

TM840

APROL Parameter Management and Recipes



Requirements

Seminars	SEM841.5 APROL Basic 1
Software	APROL Automation Runtime SuSE LINUX
Hardware	1 control computer, 1 controller

Table of contents

1 Introduction.....	4
1.1 Objectives.....	5
2 Parameters in Engineering.....	6
2.1 Creating an equipment part.....	7
2.2 Assigning the parameter variables in the function chart.....	10
2.3 Creating parameter templates (plant parts).....	12
2.4 Recipe completed.....	15
3 Parameters in the runtime environment.....	16
3.1 The ParameterCenter.....	16
3.2 New recipes.....	17
3.3 Action logging.....	18
3.4 Recipes are active.....	19
4 Uploading parameters.....	20
4.1 Parameter upload in the runtime environment.....	21
5 Exporting and importing parameter sets.....	24
6 Documentation of the parameter sets and recipes.....	25
7 Handling parameters from the logic.....	27
7.1 Recipe download using the logic.....	28
7.2 Recipe upload using the logic.....	29
7.3 Implementing automatic parameter handling.....	30
8 Creating external parameter sets.....	31
8.1 Creating parameter set description.....	31
8.2 Create parameter set.....	32
8.3 Create parameters for the parameter set.....	34
8.4 Deleting data sets.....	35
9 Summary.....	37
10 Appendix.....	38
10.1 Objectives checklist (a few on-topic questions).....	38
10.2 Responses to the objectives checklist.....	38

1 Introduction

In a process control system it is often necessary to set new values during system runtime. Values can be entered using the operating templates provided. This procedure can get relatively complicated when dealing with numerous different configurations.

To make this work a little easier and more certain for the operator, the significant parameters (i.e. values of variables) are grouped into a parameter set for a specific part of the entire process control system. The term 'recipes' is applied when multiple parameter sets are being used.



Figure 1: Recipes and parameters

To get started, let's treat ourselves to a "Green Iceberg" cocktail, recipe: • 4 tsp. crushed ice, • 4 cl peppermint liqueur, • 2 cl vodka, • 1/8 liters soda-water, • 1 mint branch



Figure 2: Green Iceberg

1.1 Objectives

Our goal is to configure and to manage a complete method for handling parameters. The relationships between system equipment, plant parts and recipes will be clarified. Instant extensions and making your own recipes, as well as integrated functionalities and SQL handling will be realized.

Participants will be able to apply this knowledge to their own systems.

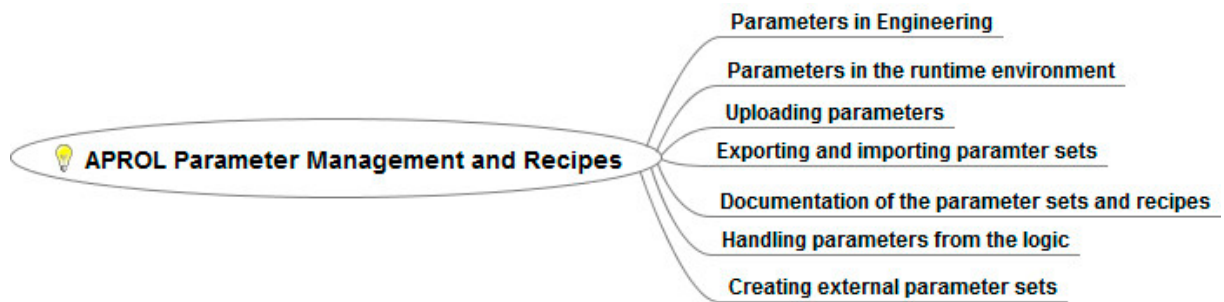


Figure 3: Overview

2 Parameters in Engineering

The ParameterCenter in the APROL RUNTIME and operating environment allows the user to manage, change and load a configuration for parameter handling (recipes) to system logic. This parameter handling is based on an SQL database. This makes it easy to implement a coupling to other higher-level systems, in order to enter production data (LEVEL 2).

To handle this, the **most important parameters** (or values of variables) for a part of the complete process control system are grouped together in a **parameter template**. The following could then be considered to be true:



Note:

A parameter template represents a plant part.

The different combinations of affected parameters (e.g. the set values for the various products) are grouped together and stored in different **parameter sets** in the template of the plant part. This means that it is no longer necessary to enter the changed equipment parameter separately when changing from product A to product B. The parameter set (the settings) for product B is selected from the plant part (e.g. "Line1") and is downloaded to the system "at the push of a button".

Another example would be the management of batches. Production data such as number of pieces, thresholds for production, descriptions, batch numbers, etc. can be maintained with internal tools or 'connected'.

To do this, the necessary **equipment** must first be defined with the parameters that will be changed. Several equipment parts, or even only one equipment part, will be created from these equipment parts afterwards. For this part of the system, the necessary parameter sets are put together for the different products and are then made available for the download.

2.1 Creating an equipment part

The **configuration of a piece of equipment** starts by adding a new project part of the same type under the menu **'Create project part / Equipment'** (right mouse button) and entering the necessary data in the **'Create equipment'** window. The instance must be unique to the entire project. Before doing this, there should be a clear and logical structure. All equipment entries correspond to a specific piece of equipment (valve, motor, etc.). By this point in the engineering task, the way in which these different pieces of equipment interact should already have been defined because these will definitely be incorporated in the structure.

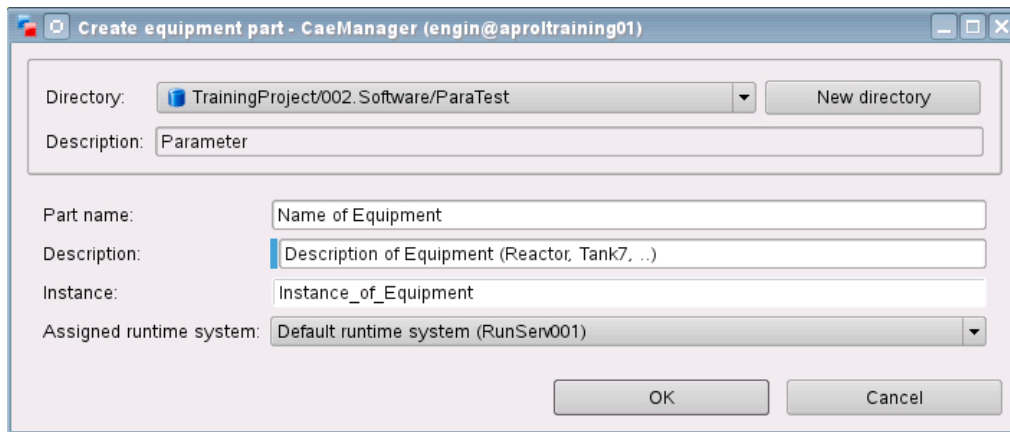


Figure 4: Create an equipment part

When the new piece of equipment is opened (double-click on the created object), an empty page appears in the **'Parameter'** tab. The variables required for this piece of equipment are added in this view. Up to this point, this involves only a list of the variables available after saving, activating and compiling for the function chart engineering.

The instance name can be edited in the **'Master data'** tan if necessary. The **'Equipment constants'** and **'Documentation'** areas are also available. These are explained in the following.

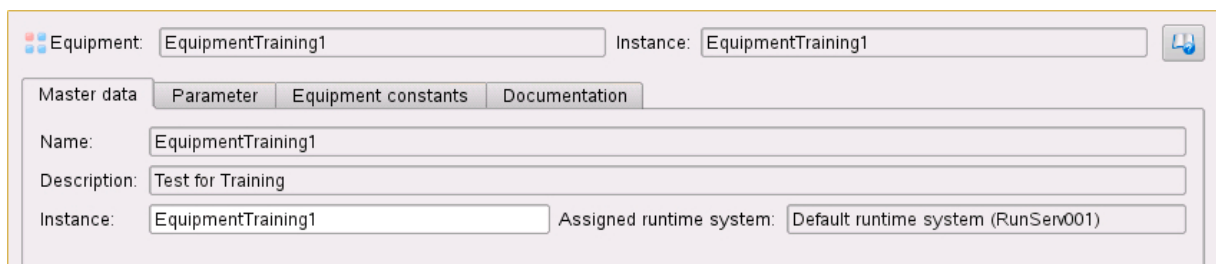


Figure 5: Equipment master data

New variables, which must then be written accordingly, can be created with the **'New'** context menu in the **'Parameter'** tab.

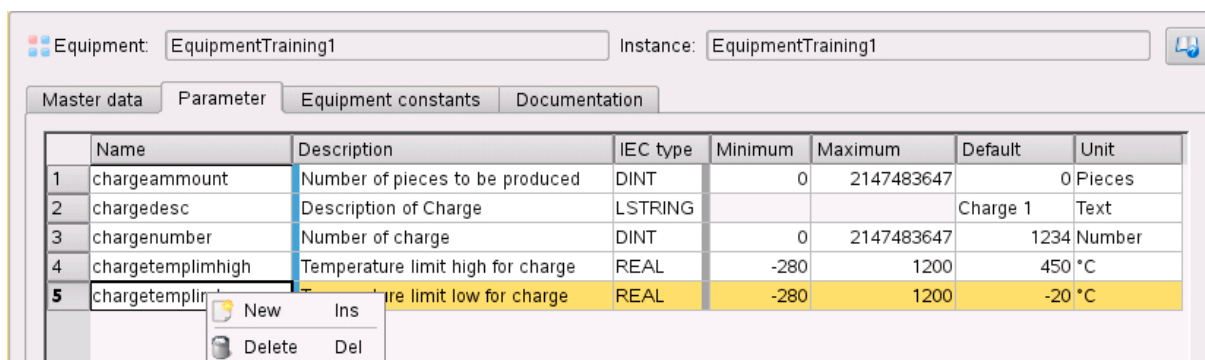


Figure 6: Entering equipment variables

Required **entries** (parameters):

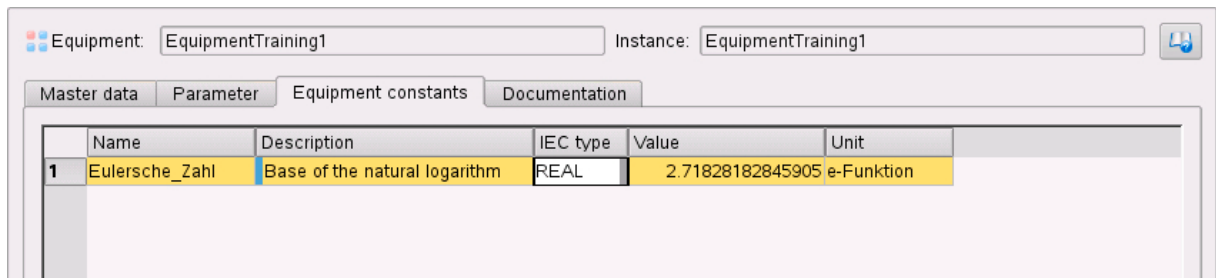
Description	Description
Name	Name of the parameter whose value can be changed using a parameter set. This parameter can be accessed on the function chart input bars once the piece of equipment has been compiled.
Description	Descriptive text (long text) This entry should be made no matter what!
Type	Selection of the variable type (data type).
Minimum	The minimum that can be assigned to a variable in a parameter set.
Maximum	The maximum that can be assigned to a variable in a parameter set.
Default	The default value of the variable in the parameter set. A sensible recipe base can be created by using sensible default values.
Unit	The physical unit of the data record.



Note:

Incorrect entries can be easily prevented by applying a minimum and maximum entry.

Equipment constants can also be defined for each piece of equipment. These are specified under the **'Equipment constants'** tab. Values are entered in the same way as for the parameters, but the minimum, maximum, and default values do not apply. This means that only the constant value is entered with the type, description, and unit. Typical constants are elements for calculations, such as Euler's number or PI. This can also be used to organize the project and depict configuration data (alarm priority, etc.).



Equipment: EquipmentTraining1 Instance: EquipmentTraining1

Master data Parameter Equipment constants Documentation

	Name	Description	IEC type	Value	Unit
1	Eulersche_Zahl	Base of the natural logarithm	REAL	2.71828182845905	e-Funktion

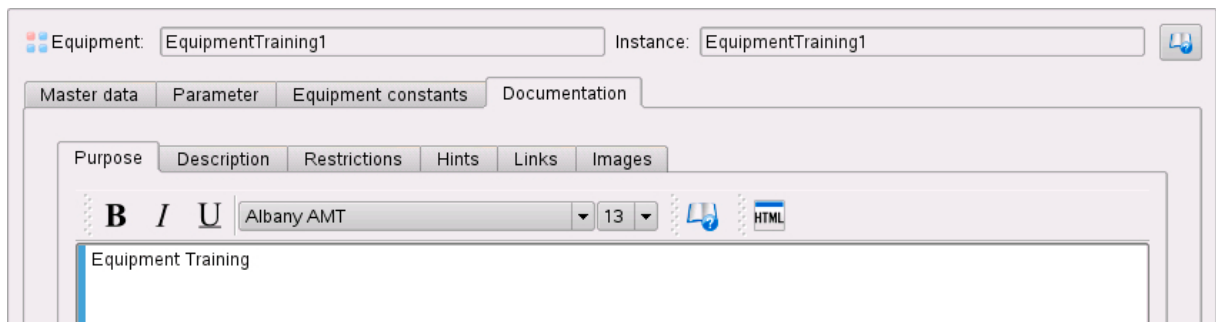
Figure 7: Equipment constants



Note:

The equipment constants must only be changed in the plant part!

The last tab is just for the equipment part documentation. This is available via the '**Show documentation (as-built)**' context menu and can be used as part of the project documentation. This documentation is created in the same way as all other project parts.



Equipment: EquipmentTraining1 Instance: EquipmentTraining1

Master data Parameter Equipment constants Documentation

Purpose Description Restrictions Hints Links Images

B I U Albany AMT 13  

Equipment Training

Figure 8: Equipment documentation

2.2 Assigning the parameter variables in the function chart

Once the equipment has been created with the respective variables, they can be accessed during function chart engineering and therefore a **connection to the program logic** can be established.

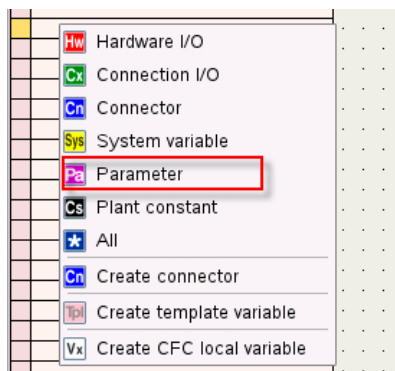


Figure 9: Parameter variable in CFC

Logical filters, which are useful during selection, simplify and speed up the engineering. The parameters must now be connected to the respective logic.

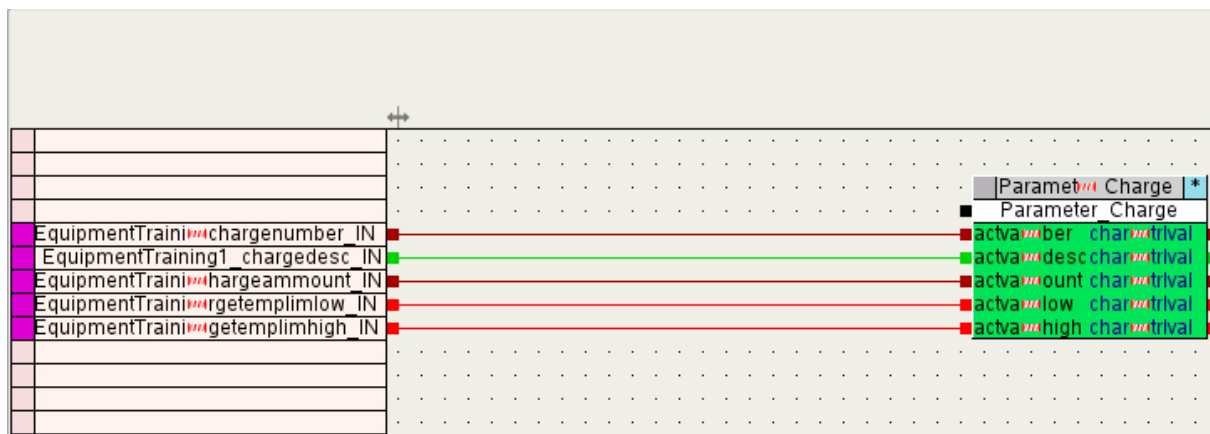


Figure 10: Parameters in the CFC

The function chart must be saved, activated and compiled once the parameters have been successfully assigned. The programming logic is thereby complete. However, the parameters must still be supplied by means of parameter download. Up-to-now, the default values were used to work in the runtime environment.

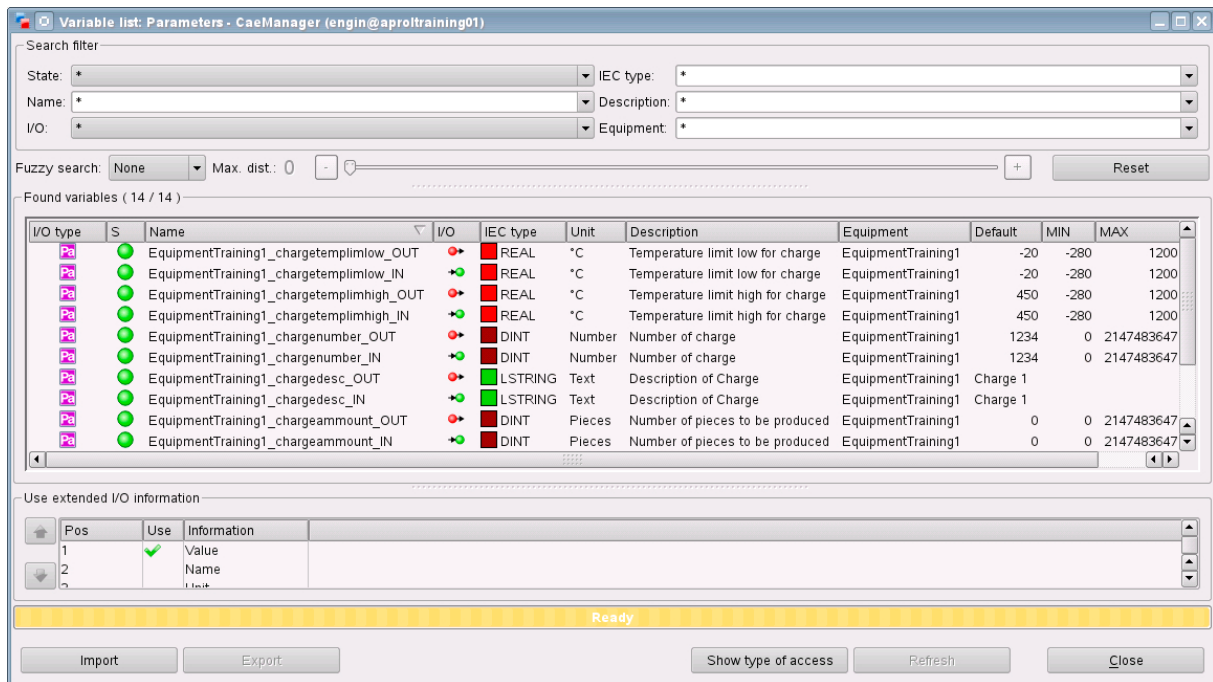


Figure 11: List of the parameter variables

The "**_IN**" and "**_OUT**" variables stand out in the list of parameter variables. This means that the system automatically creates two variables per parameter. The "**_IN**" variable is placed on the input border and is then given a value by the ParameterCenter during a parameter download.

The "**_OUT**" variable is used for a possible parameter upload, whereby the data record which is uploaded is in turn available for a parameter download. This is explained in detail in chapter 'Uploading parameters'.

2.3 Creating parameter templates (plant parts)

In the **plants parts**, the user can combine the equipment variables, which he/she would like to change "simultaneously" in the process control system during runtime.

A plant part is specified by combining parameters from the various pieces of equipment defined earlier into a parameter template. These parameters should share a functional relationship. Parameter sets are created in an additional step. Values are assigned to the "Sets of variables" and saved under various names. Each of these parameter sets can then be loaded "at the push of a button" during the process control system runtime.

This template can be created and configured in the '**Create project part / Parameter-set template**' menu. It is important that logical names correspond to a specific logic or basic function.

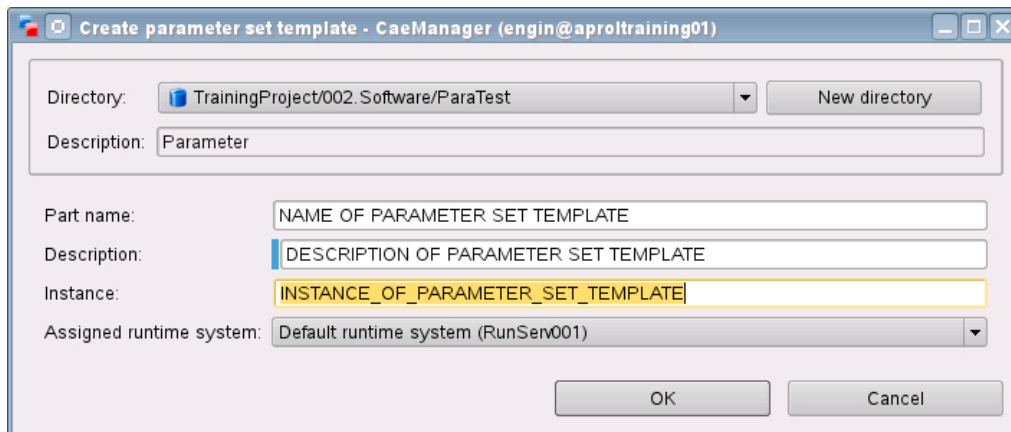


Figure 12: Create a parameter template

After being created, the **parameter template** must then be opened. The context view is opened in the navigation area of the CaeManager, where all pieces of equipment are displayed with the corresponding variables. A parameter template is now put together from pieces of equipment (and equipment variables) per 'drag-and-drop'. It is also possible to switch each individual equipment variable to FIX status. This means that this variable cannot be changed in the parameter set, in the engineering and runtime environments.

This parameter set can be made up of different pieces of equipment: It is therefore also possible to select single variables from the listed equipment.

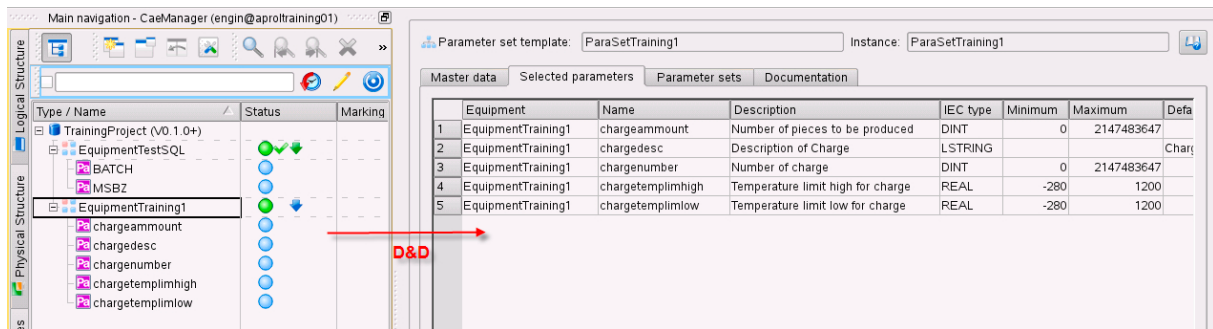


Figure 13: Parameter template

One or more parameter sets can be created in the second view '**Parameter sets**' once all necessary equipment variables have been created in the '**Selected parameters**' view. These parameter sets use the selected parameters as templates. The values for the individual equipment variables are adapted accordingly. The user now has different parameter sets which can also be used in the runtime environment as **Recipes**.

Parameter sets are created using the '**New**' button, which is shown with the right mouse button. The parameter set can be configured after entering a logical name.

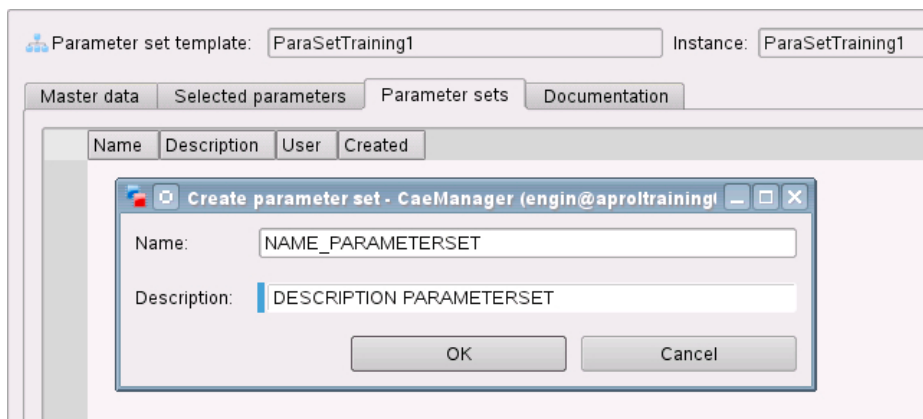


Figure 14: Create a parameter set

By **default**, the individual variables have the value defined for the piece of equipment in the engineering task. These values can now be changed according to the required circumstances. Many different parameter sets can arise in this way and correspond to certain production requirements or batches.

These parameter sets are found in the runtime environment using the ParameterCenter. Of course new sets can also be created in the runtime environment.

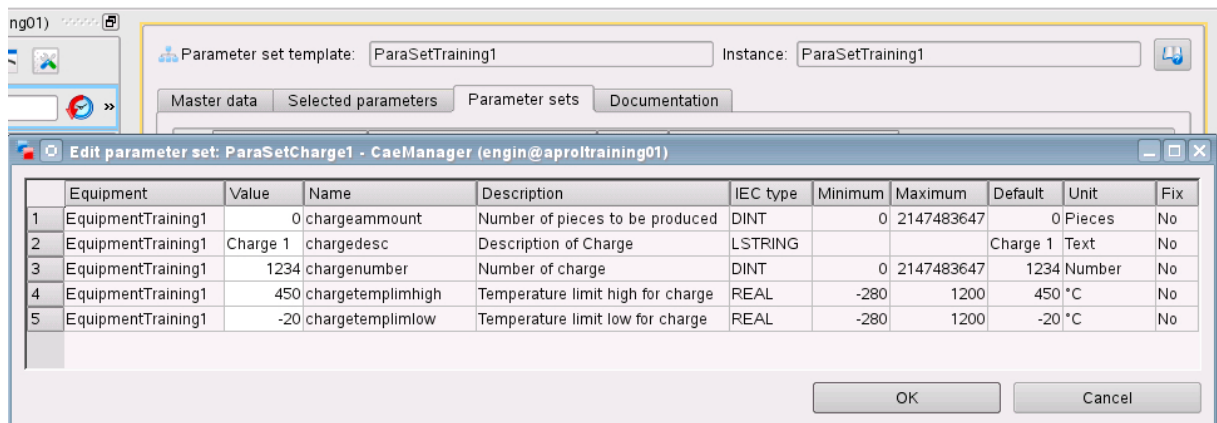


Figure 15: Edit parameter set

After saving, activating and compiling the parameter sets, they are available to be **compiled** and then **downloaded** (build tasks (APROL system), runtime download).

The instance name can also be edited in the first tab of this project part. The documentation is maintained in the last tab. (also see equipment project part)

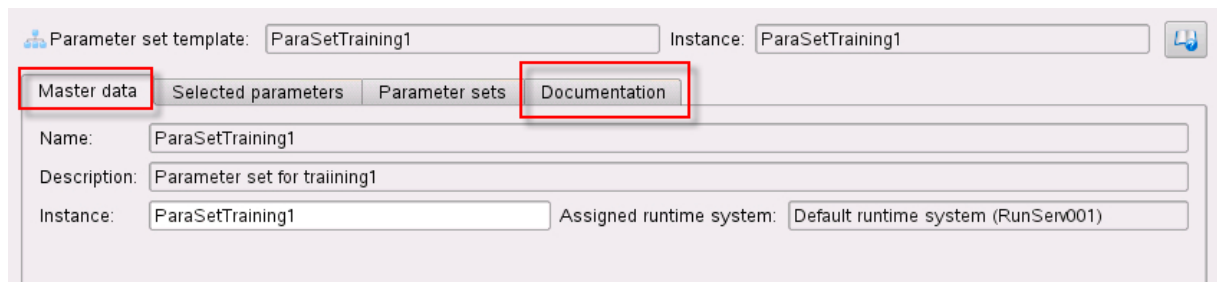


Figure 16: Master data and documentation

2.4 Recipe completed

The configuration involving engineering is now finished.

Let's take another look at the sequence of actions:

- Define and implement equipment piece(s).
- Use the equipment variables in the respective function charts.
- Put parameter templates together from the pieces of equipment.
- Put parameter sets (recipes) together from the parameter templates. This makes it possible to implement multiple recipes with a wide variety of values.
- All created and modified areas must be saved, activated and compiled.
- Perform the build for the affected tasks.
- Download the affected tasks and functions.



Figure 17: Ready for production



Task:

Carry out the management of a batch with the graphic macro which has been supplied. Please realize several different parameter sets in engineering.

3 Parameters in the runtime environment

After being successfully downloaded, the parameter sets created in engineering are available in the **ParameterCenter**. The ParameterCenter takes care of management, expansion (new sets), downloads etc. This is started the same way as other programs in the **DisplayCenter** or using the **KDE menu**.

3.1 The ParameterCenter

The ParameterCenter is designed to function intuitively. After opening the application, all parameter sets are presented in the explorer view. These can be opened for loading, changing or deleting. However, only parameter sets that were created in the runtime environment can be deleted.

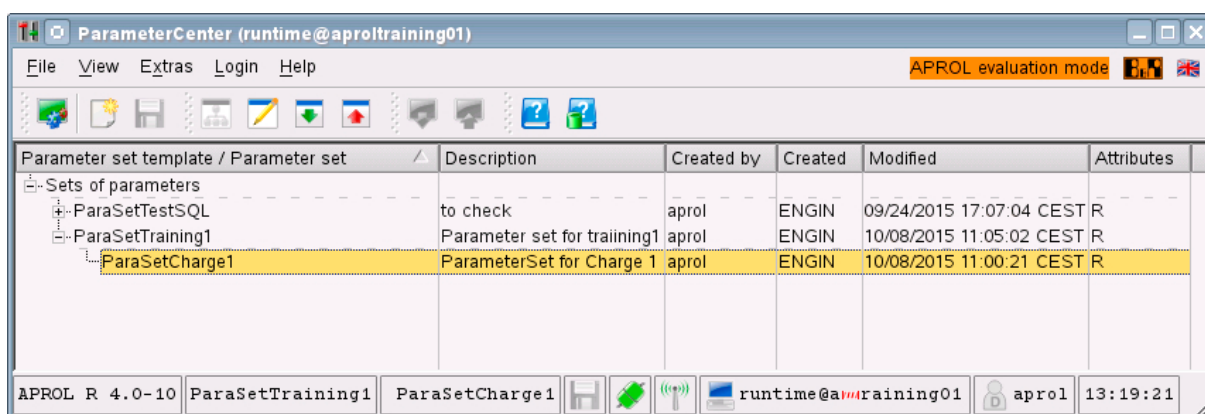


Figure 18: ParameterCenter start page

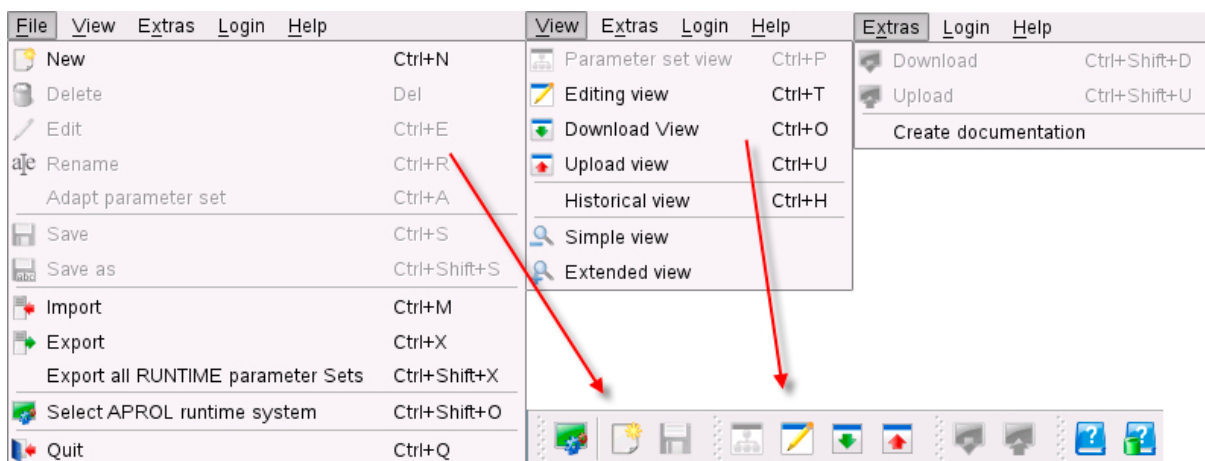


Figure 19: Toolbar in the ParameterCenter

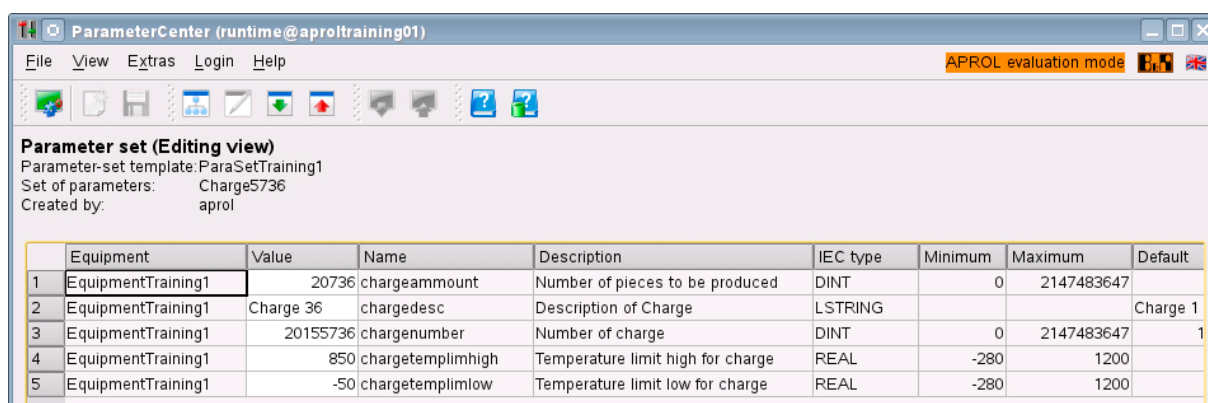
The **ParameterCenter** application allows the operator to select predefined parameter sets from different parameter templates, (i.e. different plant parts) and to **download** the **parameters** to the system. Parameter sets can be edited and saved under a different name if the operator has the corresponding rights.

The **'History'** view allows you to view the parameter downloads that have already taken place. All significant data and, in particular, parameter values are recorded during each download action.

A short description and the name of the operator is displayed for each parameter set. Furthermore, the **'Created'** entry indicates if the set was defined in the engineering task or in the runtime environment (i.e. by the operator). In addition, the date and time of the last modification are listed as well as the read and write attributes. The operator can edit or delete the selected parameter set if he/she has write [W] and read [R] access. If the operator does not have write access then any changes made can only be saved under a new name.

3.2 New recipes

New parameter sets or recipes can be created at any time by editing and changing an existing parameter set. This parameter set (recipe) can now be saved manually ("**File / Save as**" menu) or automatically by exiting this view. These new values are now available at any time.



	Equipment	Value	Name	Description	IEC type	Minimum	Maximum	Default
1	EquipmentTraining1	20736	chargeamount	Number of pieces to be produced	DINT	0	2147483647	
2	EquipmentTraining1	Charge 36	chargedesc	Description of Charge	LSTRING			Charge 1
3	EquipmentTraining1	20155736	chargenumber	Number of charge	DINT	0	2147483647	1
4	EquipmentTraining1	850	chargetemplimhigh	Temperature limit high for charge	REAL	-280	1200	
5	EquipmentTraining1	-50	chargetemplimlow	Temperature limit low for charge	REAL	-280	1200	

Figure 20: New parameter set (recipe)

Generally, several parameter sets must be changed for recipes. This makes it easy to implement a product change (new batch), rinsing process, sterilization, etc.

3.3 Action logging

Each download of a parameter set (recipe) is automatically recorded by the **APROL** action logging without requiring additional engineering.

AuditTrail report - Mozilla Firefox <2>

AuditTrail report

https://aproltraining01.br-automation.co.at/PROJECTS/TrainingProject001/standard/action/parameterdetail.ch?date=08.10.2015&from=13.54.22.231&until=13.54.22.231&id=PARAMETER.*;action=parameter*;aj

APROL

Web Portal

AuditTrail report (Reduced view) - Filtered

Parameter download details

(Back to overview)

Template	Parameter set	Operator	Surname	Firstname	Operator terminal	Server	Project	Runtime name	
08.10.2015 13:54:22ParaSetTraining1	Charge5736	aprol	unknown	unknown	aproltraining01.br-automation.co.at:2.0	aproltraining01	TrainingProject	runtime	
Equipment	Name	Old value	New value	Type	Description	Min	Max	Default	PV
			show changed only						
EquipmentTraining1	chargeamount	20736	50000	DINT	Number of pieces to be produced	0	2147483647	0	EquipmentTraining1_chargeam
EquipmentTraining1	chargedesc	Charge 36	Charge 2015_36	LSTRING	Description of Charge			Charge 1	EquipmentTraining1_charged
EquipmentTraining1	chargenumber	20155736	573620736	DINT	Number of charge	0	2147483647	1234	EquipmentTraining1_chargenu
EquipmentTraining1	chargetemplimhigh	850	150	REAL	Temperature limit high for charge	-280	1200	450	EquipmentTraining1_charger
EquipmentTraining1	chargetemplimlow	-50	-10	REAL	Temperature limit low for charge	-280	1200	-20	EquipmentTraining1_charger

Created at:

08.10.2015 13:55:22

Created on:

aproltraining01

Report:

AuditTrail report (Reduced view)

Template:

/PROJECTS/TrainingProject001/standard/action/parameterdetail.ch

File:

Version: 2.3 APROL R 4.0-10 (64Bit)

File:

parametertype: parameterid,

Figure 21: AuditTrail for parameters and recipes

In addition to the download action, the audit trail also records **who downloaded the parameter set, when it was downloaded, where it was downloaded from and which parameter set was downloaded with what content** (values). This is required, for example, to implement a record for a group of actions. This allows APROL to guarantee a seamless record when handling recipes (all other actions (process graphic) are also recorded (see SEM841.5)).

3.4 Recipes are active

The recipes or parameter sets are now loaded and active. According to these specifications, the system performs all related processes. The ParameterCenter also provides further options and status information. To learn more about this, please refer to the APROL documentation.



Figure 22: The product is being made



Task:

Check your engineering work in the runtime environment. Make changes to and save the existing parameter sets. Use the changed parameter sets in the runtime environment. Use the AuditTrail to check what the system has recorded.

4 Uploading parameters

APROL has the possibility to upload parameters which have been eventually changed in the plant. It is important to use the **"_OUT"** parameter variables for this.

If parameters in the logic can be changed by a remote connection, one normally wants to upload these changes as a new parameter set into the runtime or operating environment, in order to be able to edit and use them further.

It does not matter in which way the parameters are written. Scenarios such as VC applications, Watch, input via DisplayCenter, and also external sources (connection) can be used.

It is now necessary to use and connect the corresponding **"_OUT"** variables in engineering.

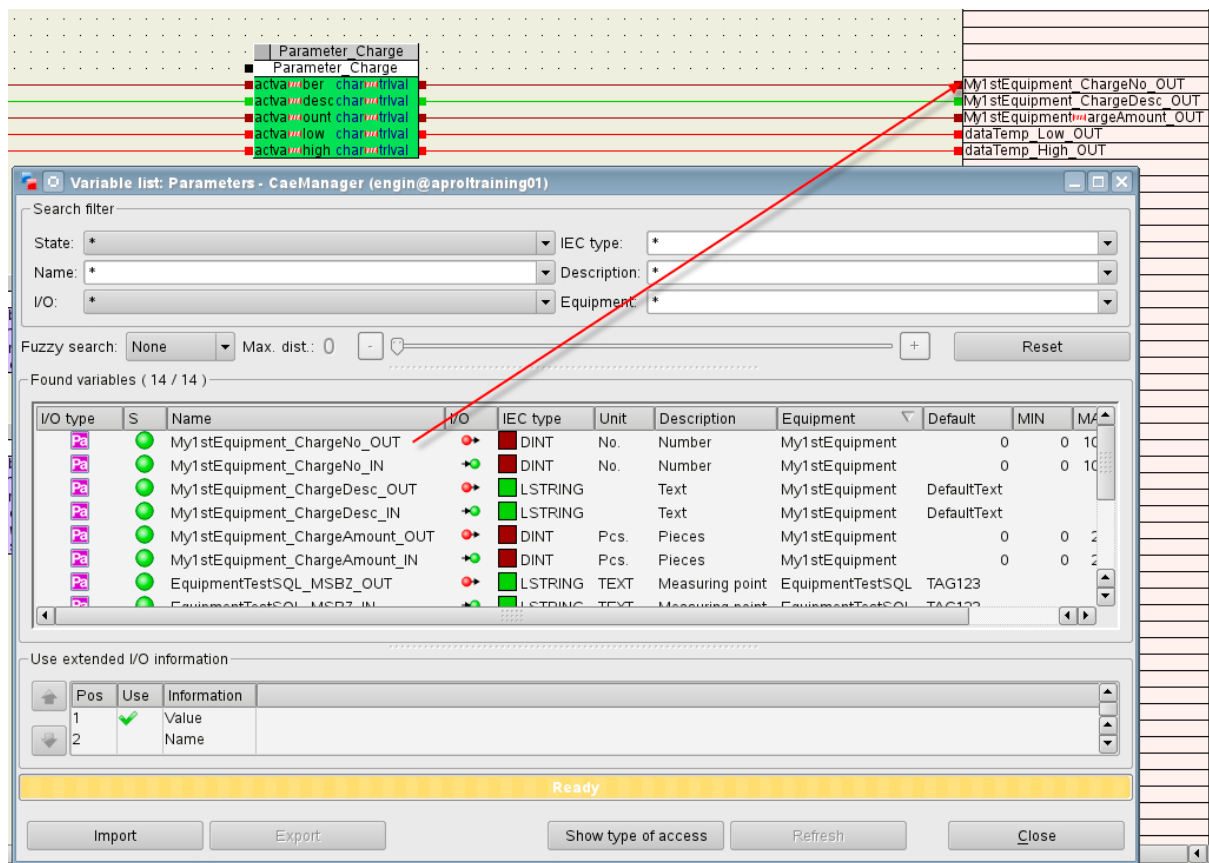


Figure 23: Upload parameters in engineering

After the known actions (compile, build, download), the parameters can be written in the runtime and operating environment and then uploaded.

4.1 Parameter upload in the runtime environment

The view of the currently loaded parameters in the runtime and operating environment can be seen in the left column, and the manual entry in the right column. This view is only as illustration.

	IN	OUT
Charge number:	123456	6789
Charge description:	Secret_Stuff	77777
Charge amount:	1500	7777
Charge temp limit low:	20.0000	77.0000
Charge temp limit high:	300.0000	700.0000

Figure 24: Illustration of parameter upload and download

If the ParameterCenter is now opened in the upload view, the different values of the "_IN" and "_OUT" variables can be seen. Depending on the logic used, changed parameter values may already be active.

Parameter upload
 Parameter-set template: ParaSetCharge
 Set of parameters: ParaSetVitaminB
 Created by: Martin

Equipment	Name	PV	Current value	Parameter value	Variance Comparison	Read	Description
dataTemp	High	dataTemp_High	700	300			H
dataTemp	Low	dataTemp_Low	77	20			L
My1stEquipment	ChargeAmount	My1stEquipment_ChargeAmount	7777	1500			Pieces
My1stEquipment	ChargeDesc	My1stEquipment_ChargeDesc	77777	Secret_Stuff			Text
My1stEquipment	ChargeNo	My1stEquipment_ChargeNo	6789	123456			Number

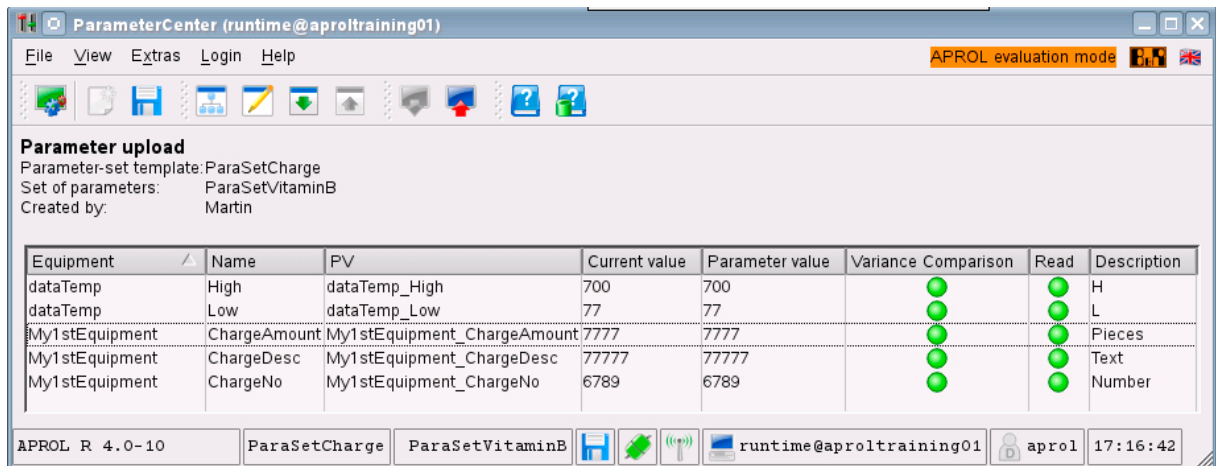
APROL R 4.0-10 ParaSetCharge ParaSetVitaminB runtime@aproltraining01 apol 17:02:40

Charge number:	123456	6789
Charge description:	Secret_Stuff	77777
Charge amount:	1500	7777
Charge temp limit low:	20.0000	77.0000
Charge temp limit high:	300.0000	700.0000

Figure 25: First step parameter upload

The changed parameter set can now be uploaded and saved under a new name. If the outgoing parameter set originates in the engineering, a new name must be assigned for saving. If a data set from the runtime is already active, it can be overwritten. After the upload, the set-actual comparison is OK and is signaled with green circle icons.

Uploading parameters



ParameterCenter (runtime@aproltraining01)

File View Extras Login Help

APROL evaluation mode

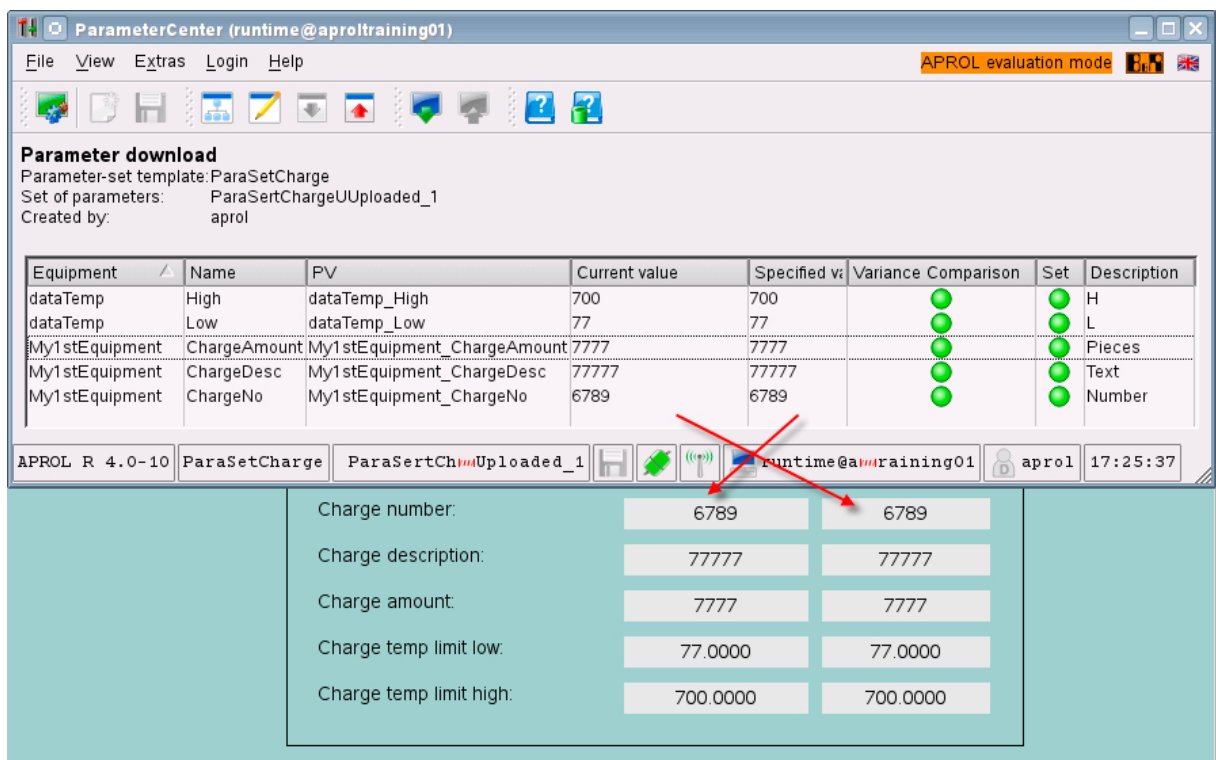
Parameter upload
Parameter-set template: ParaSetCharge
Set of parameters: ParaSetVitaminB
Created by: Martin

Equipment	Name	PV	Current value	Parameter value	Variance Comparison	Read	Description
dataTemp	High	dataTemp_High	700	700	●	●	H
dataTemp	Low	dataTemp_Low	77	77	●	●	L
My1stEquipment	ChargeAmount	My1stEquipment_ChargeAmount	7777	7777	●	●	Pieces
My1stEquipment	ChargeDesc	My1stEquipment_ChargeDesc	77777	77777	●	●	Text
My1stEquipment	ChargeNo	My1stEquipment_ChargeNo	6789	6789	●	●	Number

APROL R 4.0-10 ParaSetCharge ParaSetVitaminB runtime@aproltraining01 aprol 17:16:42

Figure 26: Applicatory changed recipe uploaded

If you now open the saved data set, you can of course use it and load it again at any time.



ParameterCenter (runtime@aproltraining01)

File View Extras Login Help

APROL evaluation mode

Parameter download
Parameter-set template: ParaSetCharge
Set of parameters: ParaSertChargeUUploaded_1
Created by: aprol

Equipment	Name	PV	Current value	Specified v	Variance Comparison	Set	Description
dataTemp	High	dataTemp_High	700	700	●	●	H
dataTemp	Low	dataTemp_Low	77	77	●	●	L
My1stEquipment	ChargeAmount	My1stEquipment_ChargeAmount	7777	7777	●	●	Pieces
My1stEquipment	ChargeDesc	My1stEquipment_ChargeDesc	77777	77777	●	●	Text
My1stEquipment	ChargeNo	My1stEquipment_ChargeNo	6789	6789	●	●	Number

APROL R 4.0-10 ParaSetCharge ParaSertChargeUUploaded_1 runtime@aproltraining01 aprol 17:25:37

Charge number: 6789 6789

Charge description: 77777 77777

Charge amount: 7777 7777

Charge temp limit low: 77.0000 77.0000

Charge temp limit high: 700.0000 700.0000

Figure 27: Downloading an applicatory changed recipe

By using this method, the ParameterCenter is not the only way to create new data sets, which can now be realized using the logic. Another example could be loading a recipe and refining it. It makes common sense to secure this "know-how" and upload it in order to be able to reuse it.

**Task:**

The respective outputs are also available if a graphic macro is used. Connect them with the corresponding "_OUT" variables. Complete your engineering duties and check the upload function in the runtime environment.

5 Exporting and importing parameter sets

Parameter sets can be exported and therefore also imported. It is also possible to import this data into the CaeManager. There are two principle functions available:

Action	Description
Export	For exporting single parameter sets. The parameter set which is selected in the list is exported.
Export all runtime parameter sets	For exporting all of the parameter sets which were created in the runtime and operator environments. It is thus possible to carry out an additional backup of all changes and imports at any time.

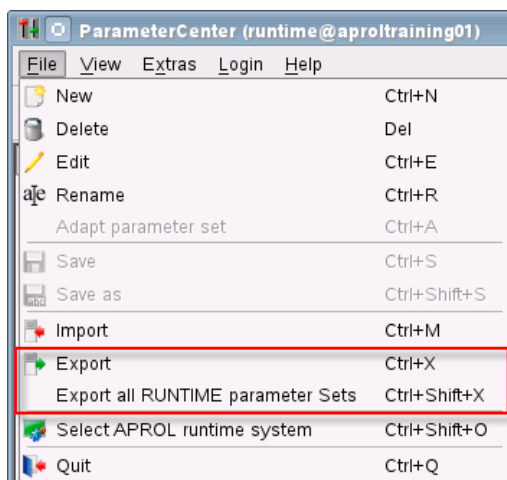


Figure 28: "Export" call of parameter sets

The result is an XML file, which is in turn available for an import. This XML file can of course be adapted and so creates new parameter sets and recipes.



Task:

Carry out an export of one parameter set and import it into the engineering.

Change the data set in the export file (XML) and import the new parameter set into the runtime environment.

6 Documentation of the parameter sets and recipes

It is necessary to document the parameter sets and recipes automatically, because they may have different sources (engineering, runtime, upload, MySQL). This is an integrated part of the ParameterCenter and can be opened with the **"Extras / Create Documentation"** menu.

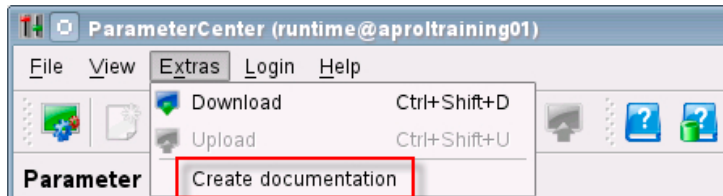


Figure 29: "Create documentation" call

The documentation which is to be created is selected with the menu which opens (all or certain (current) parameter set), and the type (HTML or CSV).

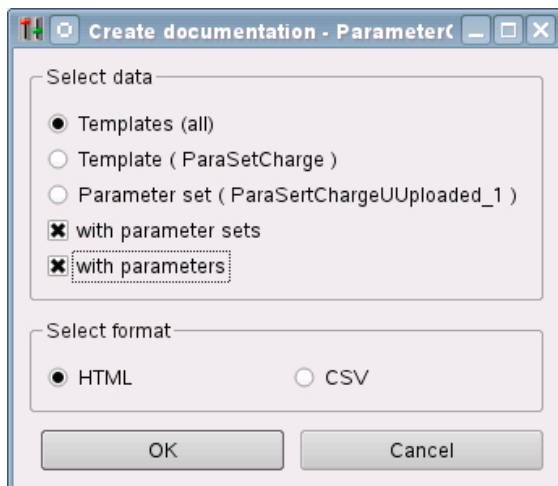


Figure 30: Data selection for parameter documentation

The result is an HTML or CSV document which can be used for further editing (PDF signing, editing in Excel, etc.).

Documentation of the parameter sets and recipes

ParameterManagement documentation - Mozilla Firefox

ParameterManagement document... X

file:///home/runtime/RUNTIME/Cnt/ParameterCenter/html/pd1.html

Parameter Management **APROL**

ParameterManagement documentation from 29.10.2015 09:22:48

Operator: aprol Host: aproltraining01 CC-Account: runtime

Parameter set template

Instance	Name	Description	Operator	Modified
ParaSetCharge	ParaSetCharge	ParaSetCharge	Martin	10/22/2015 16:44:45 CEST

Parameter set

Name	Description	Created	Operator	Modified
Produkt	Produkt	RUNTIME	aprol	10/22/2015 15:09:28 CEST

Parameter

Name	Value	Equipment instance	Equipment name	Description	Default value	Minimum	Maximum	IEC type	Unit	Fix	Deleted
ChargeAmount	567	My1stEquipment	My1stEquipment	Pieces	0	0	200000	DINT	Pcs.	No	No
ChargeDesc	Did it work???	My1stEquipment	My1stEquipment	Text	DefaultText			LSTRING		No	No
ChargeNo	3453	My1stEquipment	My1stEquipment	Number	0	0	1000000	DINT	No.	No	No
High	1000	dataTemp	dataTemp	H	300	273	6000	REAL	°C	No	No
Low	545	dataTemp	dataTemp	L	20	-273	1600	REAL	°C	No	No

Name Description Created Operator Modified

New RUNTIME aprol 10/22/2015 15:10:38 CEST

Parameter

Name	Value	Equipment instance	Equipment name	Description	Default value	Minimum	Maximum	IEC type	Unit	Fix	Deleted
ChargeAmount	676	My1stEquipment	My1stEquipment	Pieces	0	0	200000	DINT	Pcs.	No	No
ChargeDesc	dtziedzj	My1stEquipment	My1stEquipment	Text	DefaultText			LSTRING		No	No
ChargeNo	1111	My1stEquipment	My1stEquipment	Number	0	0	1000000	DINT	No.	No	No
High	1000	dataTemp	dataTemp	H	300	273	6000	REAL	°C	No	No
Low	545	dataTemp	dataTemp	L	20	-273	1600	REAL	°C	No	No

Name Description Created Operator Modified

Figure 31: Automatically created documentation

7 Handling parameters from the logic

APROL contains function blocks that can be used to upload and download recipes with the program logic. As a result, it is no longer necessary to manually intervene with the ParameterCenter. The trigger for the download can be completely variable and come from the logic for inputs, operating templates, etc.

The two most important blocks "**AprUcParamDown**" and "**AprUcParamUp**" are located in the "APROL" library in the parameter management directory. The other available blocks are for management and control.

APROL (V2.3.0)			APROL standard blocks
Alarm			Observation blocks for Process Control System
AnalogValueProcessing			Analog value processing blocks
_APROL_Internals			APROL internal group
BitFunctions			Bit orientated blocks
CommonLibs			Local common libraries
CompressedRecord			Recording compressed values
ControlComputerFunctions			Special process control system blocks
ControllerFunctions			Special PLC blocks
Conversions			Data packer/unpacker
EventDetection			Event recognition
GlobalVariables			Global Variables
MemoryFunctions			Memory operating blocks
Multimedia			Multimedia function of control computer
ParameterManagement			Parameterset management
AprUcCheckParam	●	👤	Checks the existence of a parameter set
AprUcListParam	●	👤	List all parameter sets
AprUcParamDel	●	👤	Delete parameter sets
AprUcParamDown	●	👤	Start parameter set download
AprUcParamUp	●	👤	Start parameter set upload
Reports			Standard report functions

Figure 32: Block for managing parameters from the logic

Handling parameters from the logic

7.1 Recipe download using the logic

The **"AprUcParamDown"** block is used to download a defined recipe.

If a positive edge is placed on the UcbTrigger input, the block starts the ParameterCenter in the background without graphical interface, which then performs the download.

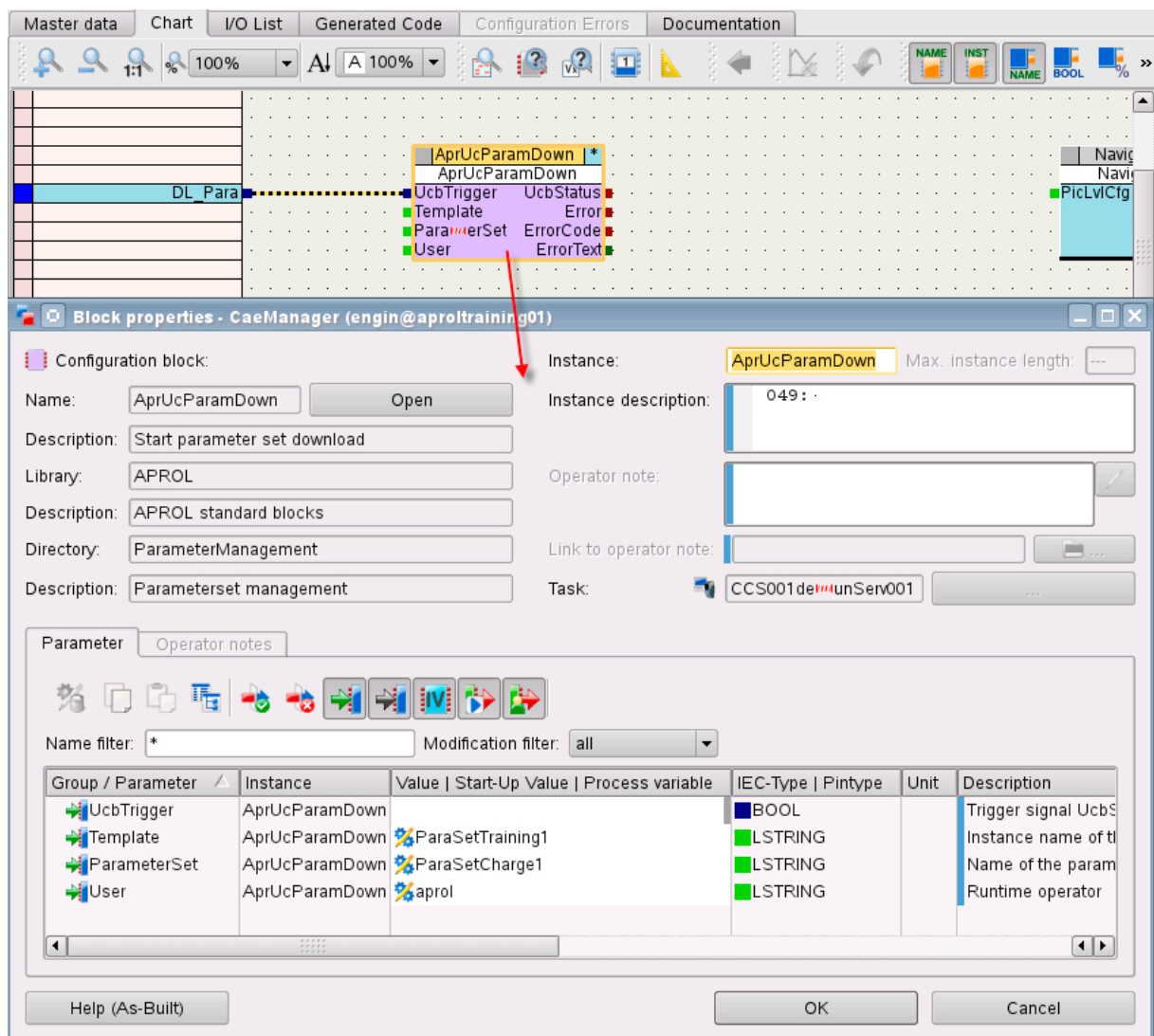


Figure 33: Block for automatic parameter download

7.2 Recipe upload using the logic

The **"AprUcParamUp"** block is used to upload a defined recipe.

If a positive edge is placed on the UcbTrigger input, the block starts the ParameterCenter in the background without graphical interface, which then performs the upload.

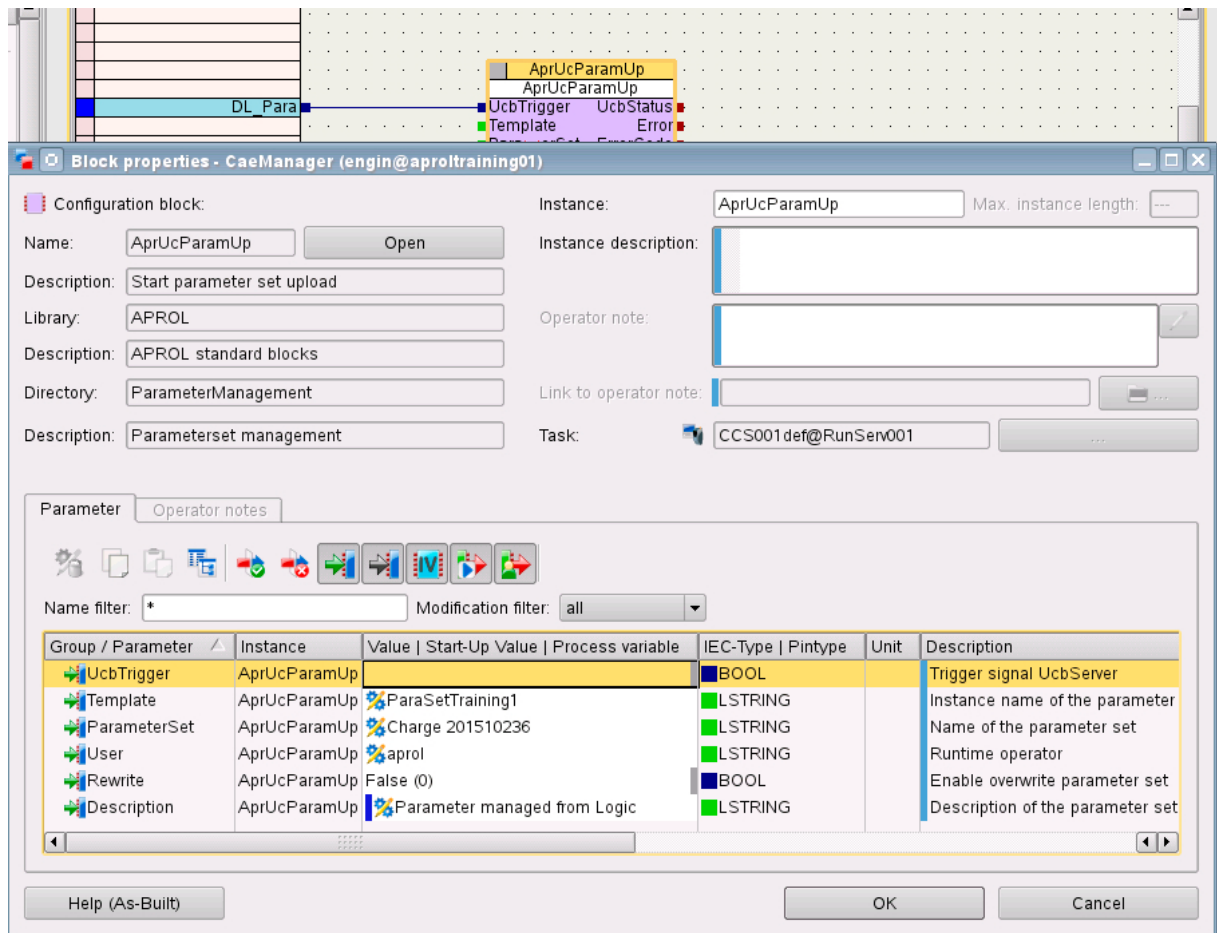


Figure 34: Block for automatic parameter upload

7.3 Implementing automatic parameter handling

These blocks can be used to implement automatic parameter handling. It could also be possible that the parameter set is variable. Therefore, completely new parameter sets could automatically be downloaded from higher-level systems. Of course, double confirmation is required in a chemical plant (two-man rule). The upload is very useful when parameters can be changed in several locations. This makes it possible at any time to save and starting using a "functioning" parameter set again.

**Task:**

In this exercise, you should choose whether recipe 1 or recipe 2 will be loaded by the programming logic. This is easily done by placing and connecting two download blocks. The trigger for the download should come from a digital input. Additionally, changed parameters should be uploaded to a new recipe using the upload block.

**Task:**

A suitable operating template should be created for the download blocks placed in the previous exercise. You should be able to easily enter the parameters required by the download block such as template and parameter set. The Ucb trigger should be set on the template using a button.

8 Creating external parameter sets

The ParameterCenter data is stored in a MySQL database. Handling this data and creating new data records therefore requires a knowledge of SQL. Simple applications and methods are described in the following example.

Database access:

In order to access the ParameterCenter database, log into the runtime server and start the MySQL client in the console with the following command:

```
mysql -u ext_runtime -p.extern
```

The MySQL database is then opened. All results are shown in the mysql prompt.

```
mysql>
```

Switch to the created runtime system database with the command "USE runtime;". If it has another name, e.g. runtime2, then use the command 'USE runtime2;'.

```
mysql> USE runtime
Database changed
mysql>
```

8.1 Creating parameter set description

The parameter set description is mandatory and must be entered first. This can only exist once (!) and is assigned to each data set as default.

You can view the table with the following command:

```
mysql> show columns from DESCRIPTION_EXT;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| D_ID | int(11) | NO | PRI | NULL | auto_increment |
| LangCode | int(11) | NO | PRI | 0 | |
| DescText | text | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01sec)
```

The following command is used to create a description for the new parameter set. This must be repeated for other languages. (ATTENTION: There is a line break in the example in order to show the entire command. This is not allowed to be entered)

```
mysql> INSERT INTO DESCRIPTION_EXT (LangCode,DescText)
```

Creating external parameter sets

```
VALUES (049,"Beschreibungstext Deutsch");
```

Ergebnis:

Query OK, 1 row affected (0.01 sec)

The ID is assigned automatically and output by the "**select * FROM DESCRIPTION_EXT;**" command. The **REPLACE** command must be used instead of the **INSERT** command to change the description.

D_ID	ID of the data record
LangCode	Language code for Germany 049
DescText	Description text

```
mysql> select * FROM DESCRIPTION_EXT;
+-----+-----+-----+
| D_ID | LangCode | DescText |
+-----+-----+-----+
| 1 | 49 | Beschreibungstext Deutsch |
+-----+-----+-----+
1 row in set (0.00 sec)
```

A new parameter set can now be created with the „D_ID=1“.

8.2 Create parameter set

The **PARAMSET_EXT** table is used to create a new parameter set. The following parameters are necessary for the creation:

```
mysql> SHOW COLUMNS FROM PARAMSET_EXT;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| PS_Name | varchar(255) | NO | PRI | | |
| T_Instance | varchar(255) | NO | PRI | | |
| User | varchar(255) | YES | | NULL | |
| Modified | bigint(20) | NO | | 0 | |
| D_ID | int(11) | YES | MUL | NULL | |
| Confirmed | tinyint(4) | NO | | 0 | |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

PS_Name	(parameter name) (free selection)
T_Instance	(parameter set template from the CaeManager engineering) (fix)

User	(user name who created the parameter set) (free selection)
modified	(UNIX timestamp of the creation/modification) (free selection)
D_ID	(ID of the data record (see chapter 'Creating parameter set description'))
Confirmed	(not relevant default = 0)

E.g. the parameter set „ParaSetVitaminD“ is created with the following command. Enter switches to the next line, because a ";" completes the command set.

```
mysql>
INSERT INTO PARAMSET_EXT (PS_Name,T_Instance,User,Modified,D_ID,Confirmed)
VALUES ('ParaSetVitaminD','ParaSetCharge','externSQL',100000000,1,0);

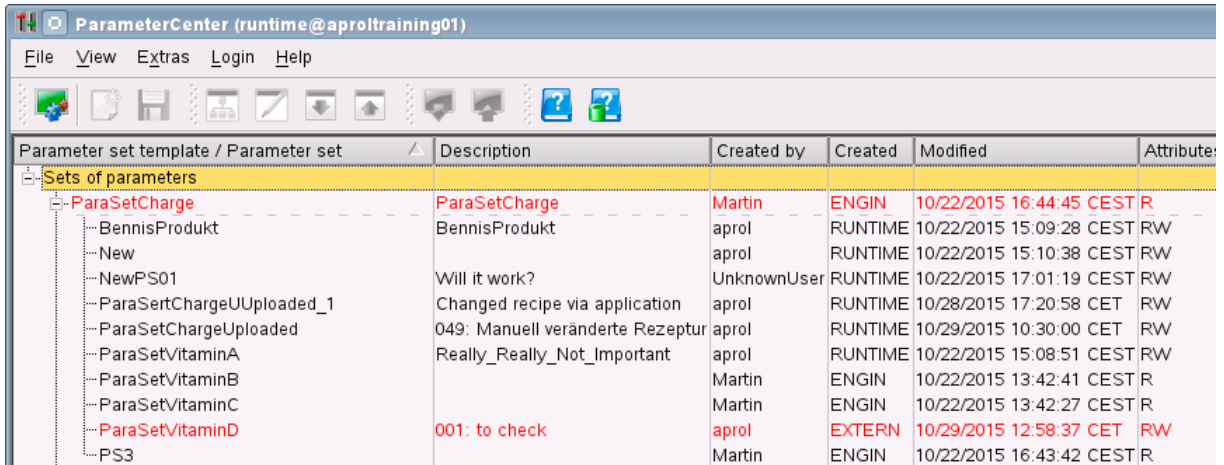
Ergebnis:
Query OK, 1 row affected (0,01 sec)
```

Use the **"SELECT"** command to find out which T_Instance (parameter set template) exists.

```
mysql> SELECT * FROM TEMPLATE;
+-----+-----+-----+-----+-----+-----+
| T_Instance | T_Name | AprotUser | Modified | D_ID | Persist |
+-----+-----+-----+-----+-----+-----+
| ParaSetCharge | ParaSetCharge | Martin | 1445525085 | 327 | 0 |
| ParaSetTestSQL | ParaSetTestSQL | aprot | 1443107224 | 8 | 0 |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

A parameter set „ParaSetVitaminD" is now visible in the „ParaSetCharge" parameter set template in the ParameterCenter. This is displayed in red, because there is no data for the individual equipment variables.

Creating external parameter sets



The screenshot shows the 'ParameterCenter' application window. On the left is a tree view under 'Sets of parameters' containing 'ParaSetCharge' and its sub-items: 'BennisProdukt', 'New', 'NewPS01', 'ParaSetChargeUUloaded_1', 'ParaSetChargeUploaded', 'ParaSetVitaminA', 'ParaSetVitaminB', 'ParaSetVitaminC', 'ParaSetVitaminD', and 'PS3'. The main table displays the details for these parameter sets.

Parameter set template / Parameter set	Description	Created by	Created	Modified	Attribute
ParaSetCharge	ParaSetCharge	Martin	ENGIN	10/22/2015 16:44:45 CEST	R
BennisProdukt	BennisProdukt	aprol	RUNTIME	10/22/2015 15:09:28 CEST	RW
New		aprol	RUNTIME	10/22/2015 15:10:38 CEST	RW
NewPS01	Will it work?	UnknownUser	RUNTIME	10/22/2015 17:01:19 CEST	RW
ParaSetChargeUUloaded_1	Changed recipe via application	aprol	RUNTIME	10/28/2015 17:20:58 CET	RW
ParaSetChargeUploaded	049: Manuell veränderte Rezeptur	aprol	RUNTIME	10/29/2015 10:30:00 CET	RW
ParaSetVitaminA	Really_Really_Not_Important	aprol	RUNTIME	10/22/2015 15:08:51 CEST	RW
ParaSetVitaminB		Martin	ENGIN	10/22/2015 13:42:41 CEST	R
ParaSetVitaminC		Martin	ENGIN	10/22/2015 13:42:27 CEST	R
ParaSetVitaminD	001: to check	aprol	EXTERN	10/29/2015 12:58:37 CET	RW
PS3		Martin	ENGIN	10/22/2015 16:43:42 CEST	R

Figure 35: Parameter set originating from SQL

8.3 Create parameters for the parameter set

The parameters can vary in contrast to the parameter set and are defined in the CaeManager by the parameters set template. There are 5 parameters in the parameter set in this example. (ChargeAmount, ChargeDesc, ChargeNo, High, Low)

Here is the listing of all parameters:

```
mysql> select * FROM PARAMSET_PARAMS;
```

T_Instance	PS_Name	P_Name	E_Instance	Value	Fix
ParaSetCharge	Produkt	ChargeAmount	MylstEquipment	567	0
ParaSetCharge	Produkt	ChargeDesc	MylstEquipment	Did it work??	0
ParaSetCharge	Produkt	ChargeNo	MylstEquipment	3453	0
ParaSetCharge	Produkt	High	dataTemp	1000	0
ParaSetCharge	Produkt	Low	dataTemp	545	0
ParaSetCharge	New	ChargeAmount	MylstEquipment	676	0
ParaSetCharge	New	ChargeDesc	MylstEquipment	dtztetdzj	0
ParaSetCharge	New	ChargeNo	MylstEquipment	1111	0
ParaSetCharge	New	High	dataTemp	1000	0
ParaSetCharge	New	Low	dataTemp	545	0
ParaSetCharge	NewPS01	ChargeAmount	MylstEquipment	7777	0
ParaSetCharge	NewPS01	ChargeDesc	MylstEquipment	77777	0
ParaSetCharge	NewPS01	ChargeNo	MylstEquipment	6789	0
ParaSetCharge	NewPS01	High	dataTemp	700	0
ParaSetCharge	NewPS01	Low	dataTemp	77	0
....					

T_Instance is the instance of the parameter definition in the CaeManager. The following command is used to create the parameters for the „ParaSetVitaminD“ parameter set with the corresponding values.

```
mysql> REPLACE INTO PARAMSET_PARAMS_EXT
(T_Instance,PS_Name,P_Name,E_Instance,Value) VALUES
('ParaSetCharge','ParaSetVitaminD','ChargeAmount','My1stEquipment',9999),
('ParaSetCharge','ParaSetVitaminD','ChargeDesc','My1stEquipment',
"My Description from SQL"),
('ParaSetCharge','ParaSetVitaminD','ChargeNo','My1stEquipment',98765),
('ParaSetCharge','ParaSetVitaminD','High','dataTemp',837),
('ParaSetCharge','ParaSetVitaminD','Low','dataTemp',-23);
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
```



Note:

Enter always also switches to the next line here. A ";" at the end completes the command.

The parameters have now been created in the parameter set and the data set is no longer displayed in red in the ParameterCenter. If the new parameter set is opened, the pre-defined values can be found there.

ParameterCenter (runtime@aproltraining01)

File View Extras Login Help

APROL evaluation mode

Parameter set (Editing view)

Parameter-set template: ParaSetCharge

Set of parameters: ParaSetVitaminD

Created by: aprot

	Equipment	Value	Name	Description	IEC type	Minimum	Maximum	Default	Unit	Fix
1	My1stEquipment	9999	ChargeAmount	Pieces	DINT	0	200000		0 Pcs.	No
2	My1stEquipment	My Description from SQL	ChargeDesc	Text	LSTRING			DefaultText		No
3	My1stEquipment	98765	ChargeNo	Number	DINT	0	1000000		0 No.	No
4	dataTemp	837	High	H	REAL	273	6000	300	°C	No
5	dataTemp	-23	Low	L	REAL	-273	1600	20	°C	No

Figure 36: Parameter set created via MySQL

8.4 Deleting data sets

In order to delete data sets from the MySQL database, the order "Delete parameter" and "Delete parameter set" must be adhered to.

Parameters are deleted with the "DELETE FROM .." command.

```
mysql> DELETE FROM PARAMSET_PARAMS_EXT WHERE PS_Name='ParaSetVitaminD';
```

Mit diesem Befehl werden alle Parameter im Parametersatz „ParaSetVitaminD“ gelöscht.

```
mysql> DELETE FROM PARAMSET_EXT WHERE PS_Name='ParaSetVitaminD';
```

Creating external parameter sets

Mit diesem Befehl wird nun der Eintrag "ParaSetVitaminD'" gelöscht.

The parameter set with the name "ParaSetVitaminD" is deleted with this command. This recipe is then no longer available in the ParameterCenter.

The parameter handling over MySQL is finished. Further details can be found in the APROL product documentation.



Task:

Carry out the creation of a new parameter set, step-for-step, as described in the chapter "Creating external parameter sets" and check the results in the ParameterCenter.

9 Summary

What have we achieved so far?

We can now correctly organize and manage the process of engineering pieces of equipment that will be used in parameter templates, which are then used for recipes. Furthermore, there are absolutely no limitations to operation since new recipes can be created and activated at any time. Of course, this is also includes a thorough record of activities.



Figure 37: Several different products created successfully

10 Appendix

10.1 Objectives checklist (a few on-topic questions)

What is a piece of equipment?

What is a parameter template?

Does working with the ParameterCenter seem familiar?

Is the automatic parameter download familiar?

10.2 Responses to the objectives checklist

Please take down any necessary additional information.

Seminars and training modules

The Automation Academy provides targeted training courses for our customers as well as our own employees.

At the Automation Academy, you'll develop the skills you need in no time!

Our seminars make it possible for you to improve your knowledge in the field of automation engineering.

Once completed, you will be in a position to implement efficient automation solutions using B&R technology. This will make it possible for you to secure a decisive competitive edge by allowing you and your company to react faster to constantly changing market demands.



Automation Studio seminars and training modules

Programming and configuration	Diagnostics and service
SEM210 – Basics SEM246 – IEC 61131-3 programming language ST* SEM250 – Memory management and data storage SEM410 – Integrated motion control* SEM441 – Motion control: Electronic gears and cams** SEM480 – Hydraulics** SEM1110 – Axis groups and path-controlled movements** SEM510 – Integrated safety technology* SEM540 – Safe motion control*** SEM610 – Integrated visualization*	SEM920 – Diagnostics and service for end users SEM920 – Diagnostics and service with Automation Studio SEM950 – POWERLINK configuration and diagnostics* If you don't happen to find a seminar on our website that suits your needs, keep in mind that we also offer customized seminars that we can set up in coordination with your sales representatives: SEM099 – Individual training day Please visit our website for more information****. ****: www.br-automation.com/academy

Overview of training modules

TM210 – Working with Automation Studio TM213 – Automation Runtime TM223 – Automation Studio Diagnostics TM230 – Structured Software Development TM240 – Ladder Diagram (LD) TM241 – Function Block Diagram (FBD) TM242 – Sequential Function Chart (SFC) TM246 – Structured Text (ST) TM250 – Memory Management and Data Storage TM400 – Introduction to Motion Control TM410 – Working with Integrated Motion Control TM440 – Motion Control: Basic Functions TM441 – Motion control: Electronic gears and cams TM1110 – Integrated Motion Control (Axis Groups) TM1111 – Integrated Motion Control (Path Controlled Movements) TM450 – Motion Control Concept and Configuration TM460 – Initial Commissioning of Motors TM500 – Introduction to Integrated Safety TM510 – Working with SafeDESIGNER TM540 – Integrated Safe Motion Control	TM600 – Introduction to Visualization TM610 – Working with Integrated Visualization TM630 – Visualization Programming Guide TM640 – Alarm System, Trends and Diagnostics TM670 – Advanced Visual Components TM920 – Diagnostics and service TM923 – Diagnostics and Service with Automation Studio TM950 – POWERLINK Configuration and Diagnostics TM280 – Condition Monitoring for Vibration Measurement TM480 – The Basics of Hydraulics TM481 – Valve-based Hydraulic Drives TM482 – Hydraulic Servo Pump Drives TM490 – Printing Machine Technology In addition to a printed version, our training modules are also available on our website for download as electronic documents (login required): Visit our website for more information: www.br-automation.com/academy
---	---

Process control seminars and training modules

Process control standard seminars	Process control training modules
SEM841 – Process Control Training: Basic 1 SEM842 – Process Control Training: Basic 2 SEM890 – Advanced Process Control Solutions	TM800 – APROL System Concept TM811 – APROL Runtime System TM812 – APROL Operator Management TM813 – APROL XML Queries and Audit Trail TM830 – APROL Project Engineering TM890 – The Basics of LINUX Visit our website for more information: www.br-automation.com/academy

* SEM210 - Basics is a prerequisite for this seminar.

** SEM410 - Integrated motion control is a prerequisite for this seminar.

*** SEM410 - Integrated motion control and SEM510 - Integrated safety technology are prerequisites for this seminar.

****Our seminars are listed in the Academy\Seminars area of the website.

*****Seminar titles may vary by country. Not all seminars are available in every country.



TM840TRE.30-ENG

V1.2 ©2016/01/07 by B&R. All rights reserved.
All registered trademarks are the property of their respective owners.
We reserve the right to make technical changes.