CREATION OF A SCHNEIDER UNICOS-CPC 6 APPLICATION

Abstract

This procedure explains how to create a Schneider UNICOS CPC6 application from the specifications using the UAB (UNICOS Application Builder) generation tool.
## HISTORY OF CHANGES

<table>
<thead>
<tr>
<th>REV. NO.</th>
<th>DATE</th>
<th>PAGES</th>
<th>DESCRIPTIONS OF THE CHANGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.0</td>
<td>2012-11-21</td>
<td></td>
<td>First version (Silvia Izquierdo)</td>
</tr>
<tr>
<td>V1.1</td>
<td>2012-02-24</td>
<td></td>
<td>General corrections after J.M. Beckers revision</td>
</tr>
<tr>
<td>V1.2</td>
<td>2012-03-16</td>
<td></td>
<td>General corrections after B. Bradu revision</td>
</tr>
<tr>
<td>V1.3</td>
<td>2012-03-26</td>
<td></td>
<td>Changes made according to new generation tool</td>
</tr>
<tr>
<td>V1.4</td>
<td>2012-04-05</td>
<td></td>
<td>General corrections after J. Ortola revision</td>
</tr>
<tr>
<td>V1.5</td>
<td>2012-05-31</td>
<td></td>
<td>Changes about WinCC Flexible generator</td>
</tr>
<tr>
<td>V1.6</td>
<td>2012-06-27</td>
<td>6</td>
<td>Installation of UAB tool added</td>
</tr>
<tr>
<td>V1.7</td>
<td>2012-06-29</td>
<td></td>
<td>Change references to point to EDMS for release v1.3.1</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

1. INTRODUCTION ................................................................................................... 5
2. REQUIREMENTS ................................................................................................... 6
3. PROCEDURE ........................................................................................................ 6
   3.1 INSTALLATION OF UAB GENERATOR TOOL .................................................. 6
   3.2 FILL IN THE SPECIFICATIONS ....................................................................... 6
   3.3 CONFIGURATION ............................................................................................ 7
      3.3.1 ARCHITECTURE OVERVIEW ..................................................................... 7
      3.3.2 CREATING THE PLC PROJECT ............................................................... 7
      3.3.3 HARDWARE CONFIGURATION .................................................................. 8
      3.3.4 ETHERNET CONNECTION CONFIGURATION ......................................... 9
      3.3.5 WINCC OA CONFIGURATION ................................................................ 12
   3.4 UAB GENERATION TOOL ............................................................................... 13
      3.4.1 GENERATION PRINCIPLES ...................................................................... 13
      3.4.2 GENERATION PROCEDURE ..................................................................... 14
      3.4.3 UAB USER REPORT ............................................................................... 21
      3.4.4 COMMON ERRORS .................................................................................. 22
      3.4.5 LOG ........................................................................................................ 23
      3.4.6 GENERATED FILES .................................................................................. 23
      3.4.7 SEMANTIC RULES .................................................................................. 24
      3.4.8 TEMPLATES ............................................................................................. 24
      3.5 IMPORTATION INTO UNITY ....................................................................... 25
         3.5.1 IMPORTATION OF INSTANCE AND LOGIC FILES ............................... 25
         3.5.2 IMPORTATION OF COMMUNICATION FILE ......................................... 26
         3.5.3 SECTIONS ORDER ............................................................................... 27
         3.7 COMPLETE LOGIC FILES ....................................................................... 28
      3.8 PLC SOURCES COMPILATION ..................................................................... 28
      3.9 CREATION OF WINCC OA PROJECT .......................................................... 29
      3.10 USAGE OF LOCAL PANELS ....................................................................... 29
   4. SUPPORT .......................................................................................................... 29
   5. REFERENCES .................................................................................................... 29
## TABLE OF FIGURES

- Figure 1 - General procedure ................................................................. 5
- Figure 2 - Schneider Front End Device IO Configuration ............................. 6
- Figure 3 - Example of a typical configuration ........................................... 7
- Figure 4 - Hardware PLC configuration .................................................... 8
- Figure 5 - CPU Configuration ................................................................. 8
- Figure 6 - Opening Ethernet connection .................................................... 9
- Figure 7 - IP Configuration ................................................................. 9
- Figure 8 - Messaging ........................................................................... 10
- Figure 9 - Opening Ethernet link ............................................................ 10
- Figure 10 - Net link .......................................................................... 10
- Figure 11 - Ethernet communication not linked ....................................... 10
- Figure 12 - Fields matching between the generator and the PLC configuration 11
- Figure 13 - Final PVSS configuration ....................................................... 12
- Figure 14 - UAB CPC Wizard ................................................................ 13
- Figure 15 - UAB Bootstrap ................................................................... 14
- Figure 16 - UNICOS Application Builder ............................................... 14
- Figure 17 - Application General Data ..................................................... 15
- Figure 18 - Schneider PLC specifications ............................................... 16
- Figure 19 - Schneider generators ........................................................... 16
- Figure 20 - Unity instance generator ....................................................... 17
- Figure 21 - Unity Logic Generator ........................................................... 18
- Figure 22 - WinCC OA instance generator ............................................... 19
- Figure 23 - WinCC Flexible Instance Generator ....................................... 20
- Figure 24 - UAB user report ................................................................. 21
- Figure 25 - Failure status .................................................................... 22
- Figure 26 - Required value is not specified ............................................. 22
- Figure 27 - Empty mandatory field ......................................................... 22
- Figure 28 - Exceeding maximum name length ....................................... 23
- Figure 29 - Not mandatory fields can become mandatory ....................... 23
- Figure 30 - Object does not exist .......................................................... 23
- Figure 31 - Importation ...................................................................... 25
- Figure 32 - Instance file importation ..................................................... 25
- Figure 33 - Logic file importation .......................................................... 26
- Figure 34 - Communication file importation .......................................... 26
- Figure 35 - Communication file importation .......................................... 27
- Figure 36 - PLC sources compilation .................................................... 28
1. INTRODUCTION

The goal of this document is to provide a procedure of how to create a Schneider UNICOS-CPC application using the Schneider PLC family (PREMIUM or QUANTUM).

The general architecture of an UNICOS Schneider application could be found in the “UNICOS CPC6 on Unity” document [1].

The general steps for the creation of a Unity UNICOS-CPC project are summarized as:

1. Filling in the specification file according to the application (Spec.xml).
2. Generation
   2.1. Running Unity Instance Generator to get the instances code for the PLC.
   2.2. Running Unity Logic Generator to get the control logic code for the PLC.
   2.3. Running WinCC OA Instance Generator to get the importation file for the supervision.
3. Importation of the object instances source files into the PLC project (Unity)
4. Importation of logic source files into the PLC project (Unity)
5. Completion of the Logic in the PLC project (it can also be done by using User Templates)
6. Downloading project to the PLC
7. Creation of the WinCC OA project
8. Importation of instances into WinCC OA
9. Completion of panels for supervision

![Figure 1 - General procedure](image-url)
2. **REQUIREMENTS**

   Developing a Schneider UNICOS application implies the usage of several software packages:
   - Specification and Generation tools: **MS Office** (Excel v2007/v2010), **Java SE Runtime Environment** (v6) and **UAB** (v1.3.x)
   - PLC: **UNITY**
   - SCADA: **PVSS**\(^1\)/**WinCC OA**

3. **PROCEDURE**

3.1 **INSTALLATION OF UAB GENERATOR TOOL**

   The different tools specified in the section “Error! Reference source not found.” must be installed. For the installation of UAB tool follow the instructions described in the following link.

   [http://cern.ch/enice/Automatic+Generation+tools+%28UAB%29](http://cern.ch/enice/Automatic+Generation+tools+%28UAB%29)

3.2 **FILL IN THE SPECIFICATIONS**

   The specification file is an **xml** file where all the UNICOS objects must be defined and parameterized. To know about how to fill the specification file check the document “UCPC Spec Documentation” [2].

   If you were a UNICOS user, you will notice there are some changes in the specifications. To convert a specification file from CPC5 to CPC6 you can use the “CPC5 to CPC6 Spec Conversion Tool” [3].

   Concerning the PLC I/O periphery addressing, is necessary to look at document “**FEPparameters.xlsx**”, that you can find in the UAB project “...\UAB_project\Specs\”. You can see a screenshot of this document in Figure 2.

![Figure 2 - Schneider Front End Device IO Configuration](image)

---

\(^1\) The software used for the supervision **PVSS** is nowadays known as **WinCC OA**, in general in this tutorial we will be referred to that software as WinCC OA
### 3.3 CONFIGURATION

#### 3.3.1 ARCHITECTURE OVERVIEW

The SCADA (WinCC OA) and the PLC are linked by the Ethernet Network. In Figure 3 you can see a typical configuration of a UNICOS-CPC6 project.

![Diagram showing the integration of SCADA and PLC via Ethernet](image)

**Figure 3 - Example of a typical configuration**

#### 3.3.2 CREATING THE PLC PROJECT

To create the Project in Unity we can open Schneider baseline, which can be found in the UAB Project “…\Project\Baseline\”. The baseline must be unzipped previously. Once it is opened in Unity, it is possible to save it with a different name in the desired location.
### 3.3.3 HARDWARE CONFIGURATION

The PLC hardware must be configured in Unity according to the existing one.

*Figure 4 - Hardware PLC configuration*

If you change the CPU, pay attention to the configuration of the address fields of the CPU (double click on the CPU). The UNICOS standard configuration is the one you get with the baseline; you can see it in Figure 5. If you don’t configure it with the standard of UNICOS, some of the project utilities may not work properly.

*Figure 5 - CPU Configuration*
3.3.4 ETHERNET CONNECTION CONFIGURATION

To configure the Ethernet connection, you can open it from the Structural view in “Communication > Networks > Ethernet_1”. See Figure 6.

![Figure 6 - Opening Ethernet connection](image1)

To configure the connection, in the tab ‘IMP Configuration’ write the corresponding “IP address” of the PLC, “Subnetwork mask” and “Gateway address”, according to the configuration chosen in the generation. In the tab ‘Messaging’ write the corresponding “Network” and “Station”.

NOTE: even if you are not using it, it is mandatory to write a second WinCC OA IP address (it is recommended to check that the address is not used in another project, it could generate conflicts).

![Figure 7 - IP Configuration](image2)
After the configuration it is necessary to make the link with the connection. Double click on the Ethernet connector on the hardware configuration, see Figure 9.

In “Net Link” choose the connection configured, see Figure 10.

If the communication is not linked properly, the icon in the structural view will look like in Figure 11.
The parameters chosen have to match with the ones in the generator, or the communication between the PLC and PVSS won’t work.

Figure 12 - Fields matching between the generator and the PLC configuration
3.3.5 WINCC OA CONFIGURATION

The PVSS configuration will be **generated automatically** according to parameters introduced within the generation procedure. Next figure only shows the final configuration in the PVSS parameterization.

![MODBUS parameterization](image)

**Figure 13 - Final PVSS configuration**
3.4 UAB GENERATION TOOL

3.4.1 GENERATION PRINCIPLES
The UAB CPC-Wizard (UNICOS Application Builder) consolidates all generation steps in one tool, making easier and intuitive the generation procedure.

The generators used for a Schneider application are four:
- Unity Instance Generator: it generates the instances code for Schneider PLC
- Unity Logic Generator: it generates the logic code (placeholders and/or complete logic) for Schneider PLC and interconnections between PCOs and Field Objects.
- WinCC OA Instance Generator: it generates the importation file for the supervision.
- WinCC Flexible Instance Generator: it generates the instance files used for touch panel applications.

In Figure 14 you can see the window where you can choose which generator you want to launch.

Figure 14 - UAB CPC Wizard
3.4.2 GENERATION PROCEDURE

Start UCPC Wizard, you can start it running UAB Bootstrap (Figure 15) and clicking ‘cpc-wizard’.

On the first window you get (Figure 16), choose ‘New UNICOS Application for Schneider, and then choose the resource package you are interested in. Finally, choose the project location and click ‘Next’.
In the ‘Application General Data’ window (Figure 17) write a **name for the project** and a **name for the application**. Then we specify the **specification file**. The rest of the fields are optional. The ‘User Tag’ field introduces a tag to indentify and categorize your application visible at the SCADA level. Click ‘Next’ for next step.

![Application General Data](image)

**Figure 17 - Application General Data**

In the window “SCHNEIDER PLC Specifications” (Figure 18) choose the PLC configuration. Write a **name for the PLC** (please use the proper naming for the PLCs following CERN-wide recommendations, e.g.: CFP-864-MyPLC), and choose the **PLC Type** (QUANTUM or PREMIUM).

Then set the **Ethernet Parameters** (PLC IP address, SCADA IP address, gateway and network mask) and the **MODBUS Parameters**, and click ‘Next’. The PLC configuration must take into account the PLC settings (e.g. other connections may be already in place).

As mentioned before, even if you are not using it, it is mandatory to write a second WinCC OA IP address (it is recommended to check that the address is not used in another project, it could generate conflicts).
Then you will get the window “UNICOS Generators for SCHNEIDER” (Figure 19), where it is possible to choose between the three PLUGINS: **Unity Instance Generator**, **Unity Logic Generator** and **WinCC OA Instance Generator**.

**Figure 18 - Schneider PLC specifications**

**Figure 19 - Schneider generators**
**Unity Instance generator**

To get the Instance source files it is necessary to run the *Unity Instance Generator*. Choose ‘Unity Instance Generator’ in Figure 19 and click “Next”. In Figure 20 you can see the generator and an explanation of the different options and fields of this generator.

It is possible to select all object types at once by clicking the ‘Select All’ button, and then you can click on ‘Generate’ to get the instance generation. The generation can be found on the folder “…\Project\Output\S7CodeGenerator\".

It is recommendable to generate everything the first time and then use partial generation whenever you need to have individual generation of devices (e.g. Analog input only).

To do the commissioning of input/output signals (instrumentation, actuators) the option “Generate Commissioning File” has to be selected, then a check list is generated in your project. You can find the file in “…\Project\Output\S7CodeGenerator\UnityInstanceGenerator\".

To know more about semantic rules check section 3.4.7.

It is possible to click on ‘Logic Generator’ button to switch to *Unity Logic Generator*.
**Unity Logic generator**

In Figure 21 you can see the *Unity Logic Generator* window, and an explanation of the different options available.

To get all the logic code, it is necessary to select all object Types, masters, and finally all the logic sections. To do so, you can click on ‘Select All’ buttons. To get the logic generation you click on ‘Generate’ button. Generation can be found on “…\Project\Output\S7LogicGenerator\”.

![Figure 21 - Unity Logic Generator](image)
Wincc OA Instance generator

And finally, we are generating PVSS importation file. To get to WinCC OA Instance Generator you can click on ‘PVSS Generator’ button on the previous window.

It is possible to select all device types by clicking ‘Select All’ button, and then just click on ‘Generate’ button to get the generation started. PVSS importation file generated can be found on “…\Project\Output\PvssInstanceCodeGenerator\”.

![Wincc OA Instance generator](image)

*Figure 22 - WinCC OA instance generator*
WinCC Flexible Instance Generator

This generator is just used if you are using a touch panel for local operation. To get to WinCC Flexible Instance Generator you can click on ‘WinCC Flexible Instance Generator’ button on the previous window (Figure 22).

It is possible to select all device types by clicking ‘Select All’ button, and then just click ‘Generate’ button to get the generation started. The WinCC Flexible importation files generated can be found on “…\Project\Output\WinCCFlexInstanceGenerator\”.

![Figure 23 - WinCC Flexible Instance Generator](image)

To know more about how to create a touch panel application take a look at the procedure “Creating and designing a local panel for a UNICOS application” [4].
3.4.3 UAB USER REPORT

To check the status of the different generation steps and the errors or warnings produced, we can check the UAB User Report.

It is possible to filter the different messages in the report by checking/unchecking the message type buttons. The meaning of the buttons can be seen in Figure 24. It is also possible to clear all messages appearing in the report by clicking ‘Clear’ button, and to copy the messages to clipboard using ‘Copy to Clipboard’ button.

Figure 24 - UAB user report

If the generation has been successful, you can read next message in the report depending on the generator used:

“The exit status of the S7CodeGenerator plug-in is SUCCEED”
“The exit status of the S7LogicGenerator plug-in is SUCCEED”
“The exit status of the PVSSInstanceCodeGenerator plug-in is SUCCEED”

The report is very useful to identify the reason of the errors, if there are any errors.
3.4.4 COMMON ERRORS

If there is any error in the generation, the generation status is FAILURE, as you can see in Figure 25. We will comment some of the most common errors.

![Figure 25 - Failure status](image)

There are different types of errors.

If the user forgets to fill a mandatory field, he will get an error like the one in Figure 26.

![Figure 26 - Required value is not specified](image)

As you can see, the message indicates the object type, the name, and the field. The object type in the figure is ‘ProcessControlObject’ called ‘DEMON_1_DemonPCO’, instance number 1, and the field is ‘FEDeviceParameters:ParReg:Manual Restart after Full Stop’, like this, it is very easy to identify the field in the spec and fill it to solve the error.

![Figure 27 - Empty mandatory field](image)

There is maximum size for object names: for Schneider it is 21 letters for PCO and 23 for all other objects, and it is not possible to use any special characters (space, !, &, ^, %, #...). If the user exceeds the maximum length, there will be an error message like the one in Figure 28.
Some options transform not mandatory fields into mandatory. We can see an example in Figure 29: when an Analog Alarm is a Full Stop, master definition is mandatory.

If the user commits a mistake writing object names in the specification file, he will get an error as the one in Figure 30, which says that the object doesn’t exist. To solve this error check the device name.

3.4.5 LOG

Every time the user runs a generation, a log file is created with general information and the warning and error messages. This files can be found in “<User project path>\Log\”.

There is also a log called ‘UABLogging.html’ in which all generation messages are recorded, it is updated with every generation.

3.4.6 GENERATED FILES

All generated files can be found in “<user project path>\Output”. The output generated is distributed in three folders: UnityInstanceGenerator, UnityInstanceGenerator and UnityLogicGenerator.
The Instance Generation output is inside the folder called UnityInstanceGenerator. The Instance Generator generates:

- A file called “communication_DS.xst”.
- An instance file called “plc_instances_file.xfm”.

The Logic Generator output is inside the folder called UnityLogicGenerator. It generates:

- A file which contains the generated logic named “plc_logic_file”

The WinCC OA Instance Generator generates two files: “wincc_oa_db_file_AppName.txt” and “wincc_oa_comm_file.txt” inside WinCCOAInstanceGenerator, these are the importation files the user will import from WinCC OA.

The WinCC Flexible Instance Generator generates five files:
- an importation file for the alarms called “winccFlex_alarm.csv”
- a text file for the main script called “winccFlex_script.txt”
- a text file for the security settings script called “winccFlex_securitySettingsScript.txt”
- an importation file with the variables called “winccFlex_tags.csv”
- an importation file with text lists called “winccFlex_textLists.csv”

3.4.7 SEMANTIC RULES

The templates used for the semantic rules analyse the specifications file against the existing constraints (if applicable) to find out if there is any semantic error. These rules take into account the existing rules into PLC and WinCC OA.

You can check these constraints for the different fields in “UCPC Spec Documentation” [2].

Example: it checks the maximum length of names mentioned before in section 3.4.4.

3.4.8 TEMPLATES

The templates used by generators are inside the folder “<user project folder>\Resources”. They are divided in several folders:

- **Device types**: description of the objects
- **WinCCOAInstanceGenerator**: templates used to generate the WinCC OA importation file

The following folders are the ones containing the templates used for the generation of code or importation files:

- **UnityInstanceGenerator**: templates used to generate instances
- **UnityLogicGenerator**: templates used to generate logic
- **SemanticCheckRules**: templates used to check semantic rules
- **Upgrade**: templates used to upgrade projects to the latest version of CPC6

To know more about the usage of User Templates you can check the document “Logic Template Usage for UNICOS-CPC Applications” [3].
3.5 IMPORTATION INTO UNITY

As it has been mentioned in this procedure before, it is necessary to create a Unity project with Schneider baseline and the PLC hardware configuration.

3.5.1 IMPORTATION OF INSTANCE AND LOGIC FILES

From the view ‘Functional view’, right click over “Baseline Premium...” and click on ‘Import…’.

Look for the instance file generated in the folder ‘...\Project\UnityInstanceGenerator\’ and click on ‘Import’.

Now do the same with the logic file generated (‘...\Project\UnityLogicGenerator\’). You will get a window indicating some names are duplicated; choose the option ‘Replace All’.
3.5.2 IMPORTATION OF COMMUNICATION FILE

Go to the folder called “Sections” (Program > Tasks > MAST > Sections), right click and click on ‘Import…’. See Figure 34.

![Figure 34 - Communication file importation](image)

Select the generated file, you can find it in “\Project\Output\UnityInstanceGenerator\”, and click on ‘Import’. See Figure 35.

![Figure 35 - Importing UnityInstanceGenerator](image)
3.5.3 SECTIONS ORDER
For a correct operation of the application, it is necessary to have the right order of the sections. The correct order is:

- CPC_Comm: Time Manager
- CPC_Comm: DS_PLCC_Command
- CPC_Comm: Extract_Date_Time
- Communication
- Inputs: DI, AI
- Parameters: DPAR, APAR
- PCOs logic and field objects: _IL, _CL, _BL, _DL, etc
- Status: AS, WS
- Alarms: DA, AA
- Field objects: Analog, OnOff, AnaDig, etc
- Outputs: AO, DO

The reason of this order is that first it is necessary to process the inputs, to execute the logic and obtain the outputs.
3.7 COMPLETE LOGIC FILES

Each PCO contains 9 main functions (BL, CDOL, CL, DL, GL, IL, INST, SL, and TL) and one function by Dependent Logic Object (DL).

All these functions are generated by the Logic Generator.

It is necessary to fill these functions with the specific logic of the project.

The meanings of the functions filled automatically with the generator are:

- **BL**: Basic Logic, contains the basic logic of the PCO.
- **CDOL**: Common Dependant Object Logic, calculate the auto requests of all dependant objects and calculate the alarm acknowledgment.
- **CL**: Configuration Logic, calculate state On/Off of the objects according to other objects.
- **INST**: Instantiation, PCO instantiation, contains a call to CPC_FB_PCO and it is executed once per each instance.

It is necessary to fill some of the functions with the specific logic of the project:

- **GL**: Global Logic, allow defining the global logic of the PCO.
- **IL**: Interlock Logic, alarm instantiation and calculation of the Interlocks of the PCO (Start, Temp, Stop Interlock).
- **SL**: Sequencer Logic, sequential behaviour of the PCO (generally with Grafcet).
- **TL**: Transition Logic, contains a call to the process grafcet.
- **DL**: Dependent Logic of every object linked of the PCO. This function conditions the behaviour of these objects according to the grafcet states. These objects can be: PCO, Controller, On/Off, Analog, AnaDig, and AnaDO. All these objects have a Dependent Logic function associated to their PCO master.

  *Note*: PCO can be its own master.

It is also necessary to complete the Dependent logic (_DL) files of every field object with their specific logic.

Once this is finished and the compilation has no errors, it is possible to download the project to the PLC and run it.

3.8 PLC SOURCES COMPILATION

Once we have imported all the files, we must compile the Project. For this we can click on the button ‘Rebuild all project’ in Unity (see Figure 36).

![Figure 36 - PLC sources compilation](image)

If there are no errors in the compilation, the Project is ready for the configuration of the specific logic of the application.
3.9  CREATION OF WINCC OA PROJECT
To create a new WinCC OA project for the supervision, you can check the procedure called “Procedure WinCC OA-CPC Application” [6].

You can design the panels for the supervision in the project and link it to the PLC project.

3.10 USAGE OF LOCAL PANELS
To design an application on a local touch panel, you can check the procedure called “Creating and designing a local panel for a UNICOS application” [4].

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

5.  REFERENCES
[1] Schneider general architecture
   http://cern.ch/enice/UNICOS-CPC+Documentation/
   https://edms.cern.ch/file/1228441/1.3.1/Specs_doc.xml
[3] CPC5 to CPC6 Spec Conversion Tool
   http://cern.ch/enice/Reverse+Engineering+Tools
[4] Creating and designing a local panel for a UNICOS application
   https://edms.cern.ch/file/1228441/1.3.1/Procedure_Touch_Panel_Application.pdf
[5] Logic Template Usage for UNICOS-CPC Applications
   https://edms.cern.ch/file/1228441/1.3.1/UCPC6_UserTemplate.pdf
[6] Procedure WinCC OA-CPC Application
   https://edms.cern.ch/file/1228441/1.3.1/Procedure_PVSS-UCPC_Application.pdf