Procedure

Recipes generation, installation and activation

Abstract

This procedure explains how to generate recipes for a UNICOS CPC6 application, and how to install, import and activate the recipes from WinCC OA.

DOCUMENT PREPARED BY: PRIETO BARREIRO, Ivan [EN/ICE CERN]
DOCUMENT CHECKED BY: [Checkers]
DOCUMENT APPROVED BY: [Approvers]
<table>
<thead>
<tr>
<th>REV. NO.</th>
<th>DATE</th>
<th>PAGES</th>
<th>DESCRIPTIONS OF THE CHANGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2012-05-23</td>
<td>17</td>
<td>First version (Ivan Prieto Barreiro)</td>
</tr>
<tr>
<td>1.1</td>
<td>2013-12-11</td>
<td>29</td>
<td>Adding documentation for dynamic recipe class creation (Ivan Prieto Barreiro)</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

1. INTRODUCTION ..................................................................................................... 5
2. REQUIREMENTS ..................................................................................................... 5
3. PROCEDURE .......................................................................................................... 5
  3.1 UAB APPLICATION GENERATION .............................................................. 5
  3.2 UAB EXTENDED CONFIGURATIONS PLUG-IN ........................................ 6
    3.2.1 OPEN THE UAB APPLICATION ................................................................. 6
    3.2.2 FILL THE RECIPE CLASS AND RECIPE INSTANCE FILES .................. 7
      3.2.2.1 RECIPE CLASS DEFINITIONS .............................................................. 7
      3.2.2.2 RECIPE INSTANCE DEFINITIONS ....................................................... 9
      3.2.2.3 RUN THE PLUG-IN GENERATION ................................................... 10
  3.3 WinCC OA ........................................................................................................ 10
    3.3.1 CREATION OF A WINCC OA – CPC 6 APPLICATION .......................... 10
    3.3.2 IMPORT THE RECIPES ............................................................................ 10
    3.3.3 USE THE RECIPE INSTANCES ............................................................... 12
      3.3.3.1 BROWSE THE IMPORTED RECIPES .................................................. 12
      3.3.3.2 ACTIVATE A RECIPE INSTANCE ....................................................... 14
      3.3.3.3 EDIT A RECIPE INSTANCE ................................................................. 14
      3.3.3.4 CREATE NEW RECIPE INSTANCES ................................................... 14
      3.3.3.5 DELETE A RECIPE INSTANCE ............................................................ 15
      3.3.3.6 INITIAL RECIPES ............................................................................... 15
      3.3.3.7 LAST ACTIVATED RECIPES ............................................................... 16
      3.3.3.8 SAVE RECIPE VALUES AS INITIAL .................................................. 16
    3.3.4 DYNAMIC RECIPE CLASSES ................................................................. 16
      3.3.4.1 RECIPE CLASS CREATION ............................................................... 17
      3.3.4.2 CREATE NEW RECIPE CLASSES ...................................................... 18
      3.3.4.3 ADD NEW DEVICES TO A RECIPE CLASS ...................................... 18
      3.3.4.4 REMOVE DEVICES FROM A RECIPE CLASS .................................. 19
      3.3.4.5 EDIT THE DEVICE ELEMENTS OF A RECIPE CLASS ...................... 20
      3.3.4.6 REMOVE A RECIPE CLASS ............................................................... 21
      3.3.4.7 DUPLICATE AN EXISTING RECIPE CLASS ..................................... 21
      3.3.4.8 EDIT THE RECIPE CLASS AND RECIPE INSTANCES PRIVILEGES ... 21
      3.3.4.9 CREATE A NEW RECIPE INSTANCE ................................................ 23
  4. TROUBLESHOOTING ......................................................................................... 23
    4.1 RECIPE ACTIVATION FAILED ................................................................. 23
      4.1.1 SCHNEIDER ......................................................................................... 23
      4.1.2 SIEMENS .......................................................................................... 24
  5. SUPPORT ........................................................................................................... 25
  6. REFERENCES ..................................................................................................... 25
APPENDIX A: RECIPE TYPE DEFINITION ......................................................... 26
## TABLE OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Open XML files from Ms. Office Excel.</td>
<td>5</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Enable recipes.</td>
<td>6</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Execute the UAB Extended Configurations plug-in.</td>
<td>6</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Main window of the UAB Extended Configurations plug-in.</td>
<td>7</td>
</tr>
<tr>
<td>Figure 5</td>
<td>User inputs folder for the UAB Extended Configurations plug-in.</td>
<td>7</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Recipe class definitions example.</td>
<td>8</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Recipe instance definitions example.</td>
<td>9</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Import Database.</td>
<td>11</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Check Ok.</td>
<td>11</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Front End Diagnostic.</td>
<td>12</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Browsing the imported recipes.</td>
<td>13</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Recipe instance panel.</td>
<td>13</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Recipe activation log.</td>
<td>14</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Create a new recipe instance.</td>
<td>14</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Initial recipes.</td>
<td>15</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Last activated recipes.</td>
<td>16</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Recipe class menu entry.</td>
<td>17</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Recipe class panel.</td>
<td>17</td>
</tr>
<tr>
<td>Figure 19</td>
<td>Create a recipe class.</td>
<td>18</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Recipe front-end creation.</td>
<td>18</td>
</tr>
<tr>
<td>Figure 21</td>
<td>Device selection for a recipe class.</td>
<td>19</td>
</tr>
<tr>
<td>Figure 22</td>
<td>Remove devices from a recipe class.</td>
<td>20</td>
</tr>
<tr>
<td>Figure 23</td>
<td>Edit device elements in a recipe class.</td>
<td>21</td>
</tr>
<tr>
<td>Figure 24</td>
<td>Privileges of the recipe class and its recipe instances.</td>
<td>22</td>
</tr>
<tr>
<td>Figure 25</td>
<td>Edit the recipe instance privileges for the operator.</td>
<td>22</td>
</tr>
<tr>
<td>Figure 26</td>
<td>Recipe activation timeout for Schneider.</td>
<td>23</td>
</tr>
<tr>
<td>Figure 27</td>
<td>Recipes source file in Step-7.</td>
<td>24</td>
</tr>
<tr>
<td>Figure 28</td>
<td>Recipe activation timeout for SIEMENS.</td>
<td>24</td>
</tr>
<tr>
<td>Figure 29</td>
<td>Recipe Type definition example.</td>
<td>26</td>
</tr>
<tr>
<td>Figure 30</td>
<td>Recipe type, class and instance relationship.</td>
<td>26</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

The goal of this document is to describe how to generate recipes for a UNICOS CPC6 application and how to install, import and activate the recipes from WinCC OA.

The necessary steps for using the recipes are summarized as:

1. UAB Application generation:
   1.1. Generate a UNICOS CPC6 application enabling the recipes.

2. UAB Extended Configurations plug-in:
   2.1. Open an existing UNICOS CPC6 application with the plug-in.
   2.2. Fill the recipe class and recipe instances files.
   2.3. Run the plug-in generation.

3. WinCC OA
   3.1. Create a WinCC OA – CPC 6 application.
   3.2. Import the recipes.
   3.3. Use the recipes (activate, edit, duplicate ...).

2. REQUIREMENTS

The recommended tool to view and edit the files used by the UAB Extended Configurations plug-in is **MS Excel**:

1. Right click on the selected file, Open with -> Microsoft Office Excel
2. In the pop-up menu select open "As an XML table".

![Open XML dialog](image.png)

Figure 1 - Open XML files from Ms. Office Excel.

3. PROCEDURE

3.1 UAB APPLICATION GENERATION

The first step of the procedure is to create and generate a UNICOS CPC6 application using the CPC Wizard as described in the following documents:

1. Schneider: [https://edms.cern.ch/file/1228441/1.5.0/Procedure_Sch-UCPC_Application.pdf](https://edms.cern.ch/file/1228441/1.5.0/Procedure_Sch-UCPC_Application.pdf)
2. SIEMENS: [https://edms.cern.ch/file/1228441/1.5.0/Procedure_S7-UCPC_Application.pdf](https://edms.cern.ch/file/1228441/1.5.0/Procedure_S7-UCPC_Application.pdf)

In the PLC specifications panel, enable the recipes and fill the recipe parameters as shown in the Figure 2 - Enable recipes.
• **Enable recipes**: this checkbox is used to enable/disable the use of the recipes in the PLC.

• **Max. number of recipe values**: is the maximum number of elements in a recipe.

• **Recipe Buffer Starting Address (Schneider ONLY)**: is the starting address of the recipe buffers in the PLC.

• **Activation Timeout (s)**: is the recipe activation timeout in seconds. If the recipe activation time is longer than the specified value, the recipe activation will be rejected by the PLC.

### 3.2 UAB EXTENDED CONFIGURATIONS PLUG-IN

#### 3.2.1 OPEN THE UAB APPLICATION

The next step is to open the UAB application created in the step 3.1 with the UAB Extended Configurations plug-in:

https://j2eeps.cern.ch/wikis/display/EN/UAB+Tools:

1. Unzip the plug-in and double-click on the file bin/ExtConfigGenerator.bat

Figure 2 - Enable recipes.

Figure 3 - Execute the UAB Extended Configurations plug-in.
2. Open the UnicosApplication.xml file located in the UAB project folder. From the menu, select Document-> Load or click on the "Load document" icon and select the UnicosApplication.xml file in the UAB project folder.

3.2.2 FILL THE RECIPE CLASS AND RECIPE INSTANCE FILES

Once the UnicosApplication.xml file is opened by the UAB Extended Configurations plug-in, a new folder will be created with the necessary resources to generate the extended configurations.

3.2.2.1 RECIPE CLASS DEFINITIONS

The recipe class definitions file is located in: Resources/ExtendedConfigGenerator/user-inputs/RcpClassDefinitions.xml.

A recipe class definition describes the list of devices and device elements that will be used in all the recipes (recipe instances) of the class.
The recipe class definitions file contains the following mandatory fields:

- **className**: is the name of the recipe class. It must be unique within the file.
- **typeName**: is the recipe type name. Currently, the only allowed value for this field is "UnRcpType".
- **classDesc**: is the description of the recipe class.
- **deviceLink**: each recipe class must be linked to a PCO device. This field contains the PCO alias.
- **deviceType**, **deviceAlias** and **dpe**: these fields are used to add a new device element to the recipe class. For example, to add the "HH" element of an AnalogAlarm device to a recipe class these fields must contain the following data:
  - **deviceType**: AnalogAlarm
  - **deviceAlias**: QSDN_4_AA1 (example of AnalogAlarm alias)
  - **dpe**: HH

The **dpe** field can be empty. In that case, the recipe class will contain all the elements defined in the recipe type specified in the **typeName** field. The Table 1 summarizes the device types and elements allowed for the recipe type "UnRcpType".

<table>
<thead>
<tr>
<th>Recipe Type</th>
<th>Device Type</th>
<th>Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnRcpType</td>
<td>DigitalParameter</td>
<td>ManReg01</td>
</tr>
<tr>
<td></td>
<td>WordParameter</td>
<td>MPosR</td>
</tr>
<tr>
<td></td>
<td>AnalogParameter</td>
<td>MPosR</td>
</tr>
<tr>
<td>AnalogAlarm</td>
<td></td>
<td>HH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>L</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LL</td>
</tr>
<tr>
<td>Controller</td>
<td>MSP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MSPH</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MSPL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MOuTH</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MOuTL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MKc</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MTi</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MTd</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MTds</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 - Elements allowed in the recipe type UnRcpType.
The recipe class definitions file can contain additional data for the recipe classes and recipe instances, like:

- Domain
- Nature
- Access control domain
- Privileged actions

The fields used to define the additional data have the ‘c’ prefix for the recipe class data and the ‘i’ prefix for the recipe instance data. For example, the domain and nature fields for the recipe class and recipe instance are named: cDomain, iDomain, cNature and iNature.

The same principle is applied to the privileged actions where it is possible to define the users who can trigger a specific action, like the recipe activation. The allowed values for the privileged actions are: “admin”, “expert” or “operator”.

### 3.2.2.2 RECIPE INSTANCE DEFINITIONS

A recipe instance definition contains the list of values for each device alias and element defined in the recipe class.

<table>
<thead>
<tr>
<th>Instance1</th>
<th>RepClass1</th>
<th>Description</th>
<th>InitialRecipe</th>
<th>DeviceType</th>
<th>DeviceAlias</th>
<th>dpe</th>
<th>Value</th>
<th>Range</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance1</td>
<td>RepClass1</td>
<td>Example of RepClass1 instance true</td>
<td>-</td>
<td>DigitalParameter</td>
<td>DEMON_1_DP1</td>
<td>MRPc1</td>
<td>FALSE</td>
<td>[0, 50]</td>
<td>cm</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WordParameter</td>
<td>DEMON_1_WP1</td>
<td>MRPc2</td>
<td>50</td>
<td>[0, 50]</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AnalogParameter</td>
<td>DEMON_1_AP1</td>
<td>MRPc3</td>
<td>50</td>
<td>[0, 200]</td>
<td>cm</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AnalogAlarm</td>
<td>DEMON_1_A12</td>
<td>HH</td>
<td>50</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Controller</td>
<td>DEMON_1_C21</td>
<td>MSH</td>
<td>50</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MSHH</td>
<td>50</td>
<td>80</td>
<td>100</td>
<td>cm</td>
</tr>
</tbody>
</table>

Figure 7 - Recipe instance definitions example.

The recipe instance definitions file contains the following fields:

- **instanceName**: is the name of the recipe instance. The pair instanceName, className must be unique within the file.
- **className**: is the recipe class name (defined in the recipe class definitions file).
- **description**: is the recipe instance description.
- **initialRecipe**: each recipe class can have one initial recipe instance. This field contains a Boolean value to specify if the recipe instance is the initial one.
- **deviceType, deviceAlias and dpe**: these fields must be the same as defined in the recipe class.
- **value**: contains the value of the recipe element.
- **range**: contains the valid range of the recipe element.
• **unit**: contains the measurement unit of the recipe element.

### 3.2.2.3 RUN THE PLUG-IN GENERATION

When the recipe class definitions file is completed, it is possible to generate an example of the recipe instance definitions file by clicking the "Generate Recipe instances file" icon. The example file will be generated in the following path: `Output/ExtendedConfigGenerator/Recipes/RcpInstanceDefinitions.xml`

Once the recipe instances file is completed it **must** be copied to the user inputs folder: `Resources/ExtendedConfigGenerator/user-inputs`

Finally, to generate the WinCC OA import file just press the generate icon. If there are no errors in the input files the generated file will be placed in the following path: `Output/ExtendedConfigGenerator/Recipes/Recipes.txt`

### 3.3 WinCC OA

#### 3.3.1 CREATION OF A WINCC OA – CPC 6 APPLICATION

Please follow the procedure described in the document “CREATION OF A WINCC OA – CPC 6 APPLICATION”

[https://edms.cern.ch/file/1228441/1.5.0/Procedure_PVSS-UCPC_Application.pdf](https://edms.cern.ch/file/1228441/1.5.0/Procedure_PVSS-UCPC_Application.pdf)

#### 3.3.2 IMPORT THE RECIPES

To import the recipes file generated by the Extended Configurations plug-in do the following steps:

1. Start the **unicosHMI.pnl** interface.
   ```
   User Interface –p vision/graphicalFrame/unicosHMI.pnl
   ```

2. Login as **admin** and start the **Import Database** panel from by clicking on ‘UNICOS icon > Configuration > Import Database’ (Figure 8).

3. Select a proper driver number. Normally it should not be same number for both S7 and Modbus PLCs.
   - *For S7 (Siemens PLC)* pick up the next unique number (increment it).
   - *For Modbus (Schneider PLC)* you could have only one active driver.

4. Set the Front End type to ‘SOFT_FE’.

5. Stop the Distribution Manager from the console if it’s running.

6. Select the importation file ‘Recipes.txt’ generated by the generator.

7. Do a **check** (Figure 9) and then an **import** if there were no errors. The ‘view log file’ button shows the details of the errors or warnings of the data checking.

8. Stop the Simulation Driver.
9. Open the Front End Diagnostic and enable the communication of the RCP_FE front end (Figure 10).

![Figure 8 - Import Database](image1)

![Figure 9 - Check Ok.](image2)
3.3.3 USE THE RECIPE INSTANCES

3.3.3.1 BROWSE THE IMPORTED RECIPES

To browse the imported recipes do the following steps:

1. From the unicosHMI.pnl panel, click on the ‘Device Tree Overview’ button (Figure 11).
2. On the ‘Device Selection’ frame check on ‘Device type’ and click on the ‘Apply filter’ button.
3. Select the UnRcpInstance device from the Device Tree.
4. Double click on a recipe instance to open the recipe instance panel (Figure 12).

Once the recipe instance panel is opened, it is possible to browse the different recipes using the ‘Recipe class’ combo box and the ‘Recipes’ table located on the left side of the panel.
Figure 11 - Browsing the imported recipes.

Figure 12 - Recipe instance panel.
3.3.3.2 ACTIVATE A RECIPE INSTANCE

To activate a recipe instance do the following steps:

1. From the recipe instance panel (Figure 12) click on the ‘Select’ button.
2. Click on the ‘Activate’ button.

When the recipe activation is triggered the following messages will appear on the recipe log frame:

```
[2013-12-09 17:03:48] Activating recipe: 5eRcpClass / Instance
[2013-12-09 17:03:48] Sending recipe data to the PLC: dest_257.PLCSimao PLC
[2013-12-09 17:03:48] All recipe data sent to the PLC: dest_257.PLCSimao PLC
[2013-12-09 17:03:48] Recipe activation completed in PLC: dest_257.PLCSimao PLC
[2013-12-09 17:03:48] Unlocking recipe buffers...
[2013-12-09 17:03:48] Recipe buffers unlocked for PLC: dest_257.PLCSimao PLC
```

Figure 13 - Recipe activation log.

3.3.3.3 EDIT A RECIPE INSTANCE

To edit the values of a recipe instance do the following steps:

1. From the recipe instance panel (Figure 12) click on the ‘Select’ button.
2. Click on the ‘Edit’ button. The next actions will be enabled:
   a. **Online Values**: get the online values for all the recipe instance elements.
   b. **Save**: save the recipe instance values.
   c. **Cancel**: cancel the recipe instance edition.
3. To modify the value of a recipe element, introduce the desired value on the ‘Value’ field of the recipe instance table.
4. Finally, press the ‘Save’ button to save the changes or the ‘Cancel’ button to discard them.

3.3.3.4 CREATE NEW RECIPE INSTANCES

Currently, there are two ways of creating new recipe instances from the recipe instance panel:

1. Create a new recipe instance by clicking the ‘New Instance’ button. A pop-up window (Figure 14 - Create a new recipe instance.) will prompt for the recipe instance name and description. The values for the new recipe instance will be taken from the PLC (online values).

Figure 14 - Create a new recipe instance.
2. Duplicate an existing recipe instance clicking the 'Duplicate' button. A pop-up window will prompt for the recipe instance name and description. The values of the new recipe instance will be copied from the original recipe instance.

3.3.3.5 DELETE A RECIPE INSTANCE
To delete a recipe instance do the following steps:
1. From the recipe instance panel (Figure 12) click on the 'Select' button.
2. Click on the 'Delete' button and press 'Ok' in the confirmation message.

NOTE: The recipe instances marked as 'Initial' cannot be deleted.

3.3.3.6 INITIAL RECIPES
By pressing the 'Initial Recipes' button in the recipe instance panel (Figure 12 - Recipe instance panel.) a new window will be opened (Figure 15 - Initial recipes.) to display all the initial recipes in the PCO hierarchy (linked to the recipe instance PCO and its PCO children).

![Initial recipes window](image)

Figure 15 - Initial recipes.

From this window it is possible to do the following actions:
- **Activate All**: triggers the activation of all the recipes in sequence.
- **Check Status**: check if the recipes are active in the PLC.
- **Clear Log**: clear the log messages in the window.
- **Close**: close the initial recipes panel.
3.3.3.7 LAST ACTIVATED RECIPES

By pressing the ‘Last Activated’ button in the recipe instance panel (Figure 12 - Recipe instance panel.) a new window will be opened to display the last activated recipes in the PCO hierarchy (linked to the recipe instance PCO and its PCO children).

![Figure 16 - Last activated recipes.](image)

From this window it is possible to do the following actions:

- **Activate All**: triggers the activation of all the recipes in sequence.
- **Check Status**: check if the recipes are active in the PLC.
- **Clear Log**: clear the log messages in the window.
- **Close**: close the initial recipes panel.

3.3.3.8 SAVE RECIPE VALUES AS INITIAL

By pressing the button ‘Save as Initial’ in the recipe instance panel (Figure 12 - Recipe instance panel.) it is possible to save the values of the selected recipe in the recipe marked as 'Initial' for the recipe class.

3.3.4 DYNAMIC RECIPE CLASSES

Since version 1.3.0 of the unRecipe component it is possible to create recipe classes dynamically in WinCC OA without importing the recipes file. This section explains the how to use the dynamic recipes.
3.3.4.1 RECIPE CLASS CREATION

To create new recipe classes from WinCC OA click on the ‘Recipe Class’ entry in the UNICOS menu (Figure 17). A new window will be opened to be able to operate with the recipe classes (Figure 18).

Figure 17 - Recipe class menu entry.

Figure 18 - Recipe class panel.
3.3.4.2 CREATE NEW RECIPE CLASSES

To create a new recipe class click on the ‘New Class’ button on the recipe class panel (Figure 18 - Recipe class panel. A popup window will appear to ask for the necessary information to create a new recipe class (Figure 19):

- **Name**: Recipe class name.
- **Description**: Recipe class description.
- **PCO Link**: PCO to which the recipe class and all its recipe instances will be linked.

![Create Recipe Class](image1)

**Figure 19 - Create a recipe class.**

If the application does not have a recipe front-end the following window will popup:

![Recipe Front End Creation](image2)

**Figure 20 - Recipe front-end creation.**

It is recommended to choose a driver number different than the ones already existing in the application so it is possible to import the recipes without stopping the drivers used to communicate with the different front-ends. Once the recipe front-end is created, the new recipe class will appear in the ‘Recipe class’ list and it is possible to add new devices to the recipe.

3.3.4.3 ADD NEW DEVICES TO A RECIPE CLASS

To add new devices to an existing recipe class do the following steps:
1. Select the recipe class on the ‘Recipe Class’ list (Figure 18) and click on the ‘Select’ button.

2. Click on the ‘Add Devices’ button. A new window will popup containing all the devices available (Figure 21). Select the required devices for the recipe class in the ‘Device Alias’ list. It is possible to filter the devices using the ‘Device Type’ list and to search devices using the ‘Device Search’ box. Once the required devices are selected click on the ‘Ok’ button to continue. By default all the allowed elements of the selected device types will be added to the recipe class for the selected devices (i.e. for the AnalogAlarm device type the elements HH, H, L and LL will be added).

3. Click on the ‘Apply Changes’ button in the recipe class window (Figure 18) to include the new devices in the recipe class and all its recipe instances (if any). On the contrary, to undo the addition of devices to the recipe class, click on the ‘Cancel’ button.

![Figure 21 - Device selection for a recipe class.](image)

3.3.4.4 REMOVE DEVICES FROM A RECIPE CLASS
To remove devices from a recipe class do the following steps:

1. Select the recipe class in the ‘Recipe Class’ list (Figure 22) and click on the ‘Select’ button.

2. In the ‘Recipe Class Elements’ table (Figure 22), select the devices to remove from the recipe class.

3. Click on the ‘Remove Sel.’ Button and accept the confirmation message.

4. Click on the ‘Apply Changes’ button to remove the selected devices from the recipe class and all its recipe instances (if any). On the contrary, to undo the removal of devices in the recipe class, click on the ‘Cancel’ button.
Figure 22 - Remove devices from a recipe class.

3.3.4.5 EDIT THE DEVICE ELEMENTS OF A RECIPE CLASS

To edit the device elements included in a recipe class do the following steps:

1. Select the recipe class in the ‘Recipe Class’ list (Figure 18) and click on the ‘Select’ button.

2. Click on the ‘Edit Elements’ button. A new window will popup (Figure 23) to allow the edition of the recipe class device elements. This window shows all the devices included in the recipe class organized by device type and the allowed elements to include in the recipe class. If a single device is selected in the ‘Device Name’ list, its elements included in the recipe class will have a grey background in the ‘Properties’ list. From this window it is possible to select one or several devices of the same device type and select the desired elements to be included in the recipe class using the ‘Device Elements’ list. Once the required device elements are selected click on the ‘Apply’ button to continue with the modification of the recipe device elements.

3. Click on the ‘Ok’ button to close the ‘Edit recipe elements’ window.

4. Finally, click on the ‘Apply Changes’ button to apply the modifications in the recipe class and all its recipe instances (if any). On the contrary, to undo the modification of the recipe device elements click on the ‘Cancel’ button.
3.3.4.6 REMOVE A RECIPE CLASS
To remove an existing recipe class and all its recipe instances do the following steps:
1. Select the recipe class in the ‘Recipe Class’ list (Figure 18) and click on the ‘Select’ button.
2. A popup window will appear to confirm the removal of the recipe class and all its recipe instances.

3.3.4.7 DUPLICATE AN EXISTING RECIPE CLASS
To duplicate an existing recipe class do the following steps:
1. Select the recipe class to duplicate in the ‘Recipe Class’ list (Figure 18) and click on the ‘Select’ button.
2. Click on the ‘Duplicate’ button in the recipe class window. A new popup window will appear (Figure 19) to ask for the necessary information to create a new recipe class.
3. Introduce the data for the new recipe class and click on the ‘Ok’ button.

3.3.4.8 EDIT THE RECIPE CLASS AND RECIPE Instances PRIVILEGES
The recipe class and recipe instance privileges allows configuring the actions assigned to the different UNICOS roles (admin, expert, operator). The different actions correspond to the buttons included in the recipe class window (Figure 18) and recipe instance window (Figure 12). The actions assigned to the operator role can be also executed by the admin and expert users, and the actions assigned to the expert role can also be executed by the admin users.

To edit the recipe class privileges do the following steps:
1. Select the recipe class in the ‘Recipe Class’ list (Figure 18) and click on the ‘Select’ button.
2. A new window will popup (Figure 24) to show the different privileges assigned to the UNICOS roles. **Note:** The recipe instance privileges will be applied to all the instances of the recipe class when modified.

3. To modify the privileged actions assigned to a specific role click on the button located at the right of the UNICOS role. A new window will popup (Figure 25) to show the current actions assigned to the selected UNICOS role (table ‘In Selection’ on the right side of the window) and the unassigned actions (table ‘Not in selection’ on the left). In the current example (Figure 25) all the privileges are already assigned so it is only possible to remove the existing privileges for the **operator** role.

4. Once the privileges are assigned to the UNICOS roles click on the ‘Ok’ button (Figure 24) to apply the changes to the recipe class and all its recipe instances. The ‘Reset’ button will restore the privileges to the previous values. The ‘Load Default’ button will restore the default values for the recipe class and its recipe instances.

---

**Figure 24** - Privileges of the recipe class and its recipe instances.

**Figure 25** - Edit the recipe instance privileges for the operator.
3.3.4.9 CREATE A NEW RECIPE INSTANCE
To create a new recipe instance from the recipe class window do the following steps:

1. Select the recipe class in the 'Recipe Class' list (Figure 18) and click on the 'Select' button.
2. A pop-up window (Figure 14) will prompt for the recipe instance name and description. The values for the new recipe instance will be taken from the PLC (online values).

4. TROUBLESHOOTING
4.1 RECIPE ACTIVATION FAILED
The recipe activation can fail if the recipe activation timeout is not properly set.

4.1.1 SCHNEIDER
To verify the recipe activation timeout in a Schneider PLC do the following steps:

1. Open the project in Unity Pro XL and double click on the “Elementary Variables” item in the project structural view (Figure 26).
2. Type “_Timeout” in the variable name and press enter.
3. The value of the variable must be different than 0 (t≠0s). Otherwise, introduce a new timeout value (e.g.: t#100s).
4. After applying the changes it is necessary to build the changes and download the project to the PLC.

Figure 26 - Recipe activation timeout for Schneider.
4.1.2 SIEMENS

To verify the recipe activation timeout in a SIEMENS PLC do the following steps:

1. Open the project in Step-7, click on the program sources and double click on the “Recipes” source file (Figure 27).
2. The value of the ActivationTimeout variable must be different than 0 (Figure 28). Otherwise, introduce a new timeout value (e.g.: 100).
3. After applying the changes it is necessary to build the changes and download the project to the PLC.

Figure 27 - Recipes source file in Step-7.

Figure 28 - Recipe activation timeout for SIEMENS.
5. SUPPORT

Please address your questions to IceControls.Support@cern.ch

6. REFERENCES

1. *UNICOS in Siemens S7 PLC*. Bradu, B., Gayet, Ph., Blanco, E.
2. *UNICOS Project, Object programming principles*. Gayet, Ph.
3. *TSPP Unicos Manager*. Grahy, J.
APPENDIX A: RECIPE TYPE DEFINITION

The recipe type definitions file contains the list of device types and device elements that can be included in a recipe class and recipe instance (e.g. a recipe can contain the MPoSR element of a WordParameter device type). The default file path for the file is: 
Resources/ExtendedConfigGenerator/config/Recipes/RcpTypeDefinitions.xml

The recipe type definitions file contains the following fields:

- **defaultType**: contains a Boolean value used to specify if the recipe type is the default one. Only one recipe type can be used as default type.
- **typeName**: is the recipe type name. The value must be unique within the file.
- **deviceType**: this field is used to add a new device type to the recipe type. It contains a UNICOS device type name (e.g. “AnalogAlarm”).
- **dpe**: this field is used to add a new device type attribute to the recipe type. For example, the value “FEDeviceManualRequests:HH Alarm” will add the “HH” attribute of the AnalogAlarm device type to the recipe type.

The Figure 30 shows the relationships between the recipe types, recipe classes and recipe instances: a recipe class must belong to a recipe type and a recipe instance must belong to a recipe class.

![Figure 30 - Recipe type, class and instance relationship.](image-url)