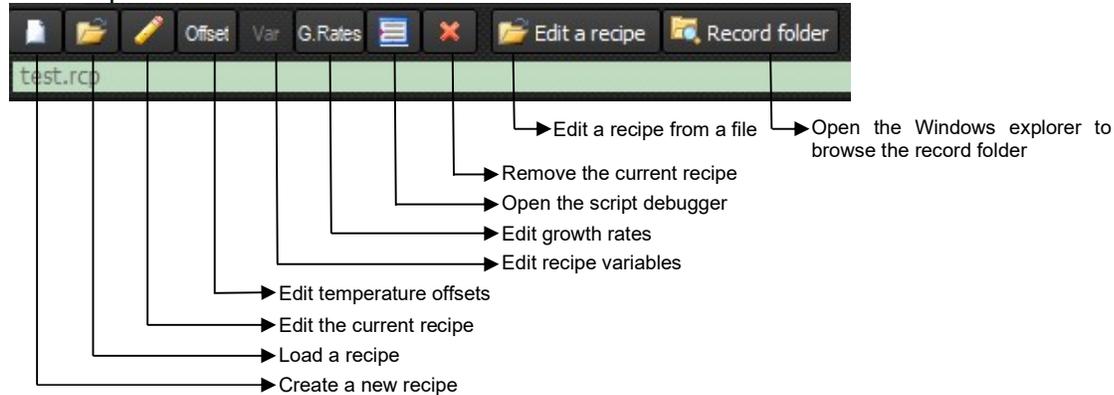
Crystal XE can manage several process chambers simultaneously and therefore several recipes can be executed in parallel.

The operating principle of the recipes is the same whether the editor was used to write them, whether with the recipe editor (“rcp” file) or by script recipes (“pas” file)

12.2. Recipe editor

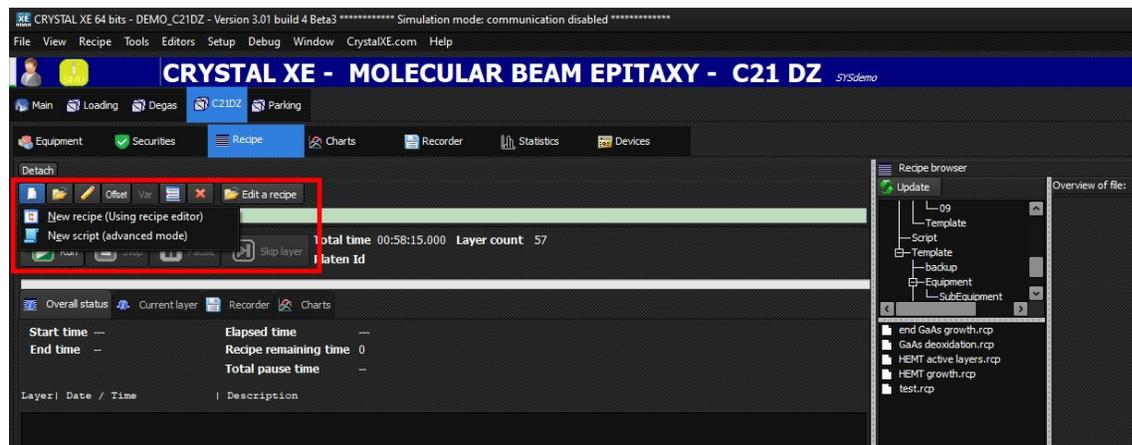
The editor offers an easy to use interface, requires no programming skills, and allows many possibilities.

The recipe tool bar



Start a new recipe

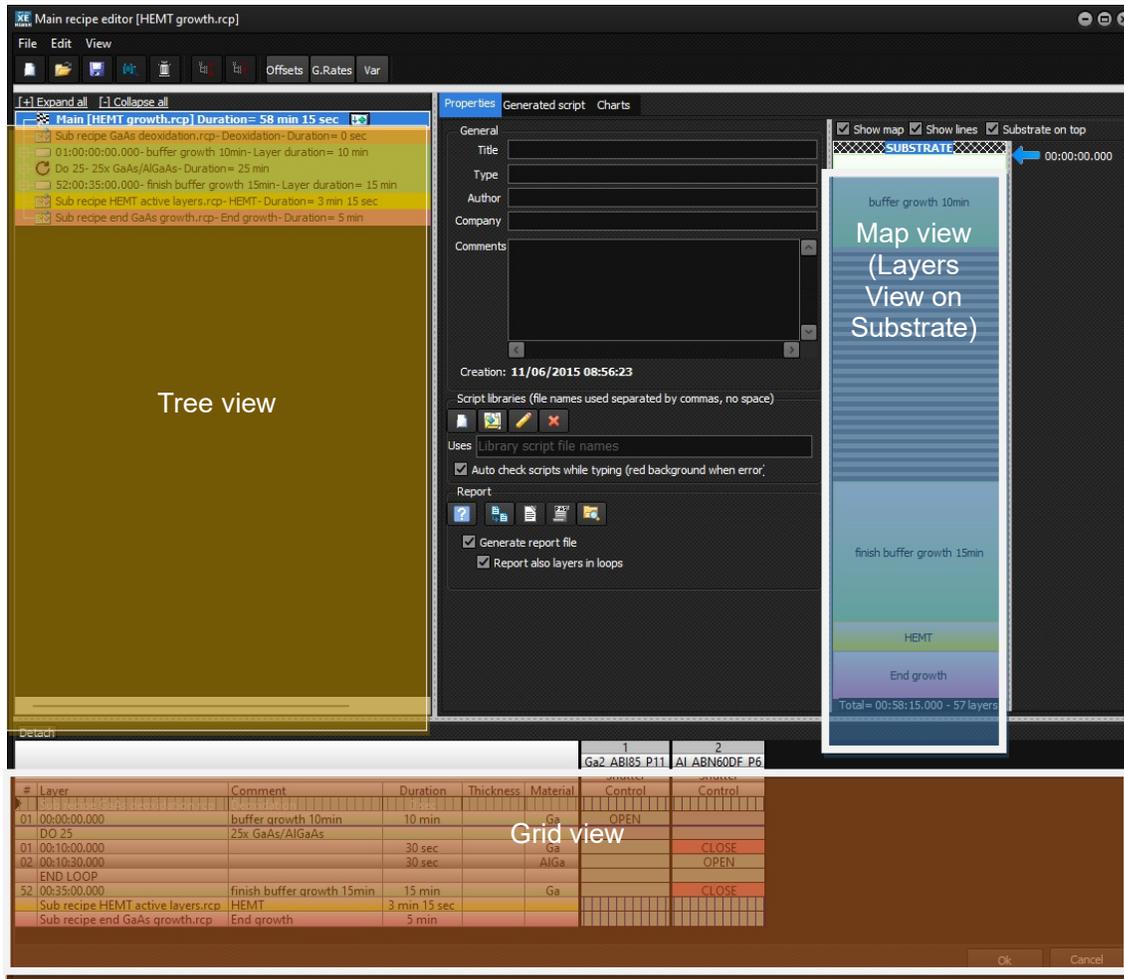
To start a new recipe, click on the button “Create a new recipe” and on the first choice “New recipe (Using recipe editor)” to open the Main recipe editor.



The recipe created is saved in a “.rcp” file, unlike script recipes which are text files with the extension '.pas' and containing a Pascal program.

The file format of the “.rcp” file is XML.

12.3. Overview of the recipe editor



Crystal XE offers two possibilities to edit your recipe.

Tree view and grid view:

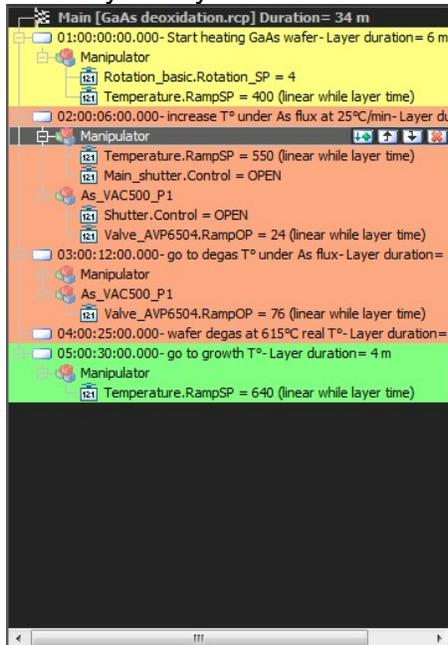
You can use either the **tree view** or the **grid view** and switch between these two views at any time.

In the **tree view**, the top-level element represents either:

- **A layer** which is defined by a list of actions to be executed and ends with a waiting time.
- **A loop** is useful to repeat a list of top-level elements. Several loops can be nested. It is also possible to add a ramp in the beginning of the loop, outside a layer, to have a ramp whose final set point is at the end of the loop.
- **A pause** to suspend the execution of the recipe until the user presses the resume button.
- **A sub-recipe** which is another recipe included. Several inclusions are possible.
- **A script** allows you to insert a piece of pascal program into a simple recipe. It is then possible to mix the two recipe possibilities.

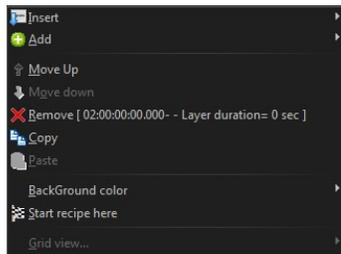
12.4. Work with the tree view

Create your layers

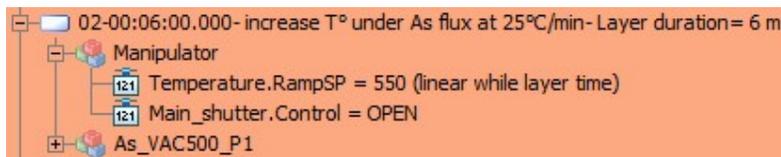


Use the button  to add a layer or  or  to move the selected layer, or  to delete the selected layer.

Right-click on any of the layers to display other available options.

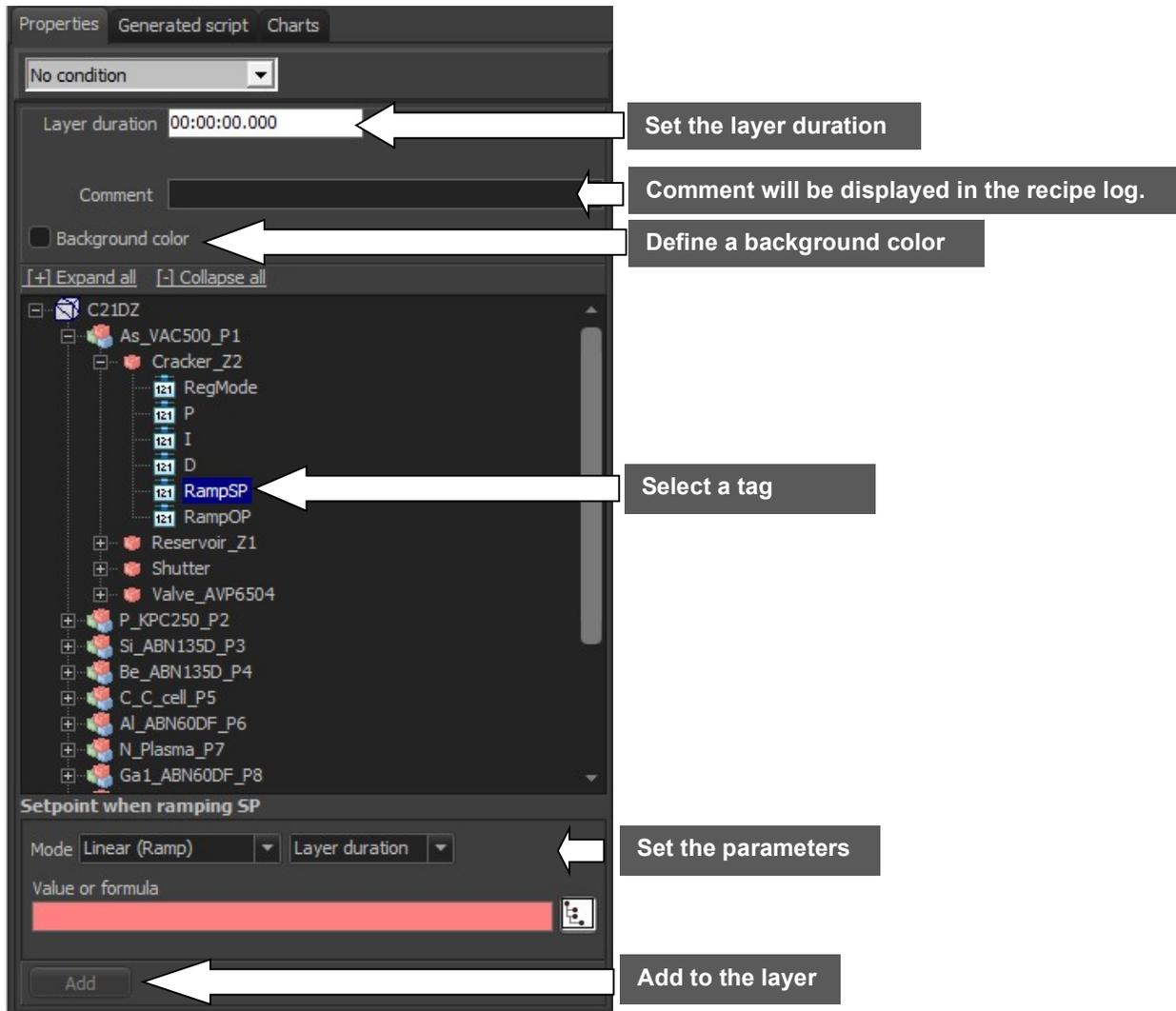


For each layer, you can expand the list of equipment and associated parameters that have been defined:



Example of a layer

Defining the parameters



In the center of the editor window, the *Properties* tab displays all equipment and sub equipment and their associated configurable properties listed in a tree view.

For each layer, you can:

- specify the layer duration,
- enter information for the user,
- select a property and set the parameters,
- add these parameters to the layer.

Expand the equipment and sub equipment lists and select the desired property to configure the parameters using the pane below.

The configured parameters will be added to the layer in the tree structure on the left side of the editor window.

12.5. Work with the grid view

How to detach the grid view

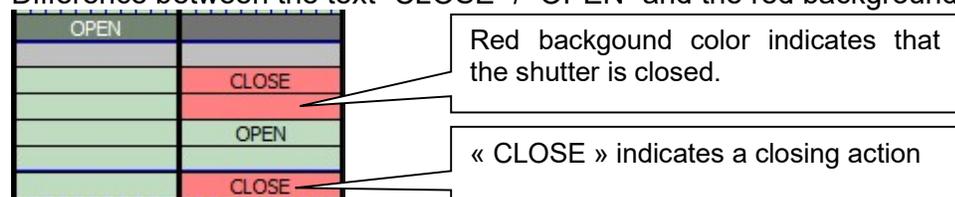
The grid view can be detached. To do this click on the detach button which is located at the top right. To restore the window to its original position, close the window with the cross at the top right.

Work faster with the grid view

The grid view allows you to display all the equipment and sub-equipment in separate columns. Each line contains a top-level element. it allows you to have a more synthetic view of the evolution of temperatures, shutters, valve positions etc ... Click on an element such as for example a temperature ramp to be able to easily modify it in the central view of the editor.

Shutters: Double-clicking on a cache cell allows you to simply change its state. If the shutter was closed, a double click will open it.

Difference between the text “CLOSE” / “OPEN” and the red background color or green.



The **background color** indicates the presumed state of the shutter, considering a previous action. For example, a green background color indicates that the shutter status is supposed to be open because an opening action has occurred previously.

The **text Open or Close** indicates that an opening or closing action has been initiated.

Double click to toggle the state of the shutter.

Right click and select  Remove [Shutter.Control = CLOSE] to delete the action.

It is not recommended to have several successive identical actions. This causes unnecessary write requests to the hardware which can in some cases slow down the frequency of communication with the devices.

Take care to check the status of the shutters because it can be wrong in some cases, especially when returning a loop, or after a script or after a sub recipe.

Material column:

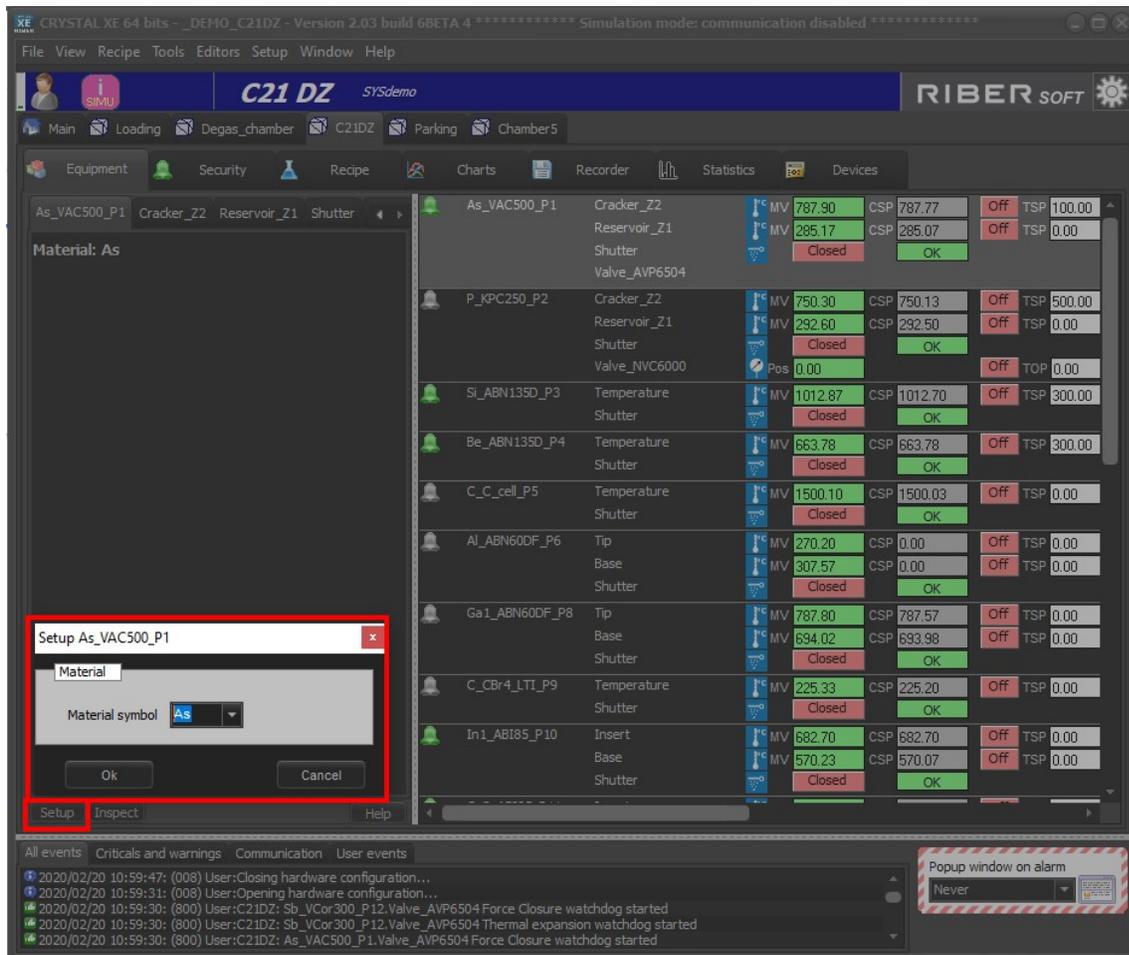
					1	2	3	
					As_VAC500_P1	Shutter	P_KPC250_P2	
					Shutter	Shutter	Cracker_Z2	
#	Layer	Description	Duration	Thickness	Material	Control	Control	RampSP
01	00:00:00.000		4 sec		P	CLOSE	OPEN	
	Script (len=32)							
	DO 2							
01	00:00:04.000		3 sec		As	OPEN	CLOSE	123 (linear while layer time)
02	00:00:07.000		2 sec		As			
	END LOOP							

The material column is only visible in the grid view.

When possible, for each layer, the material that is being grown is displayed in the material column. To know the material that is being grown, Crystal only takes into account shutters that are open. It is therefore necessary to verify this information yourself.

The symbols which are used as material can be modified within each equipment.

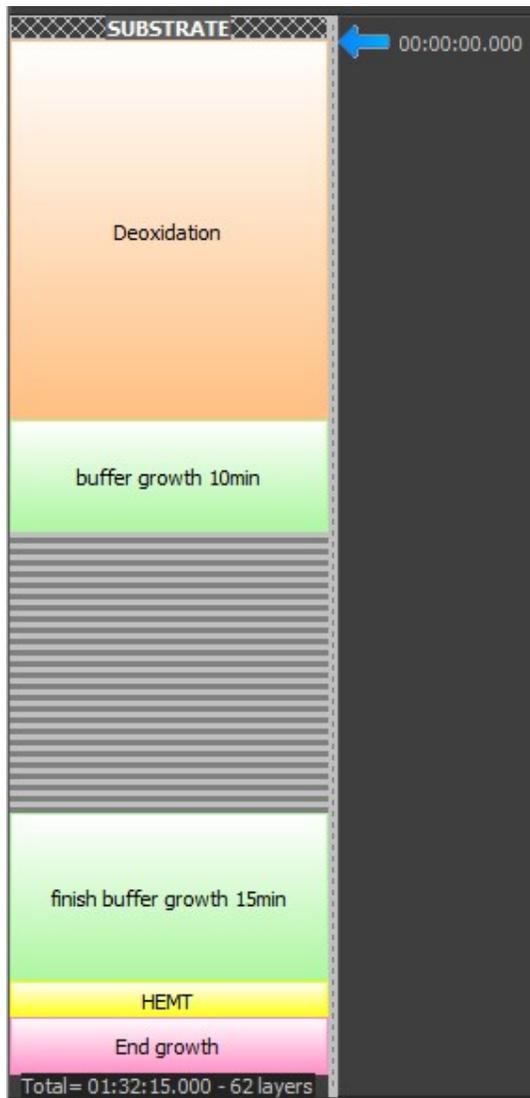
To modify the symbols, go in the Equipment view, select the appropriate equipment and push on the setup button located at the bottom right of the detail view.



How to change the material symbol of a cell which is used in the recipes.

12.6. The map view

The map view is a representation of the assembly of layers based on the time of the layers. Be careful, this does not represent the thickness of the layers but just gives an idea of the result obtained.



The map view represents the durations of all the layers.

The map view is interactive: click on a layer to select the current layer. Please note, if you click on the element or a loop, the first layer of the loop will be selected. The color of each layer can be customized.

The update time of the map view can be very important if the recipe is made up of many layers. In this case, you can deactivate the drawing of the map view. This will significantly increase the refresh times. To deactivate the drawing of the map view, uncheck the box "Show map"



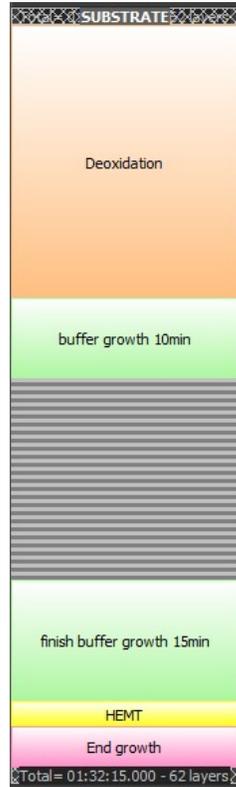
Options available to draw the map view.

The check box **Show line** will display or not the lines of separation of the layers. Disable this option to make the display faster.

Substrate on top is used either to draw the substrate at top or to draw it at the bottom.



Substrate on bottom

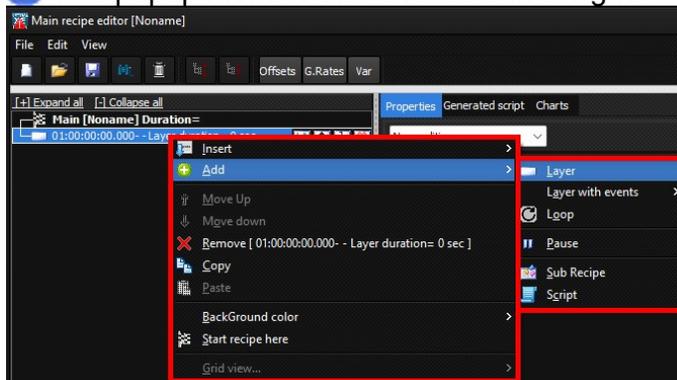


Substrate on top

12.7. How to add a top-level element

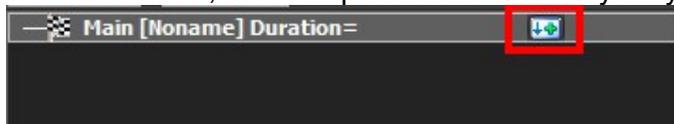
To add or insert a new top-level element, right-click with the mouse in the **tree view** or in the **grid view** to open the context menu.

 This popup menu is also available in the grid view.



Pop-up menu which is displayed when the right mouse button is clicked in the tree view or in the grid view.

In the tree view, it is also possible to add a layer by pushing the “+” button:

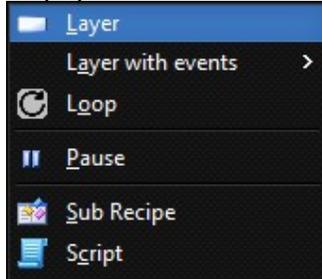


Button to add a new layer.

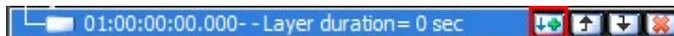
12.8. Add an empty layer

Add a layer by using the popup menu or the button in the tree view.

Popup menu



Button  in the tree view:

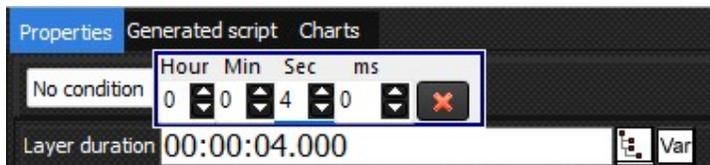


After adding a new layer or after selecting an existing layer, enter or change the layer duration.

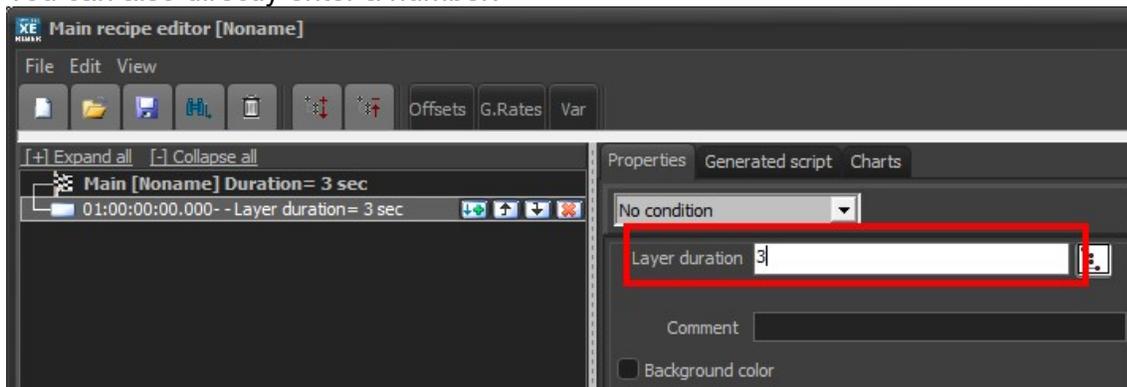
It is possible to enter either a **constant** layer duration or a **variable** layer duration.

Constant layer duration

To use a constant value, you can push on the small arrows and change the hours, minutes, seconds or milliseconds:



You can also directly enter a number:



Enter the layer duration

By default, the unit of time is the second, so if you enter “3” followed by Enter, the time will be 3 seconds but you can also enter the time in the format “HH:MM:SS.MMM”

The time format is the following

<HH=HOUR>:<MM=MINUTES>:<SS=SECONDS>:<MMM=MILLISECONDS>

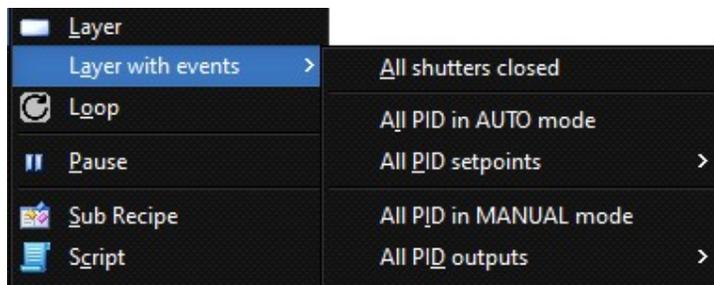
Variable layer duration

A variable layer duration can be:

- Using tags or properties like a global variable,
- A formula based on one or several tags,
- The result of a function in a script library,
- A local variable.

All these possibilities are described in the next sections.

12.9. Add a layer with events



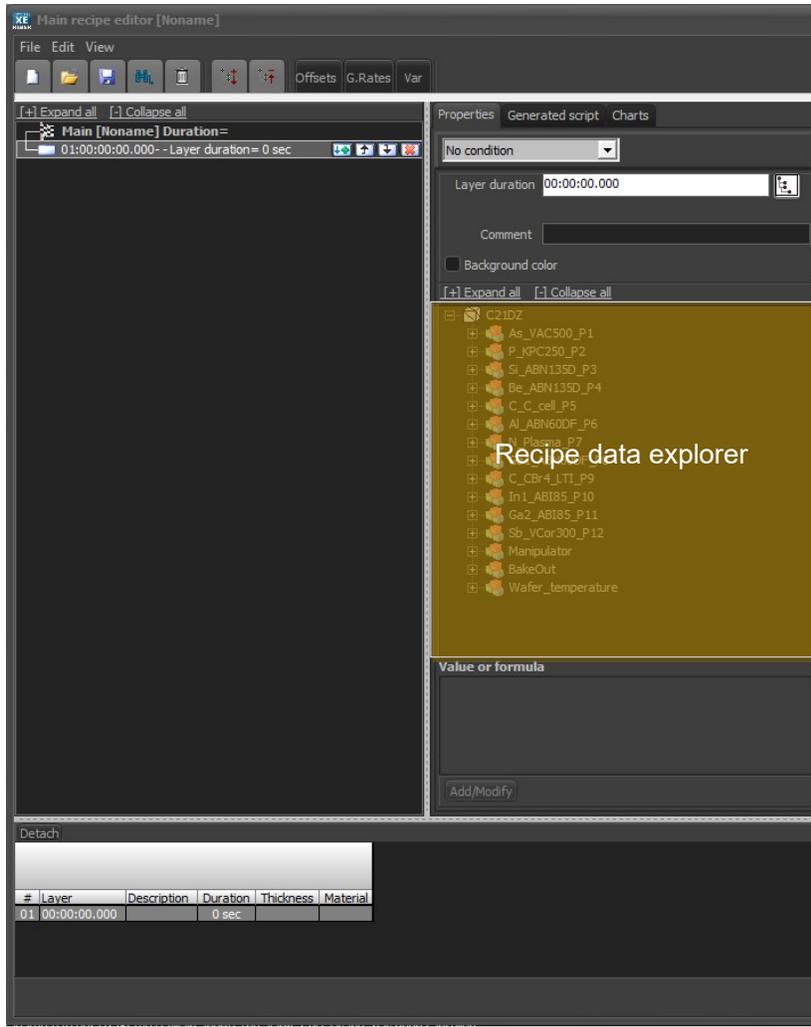
These choices allow you to add a layer while inserting predefined actions.

For example, if you want to start a recipe by ensuring that all shutters are closed, select the first item in the menu “All shutters closed”.

12.10. Add actions to a layer

Once you have added a layer in the recipe, you can add actions to execute.

The center of the window lets you select an action to be performed using the data explorer.



This screenshot shows the recipe data explorer

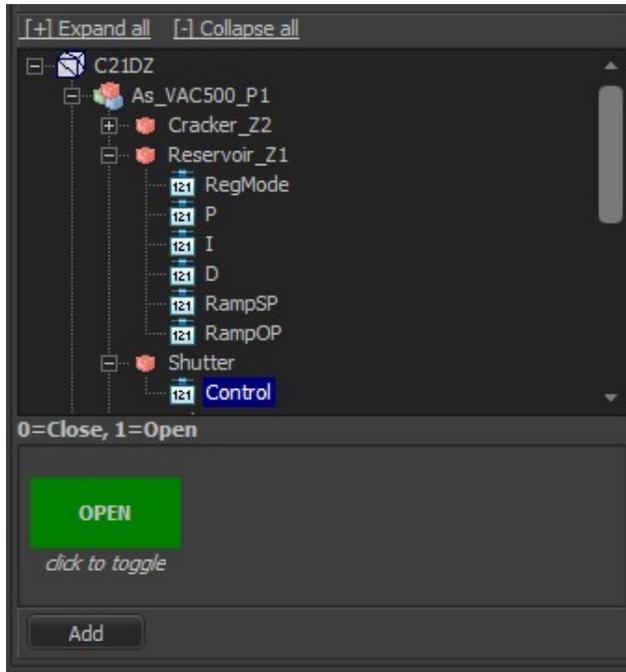
Browse the data explorer and select a tag to add actions in the recipe. You have to open a branch up to a tag.

Depending of the type of tag, several types of actions are possible:

- Binary action: ON/OFF or Open/Close
- Regulation mode Auto/Manual
- Value or formula
- Ramp using a constant value or a formula

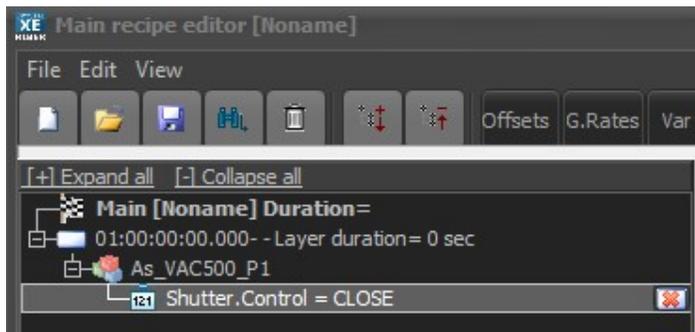
Binary action:

To change the position of a shutter, select the “Control” tag of the shutter.



This screenshot shows how to add a shutter position change in the selected layer.

Press on the button OPEN to toggle between OPEN and CLOSE and then press on the button Add to add the action to the current layer.

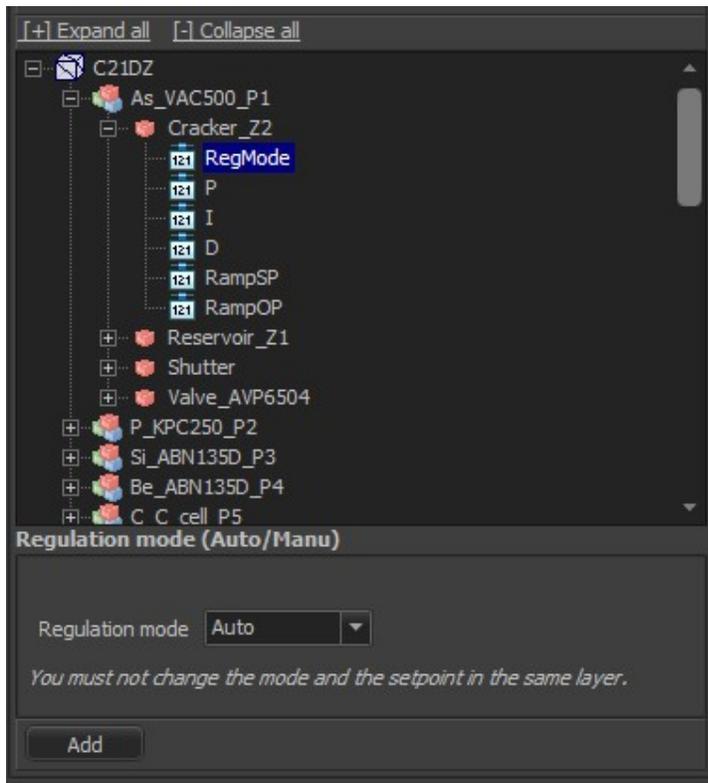


Tree view after adding a shutter event.

#	Layer	Description	Duration	Thickness	Material
01	00:00:00.000		0 sec		CLOSE

Grid view after adding a shutter event.

Regulation mode Auto/Manual:

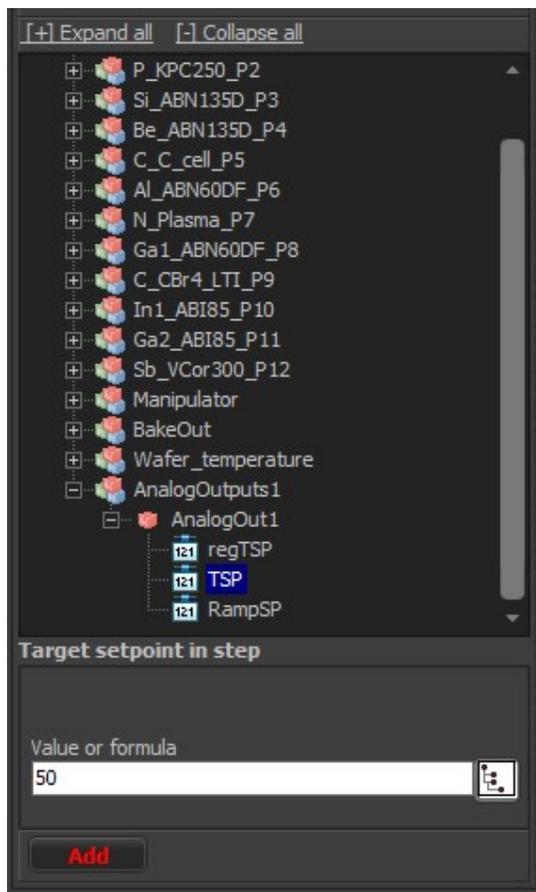


Tag to change the regulation mode of a temperature controller.

Important: when changing the control mode of a temperature controller, no other action must be performed in the same layer of the same controller such as sending a set point. The minimum duration of the layer must be 1 second to be sure that the controller has switched its operating mode.

Value or formula other than a ramp:

This type of action can be used for example to change the value of an analog output.



Simple value to change.

It is possible to enter either a constant value, a formula using tags or a script function or a combination of all.

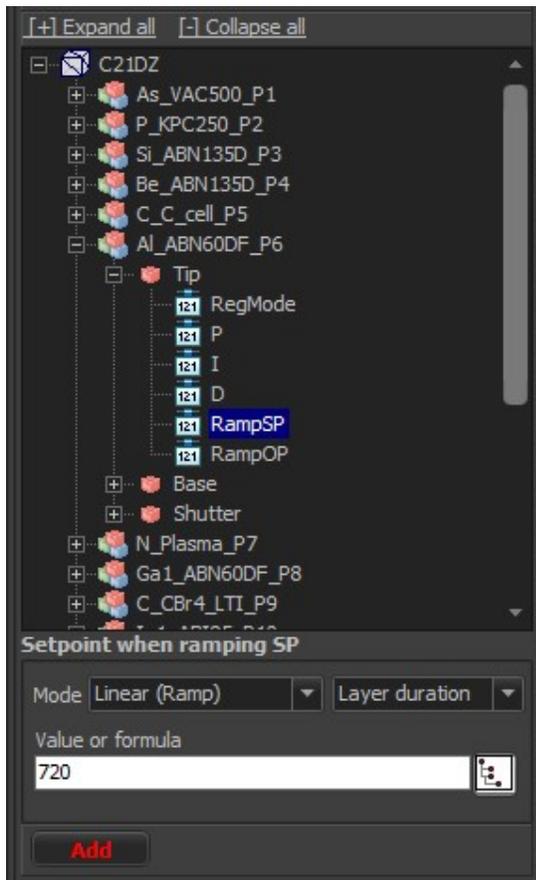
Ramp using a constant value or a formula

The ramps are generally used by temperature controllers whether in automatic mode (temperature setpoint) or in manual mode (output value in percentage).

The ramps can be used in other cases such as controlling a valve position.

Use RampSP or RampOP ?

Depending on the operating mode of the regulator (Auto or manual), you will have to choose the appropriate tag. In auto mode, choose the RampSP tag while in manual mode, choose the RampOP tag.



Change the temperature setpoint of a controller using a ramp.

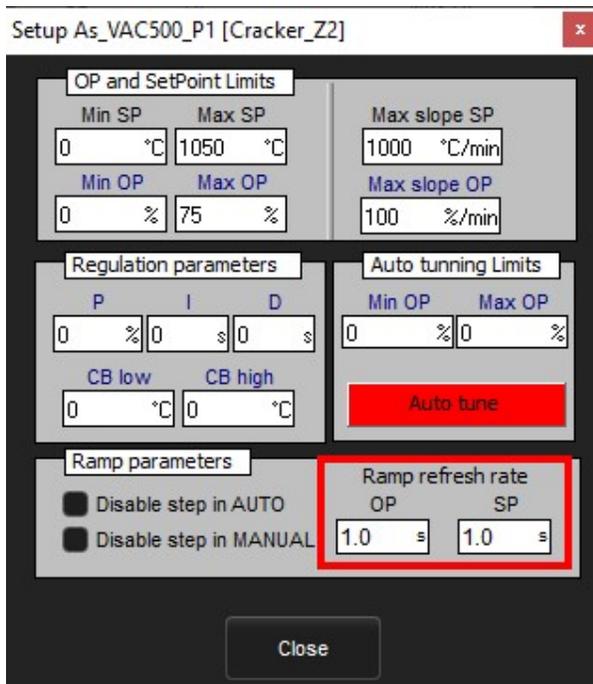
The screenshot below shows a temperature linear ramp with a final setpoint at 720°C. The controller is assumed to be in temperature control mode (AUTO). The duration of the ramp is equal to the duration of the layer in which the action is located.

You can change the mode to Step or if it exists, to another ramp profile. You must define other ramp profiles in the **options** for them to appear here.

Step mode means that the desired setpoint will be sent directly to the controller. In this case, the regulator will go as quickly as possible to reach the final setpoint.

In ramp mode, Crystal XE will periodically send a new setpoint to the temperature controller following the type of ramp selected.

The frequency of sending the setpoint to the regulator is defined in the setup of the sub equipment (go in the Equipment view select the temperature sub equipment and click on the setup button located in the bottom of the details view)



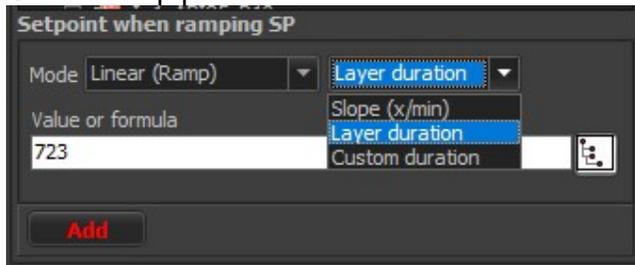
This screenshot shows how to change the period to send setpoint when ramping.

By default, the period is one second and it is not recommended to change it.

Start ramps on “Measured value” or “Current setpoint”?

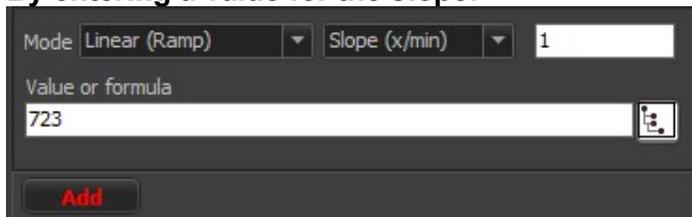
When two ramps follow one another, Crystal will start the second ramp from the measured value or from the last setpoint sent. This choice can be changed in the Crystal options. (Setup / Options menu, general tab and Ramps group)

Others ramp parameters:



This screenshot shows all the different slopes

By entering a value for the slope:

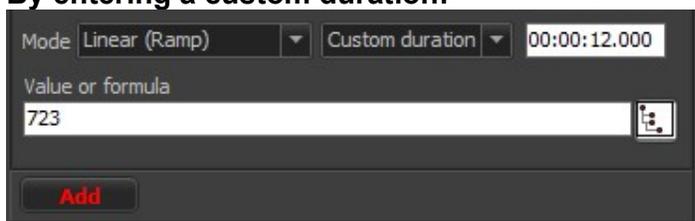


Enter a slope value

Select “Slope (x/min)” if you want to enter a slope. In this case, the duration of the ramp can either be less than the duration of the layer, or be greater than the duration of the layer and it

can even continue when the recipe is finished.

By entering a custom duration:

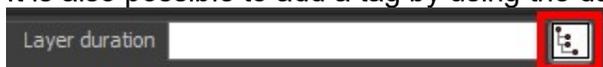


Enter a ramp duration

In this case, you are free to enter the duration of the ramp. The slope will be calculated automatically.

12.11. Using tag or properties (like global variables)

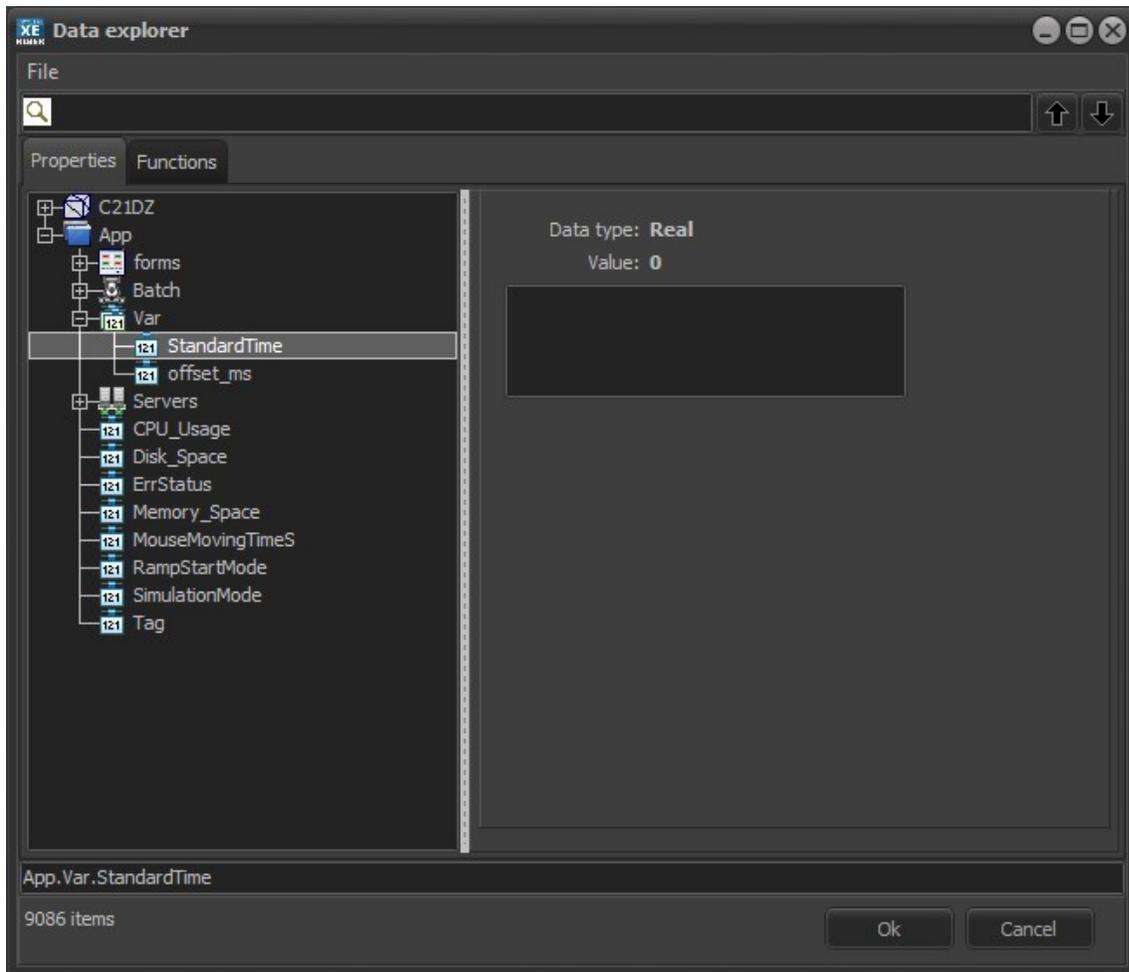
It is also possible to add a tag by using the data explorer button



Open the recipe data explorer to add a tag property

This function is available wherever this button is present, such as the layer duration or the temperature setpoint, etc.

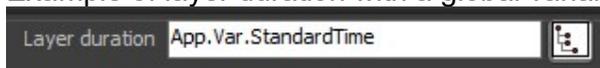
Pressing on this button open the data explorer:



Recipe data explorer that displays a global variable

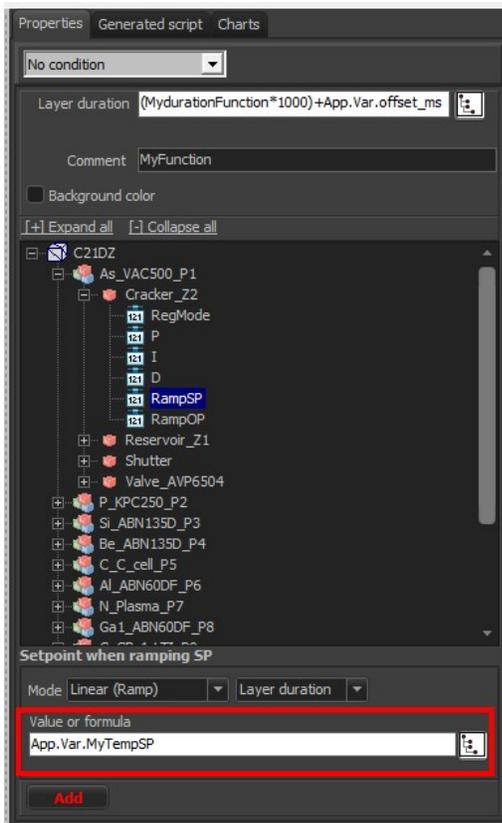
Select the data you want to insert and press OK.
The data will be inserted at the cursor position.

Example of layer duration with a global variable:



When using a global variable, the duration of the layer can be changed during the execution of the recipe by using a background script or by an Indicator in an HMI form (Human Man Interface)

Example of global variable used as a setpoint:



It is also possible to enter a formula.

Example of formula: $(App.Var.StandardTime*1000)+App.var.offset$



When using a formula, all math functions which are available in the scripts are allowed, and parentheses are also allowed. For more details, see the documentation about the scripts.

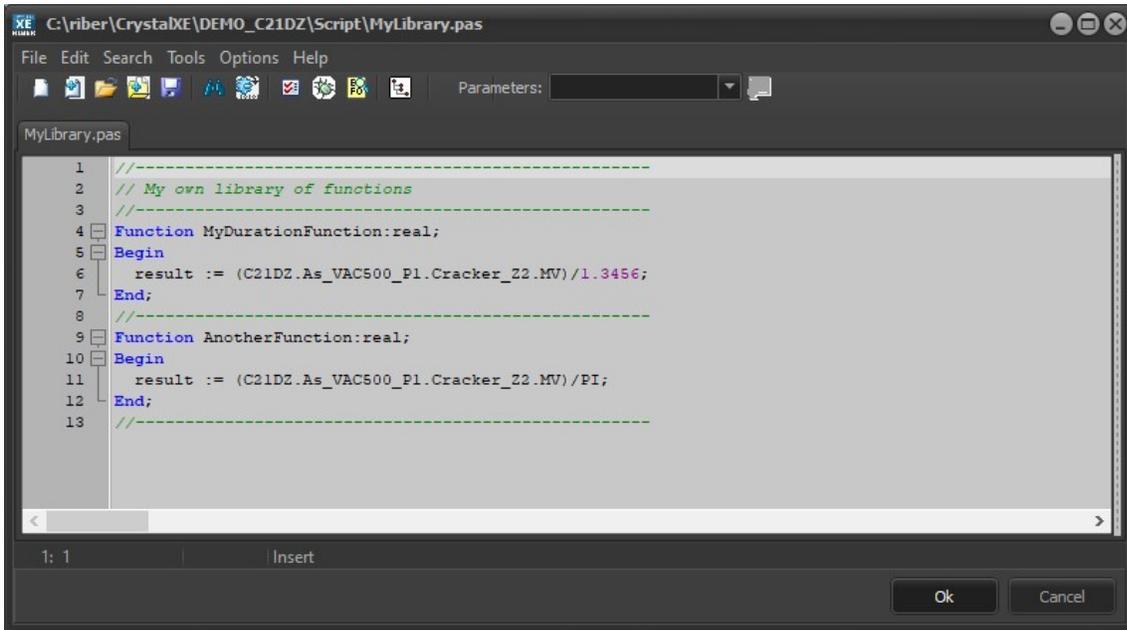
12.12. Using of an external script library

You can use one or more external libraries containing custom functions.

The result of these function can be entered in the calculation of the duration of a layer or as the setpoint value of a temperature regulator, or valve position etc...

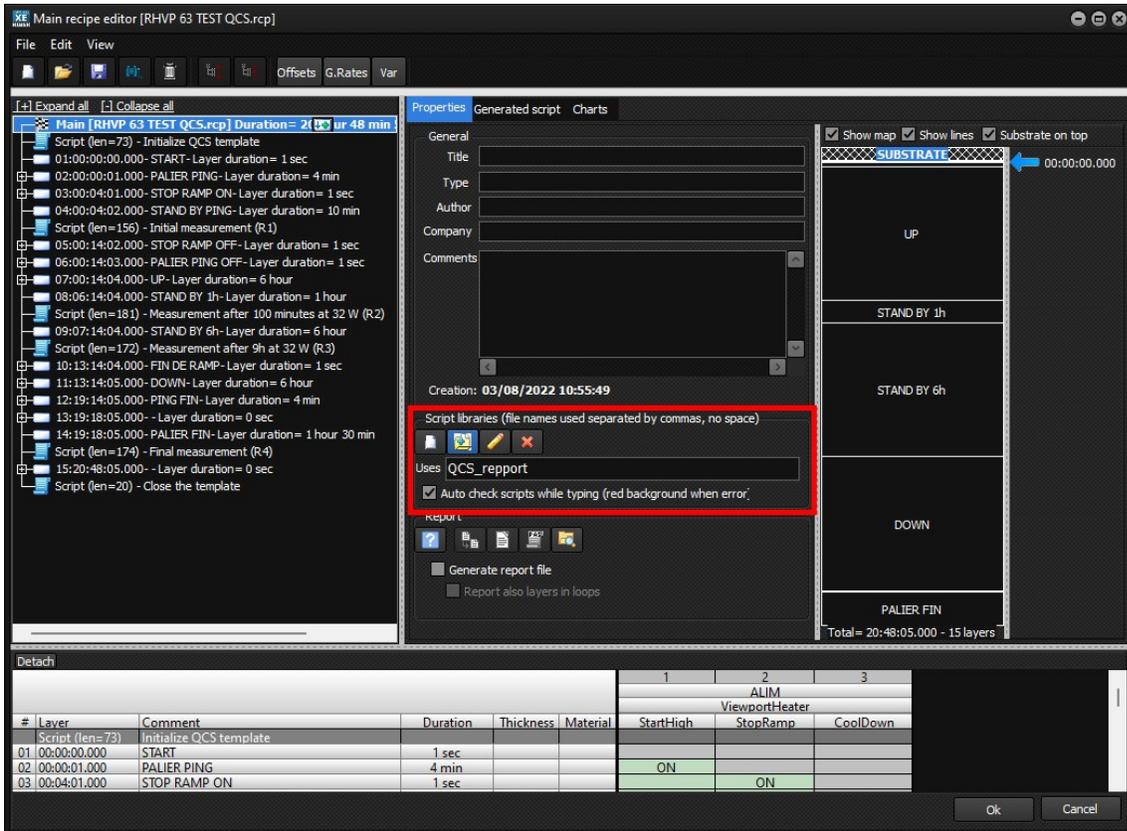
For example, create a script file with the name myLibrary.pas.

To create a new library, in the main menu of Crystal, select New script. Write the following script and save the file as "MyLibrary.pas".



Pascal script editor

Go back to the recipe editor, click on the main item and enter the name of your library:

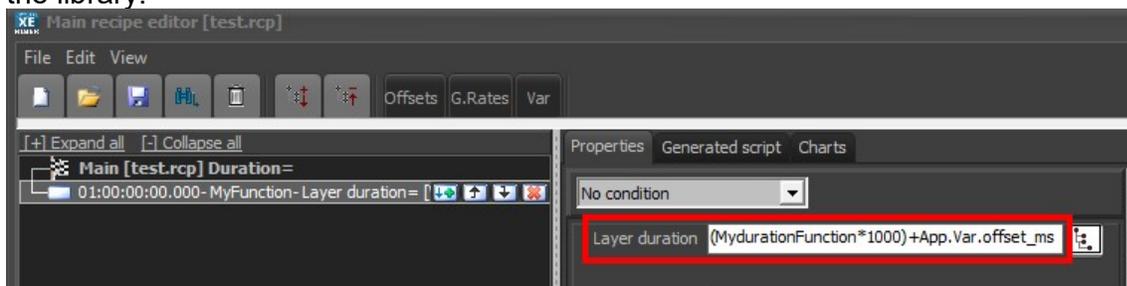


Enter the pascal library file name

Several library files can be used. In this case, each file name must be separated by a comma.

Uncheck the box “Auto check scripts while typing ...” only if the refresh takes too long.

Then, select the current layer and you can add a formula that contains the function that is in the library:

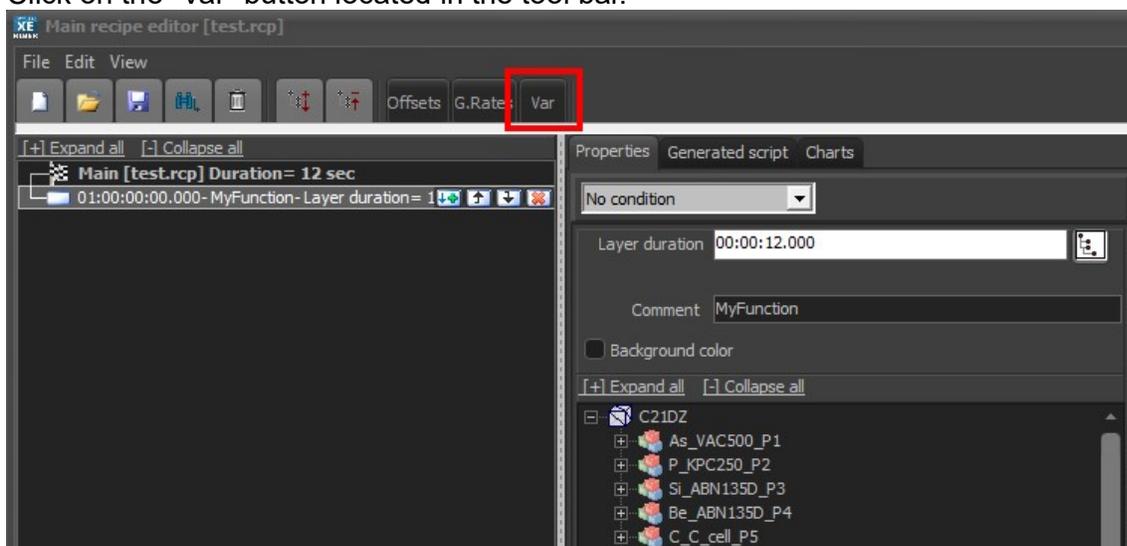


Duration of the layer composed of a formula which uses a script function.

12.13. Use of local variables

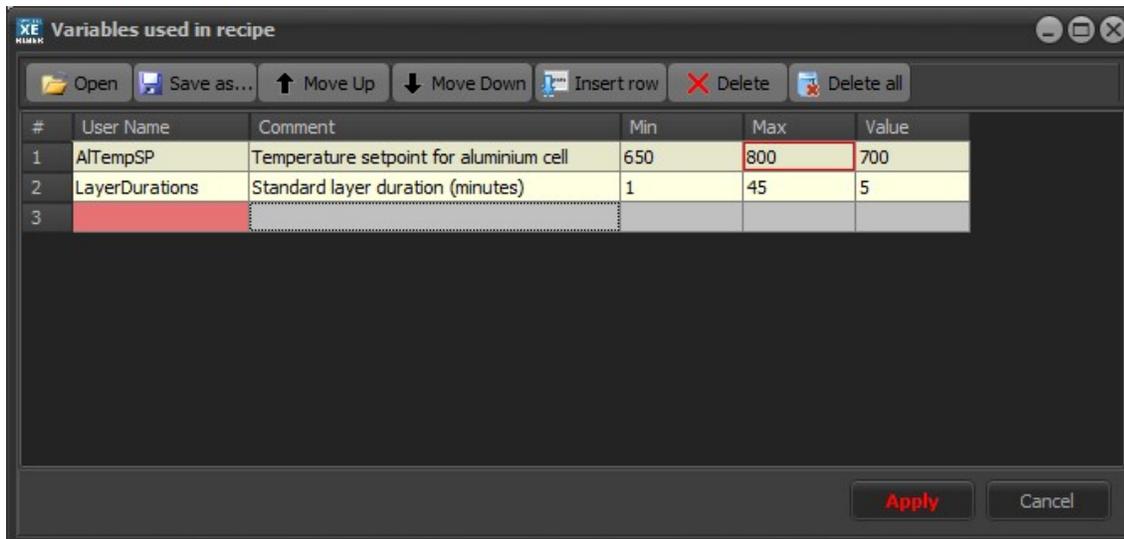
It is possible to use local variables which are only available in the recipe. When the user starts to execute the recipe, Crystal XE will ask to enter the values of the variables that will be used in the recipe.

First, you must create the local variables. Click on the “Var” button located in the tool bar.



Var button to open the local variables editor.

This button will open the local variable editor:



Local variables editor.

You can define as many variables as you want.

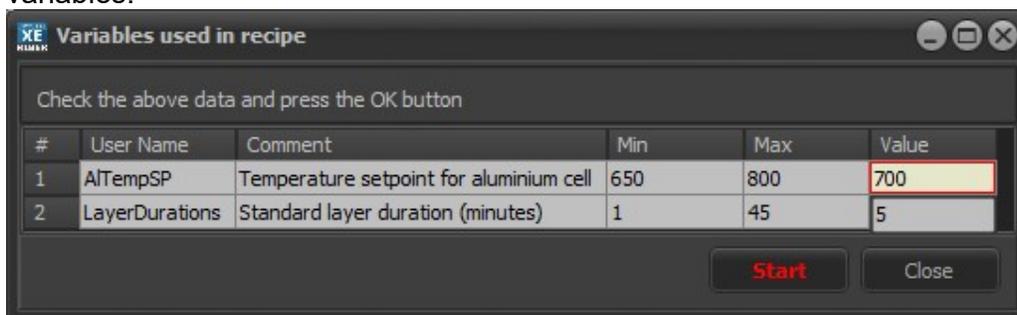
For each variable, you must enter comment, a range in which the user can enter the value and a default value which will be displayed when the recipe is executed.

These variables can be used for the durations of the layers, or for the instructions that are sent to the regulators.



Example of a local variables used in the calculation of a layer duration.

When the user launches the recipe, Crystal will ask him to enter the content of the local variables.

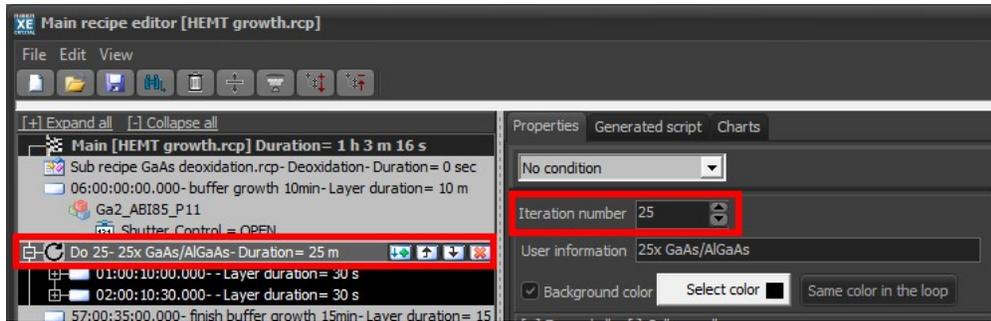


Example of a local variables requested to the user when launching the recipe.

The user can change the value of the variables and push on the start button to start the recipe. The window will remain open during the execution of the recipe to allow the user to modify the variable if necessary. Any changes will be considered instantly.

12.14. Loops

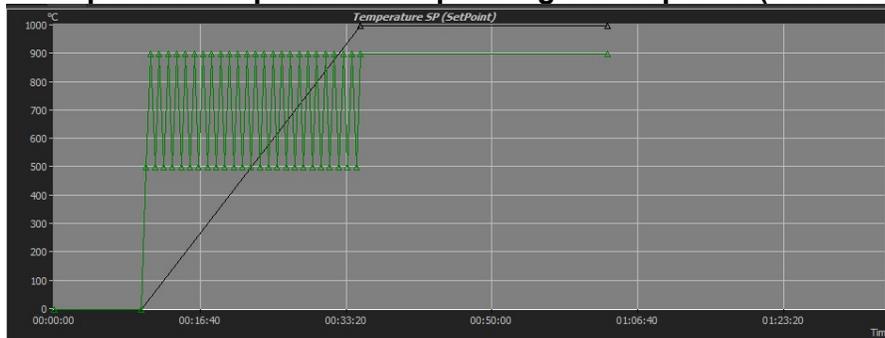
Loops are useful to repeat a list of top-level elements. Several loops can be nested. It is also possible to add a ramp in the beginning of the loop, outside a layer, to have a ramp whose final set point is at the end of the loop.



Example of loop in a recipe.

- Right-click and select Add/loop or Insert/loop.
- Select the Iteration number
- You can add a temperature ramp inside the loop but outside a layer in the case you want to make a ramp while the loop time.

Example of a temperature ramp during the loop time (black curve)



12.15. Conditional layers

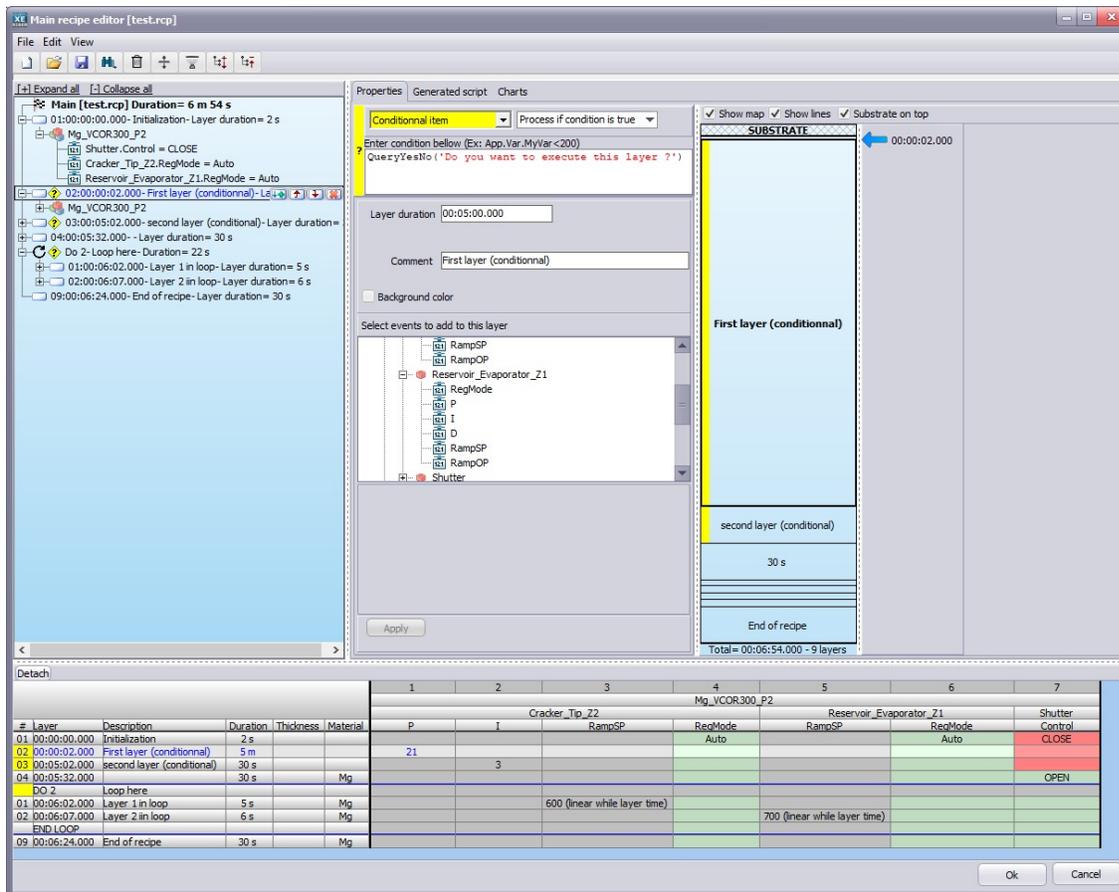
We call this feature "conditional layers" but it also applies to all the steps such as sub-recipes, the loop and the pause.

Since **Crystal XE version 1.06**, it is possible to create conditional layers.

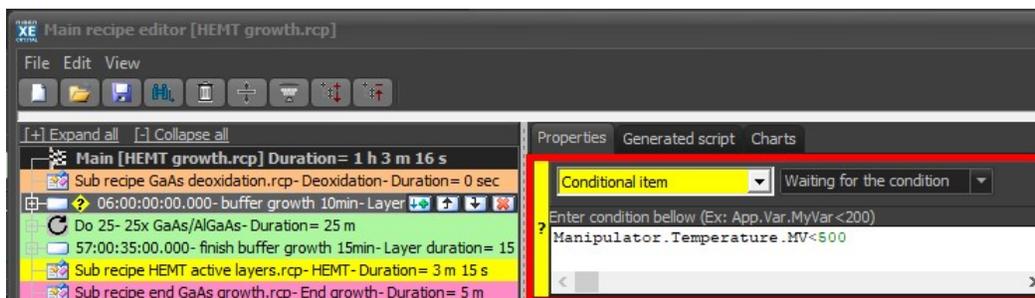
In other words, it is possible to add a condition to execute or not execute a step. This condition can also be set to **wait for an event**.

The condition can be applied on

- **A layer**
- **A loop**
- **A pause**
- **A sub-recipe**



Example of recipe using conditional layers, they are identify by the yellow color.



Turn on the conditional item.

Two modes are possible:

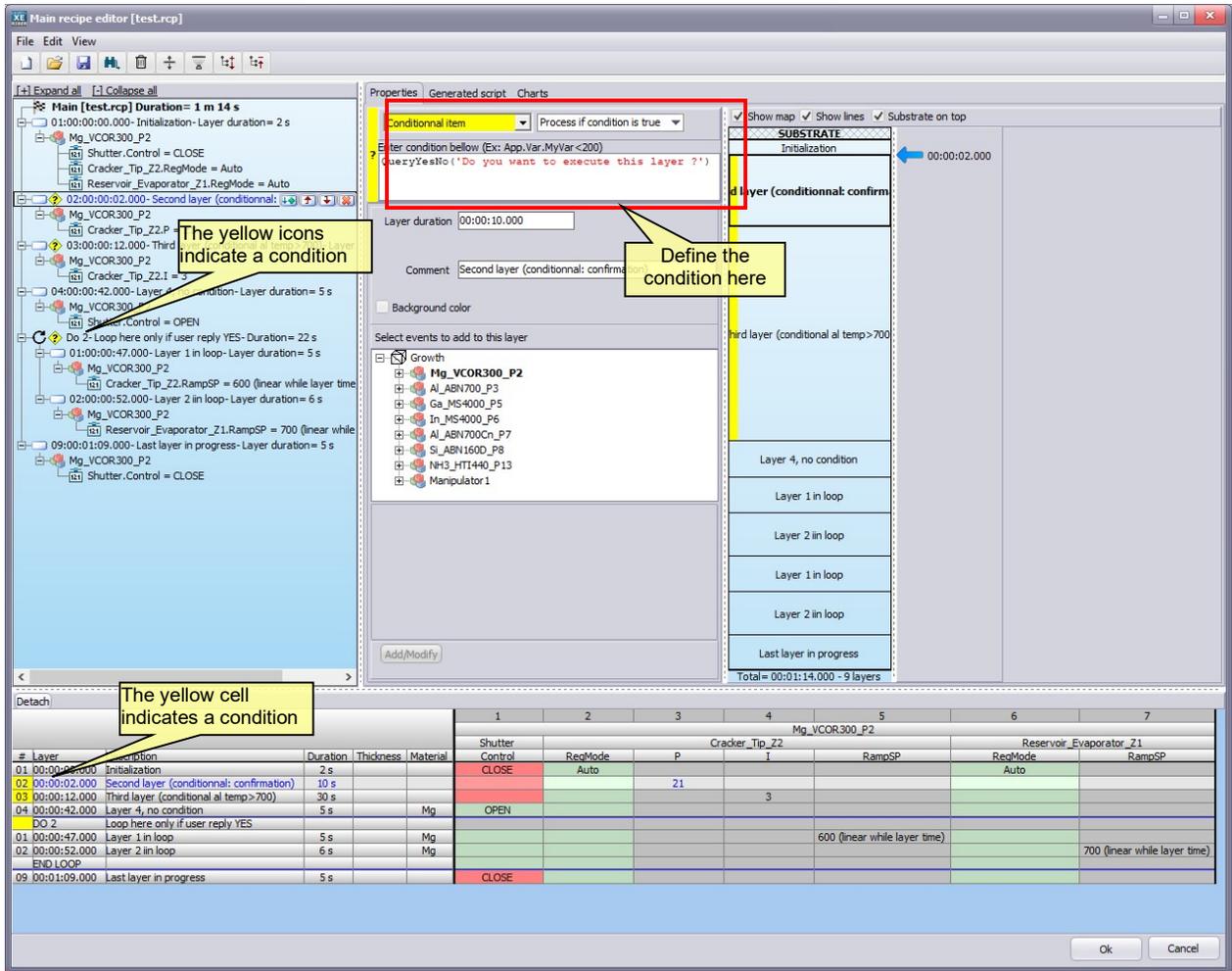
- **“Process if condition is true”** means the step will be executed only if the result of the condition is true (or different of zero) otherwise, the step will be skipped, and the next step will be executed.
- **“Waiting for the condition”** means the step will be executed only once the result of the condition will be true (or different of zero)

In the condition area, right-click and select “Insert from data explorer” to add a tag in the editor.

The condition can also be the result of a script function (A script file must be indicated in the main item).

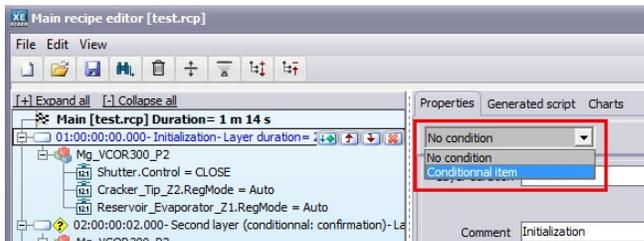
How to enter a conditional event?

Example of conditional recipe:

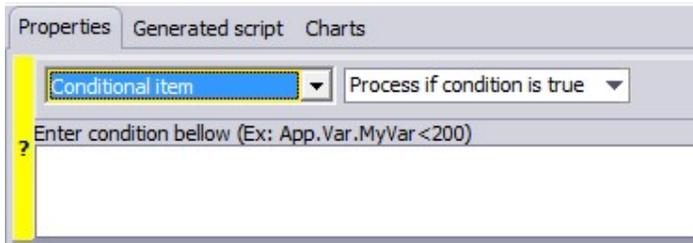


A conditional step (layer, loop, pause or sub-recipe) is identified by a yellow mark. In the tree view, there is a yellow icon, in the grid with the first cell of the line is yellow and in the map view, a vertical yellow margin indicates that the step is a conditional layer.

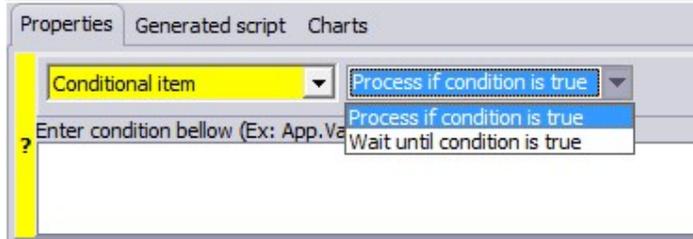
To define a new condition, click on a step and then select “Conditional item” in the combo box - see the red rectangle in the next picture:



When the conditional item is selected, the following options are available:



You can select either “Process if condition is true” or “Wait until condition is true”



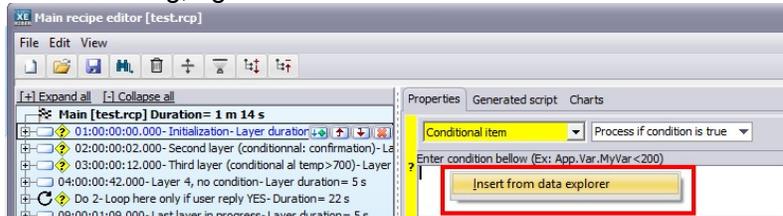
In the condition editor, you can enter any formula or condition that returns zero or a value other than zero.

The condition will be **false** if the condition returns **zero**, otherwise the condition will be **true**.

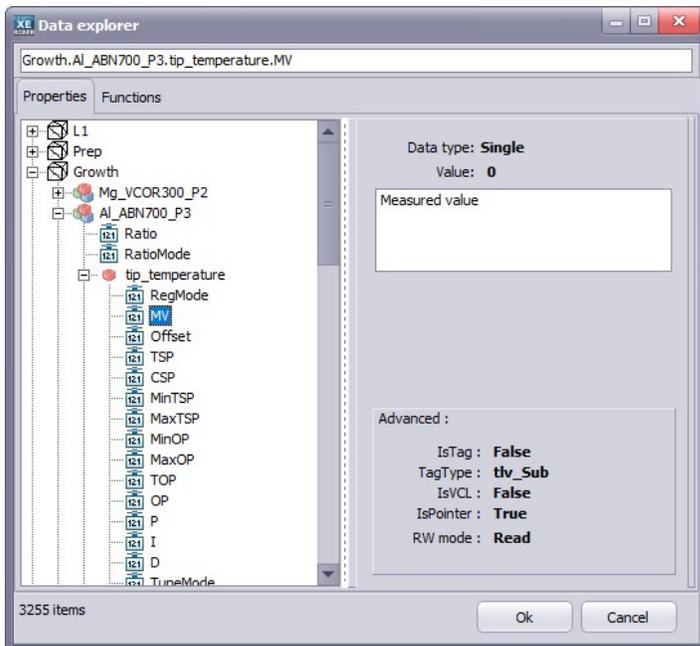
The following conditions are available:

- A simple comparison using standard comparison instructions used in Pascal (see pascal language in the user manual), example: “*GrowthPressure.pressure.MV<1E07*” (The sign “<” can be used to test if the values are different.)

To insert a tag, right click in the condition area and select “Insert from data explorer”

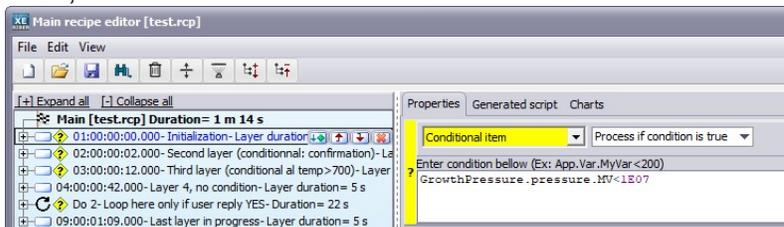


Select the tag in the data explorer

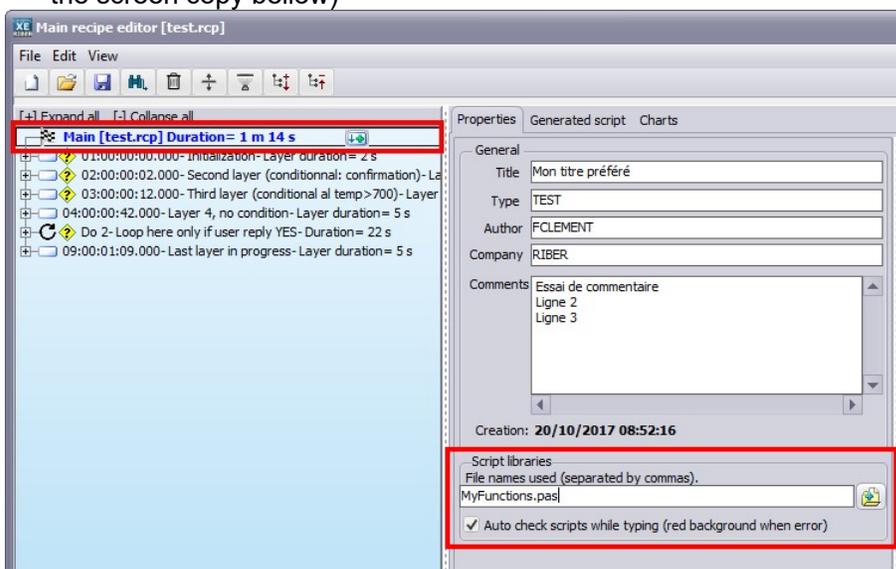


Press OK to insert the selected tag into the condition area.

Then, enter a condition



- You can use the result of a script function like “QueryYesNo(‘message’)”
- The result of a user defined function. In that case you must select the script file to use. Click in the main item of the tree view and select the file names used in the group “Script libraries” (see the screen copy below)



Then you can use all the functions that you have defined in the library.

Execute a conditional recipe

When a condition is true, no change will be displayed in the recipe log.

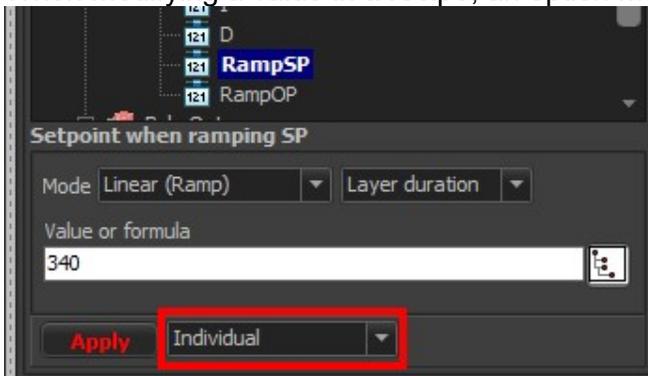
When a condition is false, the following message will be displayed in the recipe log:
"Condition is false, step skipped (followed by the comment of the item)"

Example of recipe log:

Layer	Date / Time	Description
0001	20/10/2017 11:30:36,491	Initialization
0002	20/10/2017 11:30:51,687	Second layer (conditionnal: confirmation)
	20/10/2017 11:31:01,687	Condition is false, step skipped (Third layer (conditional al temp>700))
0003	20/10/2017 11:31:01,687	Layer 4, no condition
	20/10/2017 11:31:10,394	Condition is false, loop skipped Loop here only if user reply YES
0004	20/10/2017 11:31:10,396	Last layer in progress
	20/10/2017 11:31:16,254	-----> Completed

12.16. Modify a value with repercussions in the other layers

When modifying a value in a recipe, an option may appear next to the "Apply" button.



This screenshot shows the options to make a change in all the recipe.

Several options are possibles:



- **Individual:** Only this action will be changed.
- **Same tags & value:** all the tags value of the same equipment and sub-equipment with the same name and same value will be replaced by the new value.
- **Same tags:** all the tags value of the same equipment and sub-equipment with the same name will be replaced by the new value.
- **Same tags by offset:** all the tags value of the same equipment and sub-equipment with the same name will be changed by the same offset. By example if the previous value was 200 and the new value is 245, then all the tag values will be increase of 45.

12.17. Offsets

From the menu, select Edit / Edit offsets or click on the tool bar on Offsets.



This will open the following window:

Equipment	Sub equipment	Offset
As_VAC500_P1	Cracker_Z2	0
	Reservoir_Z1	0
	Condenser_Transfer_Z3	0
	valve_NVC6000	0
P_KPC250_P2	Cracker_Z2	0
	Reservoir_Z1	0
	Condenser_Transfer_Z3	0
	Valve_NVC6000	0
Si_ABN135D_P3	Temperature	0
Be_ABN135D_P4	Temperature	0
C_C_cell_P5	Temperature	0
Al_ABN60DF_P6	Tip	0
	Base	0
Ga1_ABN60DF_P8	Tip	0
	Base	0
C_CBr4_LTI_P9	Temperature	0
In1_ABI85_P10	Insert	0
	Base	0
Ga2_ABI85_P11	Insert	0
	Base	0
Sb_VCor300_P12	Cracker_Z2	0
	Reservoir_Z1	0
	Condenser_Transfer_Z3	0
	valve_NVC6000	0
Manipulator	Temperature	0
BakeOut	temperature	0
Wafer_temperature	TC	0
	Pyro	0
AS_P121	temperature	0

Offsets window

All equipment that has a tag called "Offset" is listed here.

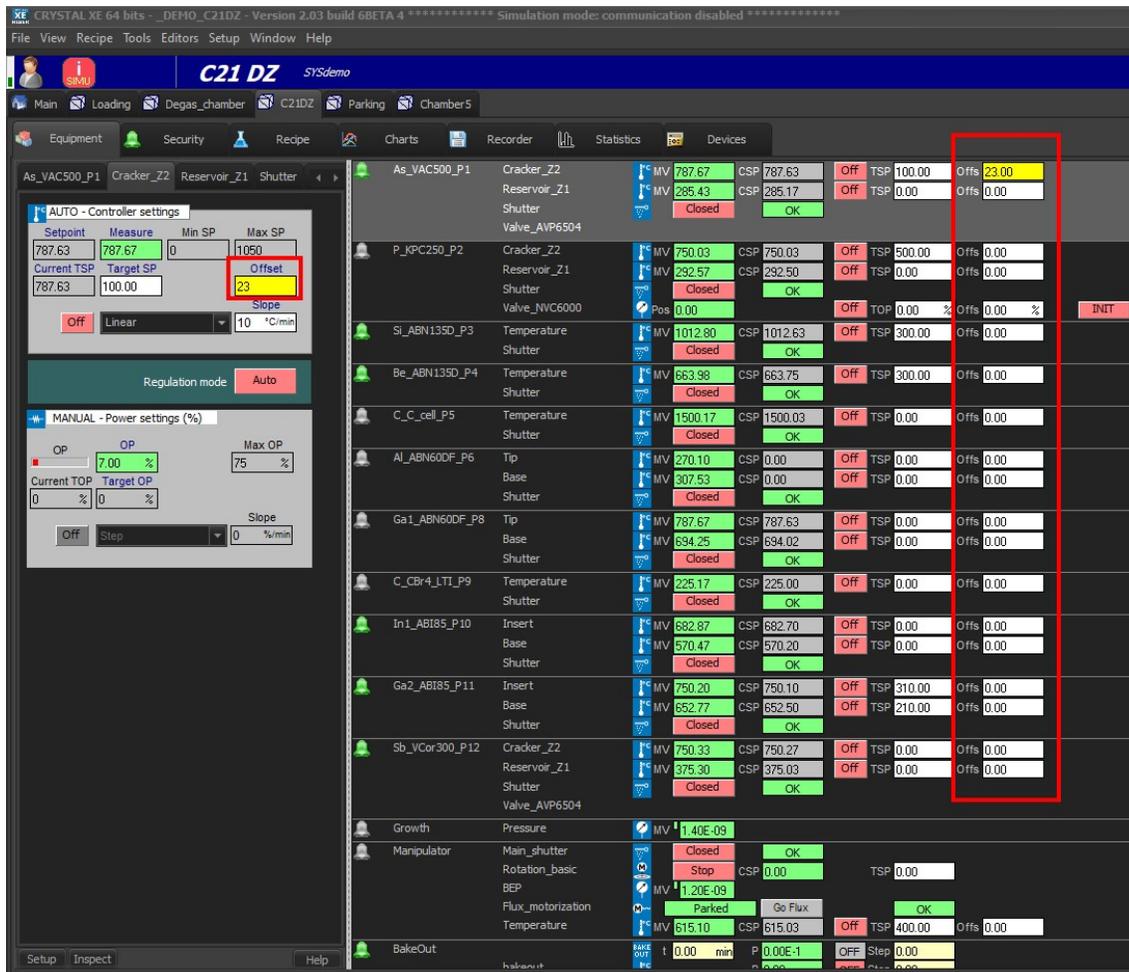
This is useful in the case where the material decreases in a cell and you do not want to modify the recipe. In this case Crystal will compensate for the loss of materials by a rise in temperature of the cell. This elevation results in an offset that you can enter in this table.

Any offset values you enter here are taken into account instantly throughout Crystal XE.

The offset values are not saved in the recipe, so by loading a recipe, the offsets will not be modified a second time.

The offsets are automatically saved by Crystal when closing it.
The offsets are automatically loaded when opening Crystal.

In the equipment view, all the offsets are displayed in one column and also in the detail view.



Offsets in the equipment view

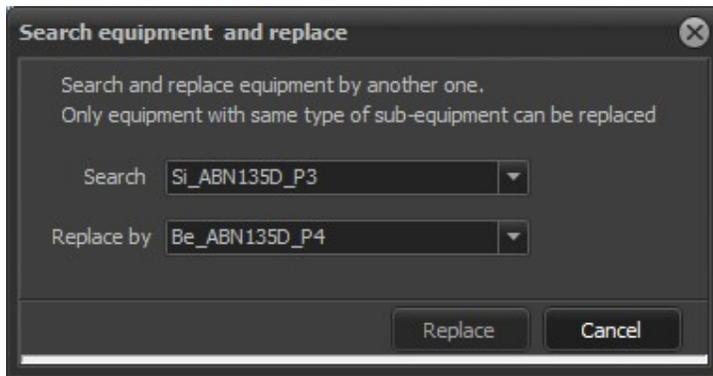
When changing an offset, all the setpoints that are sent to the device will be increased or decreased (if negative value) by the value of the offset. This is valid for step modes but also for ramps, even if they are running.

12.18. Search and replace

From the menu, select Edit / Search and replace, or in the tool bar, click on the button



This will open this window:

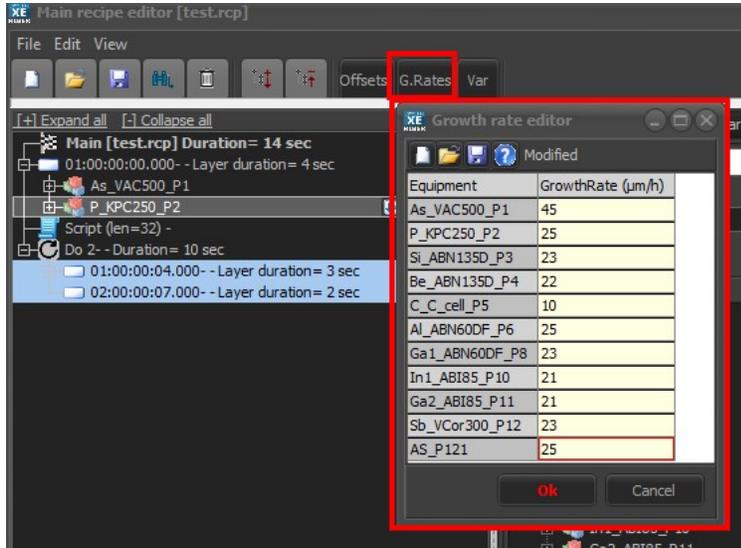


Search and replace window

This tool is useful to replace a cell by another one. For example, when you have several Ga cell in your system and you have written your recipe with the cell Ga1, you can easily change the cell Ga1 by Ga2 using this tool.

12.19. Work in thickness

Rather than entering durations for the layers, you can choose to enter layer thicknesses. In this case you must define in a table the growth rate for each of the cells. In the tool bar of the recipe editor, click on the button “G.Rates”



Click on the G.Rates button to display the growth rate editor.

The growth rate editor allows you to export or import a growth rate file.

The file format is text, and the extension is .ini.

Example of growth file:

```
[GrowthRate]
C21DZ.As_VAC500_P1=45
C21DZ.P_KPC250_P2=25
C21DZ.Si_ABN135D_P3=23
C21DZ.Be_ABN135D_P4=22
C21DZ.C_C_cell_P5=10
```

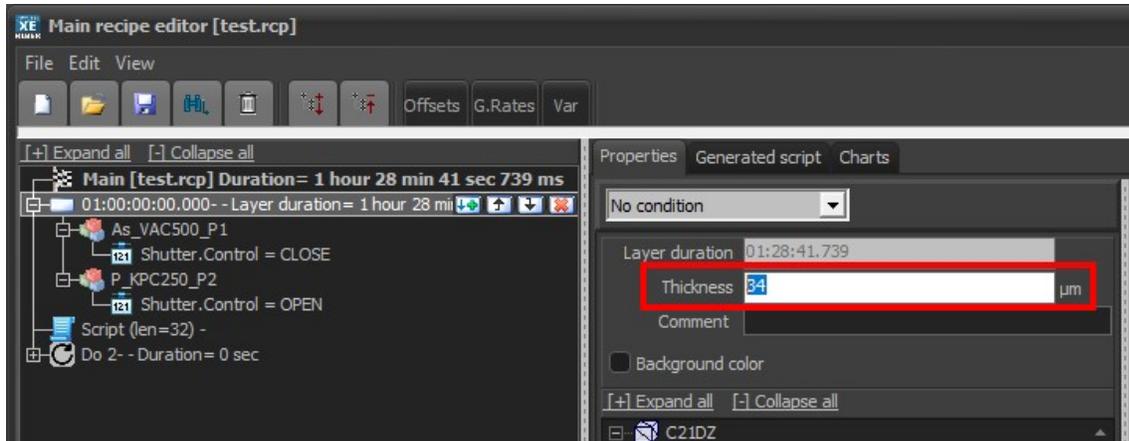
```

C21DZ.A1_ABN60DF_P6=25
C21DZ.Ga1_ABN60DF_P8=23
C21DZ.In1_ABI85_P10=21
C21DZ.Ga2_ABI85_P11=21
C21DZ.Sb_VCor300_P12=23
C21DZ.AS_P121=25

```

Important: The growth rate data are only available in the current recipe. It is saved in the recipe file (.rcp)

Once you have entered growth rates for each cell, you can start entering thicknesses for each layer.



Example to enter a thickness.

Of course, to be able to enter a thickness, the other condition is that at least **one shutter must be open**. In that case the duration of the layer is automatically calculated.

In the resulting script, the layer thickness is a constant duration. So, the growth rate cannot be changed during the execution of the recipe.

If you change the growth rate values, all the durations of the recipe will be automatically recalculated.

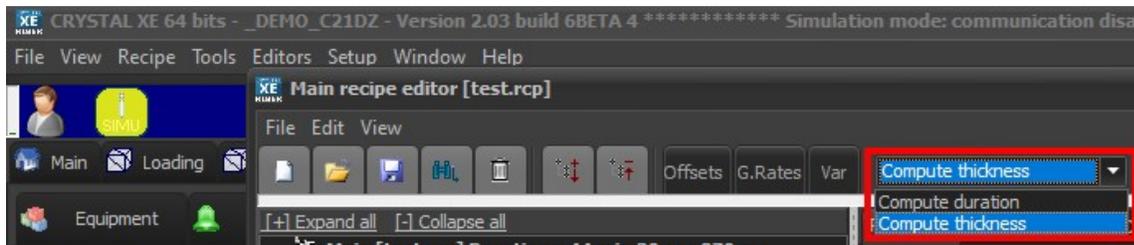
The thickness of all layers is display in the column Thickness in the grid view:

					1	2	3	
					As_VAC500_P1		P_KPC250_P2	
#	Layer	Description	Duration	Thickness	Material	Shutter Control	Shutter Control	Cracker_Z2 RampSP
01	00:00:00.000		44 min 20 sec 870 ms	34,00	P	CLOSE	OPEN	
	Script (len=32)							
	DO 2							
01	00:44:20.870		0 sec		As	OPEN	CLOSE	123 (linear while layer time)
02	00:44:20.870		0 sec		As			
	END LOOP							

Overview of all the layers thickness in the grid view

Compute duration or compute thickness?

You can choose either to enter a thickness and in this case to display a duration for the layer, or to enter a duration and in this case, to display the resulting thickness. To choose which operating mode you want, click on the combo box in the toolbar to choose either "compute duration" or "computer thickness"



Combo box to select either the Compute duration option or the Compute thickness option.

12.20. Script tab

Click on the Script tab to display the recipe's script.

This script is automatically generated by the recipe editor. It is not possible to modify it. If you want to edit a recipe using the Script editor, use this tab to copy the script and paste it in the script editor.

```

Properties  Generated script  Charts
This script is automatically generated and can not be modified  Check  Save script as...  Copy to clipboard
1  Var
2  Rtarget : Real;
3  Begin
4  Manipulator.Rotation_basic.Rotation_SP:=4;
5  Manipulator.Temperature.RampTypeSP:=1;
6  RTarget:=400;
7  Manipulator.Temperature.RampSlopeSP:= abs (RTarget-Manipulator.Temperature.MV) *60/360;
8  Manipulator.Temperature.RampSP:=RTarget;
9  Layer('Start heating GaAs wafer',360000);
10
11  Manipulator.Temperature.RampTypeSP:=1;
12  RTarget:=550;
13  Manipulator.Temperature.RampSlopeSP:= abs (RTarget-Manipulator.Temperature.MV) *60/360;
14  Manipulator.Temperature.RampSP:=RTarget;
15  Manipulator.Main_shutter.Control:=1;
16  As_VAC500_P1.Shutter.Control:=1;
17  As_VAC500_P1.Valve_AVP6504.RampTypeOP:=1;
18  RTarget:=24;
19  As_VAC500_P1.Valve_AVP6504.RampSlopeOP:= abs (RTarget-As_VAC500_P1.Valve_AVP6504.OP) *60/360;
20  As_VAC500_P1.Valve_AVP6504.RampOP:=RTarget;
21  Layer('increase T' under As flux at 25°C/min',360000);
22
23  Manipulator.Temperature.RampTypeSP:=1;
24  RTarget:=680;
25  Manipulator.Temperature.RampSlopeSP:= abs (RTarget-Manipulator.Temperature.MV) *60/780;
26  Manipulator.Temperature.RampSP:=RTarget;
27  As_VAC500_P1.Valve_AVP6504.RampTypeOP:=1;

```

Example of a script automatically generated by the recipe editor.

Click on the Check button to check the script.

Normally, there should never be an error in a script that has been automatically generated. However, this can happen if a global variable which is used in the recipe has been modified or has disappeared or if you made a mistake while entering a script that is used as an element in the recipe, etc.

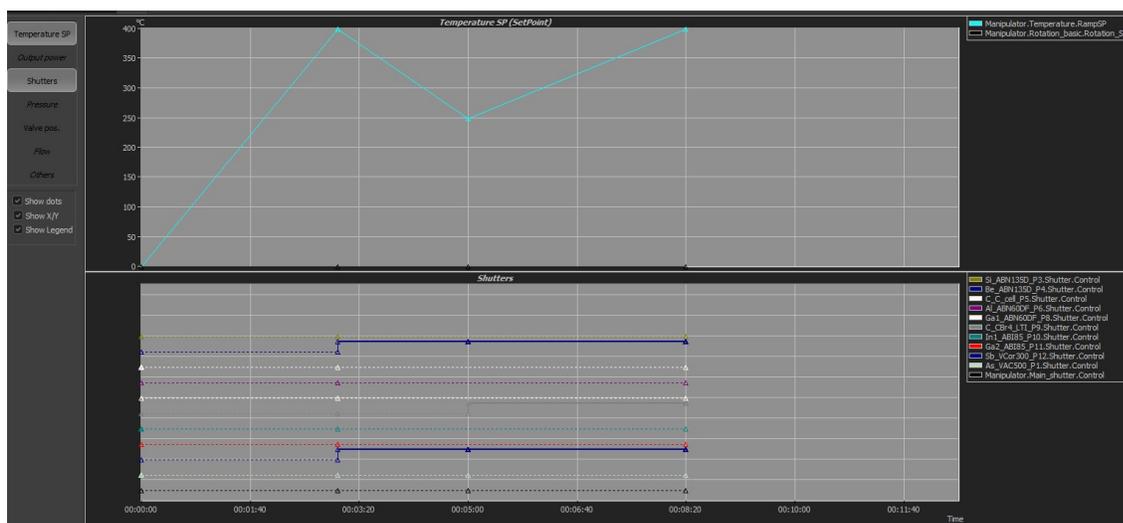
When you save the recipe or when you click the OK button to close the recipe, a check is also systematically made.

If an error occurred, Crystal indicates where the error is in the script.

Click on the button "Save script as..." to export the script into a pascal file. This can be a good start to begin a new script recipe.

12.21. Chart view

The chart view display the evolution of the temperature setpoint, output power, shutters status etc... in function of time and during all the recipe time.



Example of a charts view

This is useful to check if everything is good before to execute the recipe. By right-clicking on a graph, you can export the graphs in different formats, change the display, print, copy to the clipboard, etc.

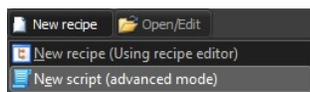
In the left panel, you can select which chart to display.



The use of the italic font indicates that no point is present in this graph.

12.22. Script recipe editor

Click on “*New recipe*” and “*New script (advanced mode)*” located in the top left-hand corner of the recipe interface, to open the *Script editor*.



Use the *Script editor* to create a recipe using Pascal programming language. The created recipe is saved as a `.pas` file.

How to write the layers ?

Remember that a layer is determined by a list of actions to be performed and a time to wait. Then you must therefore write all your actions like for example, to activate a cache, write "Growth.As_VAC500_P1.Shutter.Control:=1;" and terminate by a time to wait, but for the layers, you must use the Layer instruction. For example, write Layer('My first layer',3000); to make a layer with a duration of 3 seconds.

A good solution to start a new recipe in script is to write it using the recipe editor, to export the script which was automatically generated and to edit this script.

⇒ **For more information about the scripts, see the scripts section, later in this manual.**

12.23. Convert recipe CSV to RCP

Crystal XE CSV converter allows to convert recipe files saved as Comma-separated Values (CSV) format, edited with spreadsheet program like Excel, or exported from older version of Crystal software, to compatible Crystal XE .RCP files.

How to create a new CSV file:

To create your recipe from your favorite spreadsheet, it is necessary to create a file template. Launch the Crystal XE recipe editor: Go to the growth chamber tab, then to the Recipe tab and Click on the "New recipe" button. From the recipe editor, go to the File / Create CSV template menu. A template will be created in the Recipe folder of the chamber concerned.

To convert old Crystal Eyes recipes to Crystal XE, here we outline two methods:

- The first one is to export old "Crystal Eyes" recipes to CSV files,
- The second is to use the internal converter that convert directly old .RCP files to Crystal XE .RCP files (see the next chapter).

12.24. Export old "Crystal Eyes" recipes to CSV files

Recipes that have been edited with the **version 7.7.SP9** of Crystal Eyes software must be exported as CSV files prior to be converted to *Crystal XE* recipe files.



You need the version 7.7.SP9 of Crystal Eyes to perform the export. If you have an older version of Crystal Eyes, you must install Crystal 7.7 SP9, all older recipe files are readable from Crystal 7.7SP9.

- 1 From the Crystal Tool bar, launch the Schedule program



2. Open the desired recipe file



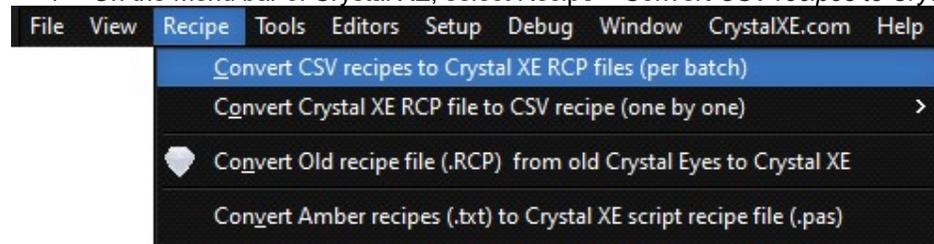
The recipe **must not** include *Sub recipe* or *Pause*. If so, you must delete all *Sub recipes* or *Pauses* prior to convert the recipe.

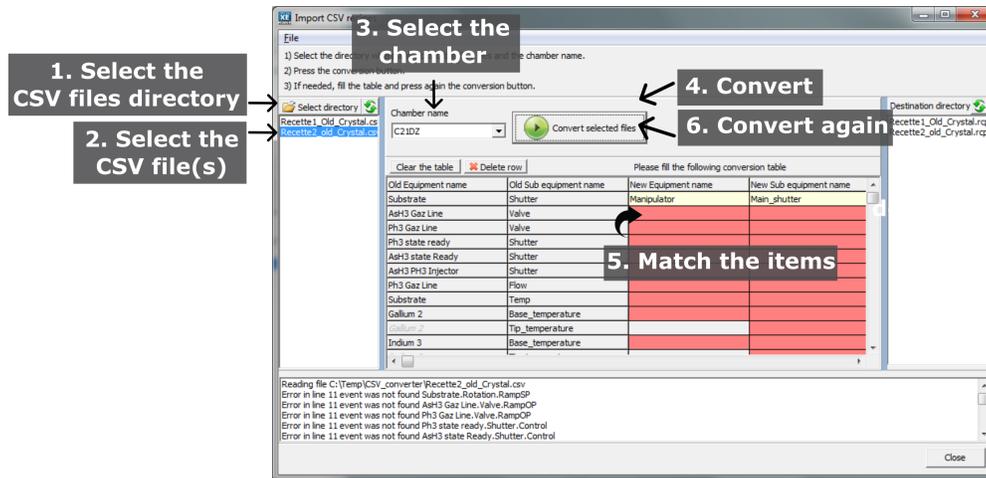
On the menu bar, select File > Export excel file to Crystal converter. The recipe is exported as a CSV file.

3. Create a directory to which you will save all your exported files (recommended).

12.25. Convert CSV files from Crystal XE

1 On the menu bar of Crystal XE, select Recipe > Convert CSV recipes to Crystal XE RCP files (per batch)





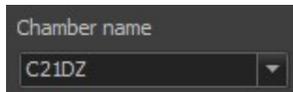
Crystal XE CSV converter

4. Select the directory to which the CSV files have been saved.
5. From the directory file list, select on or several file to convert:

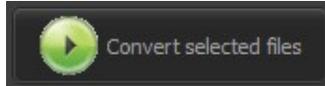


If new CSV files have been added to the directory, click on the following icon to refresh the list.

6. Select the chamber within which the recipe will be executed using the drop-down list:



7. Click on *Convert* selected files:



In the dialog box, select the directory to which the converted files will be saved.

When you first convert the CSV file(s), some equipment and sub equipment **names** (identifiers) of your older version of Crystal may not correspond to *Crystal XE* names.

Equipment and sub equipment with no corresponding names in *Crystal XE* are listed in a **conversion table**:

Clear the table		Delete row		Please fill the following conversion table			
Old Equipment name	Old Sub equipment name	New Equipment name	New Sub equipment name				
Substrate	Rotation						
<i>Substrate</i>	Shutter						
Ash3 Gaz Line	Valve						
Ph3 Gaz Line	Valve						
Ph3 state ready	Shutter						

A **history of conversion** errors is located at the bottom of the window:

```
Reading file C:\Temp\CSV_converter\Recette2_old_Crystal.csv
Error in line 11 event was not found Substrate.Rotation.RampSP
Error in line 11 event was not found Substrate.Shutter.Control
Error in line 11 event was not found Ash3 Gaz Line.Valve.RampOP
Error in line 11 event was not found Ph3 Gaz Line.Valve.RampOP
Error in line 11 event was not found Ph3 state ready.Shutter.Control
Error in line 11 event was not found Ash3 state Ready.Shutter.Control
Error in line 11 event was not found Ash3 PH3 Injector .Shutter.Control
Error in line 11 event was not found Ph3 Gaz Line.Valve.RampOP
```

8. First match the **equipment names**:

For each line, select the name of the equipment as specified in *Crystal XE* from the drop-down menu: For example, the Substrate in the older version correspond to the Manipulator in *Crystal XE*

Old Equipment name	Old Sub equipment name	New Equipment name	New Sub equipment name
Substrate	Rotation	Manipulator	
<i>Substrate</i>	Shutter	<i>Manipulator</i>	

Identical fields are filled automatically.

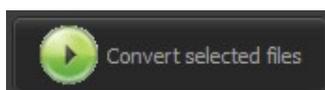
9. Then match the sub equipment names:

For each equipment line, select the corresponding name of the associated sub equipment from the drop-down list:

Old Equipment name	Old Sub equipment name	New Equipment name	New Sub equipment name
Substrate	Shutter	Manipulator	Main shutter

The drop-down list displays the pieces of sub equipment compatible with the selected equipment.

10. Once you have filled the table, click again on the conversion button:



The files are converted to XML files with '.rcp' extension to the specified directory. You can open, edit and execute the converted recipe in *Crystal XE* as any recipe files.



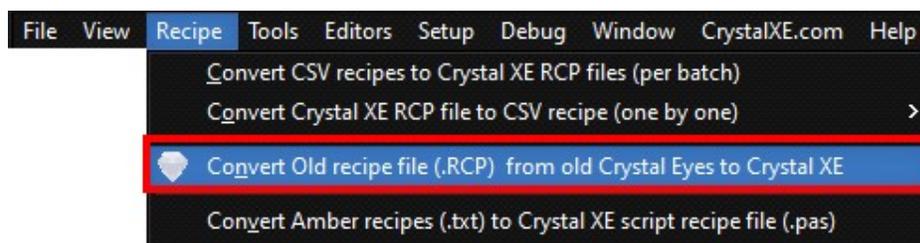
If not all equipment and sub equipment do not match, errors occur. The CSV files can be saved but lines with error are ignored and the tags are not added to the recipe.

CSV File format overview

Section	Description										
Head	Author's name, company etc. Head (all the lines from the beginning of the file up to the line beginning by « ;;; » Each column is separated by a coma <table border="1" data-bbox="695 562 1360 730" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Column #1</th> <th>Column #2</th> </tr> </thead> <tbody> <tr> <td>author</td> <td><Author's name></td> </tr> <tr> <td>company</td> <td><Company's name></td> </tr> <tr> <td>about</td> <td><comments></td> </tr> <tr> <td></td> <td></td> </tr> </tbody> </table>	Column #1	Column #2	author	<Author's name>	company	<Company's name>	about	<comments>		
Column #1	Column #2										
author	<Author's name>										
company	<Company's name>										
about	<comments>										
Blank lines	Unlimited number of blank lines										
Equipment names	A line that describe the name of all equipment used in this recipe										
::	Two empty lines										
Events titles	A line that describe all the events used in the recipe. Each event is relative to the equipment described in the section "Equipment names".										
Layers or loops	A line that describe a layer or the beginning of a loop or the end of a loop.										

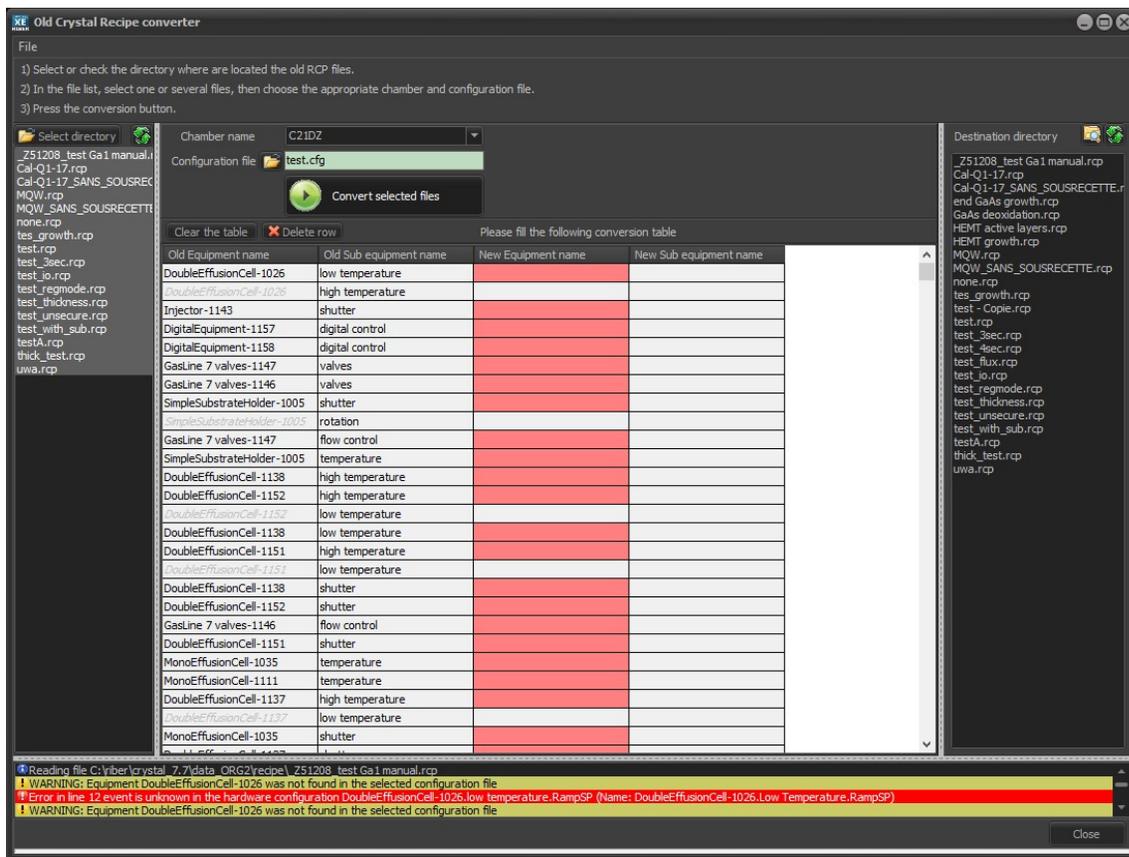
12.26. Convert old Crystal Eyes recipes to Crystal XE recipes

11. On the menu bar, click on *Recipe / Convert Old recipe file (...)*:



To convert old recipe files, you must indicate:

- Select the directory where are located the old recipe files.
- In that directory, select which files to convert.
- Select for which chamber to convert it.
- Select the old configuration file (*.cfg)
- Press "Convert selected files" and indicate where to create the resulting files.



Press the Start button “Convert selected files”

The converter will try to find the correspondences of names of the equipment and sub-equipment that are used by the recipe. If elements are missing or requests the choice of the user, they will be added to the conversion table.

The red cells of the table must be filled in by the user.

Click on each cell and select the most appropriate item.

When all the red cells are filled in, press the Start button again.

All the converted files will be saved in the indicated folder.

From the Destination directory list, double click on a file to open it.

12.27. Convert Amber recipes to Crystal XE script

12.27.1. Preliminary

A special library exists to facilitate recipe conversion.

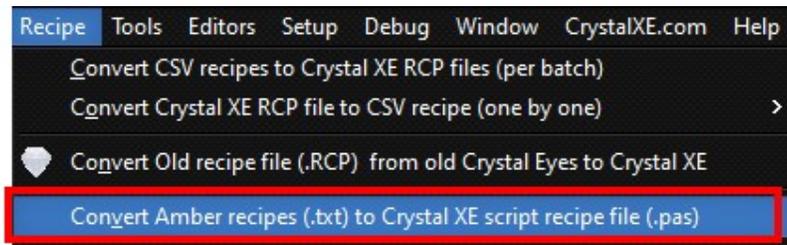
It's called Amberlib.pas and can be found in the template sub directory in the program folder: “c:\riber\CrystalXE\Template\Script”

This library is automatically added to the recipe file when using the CrystalXE recipe converter (the first line of the generated recipe contains “Uses Amberlib ;”).

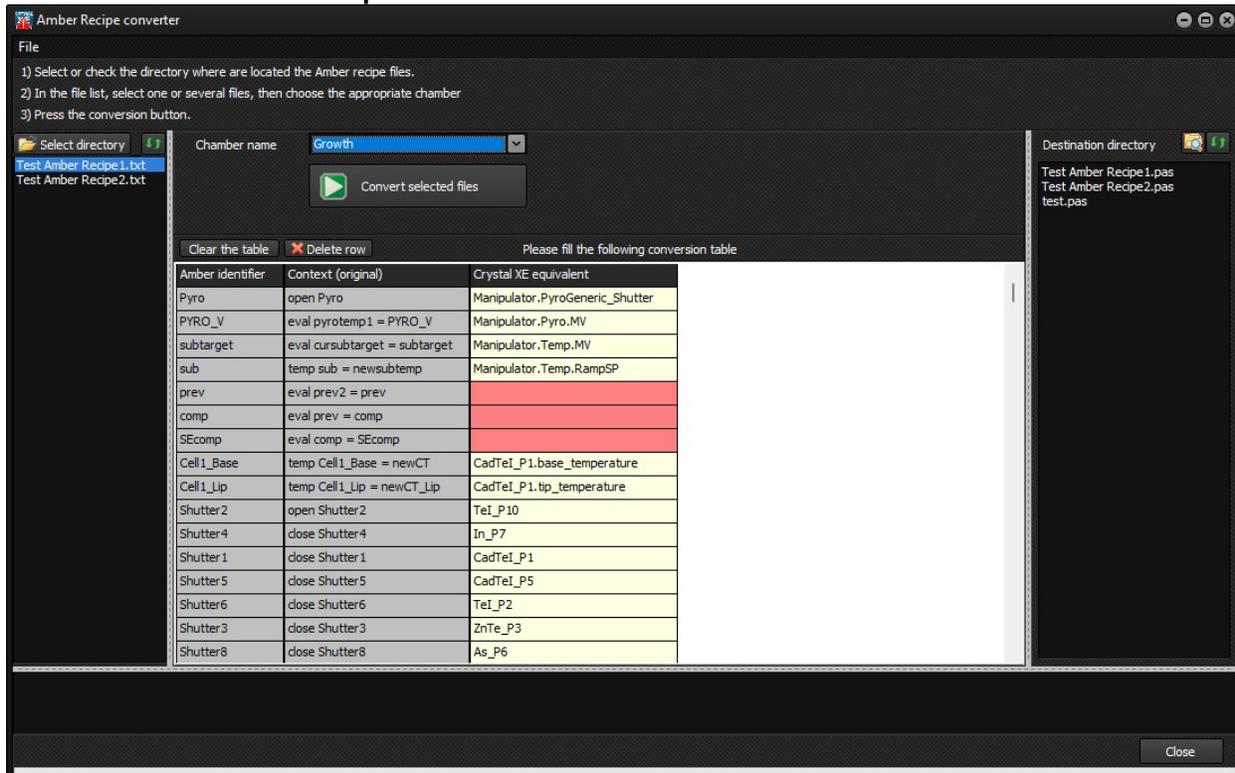
12.27.2. Using the Amber recipe converter

You need at least Crystal XE version 3.20 build 8.

To start the converter, go in the Crystal XE menu “Recipe” and select the Amber converter.



Overview of the Amber recipe converter :



- 1) On the left, click on the “**Select directory**” button to choose the folder in which your Amber recipes are stored.
- 2) The contents of the folder will appear below (only .txt files will appear) in the list.
- 3) **Select the files** you wish to convert (multiple selection is allowed). You can also double-click on a file to start conversion automatically, but you must be sure that the chamber has been selected.
- 4) At the top center, select the relevant chamber (in this example, the chamber **Growth** is selected).
- 5) Click on the button “Convert selected files” to start the conversion.
- 6) You will be asked to choose the folder in which the new Crystal XE recipes will be saved.

In the central table, lines will be created automatically when a word is unknown and requires your intervention.

The left-hand column contains the unknown word. The middle column shows the context in which the word appears in the Amber recipe. You must fill in the right-hand column (click to open the Crystal XE data explorer).

Please note that the level of the equivalent property (equipment or sub equipment or tag name with full path) varies according to the type of word to be converted.

It's worth noting that in Crystal XE, each data can be accessed by following a tree structure in which each element is separated by a dot.

As a recipe is necessarily associated with a chamber, it is not necessary to add the chamber name, so the format for a recipe is as follows: [Equipment name].[Sub-equipment name].[Parameter name]

So, for example:

- To read the manipulator temperature, the path would be as follows: Manipulator.Temp.MV
- To modify the manipulator temperature setpoint, you need to modify the following tag:
Manipulator.Temp.RampSP
- Manipulator.Temp.P to change the proportional band..

.. to display all tags, go in the main menu of Crystal XE: **View / Data explorer**.

Equivalence level table:

Amber function	Crystal XE equivalent
To control the shutter of a cell	Select the cell equipment .
To control the shutter of a pyrometer	Select the pyrometer sub equipment .
To control the shutter of the manipulator (main shutter)	Select the manipulator shutter sub-equipment .
To change a parameter of a loop (setparameter function).	Select the sub equipment

Once you've entered the equivalents, you can repeat the conversion to improve the recipes obtained.

12.27.3. Conversion table

Here's how the converter changes amber instructions into Crystal XE instructions.

The yellow elements will be included in a later version.

Amber instruction	Crystal XE equivalence
title (TEXT)	Title ('TEXT');
comment (MESSAGE)	Comment ('MESSAGE');
!	//
wait [SECONDS]	Wait (SECONDS);
wait [MINUTES:SECONDS]	WaitMS (MINUTES,SECONDS);
wait [HOURS:MINUTES:SECONDS]	WaitHMS (HOURS, MINUTES, SECONDS);
Waituntil [Time of day] <i>Ex: waituntil 6:00 AM</i>	WaitUntil (TimeOfDay);
Waituntil [Time of day, day of the week] <i>Ex: waituntil 6:00 AM, mon</i> <i>With sun, mon, tue, wed, thu, fri, sat</i>	WaitUntilDayWeek (TimeOfDay,DayOfTheWeek); <i>→ script function not defined</i>
waituntil ([lowLimit] < [tempLoop] < [highLimit]) [time in seconds]	WaitUntilRange (lowLimit, TempLoopTagName, HighLimit, time);
waitop (MESSAGE)	WaitOp ('MESSAGE');
structure (NAME)	Procedure NAME; Begin
es	End ;
writetext SE StartAcq(m43037)	Thickness quartz.Ellipso.startACQ:=1;
writevalue [instrument name] [value] <i>ex: writevalue ROT 0</i>	[EquipmentName.SubEquipmentName.tag] := [value]; <i>ex: Manipulator.Rotation.Rotation_SP:=0;</i>
open Pyro close Pyro	Manipulator.PyroGeneric_Shutter.Open; Manipulator.PyroGeneric_Shutter.Close;
eval VarNAME = VALUE_OR_FORMULA	Var VarNAME:real; // declared at the beginning of the script file. note: VarNAME must be defined only once. VarName := VALUE_OR_FORMULA;
step (x)	step (x);
temp [CELL] [SETPOINT]	CellName.sub_equipement. RampSP := SETPOINT;

ex: with constant: temp ga 945 ex:with variable: temp ga=mySetpointVar	
temp [CELL] [SETPOINT] [RampRate]	CellName.sub_equipment. RampSlopeSP := RampRate; CellName.sub_equipment. RampSP := SETPOINT;
ramprate [CELL] [RampRate]	CellName.sub_equipment. RampSlopeSP := RampRate;
writefile ([filename]; arg1, arg2, arg3, ...) columns are separated by tabs	Several possibilities: - WriteFileStrVal(wFname:string;Title:string; wVal:real); - WriteFile2 (filename,val1,val2); - WriteFile3 (filename,val1,val2,val3); - WriteFile4 (filename,val1,val2,val3);
setparameter [PARAMETER] [LOOPNAME]=[VAR] example1: setparameter XP sub 25 example2: setparameter OP Cell5_Lip 8.0	Equipment.sub_equipment.PARAMETER:=VAR The parameter is the tag relative to the sub equipment Parameters equivalence : XP=>P, OP=>TOP Example1: Manipulator.Temp.P:=25; Example2: CadTel_P5.tip_temperature.TOP:=8.0;
Repeat [INTEGER] ... er	For [VAR]:=1 to INTEGER do Begin ... end;

Please note that some functions in this table depend on the equipment name, which is specific to the configuration. It is therefore necessary to adapt it to the names given in your configuration.

About layers:

In Crystal XE recipes, there is a notion of layer that does not exist in Amber recipes.

Amber uses the Wait function, whereas Crystal XE distinguishes between two instructions:

- Layer

- sleep (The "wait" function of the Amberlib file call sleep)

The Amberlib library transforms Amber's Wait instructions into Crystal XE's sleep instructions. There is therefore no Layer instruction in the recipes generated by the converter.

Unlike sleep (or Wait for Amber), the Layer instruction allows you to record a message at each layer and quantify the layers performed.

Like the converter, systematically translates wait instructions into sleep, you are free to manually replace Wait instructions with Layer instructions in Crystal XE recipes to best suit your needs.

Compared with the wait function, the **layer function** requires an additional parameter: **the text associated with the layer**. Please note that the layer duration is expressed in **milliseconds**, whereas the basic unit of the wait function is **seconds**.

Example of table used in the converter (not completed):

Amber identifier	Context (original)	Crystal XE equivalent
Pyro	open Pyro	Manipulator.PyroGeneric_Shutter
PYRO_V	eval pyrotemp1 = PYRO_V	Manipulator.Pyro.MV
subtarget	eval cursubtarget = subtarget	Manipulator.Temp.MV
sub	temp sub = newsubtemp	Manipulator.Temp.RampSP
prev	eval prev2 = prev	
comp	eval prev = comp	
SEcomp	eval comp = SEcomp	
Cell1_Base	temp Cell1_Base = newCT	CadTeI_P1.base_temperature
Cell1_Lip	temp Cell1_Lip = newCT_Lip	CadTeI_P1.tip_temperature
Shutter2	open Shutter2	TeI_P10
Shutter4	close Shutter4	In_P7
Shutter1	close Shutter1	CadTeI_P1
Shutter5	close Shutter5	CadTeI_P5
Shutter6	close Shutter6	TeI_P2
Shutter3	close Shutter3	ZnTe_P3
Shutter8	close Shutter8	As_P6
valve	temp valve 0	
SE	close SE	
Cell4	temp Cell4 200 18	In_P7.temperature
Cell5_Base	temp Cell5_Base 250 15	CadTeI_P5.base_temperature
Cell2_Base	temp Cell2_Base 150 10	TeI_P10.base_temperature
Cell6_Base	temp Cell6_Base 150 10	TeI_P2.base_temperature
Cell3_Base	temp Cell3_Base 200 18	ZnTe_P3.base_temperature
Cell2_Lip	temp Cell2_Lip 170 4	TeI_P10.tip_temperature
Cell3_Lip	temp Cell3_Lip 300 10	ZnTe_P3.tip_temperature
Shutter 7	close Shutter 7	
Cell7	temp Cell7 176.8 10	
Shutter7	open Shutter7	
PYRO_V	eval pyrotemp1 = PYRO_V	
subtarget	eval cursubtarget = subtarget	
prev	eval prev2 = prev	
comp	eval prev = comp	
SEcomp	eval comp = SEcomp	

Cells displayed with a red background indicate the need to enter an equivalent. Otherwise, the converter will insert the original name.

12.27.4. Limitations

Read or Write tags:

The converter does not distinguish between read and write data. For example, the word “sub” can be used in both cases:

- 1) temp sub = newsubtemp
- 2) waituntil (595<sub<605) 60

In the first case, it's a write and the equivalent should be `manipulator.Temp.RampSP := newsubtemp;` and in the second case, the equivalent should be the property `manipulator.Temp.MV` but the converter does not distinguish the difference.

WriteFile

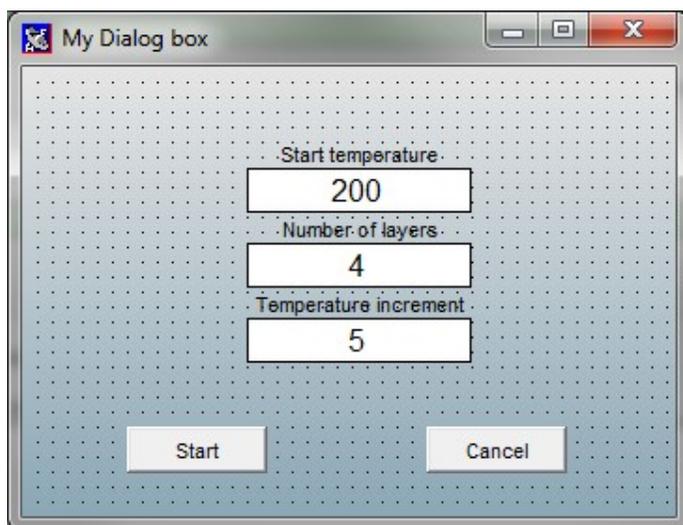
As the writefile function can accept an unlimited number of arguments, the converter does not transcribe all cases.
Only the case where the first argument is a string and the second a value is translated and call the function WriteFileStrVal (this script function is defined in the AmberLib.pas file).

12.28. Use a dialog box (form) in a recipe

12.28.1. Create the dialog box

In the main menu, select File/New/New form.

Create the following form which contain 5 objects as shown above (3 indicators and 2 buttons).



Give the following names to the objects:

(Click on each object and enter the name in the yellow edit box located in the bottom of the tool palette).

- *StartTemp* for the indicator "start temperature"
- *NbLayers* for the indicator "Number of layers"
- *TempInc* for the indicator "Temperature increment"

You can enter a text for the title (in the tool palette, enter a text in the field Window title).

Double click on the Start button and enter the following script:

```
Begin
  SaveDataToIniFile('test.ini'); //Data of all objects will be saved
  CloseModal(1); // Close the dialog box and return 1
End;
```

→ User data of the form will be saved into the file test.ini (default directory is the data directory of the project).

Click once in the window background and enter the following script in the OnCreate event of the form:



```
Begin
  LoadDataFromIniFile('test.ini');
```

```
End;
```

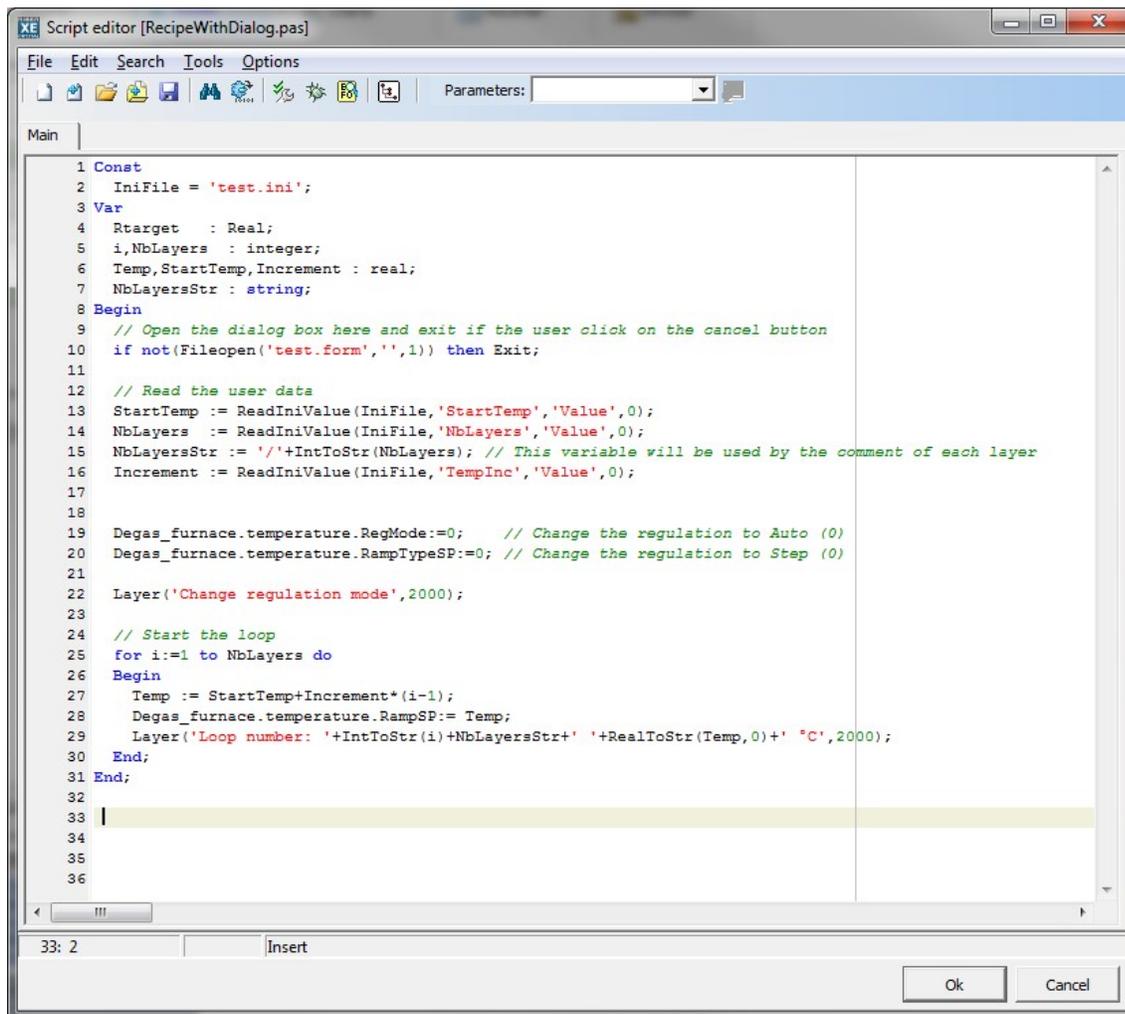
→ The previous data of the form will be loaded when the form will be shown.
Double click on the Cancel button and enter the following script:

```
Begin  
  CloseModal(0); // Close the dialog box and return 0  
End;
```

Exit the Toolpalette and give a name “**test.form**” to save the form to a file.

12.28.2. Create the Recipe

In the recipe tab, just above the recipe inspector, click on the button “New script”



The screenshot shows a 'Script editor [RecipeWithDialog.pas]' window. The code is as follows:

```
1 Const  
2   IniFile = 'test.ini';  
3 Var  
4   Rtarget   : Real;  
5   i,NbLayers : integer;  
6   Temp,StartTemp,Increment : real;  
7   NbLayersStr : string;  
8 Begin  
9   // Open the dialog box here and exit if the user click on the cancel button  
10  if not(Fileopen('test.form','',1)) then Exit;  
11  
12  // Read the user data  
13  StartTemp := ReadIniValue(IniFile,'StartTemp','Value',0);  
14  NbLayers  := ReadIniValue(IniFile,'NbLayers','Value',0);  
15  NbLayersStr := '/' + IntToStr(NbLayers); // This variable will be used by the comment of each layer  
16  Increment  := ReadIniValue(IniFile,'TempInc','Value',0);  
17  
18  
19  Degas_furnace.temperature.RegMode:=0; // Change the regulation to Auto (0)  
20  Degas_furnace.temperature.RampTypeSP:=0; // Change the regulation to Step (0)  
21  
22  Layer('Change regulation mode',2000);  
23  
24  // Start the loop  
25  for i:=1 to NbLayers do  
26  Begin  
27    Temp := StartTemp+Increment*(i-1);  
28    Degas_furnace.temperature.RampSP:= Temp;  
29    Layer('Loop number: '+IntToStr(i)+NbLayersStr+' '+RealToStr(Temp,0)+' °C',2000);  
30  End;  
31 End;  
32  
33 |  
34  
35  
36
```



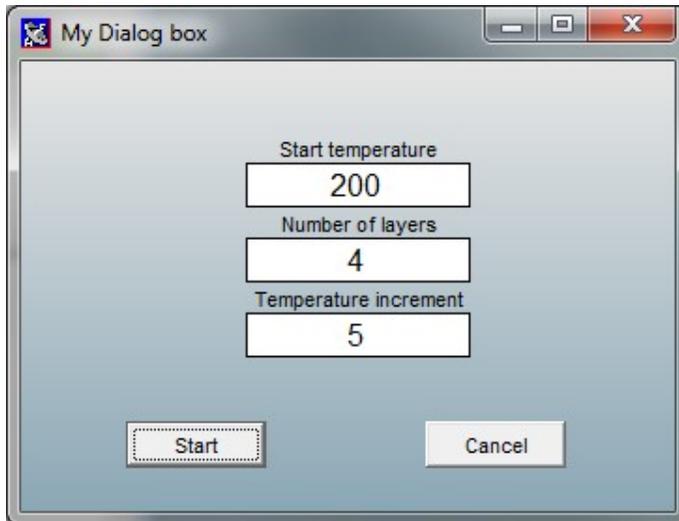
C:\riber\

Source is available here: RecipeWithDialog.pas

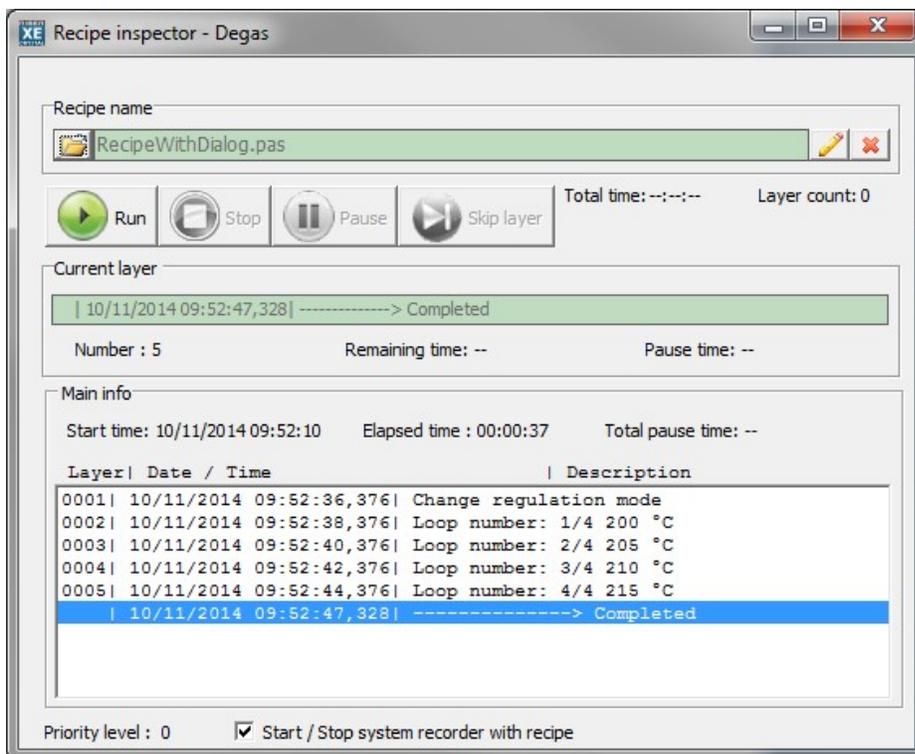
Save it to a file with the name of your choice.

12.28.3. Run the recipe

- Open the script recipe into the recipe inspector.
- Press the Run button



Press Start



12.29. Automatic report generation

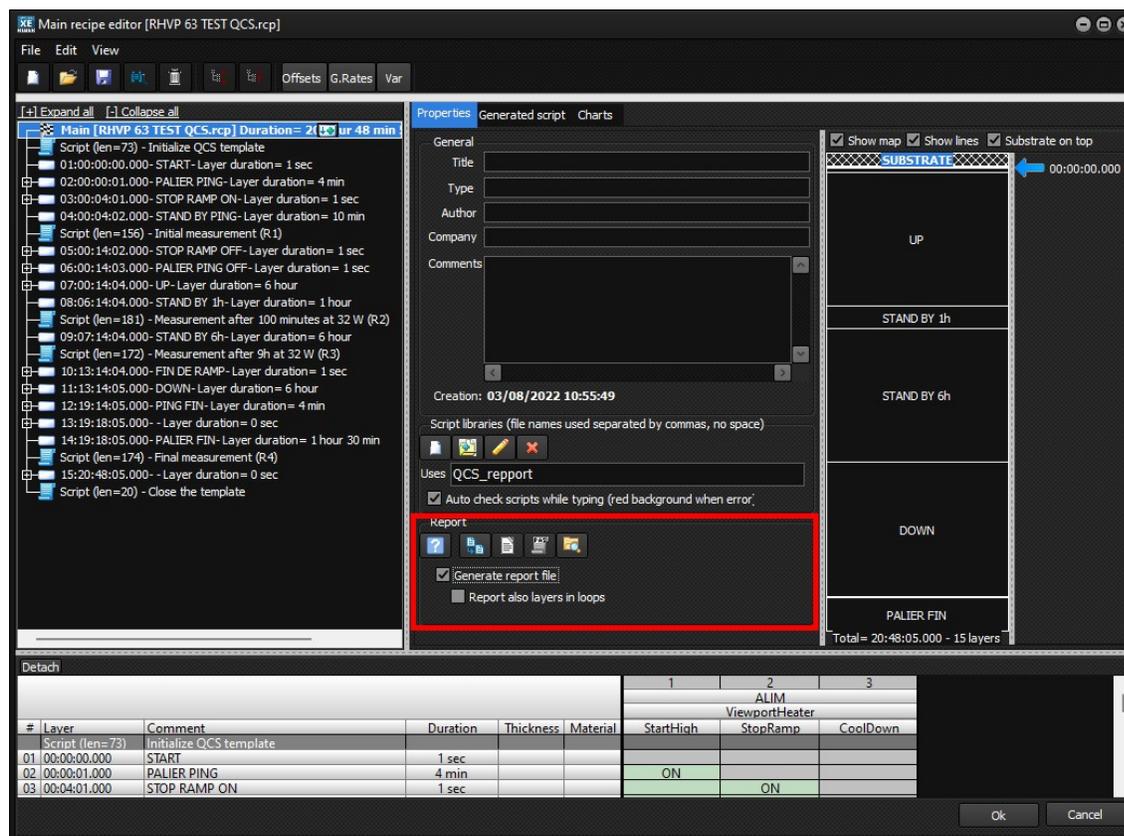
Select the first item "Main" in the tree view.

12.29.1. Default automatic report

Crystal XE allows the automatic generation of reports following a predefined template.

You must first install Libre Office on your computer, then, to activate the report generation, check the box "Generate report file".

IMPORTANT: *If the libre office application stops responding while the recipe is running, the recipe may be blocked. It is therefore not recommended to use report generation for very critical recipes.*



This option will use by default the files:

- LO_WriterReport.odt : this is the document template
- LO_WriterReport.pas: this is the script library

Do not modify directly these files as the changes may be lost in a future update of Crystal XE. To customize them, you must copy them to your project folder, see the next section for more details.

Report files will be automatically created based on the template document but with a new file name. The file name will be the same as the data CSV file created by the recorder but the extension is .DOC to be readable by MSWord.

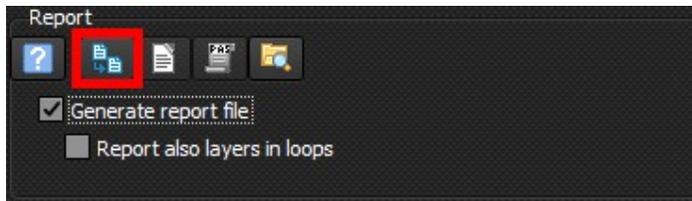
Example of file name : **2022-09-05_14-39-48_Bx-P1-C0-C21DZ-HEMT growth.doc**

In the location : **C:\riber\CrystalXE\DEMO_C21DZ\Record\2022\09**

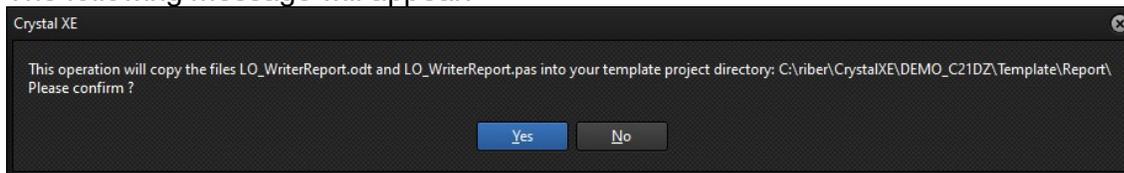
IMPORTANT: The document remains open as long as the recipe is running. Be sure not to close it until the recipe is finished to avoid errors.

12.29.2. Customize the automatic report

If you want to modify the default report or create a new one at your convenience, then click on the following button:

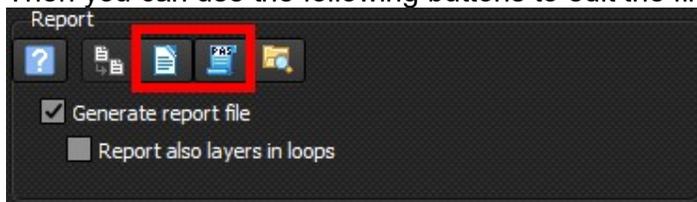


The following message will appear:



After confirmation, this action copies the LO_WriterReport.odt and LO_WriterReport.pas files into the subfolder \Template\Report\ of your project.

Then you can use the following buttons to edit the files:



As these files will be duplicated in both the program folder and the project folder, then priority will be given to the project folder.

Document LO_WriterReport.odt

- This template is loaded in the procedure “**Report_Init**” of the library file (LO_WriterReport.pas) when you check the box “Generate report file”.
- If you want to insert data from Crystal XE into this document then you must insert bookmarks. You can manage all bookmarks in the document by the menu “Insert/Bookmark...” and call the procedure App.OLE.Poke to insert data at the bookmark places.
- You can also add table in the document, add row in that table from Crystal XE, fill the cells etc...You can address the tables by their position number in the document. For more information, refer to the script manual, in the section OLE.

Library LO_WriterReport.pas

This library is automatically used by the recipe when you check the box “Generate report file”

The recipe will automatically call several functions that must be present in this library.

The mandatory functions are the following:

- **Procedure Report_Init(sChamberName:String);**

Report_Init is called only once, at the beginning of the recipe, the chamber name of the recipe is passed as a parameter.

- **Procedure Report_Layer(aLayerComment:String;aLayerTimeMS:integer);**

Report_Layer is called at each layer, before the waiting time.

Parameters are:

- aLayerComment: The comment of the layer that was defined when the recipe was written.
- aLayerTimeMS : the duration of the layer in milliseconds

- **Procedure Report_End;**

Report_end is called at the end of the recipe. This is the last thing the recipe does before it stops.

13. FORMS AND OBJECTS

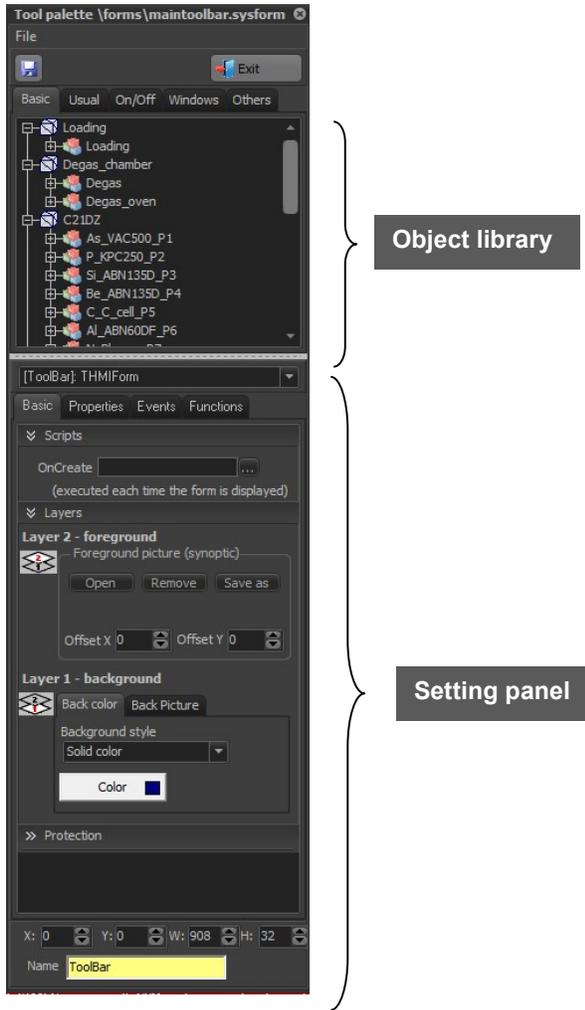
13.1. INTRODUCTION

Crystal XE is provided with a library of customizable **objects**, allowing the user to display information or to perform an action.

In *Crystal XE*, **forms** are windows on which you can place one or several objects to create customized graphic interfaces. Forms offer wide range of possibilities. For instance, you can create a form in order to perform a specific task only (monitoring equipment, controlling a recipe, etc.), or conversely design a synoptic view of your system.

You can customize the appearance of each form or object and interact with it by setting its properties and events in the *Tool palette*.

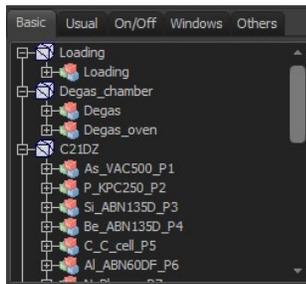
The *Tool palette* is the form and object designing interface. When editing a form, the *Tool palette* window appears on the left side and can be moved all over the screen.



Tool palette overview

13.2. Object library

All the objects you can place on a form are listed in a library, located in the upper part of the *Tool palette*:



The object library is divided into five tabs:

Tab	Types of object
Basic	Equipment type (equipment and sub equipment)
Usual	Label Indicator Group box Chart Script
On/Off	Confirmation button Light
Windows	Standard button Memo Listbox Checkbox Radio button Combo box Progress bar
Other	Image Animated image Rotary button Lines and pipes Cursor Window splitter Video camera Cassette

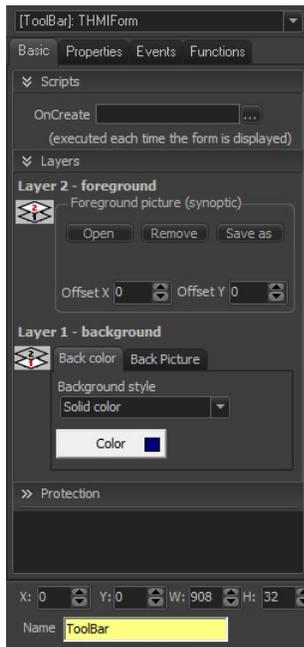
→ For more details about the different objects, please refer to the section **Type of objects** in this chapter.

You can easily add objects to a form using drag and drop.

Each object has defined type and set of properties and events that can be modified using the settings panel in the lower part of the *Tool palette*.

13.3. Settings

When a form or one of the objects placed on that form is selected, its corresponding settings appear in the *Tool palette*:



Form settings panel

The available settings depend on the type of object.

The settings panel is divided into two tabs:

- **Basic settings:**
 - customizing the text and the background style,
 - defining the values to be displayed (Property menu),
 - specifying an event (Scripts menu),
 - defining the form/object name,
 - adjusting the position and size of the form/object,
 - other available options depending on the object type.

- **Advanced settings:**

Accessing all the form or the object properties and events.

13.4. FORMS

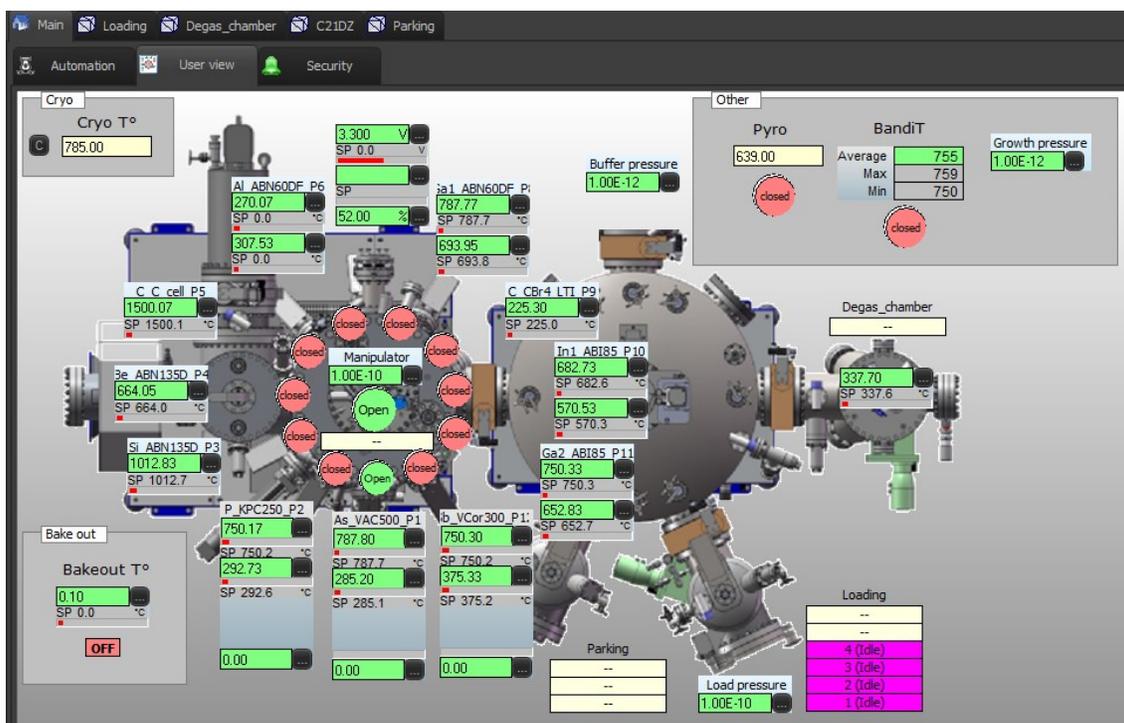
In *Crystal XE*, a form is an object (window) that contains other objects.

Forms allow the user to display or monitor information and interact with one or several objects throughout the program.

By default, the forms already created in the system are:

- **The main *User view***

On the *Main* tab, the *User view* is the main form (synoptic view) already created in the program. You can edit the *User view* and detach the form window.



- **The tool bar**

The Tool bar, located at the top of the interface, is also a form you can edit and place objects on (only in the blue area)

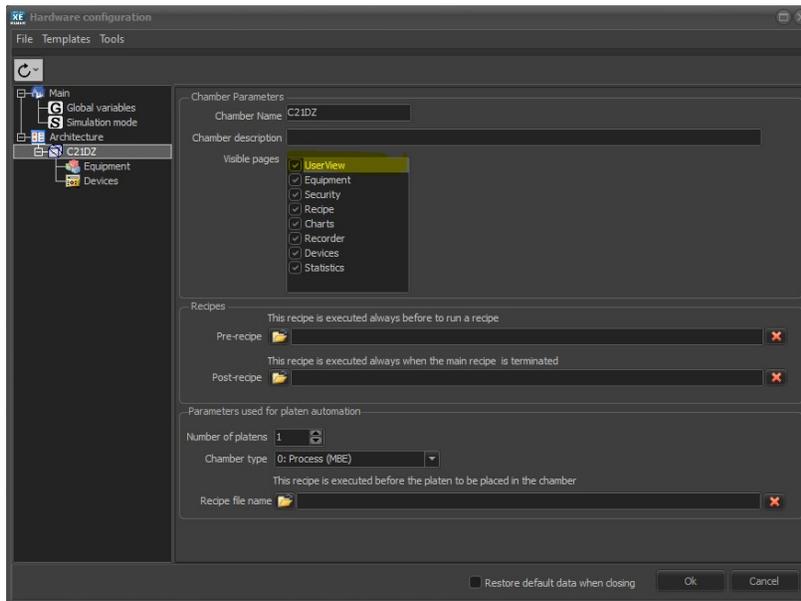


- **Chamber User view**

Crystal XE allows you to design a *User view* for each chamber, accessible with the *User view* tab.

To display the *User view* tab:

- On the *Setup* menu (menu bar), click on *Hardware configuration*
- Select the desired chamber in the tree structure and check the *User view* box (Visible pages)

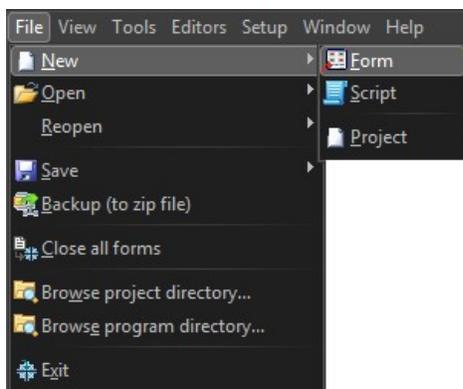


Crystal XE also enables you to create your own **separated forms** (windows). The following sections describe the creation of forms and objects using the *Tool palette*.

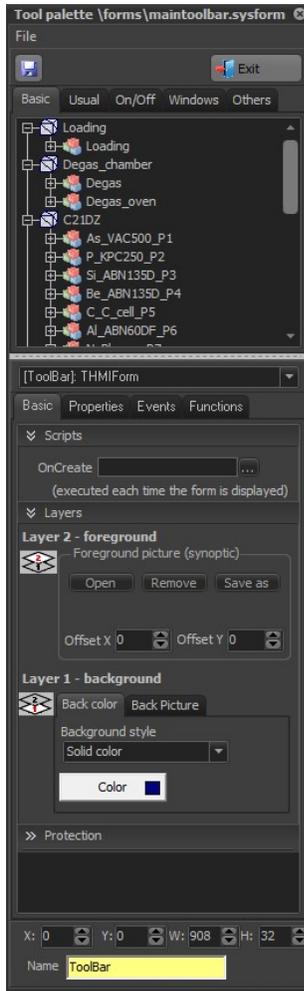
13.4.1. CREATING FORMS

Creating a new form

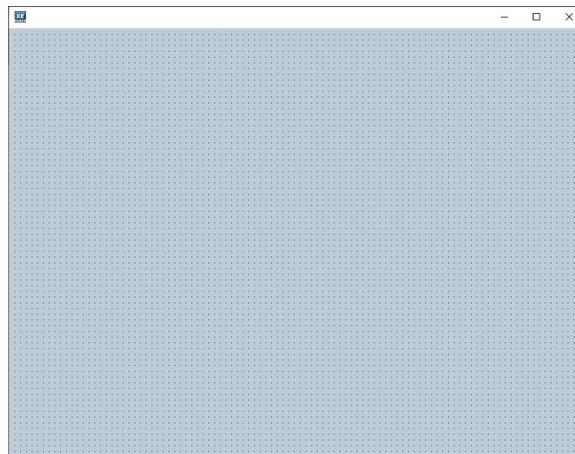
To create a new form, click on the *File* menu (menu bar) and select *New > Form*.



A new default form is created automatically and opened in the form editor (*Tool palette*).

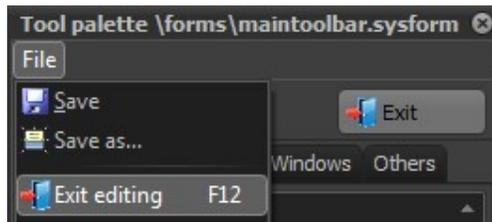


Tool palette



Default new form

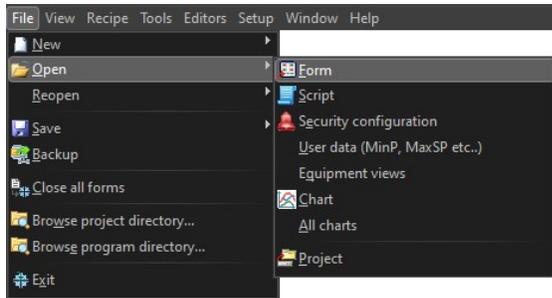
- To save a new form, click on the following icon  in the top-hand corner of the *Tool palette* or click on *File > Save as*.



The form is saved as a *.form* file to the Forms folder located in your project directory.

Opening a form

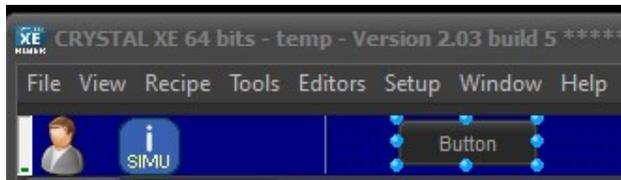
To open a form, click on the *File* menu (menu bar), select *Open > Open form* and then select the desired form file.



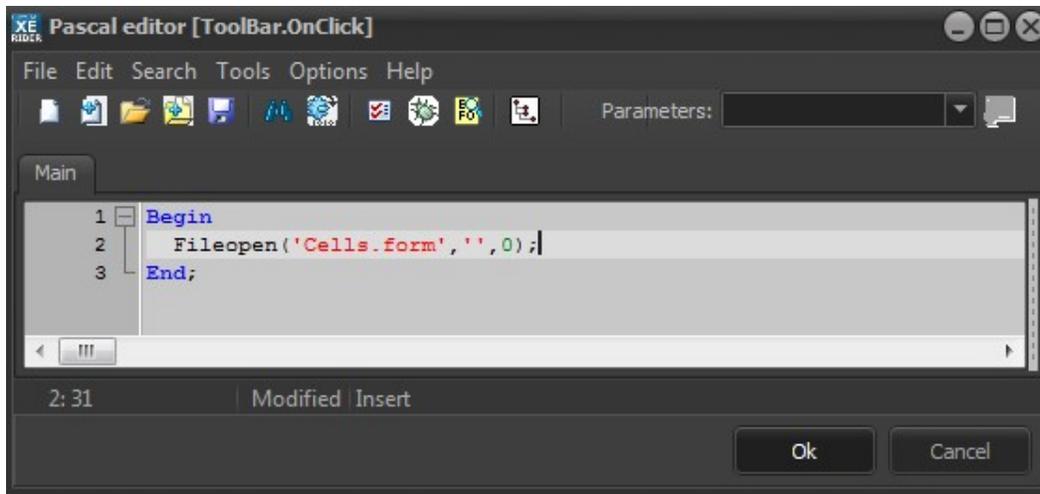
You can also open a form using a script and the function *FileOpen*.

Example: Creating a button that opens a specific form file.

- Edit the toolbar (right click on it and select “Edit the form”)
- Add a button to the *Toolbar* (drag an object “Button” from the tool palette in the Windows category)



- Double click on the button to open the script editor. This script will be executed when the button will be clicked (*OnClick event*). In the following example, the script opens the form file named ‘*Cells.form*’:

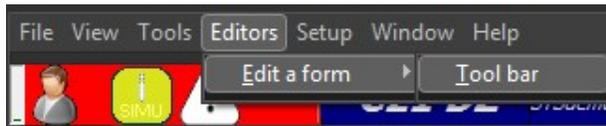


Press F1 key when the cursor is in the word “Fileopen” to open the relative help on this topic.

- Close the script editor by pressing the OK button.
- Terminate the edition of the toolbar by clicking on the “Exit” button of the Tool Palette.

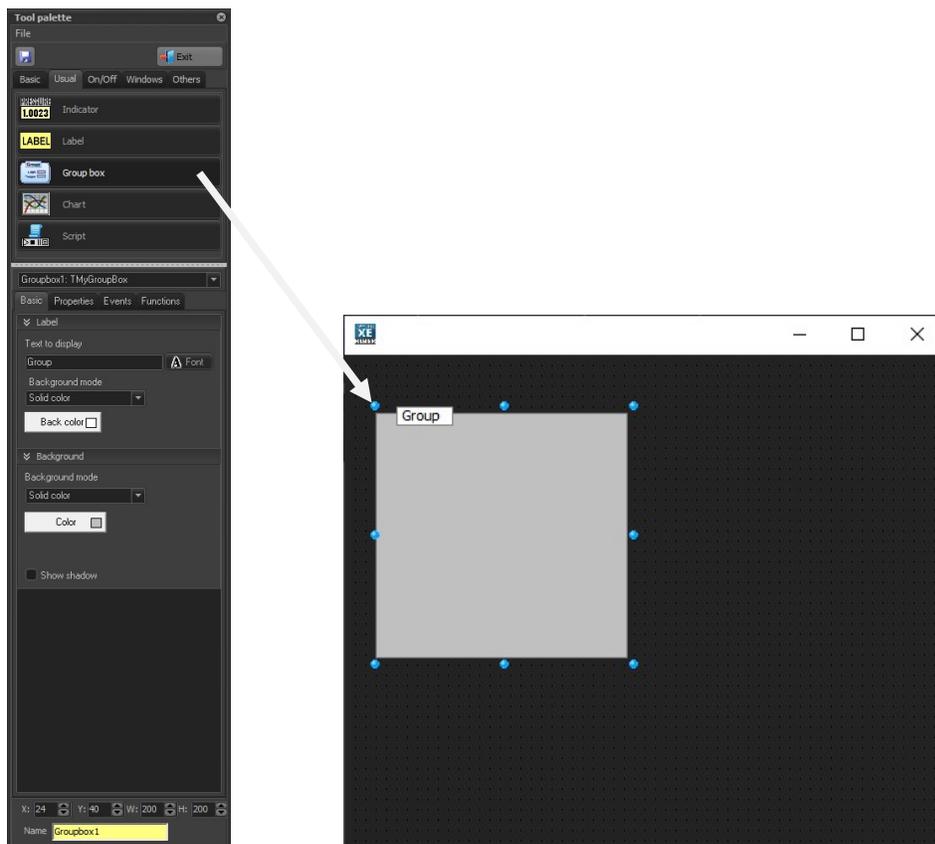
Editing forms

Only active (open) forms can be edited. First, open the form to be edited. Right-click on the form background and select *Edit the form* to open the *Tool palette*.



Adding or removing objects

- To add an object to the form, select the desired object from the object library in the *Tool palette* and place it on the form using drag and drop.



The object is placed on the form with the default properties. When editing, select it to display its settings in the *Tool palette*.

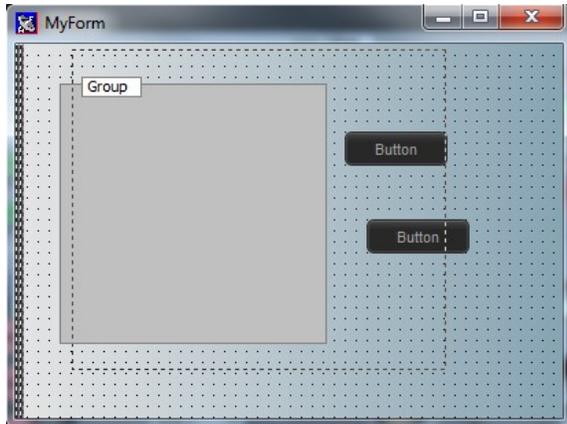
- To remove an object from a form, select it and press the [Del] key or right click on it and select *Delete*. You can also delete an object using the *Object list editor* that can be accessed by right-clicking the form background.

Selecting objects

Click an object to select it. You can also select an object using the *Object list editor* that can be accessed by right-clicking the form background.

To select multiple objects, click on the first item, press and hold down the [Shift] key, and then click on each other item you want to select.

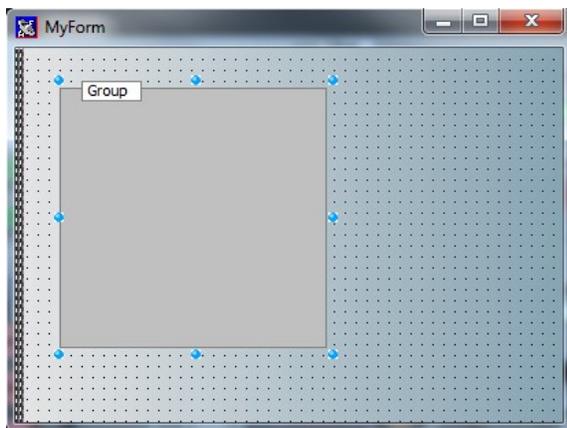
You can also drag the mouse pointer to create a rectangle selection around the outside of all the items you want to select.



When one (or several) object is selected, press the [ESC] key to unselect it. Then the form is selected and the *Tool palette* displays the form settings.

Resizing objects

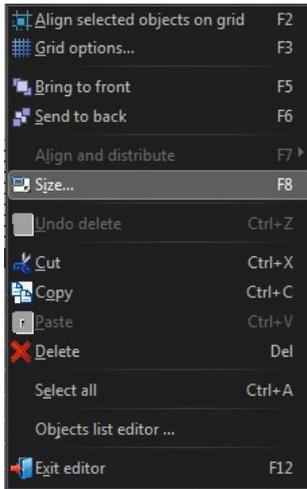
To resize an object, click one of the blue circles surrounding the object, hold down the left mouse button and drag it.



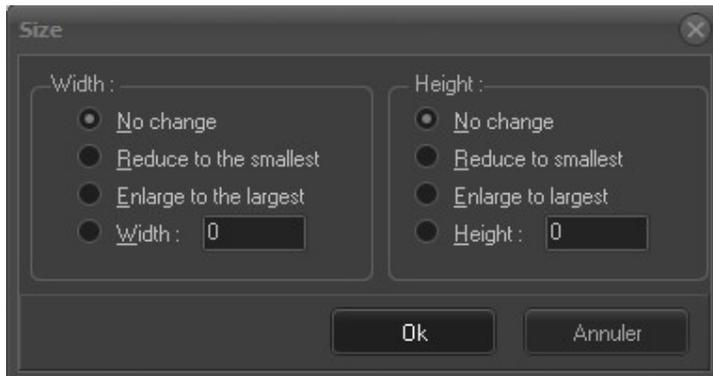
You can also resize the selected object **pixel by pixel** using the following panel from the *Tool palette* or from the Object list editor:



To adjust the size of several objects, select them and then right-click and select *Size* from the popup menu:



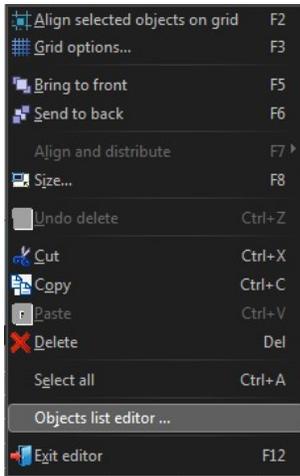
You can modify the width and height values of the selected objects:



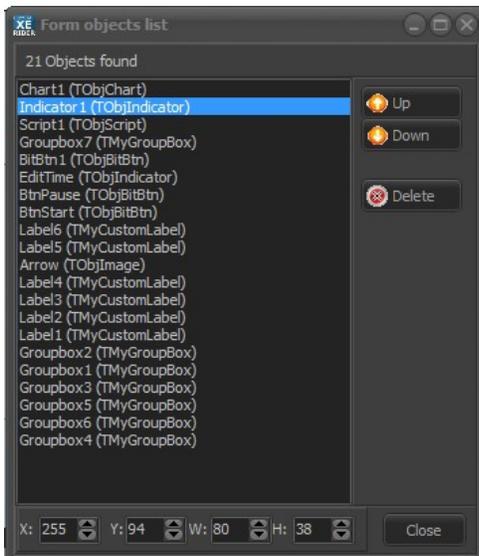
- **No change:** keep the current width or height value.
- **Reduce to the smallest:** set the width or height value of the object having the smallest width or height value among the selected object.
- **Enlarge to the largest:** set the width or height value of the object having the largest width or height value among the selected object.
- Enter the desired **width** and **height** value.

Object list editor

Right-click on a form background and to open the *Object list editor*:



The *Object list editor* lists all object contained in that form. You can change the list order, select an object, modify the object size and position or delete an object.



Moving objects

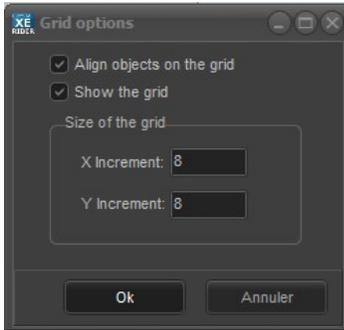
You can move an object all over the form by dragging it. Select the desired object, hold down the left mouse button and then drag it and drop it to any part of the window.

You can also select an object, and use the arrow keys to move an object over the grid.

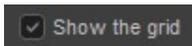
To move an object pixel by pixel, press and hold the [CTRL] key when moving the selected object using the arrow keys.

Grid options

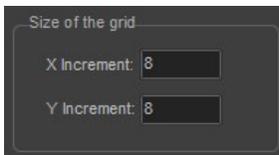
Right-click the form background and select *Grid options* to display the available options.



- To align objects on the grid, at least two objects must be selected.
- Check/uncheck the following box to show/hide the grid:

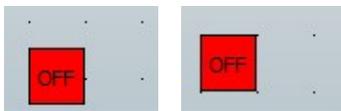


- You can define the size of the grid by adjusting X increment and Y increment values:



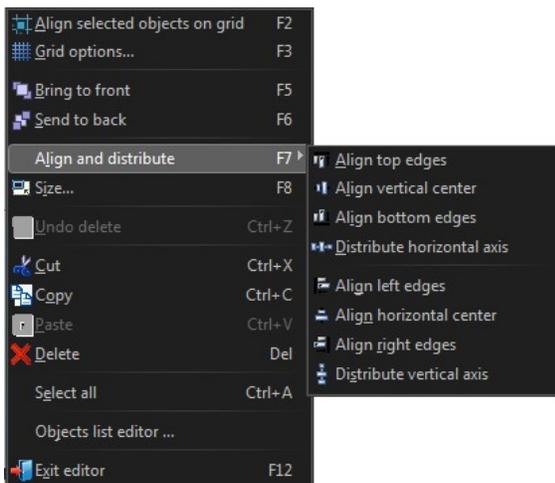
Aligning and distributing objects

If an object is not aligned to one or several objects on the grid, select them, then right-click and select *Align selected objects on grid* to align the objects at the top:



You can also align objects with each others.

After selecting several objects you want to align, right-click the form background to open the popup menu.



Move the mouse over *Align and distribute* to display the drop-down menu:

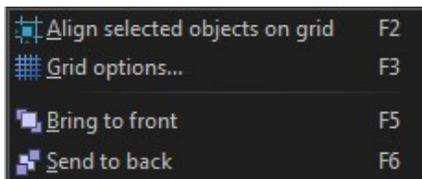
- **Align top edges:** objects will be aligned on the topmost object.

- **Align vertical center:** objects will be aligned on a virtual horizontal line located in the middle between the topmost object and the bottommost object.
- **Align bottom edges:** objects will be aligned on the bottommost object.
- **Distribute horizontal axis:** the horizontal space between each object will be distributed without changing the position of the leftmost object and the rightmost object.
- **Align left edges:** objects will be aligned on the leftmost object.
- **Align horizontal center:** objects will be aligned on a virtual vertical line located in the middle between the leftmost object and the rightmost object.
- **Align right edges:** objects will be aligned on the rightmost object.
- **Distribute vertical axis:** the vertical space between each object will be distributed without changing the position of the topmost object and the bottommost object.

Overlaying objects

You can change the layering order of the objects placed on a form.

- To lay an object on top, select it, then right-click and select Bring to front.
- If you want an object to be under other object, select it, then right-click and select Send to back.

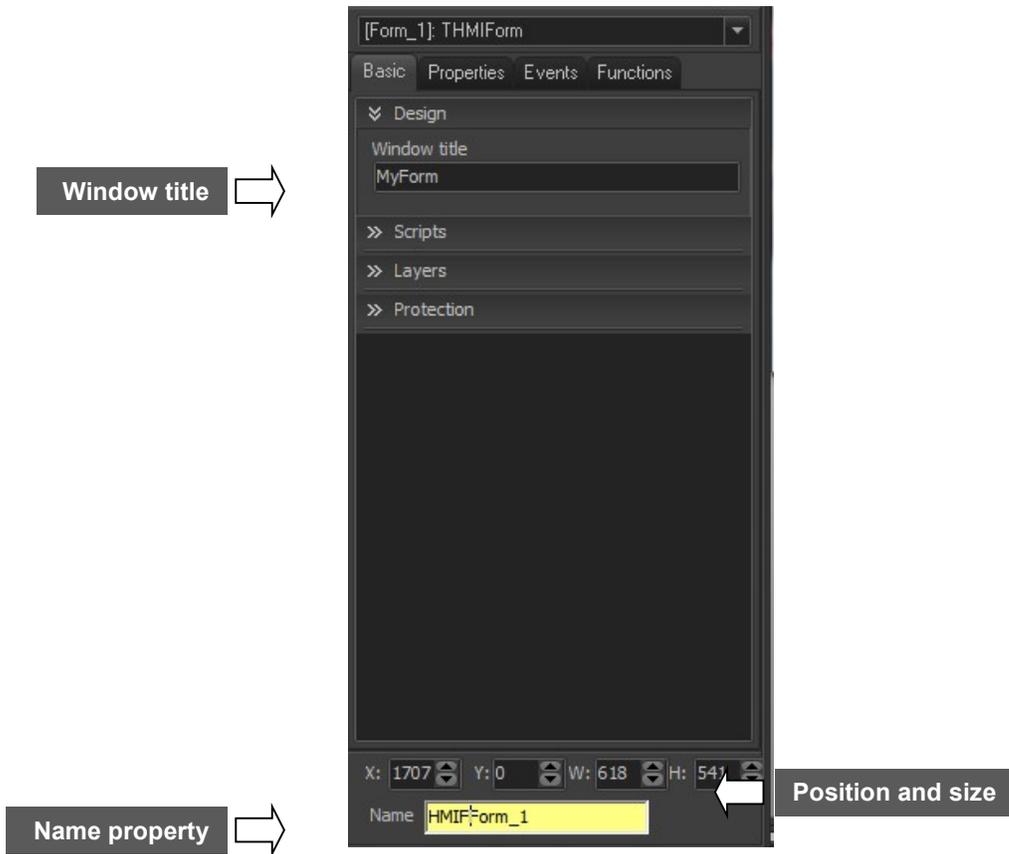


13.5. FORM PROPERTIES AND EVENTS

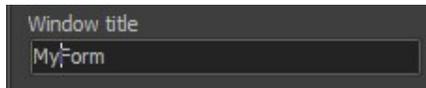
As any object, forms have their own associated properties and events. You can modify some of their properties to adjust the appearance and behavior of your forms. When a form is being edited, select the form (click on the form background) to display the form settings in the lower part of the *Tool palette*.

You can modify some of the form properties and specify an *OnCreate* event using the basic settings (*Basic* tab). All the form properties and events can be accessed using the advanced settings (*Advanced* tab).

13.5.1. Basic properties and OnCreate event



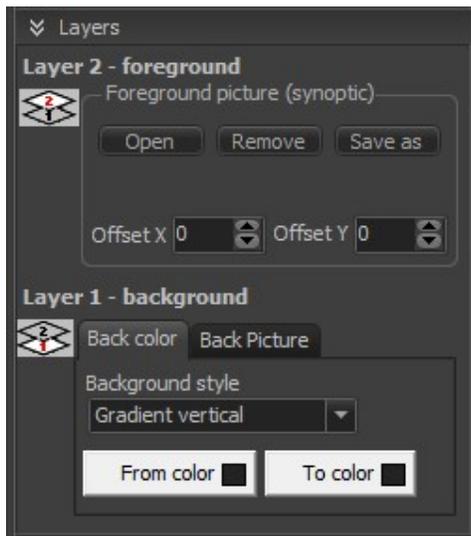
Window title



Type the title of your form into the text box. The title appears in the window's title bar. The window title differs from the *File* name of the form (*.form* file) and from the name property (used to identify the form in the program).

Background and foreground

The background of the form consists of two layers.

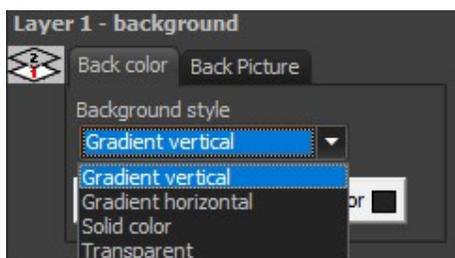


Layer 1 is below layer 2.

Usually, layer 1 is a gradient color and layer 2 is an image that represents the machine.

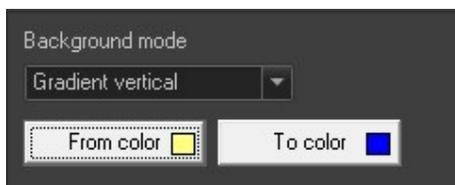
Layer 1:

Select the style of the form background from the drop-down list:



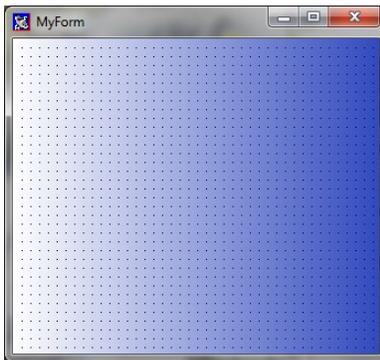
Back color

Define one (solid) or two colors (gradients) by clicking on the corresponding boxes:

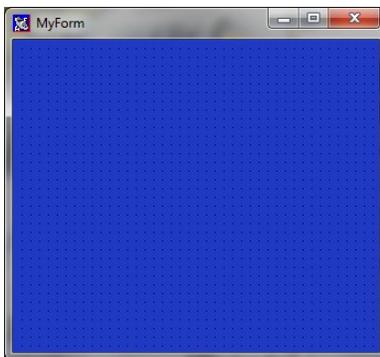




Gradient vertical



Gradient horizontal



Solid color

Transparent background

The background becomes transparent after exiting the *Tool palette*.

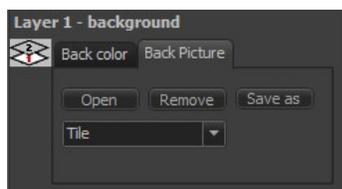


You cannot right-click on a form when its background is transparent. To edit the form, click on the Editors menu (menu bar), select Edit form and then select the form to be edited.



When designing, you cannot add objects to a form when its background is transparent. Select another background style before adding objects.

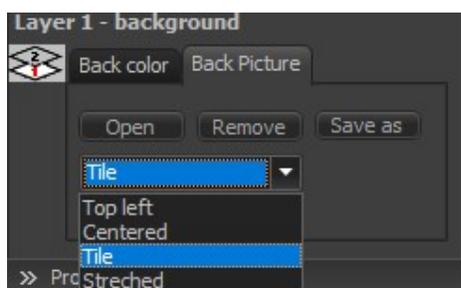
Back picture



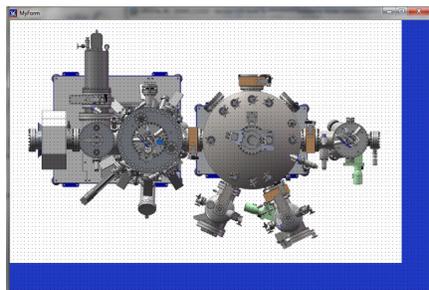
You can choose between four different layout options for the background image. If the image is smaller than your window size, a solid color border may appear around your image.

Generally, the back picture image allows the use of a repetitive texture in the background. So generally, we use the Tile option but several other options are available.

Select the desired back picture option from the drop-down list:



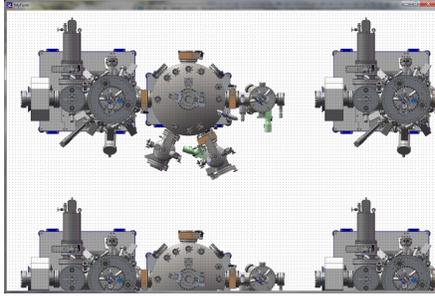
- **Top left:** the image is fixed in the left-hand corner of the window:



- **Centered:** the image is fixed in the middle of the window:



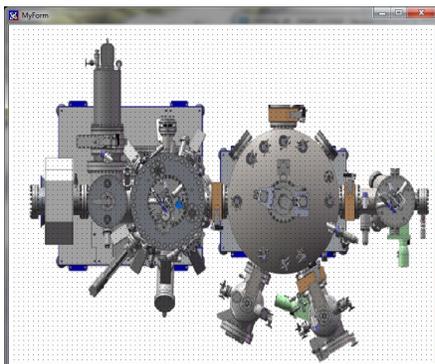
- **Tile:** the image is duplicated across the window:



Tile option is useful to allow small texture image to repeat horizontally and vertically across the window:

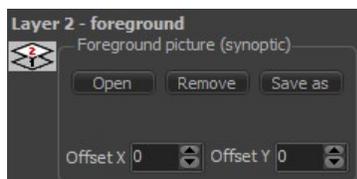


- **Stretched:** a single image is stretched across the whole window.



Depending of the window size, the ratio of the image can be changed, and the image can be distorted.

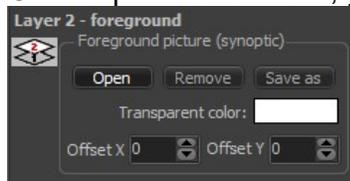
Layer 2: foreground picture



Click on the **Open** button to add a background image.

Compatible image files: Bmp, jpg, png and emf

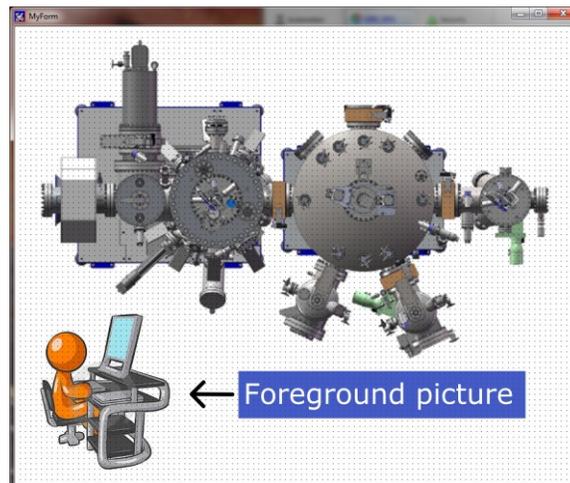
Once a picture is loaded, you can select a color to be used as a transparent color.



Click on the rectangle to change the transparent color.

All the pixels of the image which are the same color as the transparent color reveal the content of layer 1.

▪ Foreground pictures



Unlike a background image on layer 1 (fixed), a foreground image on layer 2 can be moved using the *Offset X* and *Offset Y* settings.

Use *Offset X* to adjust the position of the image from the left (of the window) and *Offset Y* to adjust the position from the top:



Form size and position

To adjust the size of the form, simply stretch it using the mouse, as any standard window. As you stretch the form, the width and height properties are adjusted.

You can move the form by dragging it directly to the desired part of the screen, as any standard window. As you move the form, the *Top* and *Left* properties are adjusted.

You can find the above properties in the following panel, located at the bottom of the *Tool palette*:



You can adjust the form position from the left (X) and from the top (Y) of the screen as well as the form width (W) and height (H) values using this panel.

Once you have selected a setting, you can increase or decrease the value using the up and down arrow keys.

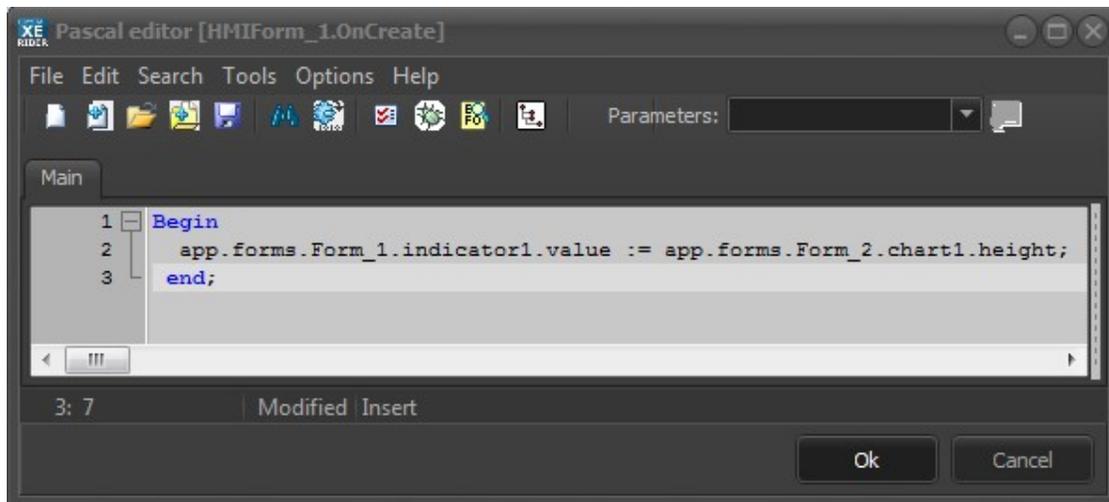
Form name

Name

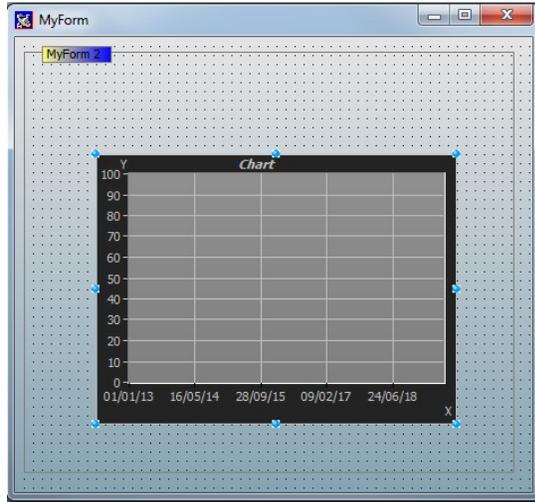
The *Name* property is a unique name identifying the form and giving access, throughout *Crystal XE* program, to all its properties and events as well as properties and events of all objects included in that form.

Example: Using a script (outside the *Tool palette*)

The object *Indicator1*, contained in the form *Form_1*, displays the height value (*height* property) of the object *Chart1*, contained in the form *Form_2*:



`app.forms.Form_1.indicator1.value := app.forms.Form_2.chart1.height`



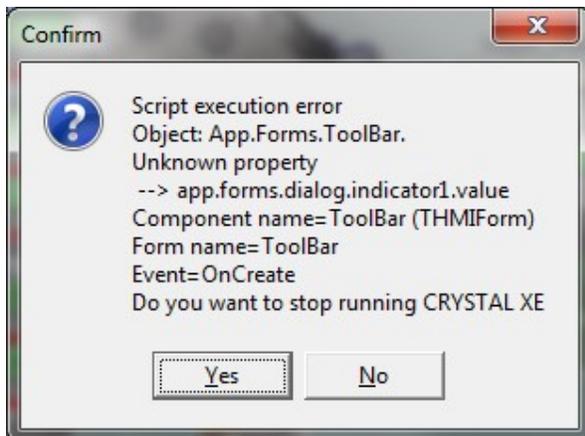
 Names are not case-sensitive

 The form must be open when editing the script (outside of the *Tool palette*) and when the script is executed.

If you attempt to **edit** the script when the form is not open, the following error message will appear:

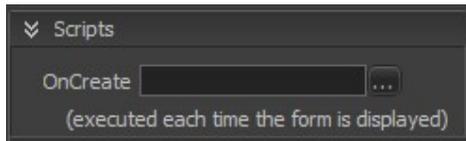


If you attempt to **execute** the script when the form window is not open, the following dialog box will appear:



From scripts (OnCreate event)

On the Basic tab, you can specify an *OnCreate* event by either using the *Scripts* menu or by double-clicking the form background.



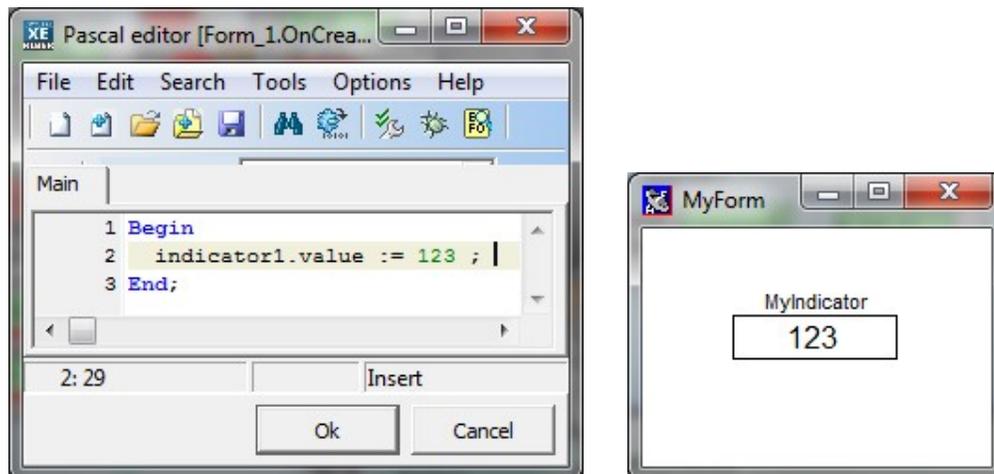
Click on the following icon  to open the Pascal editor and edit your script.

The edited script is executed when you exit the *Tool palette* and each time you open the form.



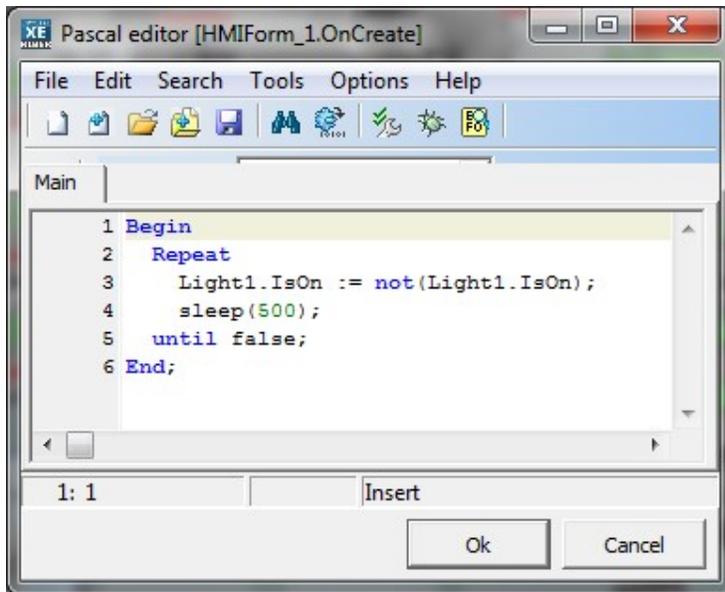
Scripts edited using either Basic or Event tabs override scripts that have been edited outside the *Tool palette*.

Example 1: Assigning a default value to an indicator when opening the form



Example 2: Infinite loop

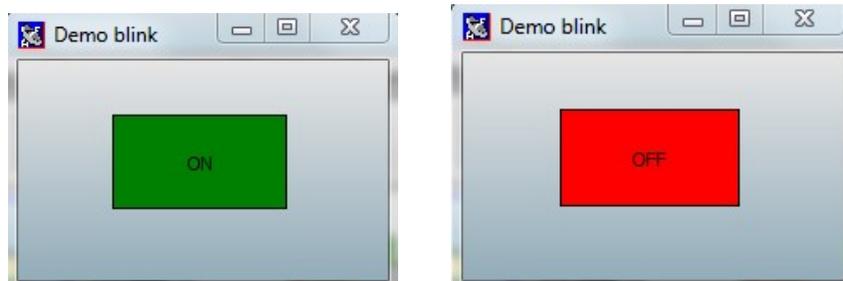
You can create an infinite loop using the reserved word *Repeat*. The task (script) will be repeating over and over until the form is closed:



It is recommended to insert the sleep statement within the infinite loop (see example above) to avoid overloading the processor.

 To insert a repetitive script in the form, we recommend that you use the object script.

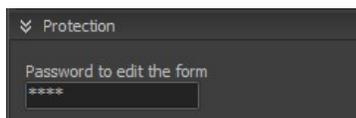
In this example, the object *Light1* will be switching from *On state* to *Off state*, at 500 milliseconds intervals (sleep) until the form window is closed (false).



Form protection

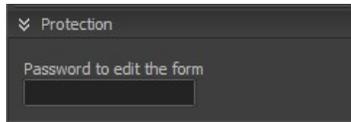
The *Protection* menu allows you to specify a password and limit access to the *Tool palette*. The password will be required when attempting to edit the form.

Specify the password using the text box:

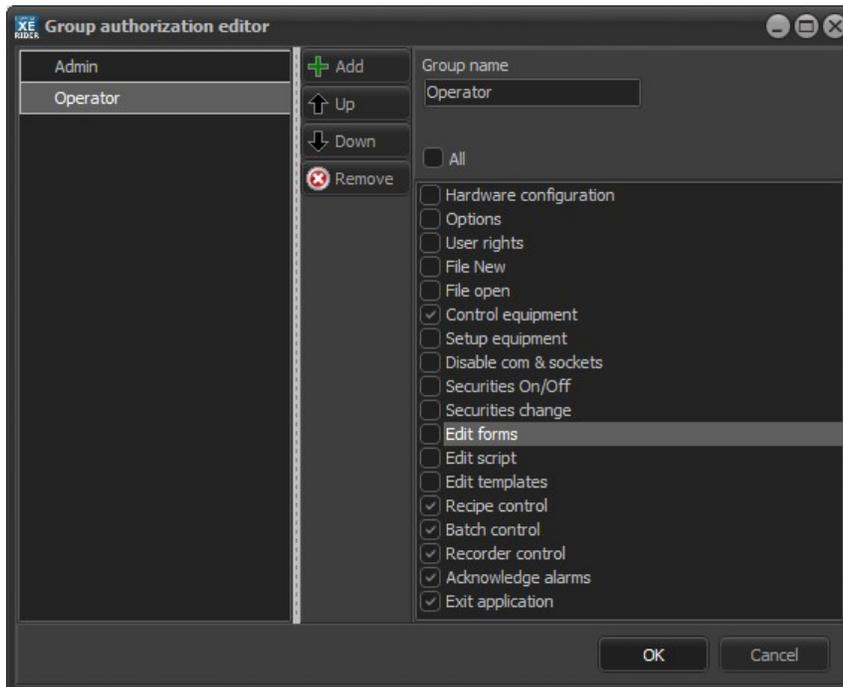


 The form can still be opened without requiring password.

To disable the password, leave the field empty as follows:



You can also allow/forbid the edition of forms to a particular group of users by checking/unchecking the *Edit forms* box in the *Group authorization* editor as follows:



The *Group authorization* editor can be accessed from the *Setup* menu > *Users* > *Edit groups* (you will need appropriate user right to perform that action).

13.5.2. Advanced properties and events

Click on the *Advanced* tab to display all the form properties and events.

Properties

Property	Value
Align	alClient
AllowClose	True
AllowMultipleInstanc	False
AllowSave	True
AlphaBlend	False
AlphaBlendValue	255
Anchors	TAnchors
AutoScroll	False
AutoSize	False
BackGround	TPicture
BkGrStyle	bsTile
BkStyle	f_Gradient_V
BorderIcons	TBorderIcons
BorderStyle	bsNone
BorderWidth	0
Caption	CRYSTAL XE - Main u
ClientHeight	645
ClientWidth	1246
Color	RGB(FFFFFFFFF000)
ColorFrom	RGB(FFFFFF)
ColorTo	RGB(808080)
ColorTransparent	RGB(FFFFFF)
Constraints	TSizeConstraints
Cl3D	True
DefaultMonitor	dmDesktop
DockSite	False

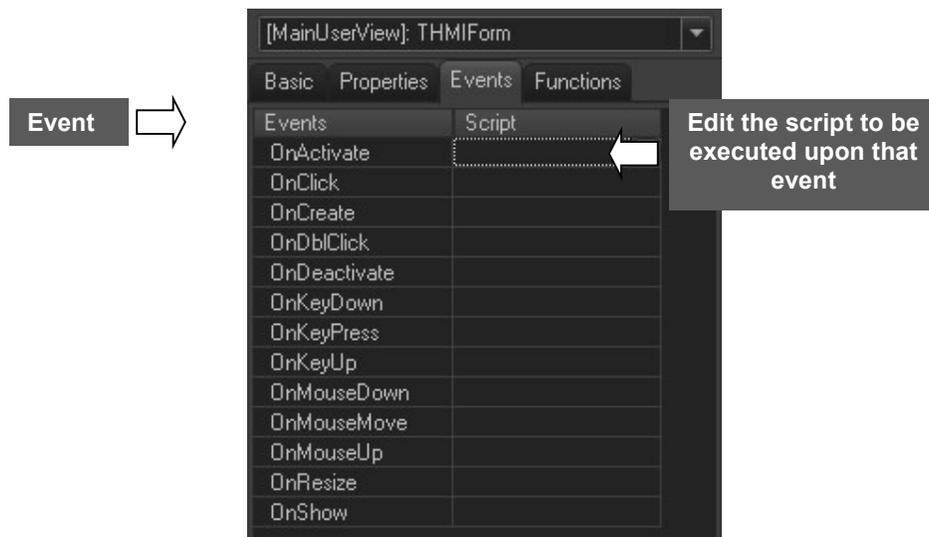
On the *Properties* tab, all the form properties and their values are listed in a table.

 The above example is for reference only. All properties have not been implemented yet in *Crystal XE*.

More details are also available on the Internet using the key words:

'Vcl.Forms.TForm Properties'

Events



The above example is for reference only. All events have not been implemented yet in *Crystal XE*.

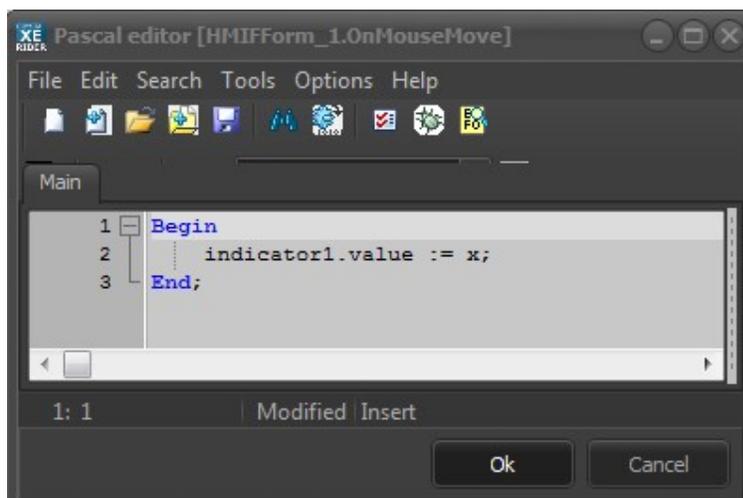
Additional help can be found on the Internet using the key words:

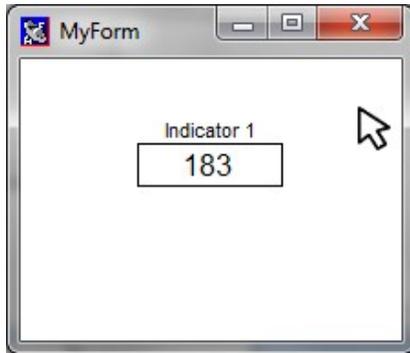
Vcl.Forms.TForm Events

Example 1: Specifying an OnMouseMove event:

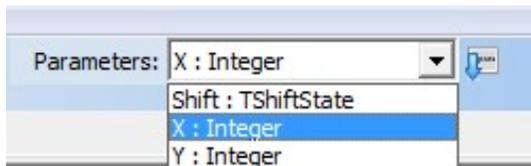


When moving the mouse over the form, the value of the Indicator 1 indicates the position of the cursor from the top left-hand corner of the window (X parameter).



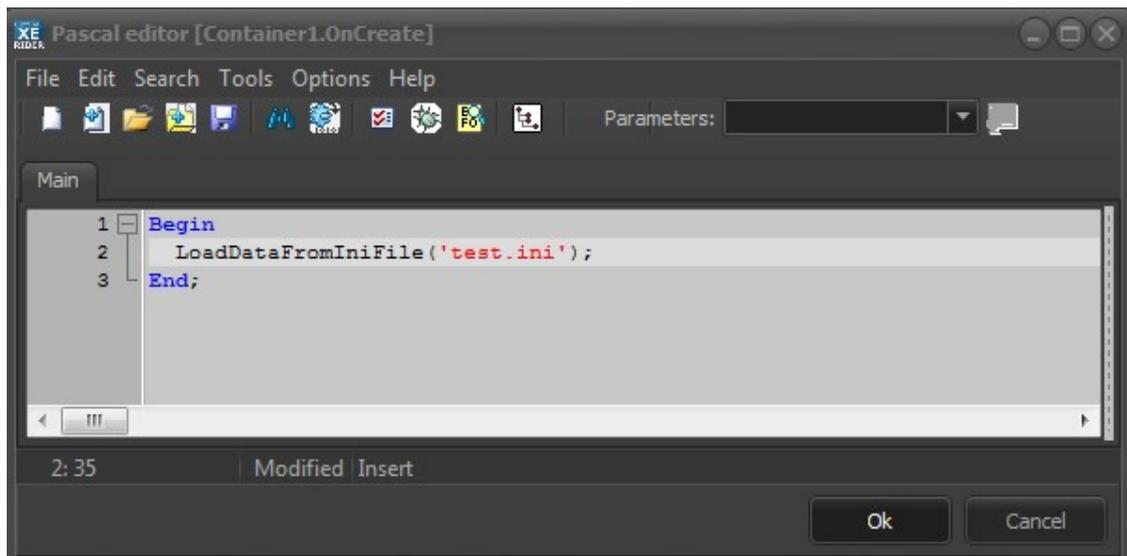


The parameters available with this event are listed in the *Parameters* drop-down list in the tool bar:



You can directly add the desired parameter to the script using the following icon: 

Example 2: Specifying an OnCreate event



In this example, when the form will be opened, the data of all the objects contained in the form will be loaded from a file.

This example assumes that the data of all objects contained in the form has been saved previously to the file 'test.ini' using the function *SaveDataToIniFile('test.ini')*.

13.6. OBJECT PROPERTIES AND EVENTS

13.6.1. Basic settings

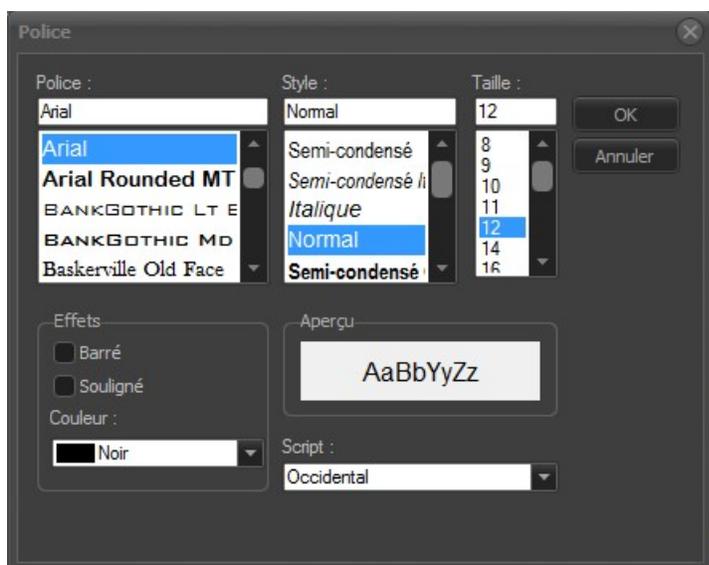
Object title and texts

Some objects are automatically accompanied by a title or a caption (indicators, group boxes, charts, buttons, etc.). If not, you can add it manually using a Label object and display static text.

You can customize the properties of all texts (titles, values, lists, etc.) provided with objects by clicking on the following icon:

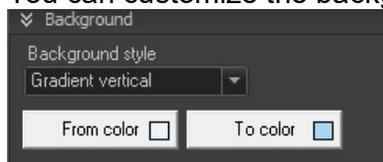


You can modify the:
Typeface, size, weight, color, etc...

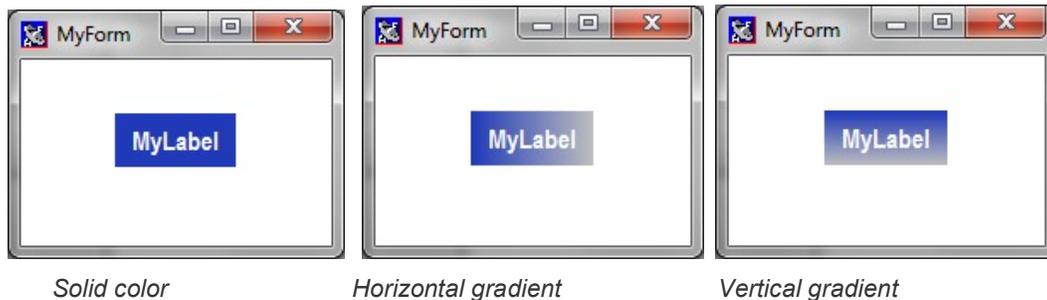
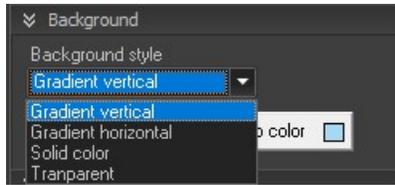


Object background style

You can customize the background of some objects such as labels, group boxes, graphs, etc.



Select the desired background style from the drop-down list and define the colors:



Transparent background option is only available with some objects (labels, indicators, etc.). Select *Transparent* in the style drop-down list or check *Transparent background* when applicable.



Object position and size



The above panel, located at the bottom of the *Tool palette* allows you to:
adjust the object **position** from the left (X) and from the top (Y) of the window
adjust the object **width** (W) and height (H) properties

Once you have selected a setting, you can increase/decrease the value using the up/down arrow keys.

Access properties and Data explorer

Some objects can be used to display one of the equipment properties and to modify its value (when writable).

Depending on the property, different types of data can be displayed:

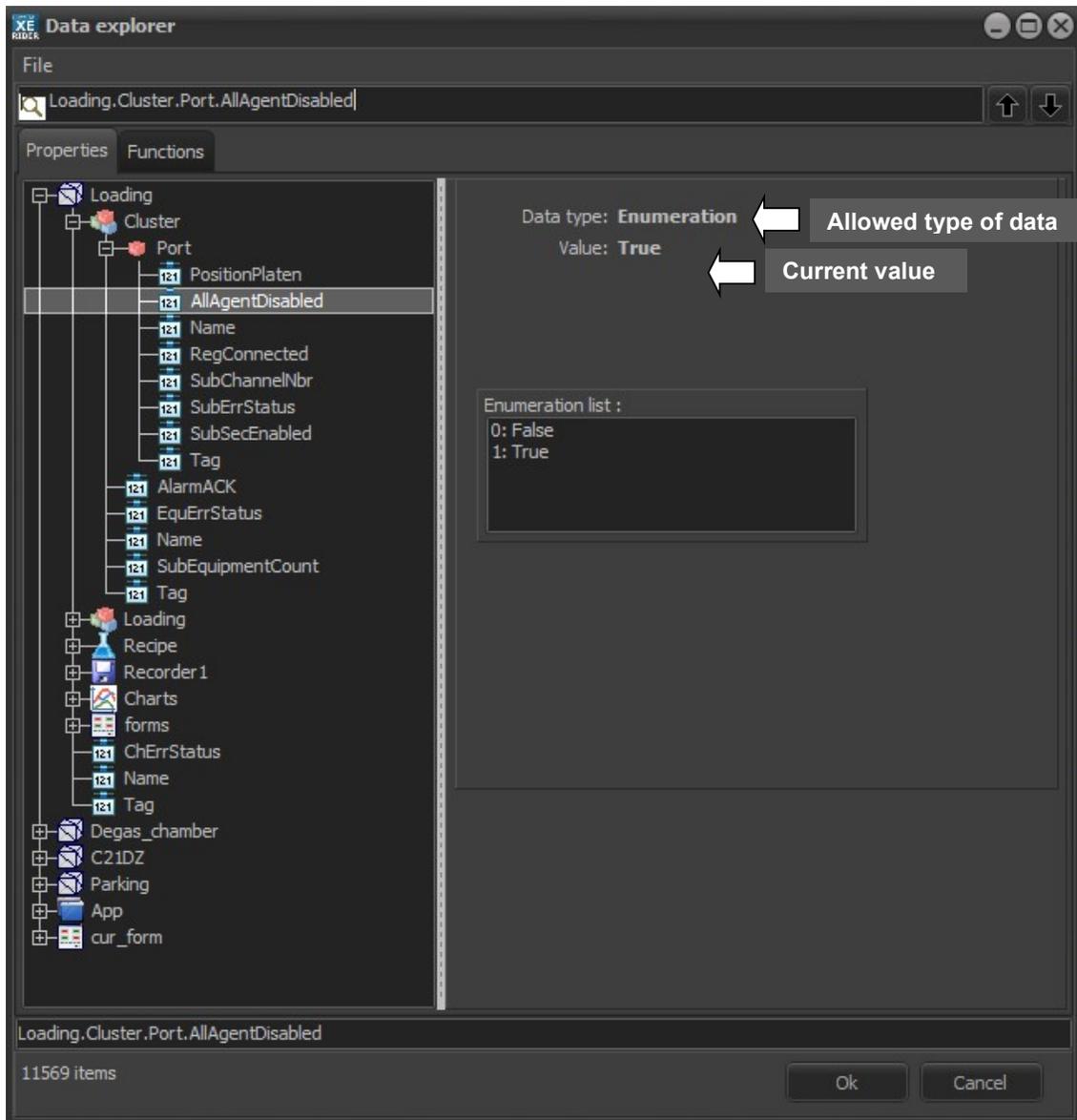
Numeric value: measured value, output power value, etc.

Enumeration: checked/ unchecked, on/off, etc.

Texts (strings): equipment name

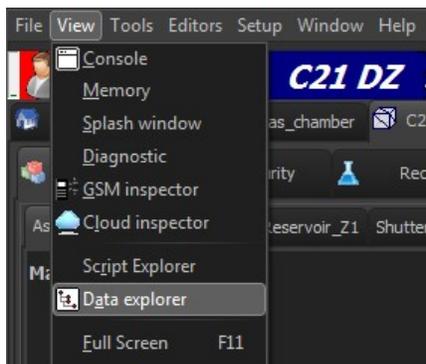
The *Data explorer* lists, in a tree structure, all the properties that can be displayed by the selected object.

When applicable, click on the following icon  to open the *Data explorer* and select the desired property:



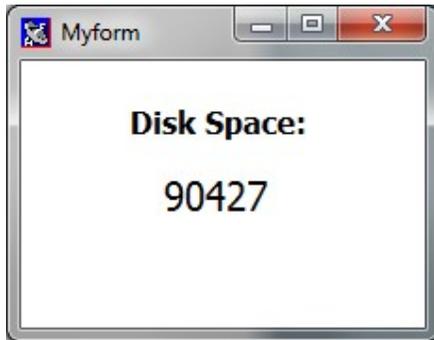
Data explorer

Outside the *Tool palette*, you can access the *Data explorer* from the menu bar by clicking on *View > Data explorer*.



Examples of displayed value:

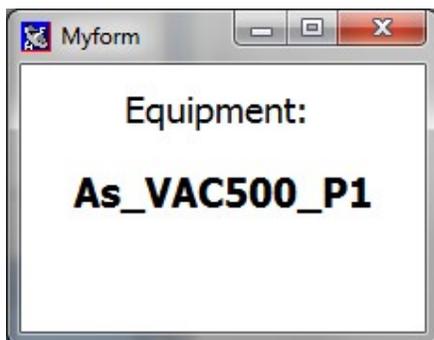
Numeric value: Disk_Space (App > Disk_Space)



Enumeration: AllAgentDisable (Loading > Cluster > port > AllAgentDisable) 1: true, 0: false

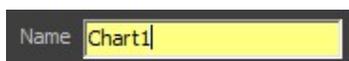


Text: Name (C21DZ>As_VAC500_P1> Name)



Object name

Use the yellow text box, located at the bottom of the *Tool palette*, to modify the name property identifying the object:



The name is an ID that gives you access to all the object properties and events in any script throughout *Crystal XE* program (and not only within the *Tool palette*).

When you first add an object to a form, a default name is allocated (Chart1, Chart2, and so on). You can rename the object by typing a new name in the text box.



If an object name has already been used in one or several scripts throughout the program, errors can occur when modifying that name.



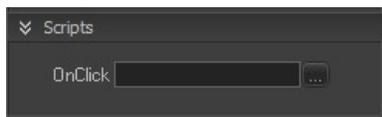
Names are not case-sensitive

Object scripts and events

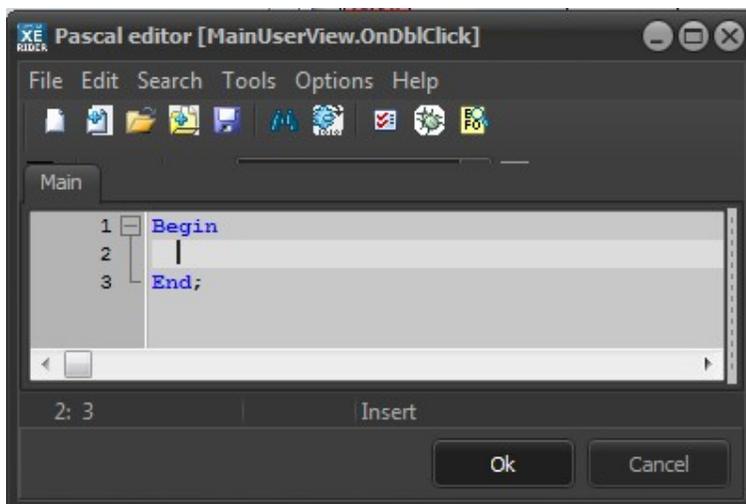
Events are used to specify an action to be performed by the program in response to a specific user interaction.

On the *Basic* tab, *OnClick* events can be specified for some interactive objects such as buttons, checkboxes, animated images, etc. The script is executed when the object is clicked.

When applicable, you can edit the script to be executed when the event occurs using the *Scripts* menu:



Click on the following icon  to open the Pascal editor:



13.6.2. Advanced settings

The Advanced tab gives access to all properties and events associated with the selected object.

Properties

Properties are specific values included within an object. Although some properties apply to most objects, the properties are different for each type of object.

The *Properties* tab gives you access to all the object properties and allows you to define their values.



Each property can only store a specific **type of data**: Booleans (true or false), numeric values or strings.

For example, the width property can only store numbers.



Depending on the property, you can modify a value by:

Selecting your choice from a drop-down list 



Opening a panel for more options



Typing the value directly into the text box:



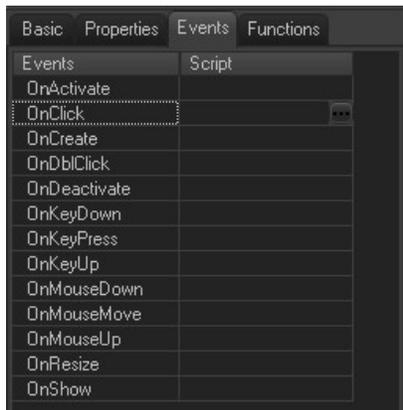


Some properties cannot be modified.

Events

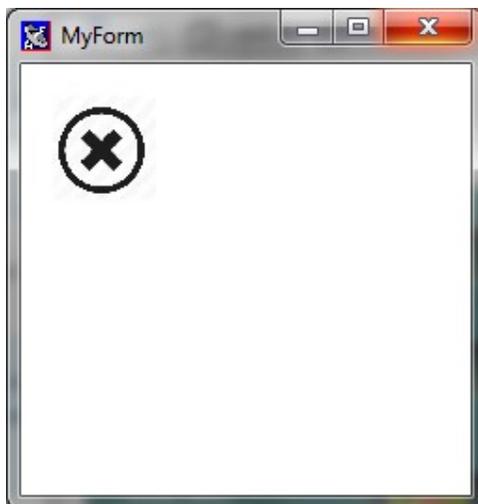
A set of events is associated with each type of objects.

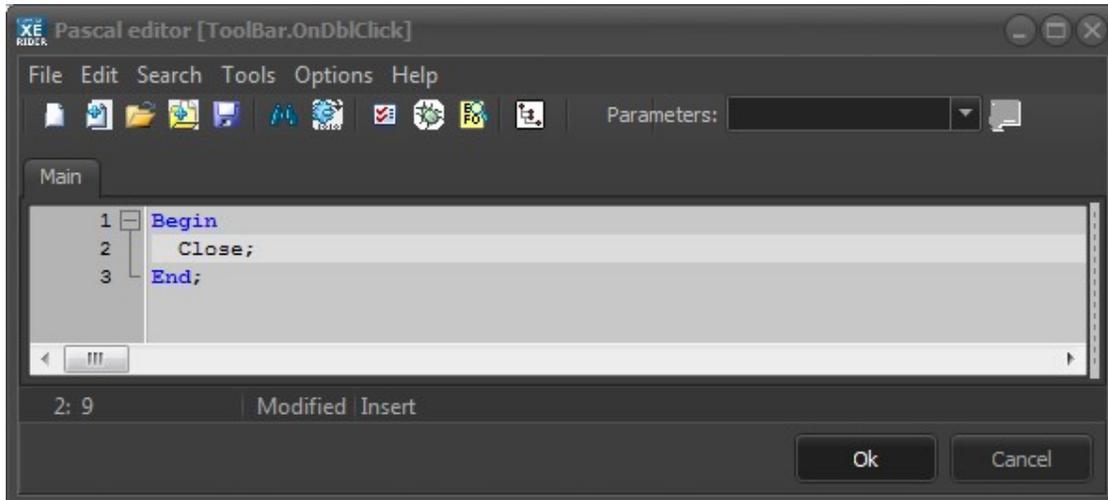
The *Events* tab gives you access to all events and allow you to edit a script to be executed when the event occurs.



Click on the following icon  to open the Pascal editor and edit your script.

Example: Double-clicking an image to close the form using an *OnDbClick* event





→ You can find more examples of events and scripts in the section **Form properties and events** in this chapter.

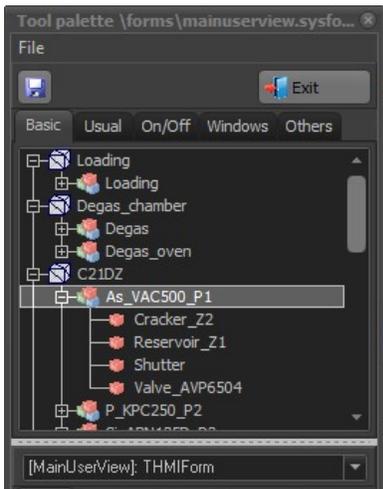
13.7. TYPES OF OBJECT

13.7.1. Basic tab (list of equipment)



To add this on a form, you can simply select it on the component palette, hold down the mouse button and release the button on the form.

The chambers and their associated equipment and sub equipment are listed in a tree structure at the top of the *Tool palette*.



Expand the lists and click on the desired item to pick up the object.



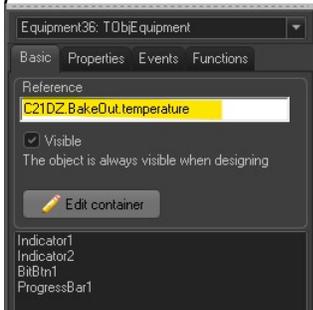
Some items do not have a corresponding visual object.

- **Drag and drop** the equipment object into the form window.

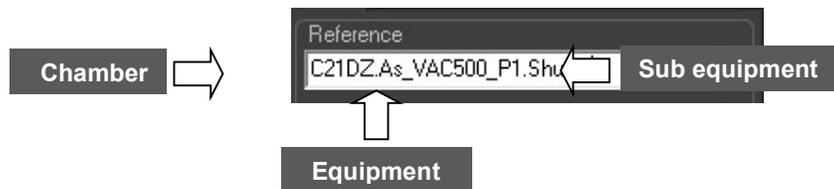


Equipment objects: Equipment and sub equipment

When the equipment is selected, its *Reference* (full path identifier) is displayed in the *Tool palette*.



The reference indicates the chamber associated with the equipment and the equipment associated with the sub equipment:

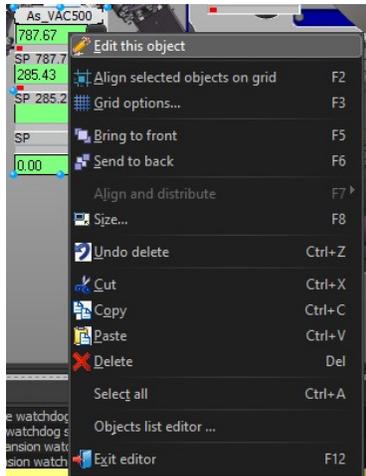


Uncheck Visible to hide the equipment or sub equipment item. The object is hidden only after exiting the Tool palette.

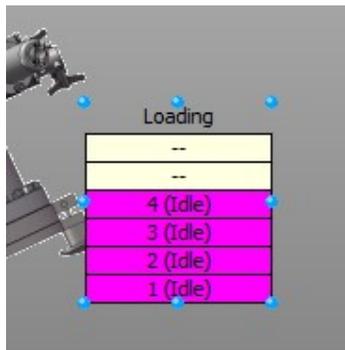


i All object in the basic tab are wrapped in another object.

It is possible to modify an object by right clicking on it and selecting “Edit this object”



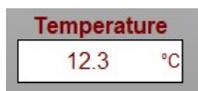
If your MBE system is equipped with a cluster, click on the chambers in the equipment library to display the cassettes. You can drag and drop the cassettes to the form.



Cassette object

13.7.2. Usual tab

Indicator object



To add this object on a form, you can simply select it on the component palette, hold down the mouse button and release the button on the form.

Indicators can display different **types of data**:

- Numeric values
- Enumeration
- Strings

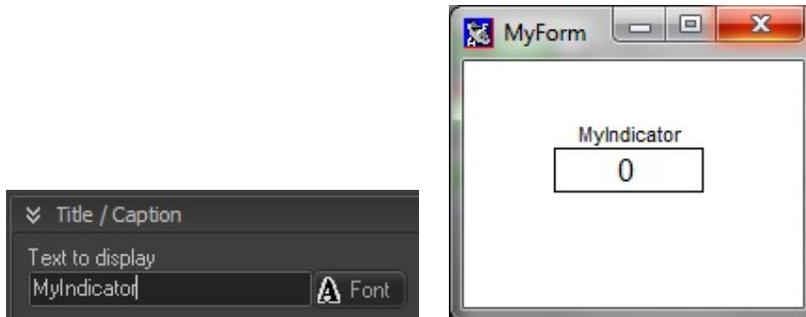
Object type: **TObjIndicator**

You can define the value to be displayed by an indicator by:

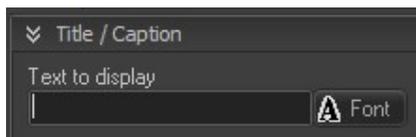
- selecting a property in the Data explorer  (Tool palette)
- typing a value directly into the text box (Tool palette)
- entering a new value in the indicator (users)
- assigning a value using any script throughout Crystal XE program

The displayed value is stored in the *Value* property (*Advanced* tab). When the specified property can be written, users can modify its value by entering the new value in the indicator.

The indicator object is automatically accompanied with a title:



To hide the title, leave the field text box empty as follows:

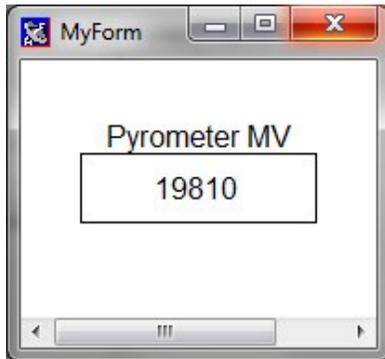


Value to display

Property

- Define a property to be displayed by clicking on the following icon  to select the desired property in the *Data explorer*.

Example: Pyrometer measured value (C21DZ>Wafer_temperature>Pyrometer>MV)



You can also type the desired value directly into the Property text box:



The defined property corresponds to the indicator *IdentInput* property value (*Advanced* tab):

IdentInput	perature.Pyrometer.MV
IdentInput	123

User entry

By default, user entries are disabled with indicators. You can enable users to modify a property value using an indicator (writable property only).

On the *Value* to display menu, check the following box to allow users to enter a value into the indicator:



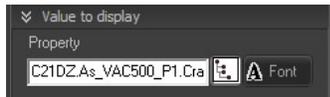
Some property cannot be modified (read only). The entered value will not be displayed if a read only property has already been defined using the *Data explorer*.

The value to be modified corresponds to the *IdentOutput* property (*Advanced* tab).

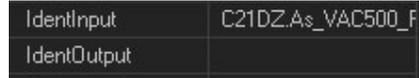
When no output property has been specified, the value is written to the input property.

Example: Modifying the cracker target setpoint

(C21DZ > As_VAC500_P1 > Cracker_Z2 > TSP)



Basic tab



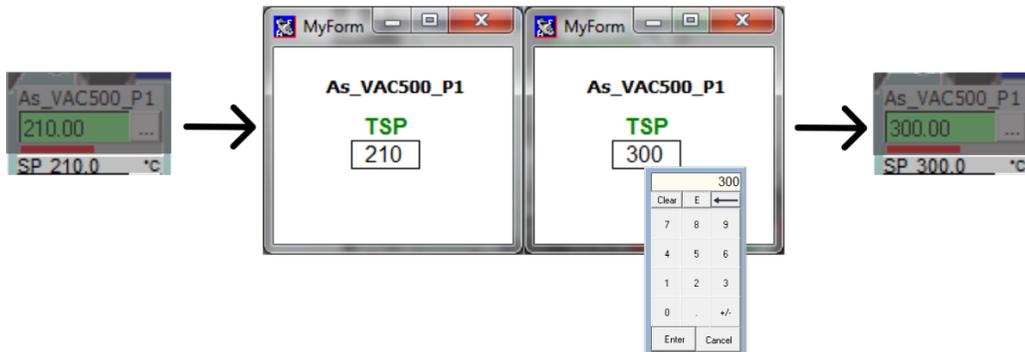
Advanced tab

Defined property (input)

Displaying the value

User entry (output)

Modifying the value



- The input tag property can be different from the output tag.

Example:

Property to display (IdentInput): Value of the indicator 2

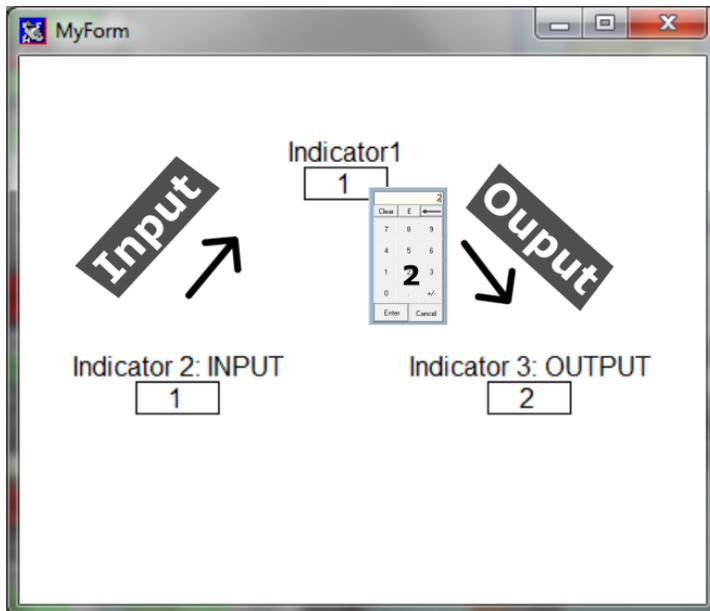
Property to write to (IdentOutput): Value of the indicator 3



Basic tab



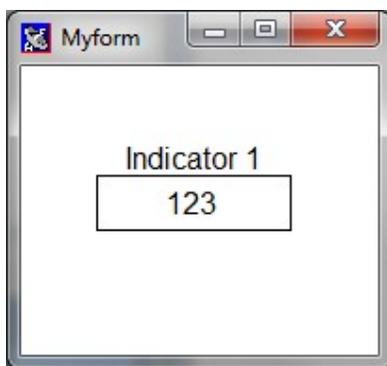
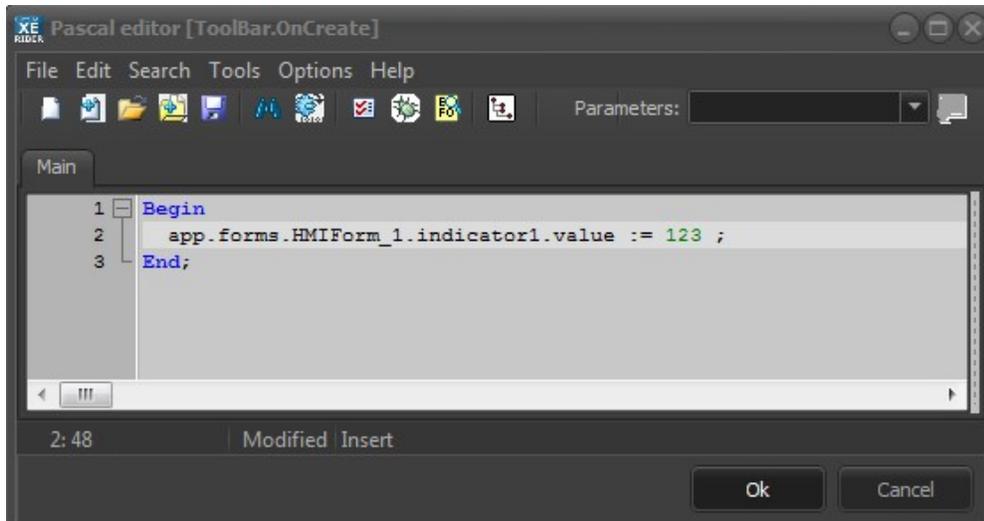
Advanced tab



External script

With any script throughout *Crystal XE* program, you can assign a value to an indicator included in a form using the form and object identifying names.

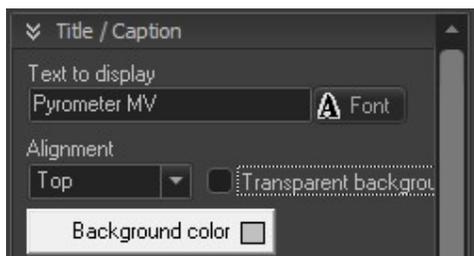
Example: Assigning a value to the Indicator1 included in the HMIForm_1 (MyForm):



 The assigned value will not be displayed if a property has already been defined using the *Data explorer*.

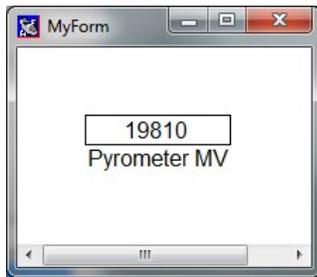
Customization

Title

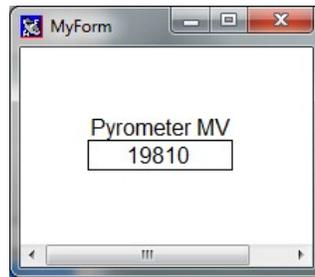


You can customize the title font  as well as the title background (transparent, solid or gradient).

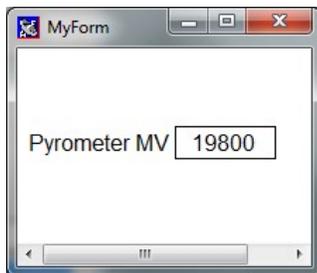
Use the drop-down list to select the desired title alignment.



Bottom



Top

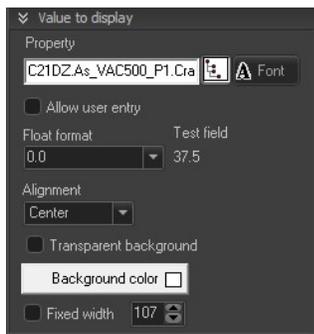


Left



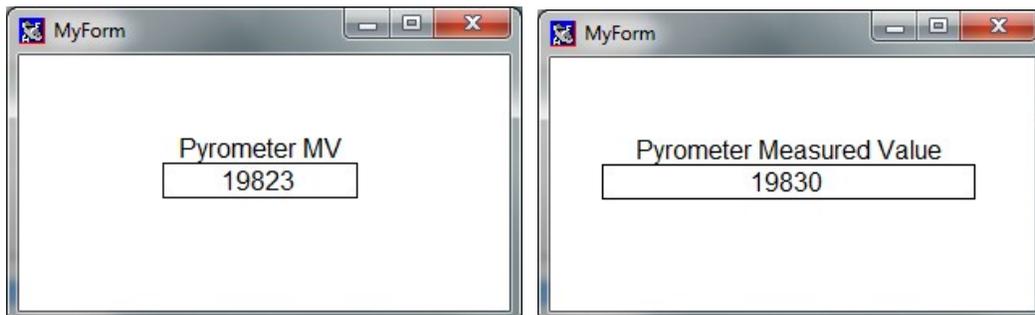
Right

Value

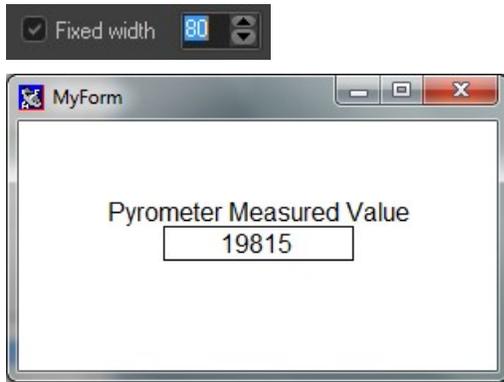


You can also customize the displayed value font  as well as the text box background (transparent, solid or gradient)

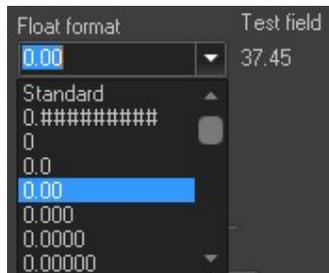
By default the text box width extends according to the title length.



Check Fixed width and specify the desired width to freeze the text box:



You can define the float format of the value using the following drop-down list:



You can also custom the float format by entering directly the format.

Example of format:

T= 0.00 degrees" will display "T= 670.34 degrees

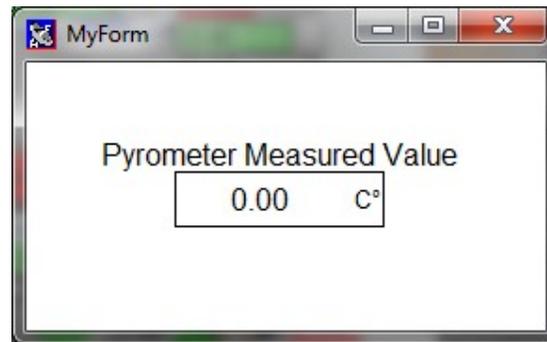
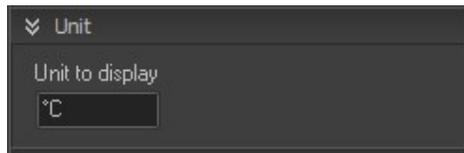


Decimal number input will be allowed according to the selected format.

Unit

Add a unit of measurement to the value using the following text box (*Unit* menu):

Example: Degrees Celsius (C°)



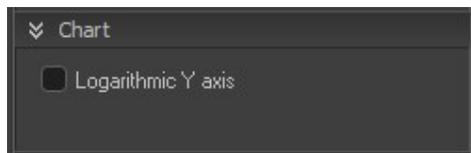
By default, the unit factor displayed is equal to 0.8 times the size of the font.

To change this factor, go to *Advanced tab > Properties* and adjust the property *UnitySizeFact*.

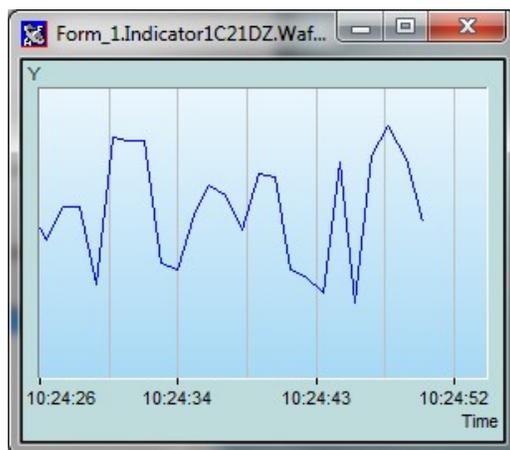
Chart

By default, graphs have a linear scale. A logarithmic scale is a nonlinear scale usually used with large range of values. Each tick mark on the scale corresponds to the previous tick mark multiplied by a certain number.

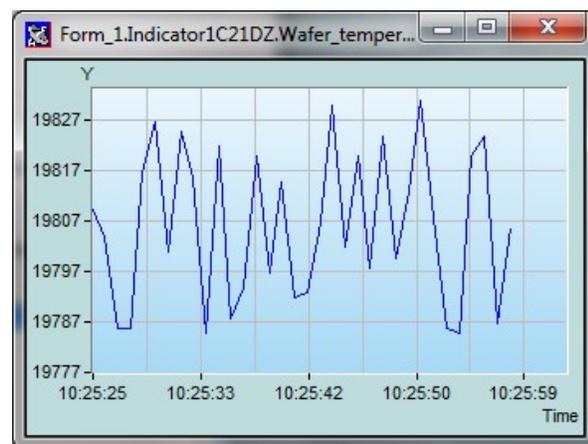
On the *Chart* menu, check the following box to follow the indicator value in a logarithmic chart (Y axis):



After exiting the *Tool palette*, right-click on the indicator and select *Follow in a chart* to view the chart in a pop-up window:

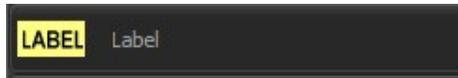


Logarithmic Y axis



Linear scale

Label object



Label objects are mainly used to display static text. They can also display values of a specified property.



To add this object on a form, you can simply select it on the component palette, hold down the mouse button and release the button on the form.

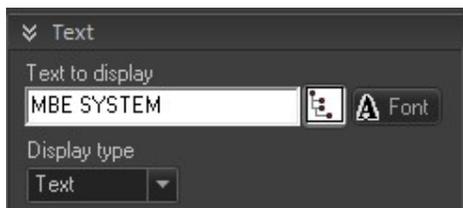
Value to display

A label object can display two different types of value: static text or property. Select the desired type from the drop-down list:

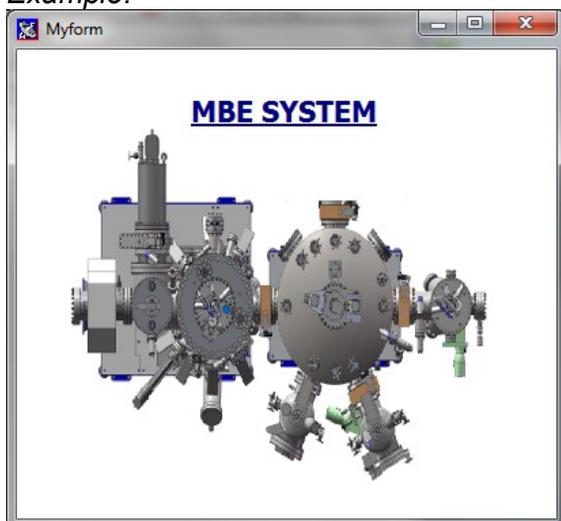


Static text

Type the text to be displayed directly into the text box:

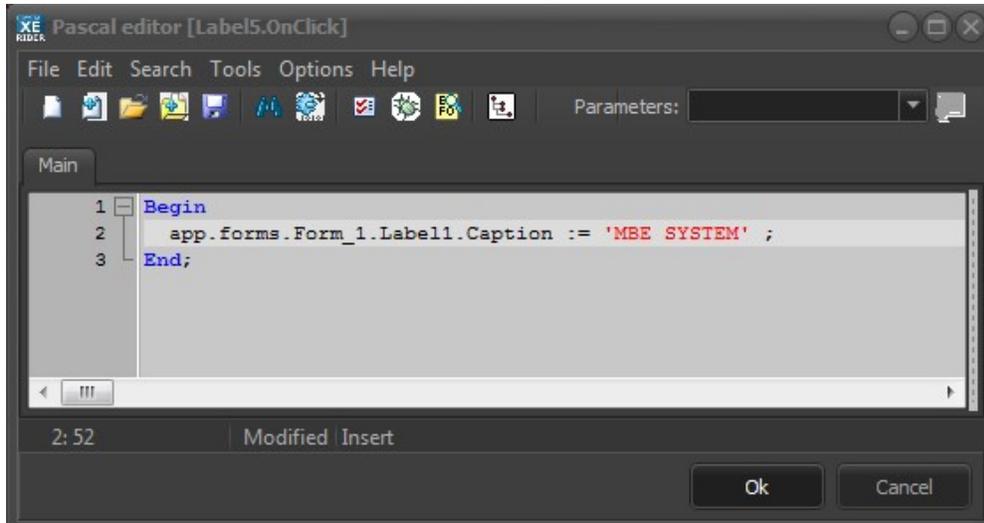


Example:



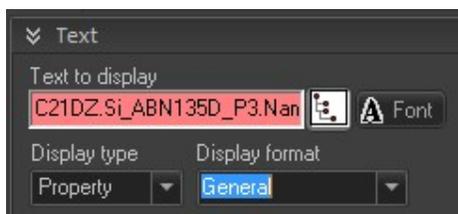
The text displayed by the label is stored in the Caption property. You can modify the text using any script throughout the program.

Example:



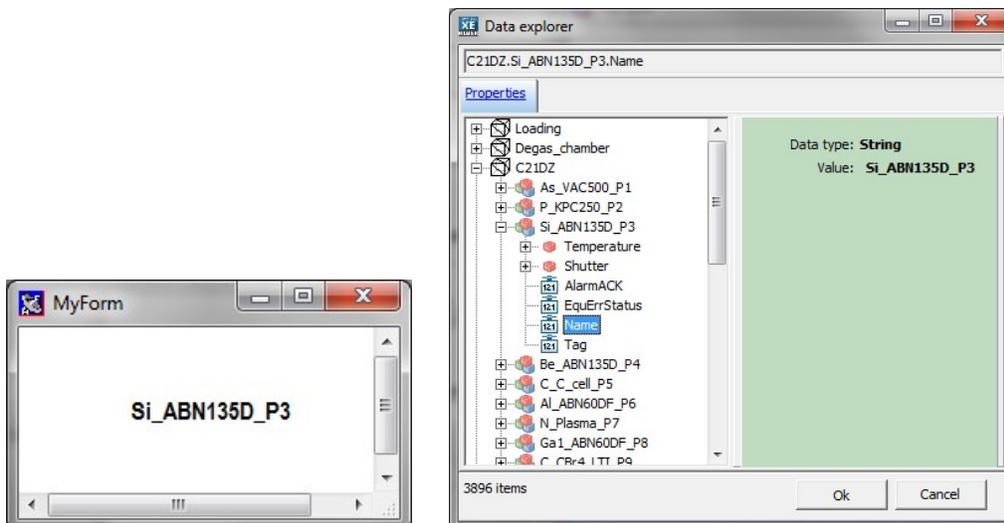
Property

Select Property in the drop-down list to display a property value:

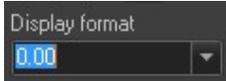


Click on the following icon  to display the available properties in the Data explorer.

Example: name of equipment (C21DZ>Si_AB135D_P3 > Name)

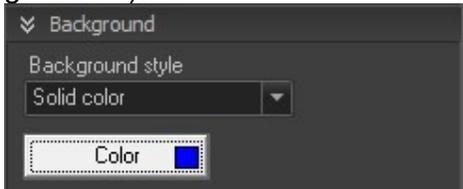


- Select the desired float format from the following drop-down list:



Customization

Click on the font icon  to customize the font style (typeface, weight, size, color, etc.). Use the Background box to customize the label background style (transparent, solid or gradients):



The following drop-down lists allow modifying the text alignment: horizontally (left, center or right) and vertically (top, center or bottom):



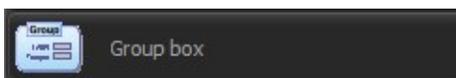
You can add a border to the text area and adjust the width using the following drop-down list:



Checking Wordwrap allows the text to move to a new line when reaching the end of the text area:



Groupbox object

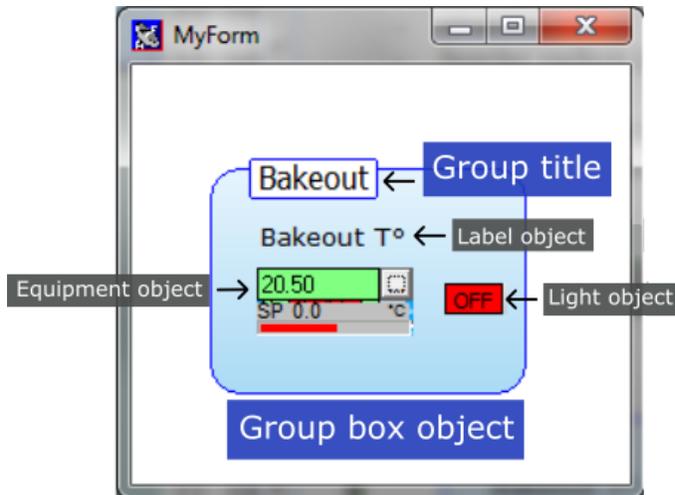


A group box is a visual object allowing you organize your form.

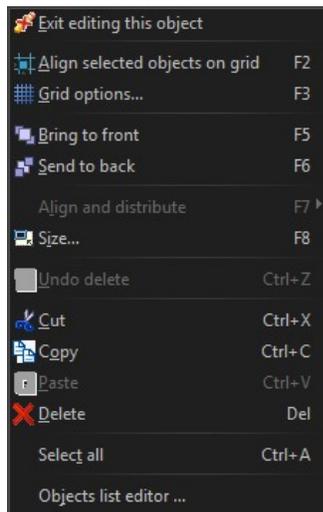
Object type: **TMyGroupBox**

To add this object on a form, you can simply select it on the component palette, hold down the mouse button and release the button on the form.

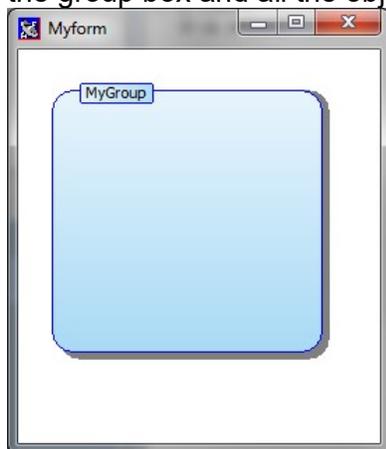
Example:



Objects can be overlaid with other objects. The objects are overlaid on the form as you add them to the window. If the group box lies on top, right-click on the item and select *Send to back* or press on the F6 key.



 If you want to move the group box and the overlaying objects together, you must select the group box and all the objects before dragging them.



Default group box

The group box is automatically accompanied with a **title**. To hide the title, leave the title input field empty.

On the *Basic* tab, you can customize:
the title font style,
the title alignment,
the title background (gradient or solid),
the box background (gradient, solid or picture).

- To define a **background image**, select *Picture* from the drop-down list, click on *Load picture* and then select the desired image file.



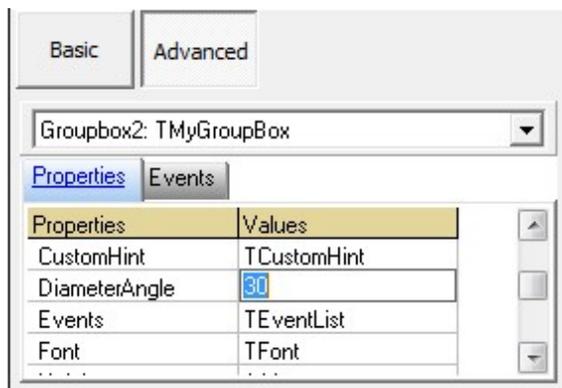
Only BMP image *File* can be loaded.

Check/uncheck Show shadow to display or hide the box shadow.



You can modify the border radius to create rounded corner or right angle group boxes using the advanced settings (Advanced tab).

On the *Properties* tab, select the *DiameterAngle* property and then type the new border radius value into the text box:



30 pixels by default

Enter 0 to set right angle boxes.

Examples of DiameterAngle value:

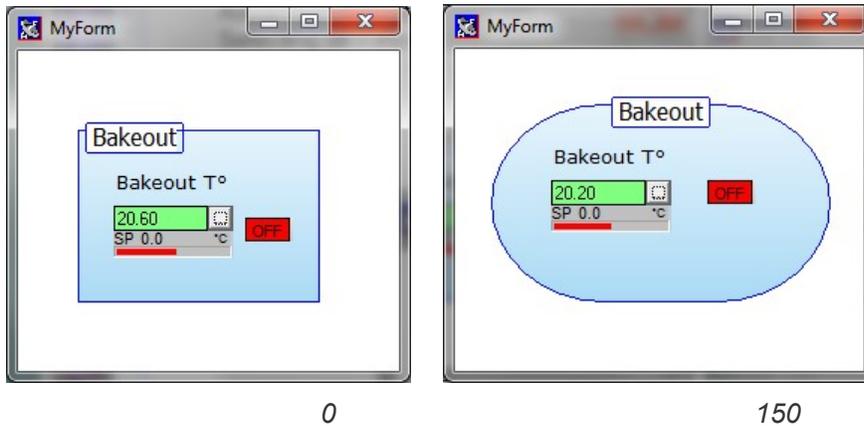


Chart object



Chart objects allow to graphically represent data using line graphs. You may have one or several chart objects included in one form. Each chart can display one or several lines (curves) acquiring data. For each curve you can define a property for the Y axis (X axis properties can be defined according to the selected type of chart).

Object type: **TObjChart**

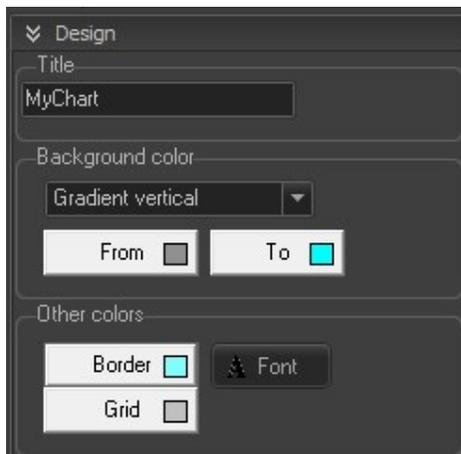
To add this object on a form, you can simply select it on the component palette, hold down the mouse button and release the button on the form.

Right-click on a chart to display the available options and modify the chart properties. To start/pause data acquisition, right-click on the chart and select/unselect *Acquiring data*.

To save the data, right-click on the chart, select *File > Save as* and then select the desired *File* format (CHR, CSV or BMP).

→ For more details about chart options, please refer to the chapter *Chambers > section [Charts](#)* in this manual.

Design



Default settings

The *Design* menu allows you to customize your chart:

title,

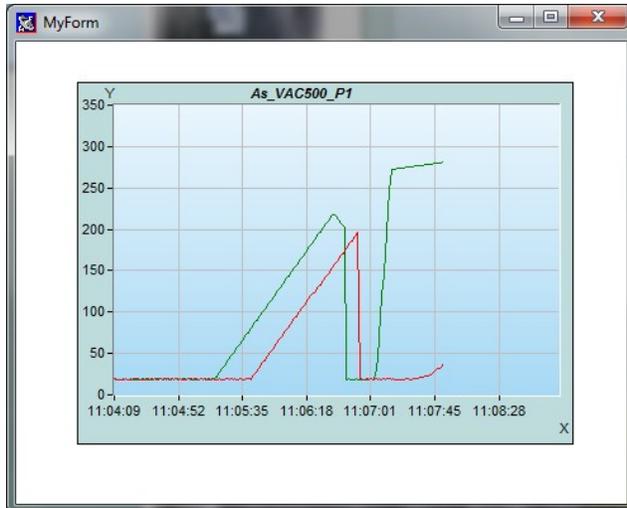
font,

background style and colors (solid or gradients),

border color,

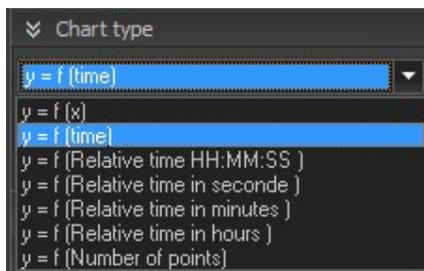
grid color.

Chart type



Default chart

Select the desired chart type from the drop-down list:



- **$y = f(x)$** : define Y property and X property
- **Relative time**: define only Y property (time on X axis)
- **$y = f(\text{Number of point})$** : define only Y property (tick marks correspond to the number of points on X axis)

Logarithmic scale

By default graphs have a linear scale. A logarithmic scale is a nonlinear scale usually used with large range of values.

You can display charts using logarithmic scale for Y axis. With the $y=f(x)$ chart type, you can use a logarithmic scale for Y axis or X axis only or for both. Check the following boxes accordingly:

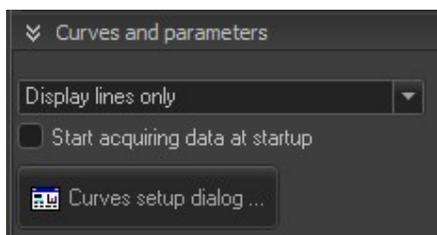


Show legend

Check *Show legend* to display the curve legends on the right side:



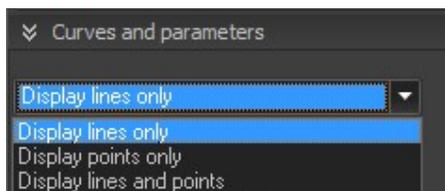
Curves



With line charts, the data is plotted by points connected by straight lines. You can display either lines or points only or you can display both lines and points.

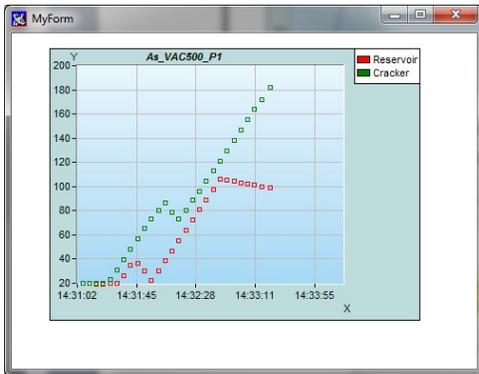
Display

Select the desired display option from the drop-down list:

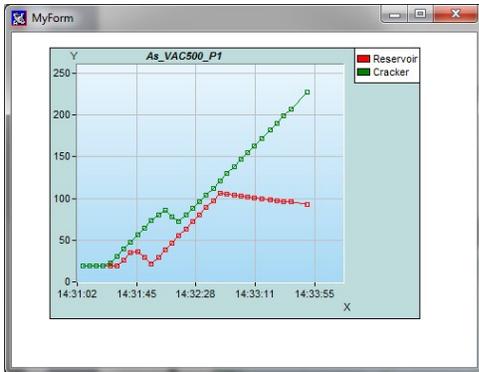




Lines only



Points only

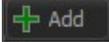


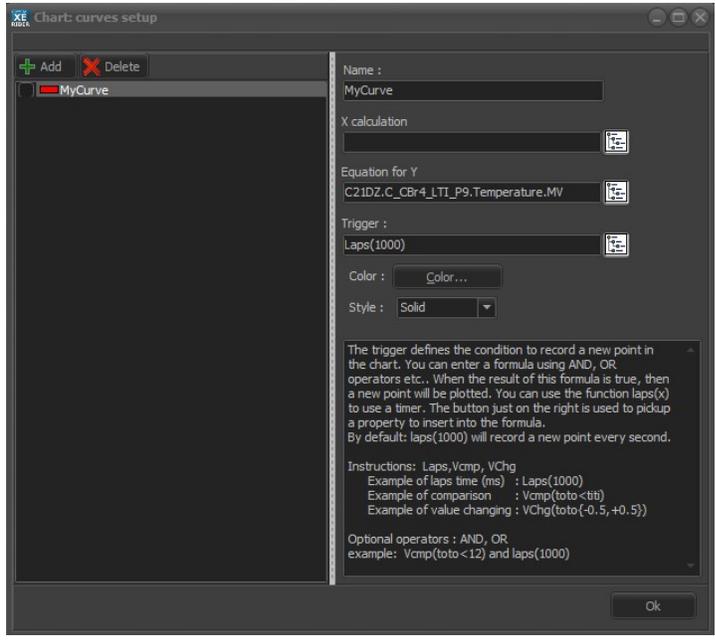
Lines and points

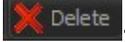
Curves setup

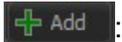
Click on the following button to open the curves set up dialog box and set the chart's curves:



- To add a new curve, click on the **Add** button 



To **delete a curve**, select the curve to be deleted and click on the **Delete** button .

After clicking on **Add**  :

Enter the curve **name**:



Define a **property** for the Y axis

Type the desired property or click on the following icon  to select a property in the *Data explorer*.

Example 1: Temperature measured value

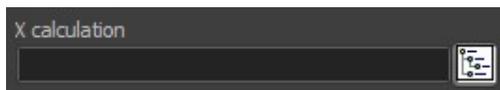


Example 2: Indicator value



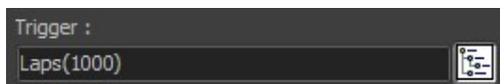


If the $y=f(x)$ chart type has been selected; you can define a property for the X axis:

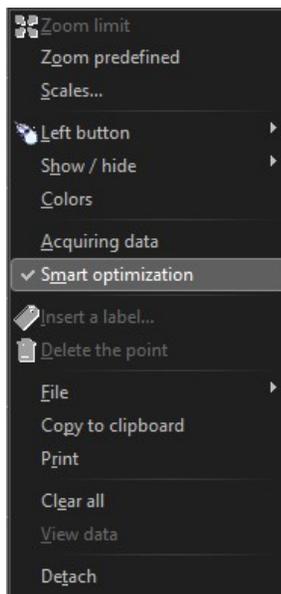


The **trigger** is a conditional statement that, if proved true, plots a point on the chart.

By default, a point is plotted each second.



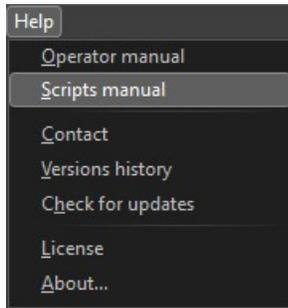
If *Smart optimization* is enabled, the curve displays the points expressing a change in value only. To enable/disable *Smart optimization*, right click on a chart and then select or unselect it:



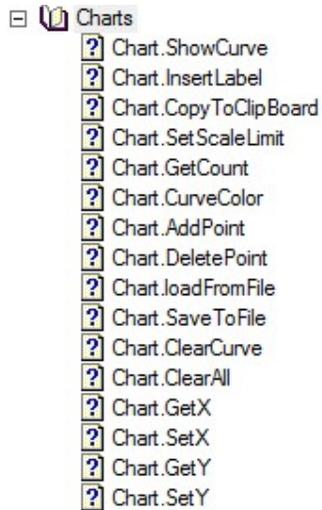
You can plot points on the chart using the chart functions in any scripts throughout the program.

→ For more details about chart functions, please refer to the Help menu.

Click on the Help menu (menu bar) and then select Scripts manual to open Crystal XE Help window.



In the tree structure, expand *Scripts > Functions reference > Charts* to display the chart functions:

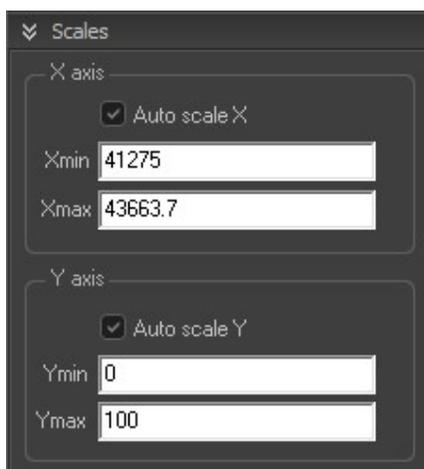


Function details appear in the box located on the right side of the window.

Scales and units

Scales

On the *Scales* menu, set the ranges of minimum and maximum values for each axis:



Default scales: $Y = f(x)$

By default **Auto scale** is enabled: the last points of the running curves are always displayed automatically. Each time a curve reaches the end of a chart window the X axis is recalculated. Uncheck the box to disable *Auto scale* option.

Units

On the Units menu, enter a unit of measurement for each axis.



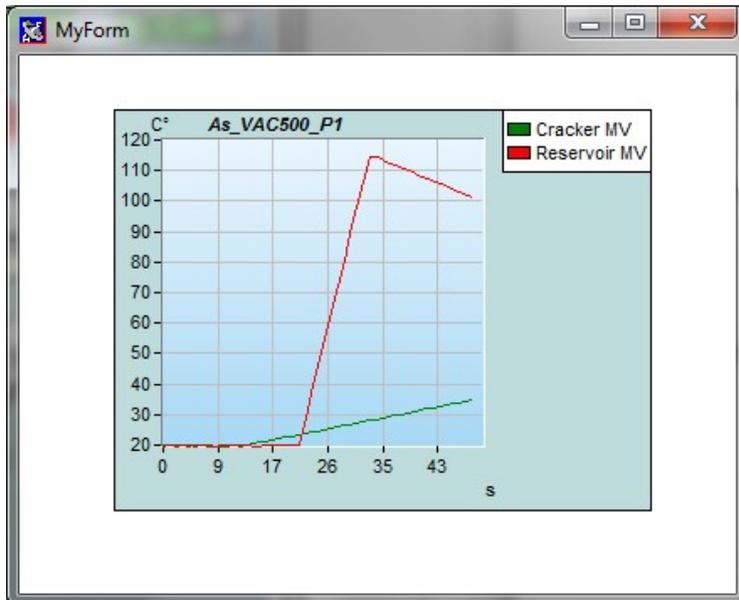
The screenshot shows a dark-themed window titled 'Units'. It contains two input fields: 'X unit' with the value 'X' and 'Y unit' with the value 'Y'.

To hide a unit, leave the corresponding field empty.

Example: Seconds (X) and Celsius degrees (Y)



The screenshot shows a dark-themed window titled 'Units'. It contains two input fields: 'X unit' with the value 'X' and 'Y unit' with the value 'Y'.



Plotting points by using scripts

The chart object is provided with functions to interact with the object from a script.

LoadFromFile

SaveToFile

ClearCurve

ClearAll

GetX

GetY

SetX

SetY

AddPoint

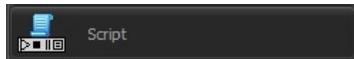
DeletePoint

CurveColor

GetCount
SetScaleLimit
CopyToClipboard
InsertLabel
ShowCurve

→ For more information about these functions, please refer to the help (menu bar), in the section *Scripts > Functions reference > Charts*. You can also find the documentation attached in annex.

Script object

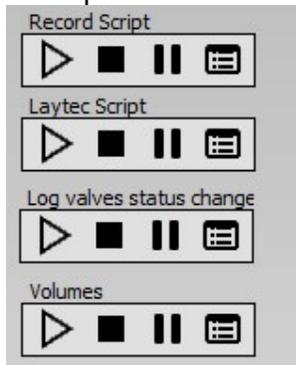


Script objects allow you to execute a specific task at a given time or permanently in the background.

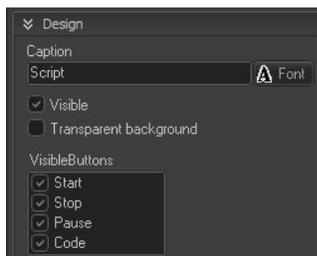
Several objects can be executed at the same time in the same form or in several forms.

To add this object on a form, you can simply select it on the component palette, hold down the mouse button and release the button on the form.

Example of several scripts in a form:



Script design



The caption of the script object is optional.

The color of the caption and other font settings can be changed by pressing the Font button. It is possible to make the object invisible by unchecking the box "Visible"

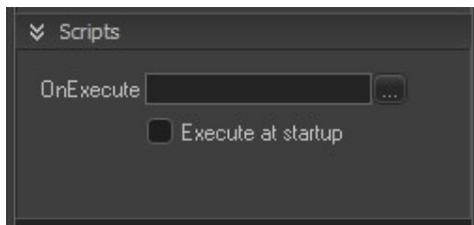
Turn on the "transparent background" to clear the background.



Some buttons can be made invisible.



Script content

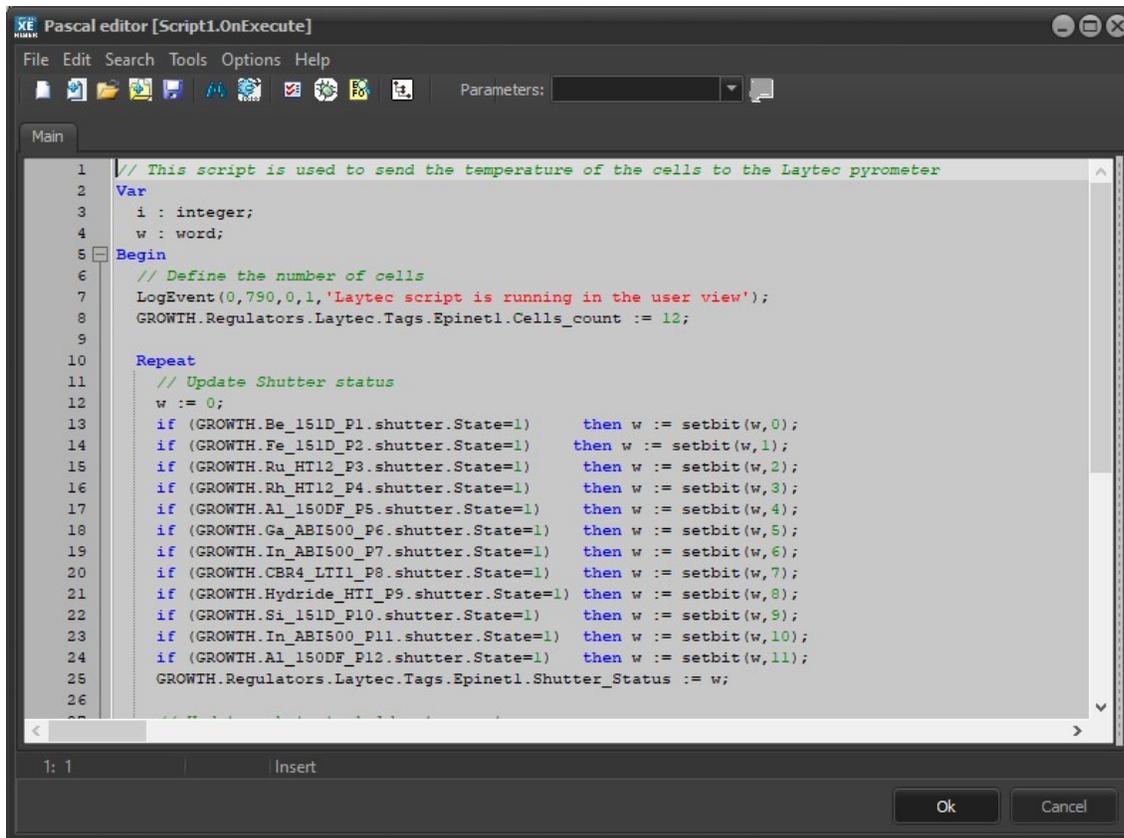


Click on the button  to edit the script.
You can also double click on the scrip object to open the script editor.

To write a script running in the background, you can use an infinite loop such as:

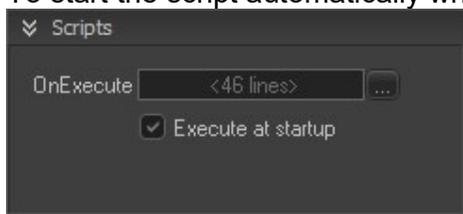
```
Repeat  
    (... code...)  
Until false;
```

Example of script:



Execute a script

To start the script automatically when the form is open, check the box “Execute at startup”

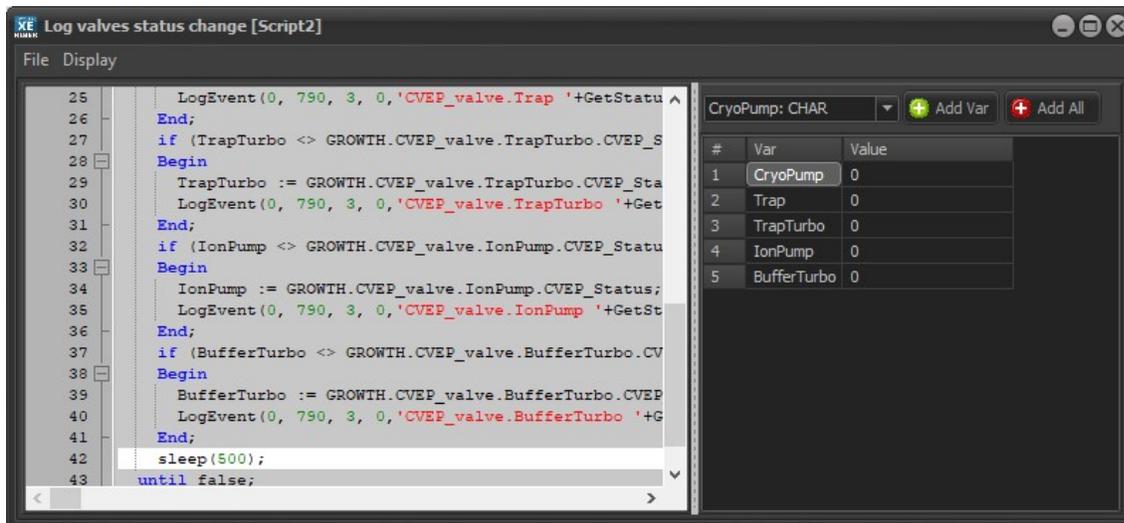


Use of the buttons:

-  Start the script.
-  Stop the script
-  Pause / Resume
-  Open the debug window

Debug a script

To open the debug window, click on the button 



Example of debug window

The content of local variables are displayed in the right part of the window. These values cannot be changed.

13.7.3. ON/OFF tab

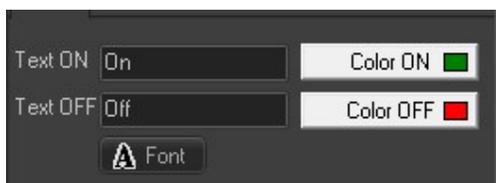
There are two types of *On/Off object*:

- Confirmation button.
- Light.

Both objects:

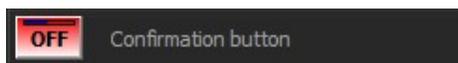
Both objects can have two different states, by default On and Off but you are free to enter what you want.

Customize the caption displayed by each state as well as the background color:

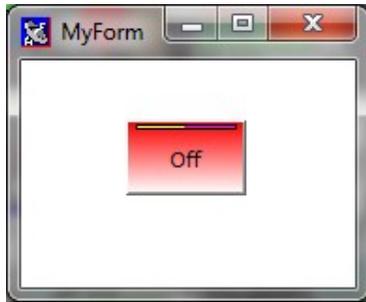


To add this object on a form, you can simply select it on the component palette, hold down the mouse button and release the button on the form.

Confirmation button



Confirmation buttons are mainly used to turn a state on or off. As a precaution, the button must be clicked twice within a certain period of time (1 second by default). When you first click on a confirmation button, a progress bar appears. Click again before the time is up.



Design

There are three ways to represent a confirmation button.

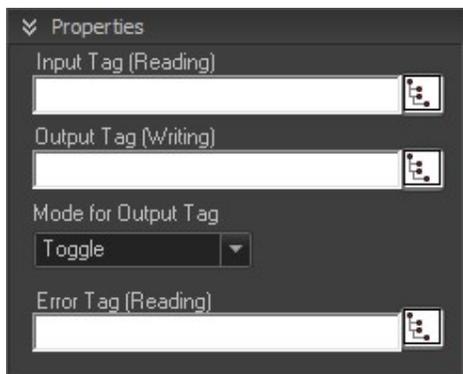


Rectangle	
Ellipse	
Custom	

Properties

On the *Properties* menu, you can set three properties:

- **Input tag:** the input tag corresponds to the *IdentInput* property value (*Advanced* tab). The button is turned On or Off according the defined property value.
- **Output tag:** the output tag corresponds to the *IdentOutput* property value (*Advanced* tab). The value is written to the specified property when the button is clicked (invert the Input value).
- **Error tag:** the error tag corresponds to the *IdentError* property value (*Advanced* tab). The button displays an alert state (grey with alert icon) according to the value of the defined property.

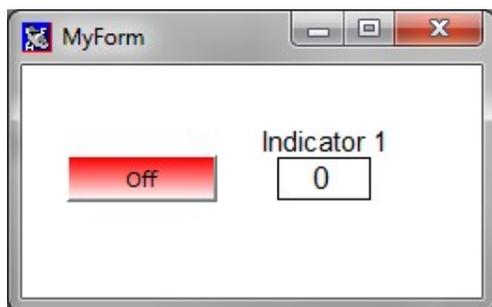
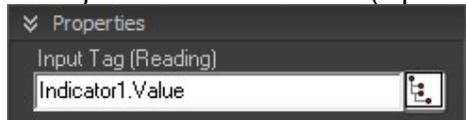


You can define same or different properties for the three tags. You can assign properties with enumeration or numeric values. The button is turned On when the value is True or not equal to zero and Off when the value is False or equal to zero.

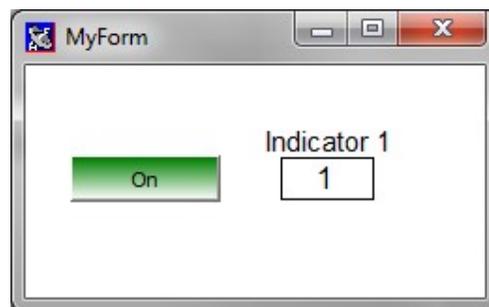
You can define the value to display by:

- **Typing the property** directly into the input field:

Example 1: Indicator value (input tag)



Indicator value = 0

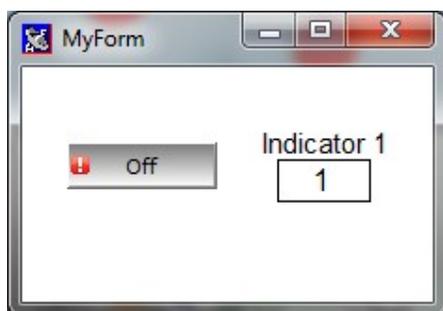
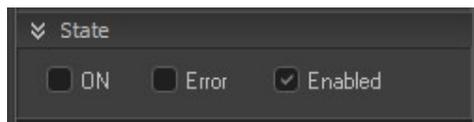


Indicator value > 0

Example 2: Indicator value (error tag)

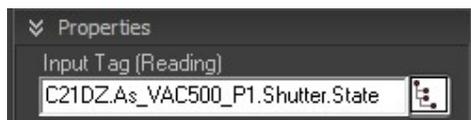


On the State menu, the *Error* box must be checked to enable Error state:

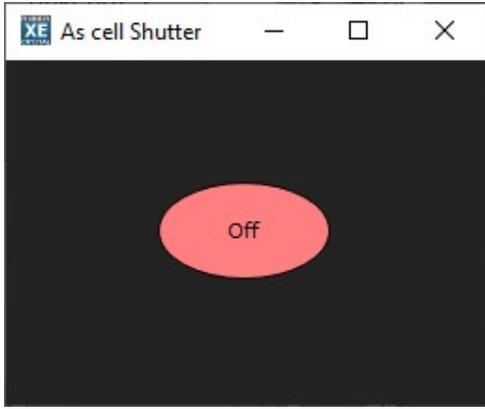


Indicator value > 0

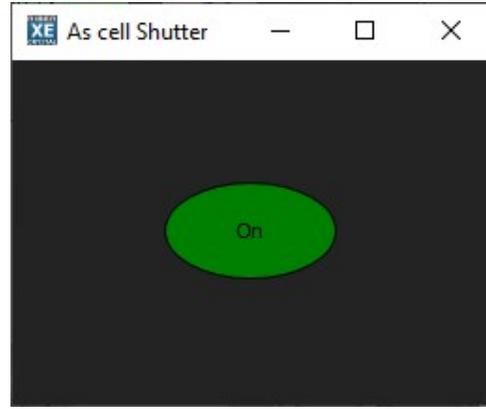
- **Selecting a property** in the *Data explorer* 



Example: State of the shutter (input)

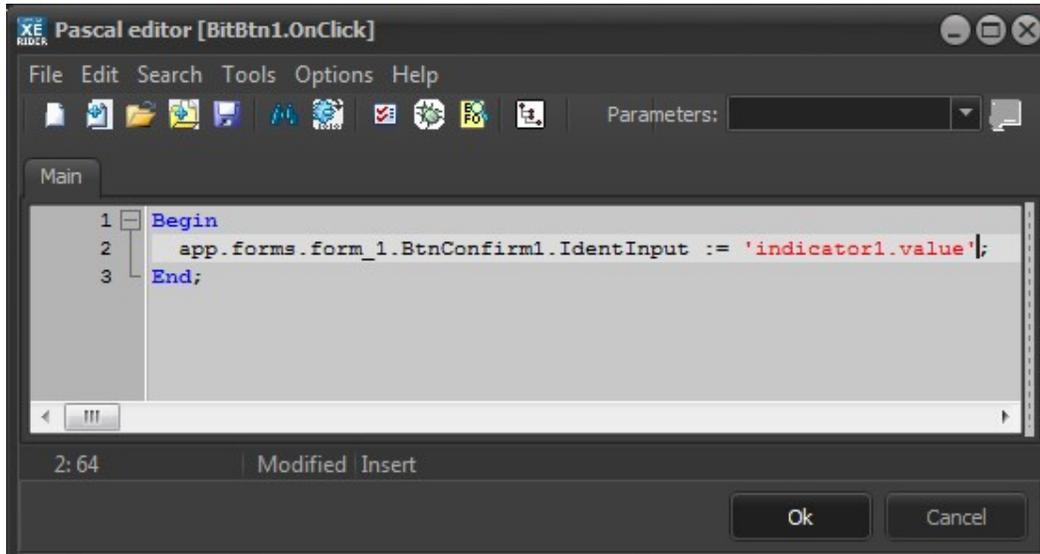


Shutter is off



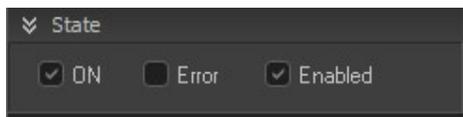
Shutter is on

- **Specifying a property** using any scripts throughout the program:



Example: indicator value (input)

State



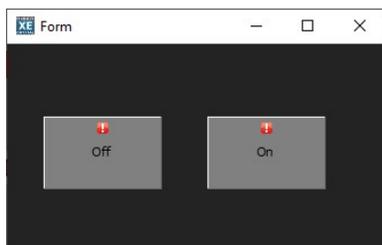
- Check the following box to display the On state and uncheck the box to display the Off state:



- Check the *Error* box to display *Error state*:



The icon turns grey and an alert icon appears (red icon), whether the button is turned On or Off:



IsError: True

- Enable/Disable the confirmation button by checking/unchecking the following box:



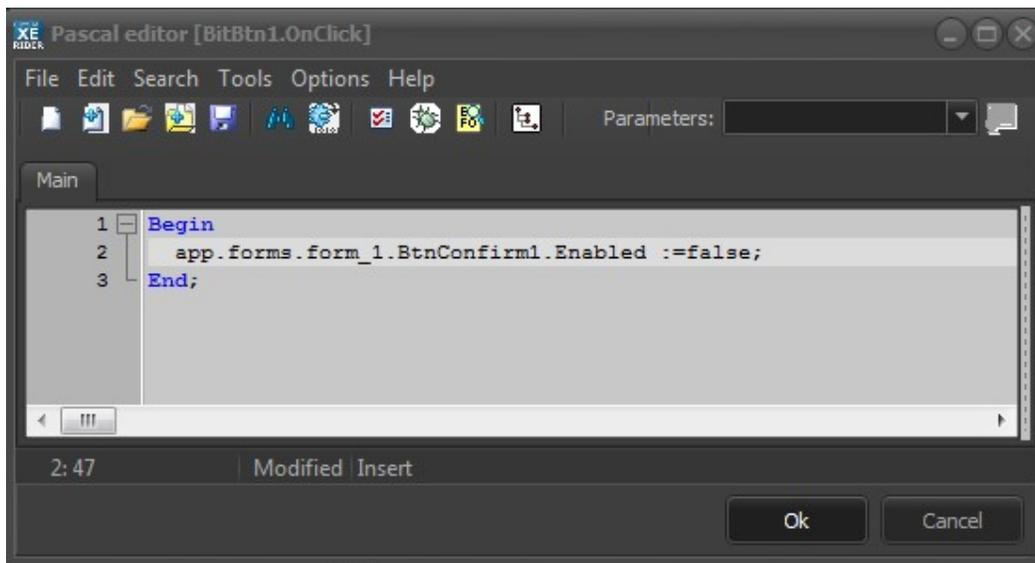
When the confirmation button is disabled, the icon turns grey, whether the button is turned On or Off:



Enabled: False

- These properties can be modified using any script throughout the program.

Example: Disabling a confirmation button



You can edit a script to be executed whenever the state turns On or Off using respectively the *OnClickOn* and *OnClickOff* events (*Advanced* tab)

LIGHT



Unlike confirmation buttons, Light objects are not designed to be interacted with. They can indicate two states (by default On and Off).

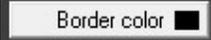
Object type: **TObjLight**

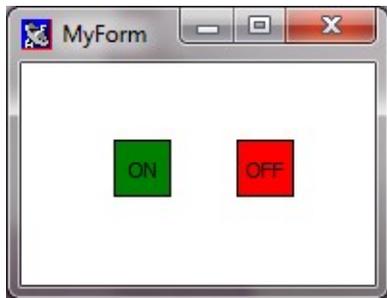
You can define one property storing enumeration or numeric value. The light is turned On when the value is *True* or not equal to zero and Off when the value is *False* or equal to zero.

Design

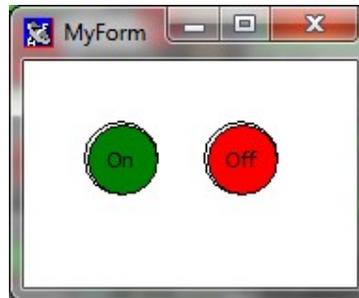


Select the desired shape between rectangles (default) or ellipses using the drop-down list.

Border color of the *Light* objects can be modified: 



Default lights (rectangles)



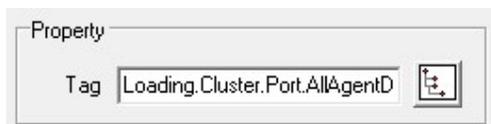
Example: shutters (ellipses)

Properties

The defined property corresponds to the IdentInput property value (Advanced tab).

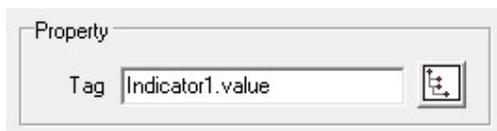
Specify the desired property:
By selecting a property in the *Data explorer*

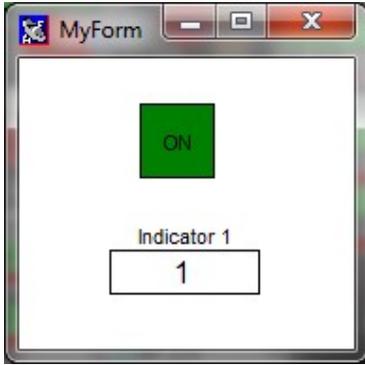
Example: Security agents state Loading.Cluster.Port.AllAgentDisabled



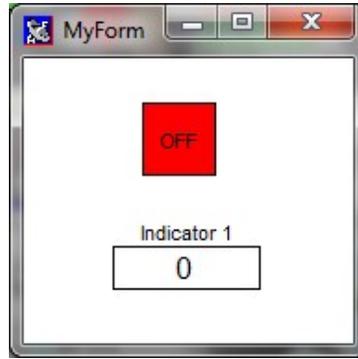
By typing a property into the input field:

Example: Indicator value





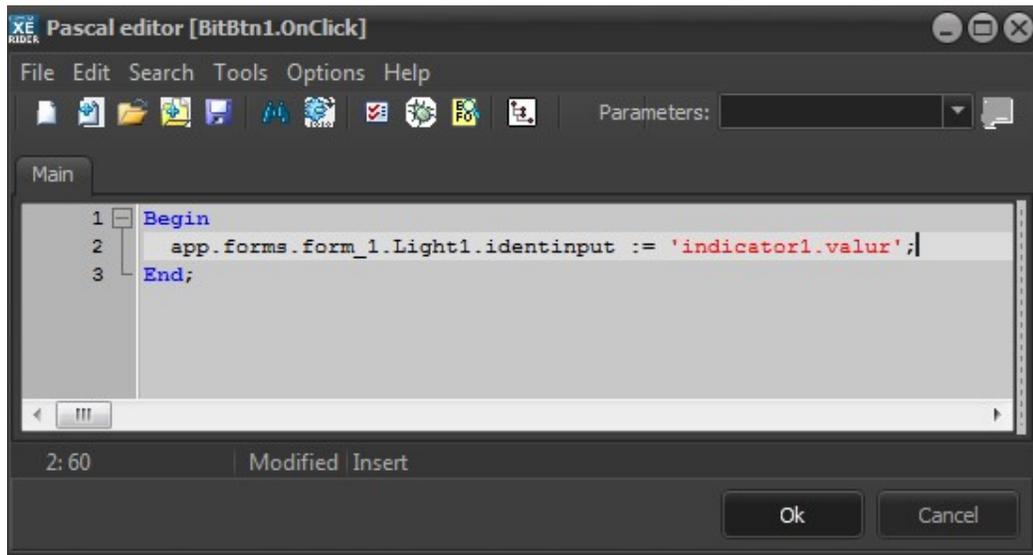
Indicator value > 0



Indicator value = 0

By specifying the property using any script throughout the program:

Example: Indicator value

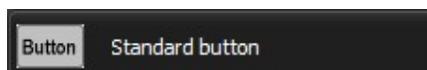


Check the following to display the On state by or uncheck the box to display the Off state:



13.7.4. Windows tab

Standard button object



Standard buttons are used to perform a certain action. The action to be performed is specified using an *OnClick* event.

Object type: **TObjBitBtn**

To add this object on a form, you can simply select it on the component palette, hold down the mouse button and release the button on the form.

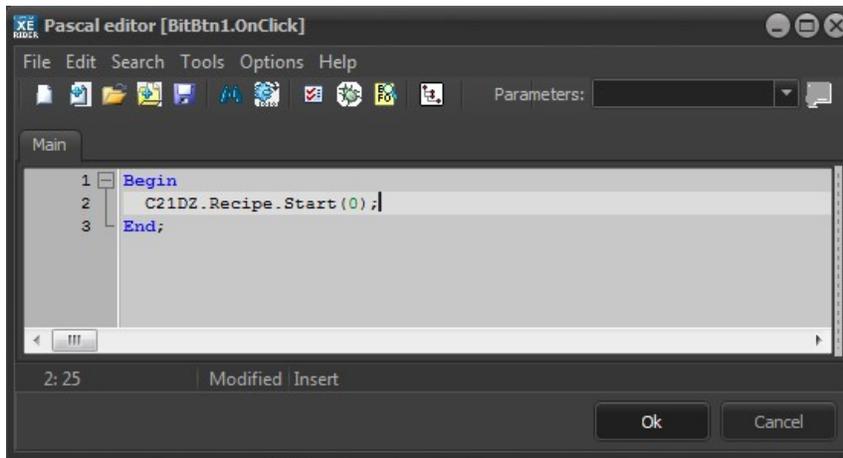
OnClick event

On the Scripts menu, click on the following icon  to edit your script in the Pascal editor:

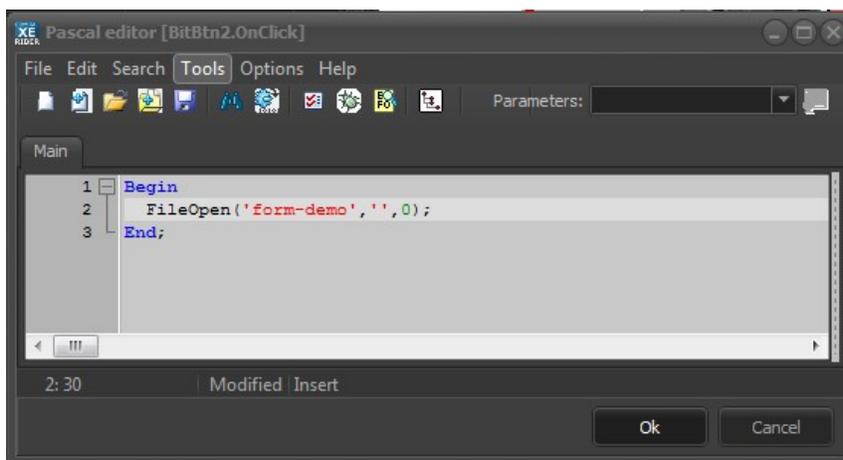


The script is executed each time the button is clicked.

Example 1: Starting a loaded recipe in the growth chamber

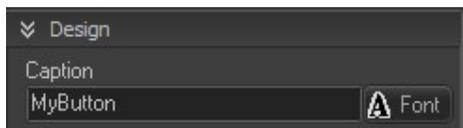


Example 2: Opening a form

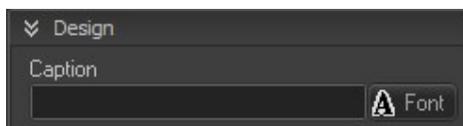


Customization

Type the **caption** of the button into the text box and customize the font:

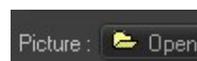


To hide the caption, leave the field empty as follows:



Buttons can have a background image:

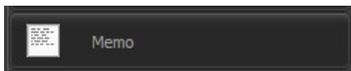
Click on the file icon to select the desired image file:



Example:



Memo object

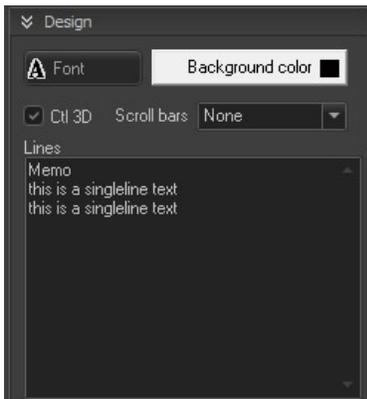


Multiline edit boxes allow the user to enter more than one line of text. They are appropriate for representing lengthy information.

To add this object on a form, you can simply select it on the component palette, hold down the mouse button and release the button on the form.

Design and content

You can change the design and the content at design time using the settings interface:



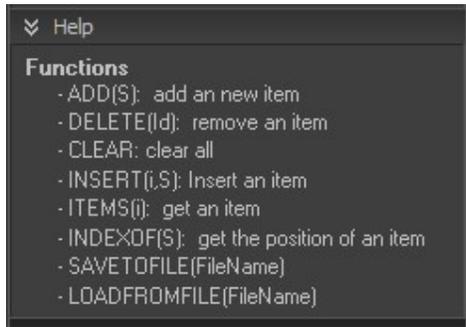
The Ctl3D option determines whether a control has a three-dimensional (3-D) or two-dimensional look.

Use of the object at runtime

In this text box, you can edit a multiline text at run time.

For example, have you added a TMemo Memo1 to your form Form1, you can use Memo1.Add('this is a singleline text'); to add a String.

List of all functions available at runtime:



See the script's online help for more information on all of these functions.

Listbox object

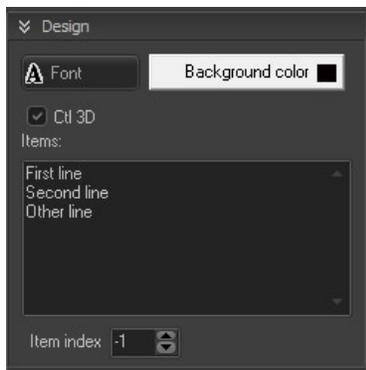


List boxes usually used to show a (scrollable) list of (short) strings and allow user to select one or more item.

To add this object on a form, you can simply select it on the component palette, hold down the mouse button and release the button on the form.

Design and content

You can change the design and the content at design time using the settings interface:

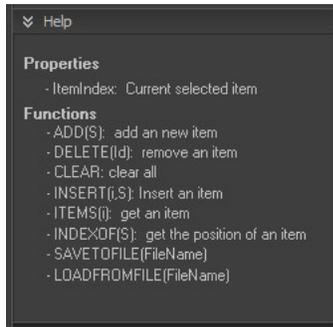


The Ctl3D option determines whether a control has a three-dimensional (3-D) or two-dimensional look.

Use of the object at runtime

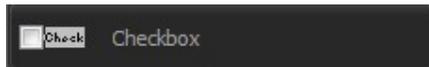
- To add a new item at run time, use the Add function.
- To know what the current selected item is, read the property ItemIndex.
- To change the current selected item at runtime, change the property ItemIndex.

List of all functions available at runtime:



See the script's online help for more information on all of these functions.

Checkbox object



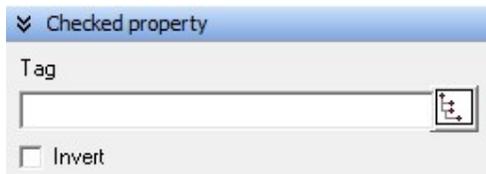
Checkboxes allow the user to choose a particular setting. Checking the box would enable the option while unchecking it would disable it. You may have several checkboxes in one form, which are independent of each other.

Object type: **TObjCheckbox**

To add this object on a form, you can simply select it on the component palette, hold down the mouse button and release the button on the form.

Checked property

Use the Check property menu to specify when the box should be checked.



By default the box is checked when the defined property value is *True* or not equal to zero and unchecked when the value is *False* or equal to zero. You can invert it by checking the following box:

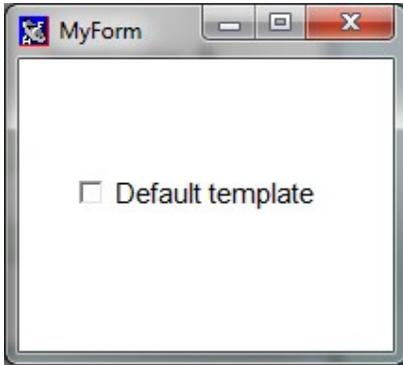


Specify the checked property by:

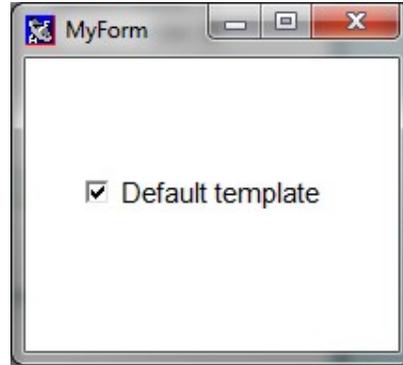
- **Selecting a property** in the *Data explorer*.

Example: Default or custom template (recorder)





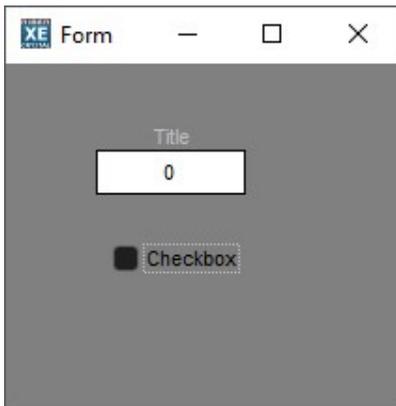
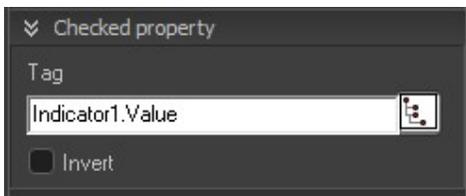
Custom template is used



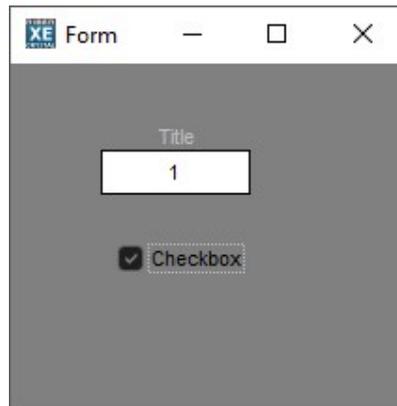
Default template is used

Typing a property directly into the input field.

Example: Indicator value



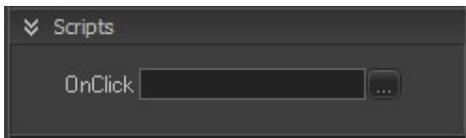
Indicator value = 0 (unchecked)



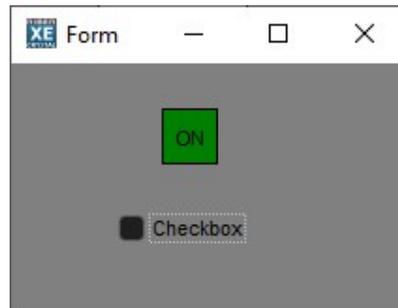
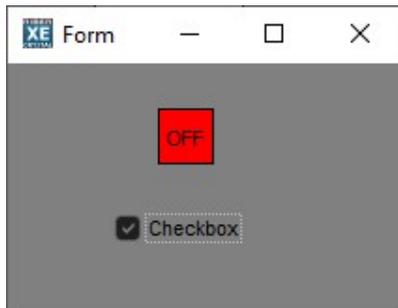
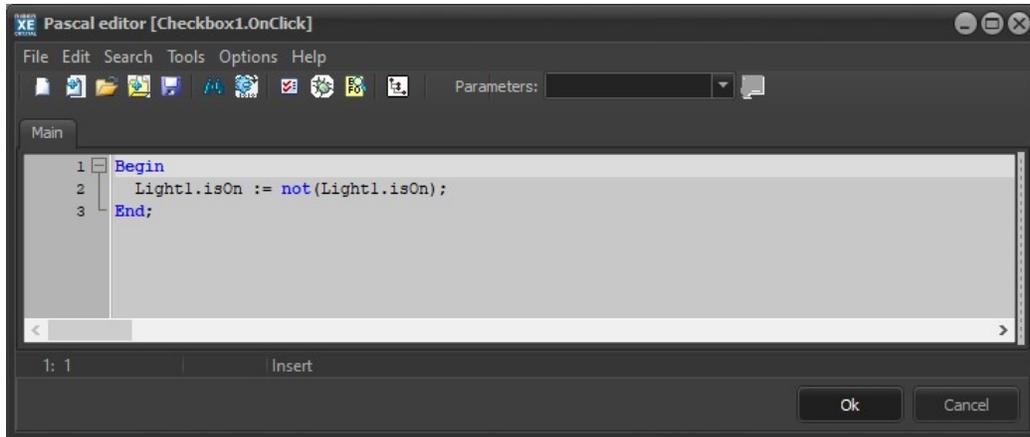
Indicator value > 0 (checked)

Script

Use the *Scripts* section to edit a script to be executed when the box is checked (*OnClick* event).



Example: Turning a Light object on



Radio button object



Radio buttons allow the user to select only one option among a predefined set of options. All radio buttons included in a form are associated with each other and only one button can be checked within one form.

Object type: **TObjRadio**

To add this object on a form, you can simply select it on the component palette, hold down the mouse button and release the button on the form.

Design

For each radio button, type the desired *caption* and customize the font:

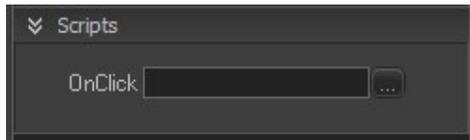


Check the following box to display the radio button checked by default:

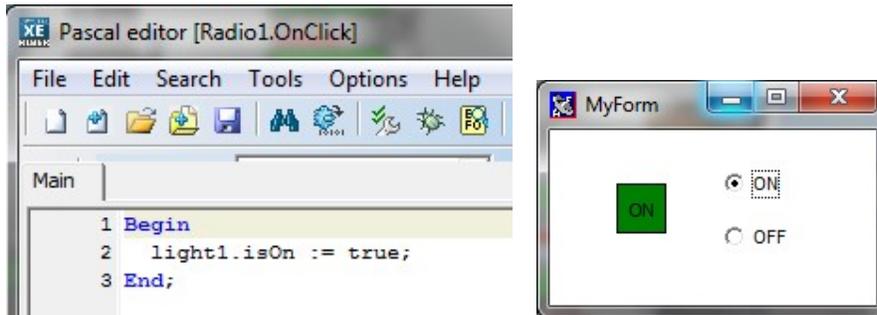


Script to execute after a click

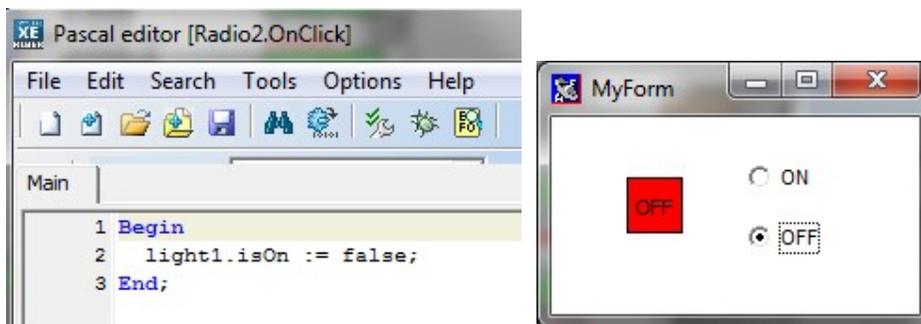
Use the Scripts menu to edit the script to be executed when the button is clicked:



Example: Turning a Light object On or Off



When Radio 1 checked the Light 1 is on



When Radio 2 is checked the Light 1 is off

Combobox object

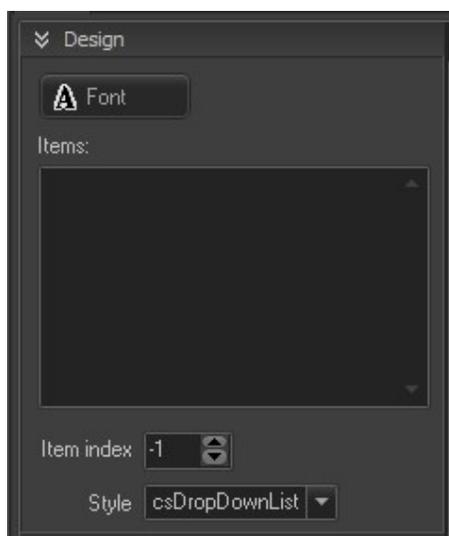


A combo box is a commonly used graphical user interface control. Traditionally, it is a combination of a drop-down list or list box and a single-line editable textbox, allowing the user to either type a value directly or select a value from the list. The term "combo box" is sometimes used to mean "drop-down list"

To add this object on a form, you can simply select it on the component palette, hold down the mouse button and release the button on the form.

Design and content

You can change the design and the content at design time using the settings interface:



Style

csDropDown	Creates a drop-down list with an edit box for manually entered text. All items are strings of the same height.	
csDropDownList	Creates a drop-down list with no edit box; the user cannot enter text manually. All items are strings of the same height.	
csSimple	Creates an edit box with a fixed list (list box) underneath. The length of the list is determined by the Height of the combo box.	

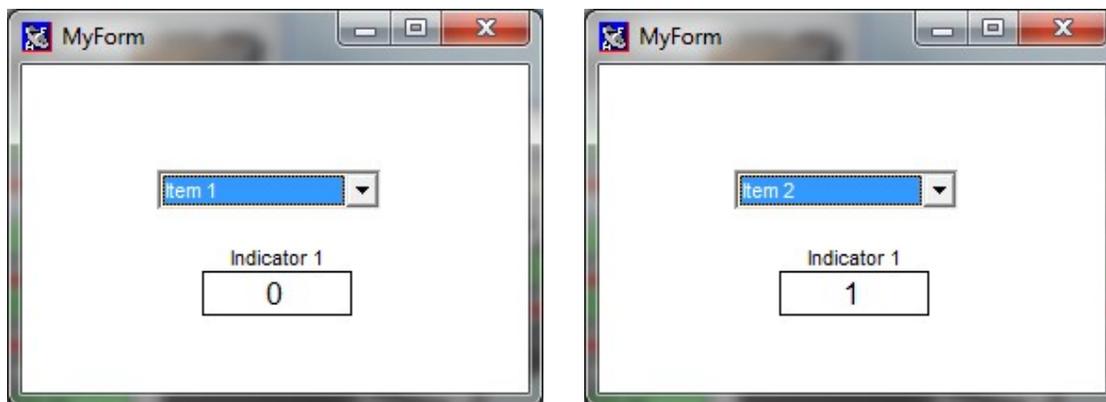
Property

Specify a property to be selected from the combo box. The value of the specified property will select the appropriate item according to its index.

Click on the following icon  to select a property in the *Data explorer* or type the desired property into the text box:

Example: Indicator value

The value of the indicator selects the items according to their indexes:



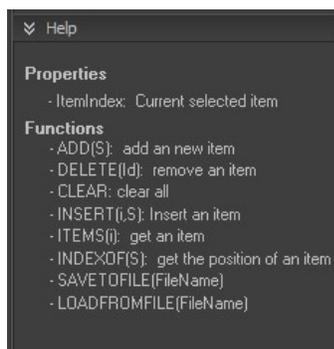
Use of the object at runtime

- To add a new item at run time, use the Add function.
- To know what the current selected item is, read the property ItemIndex.
- To change the current selected item at runtime, change the property ItemIndex.

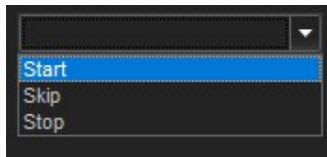
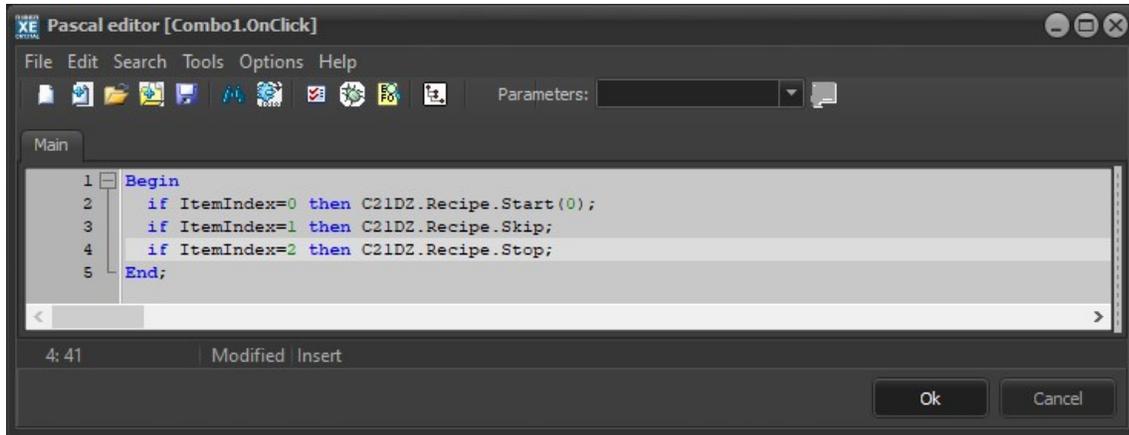


Item indexes start at 0

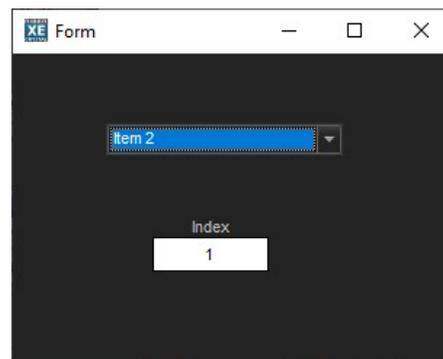
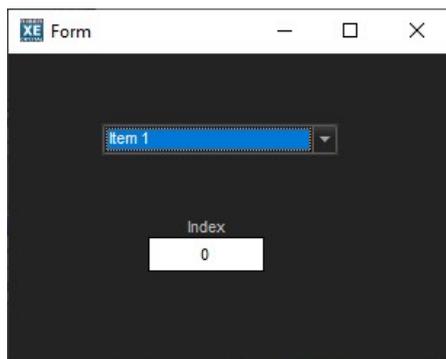
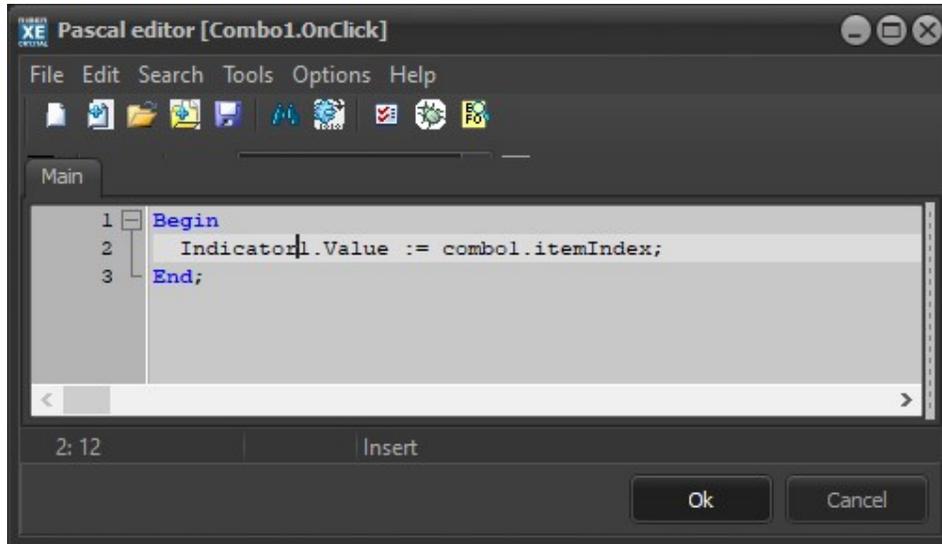
List of all functions available at runtime:



Example 1: Controlling the execution of a loaded recipe in the growth chamber

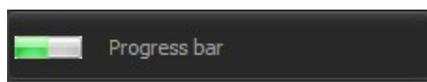


Example 2: Displaying the index of the selected item using an indicator.



See the script's online help for more information on all these functions.

Progress bar object



A progress bar is usually used as a graphical representation of a running task or a changing value. The bar is gradually filled as the task progresses or the value increases.

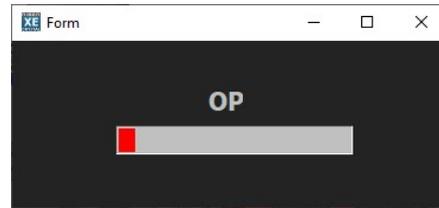
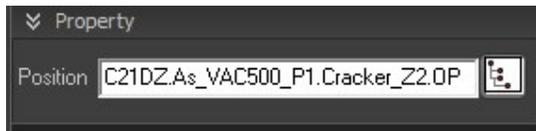
Object type: **TObjProgressBar**

To add this object on a form, you can simply select it on the component palette, hold down the mouse button and release the button on the form.

Position property

Click on the following icon  to select a property in the Data explorer

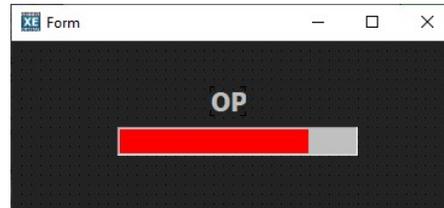
Example: Cracker output power



Cracker Output power

You can specify a static position by typing it directly into the input field:

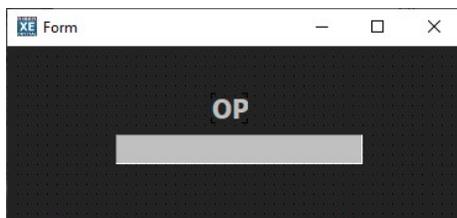
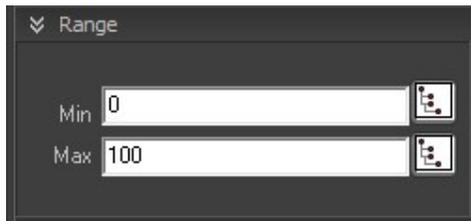
Example:



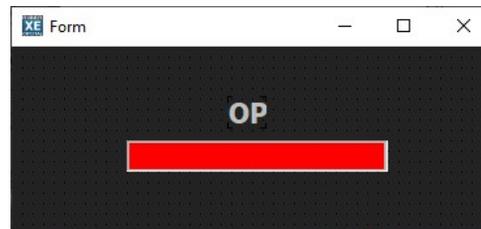
Range

Use the *Range* box to set the ranges of minimum and maximum values.

By default the bar is set in a range of 0 to 100, allowing you to visualize values expressed as a percentage, such as output power.



0



100

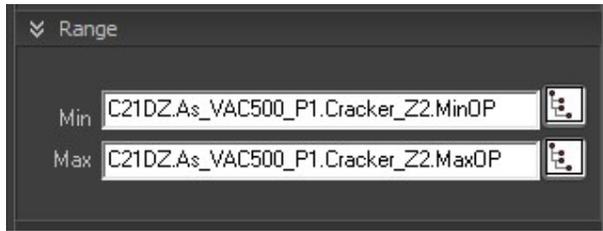
You can also enter a property for the field min and max.

Example: Cracker output power *min OP* and *max OP*

This can be used to define the minimum and maximum output power of a temperature regulator.

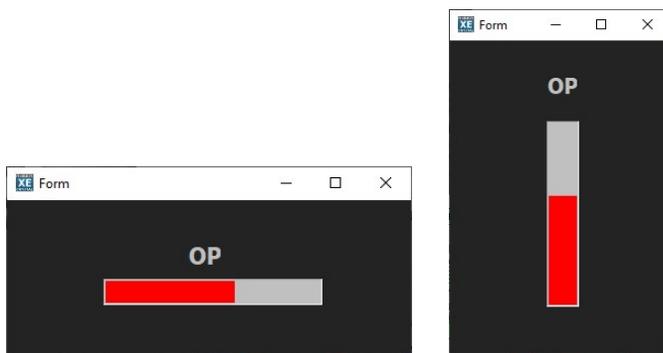
Min = C21DZ.As_VAC500_P1.Cracker_Z2.MinOP

Max = C21DZ.As_VAC500_P1.Cracker_Z2.MaxOP



Customization

Select either horizontal or vertical bars using the drop-down list:



Horizontal

Vertical

Select the desired colors:

- progress color: Bar color

- background color: Bk gr color

13.7.5. OTHERS

Image object

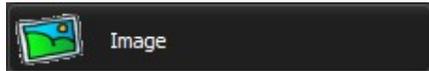
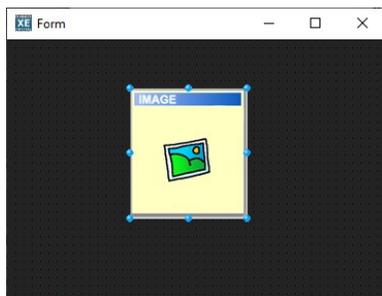


Image objects allow you to display different types of image files, which can be moved around the entire form like any other object. You can interact with image objects by specifying events (Advanced tab).

Object type: **TObjImage**

To add this object on a form, you can simply select it on the component palette, hold down the mouse button and release the button on the form.

Drag and drop the image object to the form:



Load an image

In the tool palette, click on Open to select the desired image file:  The possible file formats are: jpg, jpeg, bmp, png, ico, emf, wmf and gif.

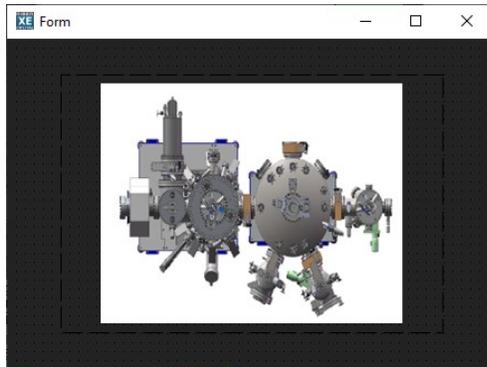
Stretch

When the Stretch box is checked  the image shrinks or extends to adapt the size of the object.

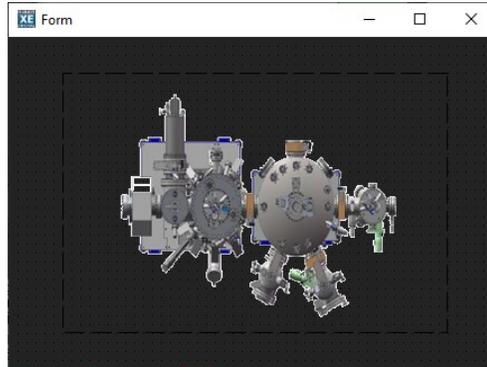
Transparent background

Check Transparent background to hide the image background

Transparent background (bottom left pixel)



Default image



Transparent background

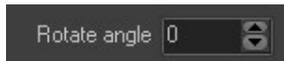


Transparency is only available with bitmap images (.bmp)

Rotation

Rotation can be define at design time or at runtime.

At the time of conception, modify the rotate angle in degree.

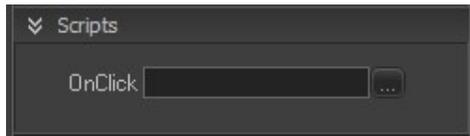


By using the script, modify the **RotateAngle** property.

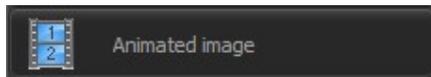
For example, this can be used to represent a rotating substrate holder.

Script to execute after a click

Use the Scripts pane to edit the script to be executed when the object is clicked:



Animated image object



Animated image objects or multiple image objects can display one picture from a list or make an animation with several pictures.

Object type: **TObjImageMultiple**

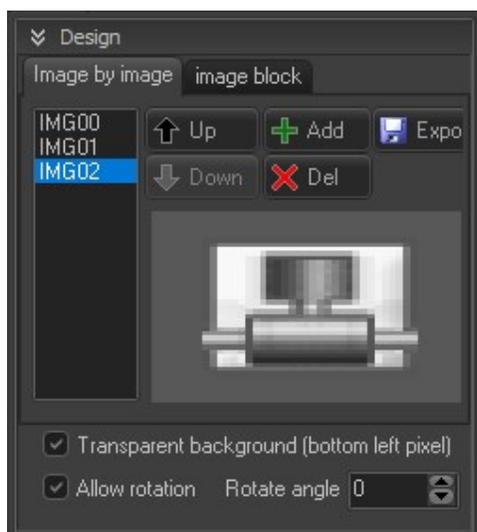
To add this object on a form, you can simply select it on the component palette, hold down the mouse button and release the button on the form.

Two modes of design are possible.

- With a list of images
- With a single image (image block) that is split into several parts.

Design with a list of images

In the Tool palette, click on the tab “Image by image”



The list contains all the images to display. The current image to be displayed is indicated by the "ImageIndex" property.

Use the Add button, to add new image in the list, Del to delete an image, Up and Down to move it and Export to export the selected image to a file.

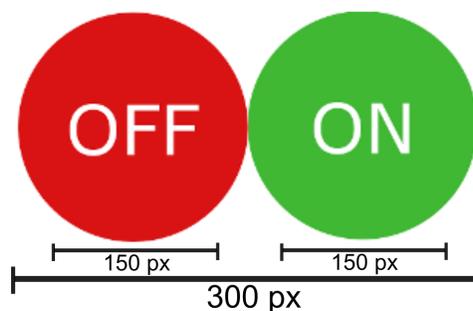
All the image must be the same size. If you add an image of a different size, it will be resized.

Design with a single image

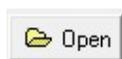
The image can be divided into several images, usually representing a state (shutter state, recipe state, etc.). You can switch from one image to another by setting a property that will select the appropriate index. Specify events to allow users to interact with animated images (*Advanced* tab).

Example: Animating On and Off buttons

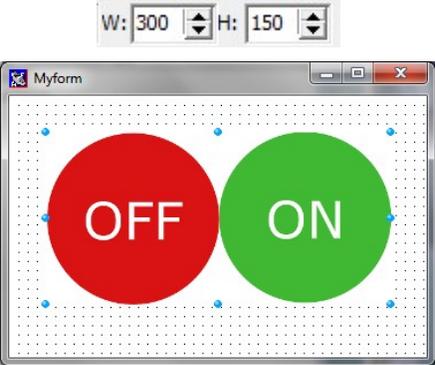
Create an image composed of two buttons (equal size):



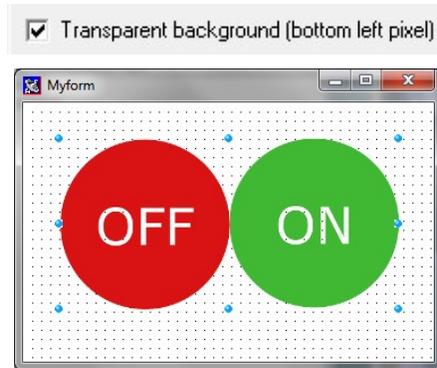
Click on the Open button to select your image file:



If you want to view the whole image first, use the panel at the bottom of the interface to adjust the width (W) and height (H) corresponding to your image.



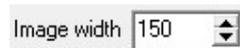
You can hide the image background by checking *Transparent background*:



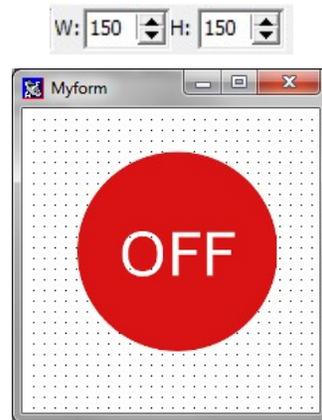
Select Horizontal images if your images are aligned horizontally or Vertical images if they are aligned vertically. In this example select Horizontal images.



Set the width of your separate images. In this example, each button measures 150 pixels:



Adjust the size accordingly using the panel at the bottom of the interface:



Only the first image should be displayed.

Display settings

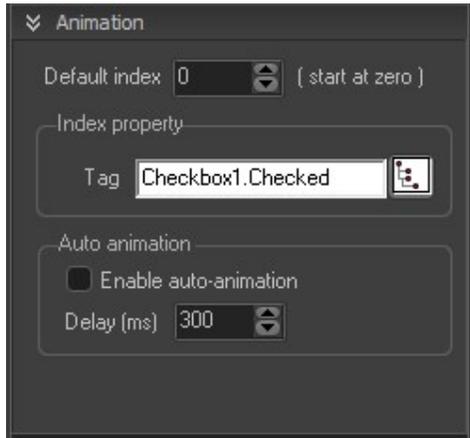


Check the "Transparent background" box to use the color identified by the lower left pixel of the image as transparent color.



Check the "Allow rotation" box if you want to rotate the image. The size of the image will be reduced to fit in the rectangle of the object during rotation.

Animation



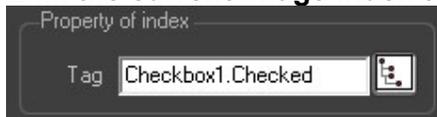
On the *Animation* pane, adjust the default index using the spinner to specify the image to be displayed by default. The first image corresponds to index 0, the second to index 2 and so on.

Selection of the current image (optional):



This is only useful if no property is defined for "IdentInput" (Property of index) and the auto-animation is not enabled.

Link the current image index to another property (optional):



If you define a tag for the "index property", the current image to be displayed will be represented by the content of this tag.

Click on the following icon  to open the Data explorer and define a **property** for the animated image:

Example: Shutter state property



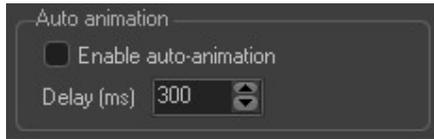
Index 0 (Shutter is off)



Index 1 (shutter is on)

Images change when the shutter is on or off

Auto animation



Check the "Activate automatic animation" box and define a delay in ms to periodically change the image.

Activation or deactivation can be activated by a script on any event.

In auto-animation mode, no Tag property are required to be set. The images will switch automatically at the defined delay.

Use the Scripts menu to edit a **script** to be executed when the user clicks on the image (OnClick event).

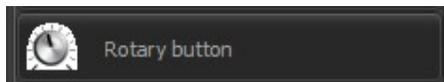


Example of use:

An animated image can be used to display the **status of a valve**. In that case, three images can be defined: valve red (closed), valve green (open) and valve gray (undefined).

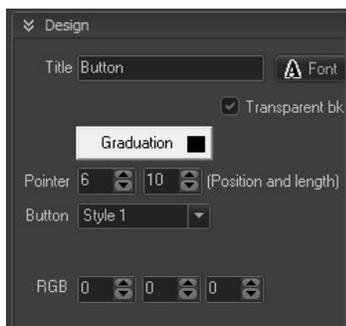
When users click on the image, the valve state is changed.

Rotary button



The rotary button object allows you to modify any value from a predefined minimum value to a predefined maximum value.

Design



The text of the title and the font use for the title can be modified here.



5 default styles are available

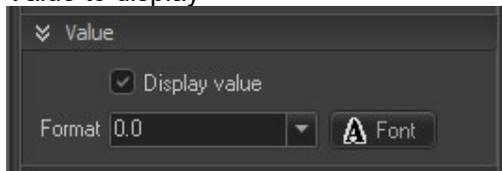
Style 1	
Style 2	
Style 3	
Style 4	
Style 5	

By selecting the "Custom" style, it is possible to choose a custom image.



Several file format are available: jpg, jpeg, bmp, emf and wmf.

Value to display



By activating the "Display value" box, the value will be displayed below the rotary button. It is possible to define the display format and to select the font of the value.

Display range



The display range can be either predefined using the min and max constants, or defined by tag properties. To select a tag, click on the data explorer button.

Wheel



This defines the pitch of the mouse wheel when the cursor is located on the object.

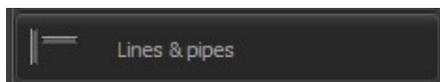
Position



The position can be either a constant value or a tag property.

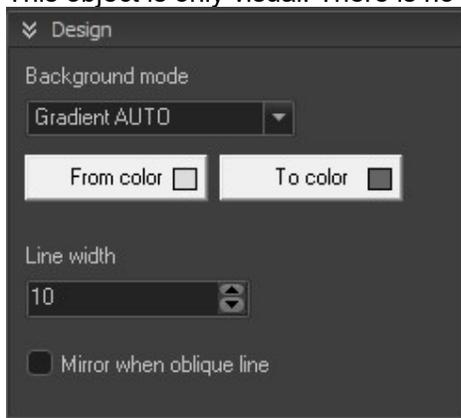
If you choose a tag, the position of the rotary button will be defined by the value contained in the tag. If the value of the tag changes, the position of the button will automatically change.

Lines and pipes



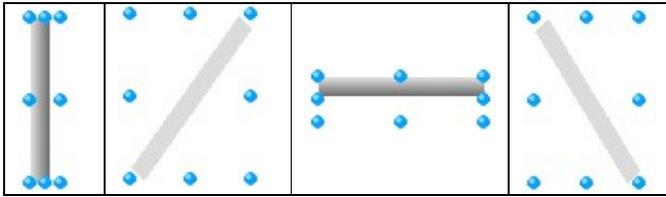
Design

This object is only visual. There is no interaction with the user.



The color and other properties can be changed by using the scripts.

The orientation is defined when designing the form. It depends of the ratio of the rectangle.



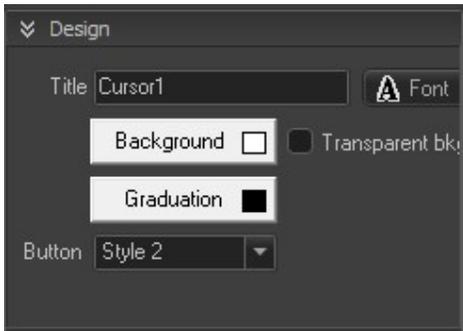
This figure shows the orientation when designing the form

Cursor



The cursor object allows to modify any value using a "mixer" type cursor

Design



The text of the title and the font use for the title can be modified in the design pane. Colors can also be changed in this pane and it is possible to make the background transparent.

Several styles of button are provided:

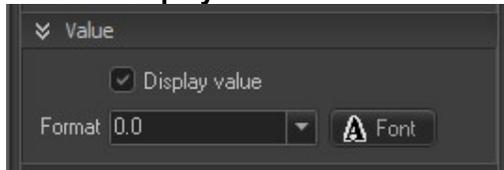
Style 1	
Style 2	
Style 3	

Orientation

The orientation is automatic when designing the form. Just resize the object to change the orientation.

Horizontal	
Vertical	

Value to display



By activating the "Display value" box, the value will be displayed below the object. It is possible to define the display format and to select the font of the value.

Display range



The display range can be either predefined using the min and max constants, or dynamical defined by tag properties. To select a tag, click on the data explorer button.

Wheel



This defines the pitch of the mouse wheel when the cursor is located on the object.

Position



The position can be either a constant value or a tag property.

If you choose a tag, the position of the cursor will be defined by the value contained in the tag. If the value of the tag changes, the cursor position will automatically change.

Window splitter



A splitter is an object that can be placed on a form as vertical or horizontal bar to separate sub-panels functionally.

The control can be used as a visual separator between two halves of your form and allows the user of your application to move it either vertically or horizontally.

When the form is divided into two parts, only two objects should be used to separate the form.

One object must be aligned on one side and the other must be aligned throughout the customer area (client area).

Example with two charts and vertical splitter:



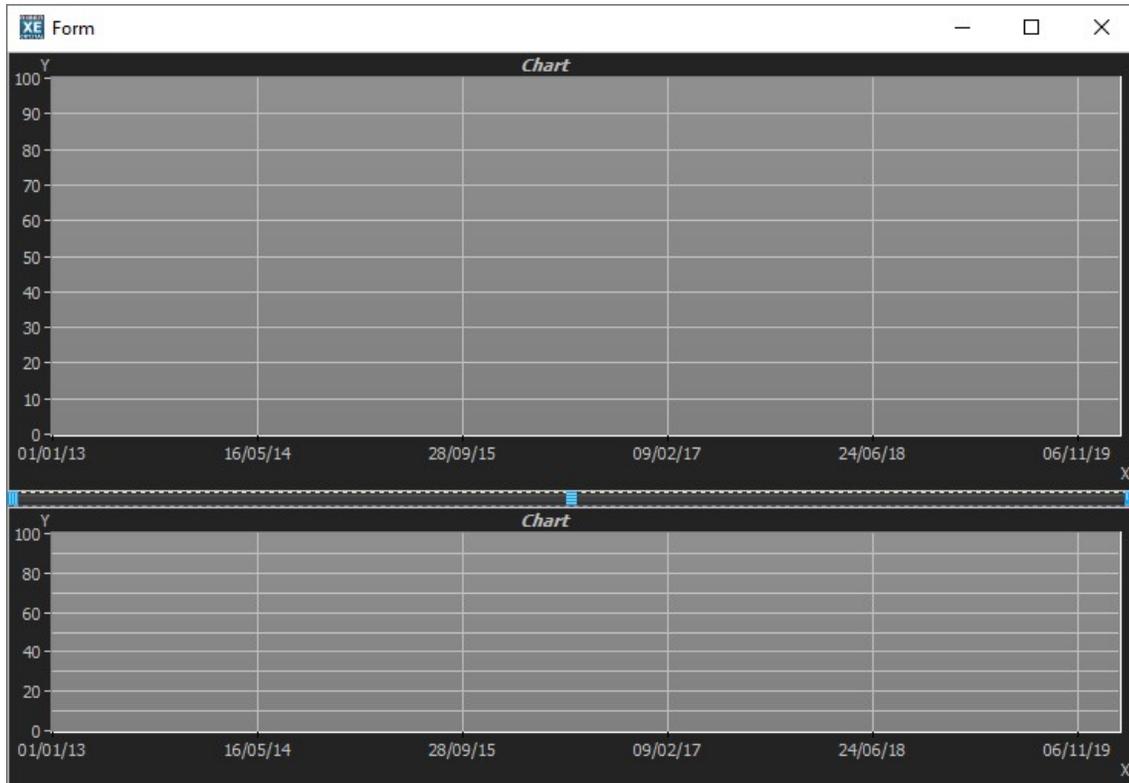
In this example, the splitter divides the form into two parts. A left part and a right part. The vertical divider can be moved from left to right.

How to design this form ?

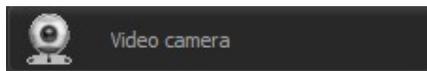
- 1) Create a new form
- 2) Add a chart in the form
- 3) Change the property "align" of the chart to "alLeft"
- 4) Add a splitter to the form
- 5) Add another chart to the form.
- 6) Change the property "align" of the chart to "alClient"

The width of the splitter can also be modified in the properties tab of the Tool palette.

Another example with a horizontal splitter:



Video camera object



Video camera objects allow displaying, in real time, video images being filmed by a video camera connected to your computer.

You can connect several devices and then several video objects.

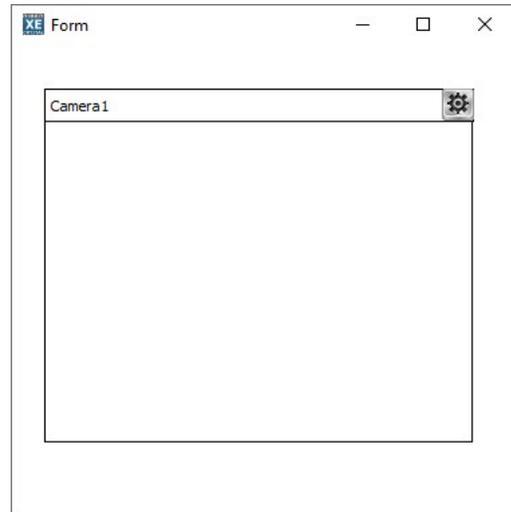


You can only connect webcams recognized by your Windows system.

Object type: **TObjCamera**

Quick start

To add this object on a form, you can simply select it on the component palette, hold down the mouse button and release the button on the form.



Drag and drop the Video camera object to the form:

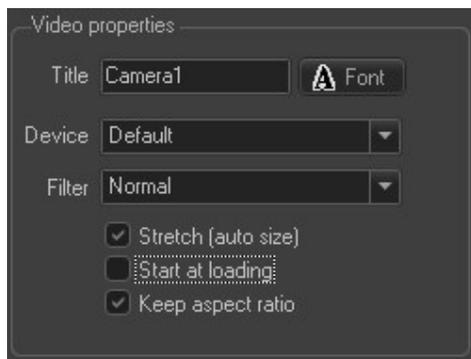
Then, check the box “Start at loading”



Exit the edition of the form.



Video properties



Customize the camera title and the title font.



Select the desired device in the drop-down list:



The list displays all camera devices connected to your computer and recognized by *Windows*. *Default* device corresponds to the device set as default by your *Windows* system.

When *Stretch* is checked, the video automatically shrinks or extends to fit the object size.



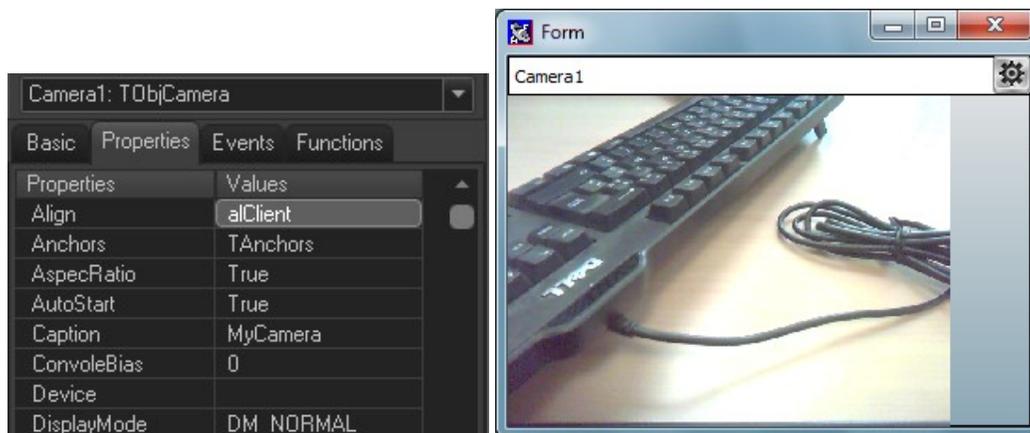
When *Start at loading* is checked, the video starts automatically after exiting the design of the form and after loading the form:



When *Keep aspect ratio* is checked, the width and height proportions are constrained to maintain the aspect ratio of the video camera and avoid distortion of the image:

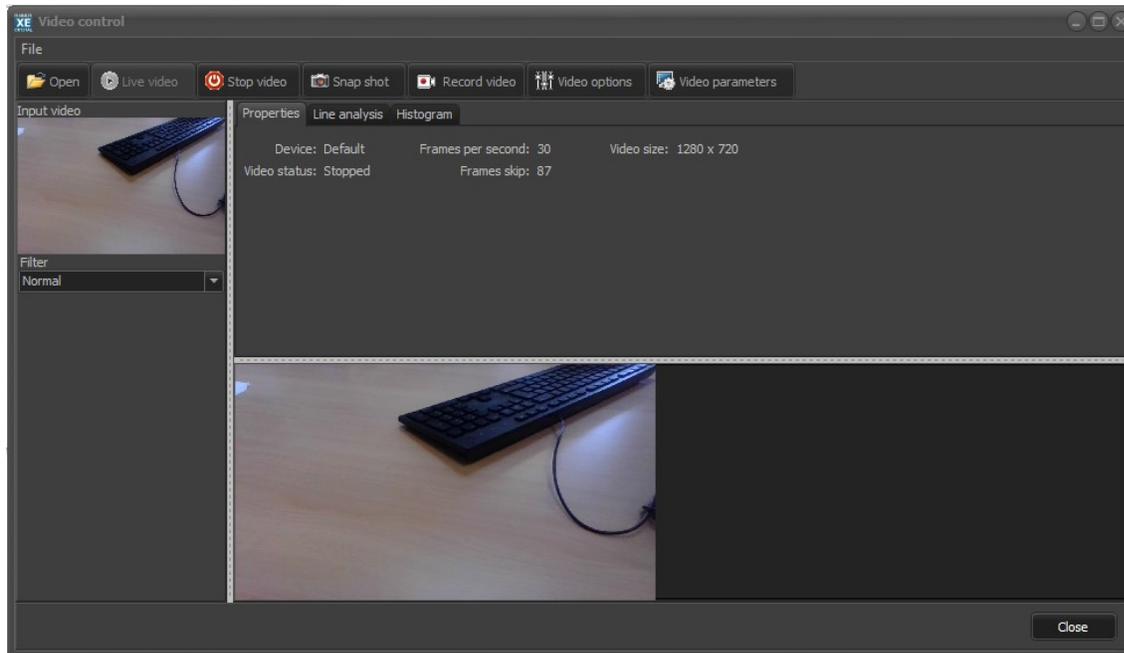


To make the video size filling the window, go in the advanced tab, then change the property *Align to alClient*.



Video control panel

At runtime, click on the following icon , located in the top left-hand corner of the Video camera object, to display the video control panel:



Several filters are provided with the video object.

Open button

You can apply a filter to a still image or a live video image.

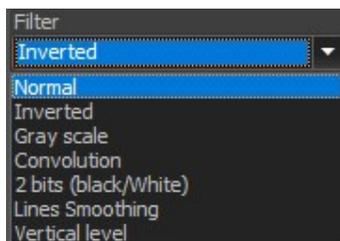
To apply a filter to a still image, stop the video and press on the Open button. Several types of image files are possible: bmp, jpg, png, gif, tif and tiff.

Live video

To start the video live, just push on the Live video button.

The selected filter will be applied and displayed in the central video output.

The following filters are available:

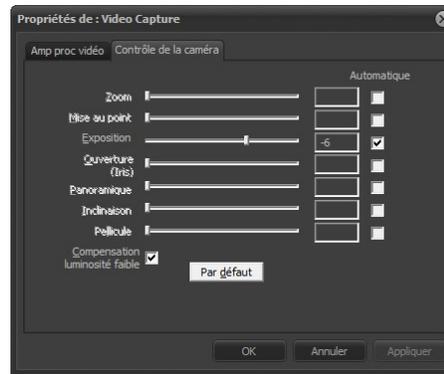
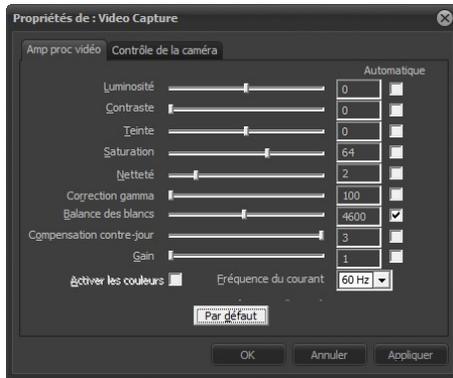


Snapshot: Click on the Snapshot button to save the frame into a bitmap file in the record sub directory of the project.

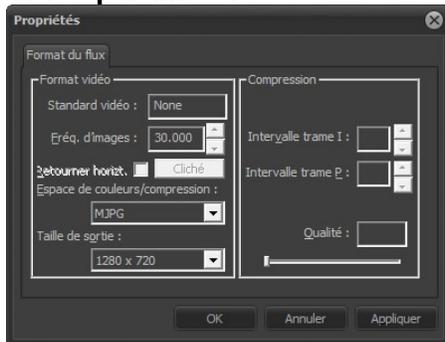
Record video

The video is recorded in a .AVI file. During video acquisition all the image are not displayed to the screen.

Video options

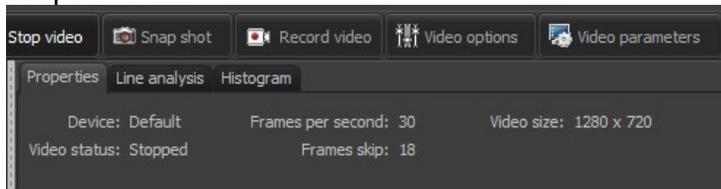


Video parameters



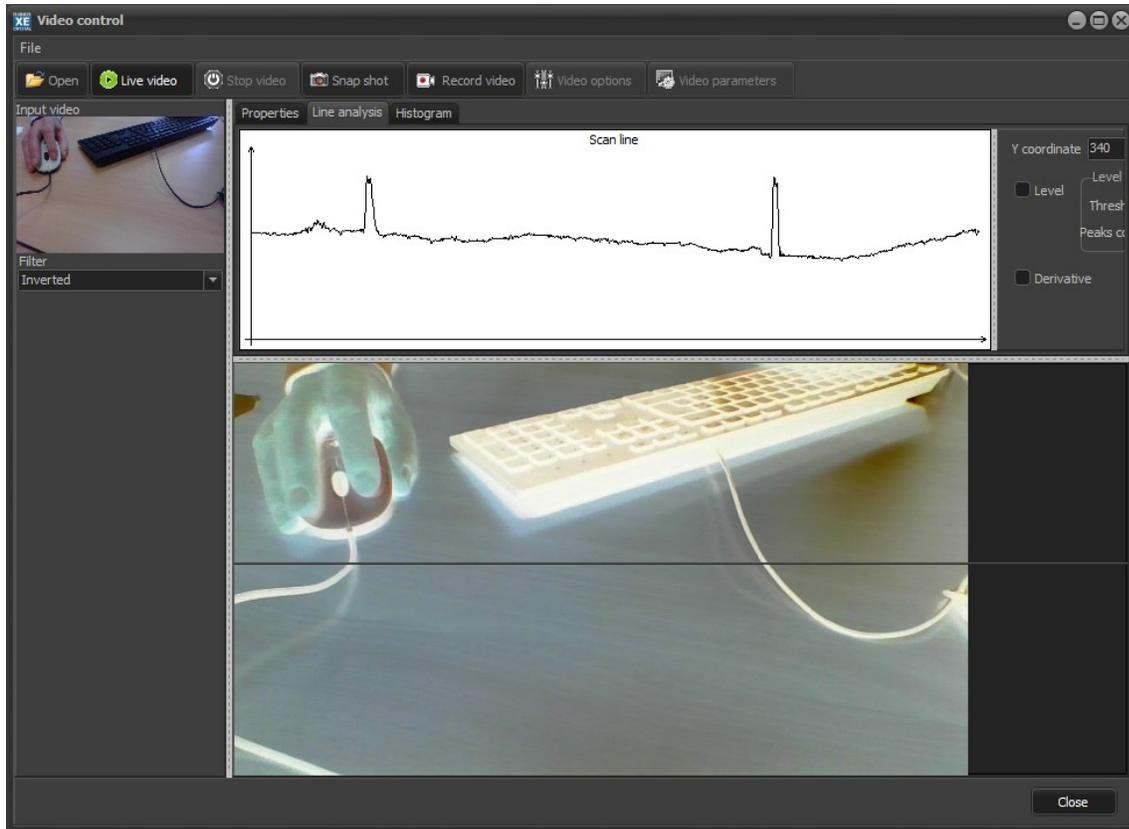
Video analysis

Properties tab



Here the data are given as an indication.

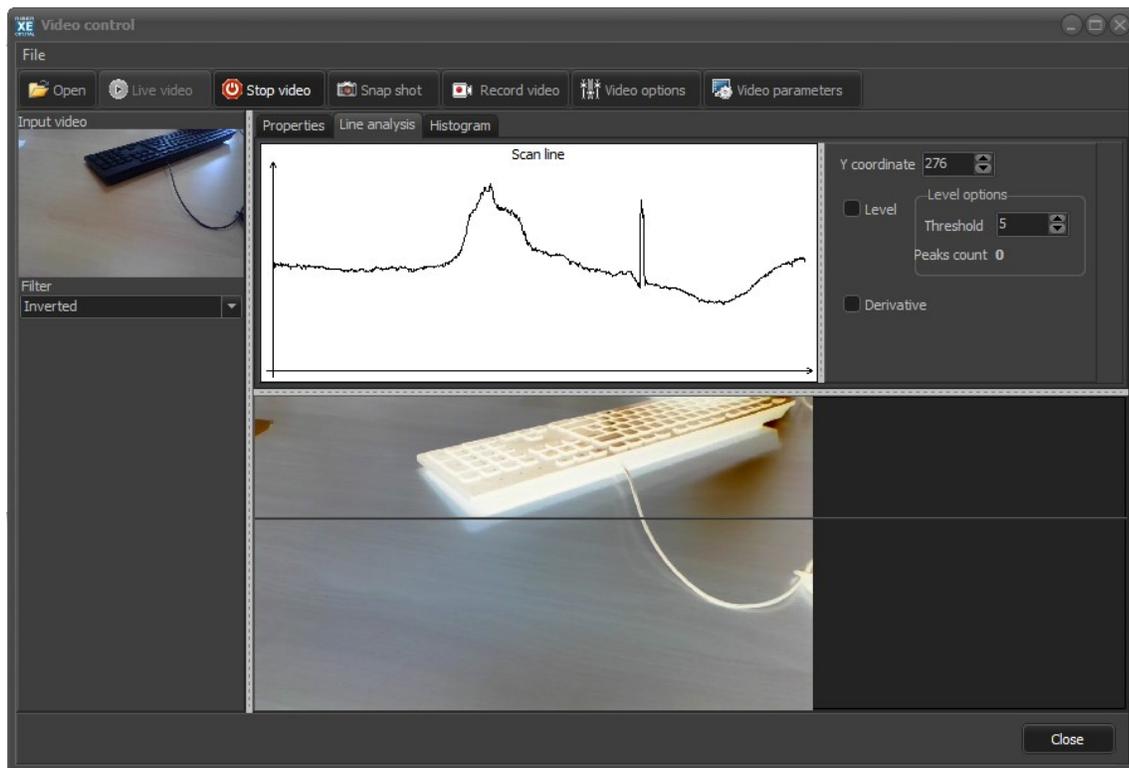
The frame skipped information is an image counter that has not been processed since the start of the video display. To reset this counter to zero, stop the video and restart it.



Line analysis

A scan line is a single row of pixels in the video image. The intensity of each pixel is displayed in a chart. Click on the central video and hold down the right mouse button to adjust the horizontal line with the mouse. The graphic just above the video shows the scan line for the selected line.

See the screen shot bellow:



Histogram

An **image histogram** is a type of **histogram** that acts as a graphical representation of the tonal distribution in a digital **image**. It plots the number of pixels for each tonal value. By looking at the **histogram** for a specific **image** a viewer will be able to judge the entire tonal distribution at a glance.

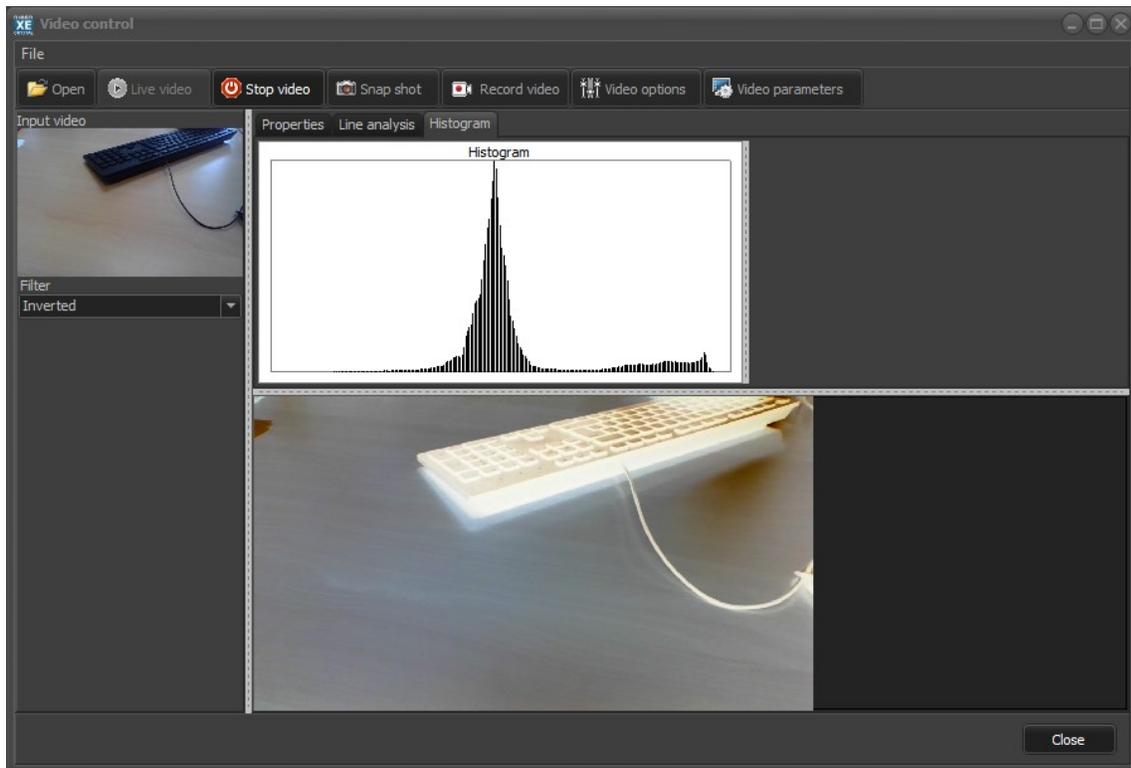
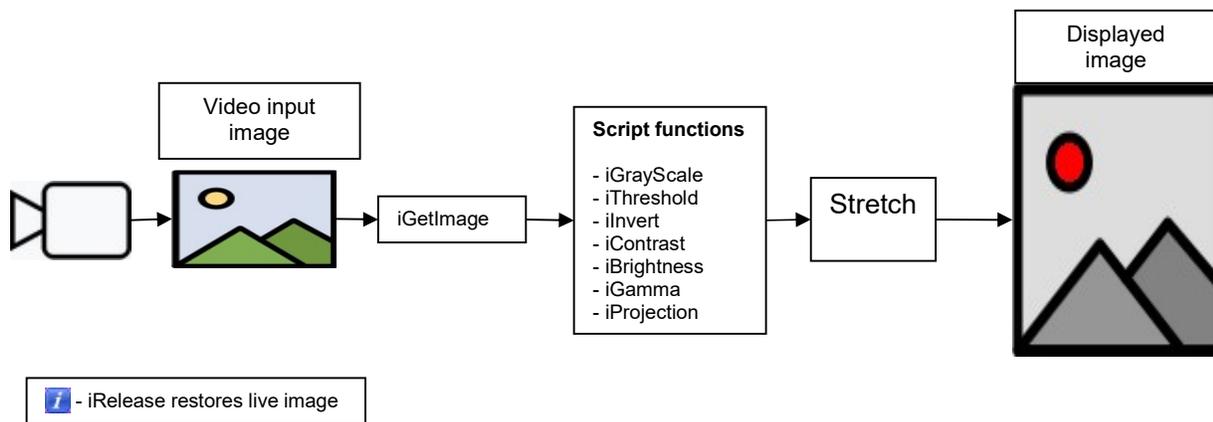
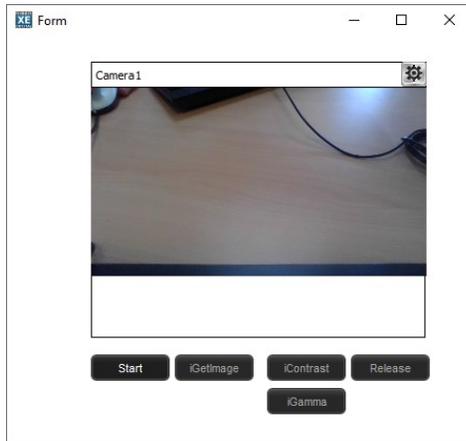


Image processing using scripts



Example of form:



In this example, push on the Start button (the script calls the function “Start”) and wait for the image to be displayed.

Push on the iGetimage, this will freeze the image.

Push on iContrast, this will execute the script: Camera1.icontrast(2000);



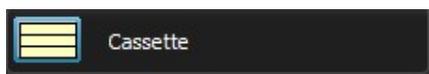
Display after iConstrat

Push on the button iGamma will modify again the picture.

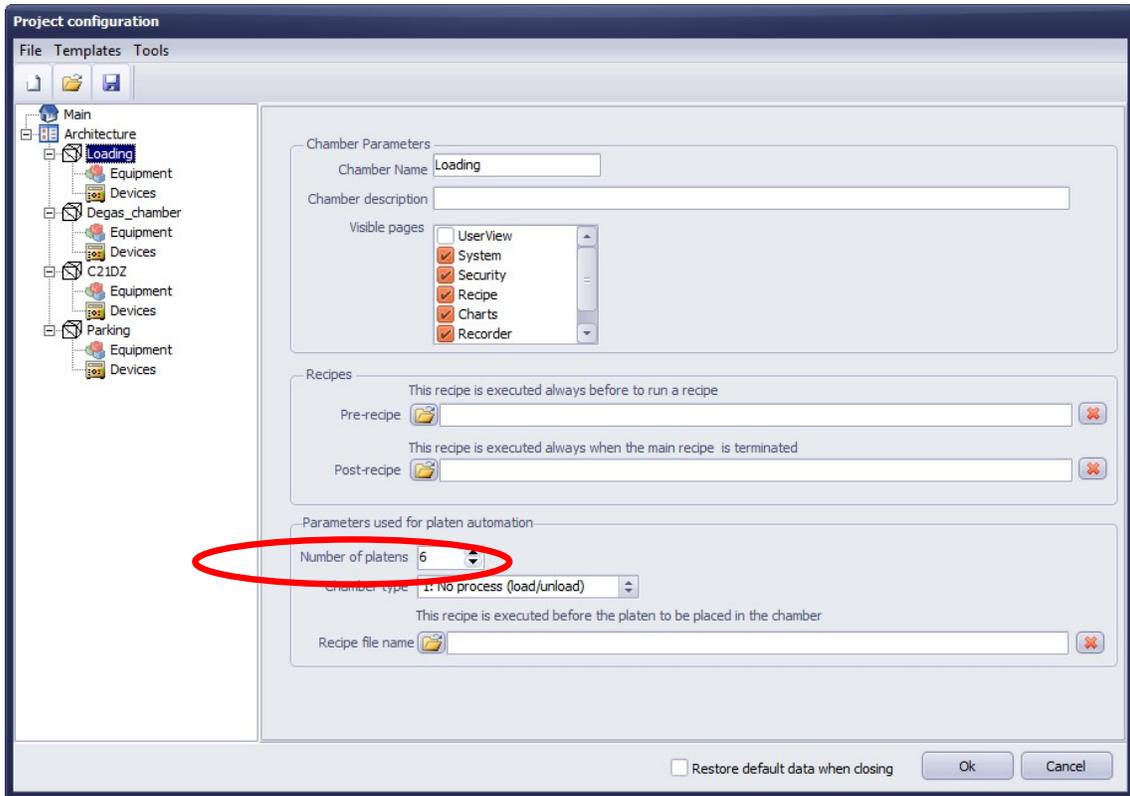
Push on the button iRelease to restore the live video.

For more details on these functions, refer to the script online help.

Cassette object



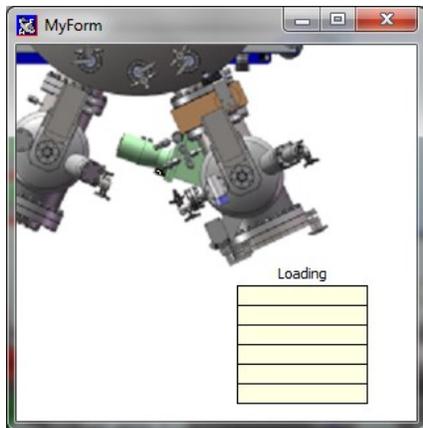
Each cassette object is associated with a chamber. *Crystal XE* automatically displays the numbers of the platens defined in the hardware configuration.



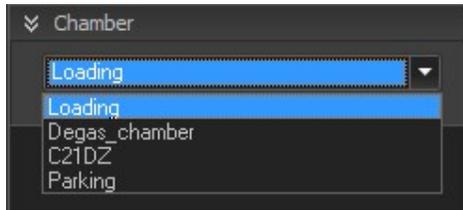
Object type: TObjCassette

To add this object on a form, you can simply select it on the component palette, hold down the mouse button and release the button on the form.

- **Drag and drop** a cassette object to the form



Select the desired **chamber** using the drop-down list:



By default, the name of the selected chamber is displayed on top of the cassette. You can hide the chamber name by unchecking the following box:



Customize the cassette font and background color:

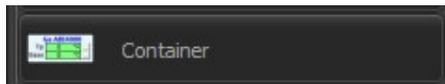


Adjust the platen height value using the spinner:



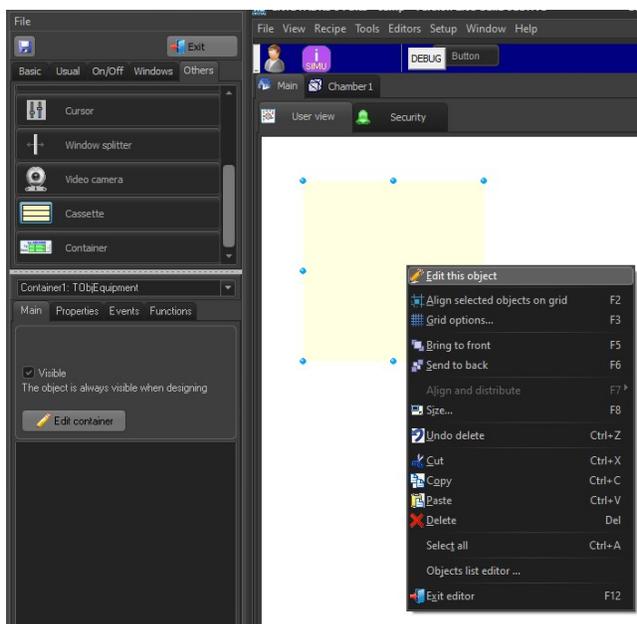
(15 pixels by default)

Container



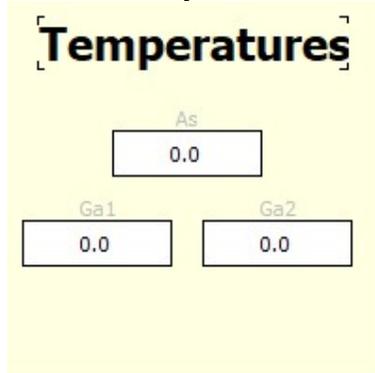
A container is a group of objects.

To add a container on a form, select it on the component palette, hold down the mouse button and release the button on the form.



Right click on the object (see screenshot above) and select “*Edit this object*”

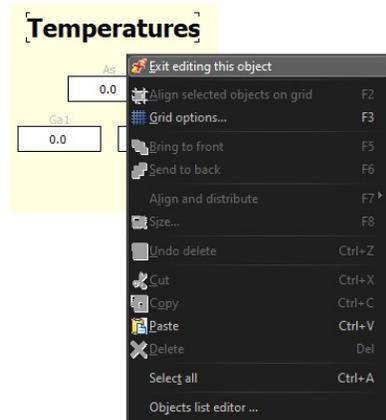
Move new objects into the container



To exit the edition of the container, you can either click on the button “End of editing” located on top of the Tool palette window (see screenshot bellow)



or right click on the container and select “Exit editing this object” (see screenshot bellow)



14. PLATEN AUTOMATION

The *Automation* tab allows you to create, edit or run production batches. The *Automation* feature is available only with the MBE systems that are equipped with a cluster or a pick and place transportation system (like MBE49, MBE6000, MBE7000 or MBE8000).

In *Crystal XE*, a batch allows you to organize the operations of several platens in the MBE system.

For each platen, you can define its path by selecting:

- The first chamber (generally a no process chamber like a load)
- One or several process chambers and at least one recipe to be executed in.
- The last chamber (generally a no process chamber like an unload)

For each of these steps, you can also define the position of the platen in the cassette.

With a MBE49, MBE6000 and MBE7000 you are not free to move the platen anywhere. You must follow a path.

The default path is:

- Load 1
- Move the platen to the preparation chamber and execute the preparation recipe.
- Move the platen to the growth chamber and execute the growth recipe.
- Move the platen to the load 1 or load 2 if it exists.

With a cluster there is no such constraint.

File location and format

The batch is saved as a **‘.batch’ file** (or **‘.pas’** if it is a script) that can be accessed in your project directory > *Batch*. The batch file format is XML (or text file if it is a script) but we do not recommend to edit directly the XML file.

To browse the batch directory, click on the *File* menu, select *Browse project directory* and go in the batch sub-directory.

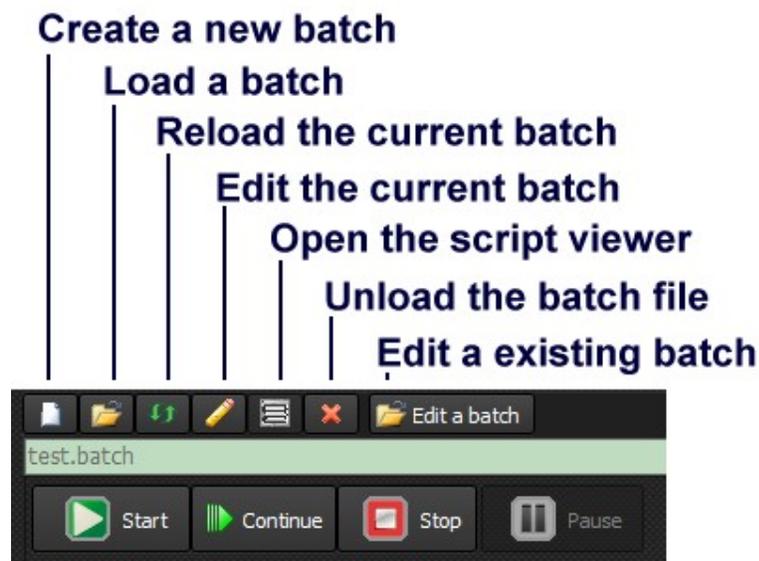
There are two types of batch:

- Using the batch editor (**.batch** file)
- Using the script editor (**.pas** file)

If the transfer machine is a cluster, you can choose between these two options.

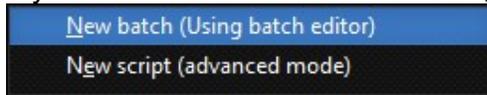
For pick-and-place transfer machines (MBE49, MBE6000, 7000, 8000), you can only choose the batch editor.

Panel control:



Create a new batch

If your transfer machine is a **cluster**, you will have two possible choices:



For **pick and place** transfer machines (MBE49, MBE6000, 7000, 8000), you'll go straight to the batch editor.

 Load a batch file in the batch engine.

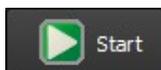
 Reload the current batch file.

 Edit the current batch either with the batch editor, or with the script editor (depending of the batch type).

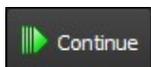
 Open the script debug window to follow the execution of the script.

 Remove the batch from the batch engine.

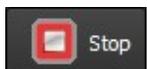
 Edit a batch Edit a other batch file using either the batch editor or the script editor (depending of the batch extension .batch or .pas)



Start the current selected batch.



Continue moving each platen that is stamped as used. The path of each platen can be changed by clicking on it.



After a user confirmation, this will stop the execution of the current batch. The stop will take effect at the end of the current movement.

If a recipe is running, the recipe will continue normally until the end of execution.



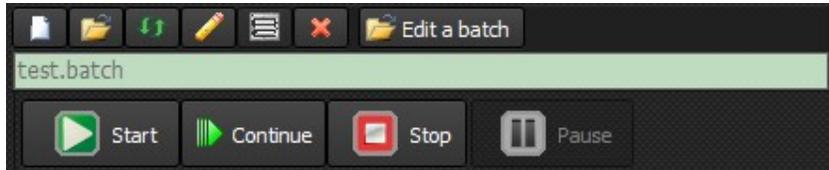
Pause or resume the batch. The Pause will take effect at the end of the current movement.

14.1. Editing batch using the batch editor

Edit an existing batch

Click on the *Edit* button  and select the desired file.

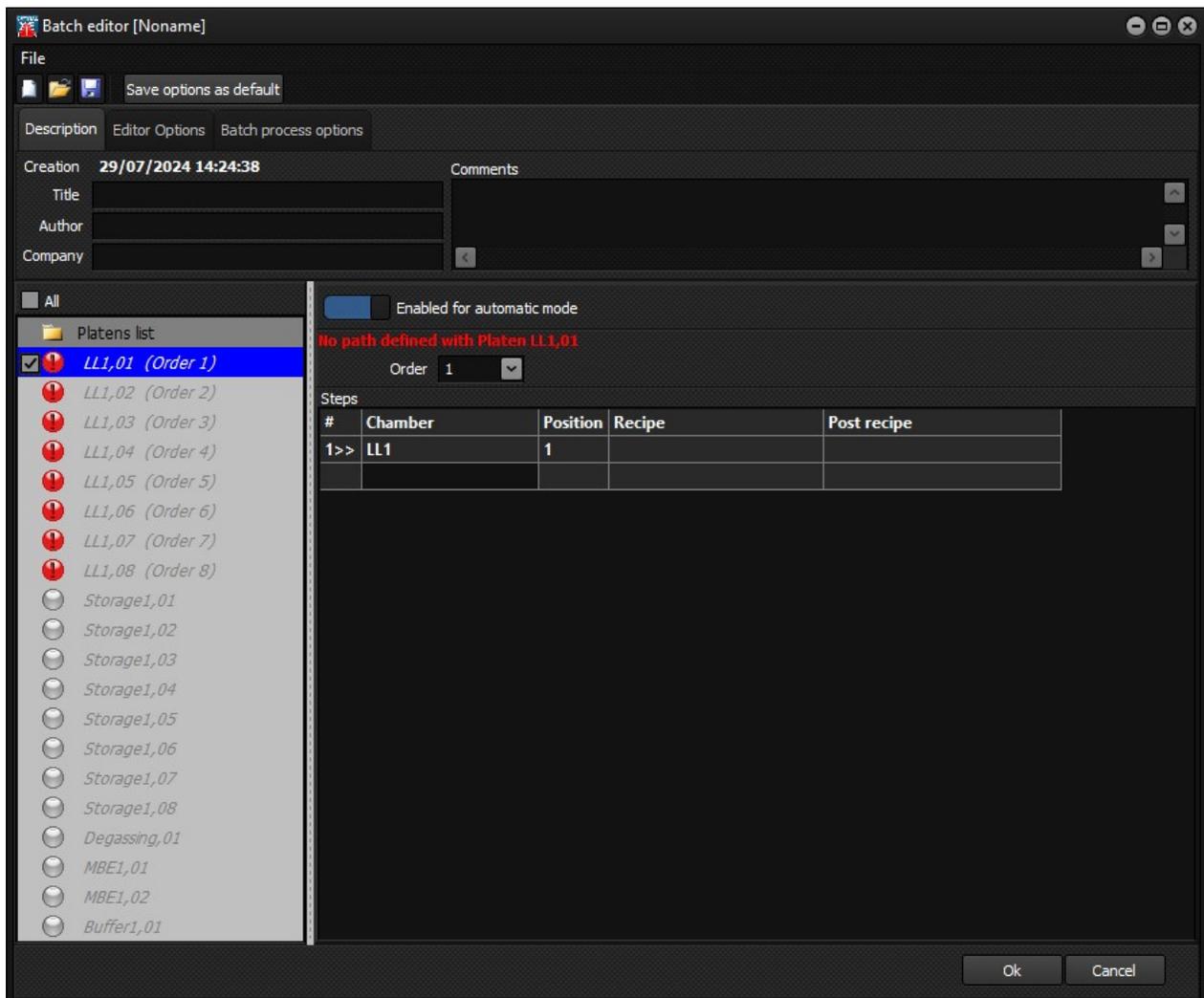
Or you can click on the *File* icon  to load the desired file and then click on the pencil icon  to open it in the Batch editor.



Clicking on the following icon  clears the Batch name *File* input field but does not delete the batch *File*.

Batch editor overview

The platens are displayed in the pane located on the left side of the Batch editor.



Use the checkboxes to select the platen to be used.
You can also use the toggle switch "Enabled for automatic mode".

Platen path

The path of the platen in the MBE system is composed of several configurable steps:

Steps				
#	Chamber	Position	Recipe	Post recipe
1>>	LL1	1		
2	Degassing	1	Degassing\degas.rcp	
3	MBE1	1	MBE1\test.rcp	
4	LL1	1		

For each step, select your choice from the corresponding drop-drop list:

- **Chamber** (required)

The platen will be picked up and dropped off in this chamber. The list shows only chambers connected to the cluster.

- **Position** (required)

The position in the cassette (Only if several platens can be stored in the chamber).

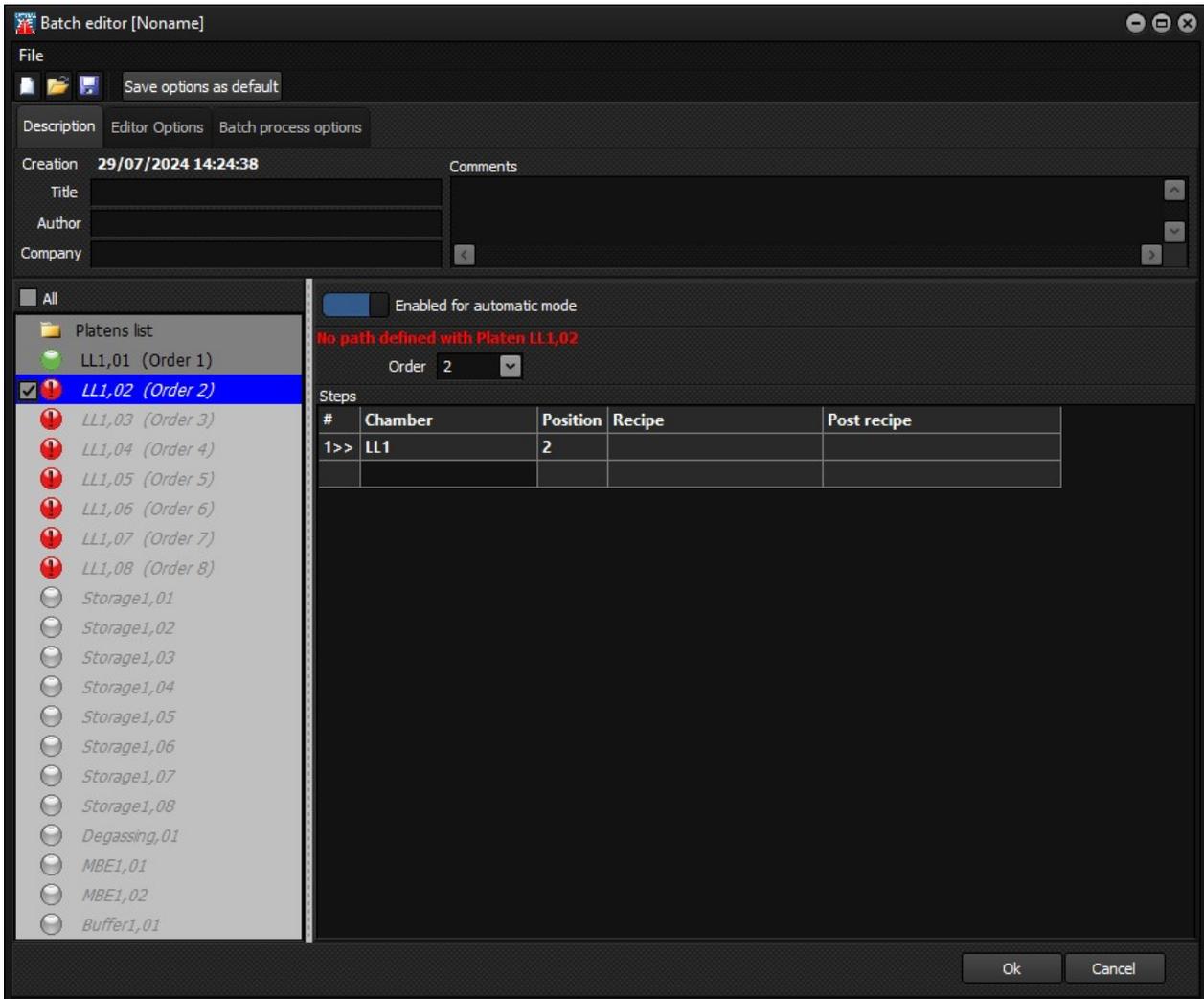
- **Recipe** (optional)

The recipe to be executed in the selected chamber

- **Post recipe** (optional)

A second recipe will start as soon as the first recipe ends.

- If the batch contains an error, a message warns you (in red) and the platen icon turns red as follows:



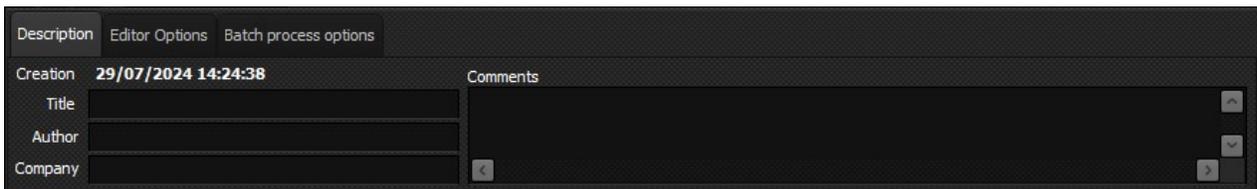
Error examples:

two platens start at the same position in the cassette.
 No path defined with platen xxx.

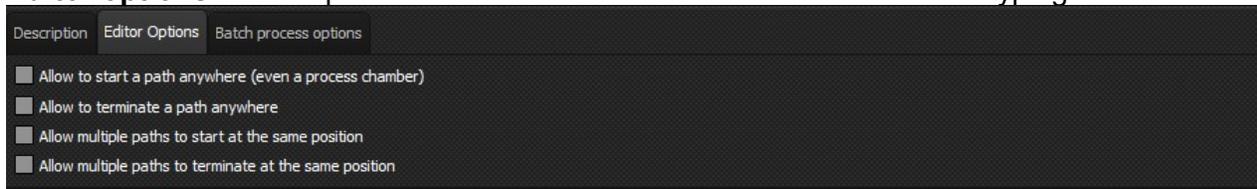
Editor tabs

Description tab

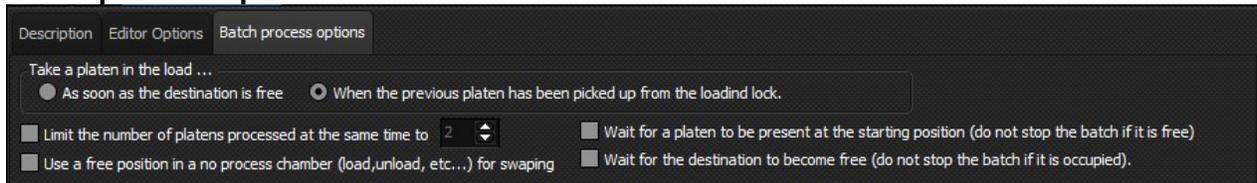
Click on the *Description* tab to edit the general information of the batch such as the title, Author, Company and comments.



Editor options These options restrict the use of the editor in order to limit typing errors.



Batch process options:



By default, when no box is checked, Crystal XE optimizes the paths so that the occupancy rate of the chambers is at the maximum. Thus, several platens can be treated at the same time if the capacity of the system allows it.

“Take a platen in the load...”



Among the platens selected in the batch, Crystal will initiate the start of a platen in the order defined by the “Order” option.

However, two choices are possible:

1) **As soon as the destination is free.**

In this case, the platen will only be removed from the first stage if the destination position is free. Otherwise, Crystal will perform the same check with the next platen in the defined order, and will only take the platen if the destination is free.

Example:

- Platen number 1 in Load at position 1 must go to the degas chamber (order 2)
- Platen number 2 in Load at position 2 must go to the growth chamber (order 1)
- Platen number 3 in Load at position 3 must go to the storage chamber (order 3)

Crystal will begin to process by the order 1 (in this example it is the platen number 2). If the growth chamber is not free, then Crystal will process with the platen corresponding to the order 2 (platen number 1). As the destination of this platen is the degas chamber, and if the degas chamber is free, then the platen number 1 will be moved to the degas chamber even if the platen number 2 stays in the load. The platen number 2 will be moved later when the growth chamber will be free.

The same reasoning will be applied to platen order 3 (platen number 3).

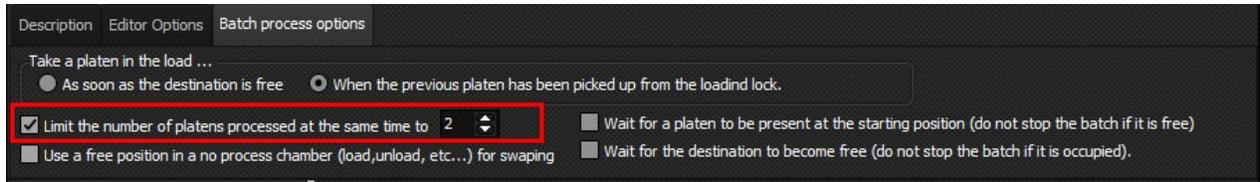
2) **When the previous platen has been picked up from the loading lock.**

In this case, the order will be scrupulously respected.

Using the previous example, Crystal will only move the platen corresponding to order 1 (platen number 2) when the degas chamber is free. Otherwise, no movement will take place. And each platen will only be moved if the one in the previous order has left its starting position.

“Limit the number of platens processed at the same time”

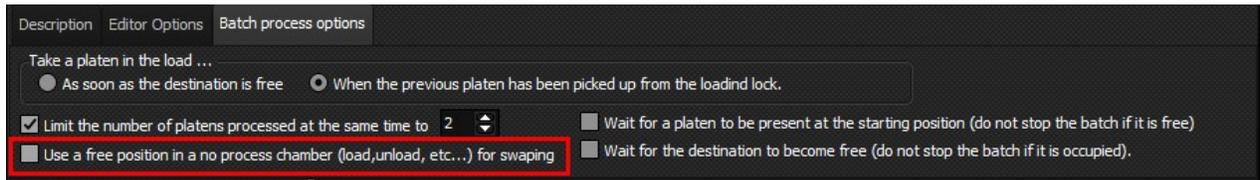
You can define a maximum number of platens to be processed at the same time by checking the following box and selecting the desired number as follows:



“Use a free position in a no process chamber (load, unload, etc..) for swapping”

If two platens need to be dropped to the same chamber, you might need swapping the platens.

Check the following box to allow the cluster to swap platens to an auxiliary storage when needed:

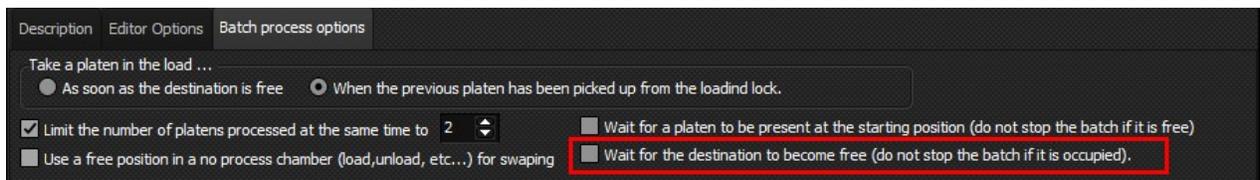


“Wait for a platen to be present at the starting position (do not stop the batch if it is free)”



This option is interesting in the case of continuous production. The starting position **must be different** from the arrival position but can be in the same loading / unloading chamber.

“Wait for the destination to become free (do not stop the batch if it is occupied)”



Like the previous option, this option is interesting in the case of continuous production. The starting position **must be different** from the arrival position but can be in the same loading / unloading chamber. When checked, if the user has not had time to remove the previous stage

which was in the same position, the batch will not stop. In this case, the platen will wait in the previous position but the other tasks will continue in order to optimize the occupancy of the chambers..

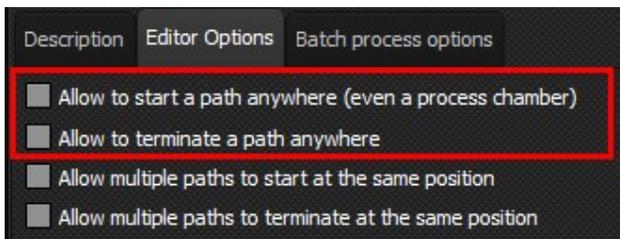
Editor options

These options restrict the use of the editor in order to limit typing errors.

“Allow to start a path anywhere (even a process chamber)”

“Allow to terminate a path anywhere”

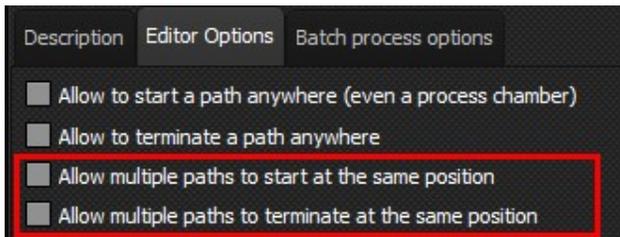
By default, a platen may not start or terminate its path in a growth chamber. Check either one or both the following boxes to allow the platens to start or/and terminate their path in any of the chambers:



“Allow multiple path to start at the same position”

“Allow multiple path to terminate at the same position”

These option disable the error "x platens use the same starting position" and "x platens use the same destination position"



14.2. Editing batch using the script editor

To begin a new script, click on the new button and select script.

The script editor will open:

```

Cluster1
1 Begin
2 // Example
3 // MovePlaten('Loading',5,'Degas',1,true); // ==> Move the platen from Loading at pos 5 to Degas at pos 1 and wait for the end of the move
4 // RunRecipe('Degas','degas.rcp',0,true); // ==> Run recipe "degas.rcp" with option=0 and wait the end of recipe
5 // you can also use: repeat until Degas.Recipe.Status=RS_STOP; // ==> Wait for the recipe end
6 // showMessage('Platen is in the degas chamber and recipe is completed');
7
8 End;

```

You can use all language functions.
To move a tray from a script, use the MovePlaten function.

Refer to the example given in the script editor (in comment) and to the online help when pressing the F1 key (or by the help menu)

Example of script in a system with two clusters:

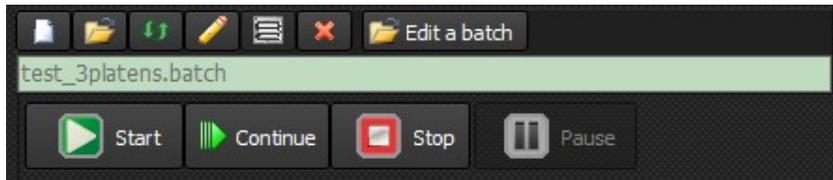
```

C:\riber\Dev_Multicuster\Batch\simulation1.pas
simulation1.pas
1 Uses Stdconst;
2 Var
3   Ok : boolean;
4 //-----
5 Procedure StopBatch(ErrMsg:String);
6 Begin
7   LogEvent(0, 790, 2, EL_Critical,ErrMsg);
8   Cluster1.Stop;
9   Cluster2.Stop;
10  exit; // Stop the script
11 end;
12 //-----
13 Begin
14 // -----
15 // Initialise cluster 2
16 // -----
17 Cluster2.stop;
18 Resettimer(1); // reset timer for time out
19 repeat until (Cluster2.Status=BS_STOP) or (GetTimer(1)>10);
20 if (Cluster2.Status<>BS_STOP) then StopBatch('Batch script in cluster 2 cannot Stop the batch of cluster 2');
21 // Load the idle script in cluster 2
22 Ok:=Cluster2.LoadFromFile('idle.pas');
23 if not(ok) then StopBatch('Batch script in cluster 2 cannot load the idle script');
24 // Start the cluster 2
25 Ok:=Ok and Cluster2.Start;
26 if not(ok) then StopBatch('Batch script in cluster 2 cannot start');
27 //-----
28 // Start moving platens
29 //-----
30 // Move M1 to degassing
31 MovePlaten('LL1',1,'Degassing',1,true); // ==> Move the platen from Loading at position 5 to Degas at position 1
32 RunRecipe('Degassing','degas.rcp',0,true);
33 MovePlaten('Degassing',1,'Buffer1',1,true); // ==> Move the platen from Loading at position 5 to Degas at position 1
34 sleep(20000);
35 //
36 Cluster2.MovePlaten('Buffer2',1,'Oxidation',1,true); // ==> Move the platen from Loading at position 5 to Degas at position 1
37 ShowMessage('The platen must be in the Oxidation chamber');
38 RunRecipe('Oxidation','test_oxid.rcp',0,true);
39 Cluster2.MovePlaten('LL2',1,'storage2',1,true); // ==> Move the platen from Loading at position 5 to Degas at position 1
40 Cluster2.MovePlaten('Oxidation',1,'Buffer2',1,true); // ==> Move the platen from Loading at position 5 to Degas at position 1
41 sleep(2000);
42 MovePlaten('Buffer1',1,'LL1',1,true);
43
44 End;

```

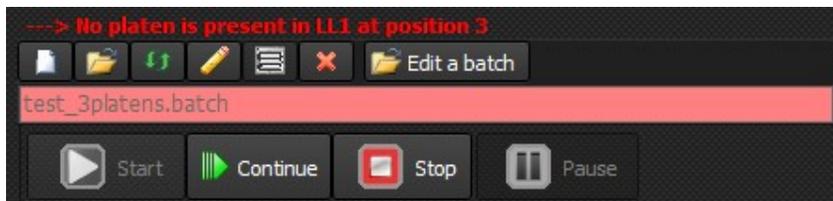
14.3. Executing a batch

If no batch file is loaded, click on the file icon  to select and open the desired batch file. Once opened, the file appears in the field on a green background. In case of error the background is displayed in red and in this case, it is not possible to launch the batch.



A green background indicates that the batch is loaded and has no error.

In case of error the background color is red:

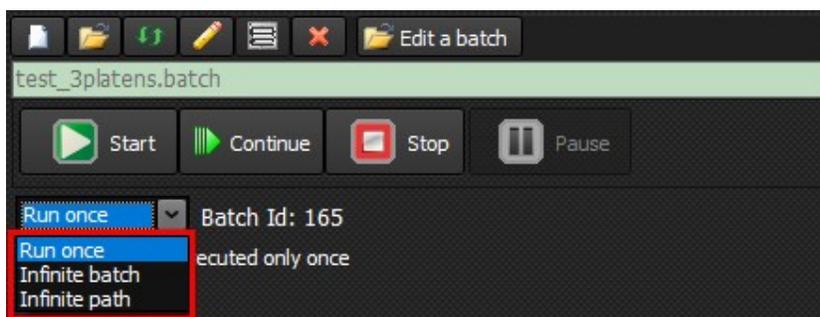


A red background indicates that the batch is loaded and has error.

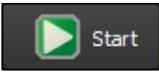
The error message is indicated in red and bold.

Execution mode

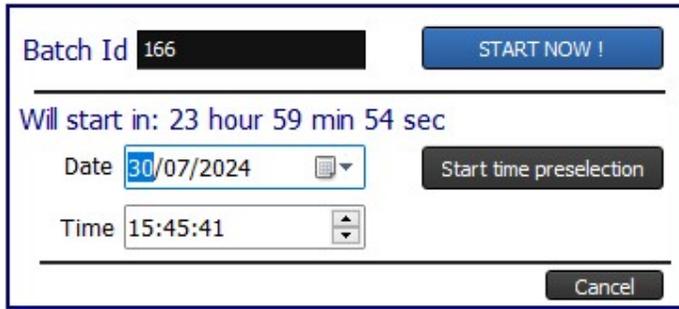
Three execution mode are possible:



- **Run once** will execute the batch only once.
- **Infinite batch**: when all the platens will be completed, the batch will restart automatically.
- **Infinite path**: when a platen is finished, it automatically restart while other platens continue to run.

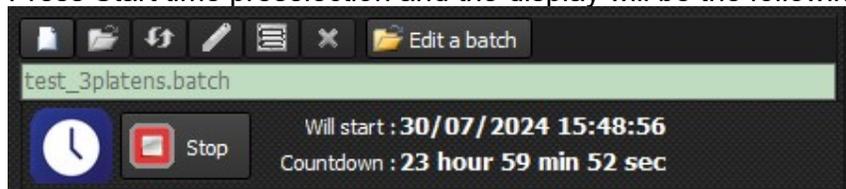
Click on the Start button  to start the batch.

The following window is displayed:



- **“Start now”** will start the batch immediately.
- **“Start time preselection”**: This second option delays the batch launch to a desired date and time.

Press Start time preselection and the display will be the following:



Execution

During the execution of a batch, an animation allows to follow the evolution of the positions of the platens:

Moving history

Step	1	2	3	4	5
Date	24/07/29	24/07/29	24/07/29	24/07/29	24/07/29
Time	15:52:20	15:52:29	15:52:33	15:52:48	15:52:52
Relative time	00:00:00	00:00:09	00:00:13	00:00:28	00:00:32
Duration	---	00:00:09	00:00:03	00:00:15	00:00:03

Platens animation

	Storage1	Degassing	MBE1	Buffer1
[8]	8	8	8	8
[7]	7	7	7	7
[6]	6	6	6	6
Completed [5]	5	5	5	5
...	4	4	4	4
...	3	3	3	3
[2]	2	2	2	2
[1]	TESTEUR	1	1	1

Batch log

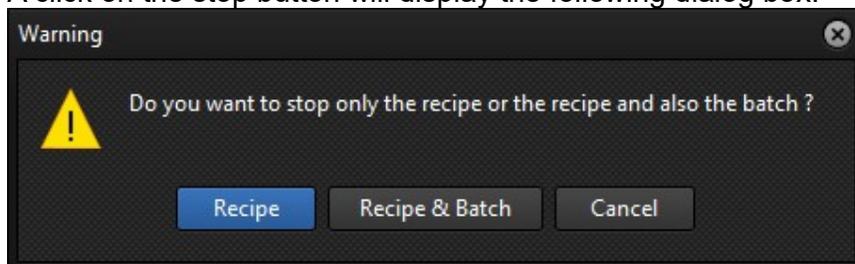
```

2024/07/29 15:52:16,874 | Loading batch 169 "test_3platens.batch" with 3 platens | Config= 11 Chambers=LL1[8]; Storage1[8]; Degassing[1]; MBE1[2]; Buffer1[1];
2024/07/29 15:52:20,013 | The start command is accepted by PLC
2024/07/29 15:52:20,512 | Start of the batch
2024/07/29 15:52:20,436 | Move platen Number 5 from LL1.5 to Degassing, 1 Sending Goto from 2005 to 7001
2024/07/29 15:52:20,830 | Moving platen from LL1.5 to Degassing, 1
2024/07/29 15:52:24,116 | Transfer is done and platen number 5 is now in chamber Degassing at pos 1
2024/07/29 15:52:24,427 | Running recipe "degas.rcp" in chamber Degassing
2024/07/29 15:52:29,963 | The recipe in Degassing is now stopped.
2024/07/29 15:52:29,966 | Move platen Number 5 from Degassing, 1 to LL1.5 Sending Goto from 7001 to 2005
2024/07/29 15:52:30,278 | Moving platen from Degassing, 1 to LL1.5
2024/07/29 15:52:33,454 | Transfer is done and platen number 5 is now in chamber LL1 at pos 5
2024/07/29 15:52:33,659 | Platen number 5 completed.
2024/07/29 15:52:33,664 | Move platen Number 4 from LL1.4 to Degassing, 1 Sending Goto from 2004 to 7001
2024/07/29 15:52:34,093 | Moving platen from LL1.4 to Degassing, 1
2024/07/29 15:52:37,597 | Transfer is done and platen number 4 is now in chamber Degassing at pos 1
2024/07/29 15:52:37,587 | Running recipe "degas.rcp" in chamber Degassing
2024/07/29 15:52:48,820 | The recipe in Degassing is now stopped.
2024/07/29 15:52:48,823 | Move platen Number 4 from Degassing, 1 to MBE1, 1 Sending Goto from 7001 to 1001
2024/07/29 15:52:49,227 | Moving platen from Degassing, 1 to MBE1, 1
2024/07/29 15:52:52,134 | Transfer is done and platen number 4 is now in chamber MBE1 at pos 1
2024/07/29 15:52:52,532 | Running recipe "test.rcp" in chamber MBE1
2024/07/29 15:52:52,641 | Move platen Number 8 from LL1.3 to Degassing, 1 Sending Goto from 2003 to 7001
2024/07/29 15:52:53,306 | Moving platen from LL1.3 to Degassing, 1
2024/07/29 15:52:56,425 | Transfer is done and platen number 3 is now in chamber Degassing at pos 1
2024/07/29 15:52:56,764 | Running recipe "degas.rcp" in chamber Degassing
2024/07/29 15:53:38,058 | The recipe in Degassing is now stopped.
  
```

You can supervise the execution of the recipe using the Recipe inspector window. On a process chamber column, click on desired platen and then click on the flask icon to open the Recipe inspector in a pop-up window.

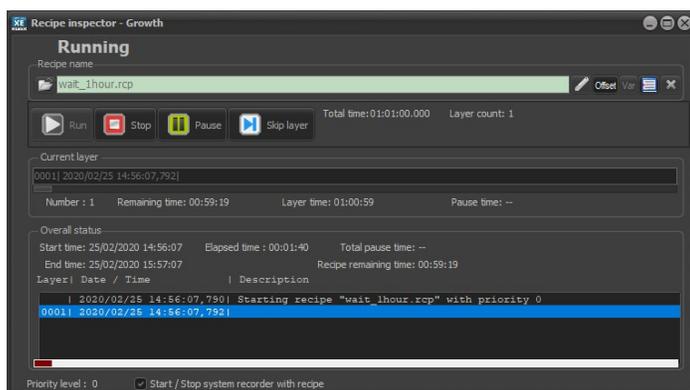
LL1	Storage1	Degassing	MBE1	Buffer1
[8] 8	-- 8	-- 1	-- 2	-- 1
[7] 7	-- 7		Recipe [8] 1	
[6] 6	-- 6			
Completed [5] 5	-- 5			
Completed [4] 4	-- 4			
-- 3	-- 3			
[2] 2	-- 2			
[1] 1 TESTEUR	-- 1			

A click on the stop button will display the following dialog box.



At this point, you can either stop the recipe only or stop the recipe and the batch.

A click on this icon will open the recipe inspector:



The **Batch log** box displays history of the batch events.

Batches related events are saved automatically as '.txt' files to the *Log folder*, in your project directory.

To open your project directory, click on the file menu and select *Browse project directory*.

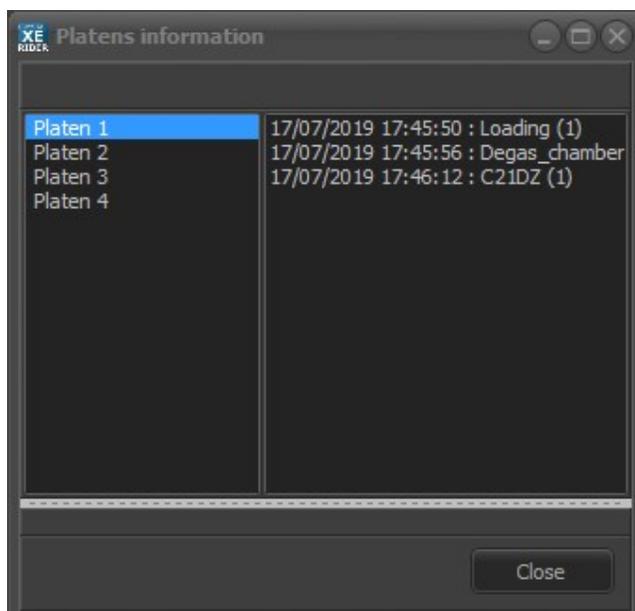
Moving history

Step	1	2	3	4	5
Date	24/07/29	24/07/29	24/07/29	24/07/29	24/07/29
Time	15:59:45	16:00:11	16:00:15	16:00:41	16:00:45
Relative time	00:00:00	00:00:26	00:00:30	00:00:56	00:01:00
Duration	---	00:00:26	00:00:03	00:00:26	00:00:03
LL1	↓ ⑤ 5	↑ ⑤ 5	↓ ④ 4		↓ ⑧ 3
Storage1					
Degassing	↓ ⑤	↑ ⑤	↓ ④	↓ ④	↓ ⑧
MBE1				↓ ④ 1	
Buffer1					
Move time	4 sec	3 sec	3 sec	3 sec	3 sec
Recipe	degas.rcp		degas.rcp	test.rcp	degas.rcp
Platen ID					

This part of the window graphically displays the paths taken by the different platens.

In the tool bar, the column width can be changed.

Click on the Info button  Platens to open the Platens information pop-up window.



For each selected platen, the *Platens information* windows gives you the date and exact time the platen was dropped into a chamber and allow you to control whether the path was correctly executed.

14.4. Modify paths individually at anytime

When the batch is running or stopped, you can edit the path followed by each platen.

To edit a platen, click on it and the following dialog box will open:

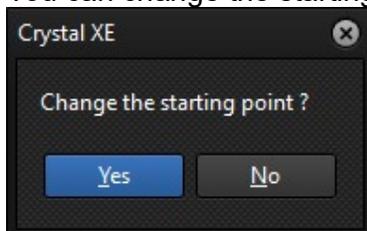
The screenshot shows a table with columns: LL1, Storage1, Degassing, MBE1, Buffer1. The rows represent different platens, with some cells containing numbers and dashes. A dialog box titled "Platen at LL1,5" is open, showing a table of steps for that platen.

#	Chamber	Position	Recipe	Post recipe
1	LL1	5		
2	Degassing	1	Degassing\degas.rcp	
3>>	LL1	5		

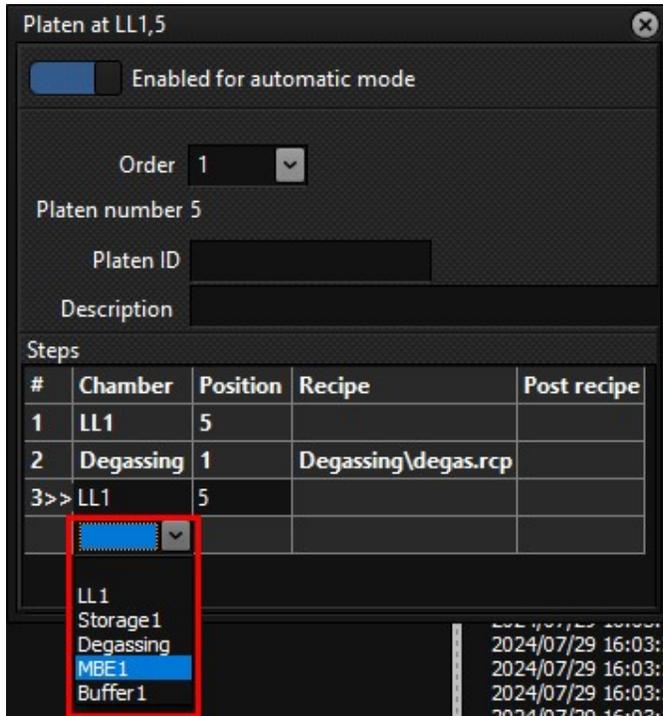
You can modify the Platen ID, the description, the path and also the starting point.

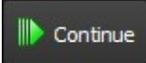
The starting point is displayed with these two characters ">>".

You can change the starting point by double clicking on a cell of the first column "#"



It is also possible to add a new step to the path by click on the next free cell in the Chamber column :



When you have finished modifying all the platens, press the “continue” button  to continue the batch from the current position.

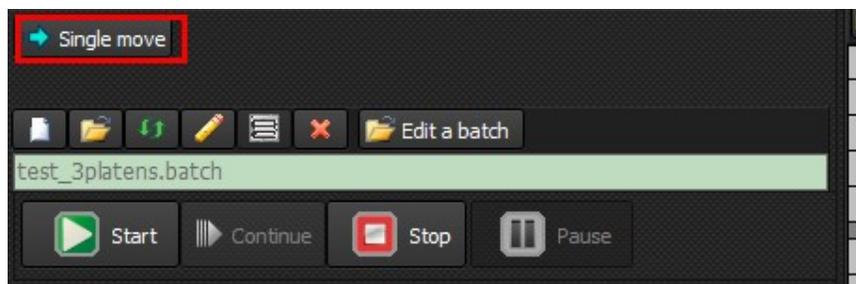
14.5. Moving a platen for a simple movement

There are two ways to move a tray from one position to another.

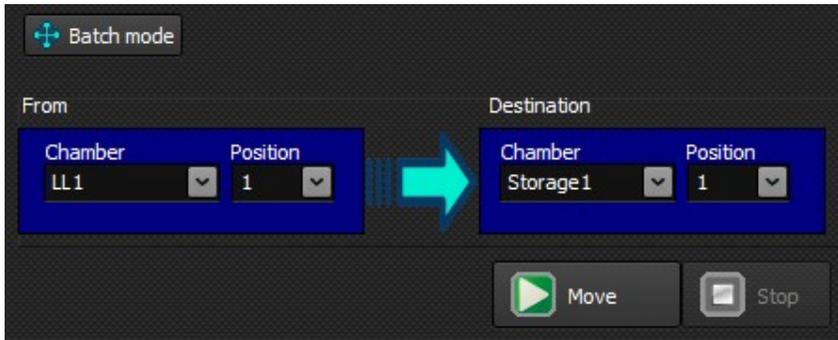
By clicking on the button “Single move” or by using “drag and drop”.

1) Single move

When the batch is stopped, click on the button “Single move”:



This will display the following information:



Select the start chamber/position and the destination chamber/position and press on the Move button to start the movement.

2) Drag and drop

You can drag and drop a platen from the automation tab view or from a user view like the main user view.

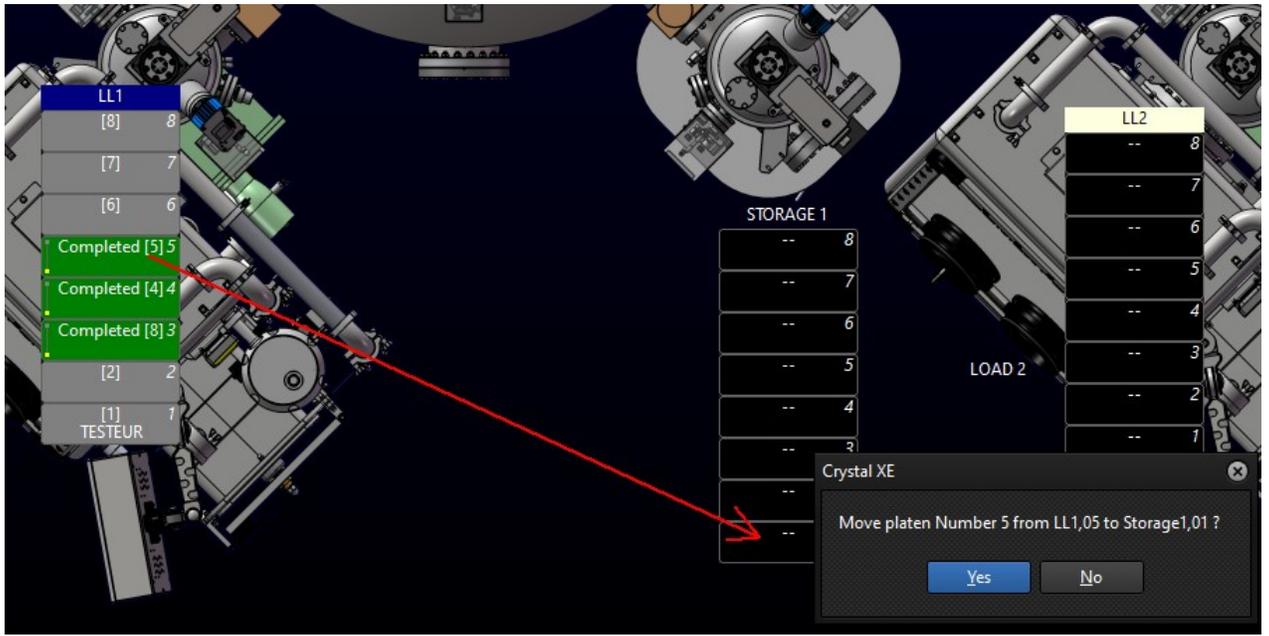
Example from the automation tab view:

LL1	Storage1	Degassing	MBE1	Buffer1
[8] 8	-- 8	-- 1	-- 2	-- 1
[7] 7	-- 7		-- 1	
[6] 6	-- 6			
Completed [5] 5	-- 5			
Completed [4] 4				
Completed [8] 3				
[2] 2				
[1] 1 TESTEUR				

Crystal XE

Move platen Number 5 from LL1,05 to Storage1,01 ?

Example from the main user view



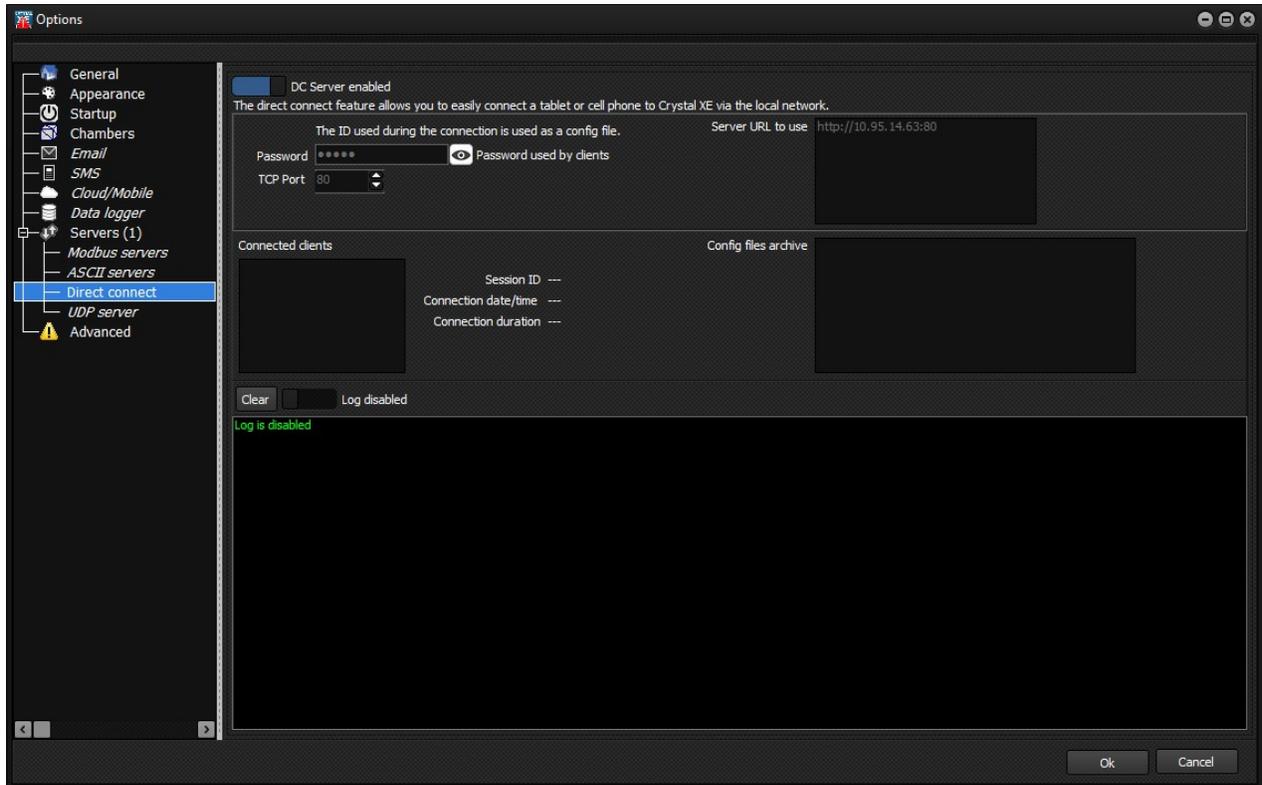
15. Direct connect

15.1. Introduction

The direct connect is an option in Crystal XE that can be enabled in the setup/options window. Crystal XE acts as a server (it responds to requests sent by other software.).

The other software requesting the data can be a specially developed program or the Riber soft mobile client application (available for download for Android and Windows from <http://www.crystalXE.com>).

To enable the direct connect feature, go in setup/options and enable the Servers / Direct connect:



Note the **server URL** and the **password** displayed on this page.
This information must be used by the application connecting to Crystal XE.

To operate, the application must be connected to the **same network as Crystal XE**. As the connection is direct, there's no need for an Internet connection, as information passes directly between the client and the server.

So if you're using a tablet or phone, they need to be connected to Crystal XE using a **Wifi link**.

15.2. Example with the RIBER soft client application

The application is available for Android and Windows.

- The Android version is available on the Google Play Store by searching for “crystalxe”



CrystalXE mobile - RIBER
Riber

- The windows application is available for download at <https://www.crystalxe.com/mycrystalxe/download/>

Note: You need to log in to your account to download the application - if you don't have an account, you can create one for free - then go to My CrystalXE and the “Download” page.



Mobile client application for windows

This application is only usefull if you enable the cloud in the main Crystal XE application.
(June 06, 2021)



DOWNLOAD

2019-02-22_MobileXE_64_bit_2.01b03.exe

Run the application (on android or windows) and enter the following information:

Mobile CrystalXE

Mobile CrystalXE v2.10b01

RIBER SOFT

Account

If you don't have an account, go at www.crystalxe.com

Email: John_PC

Password: ●●●●

Server: Custom

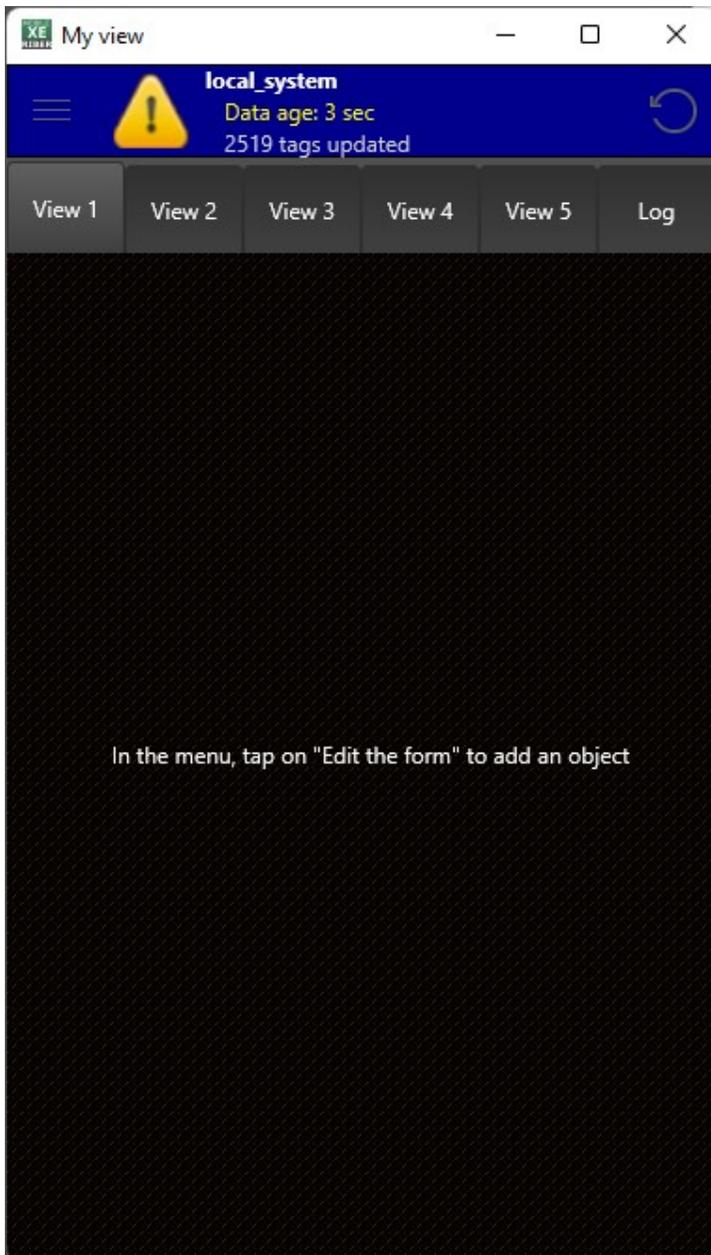
http://10.95.14.63:80

Login

In that case,

- The **email** field is an identifier (string of characters without “@”, not an email address) that will be displayed in Crystal XE to identify the client.
- The **Password** must be the same as the one defined in the options in Crystal XE.
- The **server** must be “Custom”
- And you must specify the **Crystal address** (this is the ip address of the Crystal XE computer with the indication of the port 80)

Click on Login to establish the connexion.

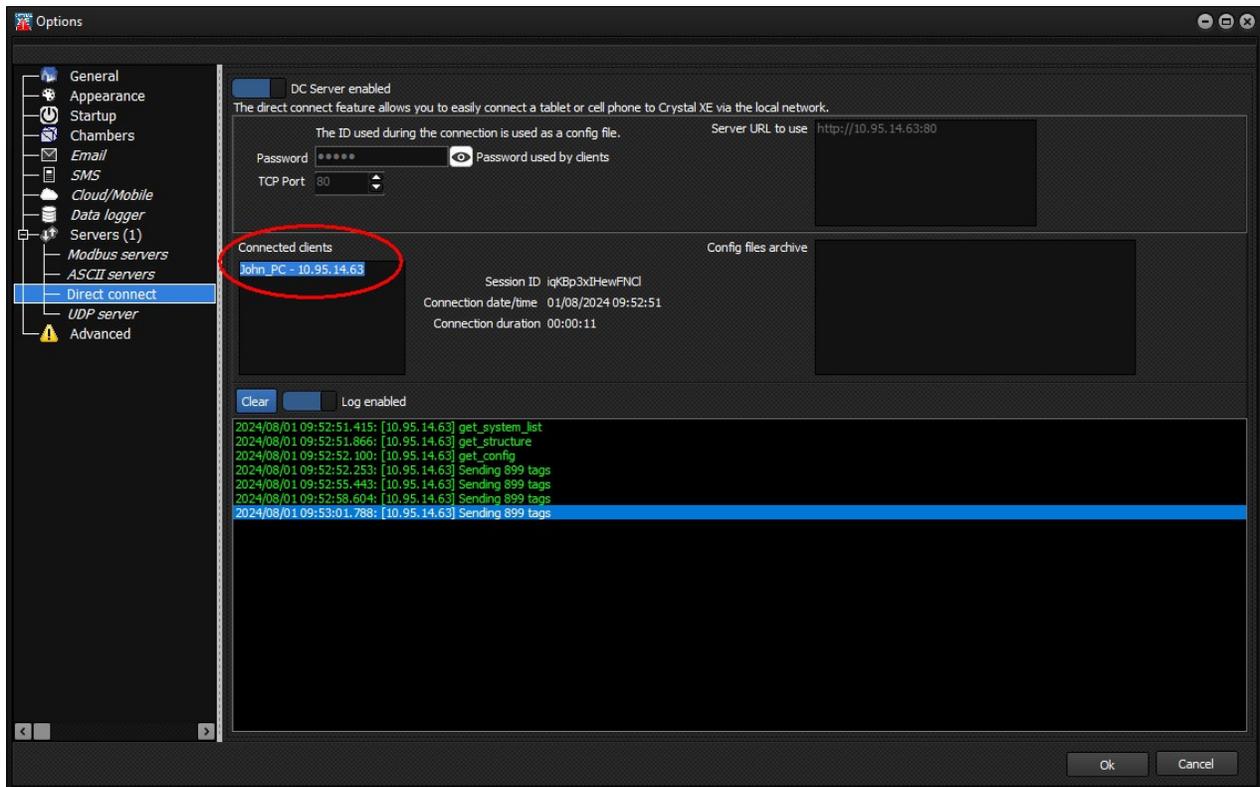


You are now ready to create your own screens directly on the application.

Screens are stored on the remote Crystal XE computer.

So when you reconnect with the same email, you'll get all the screens back. This means you can use different hardware (phone, tablet, other computer..) to display the same screens.

In the options of CrystalXE, you can see the client indicated in the connected client list:



15.3. Protocol

If you want to develop your own application, then you need to follow these guidelines. In these examples, <CR> is the carriage return character (ASCII code 13 in decimal) and <LF> is the line feed character (ASCII code 10 in decimal)

The application must follow the following steps in the order indicated (except for optional steps)
 The client will be disconnected after 10 seconds of inactivity.
 In this case, the entire sequence must be repeated.

1) Login transaction (example with password = riber)

The client must send the following string of characters:

```
action=login<CR><LF>email=John<CR><LF>password=F4E4EBF7E4E5EDE9E9E5F5E
EEEF6E0E5EBEEE7F6E3EBF4E6EA948F848394E5ECEBEEECEDF7EFE5EAE4E1E2E2E7E9F
5E9E4E3<CR><LF>
```

Example of answer:

```
OK:Array
(
  [id] => NAbCsFcbQmskS3r
  [name] => CrystalXE
  [company] => Riber
  [email] => John
  [password] => ?
```

```
[datetime] => 2024-07-31 17:17:22
[dernieraces] => 2024-07-31 17:17:22
[cnt] => 263
[addr_ip] => xxx.xxx.xxx.xxx
[NL] => 1
[AU] => 1
[level] => 1
)
```

(Each line terminates by <CR><LF>)

2) Request system list

```
action=get_system_list<CR><LF>
```

Example of answer:

```
OK:id=1,systemname=local_system,versionXE=3.20 build
5,datetime=2024-08-01 08:48:00,lastaccess=2024-08-01
08:48:00#&#<LF>
```

3) Request structure

```
action=get_structure<CR><LF>id_system=1<CR><LF>
```

Example of answer:

```
OK:id_tag=0,kind=0,vartype=8,rw=3,handle=0,level1=Load1,level2=P
ressure,level3=Load1,level4=MV,comment=Rescaled Measured
Value,value=0#&#id_tag=1,kind=2,vartype=6,rw=3,handle=0,level1=L
oad1,level2=Pressure,level3=Load1,level4=AllAgentDisabled,commen
t=,value=1#&#id_tag=2,kind=2,vartype=6,rw=3,handle=0,level1=Load
1,level2=Pressure,level3=Load1,level4=RegConnected,comment=,valu
e=1#&#id_tag=3,kind=2,vartype=6,rw=3,handle=0,level1=Load1,level
2=Pressure,level3=Load1,level4=SubChannelNbr,comment=,value=1#&#
id_tag=4,kind=2, (...) <LF>
```

4) Optional: Request configuration (screens)

```
action=get_config<CR><LF>id_system=1<CR><LF>
```

Answer: a zip file that contain the screens (format not specified, only for Crystal XE clients)

5) Optional: Request configuration file (config.xfg file)

```
action=get_configXFG<CR><LF>
```

Answer: the config.xfg file

6) Optional: Request the history log

```
action=get_history_log<CR><LF>id_system=1<CR><LF>
```

Answer:

```
OK:id=41,message=$D939|2:2024/08/01 09:22:22: (541) This version
is up to date (the latest released version is
3.20build5)##id=40,message=$2D28|1:2024/08/01 09:22:21: (002)
[splash] 11.438: Ready##id=39,message=$5F9D|1:2024/08/01
09:22:21: (002) [splash] 11.422: Checking
keys##id=38,message=$FDC3|1:2024/08/01 09:22:21: (002) [splash]
11.422: Initialization is
complete##id=37,message=$CE8C|1:2024/08/01 09:22:21: (002)
[splash] 11.406: Starting
services##id=36,message=$E7ED|1:2024/08/01 09:22:21: (002) (...)
```

7) Request tags values

```
action=get_values<cr><lf>id_system=1<cr><lf>
```

Answer:

```
OK:lastaccess=2024-08-01
09:28:00##id_tag=0,value=0,datetime=2024-08-01
09:28:00##id_tag=1,value=1,datetime=2024-08-01
09:28:00##id_tag=2,value=1,datetime=2024-08-01
09:28:00##id_tag=3,value=1,datetime=2024-08-01
09:28:00##id_tag=4,value=0,datetime=2024-08-01
09:28:00##id_tag=5,value=1,datetime=2024-08-01
09:28:00##id_tag=6,value=1,datetime=2024-08-01 (...)
```

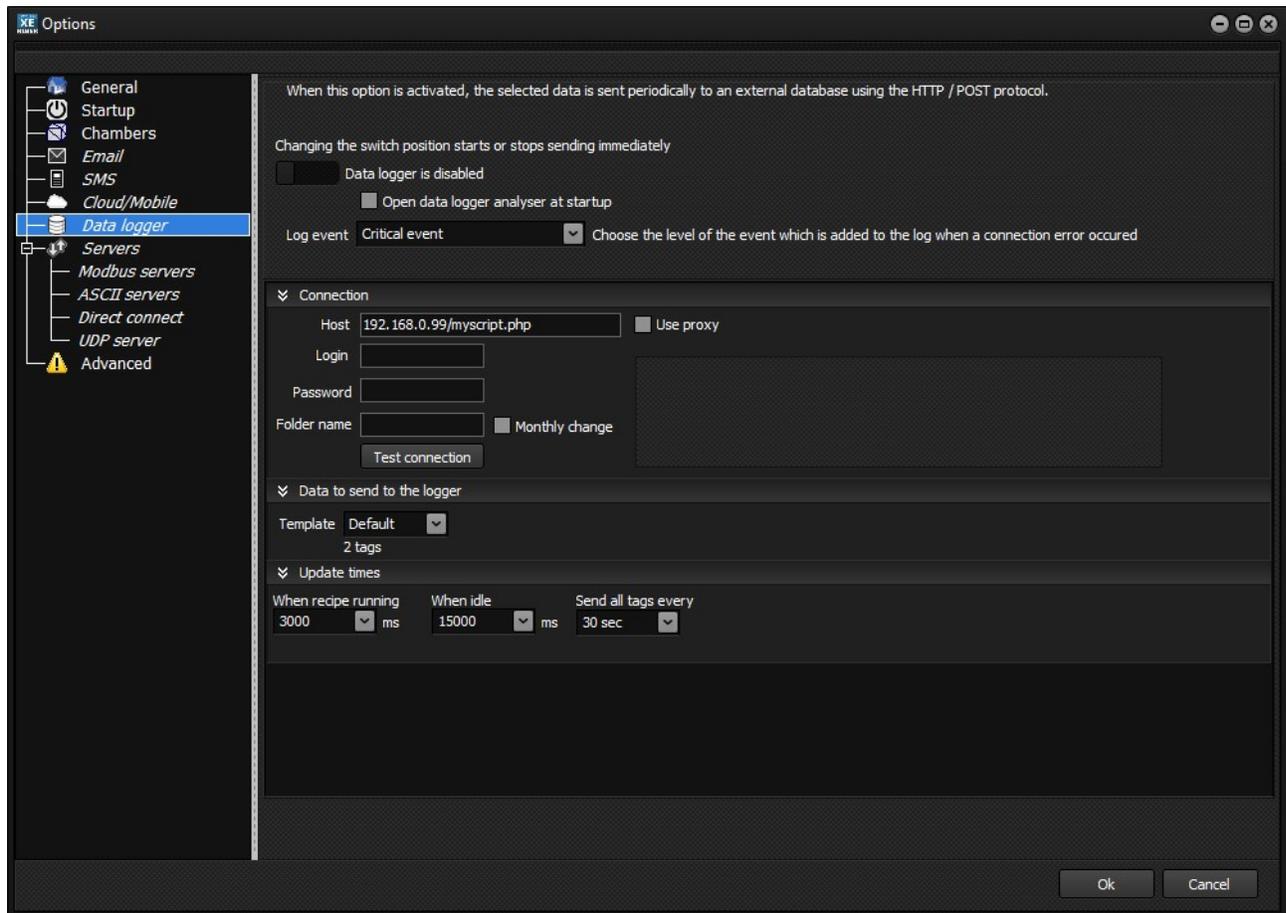
16. DATA LOGGER

16.1. Introduction

Crystal XE can send most of data to an external data logger. The data logger is an option in Crystal XE that can be enabled or not. This feature runs in the background and is invisible to the user.

16.2. Parameters

This feature can be defined in the options window and the item “Data logger”.



The main point here is the update period.

There are three update periods:

- A fast acquisition period which is activated as soon as a recipe is running.
- A slower acquisition period which is used when the system is in standby
- A period to define when all labels are to be sent.

Optimization:

In the first two cases, after each period (When recipe running or When Idle), a data packet is sent to the data logger.

In order to limit the size of the data in the data logger, **only the data that has changed is sent.**

When the third period expires, all data is sent to the data logger even if it has not changed.

16.3. Data sent to the data logger

Example of data sent to the data logger:

- For each chambers
 - o The properties of the equipment and the properties of the sub-equipment which are defined in the default recorder
 - o The status of the recipe
 - o The current layer number.

For example, for a temperature controller that is used for an effusion cell, the following data is sent (12 data): MV, Offset, TSP, regTSP, CSP, MaxTSP, MaxOP, TOP, OP, Tunemode, TampSP, RampOP.

- Properties of the Batch
- Properties of App (CPU_Usage, ErrStatus ..)
- All global variables

16.4. Protocol

Crystal XE uses the protocol HTTP over TCP/IP.

It uses the POST method.

The protocol is the same as that specified for sending data to the cloud.

16.5. Data flow structure

The chronological stages of the exchanges between crystal XE and the server are described below:

1) HTTP_CONNECT

This is the first step of the process to login to the data base. This step will create a session in the webserver.

Parameters list are:

- action=login
- email =<user identifier>
- password=<user password>
- systemname=<user define system name>

2) HTTP_CHECK_LICENSE

After sending the connection information, the server will respond with "OK:" followed by a long character string containing useful data.

To avoid a conflict, this step will check if another computer is using the same system name with the same user name. If so, Crystal XE will ask the user if they wish to continue.

Example of string returned by the server (#\$A is a line feed character)

```
'OK:a_mycrystalxe=Array'$A'('$A' [id] => 1'$A' [id_user] => 94'$A' [systemname] =>
test'$A' [versionXE] => 2.03 build 9'$A' [license] => 048E30572345511D'$A'
[datetime] => 2020-01-02 10:50:49'$A' [client_cnt] => 0'$A' [server_cnt] => 24'$A'
[client_ip] => '$A' [server_ip] => 192.168.1.184'$A' [lastaccess] => 2020-01-06
09:29:15'$A' [lastaccessclient] => 0000-00-00 00:00:00'$A' [client_send] => '$A' [AB]
=> 0'$A')A
```

3) HTTP_UPDATE_LICENSE

Crystal XE sends its license number to the server. This information will be saved in the data base.

Parameters of the post are:

- action=update_license
- license=<current registered license number>

4) HTTP_SEND_TAGS_DEFINITION

Send the tags name to the server.

Parameters of the post are:

- action=tags_def
- tag_names=<frame>

Example of frame:

0=Chamber1.BakeOut1.temperature.MV;1=Chamber1.BakeOut1.temperature.Offset;

5) HTTP_REFRESH_DATA

When all the previous steps are successfully completed, the process will remain at that step until no communication errors occur.

Refresh tags values:

Every n seconds (n=3 seconds by default), Crystal XE will send only the data that has changed to the server.

Parameters of the post are:

- action=Mupdate_values
- frame=<frame>

Example of frame:

*0=750.233337402344`8=5`20=750.166625976563`28=7`57=0`77=0.133333340287209`
85=7`(...)*

Send the history log:

When new data is recorded in the log, it is sent to the data logger.

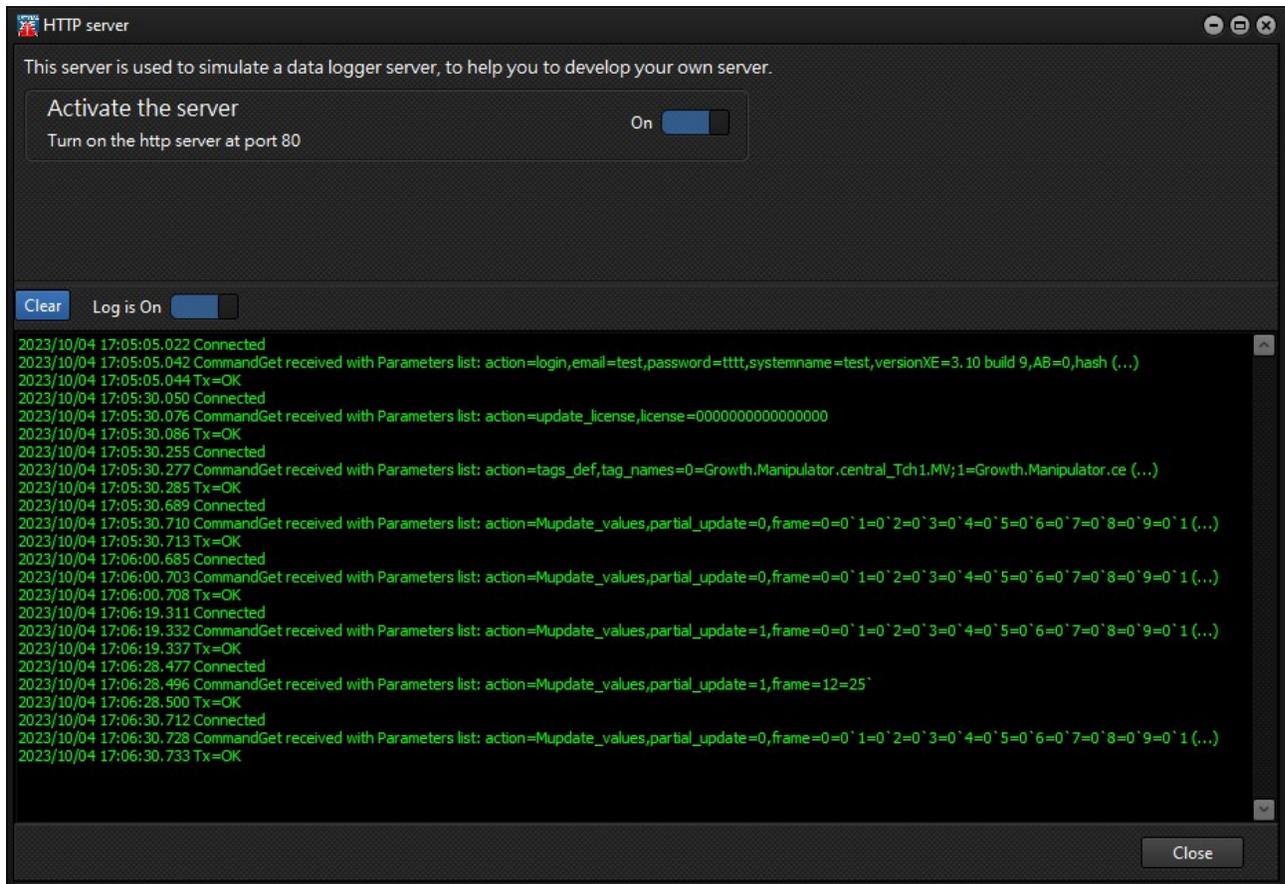
Parameters of the post are:

- action= send_historylog
- logs=<log content separated by '#&#>

16.6. Example of data exchanged with a server

In the following example, we use the http server tool integrated into Crystal XE in the "Debug/Test tools/Http server" menu.

The tool's behavior is straightforward, as it systematically responds "OK" to every http request.



- In the first step, we can see the action=login with other parameters like email, password,systemname... etc
- In the second step, the Crystal XE license key is sent to the data logger to identify the single computer on which Crystal XE is licensed.
- In the third step, the tags names are sent following this format:
tag_names=<tagid_0>=<tagname_0>;<tagid_1>=<tagname_1>; etc...

In the following steps, Crystal XE sends either complete (partial_update=0) or partial data updates (partial_update=1).

17. DYNAMIC DATA EXCHANGE (DDE)

17.1. Introduction

Dynamic Data Exchange (DDE) is a technology for interprocess communication used since the first versions of Microsoft Windows.

Thanks to the DDE protocol, it is possible to access all the data (properties) of Crystal XE but Crystal XE can also send data or execute actions to other compatible applications.

Four script functions allow these exchanges:

- App.DDE.Check
- App.DDE.Poke
- App.DDE.Send
- App.DDE.Request

For more details on these functions, please refer to the DDE section in the script help or in the Crystal XE user manual.

This data exchange protocol can be used for report editing or for real-time calculations in Excel and interaction with Crystal XE.

MS Excel ©, MS Word © and LibreOffice are fully compatible with DDE.
Other software like Matlab that could be useful are also compatible with this protocol.

Using DDE over the network

Netdde.exe is a Microsoft Windows Network Dynamic Data Exchange service. It is used to help exchange data over the network.

Remark:

If DDE does not work with MSWord, add the following key in the register:
In HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Word\Security (check this path depending of the version you have), add a DWORD 32 bit and name it AllowDDE. Then change the value to 2.

Load Excel or Word or other application from a script

From a script in Crystal, to open an application in Windows, use either the function **WinExec** or **ShellExecute**.

Go to the script help from the script editor for more details on these functions.

Example of using WinExec:

```
WinExec('C:\Program Files\Microsoft Office\root\Office16\excel.exe  
c:\temp\test.xlsx',1)
```

17.2. DDE Exchanges with EXCEL

About the decimal separator

By default, Excel uses the decimal separator **defined in Windows**: Control panel / Regional settings / Button Additional settings / Decimal symbol.

But you can also change the default decimal separator **in Excel**: Menu File / tab Options / tab Advanced / uncheck the box “Use system separator” and enter the new Decimal separator.

17.2.1. Display live data from Crystal XE: Hot DDE Link

Syntax: “=**crystalxe|property!**<property name>”
(The ASCII code of the char | is 124)

Example to display the error status of Crystal XE

In a cell enter the following text: =**crystalxe|property!App.ErrStatus**
This will display the main error status flag of Crystal XE (0 or 1)

Other example to display the temperature value of a cell

=**crystalxe|property!chamber1.EffusionCell1.tip_temperaure.MV**

The cell will be automatically updated.

To browse all property of Crystal XE, in the main menu click on View/Data explorer

Remark about the decimal séparator:

Crystal will send values with the decimal separator that is defined in Windows (Control panel / Regional settings / Button Additional settings / Decimal symbol)

17.2.2. From Crystal XE request data to Excel (Using script / DDE_Request)

In Crystal XE, uses the script function **DDE_Request**

The syntax is:

```
DDE_request('excel', [bookFileName_with_extension]SheetName, <CellReference>);
```

The cell reference depends on the country language used.

With an English Excel the cell reference is **RxCy** (x is the row number and y is the column number)

With a French excel, the cell reference is **LxCy**

Example:

```
Var  
  S: String;  
Begin  
  S := App.DDE.request('excel', '[Book1.xls]Sheet1', 'R1C1');  
  if length(S)>0 then ShowMessage(S);  
End;
```

17.2.3. From Crystal XE send data to Excel (Using script / DDE_Poke)

The syntax depends on the country language used in Excel.

In Crystal XE, uses the script function **App.DDE.Poke**.

The syntax is:

```
App.DDE.poke('excel', [bookFileName_with_extension]SheetName, <CellReference>, <data>);
```

The cell reference depends on the language used.

With an English Excel the cell reference is **RxCy** (x is the row number and y is the column number)

With a French excel, the cell reference is **LxCy**

Example:

```
Var  
  S: String;  
  n : integer;  
Begin  
  n := random(1000);  
  S := App.DDE.poke('excel', '[Book1.xls]Sheet1', 'R1C1', IntToStr(n));  
  if length(S)>0 then ShowMessage(S);  
End;
```

17.2.4. From Crystal XE execute a method in Excel

Unlike Word, the following does not work with Excel

- Print preview...
- Save document as...

But the following works:

-To create a new sheet

```
App.DDE_Execute('excel','system','[NEW(1)]',true);
```

-To close the current sheet

```
App.DDE.Execute('excel','DDE_VB.xlsm','[File.Close()]',true);  
(or 'system')
```

→ known issue: the last opened file is closed (not the one selected)

-To run a macro

```
App.DDE.Execute('excel','DDE_VB.xlsm','[Run("Sheet1.MyMacro1"]',true);
```

Or

```
App.DDE.Execute('excel','DDE_VB.xlsm','[Run("ThisWorkbook.MyMacro1")]',true);
```

Or

```
App.DDE.Execute('winword','c:\temp\ddetes.docx','[Call MyMacro]',true);
```

Remark: it is not possible to pass arguments to the macro. You must use a cell.

-Display an alert message

```
App.DDE.Execute('excel','DDE_VB.xlsm','[ALERT("Hello world")]',true);
```

More methods

CELL SELECTION

```
-[Select("L1C1:L7C5")]
```

FONT CONTROL

```
-[Format.Font("Arial",12,False,False,False,False,0)]
```

(Parameters are: name of font (such as Arial, Courier, Times), height in points (72 points to an inch),bold, italic, underline, strike, color)

```
-[error(false)]
```

```
-[Workbook.activate("Header Sheet")]
```

PAGE SETUP

landscape page setup with fit to page and center horizontally options specified

```
-[Page.Setup(" ", " ", 0.75, 0.75, 1, 1, False, False, True, False, 2, 1, True, 1, 1, False)'];
```

(Parameters are: head, foot, left-, right-, top-, bottom- margins, heading, grid, horizontal center, vertical center, orientation, paper size, scale, page number, page order, cell color)

COLUMN WIDTHS

specify widths of the row label

```
[select("c1")]
```

```
[column.width(%trim(&rowwid))]
```

OTHER MACRO

```
-[PRINT(1,,,,,1,,,,,,,,,2)]
```

```
-[file.close(false)]
```

```
-[QUIT()]'
```

17.2.5. From Excel write data to Crystal XE (Using macro / DDEPoke)

This is useful if you want to send the content of a cell to a property in Crystal XE.
The property can be a single property or a list of strings like for Listbox, ComboBox or Memo

Example 1 with EXCEL change a single property

This example changes the temperature setpoint of the manipulator.
The temperature setpoint value is supposed to be at the cell at row 2 and column 1

```
Sub Poke()  
    DDEChannel = DDEInitiate("crystalxe", "property")  
    Set MyRange = Cells(2, 1)  
    DDEPoke DDEChannel, "C21DZ_Manipulator.Temperature.TSP", MyRange  
    DDETerminate DDEChannel  
End Sub
```

Example 2 with EXCEL change a list of strings

It is also possible to send a range of data into a listbox:

```
Sub Poke()  
    DDEChannel = DDEInitiate("crystalxe", "property")  
    Set MyRange = Range("A1:A3")  
    DDEPoke DDEChannel, "App.forms.MainUserView.Listbox1.Items", MyRange  
    DDETerminate DDEChannel  
End Sub
```

*For a comboBox, use also the property **items** of the comboBox.
For a memo, use the property **lines**.*

17.2.6. From Excel read data in Crystal XE (Using macro / DDERequest)

Example1: Request the temperature of the manipulator

```
Sub ReadTemperature()  
    DDEChannel = DDEInitiate("crystalxe", "property")  
    ReturnValue = DDERequest(DDEChannel,  
"C21DZ.Manipulator.Temperature.MV")  
    Cells(1, 1).Formula = ReturnValue  
    DDETerminate DDEChannel  
End Sub
```

Example2: Request a list of strings

(Each string of the list is separated by CR and LF)

In this example, we request all the items in Listbox1 which is in the main user view.

```
Sub ListBox()  
    DDEChannel = DDEInitiate("crystalxe", "property")  
    ReturnList = DDERequest(DDEChannel,  
"App.forms.MainUserView.Listbox1.items")
```

```

Range("A1:A3") = ReturnList
DDETerminate DDEChannel
End Sub

```

17.3. DDE Exchanges with Word

To open Word and load a document, use WinExec

Example

```

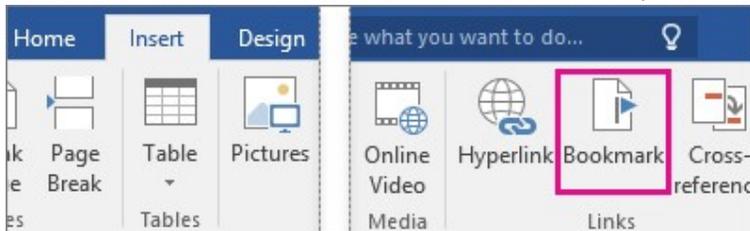
WinExec('C:\Program Files\Microsoft Office\root\Office16\winword.exe
c:\temp\test.docx',1)

```

17.3.1. From Crystal XE send data to Word

In MSWORD, add a bookmark following this procedure:

- Enter a text
- Select this text and in the toolbar menu select Insert / bookmark



- Then give a name for this bookmark.
Example in video: <https://www.youtube.com/watch?v=C3p2FS0cK18>

In Crystal XE, uses the script function **App.DDE.Poke**.

The syntax is:

```
DDE_poke('winword', <File name with extension>, <Bookmark name>, <data>);
```

For example (tested with MSword 365 version 2201), insert a button to the main user view and enter the following code:

```

Var
  S: String;
  n : integer;
Begin
  n := random(1000);
  S := App.DDE.poke('winword', 'test.docx', 'MyBookmark', IntToStr(n));
  if length(S) > 0 then ShowMessage(S);
End;

```

Create a word document named test.docx and insert a bookmark named MyBookmark.

In Crystal XE, push on the button.

The bookmark must change and a randomized value must be displayed.

17.3.2. From Crystal XE execute a method in Word

Print preview (The document must be already loaded)

```

App.DDE.Execute('winword', 'c:\temp\test.docx', '[FILEPRINTPREVIEW]', false);

```

Save document as...

```
App.DDE.Execute('winword', 'C:\temp\test.docs', '[FileSaveAs.Name="NewName.docx"]', false);
```

17.4. From Excel or Word, send data to Crystal XE (Using Macro / DDEExecute)

In Excel or Word, this is possible by writing a macro, so you must show the developer tab. The Developer tab isn't displayed by default, but you can add it to the ribbon.

1. On the File tab, go to Options > Customize Ribbon.
2. Under Customize the Ribbon and under Main Tabs, select the **Developer** check box.

You need to use the method App.DDE.Execute and to give as parameter the name of the function with eventually the appropriate parameters for this function preceded by "Run|"

The syntax is : DDEExecute DDEChannel, "<PropertyName> := <Constant Value or String>"

Example1: Open and Close the shutter of Cell As

From the *Developer* tab, in the group *Controls*, click on the button *Insert* and select a button (from the control and not from the ActiveX).

Draw the button on the sheet and enter the following macro:

```
Sub OpenAs_Click()  
DDEChannel = DDEInitiate("crystalxe", "property")  
DDEExecute DDEChannel, "C21DZ.As_VAC500_P1.Shutter.Control:=1"  
DDETerminate DDEChannel  
End Sub
```

```
Sub CloseAs_Click()  
DDEChannel = DDEInitiate("crystalxe", "property")  
DDEExecute DDEChannel, "C21DZ.As_VAC500_P1.Shutter.Control:=0"  
DDETerminate DDEChannel  
End Sub
```

It is also possible to execute several actions:

```
Sub OpenAs_Click()  
DDEChannel = DDEInitiate("crystalxe", "property")  
DDEExecute DDEChannel, "C21DZ.As_VAC500_P1.Shutter.Control:=1"  
DDEExecute DDEChannel, "C21DZ.P_KPC250_P2.Shutter.Control:=1"  
DDETerminate DDEChannel  
End Sub
```

Example2: Change the selected index of a combo Box

Add a comboBox in the main user view.
Add at least 3 items and execute this macro.

```
Sub SelectId1()  
DDEChannel = DDEInitiate("crystalxe", "property")  
DDEExecute DDEChannel, "App.forms.MainUserView.Combo1.ItemIndex:=1"  
DDETerminate DDEChannel
```

End Sub

The second index (begin at zero) will be selected.

17.5. DDE Exchange with LibreOffice Calc

17.5.1. Display live data from Crystal XE (Hot DDE Link)

In Calc the syntax is `=DDE("crystalxe";"property";"<Property name>")`

Example to display the error status of Crystal XE

In a cell enter the following text: `=DDE("crystalxe";"property";"App.ErrStatus")`

This will display the main error status flag of Crystal XE (0 or 1)

Other example to display the temperature value of a cell:

`=DDE("crystalxe";"property";"Chamber1.EffusionCell1.tip_temperature.mv")`

17.6. More information on DDE

<https://www.angelfire.com/biz/rhaminisys/ddeapps.html#DDEExcelSpec>

18. OBJECT LINKING AND EMBEDDING (OLE)

Object Linking & Embedding (OLE Automation) is a proprietary technology developed by Microsoft that allows embedding and linking to documents and other objects. Crystal XE uses this protocol to exchange data and control Open office, Libre Office (Writer, Calc, Impress), Excel and Word.

Thanks to the OLE automation, Crystal XE can send data or execute actions to other compatible applications.

Script functions allow these exchanges:

- App.OLE.LoadDocument
- App.OLE.SaveDocument
- App.OLE.Close
- App.OLE.Poke
- App.OLE.Request
- App.OLE.ExportPDF
- App.OLE.RunMacro
- App.OLE.TableCreate
- App.OLE.TableAddRows
- App.OLE.TableSetCell
- App.OLE.ShowDocument

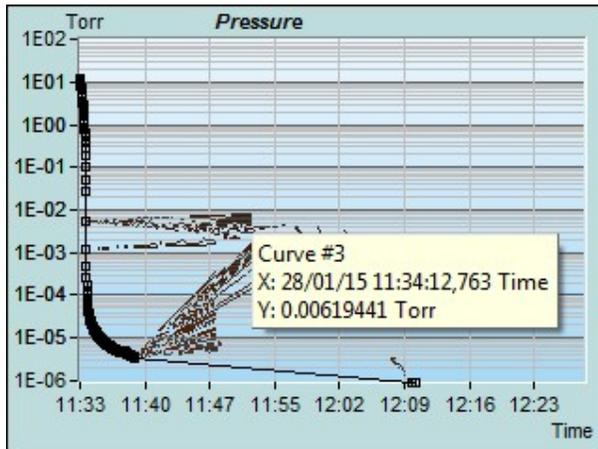
All these functions are used for example to generate automatic reports in the recipes. These reports can be PDF, XLS, DOC, ODT, etc...

For more details on these functions, please refer to the OLE functions in the script help or in the Crystal XE user manual.

19. FAQ

19.1. Display problem in charts

If you encounter any problems of display like in the following picture:



If you are under Windows XP:

- There is no alternative but to disable the display of information balloon.

If you are under Windows 7 or windows 8:

Turn on the aero mode:

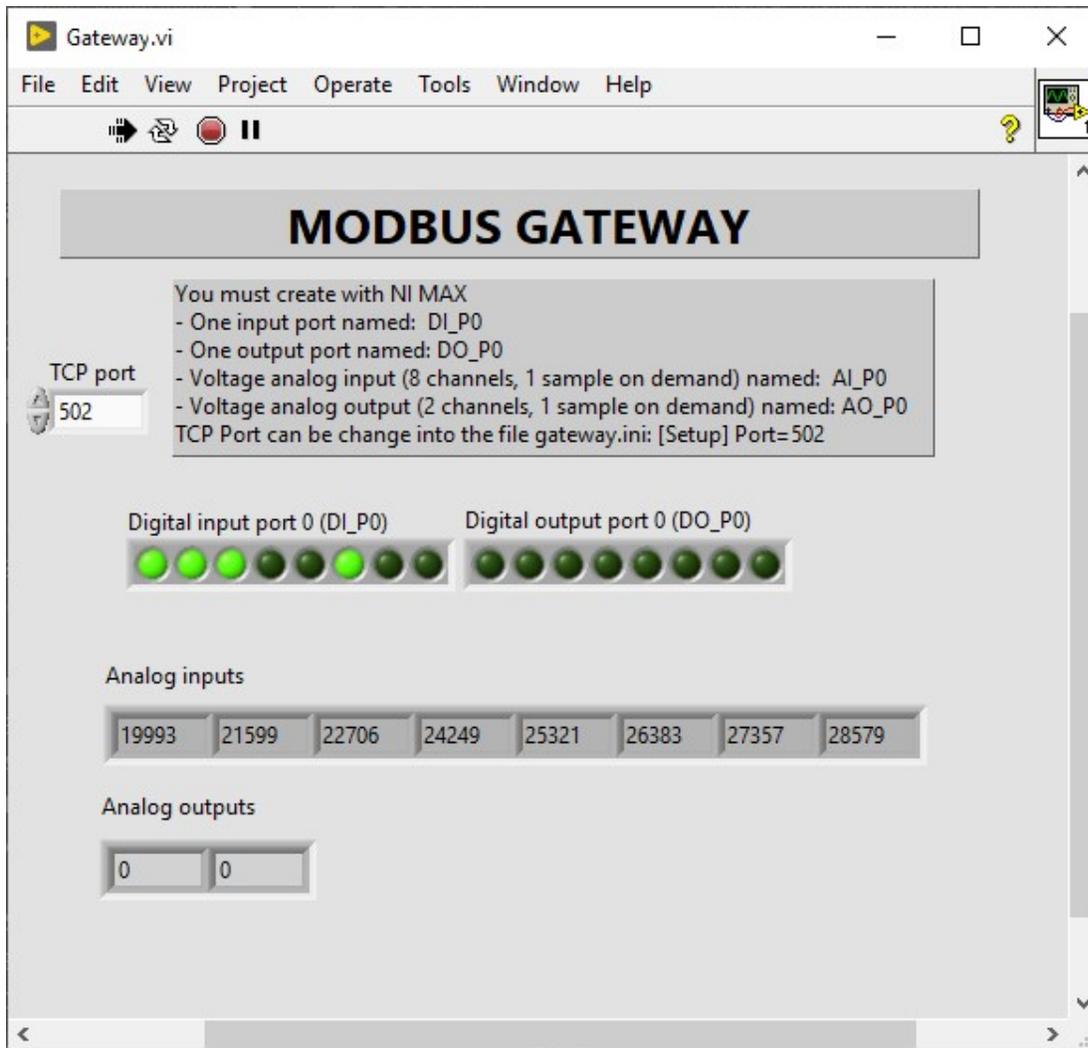
Right click on the desktop background and then click on the item "Personalize".
Select aero Theme.

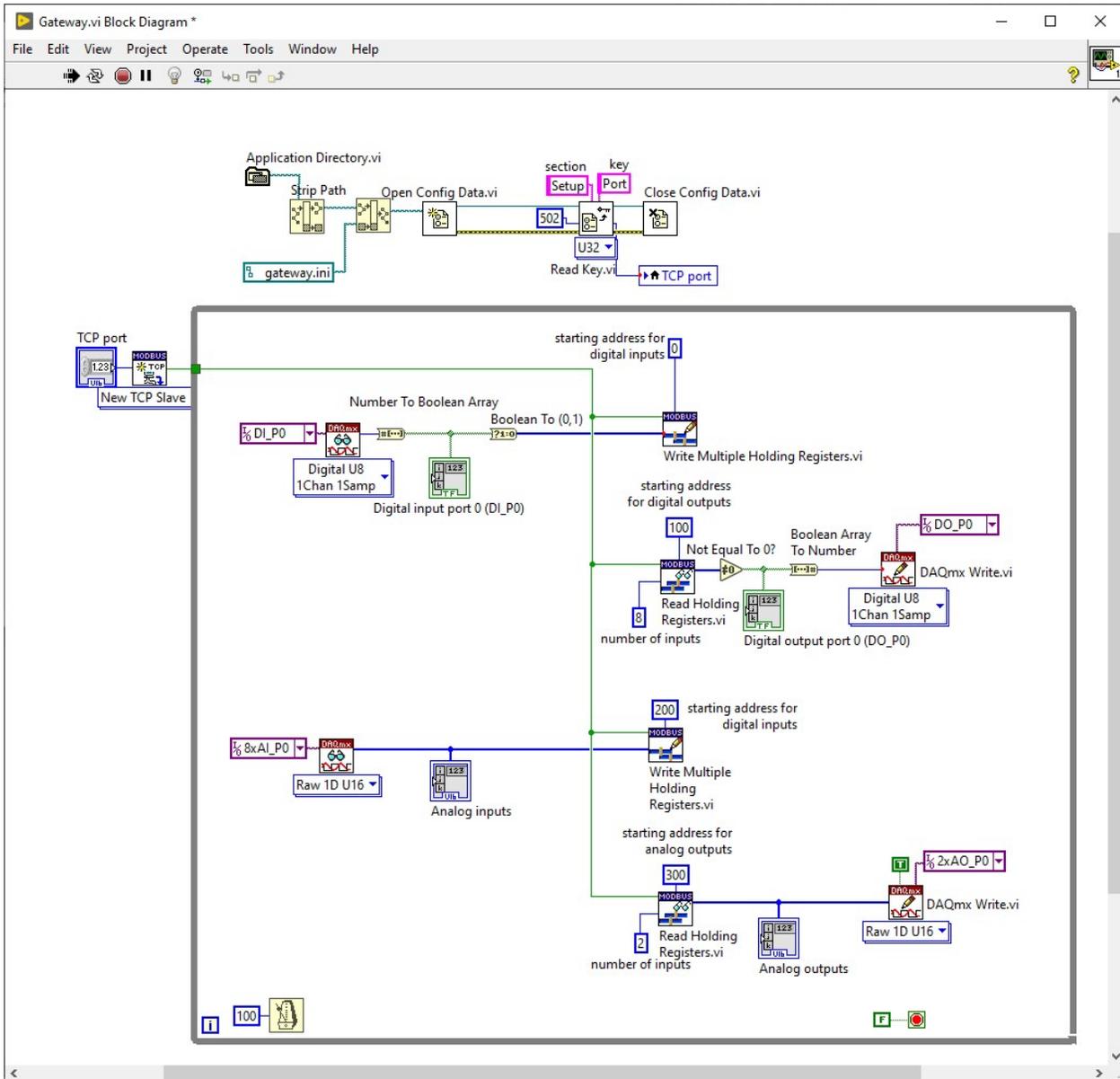
If the problem persists, enable the DWM (Desktop Window Manager)
Click on Start and type "services.msc" in the search field and press enter.
In the "Services" window, look for the entry "Desktop Window Manager Session Manager"
Double click and set the "Startup type" as "Enabled" and click on Start

19.2. Using of National Instruments cards

One of the solutions for using cards from National Instruments is to use a modbus gateway developed in Labview.

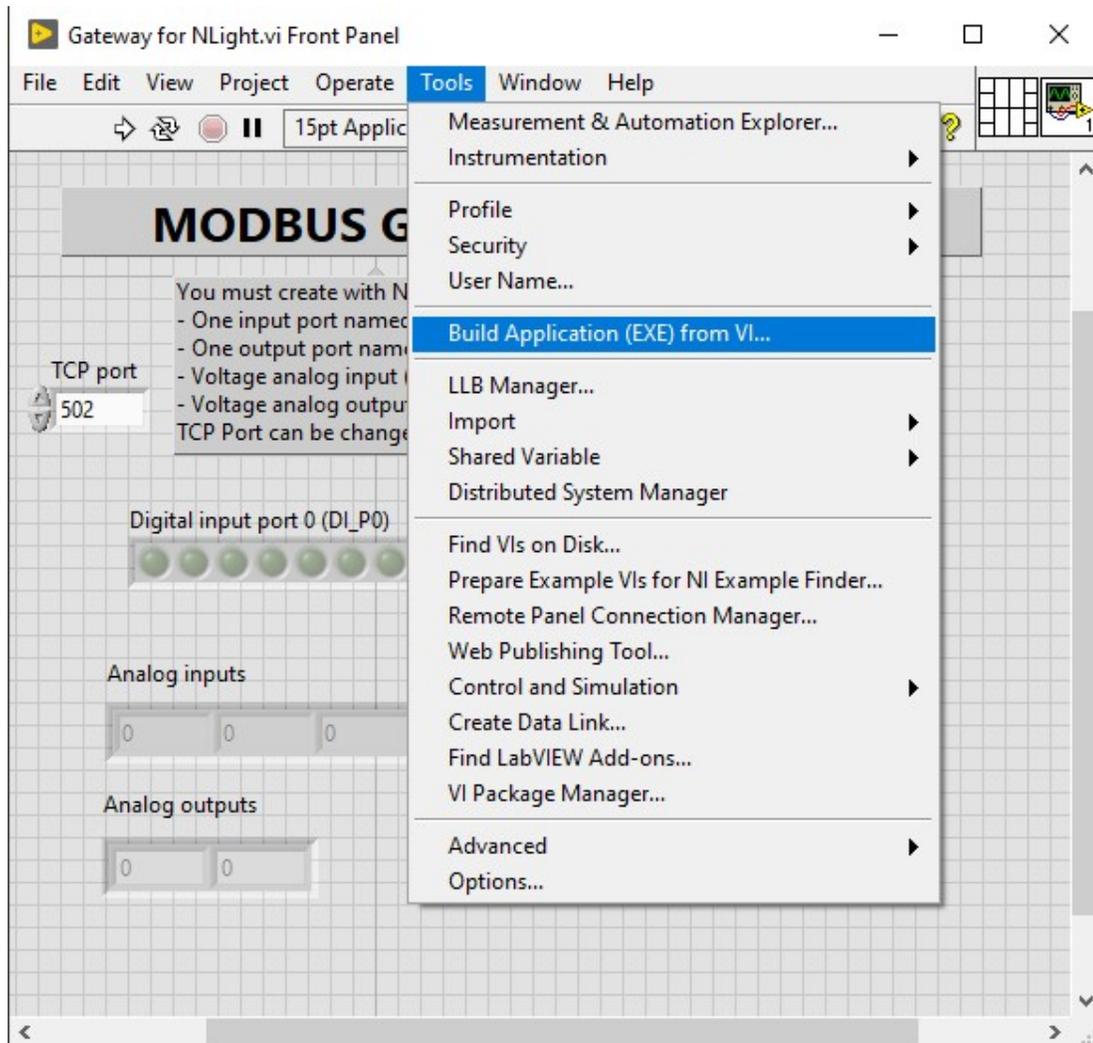
The virtual instrument file which is given in the example below is available on request.





19.2.1. Installation procedure

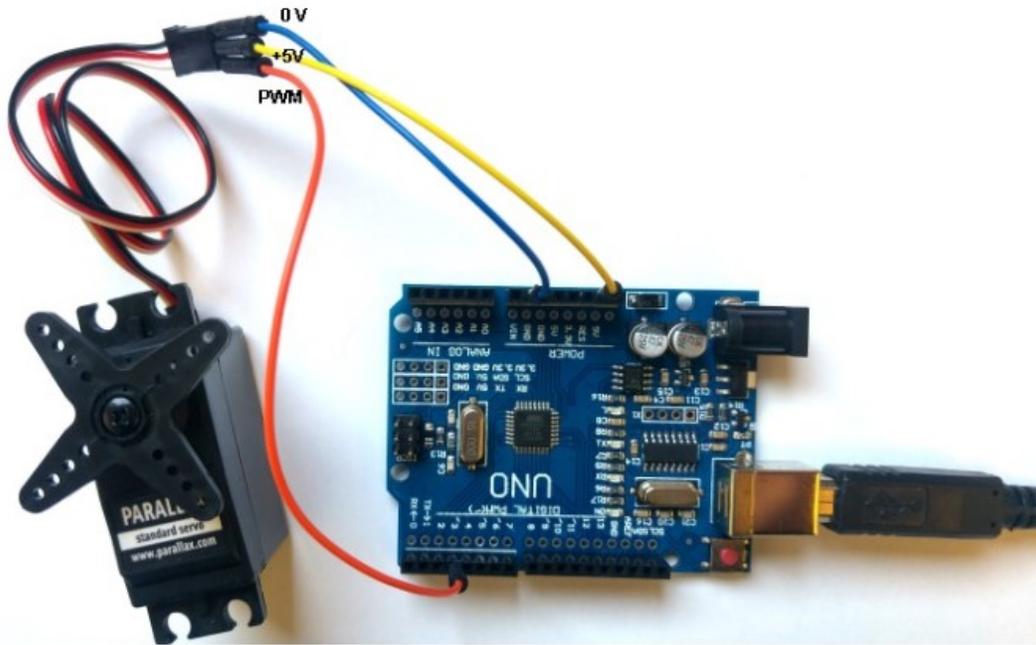
- 1) Create the executable file



- 2) Copy all files needed for the gateway (executable files and INI files) in a folder of the computer, by example, in the folder c:\Riber\Gateways
- 3) Download and Install NI DAQMx (today in 2020, version 19.5)
Link: <https://www.ni.com/fr-fr/support/downloads/drivers/download.ni-daqmx.html#325032>
- 4) Download and Install the Labview Runtime
Link: <https://www.ni.com/fr-fr/support/downloads/software-products/download.labview.html#329458>
- 5) Run the executable files of the gateway
- 6) Run Crystal XE
- 7) In Crystal XE, create a Modbus client that will communicate with the gateway, either in localhost or on a remote computer.

19.3. Using Crystal with Arduino

See the Application note AN005 available on <http://www.crystalxe.com/faq>



20. ANNEXE A - Write a recipe by example

The following example details how to create a new recipe based on the demo C21 DZ project.

This recipe is intended to grow a simple conventional HEMT structure As-based on a C21 DZ using RIBER cells mounted onto the system. Namely a VAC 500 for As, an ABI85 for Ga, a ABN60DF for Al and a ABN 135 DC8 for Si. Manipulator is the standard 4" oven allowing to grow on up to 3" diameter wafers.

It is just an example to show the main features of the recipe editor, set-point temperature values and valve positions varying from system to system. Indeed end-user own procedure may differ.

For demonstration purpose this chapter has been splitted into several parts:

Section 8.1.1: We create a first recipe, intended to be a sub-recipe named *deoxidation GaAs* in the final *HEMT* recipe. It will be used to remove oxide from the GaAs wafer prior to growth. Thus it only concerns manipulator and As cell parameter management.

Section 8.1.2: More information about main recipe editor features, based on the recipe edited in part 1.1.

Section 8.2.1: The final recipe will include at first step the execution of the sub-recipe prepared above, that is oxide removal prior to growth. Then we will create layers that will be directly included in the final recipe, corresponding in fact to the buffer growth. For demonstration purpose, this series of layers will include a 'loop', corresponding to the growth of a super-lattice GaAs/AlGaAs prior to active layer growth.

Section 8.2.2: More in-depth information about loop usage.

Section 8.3.1: after the series of layers created to describe the buffer growth, we will create two more sub-recipes. Namely, the *HEMT active layers* and the *end GaAs growth* sub-recipes. They will be edited in a very similar way to the *deoxidation GaAs* sub-recipe so no further detailed information here.

Section 8.3.2: More in-depth information based on final recipe edited in part 3.1, such as simple script usage, use of pause and offset function.

Now recipe is fully edited. In Section 8.4 one can find the other two presentations of the recipe: in form of a table, the recipe overview, and in form of graphs for the various parameters such as temperatures and valve positions.

Section 8.5 briefly describes some specific features that can be useful, during or after recipe execution.

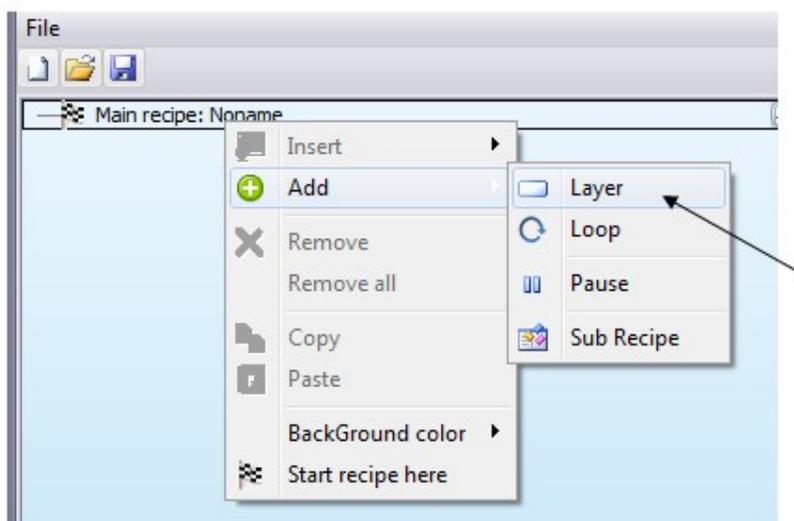
1.1	Operating basics	Remove oxide from the wafer
1.2	More information about recipe edition Layer information Background color More about ramping Insert, copy/paste layers Sub-recipe Ramp duration smaller than layer duration Recipe start at a specified layer number Use of non-linear ramps Where is my recipe window?	
2.1	Loop edition	Buffer growth
2.2	In-depth information about loop usage Nested loops 'Slow' ramp Shutter initialization at loop start Case study: super-lattice InAs/GaSb	
3.1	Recipe edition: finalization	Active layer growth Growth stop
3.2	In-depth information about recipe edition Pause function How to mix script and recipe Configurable recipes Offset editor Growth rate editor	
4	Recipe edition: additional viewings Graph display Table view	
5	Other features: How to follow recipe execution progress? How to retrieve data from recorded files	

20.1. Operating basics

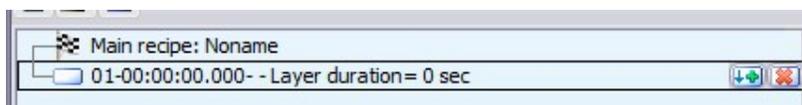
- First create a new recipe:



- Right-click on the 'Main recipe' line left side and create a first layer by selecting 'Add\Layer':



- An empty layer is added to the recipe tree left side of the recipe editor:



A 'layer' in Crystal terminology is a recipe line within which, when executed, various parameters such as temperature set-points, shutter positions, valve positions, ... will be modified. As explained below, some parameters will be modified immediately, some others as a ramp generally based on the layer duration. A parameter modification is called an 'event'.

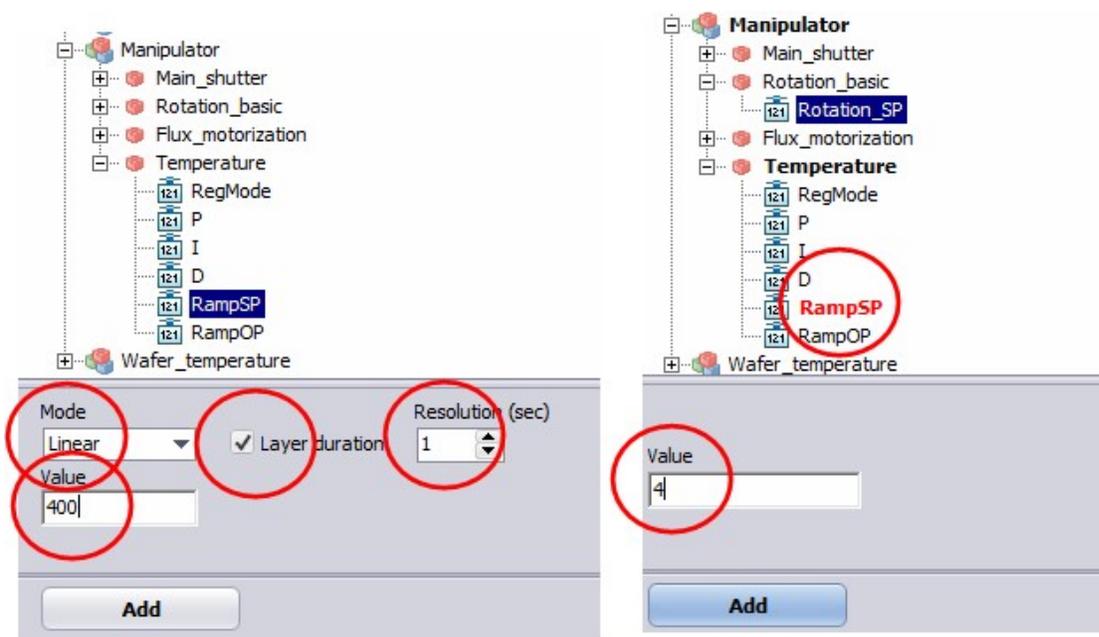
This first layer will be used to start heating the manipulator on which the platen equipped with the GaAs wafer has been transferred. Typically transfers are performed with no power applied to the manipulator filament, and a temperature reading below 300°C. Thus as a first step the manipulator set-point is increased here to 400°C (with no As incoming flux yet), and a low speed rotation, here 4 rpm, set (to check from time to time RHEED pattern appearance in all directions).

The manipulator temperature could be increased as a step, or linearly. If **linear** is chosen, as in this example, a new set-point will be sent to the corresponding temperature controller channel according to the set resolution, in this case every **1 sec**. Indeed in most cases temperature set-points are modified as ramps for a better control.

Note that the ramping rate will be automatically calculated from the layer duration, based on the starting and ending temperature set-points.

If there is no starting temperature, such as in this case since this is the very first layer of the recipe, the starting temperature will be the current set-point applied.

Ramping rate calculation is straightforward : for example if the starting set-point is 250°C (the set-point taken from the previous layer) and the ending set-point 400°C (the one which is set as a new event), with a layer duration set to 6 min, the calculated ramping rate will 25°C per min.

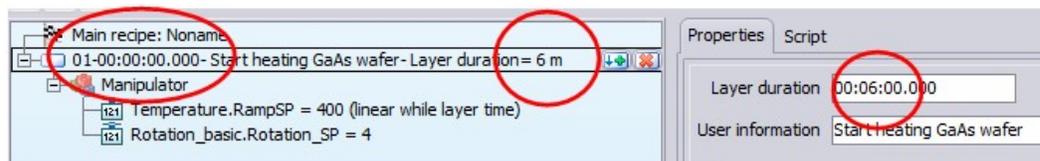


Events can be edited one by one or simultaneously. In the second case, already edited events appear in red color in the event tree window (such as RampSP above).

The rotation speed of the manipulator is entirely controlled by the RIBER ACM controller (acceleration, deceleration), no ramping ramp can be used for this parameter.

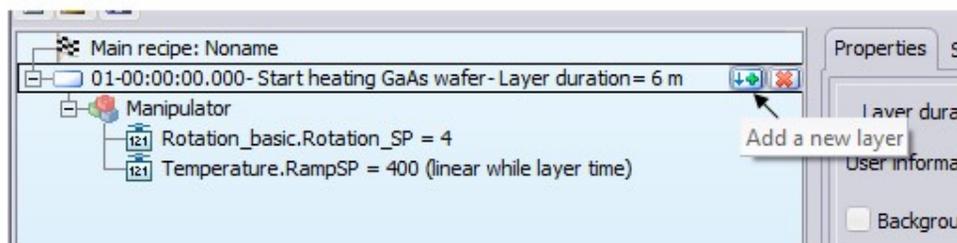
Typically before transferring the platen into the growth chamber, the temperature is set to 250°C, to make sure as explained above that real temperature is low enough and no power is applied to the filament.

- So setting a duration of 6 min, 00:06:00:000, will correspond to a ramp rate of 25°C/min to reach 400°C after 6 min.



Next the temperature of the manipulator will be slowly further increased under As flux. At some point deoxidation will be observed by eyes, but the temperature will be increased slightly higher, to make sure deoxidation is fully completed, and the wafer degased at a temperature high enough to remove contaminants such as C. So both main shutter and As cell shutter will be opened at the beginning of the second layer. In order not to spill too much As, the valve will be slowly opened as the temperature increased. Also for a better control the ramping rate will be decreased at an intermediary temperature, so two layers will be used for this step.

- Create a second layer, either by selecting again the very first line of the recipe tree and right-clicking, or selecting the first line and clicking on the small dedicated icon:

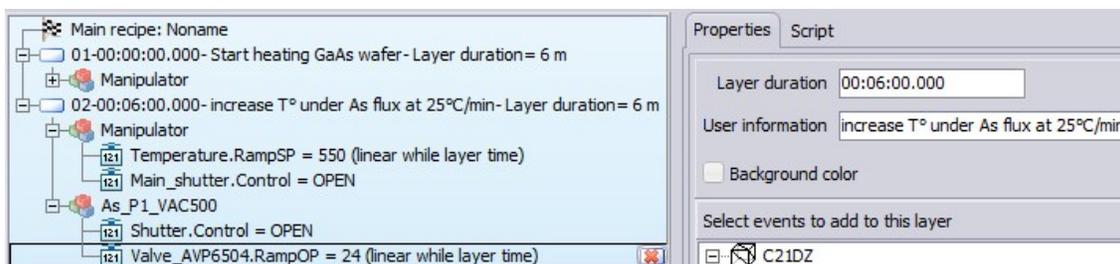


- In this second layer the manipulator temperature is increased (400°C) to 550°C in 6 min (00:06:00.000), so still at 25°C/min ramping rate, and the As valve linearly increased from its starting position (most likely 0%) to 24% in 6 min as well (so at a ramping rate of 4%/min). Both As cell and main shutter are opened.

There is no further event for the manipulator rotation speed, so the previous setting, 4 rpm, will be kept on until a new event is declared.

There is no event for both the cracker and the reservoir temperatures of the As cell, so the current set-points will be kept, until a new event, if any, is declared.

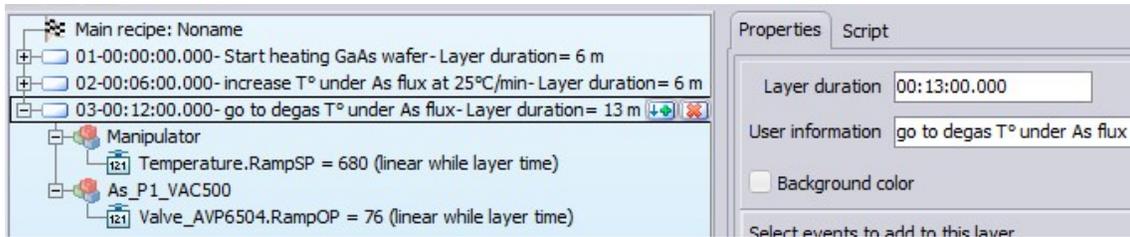
Note that if no event is defined for a given equipment, when running a recipe, the corresponding tag will be kept at its last known value from its controller.



A shutter position modification, open <=> closed, is immediately set at the start of the layer, independantly of the layer duration.

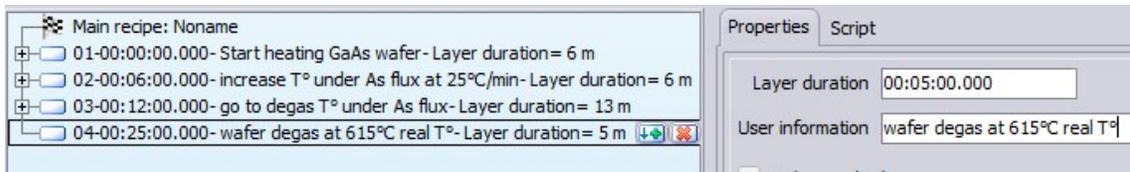
- Create a third layer. In this third layer the manipulator temperature will be increased from 550°C to 680°C in 13 min (00:13:00.000), so at a ramping rate of 10°C/min. Meanwhile the As valve will be linearly increased to 76% (so still at a ramping rate of 4%/min).

There is no further event for As cell shutter and main shutter, so both are kept open until a new event is declared.



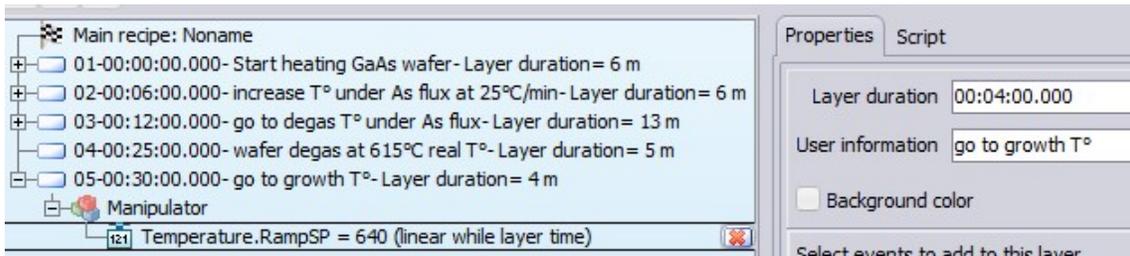
These conditions will be then kept for 5 min in order to degas the wafer. In this example 680°C is the set-point value for the manipulator thermocouple, the corresponding real temperature being roughly 615°C. We assume in this example that deoxidation occurred at about 640°C, that is a real temperature of about 585°C. The As valve position, 76%, correspond to the targeted As flux for the following growth initial process.

- Create a fourth layer with no event at all, just a duration of 5 min (00:05:00:000). That means all conditions of previous layer will be kept on.



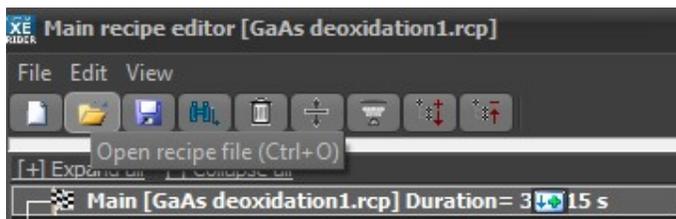
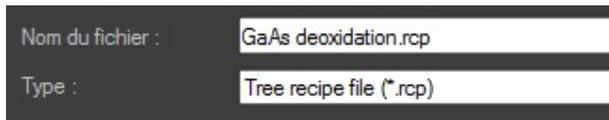
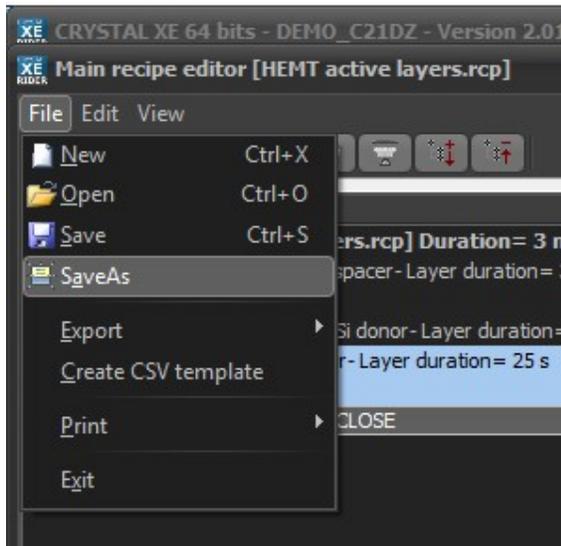
Then the manipulator temperature set-point will be brought down to the value intended for growth, typically about the temperature at which has been seen deoxidation. In this example 640°C, that is a real temperature of about 585-590°C.

- Create a fifth and last layer within which the manipulator temperature will be brought down from 680°C to 640°C in 4 min (00:04:00.000), so still at a ramping rate of 10°C/min.

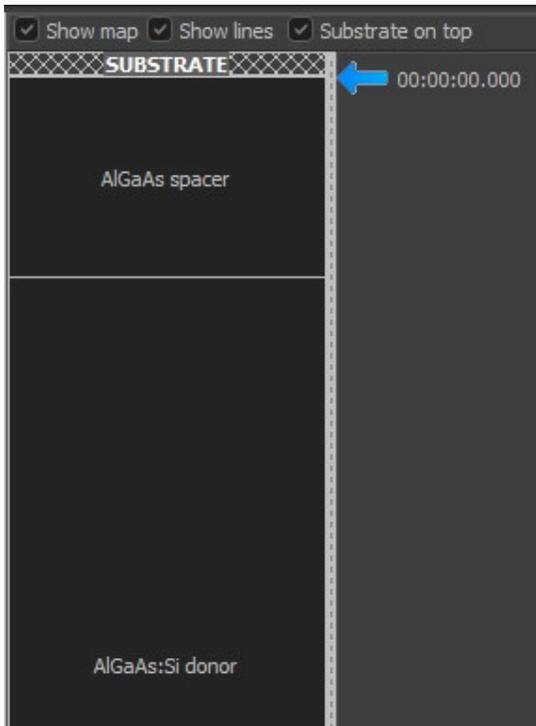


Note that in this example the manipulator rotation speed is kept at 4 rpm. This is because most likely upon starting the GaAs buffer, one wants to have a look to the RHEED pattern to see if the growth conditions are right. But then the manipulator speed would be increased, typically up to 20 or 30 rpm.

- Save the recipe. The default recipe path can be changed to a specific path if needed. Indeed recipe can be saved any time.



- After saving a schematic of the intended sequence of layers is shown on the right side:

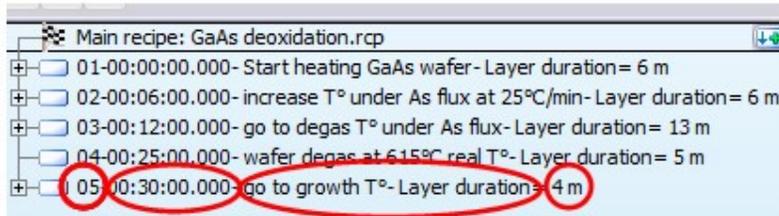


20.2. More information about recipe edition

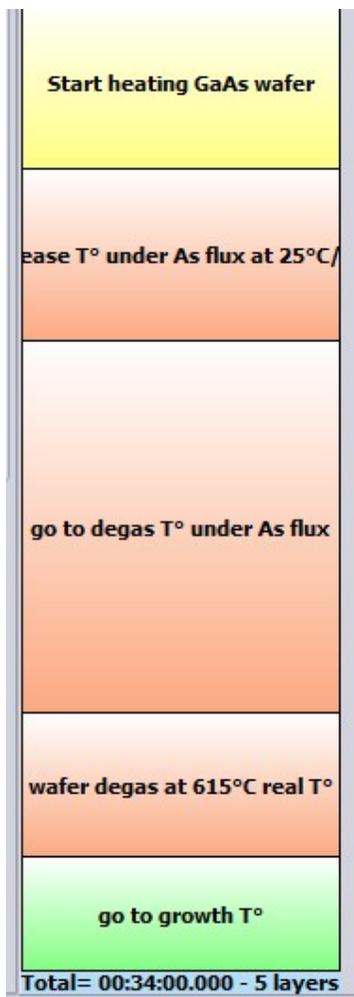
- Layer information:

The recipe tree recalls for each layer some general information:

- its position in the recipe,
- the time at which it will be executed after the recipe has been launched,
- the comments set by the user in the corresponding box left side,
- the duration of the layer.

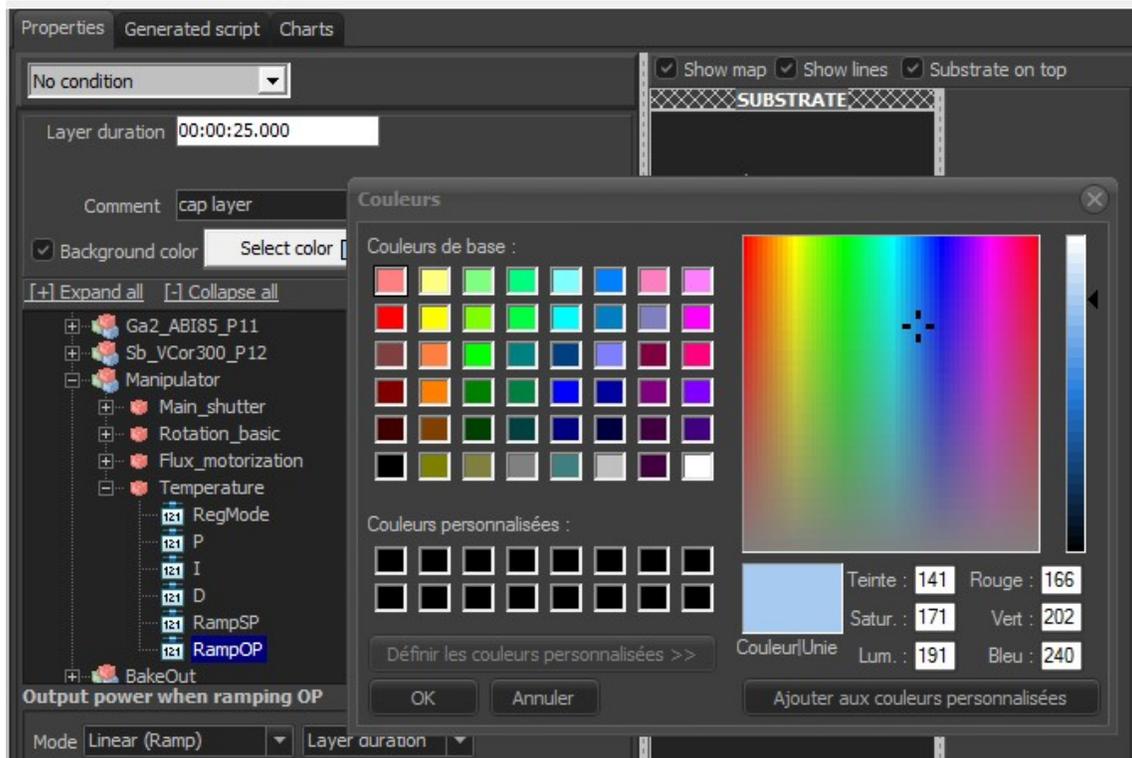


The layer stack displays the total number of layers. Layer number currently under process while running a recipe is also display in the recipe information box.

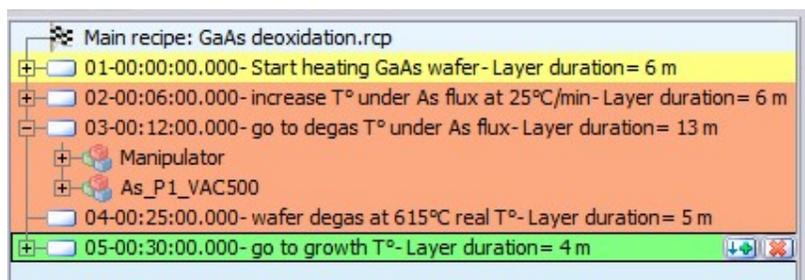
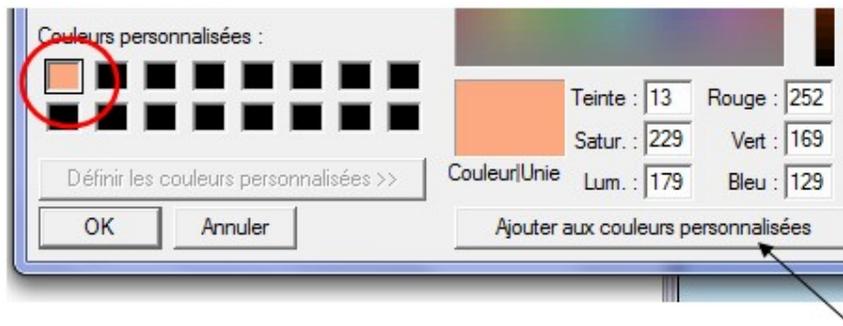


- Background color:

To make viewing easier it is possible to attribute some colors to any of the layers included in a recipe:



There is no propagation mode but one can easily make its own color code to separate between recipe layer sequences:

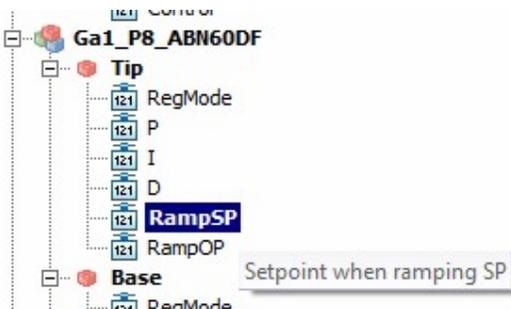


- More about ramping:

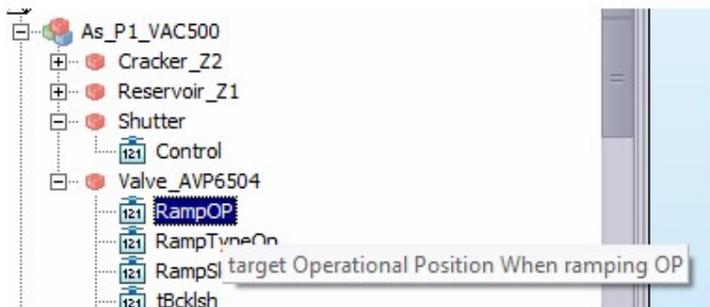
This is the Crystal XE software that manages all set-point modifications while ramping during recipe execution, not the dedicated controllers themselves. That is every x seconds, depending on the resolution chosen, a new set-point is sent to the corresponding controller. This solution has been chosen to be consistent with the offset management, see dedicated section.

Crystal XE terminology:

Whenever ramping a temperature set-point, modify the tag **RampSP**:



Whenever ramping the valve set-point (% opening) of a valved cell, modify the tag **RampOP**:



- Insert, copy/paste layers:

To insert a layer (as well loop or sub-recipe), select the line above which you want the new layer to be positioned and right click to do the action.

To copy/paste:

Right click a layer to copy its content.

Then select the layer below which you want this layer to be copied.

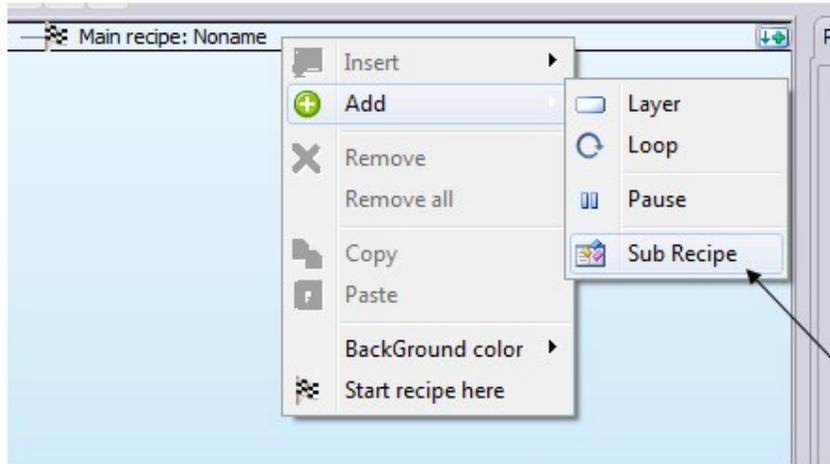
For example select layer #03 and copy it. Then select layer #04 and paste it. The copy of layer #03 will then be inserted between layer #04 and previous layer #05, which becomes now layer #06.

To copy a layer and insert it in first position, insert first a fake layer (New\Layer) in first position, then paste the copied layer below it, and last remove the fake layer.

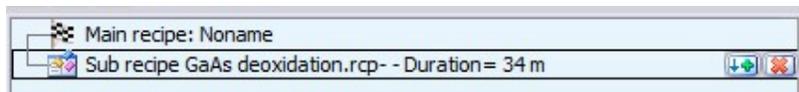
- Sub-recipe:

Any created recipe can be inserted as a whole in another recipe, to make easier viewing and minimize mistakes while editing/copying an existing recipe. In this example the recipe created will be included as a sub-recipe placed in first position in the final recipe we intend to edit.

Click as follows to add/insert a sub-recipe:



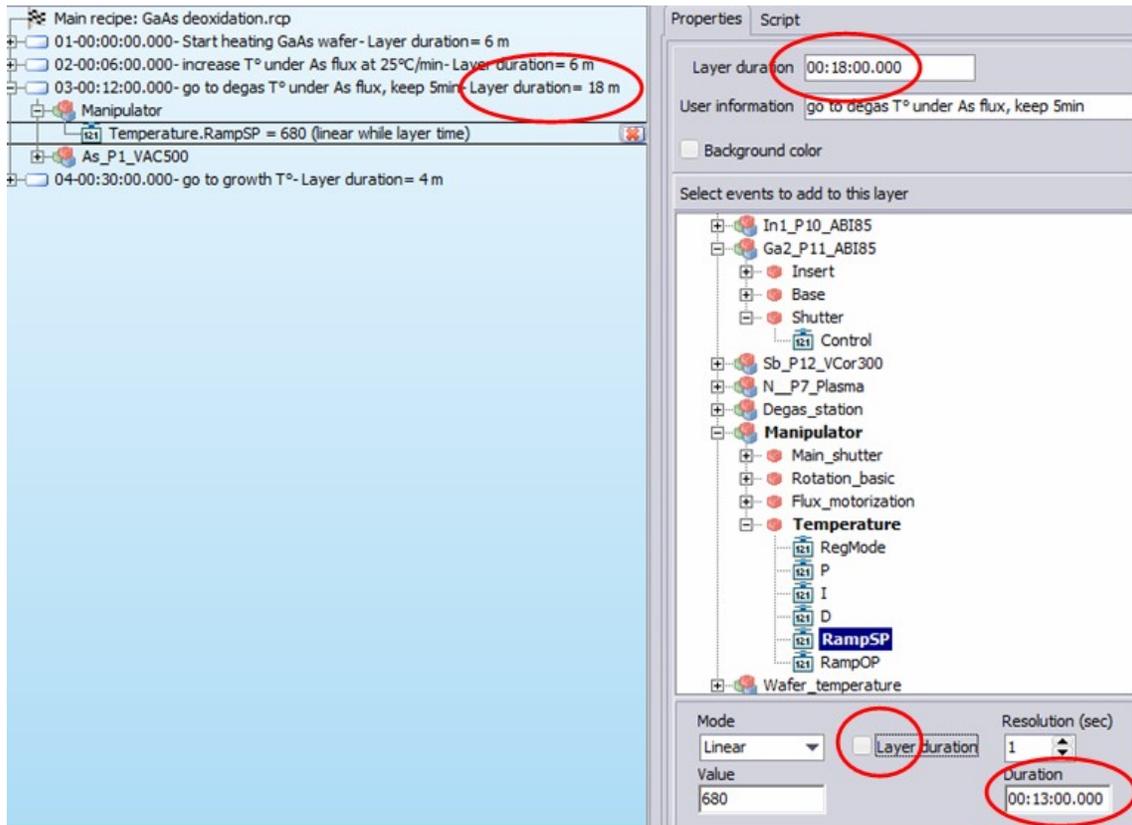
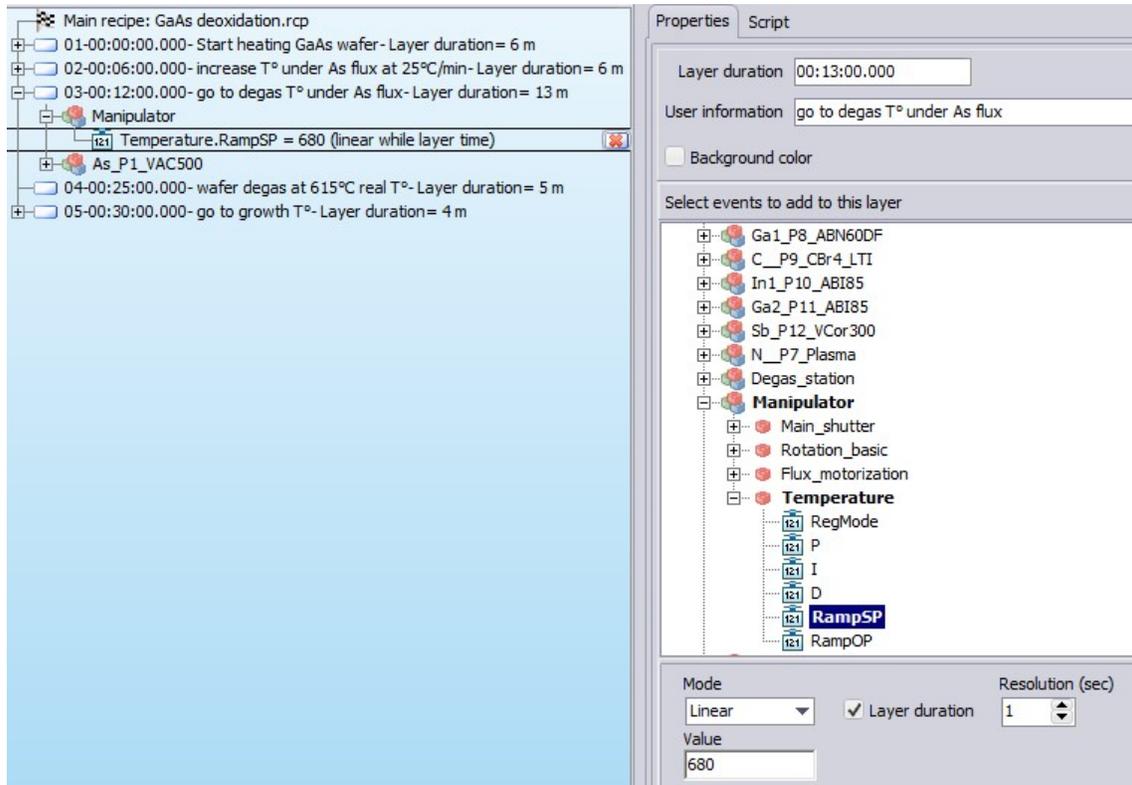
Click on the sub-recipe to see its detailed sequence.



A sub-recipe can be inserted in any position, first line, in-between layers, last line. It can also be inserted if needed into a loop. See dedicated section for loop usage in recipe.

- Ramp duration smaller than layer duration:

It is possible to make a ramp duration to be shorter than the duration intended for the layer within which it is incorporated. To do that unclick the 'Layer duration' box, a new box 'Duration' will appear. Set then the intended duration for the parameter ramp.



In above example the total duration of layer #03 has been modified to be 18 min. But the manipulator new set-point, 680°C, will still be reached in 13 min. That means for the next 5 min of this layer, the manipulator temperature will be kept at the same value, 680°C.

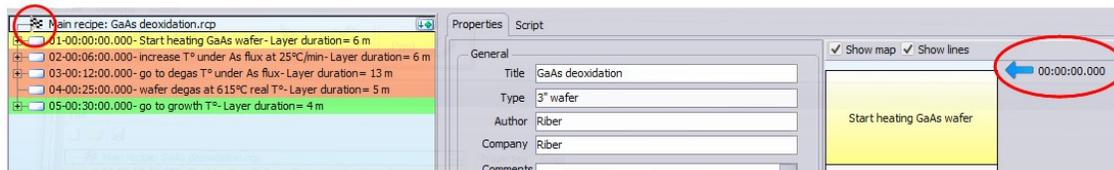
If the same has been done for the valve position, ramp up to 76% in 13 min, the execution of layer #03 of the second case will strictly be equivalent to the execution of layers #03 and #04 of the initial case. This feature permits to simplify editor viewing. Just make sure of what has been done since there is no warning that a ramp duration is smaller than the layer it is incorporated in.

- Recipe start at a specified layer number:

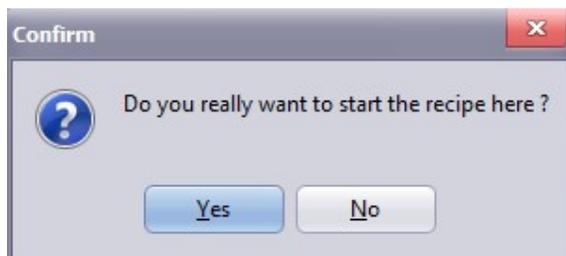
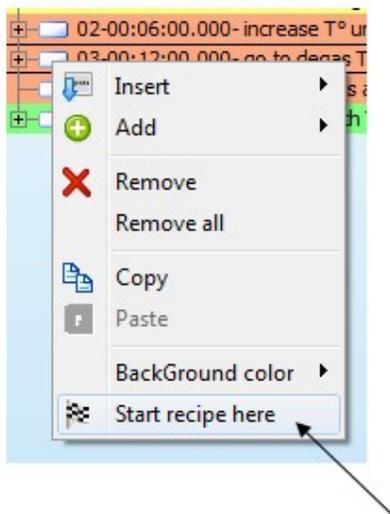
It can be interesting to start a recipe not at the very first line, but at a user-defined line number.

For example recipe execution has been aborted for some reason, and after some checks one wants to restart at the position it has been stopped, all parameters still being set according to the last layer executed.

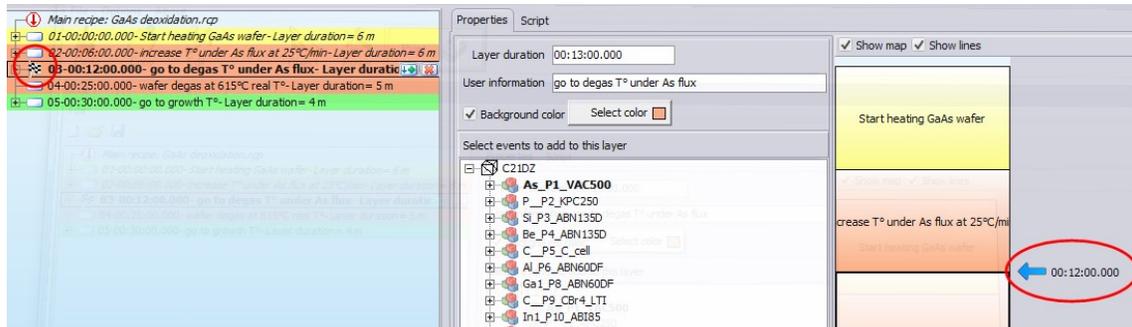
By default the start flag is positioned at the first line, so the recipe starts from first line:



- Select the layer at which you want the recipe to start and right-click to select the 'Start recipe here' option:



In the example below, the recipe will start now from layer #03, layers #01 and #02 being ignored:



- Use of non-linear ramps:

In case some layers differ only by set-point variations, no shutter commands, it is even possible to compress further the number of layers by using the non-linear ramp option possibilities.

Indeed this specific feature can also be used if a very particular profile is expected, such as a non-linear temperature variation in case of graded composition or doping concentration within one layer.

Going back to the initial recipe detailed above, layers #2, 3, 4 and 5 only differ by their manipulator temperature set-point and their As cell valve position.

In details, the manipulator temperature set-point varies as follow:

- go from 400°C to 550°C in 6 min => ramping rate 25°C/min
- go from 550°C to 680°C in 13 min => ramping rate 10°C/min
- keep at 680°C for 5 min => plateau
- go from 680°C to 640°C in 4 min => ramping rate 10°C/min

It is possible to create a specific non-linear profile nearly fitting these variations:

One just have to define some reference points for the X and Y-axis, scaled from 0 to 1 both of them.

X-axis: 0 corresponds to time '0', 1 corresponds to time '28 min'

Y-axis: 0 corresponds to temperature '400°C', 1 corresponds to temperature '640°C'.

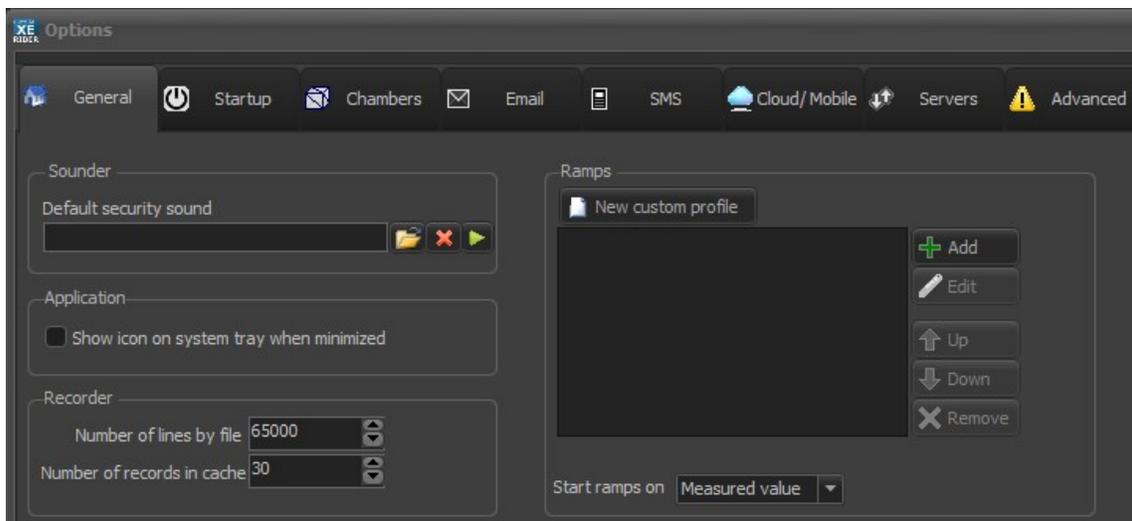
It is easy to calculate that the following data point series will nearly describe the expected profile:

- (0, 0)
- (0.214, 0.625)
- (0.679, 1.167)
- (0.857, 1.167)
- (1, 1)

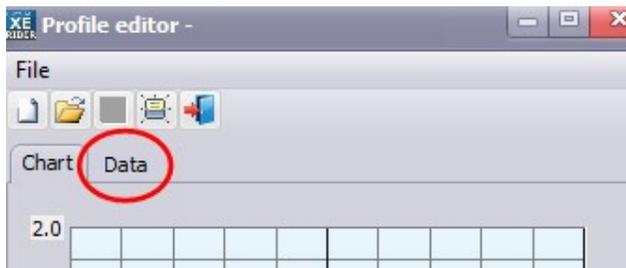
- To create this profile, open the option menu displayed in the main window upper right corner:



- Open the profile editor by clicking on 'General\New profile':

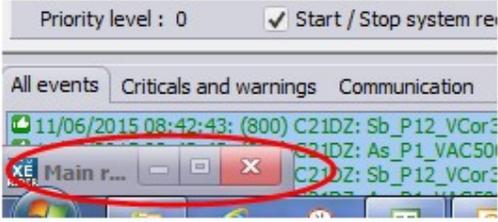


- Enter the data point series:



- Where is my recipe window?

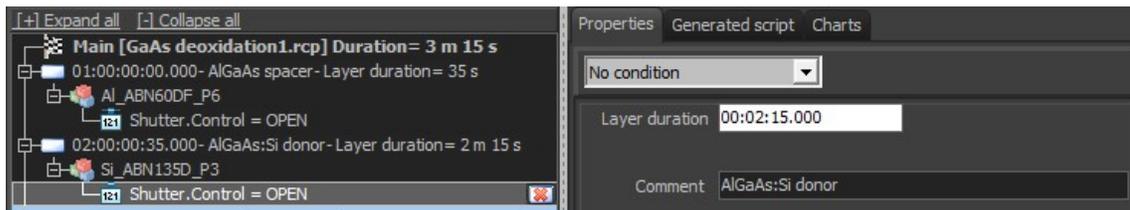
Look at the left bottom of the screen!



20.3. Loop edition

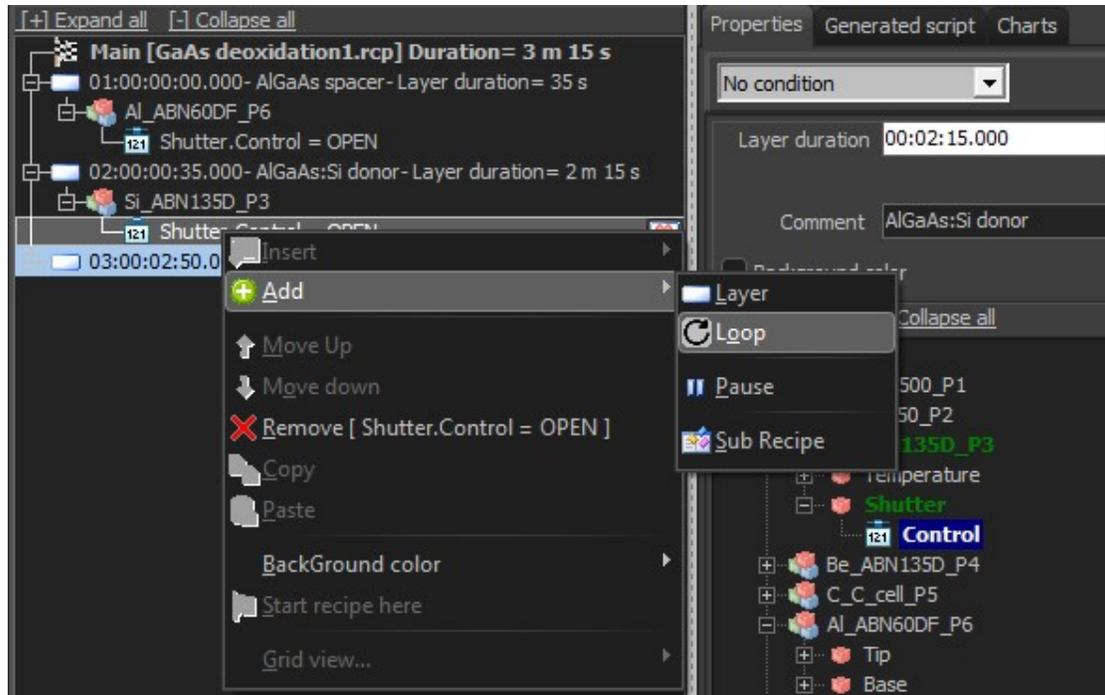
A series of layer will be edited corresponding to the GaAs buffer growth. For demonstration purpose, this buffer will include a super-lattice sequence consisting of 25 pairs of GaAs and AlGaAs thin layers, typically used to minimize dislocation propagation as well reduce the buffer growth time. We will use the loop function to edit this part of the buffer growth.

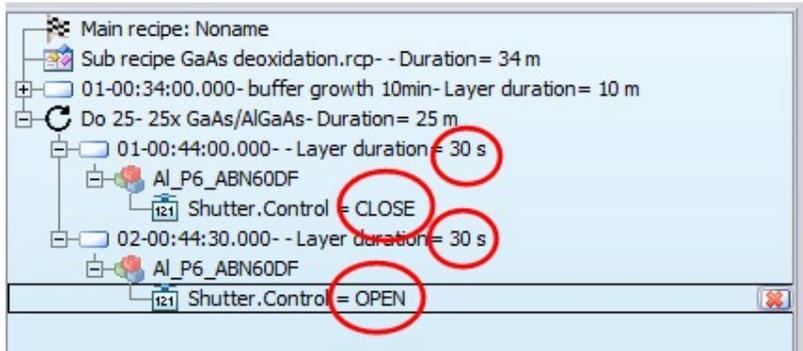
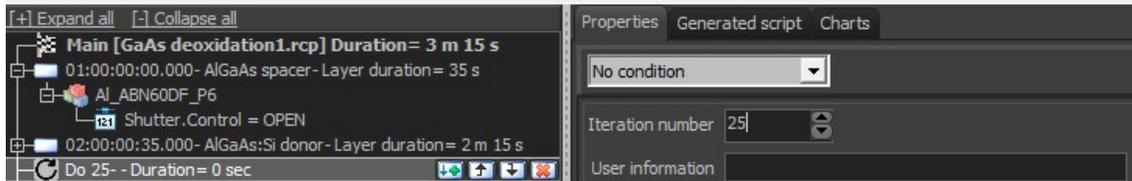
- Insert as a sub-recipe the *deoxidation GaAs* recipe created in part 1.1.
- Edit a first layer corresponding of the growth of a GaAs buffer for 10 min:



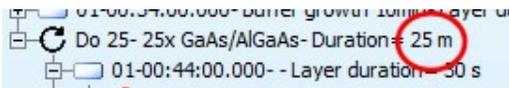
Note that it is only necessary to edit the Ga cell shutter position to effectively grow GaAs, since both the As and the main shutters are already opened from the sub-recipe executed right before this layer.

- Select the line just edited and create the loop corresponding the 25 pairs of GaAs and AlGaAs layers:

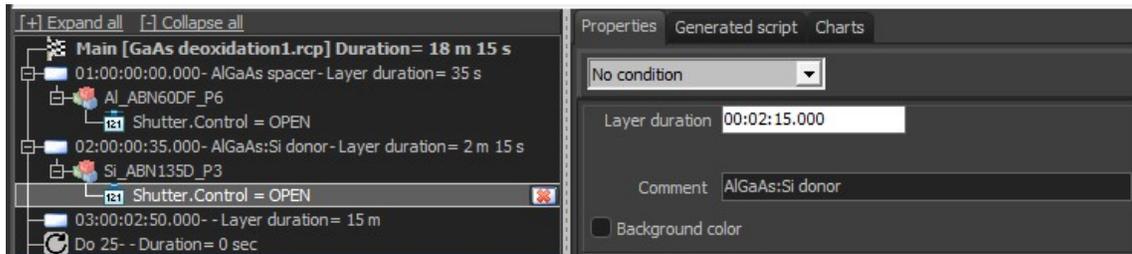




The total duration of the loop is displayed at the 'Do' level :



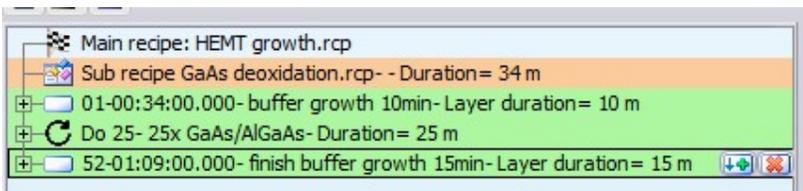
- Edit a last layer to finish buffer growth



Since the very last layer just before this layer corresponds to an AlGaAs layer, just close the Al shutter to grow again GaAs.

Layer numbering takes into account all layers included into the loop, so this new layer will be positioned number #52.

- Save the file, here *HEMT growth*:

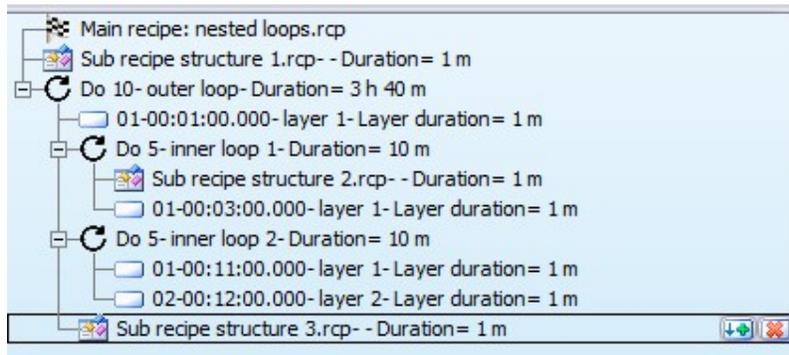


20.4. In-depth information about loop usage

- Nested loops:

There is no limitation to the number of loops that can be inserted into a recipe or sub-recipe.

The following example shows various possibilities, just for demonstration purposes:



Basically the recipe contains two main lines, first a sub-recipe, then a 10 iteration loop.

Then within this loop for each iteration execution, there are successively one simple layer, then a first nested 5 iteration loop, then another 5 iteration loop, and last one sub-recipe.

Deeper in the first 5 iteration loop consists of one sub-recipe followed by a single layer. The second 5 iteration loop consists of two simple layers.

Of course each sub-recipe used for this recipe could contain itself loops and nested loops.

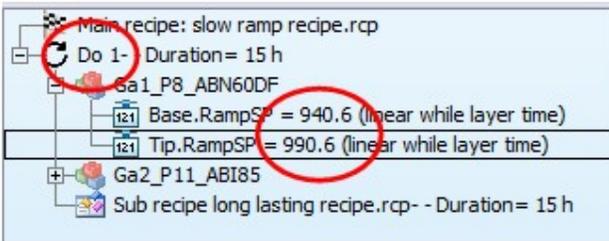
Note that the total number of layers calculated and displayed at the bottom of the layer schematic does not take into account sub-recipes, only simple layers.

- 'Slow' ramp:

In standard effusion cells, it may be necessary to take into account material depletion into the crucible to keep constant the growth rate of the corresponding element, especially if the recipe to be executed lasts for 10 hours or more. That is in fact increase the set-point temperature of the cell, to keep constant the outcoming flux, thus the growth rate.

After some calibration one can estimate by how much the temperature should be increased over the total duration of the recipe.

Then to smoothly increase over time the temperature, it is possible to modify the corresponding set-point by inserting the entire recipe to be executed within a loop consisting of one single iteration, and at the loop level, define the needed event:



When this recipe starts, the temperature will be linearly increased to reach respectively 940.6°C for the base, and 990.6°C for the tip after 15 h. The ramping rate will be calculated based on starting value. Since there is no temperature set in this recipe, the linear ramp will start from the actual set-point value. For example, the temperatures were last set to 940°C and 990°C before starting the recipe.

This means that the ramping rate will be in this case 0.04°C/min.

If a new recipe, one can directly create first the loop, then include all needed layers within this loop. If the recipe already exists, just add it as a sub-recipe in the loop.

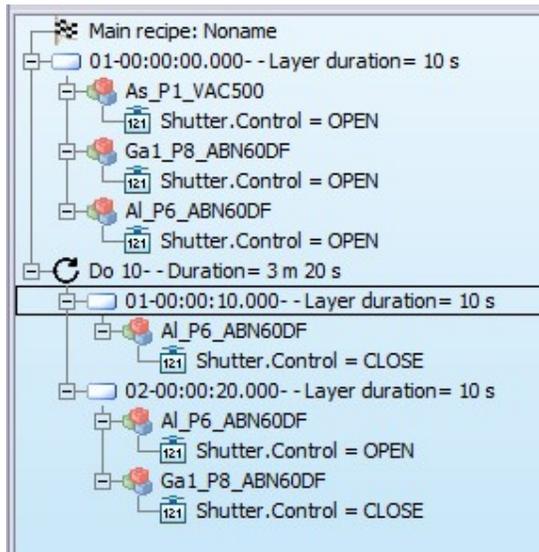
This specific feature could be used as well to linearly modify a temperature over a loop execution contained inside a recipe, thus vary overthrough it the composition or the doping level.

- Shutter initialization at loop start:

There is a special case for which, if not properly edited, the execution of a loop inside a recipe will not be as expected.

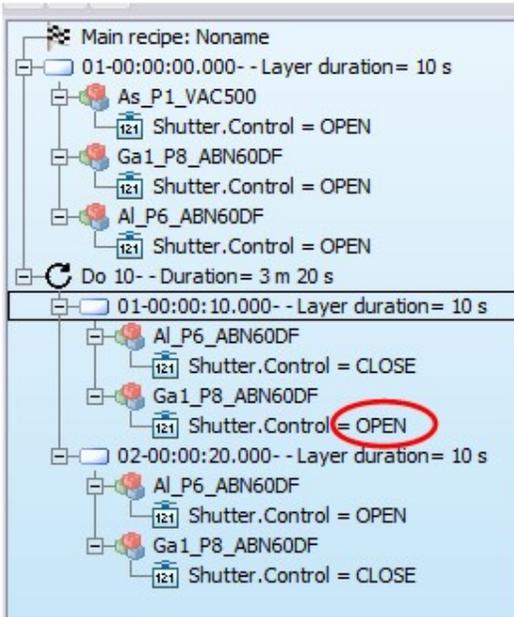
Let's take the following case as an example:

First scenario:



We grow AlGaAs just before loop start. For the first cycle, the first layer will be GaAs, since we close the Al shutter. The second layer of this first cycle will be AlAs. But for the second cycle, and all following, there is no indication for Ga shutter status in the first layer, so this Ga shutter would be kept closed, meaning that the loop is not executed as most likely expected.

Therefore, Ga shutter status should be specified, open, for the first layer as shown in second scenario:



- Case study, super-lattice InAs/GaSb:

Photodetector structure requests the use of many iterations of thin InAs and GaSb layers, thus, many shutter and valve position changes over time to be included in the corresponding loop.

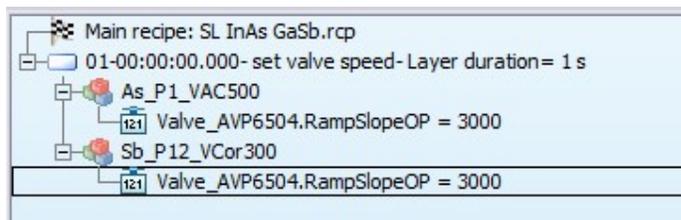
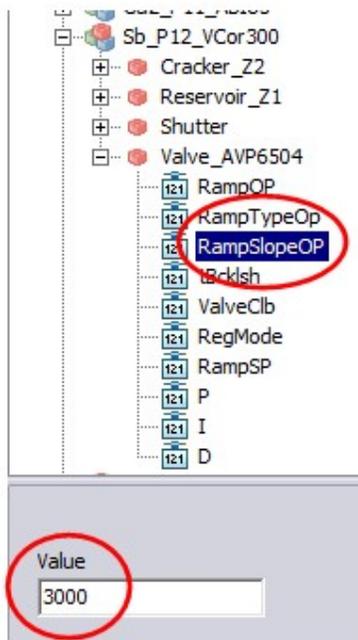
Indeed some special attention has to be brought to the interface quality, and the use of a quite sophisticated series of layers to be integrated in the loop for each iteration.

Some precaution have to be taken as well to minimize stress applied to the mechanical parts while moving the valves.

But the aim of this section is to pin-point some recommendation about valve controller usage while editing the loop. So only the valve tags will be included.

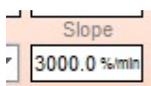
The loop could be edited as follows regarding valve parameter settings, when using RIBER valve controllers model AVP6504:

- Edit the following layer immediately above the loop:



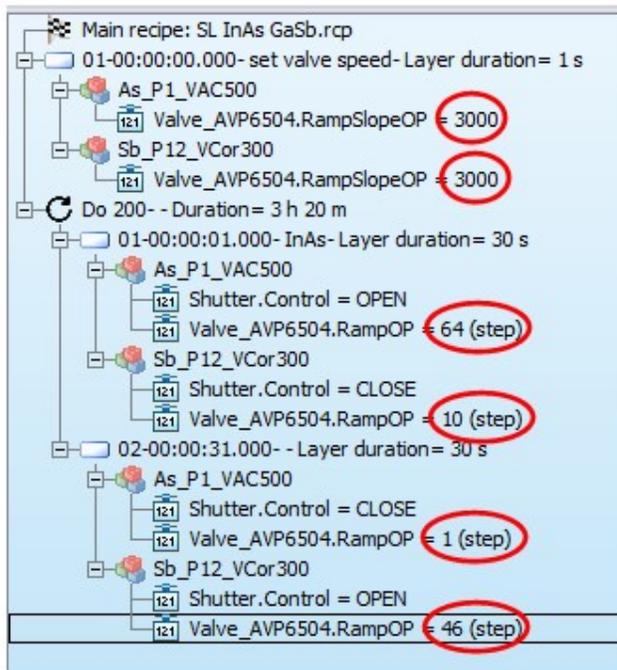
In this layer, we set the speed at which the valve will be moved during execution of the loop next. Set 3000, that is the valves will move at a speed of 3000% per min (so valves would move from 0 to 100% in 2 s for example).

It corresponds to the 'Slope' parameter that is shown on the 'Valve' page of the cell inspector:



The valve positions will be moved as steps in the loop, not linearly over time, to quickly establish/cut GV fluxes. But still one can optimized the speed at which the valves will move, taking into account both the flux requirements and the necessity to minimize mechanical stress. 3000 is a good starting point, that can be optimized later depending on grown structure ex-situ quality checks, such as PL or HRXRD measurements.

- Then the loop next to this initialization layer would look as follow regarding valve tags:



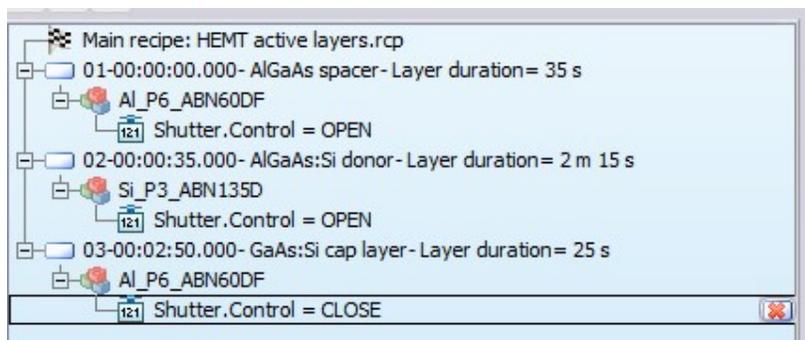
Note that the valve is not fully closed (0%) when growing the layer with the other GV element. One can just use a 'closed' position cutting enough the flux, not to spoil interface quality. Values shown here can be used as starting values prior to optimization.

20.5. Recipe edition: finalization

So far we have remove oxide from the GaAs wafer and grow the GaAs buffer by this recipe. Next we will edit another sub-recipe consisting in all active layers for the HEMT structure, and a last one to end the growth process.

These sub-recipes will be edited in a similar way to the deoxidation GaAs sub-recipe, so only the final edition is shown for both of them:

- Sub-recipe HEMT active layers:



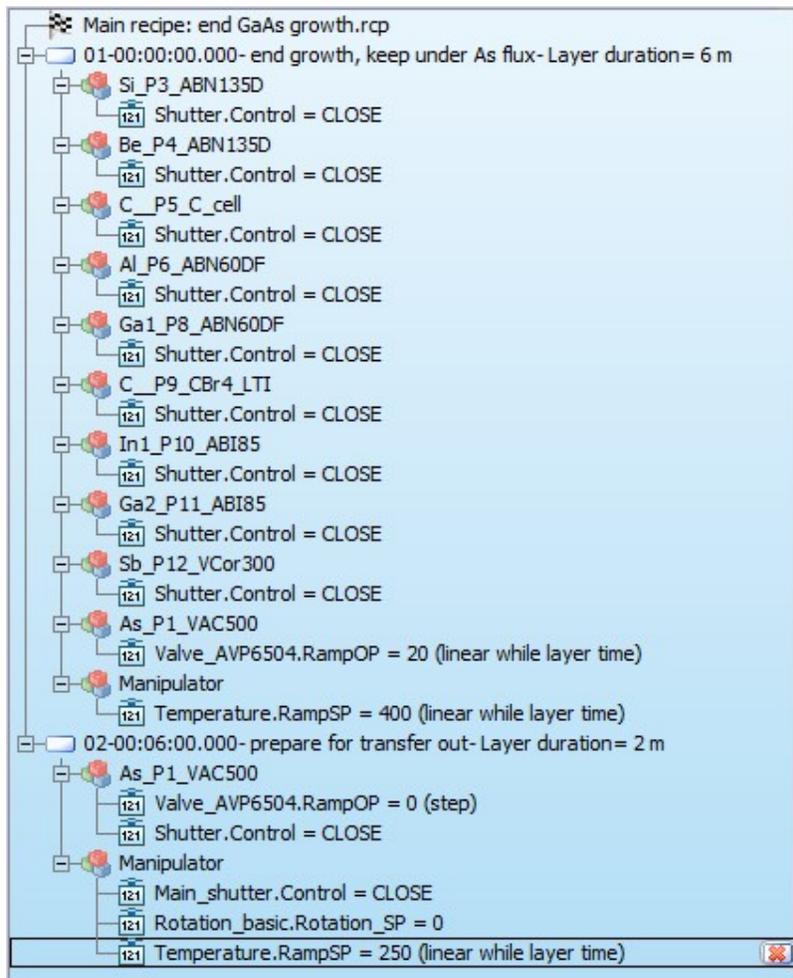
- Sub-recipe *end GaAs growth*:

In order this sub-recipe to be used for any growth for which we keep under As flux meanwhile decrease manipulator/grown wafer temperature, all shutters potentially opened previously for active layers are forced to the closed position in this sub-recipe.

In our simple case indeed only Ga cell position #11 and Si cell shutters needed to be closed.

In the first layer we keep under As flux, so both the As cell shutter and the main shutter are kept opened, and we linearly decrease the manipulator temperature. Meanwhile we also linearly decreased the As valve position since less and less As is needed to protect the surface. And also it reduces the As pressure in the growth chamber, we want as little As as possible before to open the transfer valve to the buffer chamber to take out the platen.

In the second and last layer we prepare for transfer out. Both As and main shutters are closed. The As valve is closed. The manipulator rotation is stopped.



- Now the complete recipe can be edited:

Main recipe: HEMT growth.rcp

- Sub recipe GaAs deoxidation.rcp - Duration= 34 m
- 01-00:34:00.000- buffer growth 10min- Layer duration= 10 m
- Do 25- 25x GaAs/AlGaAs- Duration= 25 m
- 52-01:09:00.000- finish buffer growth 15min- Layer duration= 15 m
- Sub recipe HEMT active layers.rcp - Duration= 3 m 15 s
- Sub recipe end GaAs growth.rcp - Duration= 5 m

Properties Script

General

Title: HEMT growth

Type: conventional HEMT

Author: Ribier

Company: Ribier

Comments:

Creation: 11/06/2015 08:56:23

00:00:00.000

34 m

buffer growth 10min

finish buffer growth 15min

3 m 15 s

5 m

Total= 01:32:15.000 - 52 layers

- Rename sub-recipes to display titles instead of duration in the layer block view:

Layer duration 00:34:00.000

User information **Deoxidation**

Background color Select color Edit sub recipe

Sub recipe: GaAs deoxidation.rcp

- 00-00:00:00.000- Start heating GaAs wafer- Layer duration= 6 m
- 00-00:00:00.000- increase T° under As flux at 25°C/min- Layer du
- 00-00:00:00.000- go to degas T° under As flux- Layer duration=
- 00-00:00:00.000- wafer degas at 6 15°C real T°- Layer duration=
- 00-00:00:00.000- go to growth T°- Layer duration= 4 m

Deoxidation

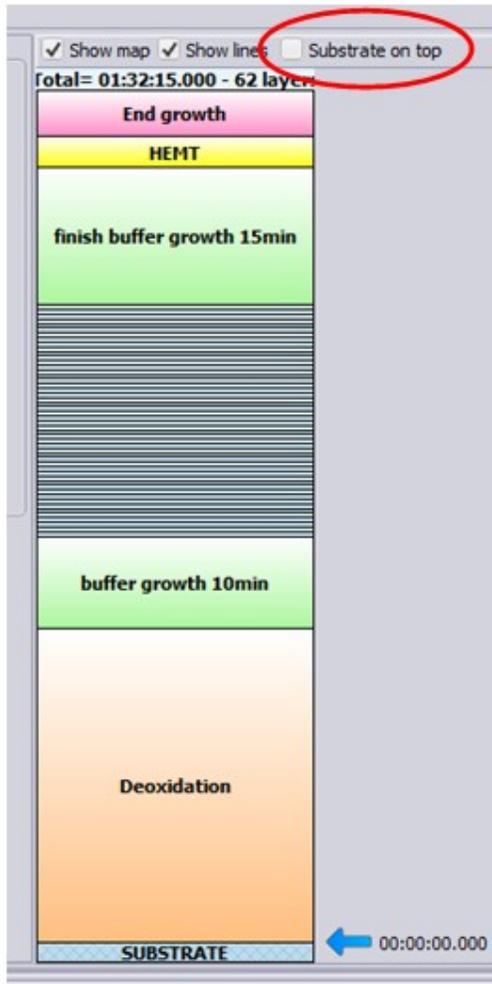
buffer growth 10min

finish buffer growth 15min

HEMT

End growth

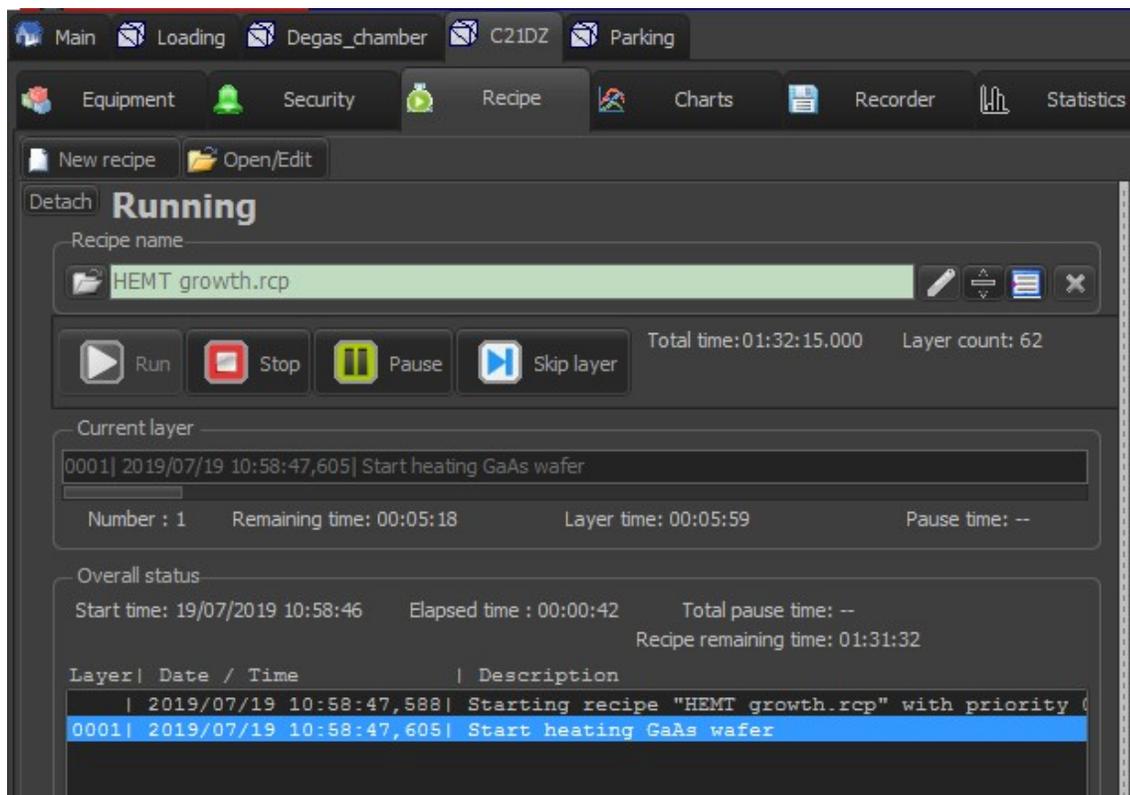
- Note that it is also possible to display the layer stack from bottom by unclicking the 'Substrate on top' tick box:



20.6. In depth information about recipe edition

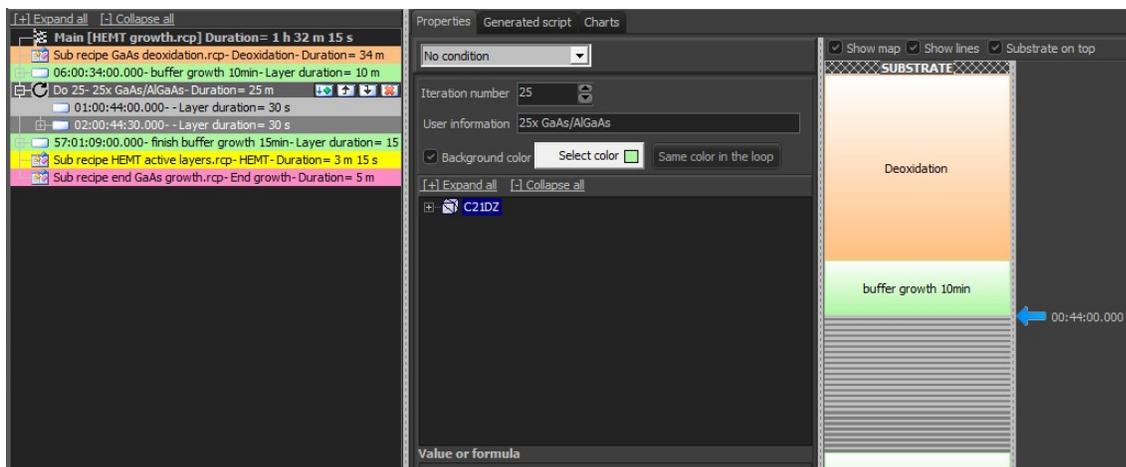
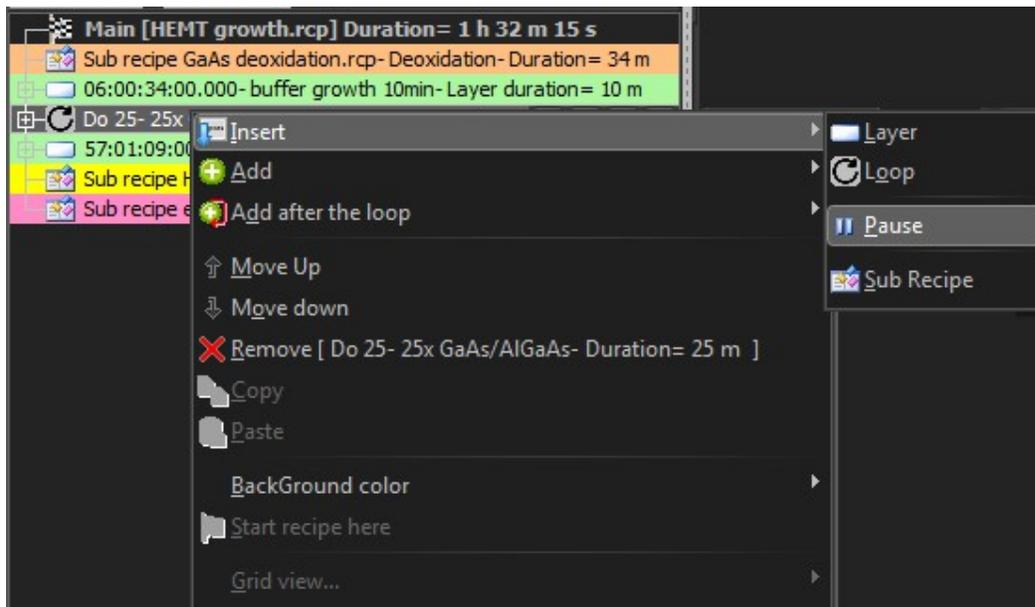
- Pause function:

During its execution the recipe can be paused by clicking on the 'Pause' button:



The recipe will then be frozen in its actual status, all parameters being kept at their current set-point values. In the particular case where a recipe is paused while ramping a set-point, clicking the 'Resume' button will restart the ramp at the exact point it was left.

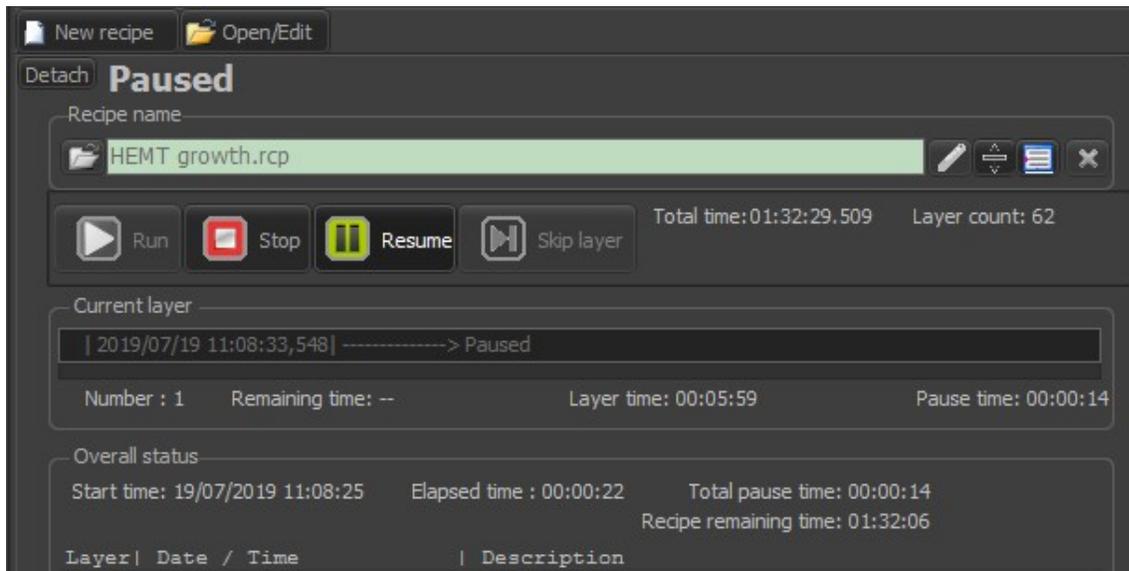
But it is also possible to directly integrate a pause in the recipe. The recipe execution will then be frozen until end-user manually clicks the 'Resume' button to continue recipe execution:



A blue horizontal arrow signals that a special event, here the pause function, is intended during this recipe execution

Then during recipe execution, a red flashing message will inform that recipe execution has been frozen at the pause level.

Click on the 'Resume' button to resume the recipe execution.



Paused time is displayed as shown above.

- How to mix script and recipe?

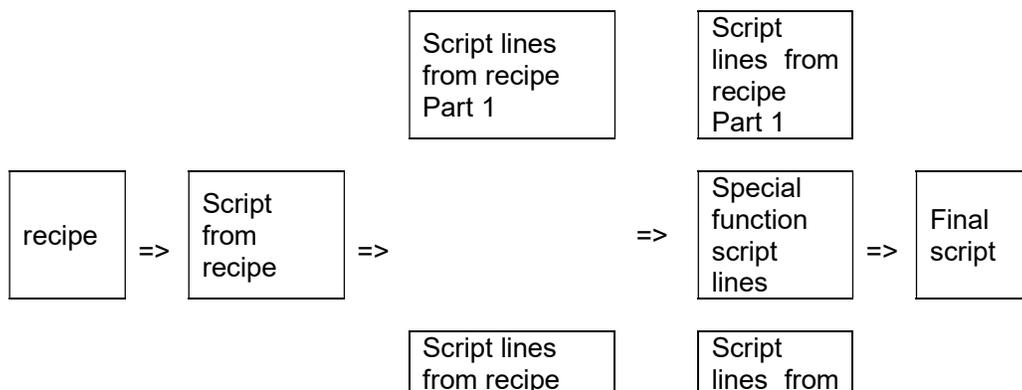
Let's suppose we want to execute the *HEMT growth* recipe as defined previously, but with a special function inserted in-between two layers, a special function that can only be described by the use of a script.

As an example, we want to remove oxide and get ready for growth, but before to grow we want to make sure the temperature is correct by adjusting the manipulator set-point according to the pyrometer reading.

The check will consist in averaging pyrometer reading over 10 measurements taken at 1s interval, and compare the average value to the target value. If within defined temperature range, we proceed with buffer growth. If not, we modify the manipulator set-point according to the difference found, and repeat the check until it is OK.

There is no possibility to directly integrate script lines into a recipe tree. But as an alternative, it is possible to edit a new script that will contain the recipe transformed into its equivalent in script lines, split it into two parts by inserting the series of script lines corresponding to the special function at the proper position.

And doing so run this specific script to execute what we originally plan to do, remove oxide, wait for the proper temperature to be obtained, then start to grow buffer and following steps.

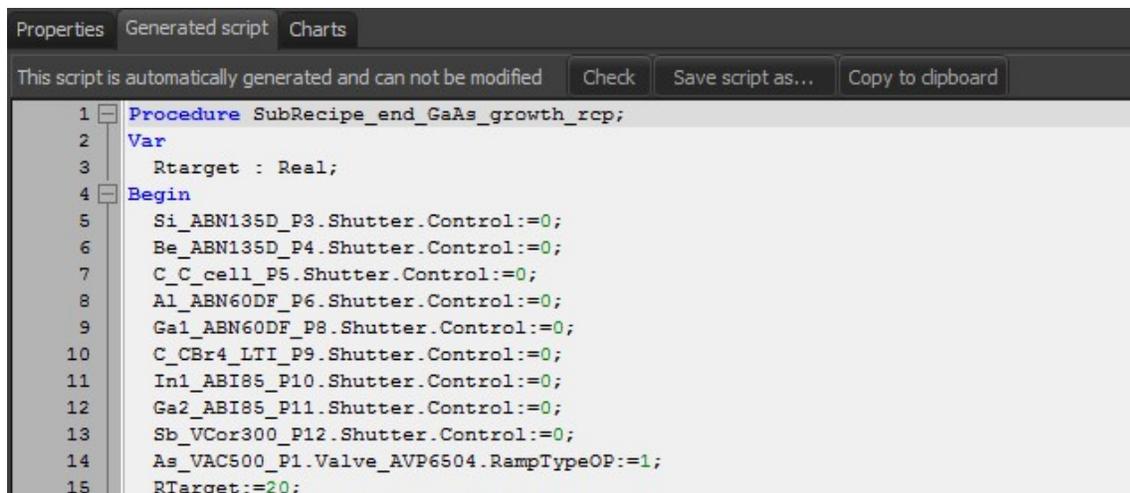
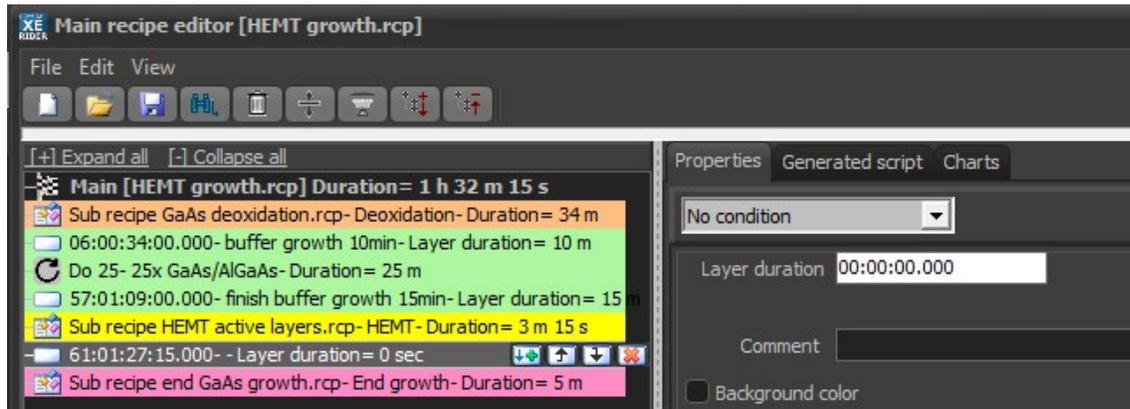


Part 2

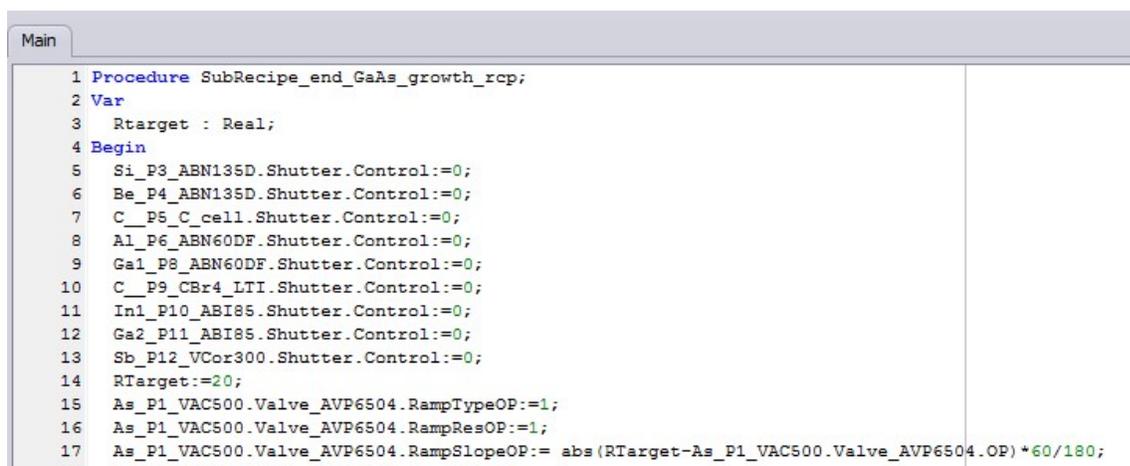
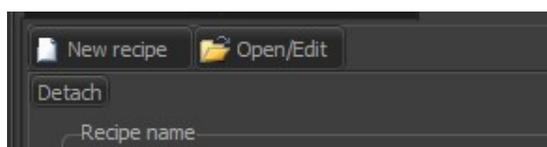
recipe
Part 2

In details:

Open the *HEMT growth* recipe and click on 'Script' to get its equivalent in script lines:



Copy all script lines and paste them into a newly created script:



Pay attention to how is organized the script, there are in this example first a series of 3 'Procedure' that describe the 3 sub-recipes that will be included into the recipe.
For example the middle one is described lines 38-48:

```

36 End;
37 //=====
38 Procedure SubRecipe_HEMT_active_layers_rcp;
39 Var
40   Rtarget : Real;
41 Begin
42   Al_P6_ABN60DF.Shutter.Control:=1;
43   Layer('AlGaAs spacer',35000);
44   Si_P3_ABN135D.Shutter.Control:=1;
45   Layer('AlGaAs:Si donor',135000);
46   Al_P6_ABN60DF.Shutter.Control:=0;
47   Layer('cap layer',25000);
48 End;
49 //=====

```

And then the series of lines corresponding to the execution of the recipe as a whole, starting line 97,

```

90 Manipulator.Temperature.RampSP:=RTarget;
91 Layer('go to growth T',240000);
92 End;
93 //=====
94 Var
95   Rtarget : Real;
96   i : integer;
97 Begin
98   //-----Call sub recipe here
99   SubRecipe_GaAs_deoxidation_rcp;
100 Ga2_P11_ABI85.Shutter.Control:=1;
101 Layer('buffer growth 10min',600000);
102 // 25x GaAs/AlGaAs

```

and ending line 116.

To proceed as needed, insert the necessary script lines at the proper position, in this case in-between lines 99 and 100.

```

97 Begin
98 //-----Call sub recipe here
99   SubRecipe_GaAs_deoxidation_rcp;
100
101
102 // edit special function here
103
104
105 Ga2_P11_ABI85.Shutter.Control:=1;
106 Layer('buffer growth 10min',600000);
107 // 25x GaAs/AlGaAs
108 for i:=1 to 25 do
109   Begin

```

- Configurable recipes:

Crystal XE allows the user to define a series of properties, named in Crystal XE 'global variables', that can be used while editing a recipe or a script.

There are several ways to use this feature, some dedicated to specific cases such as evaporation gun control during recipe execution, and they are described elsewhere.

But it can also be used as described below:

As an example, in the superlattice InAs/GaSb recipe described above, it is possible to refer to global variables instead of specific values for the percentage opening of both the As and the Sb valves.

Indeed in this simple case, it does not change much to modify the % values directly in the recipe editor (based on ex-situ calibrations after growth), or modify two or four global variable values.

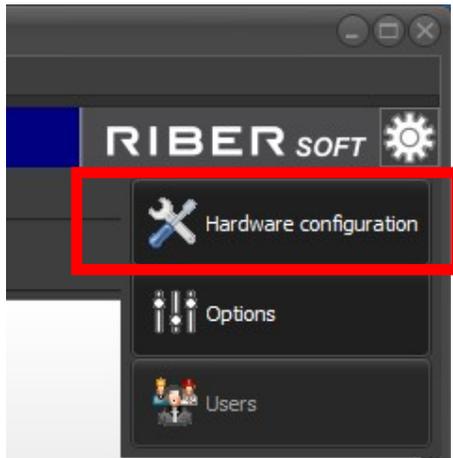
But if the recipe is more complicated, containing hundreds of layers, with usage of a series of different percentage openings for both valves (depending on the target composition in each corresponding layer), it can be quite tricky to make sure all changes have been correctly made in all the layers, since these changes have to be looked for, and made one by one.

Thus in this second case it is much simpler to define a series of global variables, make oneself a table inserted in the 'User view' of the 'Main' page that will contain those global variable values, and modify from this page the values whenever necessary. That is only one time for each of them, instead of hundreds of time maybe if done directly by editing the recipe. So in this case no need to check on the recipe itself: upon execution, it will automatically read all values from this table, and apply them to the corresponding parameters each time they are called in.

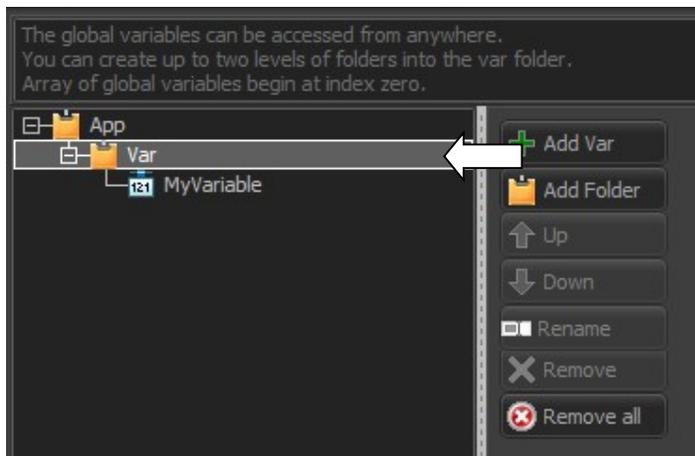
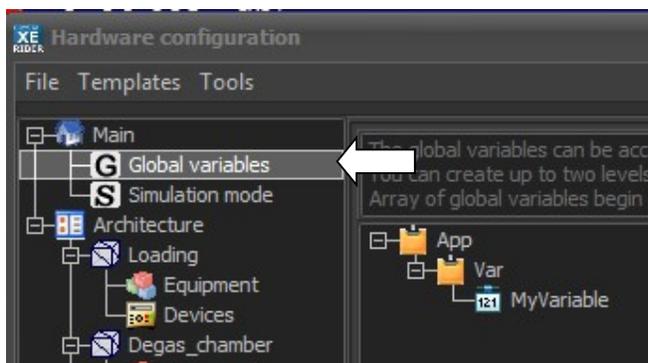
Indeed as an alternative method, it is also possible to make oneself a spreadsheet from which those values can be more easily modify, and import the spreadsheet into the recipe to take into account the new set of values.

In details:

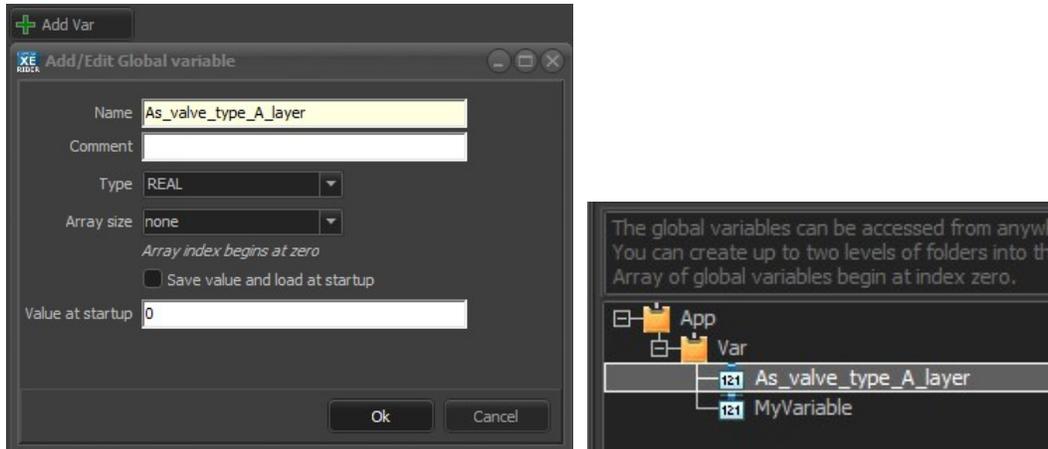
Open the configuration window to create the series of global variables:



Open the 'Global variables' window, and click on 'Add Var' button:

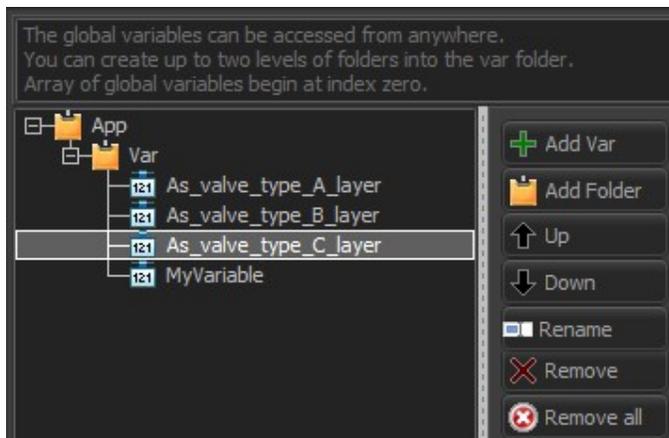


Create the first variable, for example here for the As valve:



Name field reddish if it contains forbidden characters, or spaces
No need yet to input a value
Real type in this case since this is a valve opening percentage

Create all necessary global variables:



More than necessary here for demonstration purpose

Press 'OK' to confirm the selection, and exit the 'Configuration' menu (it is automatically saved).

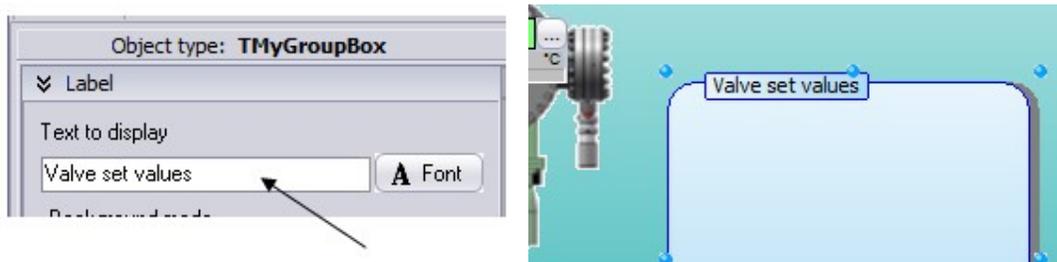
Go to the 'User View' page of the 'Main' display set and create a table containing these global variables.

Starting from scratch:

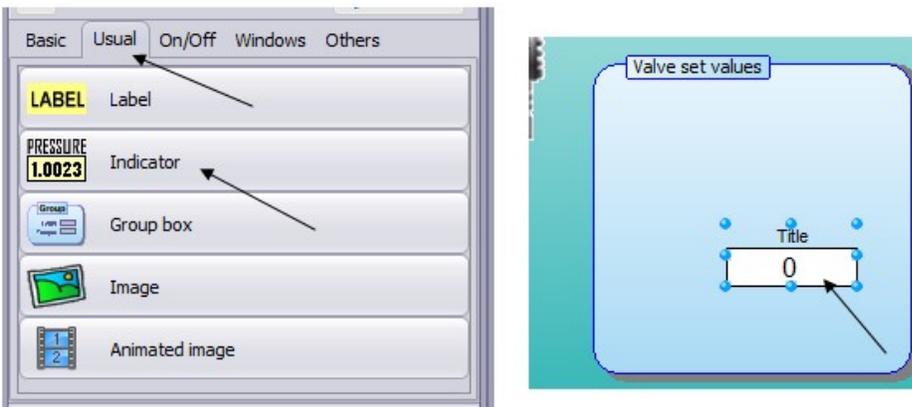
Edit the view form, and drag and place a new 'Combo box':



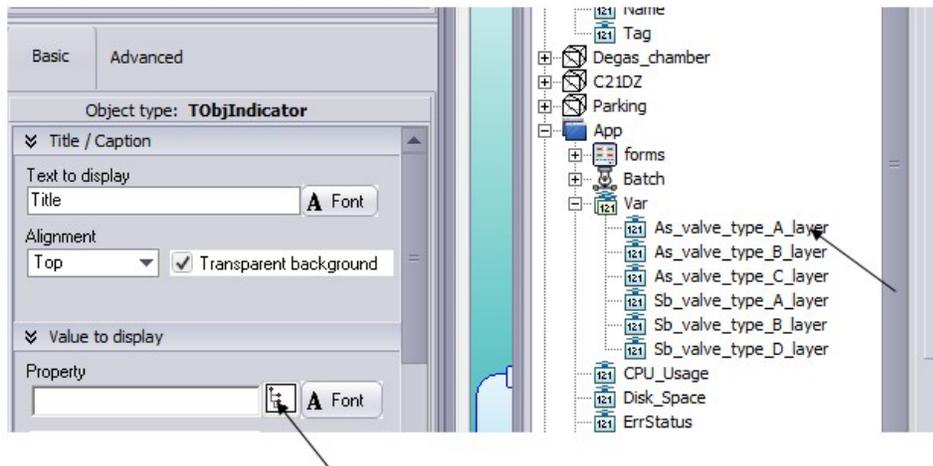
Change title to your liking:



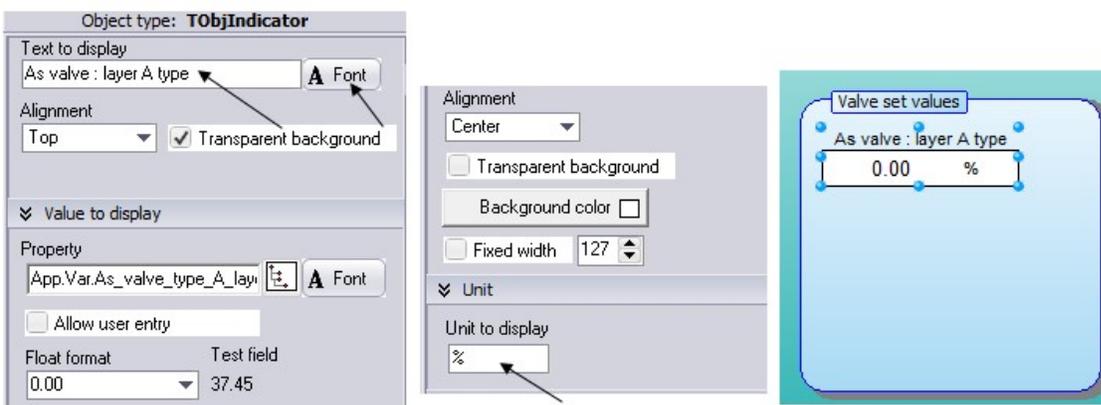
Unselect the Combo box, and drag and place a first 'Indicator':



Set the proper global variable as follows:

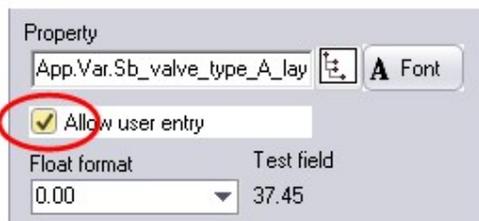


Add title, modify font, add unit:

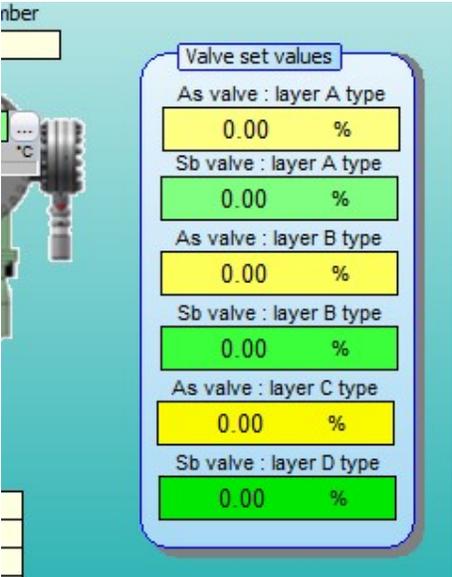


Add blanks after unit to make it to appear more centered.

Make sure to tick the 'Allow user entry' box to be able to modify the values from this table:



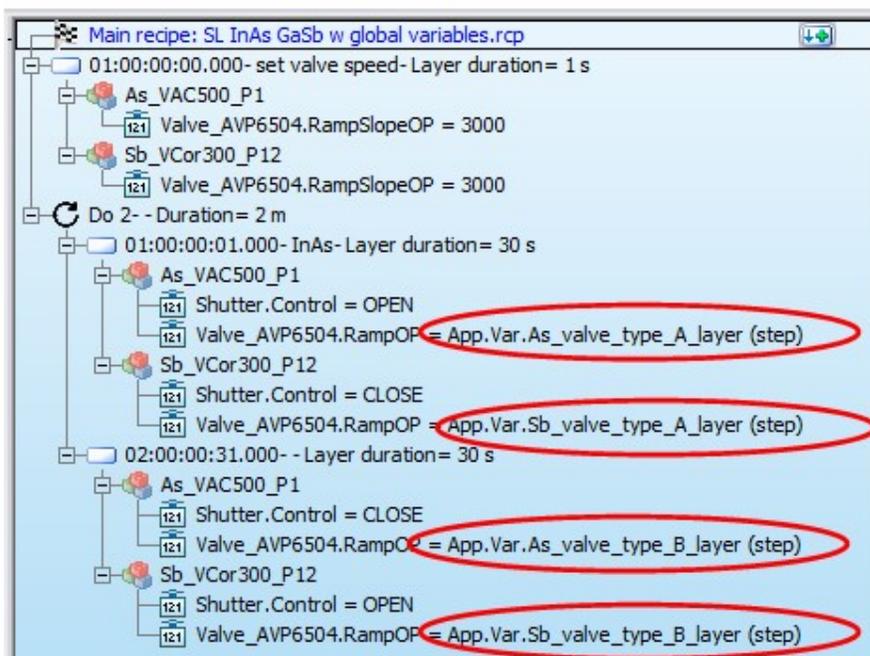
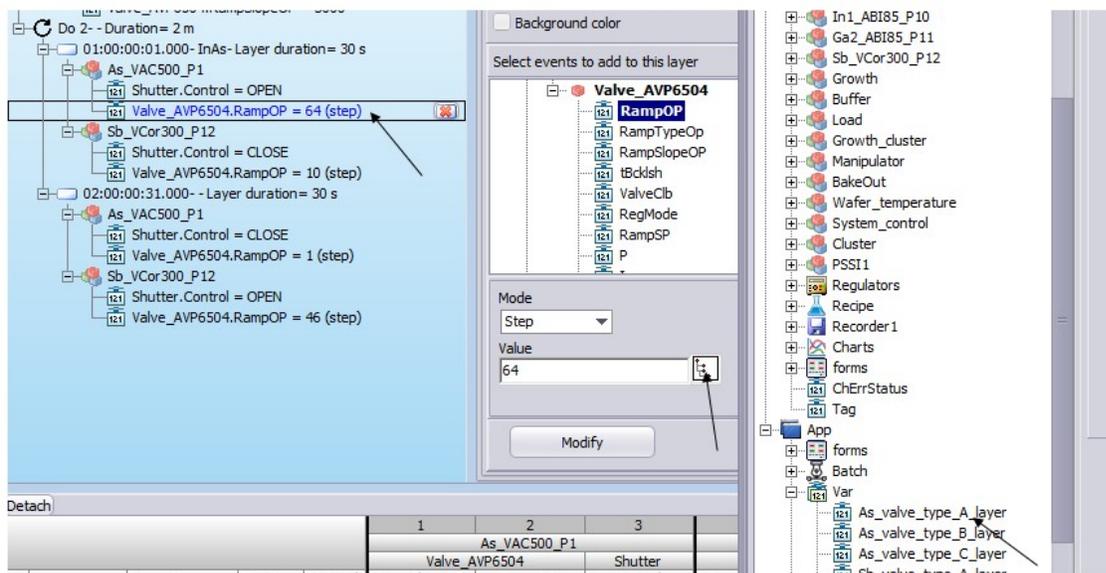
Complete with all necessary global variables, and adjust window size, background color to your liking:



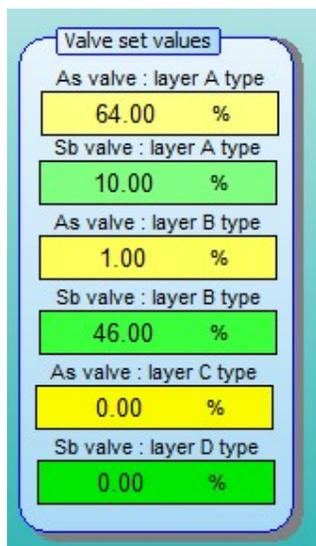
Copy/paste the first 'Indicator' to create the other ones.

Exit and save the new 'User View'.

Modify then the 'SL InAs GaSb' recipe shown as an example in above section:



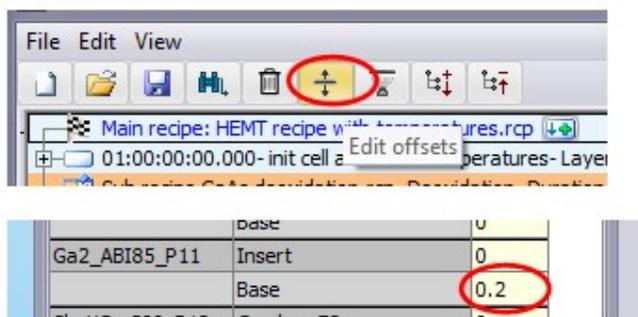
Then before recipe execution, fill in according to last calibration runs the global variable field in the 'User view':



- Offset editor:

Due to material depletion into the crucible, with standard effusion cells it will be necessary to modify/increase the cell temperature to keep the same outcoming flux.

This can be done directly from the recipe editor by opening the offset sub-menu and entering values, or downloading an existin offset file.



This can also be done directly from the 'Recipe' page:



- Growth rate editor:

A dedicated function permits to convert durations into thicknesses, and reversely thicknesses into durations, to make sure the layers will be as originally planned.

In this version, Crystal XE calculates duration or thickness based on following parameters:

- a valid growth rate has been defined,
- the cell shutter is open.

Note that this features is meant to help in simple cases. For complicated structures that would imply growth rate changes, it is possible to import an accordingly defined spread sheet.

Also:

The main shutter status - open, closed or not specified - is not taken into account, the calculation is done independantly.

If the temperature is changed while the shutter is open, the calculation will still be done according to the set growth rate.

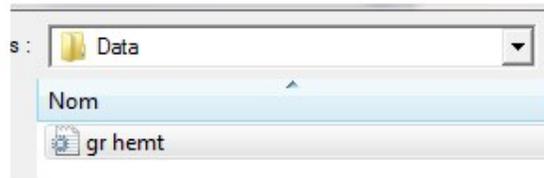
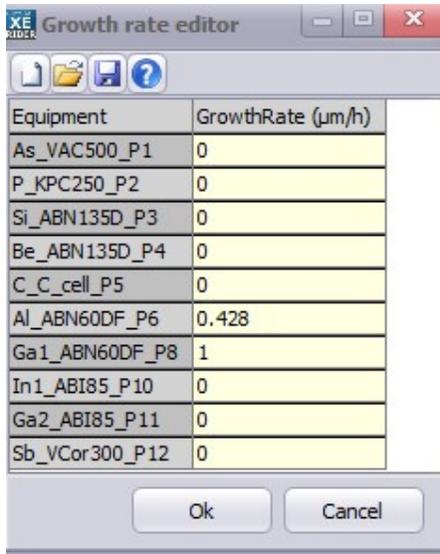
To define growth rate and thickness units, go to the 'General' page of the 'Options' window.



Indeed this choice can be modified anytime. Upon opening the recipe window, the values will be modified accordingly.

Open the recipe and open the 'Growth rate editor' window to set the growth rates to be taken into account in the calculations:





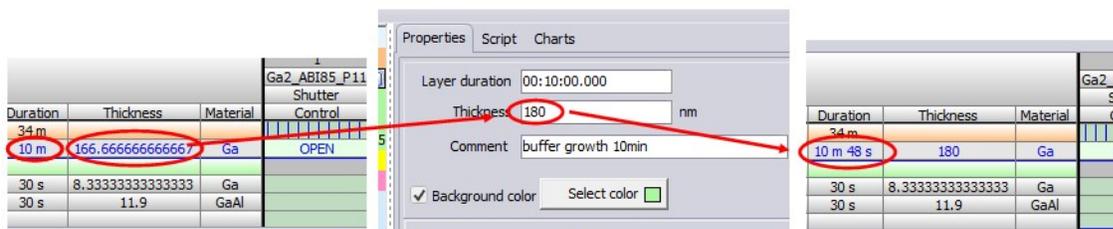
Either manually enter the values or download a previously saved file.

Upon entering nonzero values, the column 'Thickness' of the table view and the 'thickness' box of the layer will be filled in accordingly. If no nonzero growth rate can be found for one layer, the column/box will remain empty.

Detach						1	2
						Ga2_ABI85_P11	Al_ABN60DF_P6
#	Layer	Description	Duration	Thickness	Material	Shutter Control	Shutter Control
	Sub recipe GaAs deoxidation.rcp	Deoxidation	34 m				
06	00:34:00.000	buffer growth 10min	10 m	166.666666666667	Ga	OPEN	
	DO 25	25x GaAs/AlGaAs					
01	00:44:00.000		30 s	8.33333333333333	Ga		CLOSE
02	00:44:30.000		30 s	11.9	GaAl		OPEN
	END LOOP						

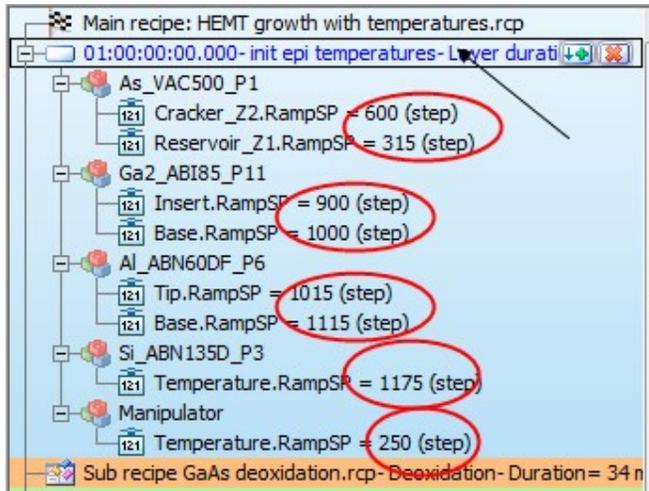
Note that in case a growth rate is changed, this is the thickness value that will be changed, the duration value remains unchanged.

Indeed selecting a nonzero value from the Table view open the corresponding tag that can then be directly modified:



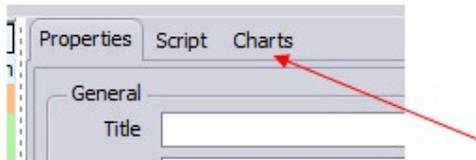
20.7. Recipe edition: additional viewings

For a better understanding, an init line setting growth temperatures has been inserted as first layer in the 'HEMT' growth' recipe, renamed 'HEMT recipe with temperatures'. It will permit to detail the two functions presented in this section.

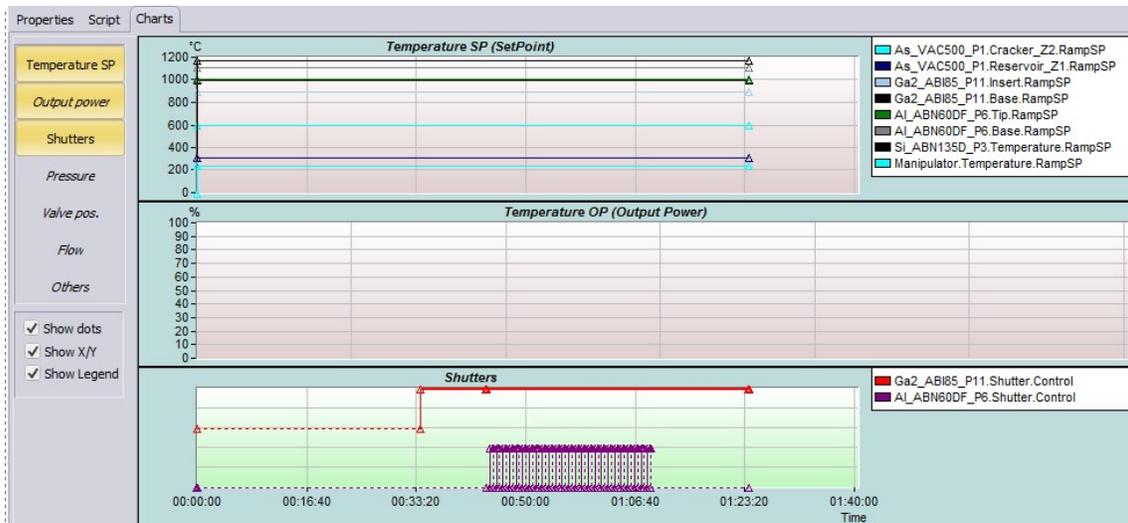


- Graph display:

Click on the 'Charts' page



Graph display and related features are identical to the 'Charts' page of the corresponding chamber in the configuration. It is not possible to have the measured values to be plotted on their graphs.



- Table view:

The recipe is also represented as a table from where one can have an overview of all parameters with a defined setting in one of the layers, except for subrecipes that are not taken into account.

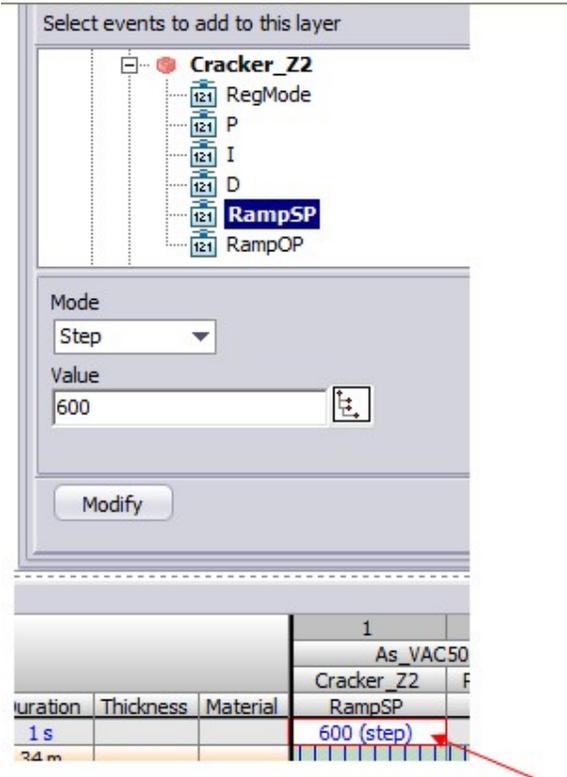
This table is interactive, you can modify all parameters:

- Durations.
- Shutter positions.
- Temperature and valve set-points.

Detach					
#	Layer	Description	Duration	Thickness	Material
01	00:00:00.000	init epi temperatures	1 s		
	Sub recipe GaAs deoxidation.rcp	Deoxidation	34 m		
07	00:34:01.000	buffer growth 10min	10 m		Ga
	DO 25	25x GaAs/AlGaAs			
01	00:44:01.000		30 s		Ga
02	00:44:31.000		30 s		GaAl
	END LOOP				
58	01:09:01.000	finish buffer growth 15min	15 m		Ga
	Sub recipe HEMT active layers.rcp	HEMT	3 m 15 s		
	Sub recipe end GaAs growth.rcp	End growth	5 m		

1	2	3	4	5	6	7	8	9	10
Ga2_ABI85_P11			Al_ABNG0DF_P6			As_VAC500_P1		Si_ABN135D_P3	Manipulator
Shutter	Insert	Base	Shutter	Tip	Base	Cracker_Z2	Reservoir_Z1	Temperature	Temperature
Control	RampSP	RampSP	Control	RampSP	RampSP	RampSP	RampSP	RampSP	RampSP
	900 (step)	1000 (step)		1015 (step)	1115 (step)	600 (step)	315 (step)	1175 (step)	250 (step)
OPEN									
			CLOSE						
			OPEN						
			CLOSE						

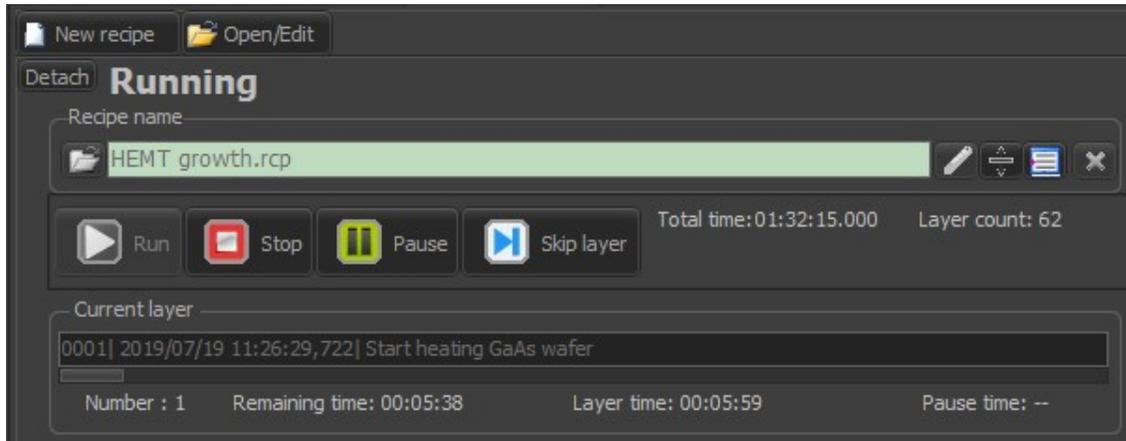
Clicking on one parameter directly opens the tree view to allow modification. Select the 'Properties' page and click on the parameter to be modified:



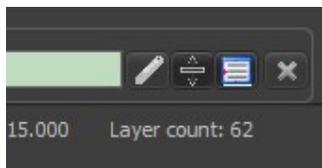
Note that it is also possible to add a parameter by simply clicking on an empty box of this table.

8.5 Other features

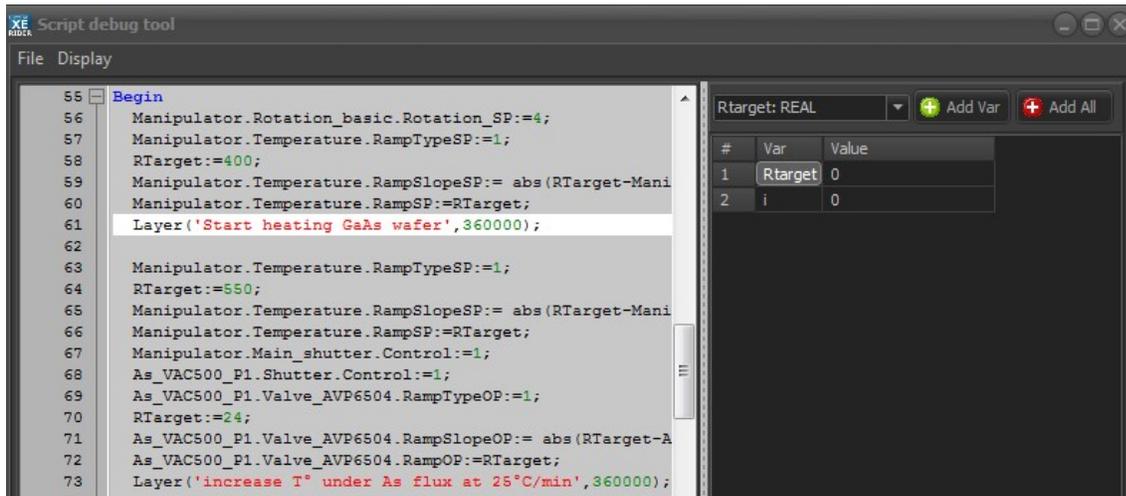
- How to follow recipe execution progress?



In addition to the information contained in the integrated log file, it is possible to exactly follow the recipe execution by opening the 'Script debug tool' window:

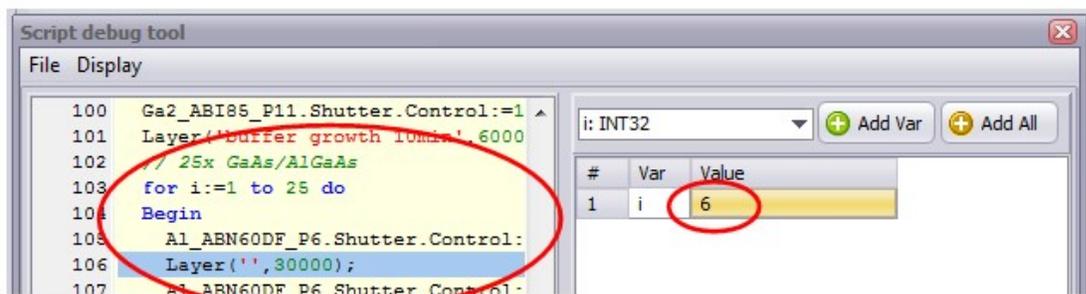
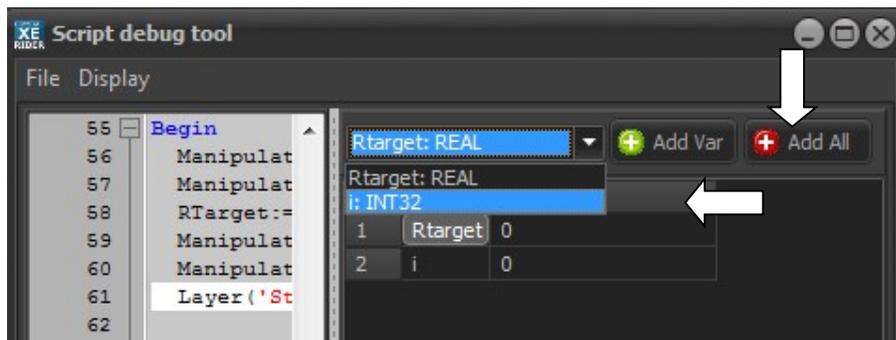
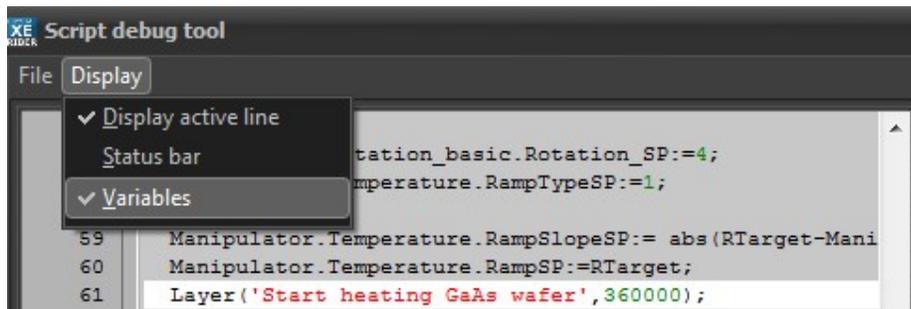


The recipe line currently executed is highlighted in blue:



This feature can be very useful for example to exactly know which iteration of a loop is currently executed, by displaying the parameter of interest, in this case the iteration number.

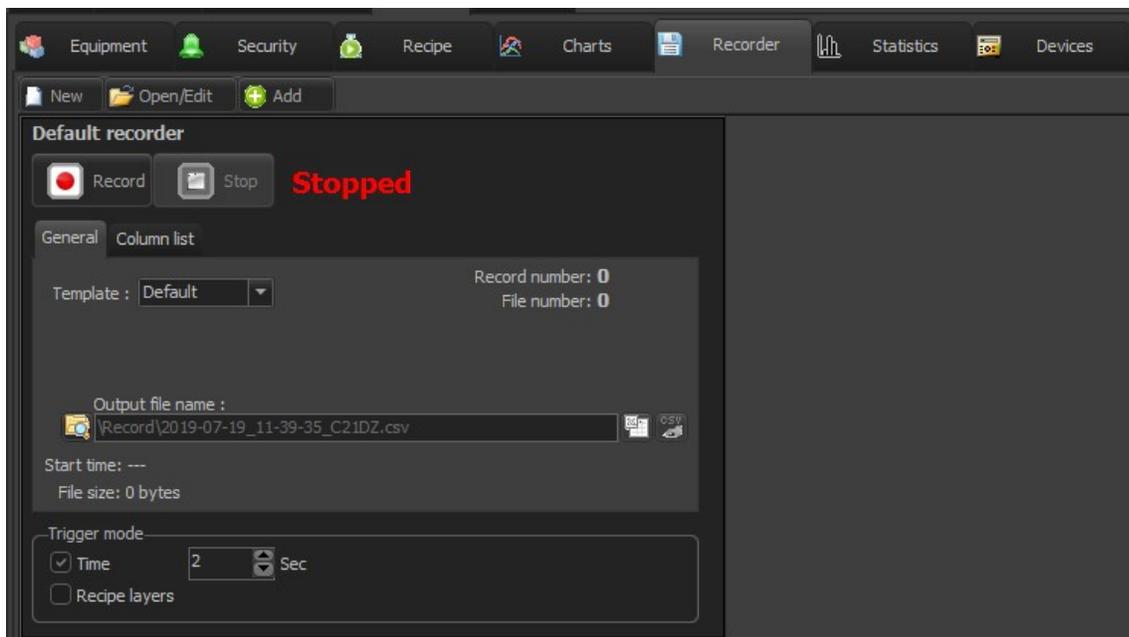
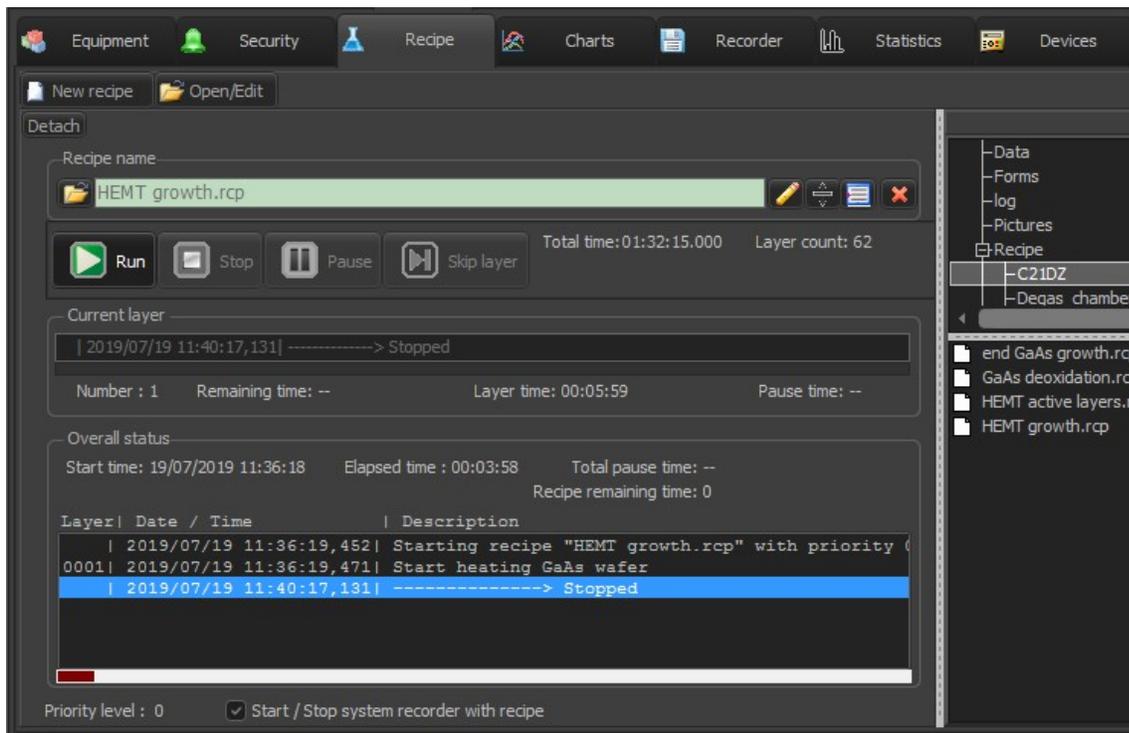
Open the 'Display' menu to select which variables will be displayed:



So the sixth iteration of the loop is currently being executed in this example.

- How to retrieve data from recorded files?

Unless unticked in the 'Recipe' page, a file containing all parameters defined in the record template in current use will be saved:

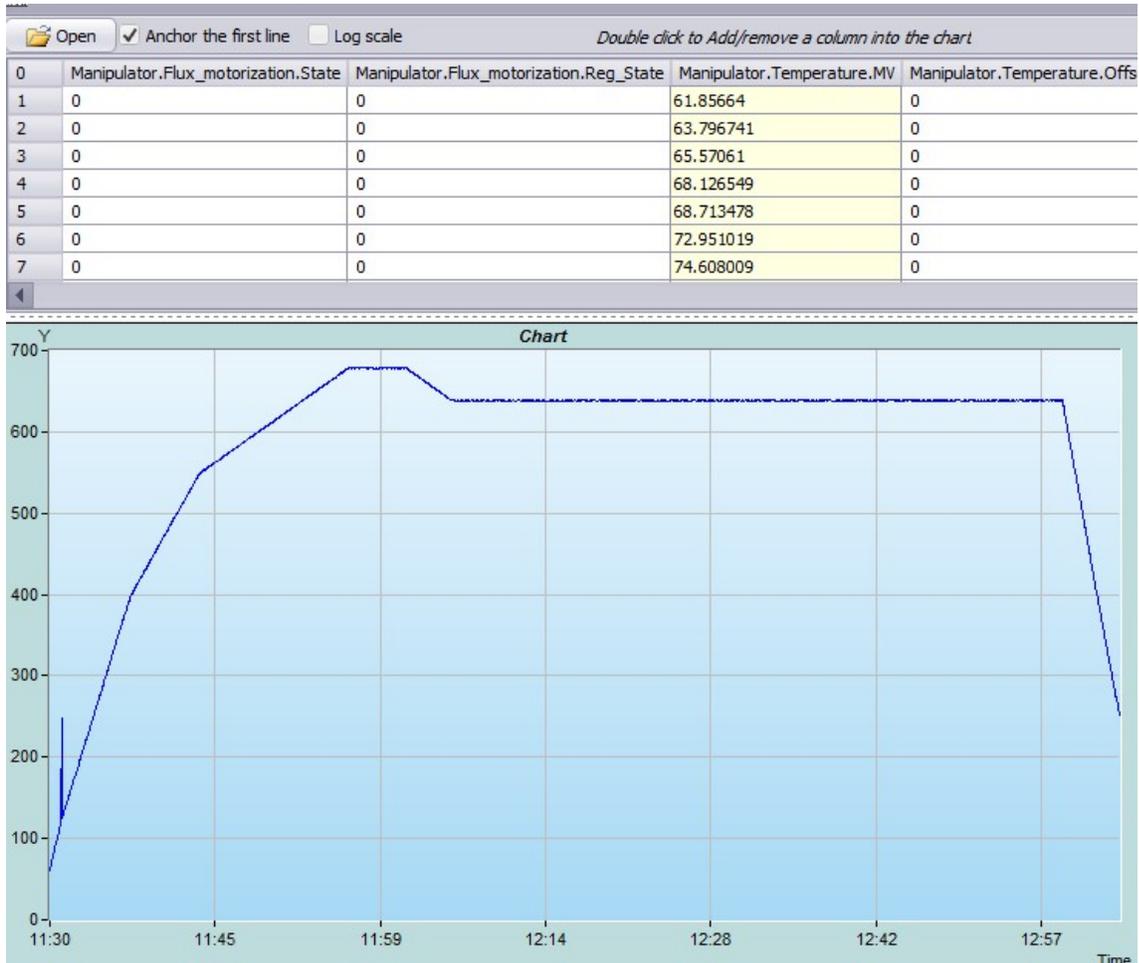


If the template 'Default' is selected, all parameters will be saved. Create your own template to fine-tune the selection.

After recipe completion, double-click on the recipe csv file to pop-up a window displaying all data:



Double click on any tag to display it as a graph view:



Similarly to the 'Charts' page, one can display several parameters on the same graph of the same type, zoom, activate the 'balloon' function to display values, save the data displayed in one graph as a new csv file.

21. License

CRYSTAL XE
 Copyright (C) 2016-2022 RIBER SA.
 All Rights Reserved

NOTICE TO USER:

THIS IS A CONTRACT BETWEEN YOU AND RIBER S.A.
BY INSTALLING THE SOFTWARE (AS DEFINED BELOW), YOU ACCEPT ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT.

This RIBER End User Licence Agreement accompanies the CRYSTAL product and related explanatory materials ("Software"). The term "Software" shall also include any upgrades, modified versions or updates of the Software licensed to you by RIBER S.A.

Please read this Agreement carefully. At the end, you will be asked to accept this Agreement and continue to install or, if you do not wish to accept this Agreement, to decline this Agreement, in which case you will not be able to use the Software.

If you cannot understand this licence or do not think this licence applies to you then please contact the RIBER S.A. Customer Service Department.

This is a licence agreement and not an agreement for sale. RIBER S.A. continues to own the copy of the Software and the physical media contained in this package and any other copy that you are authorised to make pursuant to this Agreement.

Unless you have a different licence agreement signed by RIBER SA. your use of this software indicates your acceptance of this licence agreement and warranty.

A. Use of the Software

One registered copy of CRYSTAL may either be used by a single person who uses the software personally on one or more computers, or installed on a single workstation used nonsimultaneously by multiple people, but not both.

Unregistered use of CRYSTAL is in violation of international copyright laws.

B. Copyright and Trademark Rights

The Software is the intellectual property of RIBER S.A. and is protected by French copyright law, international treaty provisions, and applicable laws of the country in which it is being used.

The structure, organisation and code of the Software are the valuable trade secrets and confidential information of RIBER.

You may use the trademarks only to identify printed output produced by the Software, in accordance with accepted trademark practice, including identification of trademark owner's name.

Such use of any trademark does not give you any rights of ownership in that trademark. Except as stated above, this Agreement does not grant you any intellectual property rights in the Software.

C. Restrictions

The Software is licensed and distributed by RIBER S.A. for working with RIBER Molecular Beam Epitaxy (MBE) systems.

Because of its hardware dependency (extension cards, controllers, regulators, automates), the software is intended to run conforming with the hardware compatibility guide.

Please, contact RIBER S.A. Customer Service Department to get this compatibility guide.

D. Disclaimer of Warranty

The Software is being delivered to you "AS IS" and RIBER makes no warranty as to its use of performance.

RIBER DOES NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THE SOFTWARE OR DOCUMENTATION. IN NO EVENT WILL RIBER BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL, INDIRECT OR SPECIAL DAMAGES, INCLUDING ANY LOST PROFITS OR LOST SAVINGS, EVEN IF A RIBER REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY THIRD PARTY.

In particular, there is no warranty for the virus scanning (CRYSTAL does not scan for viruses, use external programs to perform this function).

Some states or jurisdictions do not allow the exclusion or limitation of incidental, consequential or special damages, or the exclusion of implied warranties or limitations on how long an implied warranty may last, so the above limitations may not apply to you.

Good data processing procedure dictates that any program be thoroughly tested with non-critical data before relying on it.

The user must assume the entire risk of using the program.
ANY LIABILITY OF THE SELLER WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT.

E. Distribution

You are specifically prohibited from charging, or requesting donations, for any such copies, however made; and from distributing the software and/or documentation with other products (commercial or otherwise) without prior written permission.

CONTACT :

To contact us : RIBER
31, rue Casimir Périer
BP 70083
92873 BEZONS CEDEX
FRANCE

Mail: customerservice@riber.fr

Our web site : <http://www.riber.com> and <http://www.crystalxe.com>