SIEMENS

SIMATIC

Process Control System PCS 7
PCS 7 BRAUMAT Library V7.1

User Manual

Table of Contents

This manual has the order number:

04/2012
A5E03447308-02

Safety Guidelines

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring to property damage only have no safety alert symbol. The notices shown below are graded according to the degree of danger.

Danger
indicates that death or severe personal injury **will** result if proper precautions are not taken.

Warning
indicates that death or severe personal injury **may** result if proper precautions are not taken.

Caution
with a safety alert symbol indicates that minor personal injury can result if proper precautions are not taken.

Caution
without a safety alert symbol indicates that property damage can result if proper precautions are not taken.

Notice
indicates that an unintended result or situation can occur if the corresponding notice is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel.** Within the context of the safety notices in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

Prescribed Usage

Note the following:

Warning
This device and its components may only be used for the applications described in the catalog or the technical description, and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens.
Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG.
The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E03447308-02

04/2012

Table of Contents

1 PCS 7 BRAUMAT Library - basic concepts

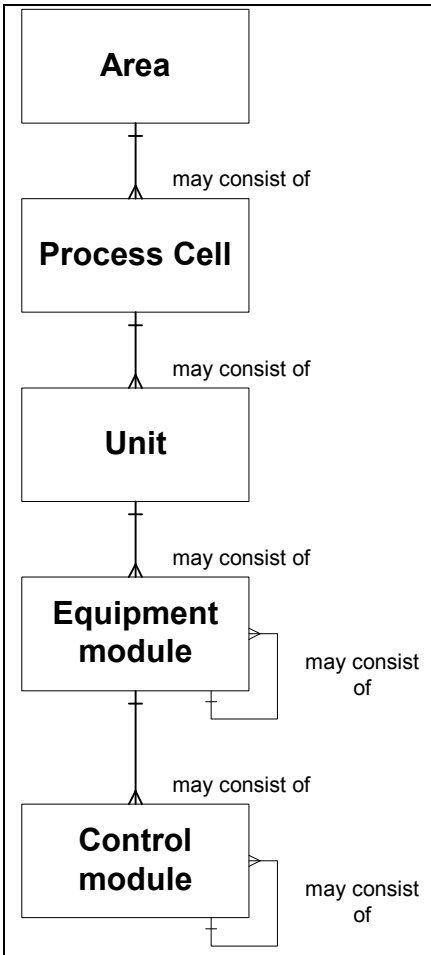
1.1 Brief overview

The PCS 7 BRAUMAT Library is based on the ISA-88 standard. This chapter describes the general concepts of the PCS 7 BRAUMAT Library to implement the ISA-88 physical model and the equipment procedure model by using the PCS 7 toolset.

Remarks

A general understanding of the ISA-88 standard is assumed.

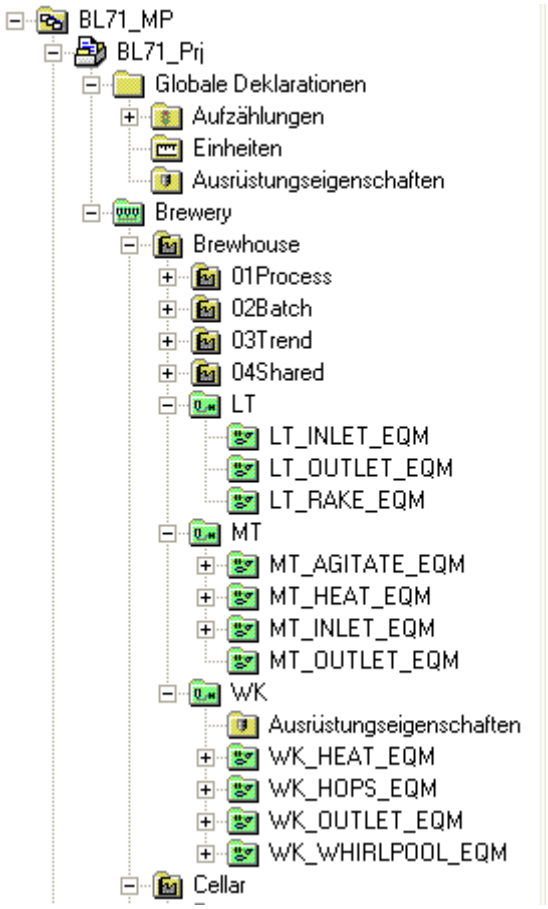
1.1.1 Physical model



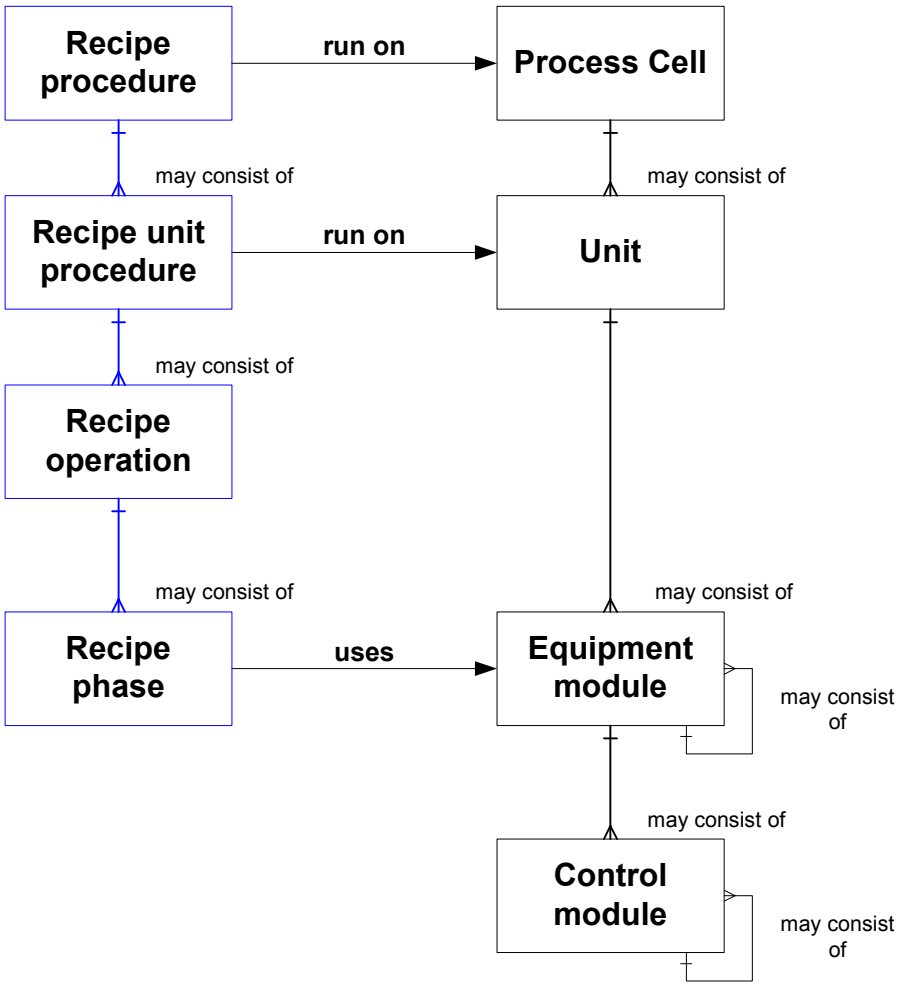
In the scope of the PCS 7 automation project there are the layers:

- Area (e.g. Brewery, Production)
- Process Cell (e.g. Brewhouse, Cellar)
- Unit (e.g. Mash Tun, Lauter Tun)

Representation in PCS 7 plant hierarchy



1.1.2 Procedural model

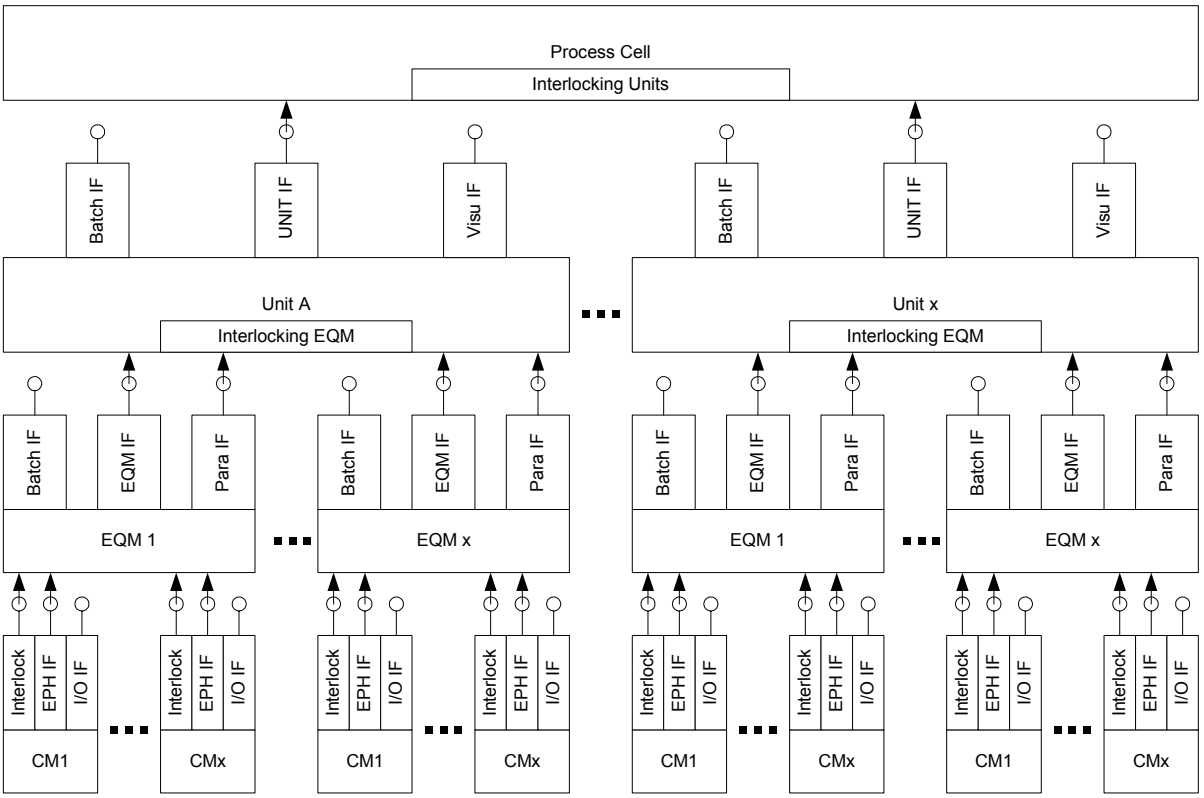


The control modules consists of blocks from the PCS 7 BRAUMAT Library and the PCS 7 Advanced Process Library (APL) and and prepared and tested for the use with two level recipes (Recipe operations and Recipe phases. The recipe phases corresponds with equipment modules implemented with SFC-Type). However the library blocks may be used basically also with single level recipes, where recipe operations corresponds with equipment operations which are implemented with SFC-Type.

1.1.3 Hierarchy levels

Layer	Implemented in PCS 7
Area	Depending on the plant complexity the production or process scope of a brewery may be represented by one PCS 7 Multi-Project or may be partitioned into several PCS 7 Multi Projects. A PCS 7 Multi-Project may consist of several Controllers and OS Servers or OS-Server Pairs. To every PCS 7 Multi-Project one SIMATIC BATCH Server-Pair and for that reason one SIMATIC BATCH project is assigned.
Process Cell PCell	Standard folder in the plant hierarchy view. A Process Cell is assigned to one SIMATIC BATCH Project. Coordination tasks for the unit and common functions for all units of the process are implemented in CFC charts in this folder. Consequently a Process Cell contains all Units which may be used in a recipe.
Unit	Unit folder in the plant hierarchy view One CFC chart contains at least the unit interface blocks (see following chapters). Interlocking and data processing for the subordinate equipment modules (EQMs) are implemented here or better in subordinate CFC charts.
Equipment module EQM	EQM Folder in the plant hierarchy view All EQMs have at least one standardized CFC chart for the unit interface and the phase logic (SFC-Type instance)
Control modules CM	Each control module is represented by one CFC chart. For many of the control modules templates of process tag types are contained in the scope of delivery of the PCS 7 BRAUMAT Library
Shared equipment	To handle shared equipment the PCS 7 BRAUMAT Library provides management blocks. These blocks can be used by the phase logic to allocate, control and monitor the shared equipment.

1.1.4 Software overview



1.1.5 Interfaces

Each layer has clear defined interfaces to the next higher level. Cross communication within one level is not used.

Exception: Project specific Interlocks of EQMs and control modules (CMs).

The interface between Unit and EQM is implemented by using the special PCS 7 BRAUMAT Library blocks BlUnitIf and BlEqmIf.

1.1.6 Naming convention

The names of the equipment modules should follow a project-wide naming scheme.

Recommendation:

<unit abbreviation>_<short name of the module>_EQM<optionally variant number>

Example: LT_Inlet_EQM01

Depending on the technical scope of the equipment modules different version of the same module may exist (e.g. heater module with two or three heating zones).

1.1.7 Modes

AUTO/MANUAL mode selection

All EQMs and CMs can run in automatic or manual mode. The mode can be activated centrally for each Unit by the BlUnitIf block and its according OS-Faceplate. The signals are propagated to all subordinate objects (EQMs, CMs) by data structure ‚EqmData'and ‚CmData' .

SIMULATION mode selection

A simple process simulation is implemented in the library. The Simulation-mode can be activated centrally for each unit by the BlUnitIf block and its according OS-Faceplate. In doing so, the control module feedback signals are generated in the adapter blocks itself (time delayed on demand) and signaled to the control module blocks via the input drivers. Independently the local simulation may be activated in the OS-faceplate of each control module.

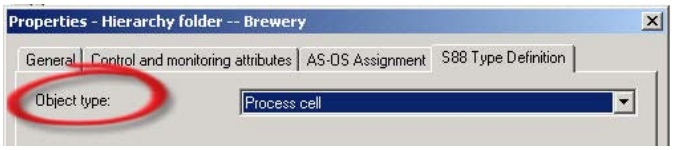
FORCE AUTOMATIC mode selection

The PCS 7 BRAUMAT Library has implemented this mode switchover strategy in such a way, that all EQMs and subordinated CMs can be enforced to automatic mode when the unit is occupied by SIMATIC BATCH. For activating this mode the feature bit 0 of the UNIT specific BlUnitIf block must be set true. Furthermore the input parameter UPLC_BA_INFO must be connected with the BA_INFO output of the UNIT_PLC block of SIMATIC BATCH, which allows detection of status changes in the SIMATIC BATCH unit.

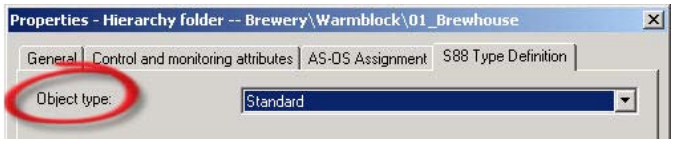
1.2 Plant hierarchy

1.2.1 Area/Process Cell - Folder

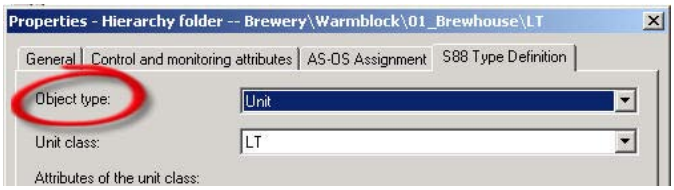
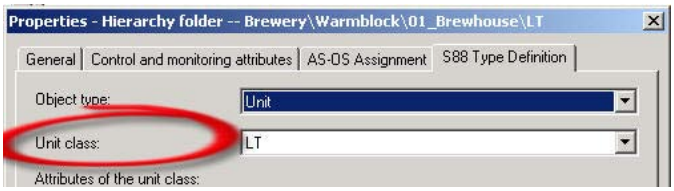
This physical model is converted into a plant hierarchy of the SIMATIC manager.

	SIMATIC BATCH handles only one Area as well as one Process Cell per SIMATIC BATCH server. The area folder (e.g. 'Brewery') must have the object type 'Process cell'. 

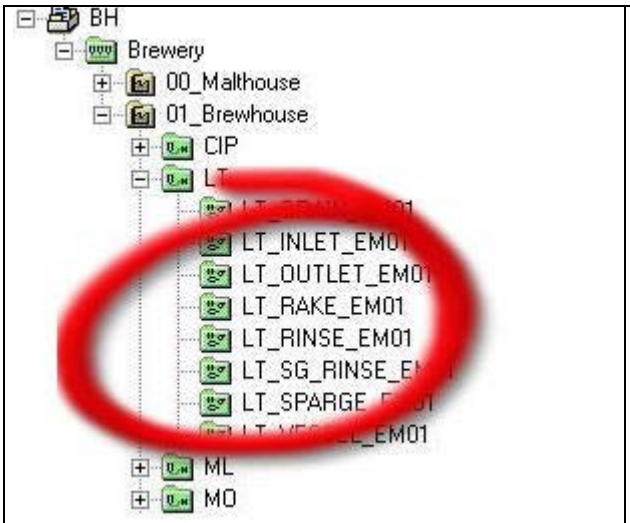
Standard - Folders for subdivision

	BRAUMAT PCS 7 uses folders with object type 'Standard' (= neutral, viz. 'not batch relevant') for one or more physical Process Cell(s) 

1.2.2 Unit - Folder

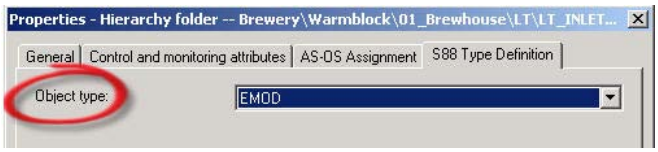
	The next hierarchy contains the units of the Process Cell. This folder must have the object type 'Unit'.  Each unit must be assigned to a unit class. 
	Each Unit-folder contains a CFC chart which holds the unit specific interface blocks for SIMATIC BATCH and the PCS 7 BRAUMAT Library.

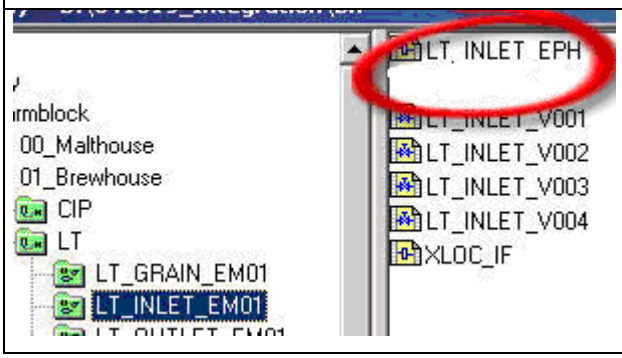
1.2.3 Equipment module – Folder



Each EQM is represented by a subfolder of the unit.

This folders must have the object type 'EMOD'.

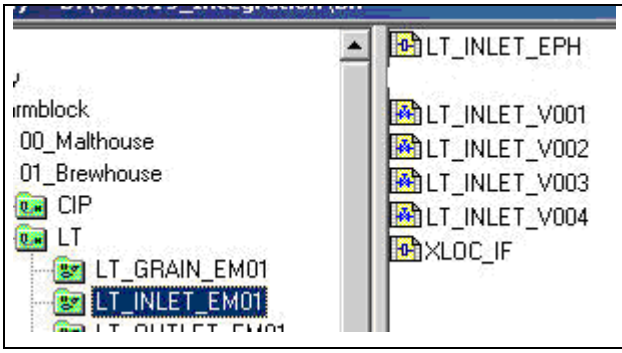




Each EQM folder contains CFC charts for:

- one chart for each of the control modules.
- one chart for the equipment phase

1.2.4 Control modules - Charts



Control modules are implemented as CFC charts below

- a EQM folder or
- a shared equipment folder.

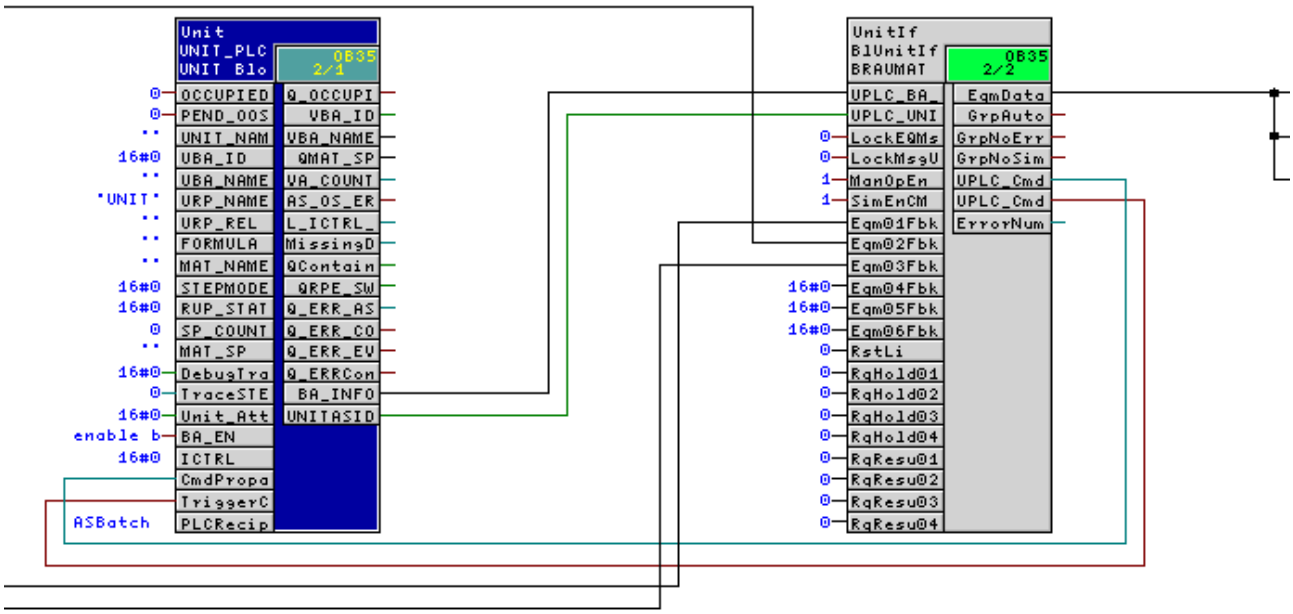
1.3 Units

1.3.1 Interface blocks on Unit level

The interface between the Unit related application program and SIMATIC BATCH is built up by the following blocks:

1. UNIT_PLC block of the SIMATIC BATCH block library
2. BlUnitIf block of the PCS 7 BRAUMAT Library

These two interface blocks have to be placed in a Unit specific CFC-chart. The basic Interconnections between both blocks is shown in the following screenshot



Signal routing from UNIT_PLC to BlUnitIf:

- BA_INFO → UPLC_BA_INFO (Unit Name, Batch Name, Batch ID, Step No und Q_STATUS)
- UNITASID → UPLC_UNITASID (AS-ID und UNIT-ID)

Signal routing from BlUnitIf to UNIT_PLC:

- UPLC_CmdProp → CmdPropagation (sende ISA-88 Kommando zur Unit)
- UPLC_CmdTrig → TriggerCmdPropagation (Trigger für ISA-88 Kommando)

Additional note on BlUnitIf/UNITASID connector:

The BlUnitIf block for itself generates no messages. Nevertheless there is an alarm view in its OS-Faceplate, which shows the messages of all subordinate EQMs/CMs as well as the shared equipment CMs which are assigned to the unit dynamically. To enable this message filtering, this input parameter must receive a plant-unique identifier, composed of AS- and UNIT-ID from SIMATIC BATCH block UNIT_PLC.

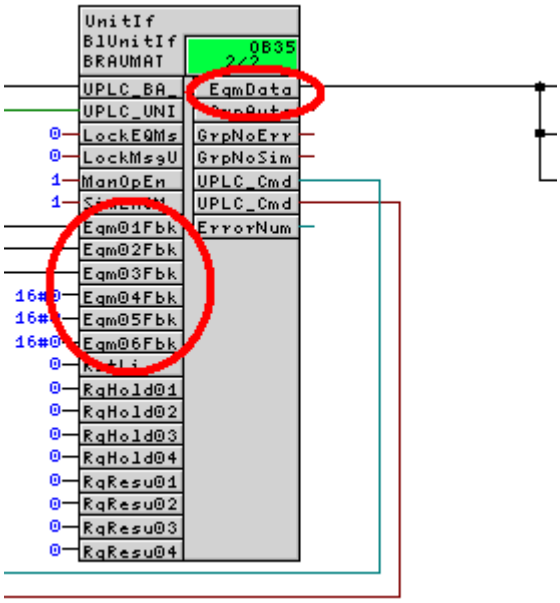
1.3.2 Interface from Unit level to EQMs

Up to 12 BlEqmIf instances (EQMs) may be connected to one BlUnitIf interface block. This connection is built up through the 1:n 'downstream' data structure connection to the subordinate EQMs

- BlUnitIf/EqmData → BlEqmIf/EqmData of each according EQM

and the 1:1 'upstream' Feedback data structure connections

- BlEqmIf/EqmFbk → BlUnitIf/Eqm01Fbk…Eqm12Fbk from each EQM back to the BlUnitIf block



Runtime signal exchange to EQMs via EqmData data structure output

- unit status and batch information
- hold propagation
- mode propagation AUTO/MAN
- UNIT - Simulation
- Force AUTO for CMs/EPEs
- Lock/Release MANUAL mode for CMs/EPEs
- Reset Signal

Feedback signals from EQMs to UNIT via Eqm<xx>Fbk data structure inputs

- Hold/Resume propagation upstream
- SFC Error and ISA-88 Status
- Batch StepNo

1.3.3 Hold handling

1.3.3.1 Hold command from unit level

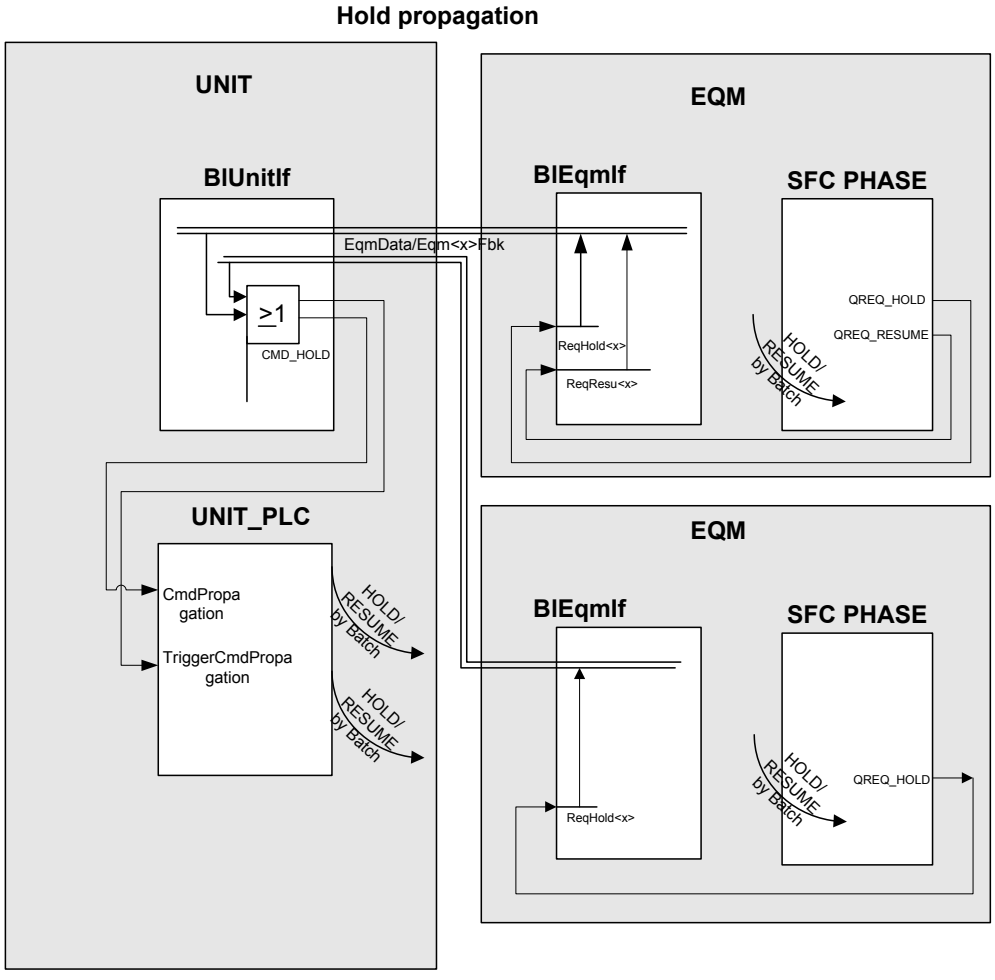
From the Unit-level all subordinate EQMs can be set to the HELD state by an operator activation whether in the Batch Control Center or in the OS-Process-Picture via the BCC-OCX control.

Furthermore the BlUnitIf inputs RqHold01 … RqHold16 and RqResu01 … RqResu08 allow to send requests for hold/resume to the Unit_PLC block. In this way running equipment phases belonging to this unit may be held / resumed centrally by SIMATIC BATCH (see description of SIMATIC BATCH for further information).

1.3.3.2 Request Hold/Resume from the EQM level

An EQM can request the HELD state. This is initiated by the SFC-Type instance block of this EQM through the BlEqmIf inputs RqHold01 … RqHold16 und RqResu01 … RqResu08 which allow to send requests for hold/resume to SIMATIC BATCH via the BlUnitIf → UNIT_PLC feedback connection. Likewise the running EQMs belonging to this unit may be held / resumed by SIMATIC BATCH (see description of SIMATIC BATCH for further information).

The following schematic diagram shows the signal routing mechanism between the PCS 7 BRAUMAT Library and SIMATIC BATCH UNIT_PLC.



1.4 Equipment module

1.4.1 BRAUMAT equipment interface block

The data exchange between the unit and the EQM is realized within the EQM specific CFC chart with a BRAUMAT block BlEqmIf. The block supports the BlUnitIf block and its OS-Faceplate to maintain the monitoring and control features via the data structure connections containing the Unit related interface signals (Unit-State, Batch-Info, Modes and commands) which have been described in chapter 1.3.2 ‘Interface from Unit level to EQMs’.

The BlEqmIf block is able to control up to 32 control modules in association with the SFC-Type. The following CFC connections are required:

Signal routing BlEqmIf to EQM SFC instance

- AUT → AUT (Switch to automatic mode)
- ENAUT → ENAUT (Enable switching to automatic mode)
- MAN → MAN (Switch to manual mode)
- ENMAN → ENMAN (Enable switching to manual mode)

Signal routing from EQM SFC instance to BlEqmIf

- QSTEP_NO → STEP_NO
- BA_STATE → BA_STATE

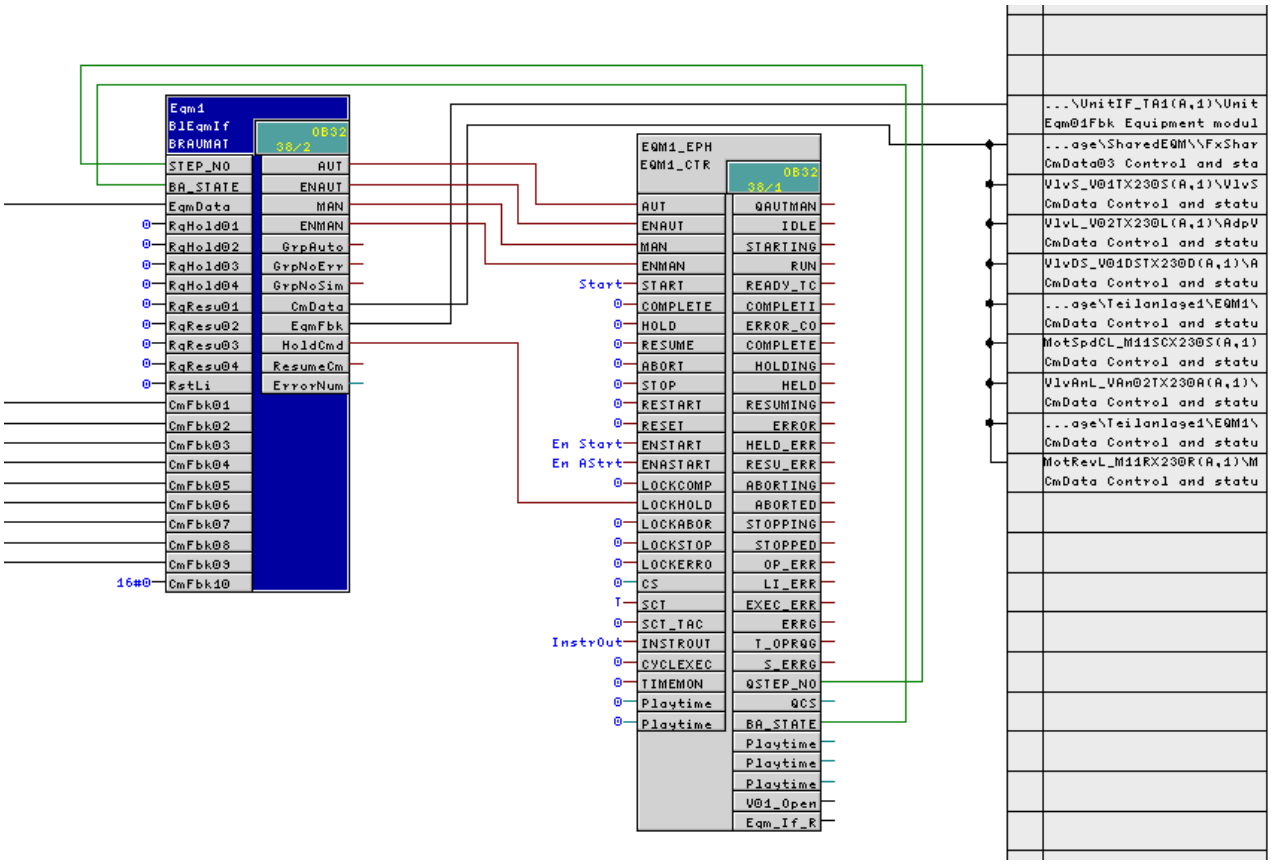
Signal routing from BlEqmIf to all subordinate CM-adapter blocks

- CmData → CmData

Signal routing from CM-adapter to BlEqmIf

Each adapter block transfers a feedback data word back to the BlEqmIf block

- CmFbk → CmFbk01 … CmFbk32


Eqm1
BlEqmIf
BRAUMAT
OB32
38/2
STEP_NO
BA_STATE
EqmData
RqHold01
RqHold02
RqHold03
RqHold04
RqResu01
RqResu02
RqResu03
RqResu04
RstLi
CmFbk01
CmFbk02
CmFbk03
CmFbk04
CmFbk05
CmFbk06
CmFbk07
CmFbk08
CmFbk09
CmFbk10
AUT
ENAUT
MAN
ENMAN
GrpAuto
GrpNoErr
GrpNoSim
CmData
EqmFbk
HoldCmd
ResumeCm
ErrorNum
EQM1_EPH
EQM1_CTR
OB32
38/1
AUT
ENAUT
MAN
ENMAN
START
COMPLETE
HOLD
RESUME
ABORT
STOP
RESTART
RESET
ENSTART
ENASTART
LOCKCOMP
LOCKHOLD
LOCKABOR
LOCKSTOP
LOCKERRO
CS
SCT
SCT_TAC
INSTROUT
CYCLEXEC
TIMEMON
Playtime
Playtime
QAUTMAN
IDLE
STARTING
RUN
READY_TC
COMPLETI
COMPLETE
HOLDING
HELD
RESUMING
ERROR
HELD_ERR
RESU_ERR
ABORTING
ABORTED
STOPPING
STOPPED
OP_ERR
LI_ERR
EXEC_ERR
ERRG
T_OPRQG
S_ERRG
QSTEP_NO
QCS
BA_STATE
Playtime
Playtime
Playtime
V01_Open
Eqm_If_R
...\UnitIF_TA1(A,1)\Unit
Eqm01Fbk Equipment modul
...age\SharedEQM\\FxShar
CmData03 Control and sta
VlvS_V01TX230S(A,1)\VlvS
CmData Control and statu
VlvL_V02TX230L(A,1)\AdpV
CmData Control and statu
VlvDS_V01DSTX230D(A,1)\A
CmData Control and statu
...age\Teilanlage1\EQM1\
CmData Control and statu
MotSpdCL_M11SCX230S(A,1)
CmData Control and statu
VlvAnL_VAn02TX230A(A,1)\
CmData Control and statu
...age\Teilanlage1\EQM1\
CmData Control and statu
MotRevL_M11RX230R(A,1)\M
CmData Control and statu

1.4.2 Equipment phase

For each EQM one CFC - chart is to be implemented. This chart should have the name of the EQM with the extension '_EPH' (equipment module phase). The phase logic of the EQM is implemented primarily within an SFC-Type. The different functionality may be implemented as control strategies.

1.4.2.1 General

Sequences

Each SFC-Type has a standard structure consisting of the following sequences:

- Starting (only when necessary)
- Running
- Completing (for the most part used commonly for the ISA-88 states completing and aborting)
- Stopping (only when necessary)
- Holding
- Resuming

Self terminating and non self - terminating phases

- PCS 7 BRAUMAT Library phases are implemented as self terminating or non self terminating phases according the technological demands.

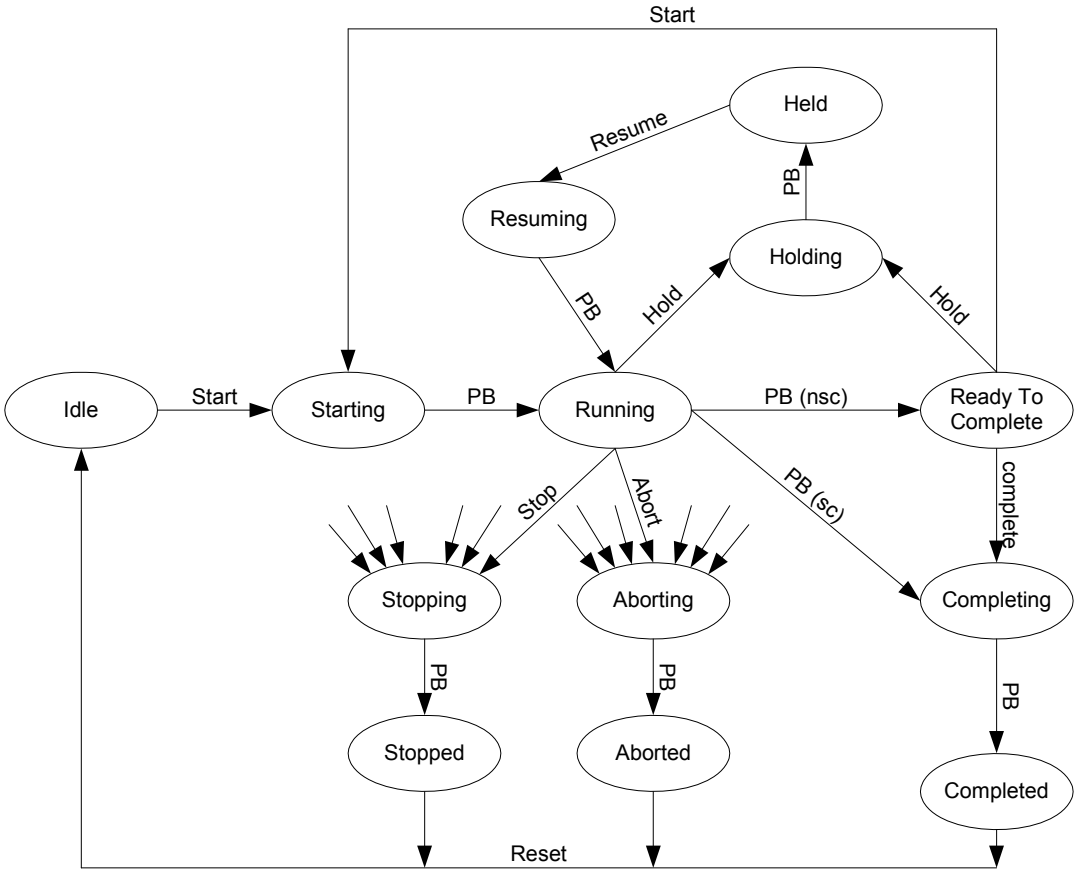
Target step number and target sequence number in SFC

- This feature is not used in the library.

Unused sequences

- Unused sequences should be deleted.

1.4.2.2 ISA-88 states and transitions



Note:
PB = „Prozessbedingung“ (EN→process condition)

1.4.2.3 Control strategies

All equipment phases uses control strategies. The control strategies represent the different technical activities the equipment module can execute. This results in alternative branches (each strategy one branch) in a SFC-Type.

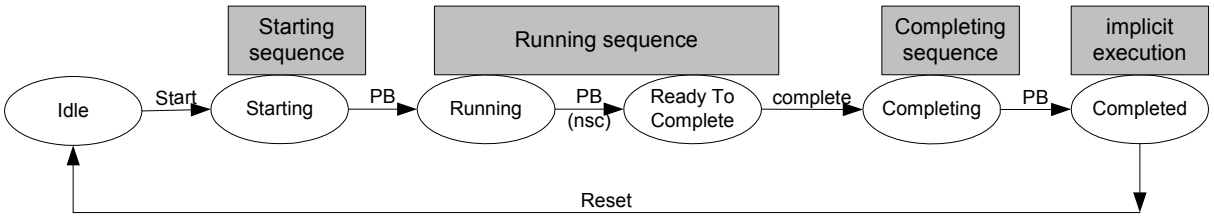
In combination with the non self completing functionality it is possible to keep all the equipment active during a step change of the batch system.

The system allows at maximum 32 control strategies.

1.4.2.4 State sequence ‘normal execution’

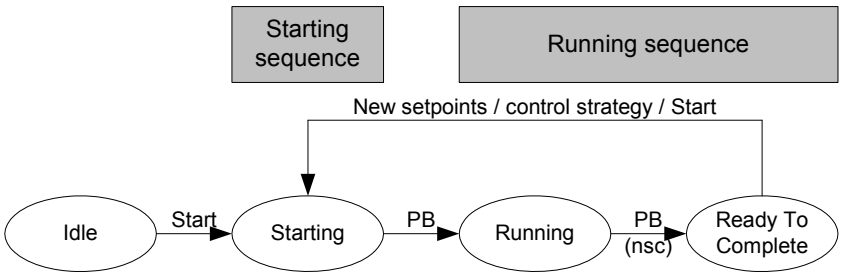
Scenario:

- the phase is started by SIMATIC BATCH
- phase logic in the PLC executes the ‘starting’sequence; starting will be left after the process conditions for ‘end starting’ is fulfilled
- phase logic executes the running sequence; when the process condition for ‘ready to complete’ (RTC) is fulfilled the phase signals this to the SIMATIC BATCH via setting READY_TC
- the batch system checks the conditions within the recipe and complete the phase
- the phase logic executes the ‘completing’ sequence; when the process condition for ‘end completing’ is fulfilled the phase leave the completing sequence
- the batch system resets the phase



1.4.2.5 State sequence ‘continuous execution’

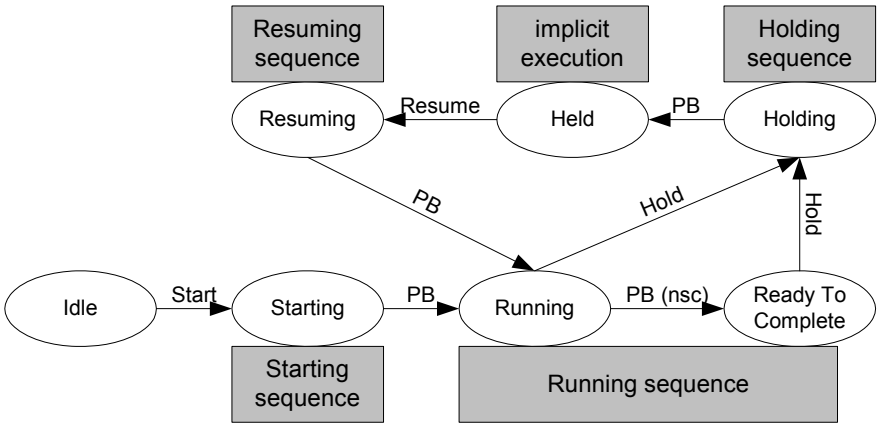
- the phase is started by SIMATIC BATCH
- phase logic in the PLC executes ‘starting’ sequence; starting will be left after the process conditions for ‘end starting’ is fulfilled
- phase logic executes the running sequence; when the process condition ‘ready to complete’ (RTC) is fulfilled the phase signals this to SIMATIC BATCH via setting READY_TC
- SIMATIC BATCH checks the conditions within the control recipe
  when the selection ‘continuous’ is set for this recipe phase the equipment phase will be provided with new set points, control strategy and start again
- when no recipe phase uses the equipment phase, it will be completed by SIMATIC BATCH via the ‘completing’ sequence
- in any case the status will be set to completed before the unit will be released



1.4.2.6 State sequence ‘holding’

Scenario:

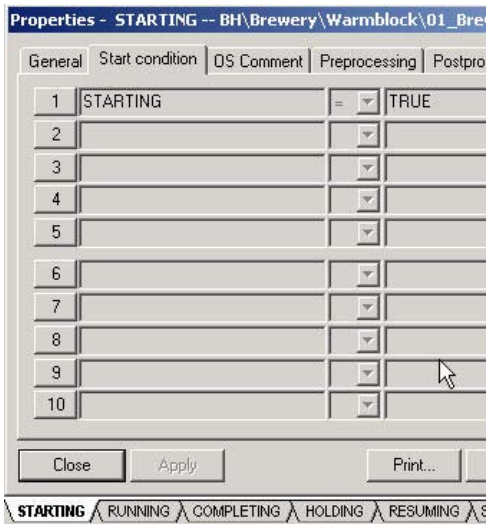
- Start of the phase as ‘normal execution’
- With the command ‘hold’ set by SIMATIC BATCH or PLC program the running sequence will be left and the holding sequence will be executed
- When the process condition for terminating the ‘holding’ sequence is fulfilled the Held status is reached
- The ‘held’ status is implicit executed by the SFC-Type; no activities are implemented in the PCS 7 BRAUMAT Library
- With the command ‘resume’ set by batch system or PLC program the held status will be left
- Within the ‘resume’ sequence the phase can be re-initialized
- When the process condition for terminating the ‘resume’ sequence is fulfilled, the phase goes back to the ‘running’ sequence



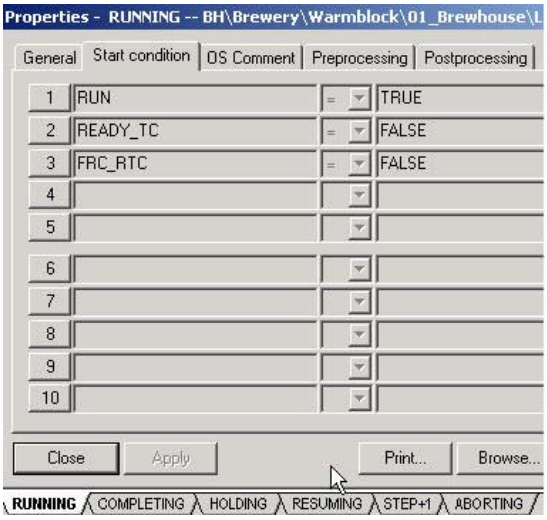
1.4.2.7 Phase sequences

Start conditions

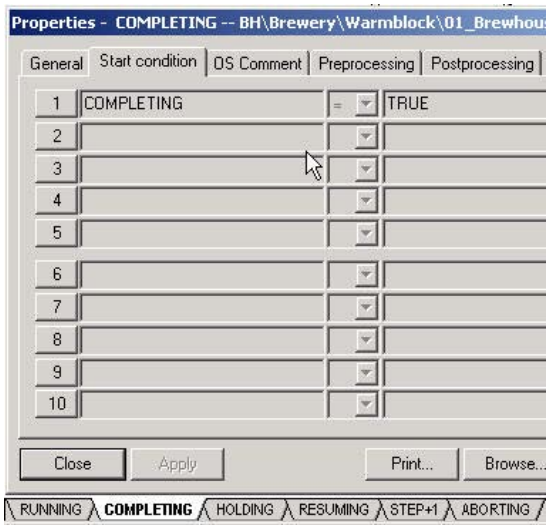
Starting sequence



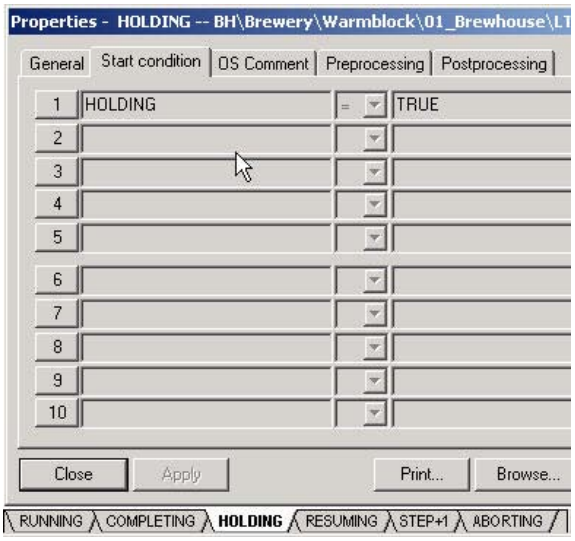
Running sequence



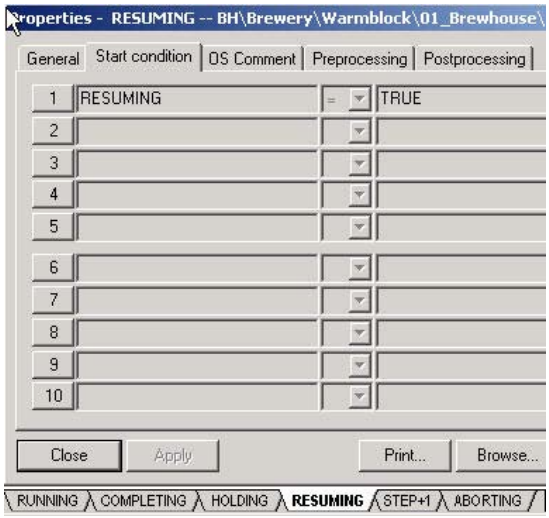
Completing sequence



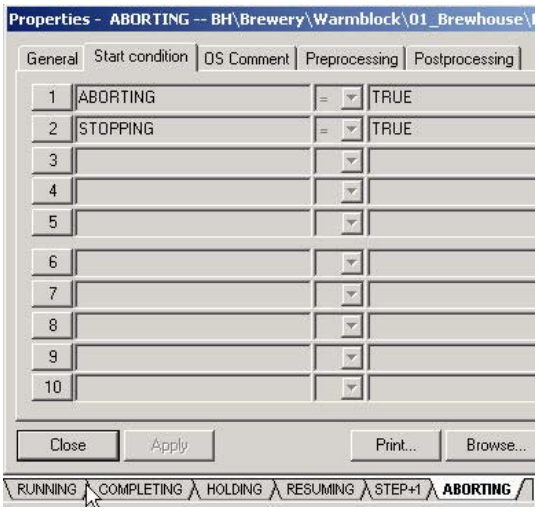
Holding sequence



Resuming sequence



Aborting sequence

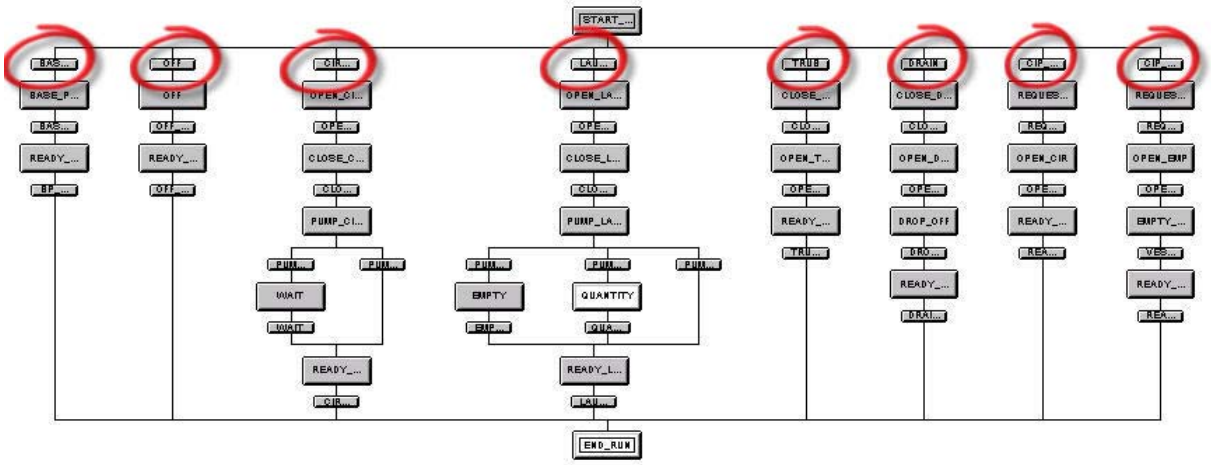


Control strategies

The control strategies represent the different technical steps the equipment module can execute. This results in alternative branches (each strategy one branch) in a SFC-Type.

In each sequence (starting, running, …) the behavior for the control strategies must be defined.

Example for running sequence with control strategies



The entry transitions for the branches evaluate the current control strategy value to select the right branch (switch case approach). The output value QCS must be used for the evaluation of the control strategy.

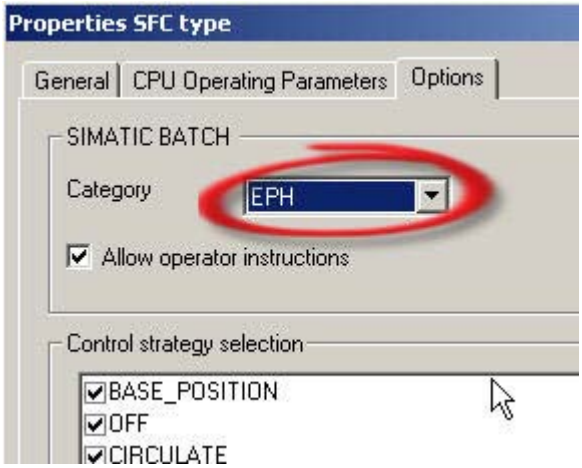
The control strategy is a number representing the strategy enumeration value defined in the characteristics of the SFC-Type.

1.4.2.8 SIMATIC BATCH interface

Handshake

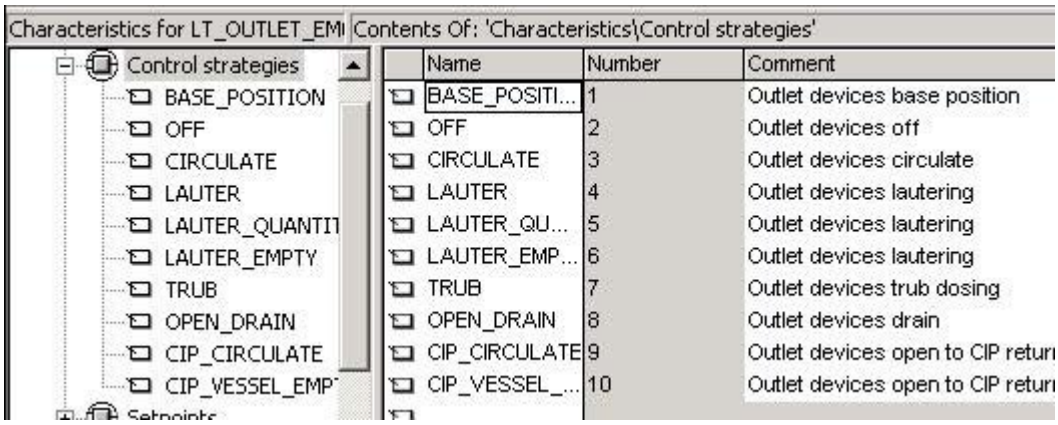
For the communication with the SIMATIC BATCH system the SFC contains all needed In- and Outputs.

The engineer must set in the SFC-Type property page the category to ‘EPH’ or ‘EOP’ respectively.



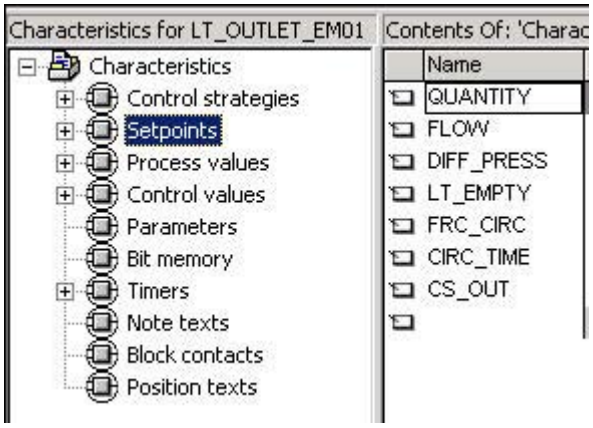
Control strategies

The control strategies must be defined with the interface editor of the SFC-Type.



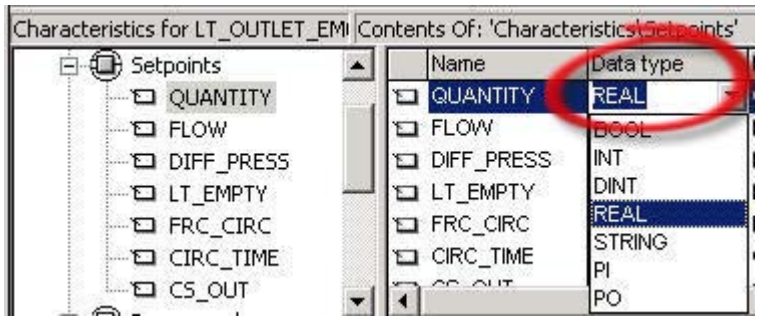
Recipe set points

The recipe set points must be defined with the interface editor of the SFC-Type.



Data types for recipe set points

For each set point a data type must be set.

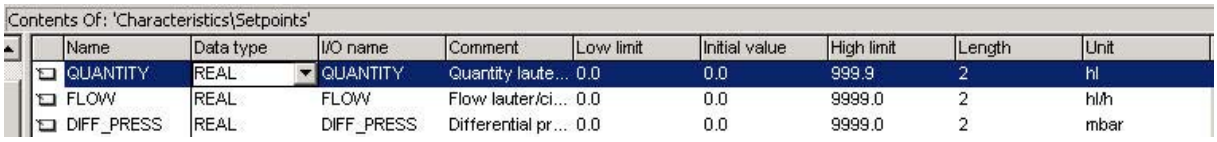


Possible data types

Data type	
BOOL	Boolean value
INT	16 Bit signed integer value
DINT, SOURCE, DEST and VIA	32 Bit signed integer value
REAL	32 Bit S7 real value
STRING	Character field with maximum 16 chars
PI / PO	Process input data type - refer SIMATIC BATCH documentation

Limits and dimension

The limits must be set to each set point. This limits are transferred to SIMATIC BATCH and are valid for the usage in recipe phases.

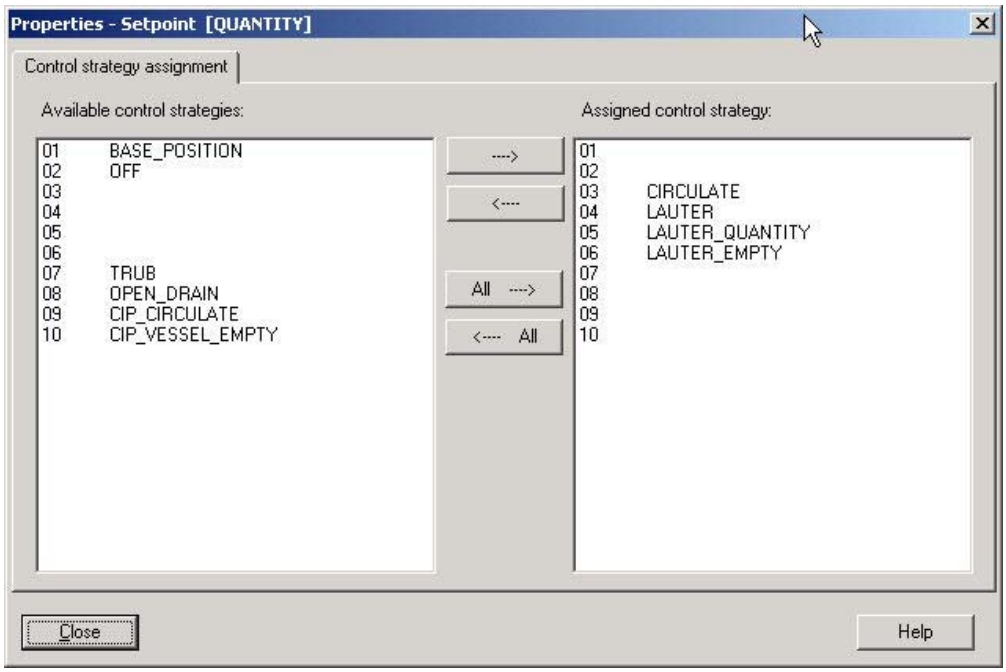


Control strategy - set point assignment

Each set point must be assigned to a control strategy. A new inserted set point is assigned to all control strategies.

The dialog can be opened by a double click on the last column in the set point table.

Then the dialog pops up and the assignment can be made.

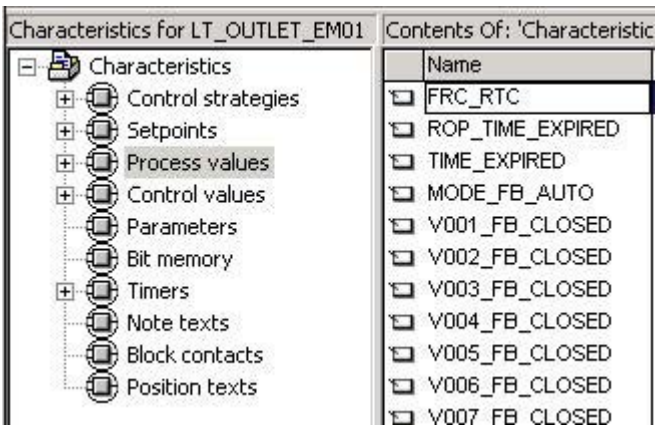


1.4.2.9 Process interface

Process values

Process values represent the feedback from the process to the phase (e.g. valve feedback, analogue values). Process values are primarily used for step enabling conditions in the sequencer steps.

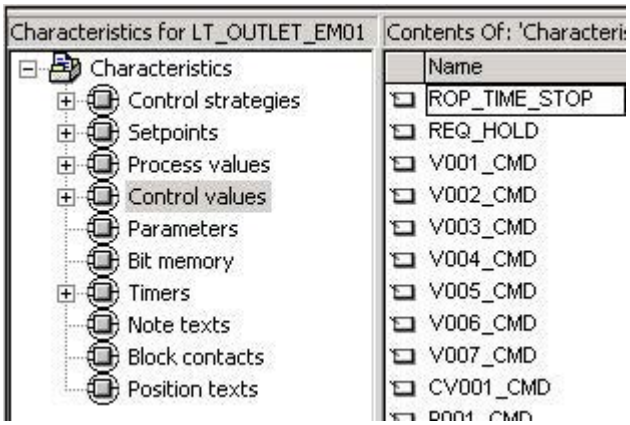
The process values must be defined with the interface editor of the SFC-Type.



Control values

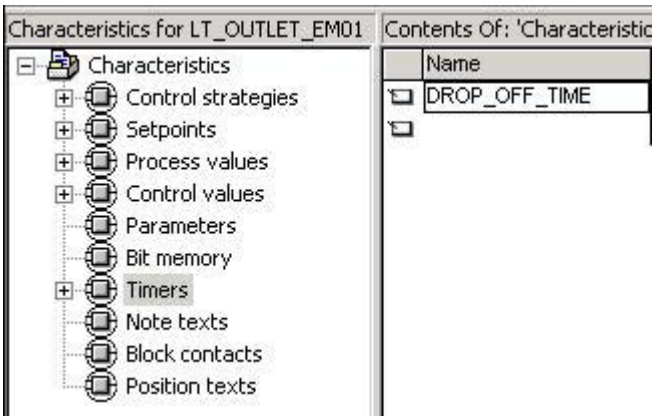
Control values are commands or set points for the equipment of the EQM.

The control values must be defined with the interface editor of the SFC-Type.

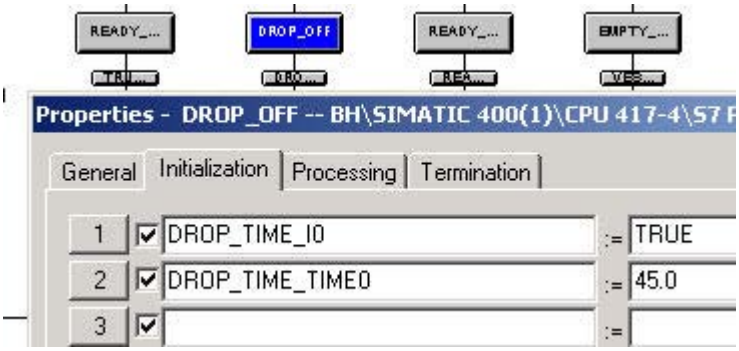


1.4.2.10 Timers

Definition

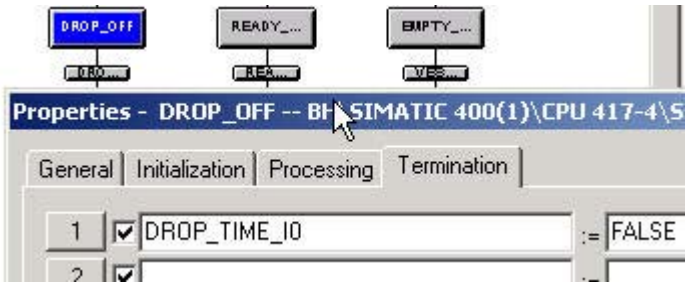


Start and set time value



Evaluate timer expired

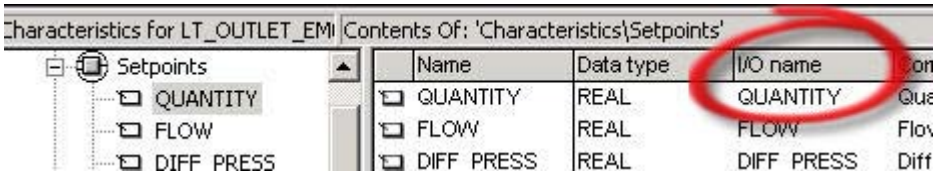
Stop the timer



1.4.2.11 In- outputs of SFC Instance

Set points

For each set point e.g. of type REAL the SFC editor creates 13 in- or output signals. The according name-suffix is automatically preset by the editor, but can be modify manually.



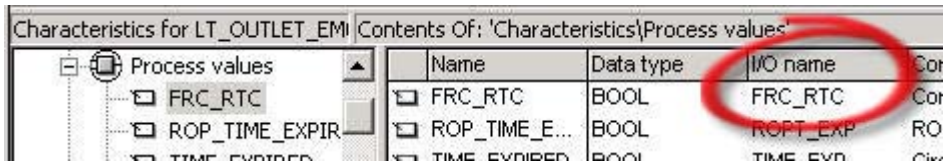
IO signals

Signal	in/out	Outputs
„name“	IN	Automatic process value input (= set point from SIMATIC BATCH)
„name“_AI	IN	→ Actual value input
„name“_AO	OUT	→ Actual value output
„name“_CS	IN_OUT	Assignment to control strategy
„name“_ENOP	IN_OUT	Enable operator input in the according faceplate
„name“_ENPOP	IN_OUT	Enable operator to set prepared set point value
„name“_ERROP	OUT	Set point error (typically out of limits)
„name“_HL	IN	Upper limit of the set point
„name“_LL	IN	Low limit of the set point
_OP	IN	→ Operator input value
_OPP	IN	Prepared operator values
_Q	OUT	→ Valid set point output
_QP	OUT	Valid set point prepared

The signal marked with '→' are typically used in the library modules.

Process values

For each process value the SFC editor create one input signal.

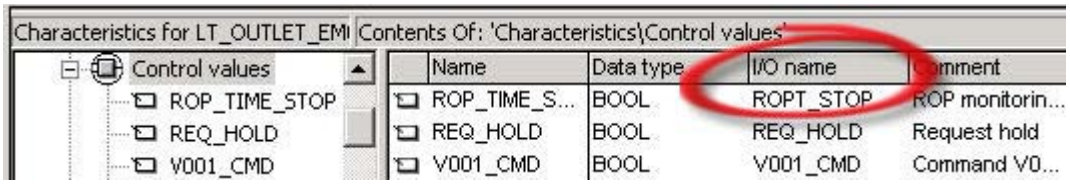


IO signals

Signal	in/out	Outputs
„name“	IN	Input for the process value

Control values

For each control value the SFC editor create one output signal. The name is automatically preset by the editor, but can be modify manually.



IO signals

Signal	in/out	Outputs
„name“	OUT	Control value

Timer

For each control value the SFC editor create one output signal. The name-suffix is automatically preset by the editor, but can be modified manually.

IO signals

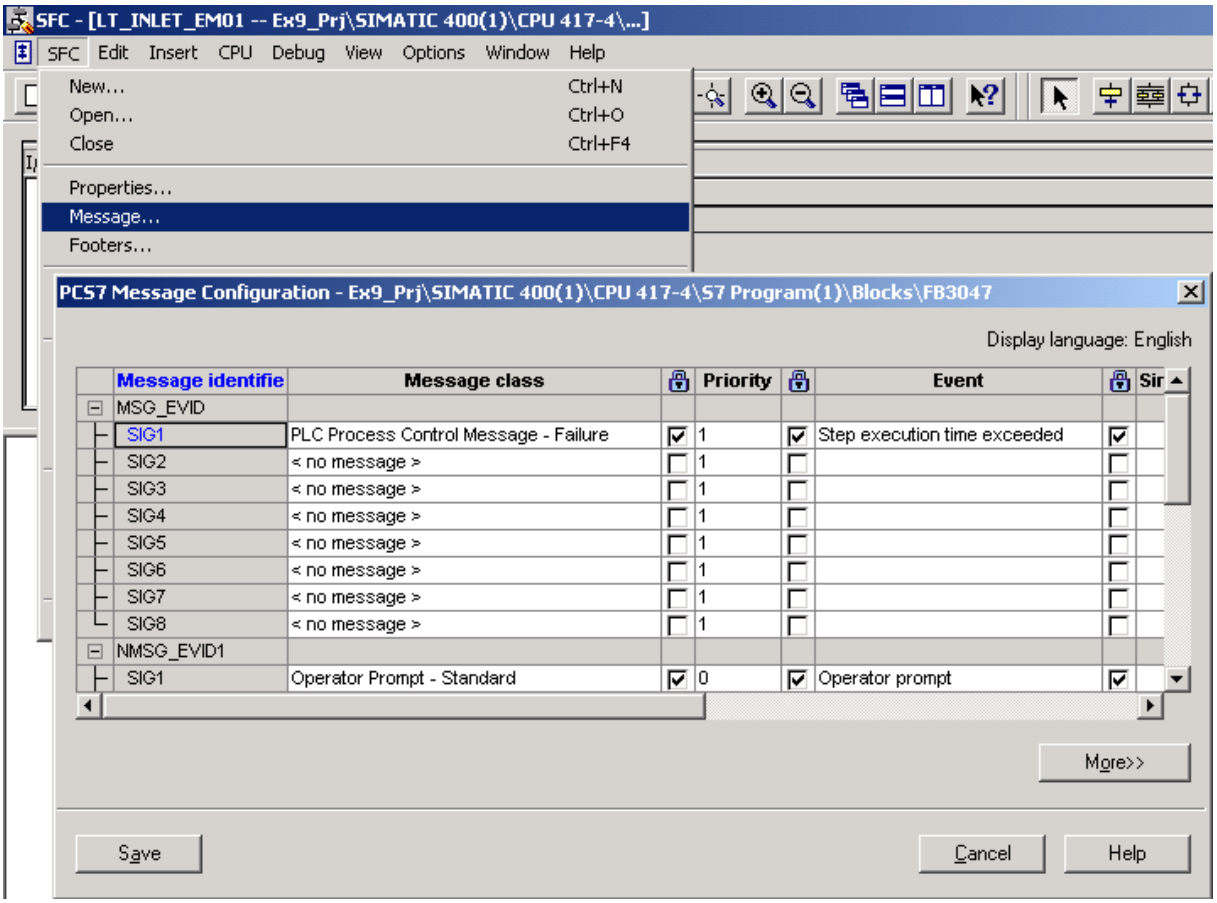
Signal	in/out	Outputs
SAMPLE_T	IN	Task sampling time in sec (is set by CFC compiler with the adequate OB cycle time of the SFC instance).
„name“_IO	IN	Input pulse to start or stop the timer
„name“_MODE	INT	Operating mode: 0: Start timer as pulse "Pulse" 1: Start timer as extended pulse "ExtP" 2: Start timer with on delay "OnDel" 3: Start timer with retentive on delay "RetOn-D" 4: Start timer with off delay "Off-D"
„name“_PTIME	OUT	Time to go
„name“_QO	OUT	Output pulse
„name“_QERR	OUT	Error
„name“_RESET	IN_OUT	Reset
„name“_TIME0	IN_OUT	Time in seconds

1.4.2.12 Alarms in SFC-Type

The message section in SFC-Type is used for defining alarms and operator messages. You can have a maximum of 12 user defined messages in one SFC-Type.

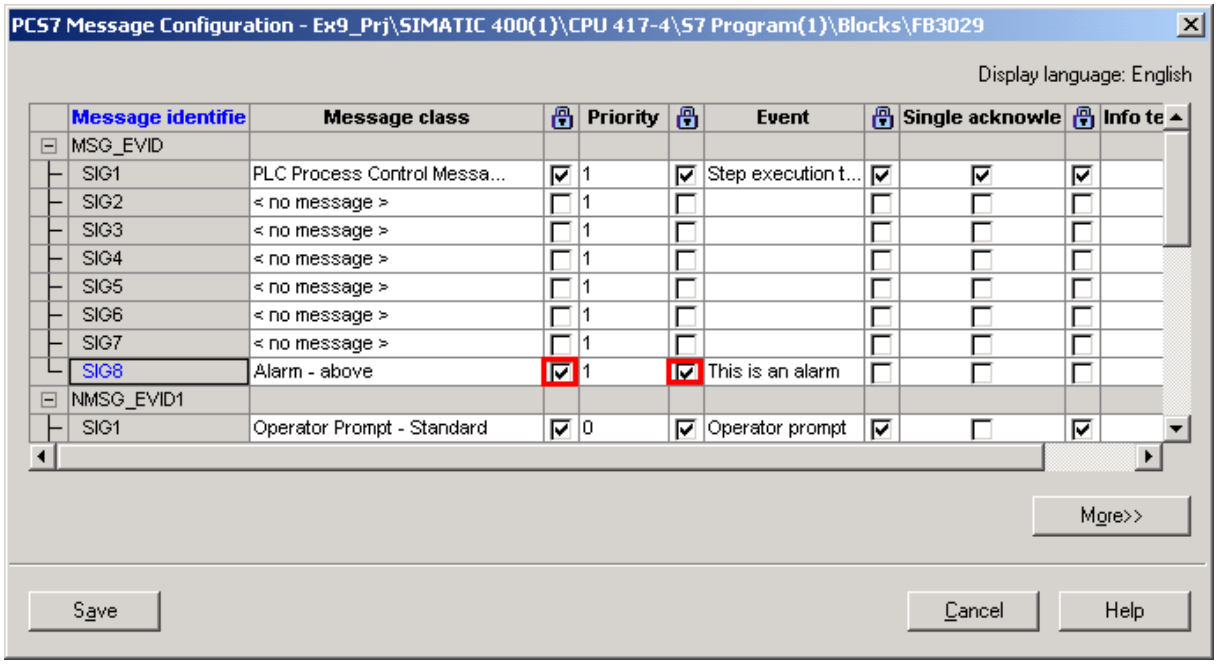
Message preparation

For each message one input is reserved in SFC-Type. For the first 7 inputs SIG_2 – SIG_8 ALARM_8P messages and for the next 5 the inputs NSIG_12 – NSIG_16 NOTIFY_8P messages are generated.



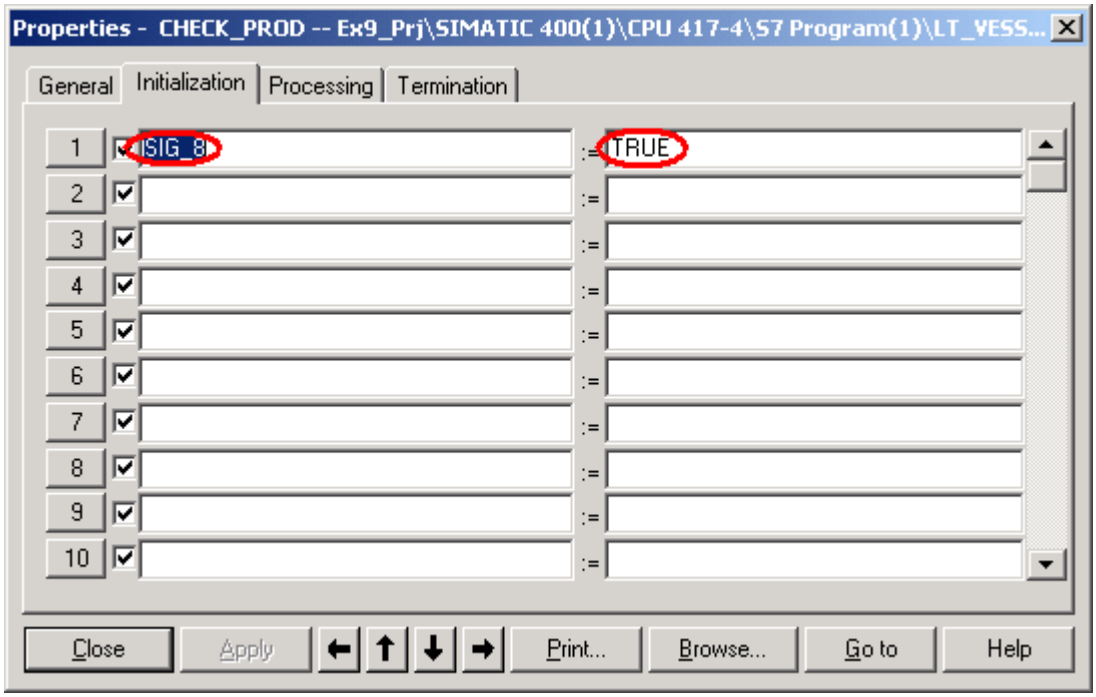
In message section you define message class, priority, message text as an event and acknowledge action. Message class should be alarm, warning, tolerance and etc. In the Event field you can type the message text.

For proper functionality of message, you have to check following check boxes:



Message activation.

Each message is activated by setting the according SIG_x input = TRUE in the desired Step of the SFC-Type.



For reactivating the same alarm you have to reset the input signal. This should be located in the END step of SFC-Type.

Numbering of messages are following:

Signal in SFC-Type step	Signal in Message part of SFC-Type
SIG_2	MSG_EVID / SIG2
SIG_3	MSG_EVID / SIG3
SIG_4	MSG_EVID / SIG4
SIG_5	MSG_EVID / SIG5
SIG_6	MSG_EVID / SIG6
SIG_7	MSG_EVID / SIG7
SIG_8	MSG_EVID / SIG8

Signal in SFC-Type step	Signal in Message part of SFC-Type
NSIG_12	NMSG_EVID1 / SIG4
NSIG_13	NMSG_EVID1 / SIG5
NSIG_14	NMSG_EVID1 / SIG6
NSIG_15	NMSG_EVID1 / SIG7
NSIG_16	NMSG_EVID1 / SIG8

1.5 Control Modules (CMs)

1.5.1 General

The control modules are implemented by using the process tag types (CFC-charts) consisting of blocks from the PCS 7 BRAUMAT Library as well as the PCS 7 Advanced Process Library (APL). In the scope of delivery a profile of frequently used process tag types is included which contain for example the following function templates:

- Analogue value monitoring
- Digital value monitoring (with/without messages)
- On / Off Motor
- Speed controlled motor
- Reversible motor
- PID controller
- Open / Close valve
- valves with two positions (double seat valves)
- Position controlled valves

The available process tag types (description see chapter 2

PCS 7 BRAUMAT Library – process tag types) are delivered as a Master Data Library inside the Examples project. For using the Templates in the user project this Library may be inherited completely or partly in the Project Master Data Library.

1.5.1.1 Plant code of the control modules

The plant hierarchy of BRAUMAT PCS 7 project is used for structuring the multitude of equipment according the plant structure. This hierarchy are not part of the CM name. BRAUMAT PCS 7 uses for each CM one CFC chart representing the process tag of that CM. This chart gets the name of the plant code. So all alarms of the CM are using the plant code in the source field.

The displayed name in the alarm view consist of the plant code plus the instance name of the used block. This name should assigned as follows:

- /M for motors / pumps
- /V for valves
- /DI for digital monitoring
- /AI for analogue monitoring
- /PID for controller

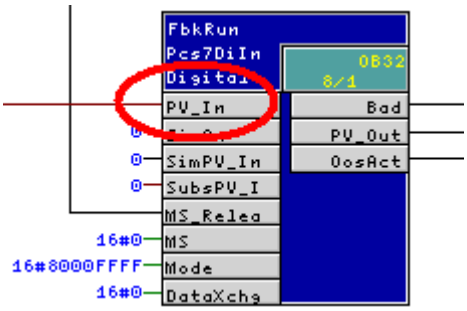
1.5.1.2 Textual interconnections

The interface of the CMs are realized by using textual interconnections of the CFC chart with the following general properties:

- Hierarchical text can be used (e.g. <chart name>\ EqmIf02.CmFbk01)
- Must not be closed (warning after compilation)
- May be closed manually
- Are closed automatically when text string and chart-/block- and input or output signal name fits together

1.5.1.3 I/O connection

The I/O connections of the PCS 7 BRAUMAT Library blocks are realized with PCS 7 channel driver blocks.



The input or output signal are defined as textual interconnection in the CFC chart.

This will be closed automatically by give the correct symbolic name whether in the process tag's CFC-chart or during import of the process tags via Import/Export- Assistant

1.5.1.4 External message inputs

Those PCS 7 BRAUMAT Library blocks which generates messages, provides external inputs ‚ExtMsg<x>' for ALARM_8P signals which are not used internally. With that feature, user specific messages per block instance may be generated by connecting application logic to these inputs.

The Message-Trigger inputs are forwarded to ALARM_8P inputs internally the block. The same auxiliary 'ExtVa<xxx>' values are available as with the internal messages. For activation the block special object properties/messages dialog must be engineered with the desired setting.

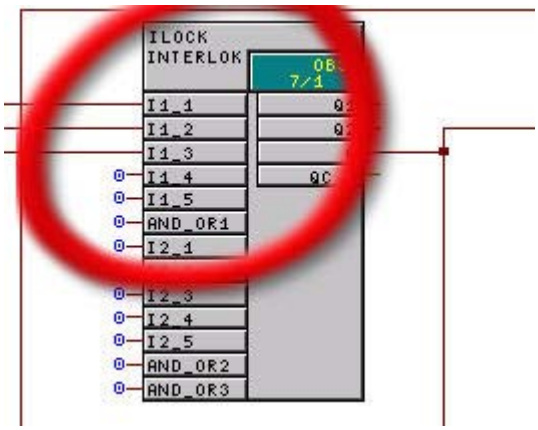
1.5.1.5 Interlocks

The APL blocks contains three interlock inputs:

- **Activation enable ("Permission"):**
  The activation enable (input `Permit = 1`) makes it possible to leave the neutral position of the block in response to operator input or a command from the program (CFC/SFC). The activation enable has no effect if the block is not in the neutral position.
- **Interlock without reset ("Interlock"):**
  An active interlock condition brings the block to the neutral position (input `Intlock = 0`). After the interlock condition has gone, the currently active control function becomes active again in automatic or local mode. In manual mode the faceplate can be operated again after the interlock condition has gone.
- **Interlock with reset ("Protection"):**
  An active interlock condition brings the block to the neutral position (input `Protect = 0`). After the interlock conditions are cleared, the operator or an activation sequence must perform a reset to once again enable activation of the control according to the input parameters.

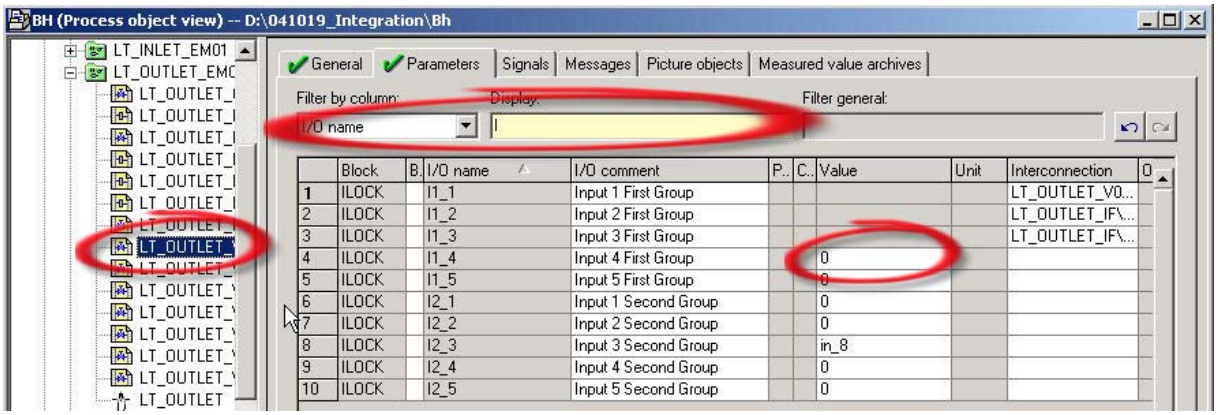
You can influence the reset using various Feature bits. Refer to chapter "Resetting the block in case of interlocks or errors" in the APL-documentation for more on this.

Additional Standard Library interlock blocks may be connected upstream, if needed.



Three interlocks are connected to textual interconnections, the other interlocks are unused. Interlocks are engineered after import of the modules by using the symbolic names of the I/O signals.

Usually the interlocks are configured in a very late engineering phase. This can be done very comfortable in the process object view of the SIMATIC manager.



By selected the right chart within the plant hierarchy (Tree-view on the left) all signals which are labeled as ‘parameter’ are shown in one list. By using a suitable filter only the relevant signals for the interlocking are in one list.

1.5.1.6 Superior mode handling of the CMs

Each control module can be run in manual or automatic mode. Furthermore the CMs may be switched into ‘Unit Simulation Mode’ via the according adapter blocks, additional to the manually selectable ‘individual local simulation’ of each control module.

The mode of operation is as demanded centrally from the BlUnitIf block via the BlEqmIf- block for all subordinate CMs. The data connection of an equipment module with the adapter blocks is carried out on the basis of a 1:n relationship, meaning that a connection of the EQM interface block (BlEqmIf, BlFxCtrl) from the output "CmData" to each assigned CM is required.

This contains the following individual information:

- Batch data
- Unit identifier
- Mode switch / enabling signals

Parameters of the CmData data structure connection

Structure element	***Description***	***Type***
`dwUnitID`	Unit ID	`DWORD`
`dwBatchID`	Batch ID	`DWORD`

sBatchName	Batch name	STRING [32]
dwBatchStepNo	Batch step number	DWORD
xBatchOcc	Batch occupied	BOOL
xBatchEn	Batch enabled	BOOL
xSimEnCM	Übergeordnete Simulation ist freigegeben	BOOL
xSimOnParent	Übergeordnete Simulation ist aktiv	BOOL
xEqmLock	Sperren EQM / unterlagerte CMs	BOOL
xManOpEn	Operator kann in MANUAL mode schalten	BOOL
xResetPls	Reset signal (Impuls)	BOOL
xModLiOp	Umschaltung linked/operator Betrieb	BOOL
xAutModLi	Kommando AUTO mode	BOOL
xManModLi	Kommando MANUAL mode	BOOL
xMsgLock	Sperren Meldungen	BOOL
xRes1		BOOL
xRes2		BOOL
xRes3		BOOL
xRes4		BOOL
xRes5		BOOL
byFbkDelayRatio	Faktor Rückmeldeverzögerung (in %)	BYTE

Note:

This information can also be switched groupwise to external inputs via various "Overwrite" signals in order to allow the CM-individual control:

The following table lists the individual signals per category as well as the associated switchover signals:

Category	*Input signal*	*Switchover signal*
Unit-Identifier	UnitID	OvwUnitID
Batch data	BatchID BatchName StepNo	OvwBatchInfo

	BatchOcc BatchEn	
Manual switching operation	ManOpEn	OvwManOpEn
Mode switch	ModLiOp AutModLi	OvwModSel
Controls	(Typspezifisch)	OvwCommands

The following signals cannot be switched over:

- Simulation enable
- Signal "Higher-level simulation is active"
- EQM lock
- Reset pulse
- Request for occupation and downstream function designation (only relevant for "shared equipment")
- Relationship of the feedback simulation time to the CM monitoring time

Further details can be found in the Function manual of PCS 7 BRAUMAT Library in chapter „Basics of adapter blocks“.

2 PCS 7 BRAUMAT Library – process tag types

With the PCS 7 BRAUMAT Library an Examples Project is delivered which contains the Master Data Library „FB_Example_Lib“. In folder „Templates“ a number of CFC charts is contained which covers the main control module applications with standard I/O interface as well as with Profibus PA and SIMATIC Route Control Interface. This process tag types serves solely as engineering templates and have to be changed or extended to satisfy the application’s demands. The process tag types are constructed from blocks of the PCS 7 Advanced Process Library V7.1 and the PCS 7 BRAUMAT Library.

The following information is contained in subchapters per tag type:

- General
- Used block types
- Standard connections in CFC

2.1 BIDin

Monitoring a digital process valuel without messages

This process tag type serves as a basis for monitoring a digital process tag with the block BIDin.

- The digital value is read from the I/O through the PCS7DiIn block.
- The active signal level may be preset per “Feature bit” Bit 2 (1= high level active, 2= low level active)
- The signal edge (1→ 0 or 0 → 1) may be delayed each with a configurable or linkable delay time value
- The block has no messages for minimal resource overhead

Used Block types

Block type	*Library*	*Comment*
Pcs7DiIn	PCS 7 APL-Library	Standard-driver block for digital input
BIDin	PCS 7 BRAUMAT Library	BRAUMAT digital monitoring

Signal connections in CFC-Chart

Block	Connector	Comment
Pcs7DiIn	`PV_In`	Process value input
BIDin	`CmData`	EQM status information

Note:

For message enabled digital process tags the APL-Template “DigitalMonitoring” may be used. The decoding and propagation of the SIMATIC BATCH info signals should be done with the PCS 7 BRAUMAT Library block ‘BICmDec’ in this case (see process tag type ‘BIMonAn08’ for reference).

2.2 BlMof

Configurable and combined monitoring of one digital and up to four analog values with message generation

This process tag type serves as a basis for monitoring one digital and up to four analog inputs with optional alarm function and acknowledgement. The input values can be operated or linked, with the switchover being carried automatically with the interconnection in CFC.

- The process values are read from the I/O through Pcs7DiIn and Ps7AnIn driver blocks.
- The compare and monitoring functions are configured via the 'CompType' and 'MofType' parameters (see Function manual for details).
- The result of the compare may be output as a binary signal directly or with storing behavior
- The output signal changes (1 → 0 or 0 → 1) may be delayed with configurable or linkable delay times each
- For the binary result a message can be generated with configurable class (alarm, warning, tolerance, above or below).

Used Block types

Block type	*Library*	*Comment*
Pcs7DiIn	PCS 7 APL-Library	Standard-driver block for digital input
Pcs7AnIn	PCS 7 APL-Library	Standard-driver block for analog input
BlMof	PCS 7 BRAUMAT Library	BRAUMAT configurable monitoring function

Signal connections in CFC-Chart

Block	Connector	Comment
Pcs7DiIn	PV_In	digital process value input
Pcs7AnIn	PV_In	Analog process value input
Pcs7AnIn	PV_In	Analog process value input
Pcs7AnIn	PV_In	Analog process value input
Pcs7AnIn	PV_In	Analog process value input
BlMof	CMData	EQM status information

2.3 BlMonAn08

Monitoring an analog process valuel

This process tag type serves as a basis for monitoring an analog process tag with a maximum of 8 free configurable limit values.

- The analog value is read from the I/O through the PCS7AnIn block.
- The SIMATIC BATCH information is connected as a data structure connection from the equipment module to the decoder block BlCmDec. This provides the distinct signals `UnitID, BatchID, BatchName, StepNo, BatchOcc` for the BlMonAn08 block.
- The limits may be configured individually for lower/upper limit and one of the message classes alarm, warning or status.
- The limit violation is signaled on a corresponding output (`PV_LimxAct`) and maybe supplied with an additional warning message (see chapter “message behaviour” of the function manual).
- Each limit monitoring may be delayed by the parameter `DelayTx`

Note:

For smoothing the input value, the “Smooth” block of the advanced process library may be used.

Used Block types

Block type	***Library***	***Comment***
Pcs7AnIn	PCS 7 APL-Library	Standard-driver block for analog input
BlMonAn08	PCS 7 BRAUMAT Library	BRAUMAT monitoring
BlCmDec	PCS 7 BRAUMAT Library	BRAUMAT CM-Info decoder block

Signal connections in CFC-Chart

Block	**Connector**	**Comment**
Pcs7AnIn	`PV_In`	Process value input
BlCmDec	`CmData`	EQM status information

2.4 BlMotL

Controlling a motor with feedback and protection

This process tag type can be used for motors. The motor is controlled by binary output and its status is monitored via feedback from motor or motor contactor.

- The feedback signal of the motor or motor circuit breaker is read from the I/O through Pcs7DiIn driver block.
- The interlock signals of MotL are connected to the IntLk02 interlock blocks. These interlock blocks in turn are connected to other blocks, for example, to digital process tags via Pcs7DiIn.
- The digital output signals are sent to the I/O through the PCS7DiOu blocks.
- In unit simulation mode, the feedback signal is generated from the PCS 7 BRAUMAT Library adapter block (delayed on demand)

Used Block types

Block type	*Library*	*Comment*
Pcs7DiIn	PCS 7 APL-Library	Standard-driver block for digital input
Pcs7DiOu	PCS 7 APL-Library	Standard-driver block for digital output
BlMotL	PCS 7 BRAUMAT Library	BRAUMAT adapter block for motor control
MotL	PCS 7 APL-Library	APL motor control block
IntLk02	PCS 7 APL-Library	APL interlock with 2 inputs

Signal connections in CFC-Chart

Block	Connector	Comment
Pcs7DiIn	PV_In	Feedback On
BlMotL	CMData	EQM status information
Intlk02	In01	Activation enable ("Permission"):
Intlk02	In01	Interlock without reset ("Interlock"):
Intlk02	In01	Interlock with reset ("Protection"):
Pcs7DiOu	PV_Out	Command motor on

2.5 BlMotRevL

Controlling a motor with two directions

This process tag type can be used for handling reverse motors with the APL block MotRevL. The motor is controlled via command two binary outputs and relevant feedbacks for motor running and direction.

- The feedback signals of the motor or motor circuit breaker are read from the I/O through Pcs7DiIn driver blocks.
- The interlock signals of MotRevL are connected to the IntLk02 interlock blocks. These interlock blocks in turn are connected to other blocks, for example, to digital process tags via Pcs7DiIn.
- The digital output signals are sent to the I/O through the PCS7DiOu blocks.
- In unit simulation mode, the feedback signals are generated from the PCS 7 BRAUMAT Library adapter block (delayed on demand)

Used Block types

Block type	*Library*	*Comment*
Pcs7DiIn	PCS 7 APL-Library	Standard-driver block for digital input
Pcs7DiOu	PCS 7 APL-Library	Standard-driver block for digital output
BlMotRevL	PCS 7 BRAUMAT Library	BRAUMAT adapter block for reversed motor control
MotRevL	PCS 7 APL-Library	APL motor control block with two directions
IntLk02	PCS 7 APL-Library	APL interlock with 2 inputs
Or04	PCS 7 APL-Library	APL digital logic block

Signal connections in CFC-Chart

Block	**Connector**	**Comment**
Pcs7DiIn	PV_In	Feedback forward
Pcs7DiIn	PV_In	Feedback reverse
BlMotRevL	CMData	EQM status information
Intlk02	In01	Activation enable ("Permission"):
Intlk02	In01	Interlock without reset ("Interlock"):
Intlk02	In01	Interlock with reset ("Protection"):
Pcs7DiOu	PV_Out	Command motor forward
Pcs7DiOu	PV_Out	Command motor reverse

2.6 BlMotS

Controlling a motor with feedback and protection

This process tag type can be used for motors. The motor is controlled by binary output and its status is monitored via feedback from motor or motor contactor.

- The feedback signal of the motor or motor circuit breaker is read from the I/O through Pcs7DiIn driver block.
- The interlock signals of MotS are connected to the IntLk02 interlock blocks. These interlock blocks in turn are connected to other blocks, for example, to digital process tags via Pcs7DiIn.
- The digital output signals are sent to the I/O through the PCS7DiOu blocks.
- In unit simulation mode, the feedback signal is generated from the PCS 7 BRAUMAT Library adapter block (delayed on demand)

Used Block types

Block type	*Library*	*Comment*
Pcs7DiIn	PCS 7 APL-Library	Standard-driver block for digital input
Pcs7DiOu	PCS 7 APL-Library	Standard-driver block for digital output
BlMotL	PCS 7 BRAUMAT Library	BRAUMAT adapter block for motor control
MotS	PCS 7 APL-Library	APL motor control block
IntLk02	PCS 7 APL-Library	APL interlock with 2 inputs

Signal connections in CFC-Chart

Block	Connector	Comment
Pcs7DiIn	`PV_In`	Feedback On
BlMotL	`CMData`	EQM status information
Intlk02	`In01`	Interlock without reset ("Interlock"):
Pcs7DiOu	`PV_Out`	Command motor on

2.7 BlMotSpdCL

Motor with variable speed and two directions

This process tag type can be used for handling of variable speed motors with two directions via the APL block MotSpdCL.

- The direction feedback signals of the motor are read from the I/O through Pcs7DiIn driver blocks.
- The speed feedback signal is read from the I/O through Pcs7AnIn driver block.
- The interlock signals of MotSpdCL are connected to the IntLk02 interlock blocks. These interlock blocks in turn are connected to other blocks, for example, to digital process tags via Pcs7DiIn.
- The digital and analog output signals are sent to the I/O through the PCS7DiOu/ PCS7AnOu blocks.
- In unit simulation mode, the direction feedback signals are generated from the adapter block (delayed on demand). The speed feedback is derived from the speed setpoint in this case.

Used Block types

Block type	*Library*	*Comment*
Pcs7DiIn	PCS 7 APL-Library	Standard-driver block for digital input
Pcs7AnIn	PCS 7 APL-Library	Standard-driver block for analog input
Pcs7DiOu	PCS 7 APL-Library	Standard-driver block for digital output
Pcs7AnOu	PCS 7 APL-Library	Standard-driver block for analog output
BlMotSpdCL	PCS 7 BRAUMAT Library	BRAUMAT adapter block for speed controlled motor
MotSpdCL	PCS 7 APL-Library	APL motor control block for speed controlled motor
IntLk02	PCS 7 APL-Library	APL interlock with 2 inputs
Or04 / Or08	PCS 7 APL-Library	APL digital logic block

Signal connections in CFC-Chart

Block	Connector	Comment
Pcs7DiIn	PV_In	Feedback forward
Pcs7DiIn	PV_In	Feedback reverse
Pcs7AnIn	PV_In	Speed feedback actual value
BlMotSpdCL	CMData	EQM status information
Intlk02	In01	Activation enable ("Permission"):
Intlk02	In01	Interlock without reset ("Interlock"):
Intlk02	In01	Interlock with reset ("Protection"):
Pcs7DiOu	PV_Out	Command motor forward
Pcs7DiOu	PV_Out	Command motor reverse
Pcs7AnOu	PV_Out	Setpoint value motor speed

2.8 BlPidConL

Extended PID control

This process tag type forms the basis for generating PID control instances for continuous processes. In addition to the actual PID block, it contains functionality which are implemented in the PCS 7 BRAUMAT Library adapter block BlPidConL. The process tag contains the following functionality:

- An analog input channel block for the actual value PV plus an analog output channel block for the manipulated variable MV.
- Certain logic blocks that change the control loop to a safe mode if the measurement of the process value fails. This state is signaled by the signal status of PV.
- With the BRAUMAT adapter block BlPidConL 5 different tracking values may be applied (1x `ExtTrkVal`, 4x directly or time controlled)

Note:

Read the information for process tag types with controller blocks in “Introduction to process tag types” in the APL-documentation.

Used Block types

Block type	***Library***	***Comment***
Pcs7AnIn	PCS 7 APL-Library	Standard-driver block for analog input
Pcs7AnOu	PCS 7 APL-Library	Standard-driver block for analog output
BlPidConL	PCS 7 BRAUMAT Library	BRAUMAT adapter block for APL PidConL contoller block
PidConL	PCS 7 APL-Library	APL continuous controller block
Or04	PCS 7 APL-Library	APL digital logic block

Signal connections in CFC-Chart

Block	**Connector**	**Comment**
Pcs7AnIn	`PV_In`	Process value input
BlPidConL	`CMData`	EQM status information
Pcs7AnOu	`PV_Out`	Setpoint value output

2.9 BlVlvAnL

Motor Valve with position control output and two feedbacks

This process tag type serves as a basis for controlling a motor valve (0…100%) using the APL VlvMotL block.

- The feedback signals of the valve is read from the I/O through Pcs7DiIn driver blocks.
- The feedback signal of the analog position is read from the I/O through the PCS7AnIn block.
- The interlock signals of VlvAnL are connected to the IntLk02 interlock blocks. These interlock blocks in turn are connected to other blocks, for example, to digital process tags via Pcs7DiIn.
- The analog output signals is sent to the I/O through the PCS7AnOu block.
- In unit simulation mode, the feedback signal is generated from the adapter block (delayed on demand)

Used Block types

Block type	*Library*	*Comment*
Pcs7DiIn	PCS 7 APL-Library	Standard-driver block for digital input
Pcs7AnIn	PCS 7 APL-Library	Standard-driver block for analog input
Pcs7AnOu	PCS 7 APL-Library	Standard-driver block for analog output
BlVlvAnL	PCS 7 BRAUMAT Library	BRAUMAT adapter block for analog valve
VlvAnL	PCS 7 APL-Library	APL analog valve control block
IntLk02	PCS 7 APL-Library	APL interlock with 2 inputs

Signal connections in CFC-Chart

Block	**Connector**	**Comment**
Pcs7DiIn	PV_In	Feedback open
Pcs7DiIn	PV_In	Feedback close
Pcs7AnIn	PV_In	Feedback position
BlVlvAnL	CMData	EQM status information
Intlk02	In01	Activation enable ("Permission"):
Intlk02	In01	Interlock without reset ("Interlock"):
Intlk02	In01	Interlock with reset ("Protection"):
Pcs7AnOu	PV_Out	Output valve position (0…100%)

2.10 BlVlvL

Valve with one control output (open/close) and two feedbacks

This process tag type can be used for valves with Open/Close position with the APL block VlvL.

- The feedback signal of the valve is read from the I/O through Pcs7DiIn driver blocks.
- The interlock signals of VlvL are connected to the IntLk02 interlock blocks. These interlock blocks in turn are connected to other blocks, for example, to digital process tags via Pcs7DiIn.
- The digital output signals are sent to the I/O through the PCS7DiOu blocks.
- In unit simulation mode, the feedback signal is generated from the adapter block (delayed on demand)

Used Block types

Block type	*Library*	*Comment*
Pcs7DiIn	PCS 7 APL-Library	Standard-driver block for digital input
Pcs7DiOu	PCS 7 APL-Library	Standard-driver block for digital output
BlVlvL	PCS 7 BRAUMAT Library	BRAUMAT adapter block for valve controls
VlvL	PCS 7 APL-Library	APL valve control block
IntLk02	PCS 7 APL-Library	APL interlock with 2 inputs

Signal connections in CFC-Chart

Block	Connector	Comment
Pcs7DiIn	PV_In	Feedback open
Pcs7DiIn	PV_In	Feedback close
BlVlvL	CMData	EQM status information
Intlk02	In01	Activation enable ("Permission"):
Intlk02	In01	Interlock without reset ("Interlock"):
Intlk02	In01	Interlock with reset ("Protection"):
Pcs7DiOu	PV_Out	Command valve open/close

2.11 BlVlvS

Valve with one control output (open/close) and two feedbacks

This process tag type can be used for valves with Open/Close position with the APL block VlvL.

- The feedback signal of the valve is read from the I/O through Pcs7DiIn driver blocks.
- The interlock signal of VlvS is connected to the IntLk02 interlock block. This interlock block in turn is connected to other blocks, for example, to digital process tags via Pcs7DiIn.
- The digital output signals are sent to the I/O through the PCS7DiOu blocks.
- In unit simulation mode, the feedback signal is generated from the adapter block (delayed on demand)

Used Block types

Block type	*Library*	*Comment*
Pcs7DiIn	PCS 7 APL-Library	Standard-driver block for digital input
Pcs7DiOu	PCS 7 APL-Library	Standard-driver block for digital output
BlVlvL	PCS 7 BRAUMAT Library	BRAUMAT adapter block for valve controls
VlvS	PCS 7 APL-Library	APL valve small control block
IntLk02	PCS 7 APL-Library	APL interlock with 2 inputs

Signal connections in CFC-Chart

Block	Connector	Comment
Pcs7DiIn	PV_In	Feedback open
Pcs7DiIn	PV_In	Feedback close
BlVlvL	CMData	EQM status information
Intlk02	In01	Interlock without reset ("Interlock"):
Pcs7DiOu	PV_Out	Command valve open/close

2.12 BlVlvDS

Double seat valve with four control outputs and two feedbacks

This process tag type can be used for double seat valves with feedback for close/open position and two digital outputs for valve control and two lifting digital outputs with the block VlvDS.

- The feedback signals (open/close) of the double seat valve is read from the I/O through Pcs7DiIn driver blocks.
- The interlock signals of VlvDS are connected to the IntLk02 interlock blocks. These interlock blocks in turn are connected to other blocks, for example, to digital process tags via Pcs7DiIn.
- The digital output signals (open/close, SLTop, SLBottom) are sent to the I/O through the PCS7DiOu blocks.
- In unit simulation mode, the feedback signal is generated from the adapter block (delayed on demand)

Used Block types

Block type	*Library*	*Comment*
Pcs7DiIn	PCS 7 APL-Library	Standard-driver block for digital input
Pcs7DiOu	PCS 7 APL-Library	Standard-driver block for digital output
BlVlvDS	PCS 7 BRAUMAT Library	BRAUMAT adapter block for valve controls
VlvDS	PCS 7 APL-Library	BRAUMAT double seat valve
IntLk02	PCS 7 APL-Library	APL interlock with 2 inputs

Signal connections in CFC-Chart

Block	**Connector**	**Comment**
Pcs7DiIn	PV_In	Feedback open
Pcs7DiIn	PV_In	Feedback close
BlVlvDS	CMData	EQM status information
Intlk02	In01	Activation enable ("Permission"):
Intlk02	In01	Interlock without reset ("Interlock"):
Intlk02	In01	Interlock with reset ("Protection"):
Pcs7DiOu	PV_Out	Command valve open/close
Pcs7DiOu	PV_Out	Inverted command valve open/close
Pcs7DiOu	PV_Out	Command valve seat lifting top
Pcs7DiOu	PV_Out	Command valve seat lifting bottom

2.13 BlVMan

Manual operated valve with two feedback signals

This process tag type can be used for manual valves with two positions (open/close) and feedback signals for close and open position. The process tag type is based the APL block VlvS.

- The feedback signals of the valve are read from the I/O through Pcs7DiIn driver blocks.
- As no delay times for control and monitoring or higher ranking simulation mode, maintenance data are used in this application, there is no BRAUMAT adapter block used.
- The unit related batch information in the `CmData` structure is routed via the BRAUMAT decoder block BlCmDec on to the valve block.
- In order that the APL block VlvS behaves like a manually operated valve (flap, repositionable bow) without control outputs, it must be forced to 'Local mode'. This is done via preset of the following input parameters:
  - `LocalLi = 1` → „Local mode" activating only in CFC
  - `LocalSetting = 5` → dynamic monitoring active and operation in faceplate limited (only monitoring time may be changed)
- The monitoring time `MonTiDynamic` should be set to >= 10sec
- It is suggested, to deactivate the following operating permissions
  Bit 9 = 0 → operator can not deactivate the dynamic monitoring
  Bit 19 = 0 → operator can not switch to local simulation

Used Block types

Block type	*Library*	*Comment*
Pcs7DiIn	PCS 7 APL-Library	Standard-driver block for digital input
VlvS	PCS 7 APL-Library	APL valve control block small

Signal connections in CFC-Chart

Block	Connector	Comment
Pcs7DiIn	`PV_In`	Feedback open
Pcs7DiIn	`PV_In`	Feedback close
BlCmDec	`CMData`	EQM status information

2.14 Control modules with SIMATIC Route Control

2.14.1 General

These process tag types supports control of motors and valves with PCS 7 BRAUMAT Library via SIMATIC Route Control.

The CFC charts of this process tag types corresponds in principle to the 'normal' process tags with the difference of one additional RC-IF block for each control output. Here the following signal connections are of concern:

- Control signal `QAUTO_OC`
  routing from RC_IF–block (instead of EQM-SFC) → BRAUMAT adapter → APL-CM.
- Batch info signals (`QBA_EN, QOCCUPIED, QBA_ID, QBA_NAME, QSTEP_NO`)
  routing from RC-IF block (instead of BlEqmIf) → BRAUMAT Adapter → APL-CM.
  Note:
  Set the adapter switchover parameter `OvwBatch=1` for this purpose.
- Feedback signals (`FbkOpenOut, FbkCloseOut, AutAct, LockAct, GrpErr`)
  routing from APL-CM → RC_IF-Baustein

2.14.2 List of RCS-process tag types

BlMotL_RCS	motor with two feedbacks and RCS
BlMotS_RCS	motor 'small' with two feedbacks and RCS
BlMotRevL_RCS	Reversible Motor with two feedbacks and RCS
BlMotSpdCL_RCS	Speed controlled Motor with two feedbacks and RCS
BlVlvL_RCS	valve with two feedbacks and RCS
BlVlvS_RCS	valve 'small' with two feedbacks and RCS
BlVlvDS_RCS	double seat valve with seat lifting top/bottom, two feedbacks and RCS

2.14.3 BITransfer – skeletal structure for transfer control

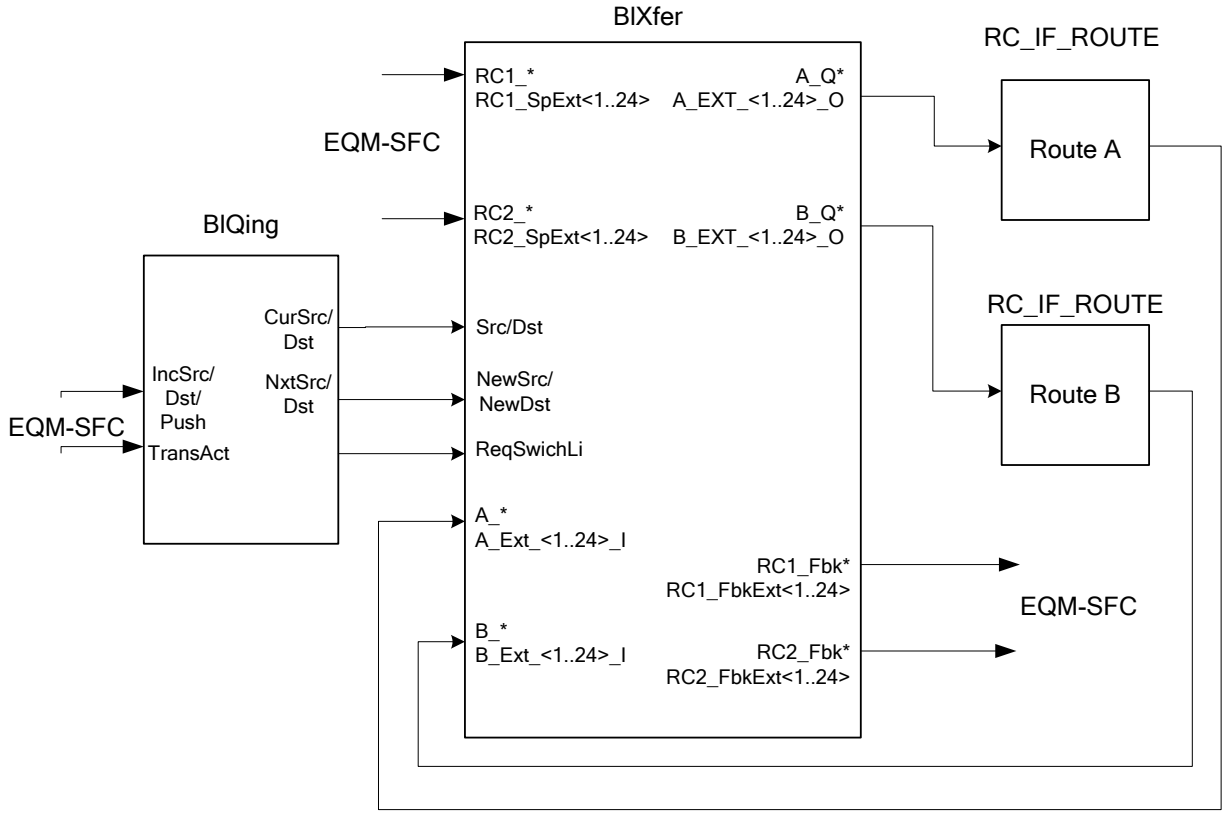
General

This CFC-Chart may be used as a basis for material transfer with SIMATIC Route Control with a transfer-/pushout-list via the PCS 7 BRAUMAT Library block BIQing and a ‘bumpless switchover’ with the block BIXfer. For a full working transfer application a EQM-SFC is needed, which is not contained in this CFC-Template. But the demo project shows an example of an complete transfer EQM based on this CFC which contains the SFC-type and its connections (chapter ‘XFER2_EX’).

Used block types

Blocktype	Library	Comment
BIQing	PCS 7 BRAUMAT Library	BRAUMAT queue transfer control block
BIXfer	PCS 7 BRAUMAT Library	BRAUMAT transfer routes control block
RC_IF_ROUTE	RC Library	SIMATIC Route Control Interface block

Routing prinziple



The two routes are controlled through the Route Control interface blocks RC_IF_ROUTE. Activation of these blocks is carried out through the 'BIXfer' block. The first RC_IF_ROUTE block controls the active route, the second route block is started in standby mode and is only still waiting for its activation. After the switchover through block ‚BIQing’ (`ReqSwitchLi`) the two RC_IF_ROUTE blocks swap their roles. The second block becomes the master, the first one becomes the standby and can now have new values for source / destination assigned to it. As the 'BIXfer' block handles this mechanism internally and transparently, the upstream BIQing block thus operates only on one fixed interface instead of on the two blocks with the switchover.

Note that the specified blocks of this sequence including EQM-SFC have to run in the same OB to ensure that the modes and controls requested by the EQM SFC are not lost before the RCS interface block is reached. The installation sequence is:

→ EQM-SFC → BlQing → BlXfer → RC_IF_ROUTE (A) → RC_IF_ROUTE (B)

The required CFC connections between the BlQing / BlXfer block as well as to the RCS–IF blocks are manifold and are specified in the supplied CFC template.

2.15 Control modules with Fieldbus Interface

2.15.1 General

Various types of channel blocks are provided in this PCS 7 BRAUMAT Library in connection with the PCS 7 APL library processing signals from inputs and outputs.

Standard Channel blocks

This applies to the following blocks:

- Pcs7AnIn
- Pcs7AnOu
- Pcs7DiIn
- Pcs7DiOu
- Pcs7DiIT

These blocks are used only for processing the signals of S7-300/400 SM modules. Use these standard blocks if you want to optimize memory and runtime utilization and do not need to process any PA devices.

FF/PA channel blocks

This applies to the following blocks:

- FbAnIn
- FbAnOu
- FbDiIn
- FbDiOu

These blocks are designed especially for use with PA field devices and the PROFIBUS 3.0 Class A and B or with FF field devices. In particular, you should use these blocks if you want to make use of the special features of these devices. In contrast to standard channel blocks, PA channel blocks not only process the signal itself but also all variables, according to the desired device configuration selected in the hardware configuration.

2.15.2 Process tags with FF/PA-channel driver blocks

When you adopt these channel driver blocks for the process tag types introduced in the earlier chapters, the following adaptation is necessary:

- Copy the process tag type and replace the applied channel blocks Pcs7DiIn, Pcs7DiOut, Pcs7AnIn, Pcs7AnOut with the accordant fieldbus interface blocks FbDiIn, FbDiOut, FbAnIn, FbAnOut
- Feedback signals as well as process values (`FbkOpen, FbkClose, PV, Rbk`) on the PCS 7 BRAUMAT Library adapter blocks or APL-blocks have to be linked with the corresponding channel driver outputs.
- Output signals for the CM-commands and for maintenance mode (`MS_Release`) on the APL-block must be linked to the corresponding inputs of the channel In-/Output driver blocks.
- Output signals for higher ranking simulation on the BRAUMAT adapter block have to be linked to the corresponding channel driver In-/Output driver blocks.
- Inputs of the OR gates for „Out of service“ (OR_Oss) and „peripheral error“ have to be linked to the corresponding channel driver outputs (`OosAct, Bad`).

3 PCS 7 BRAUMAT Library – example project

3.1 Hard- and Software-Requirements

To use the Example-Project of PCS 7 BRAUMAT Library, at least the following SIMATIC SW-Packages are required:

- SIMATIC PCS 7 Version 7.1 SP3 (includes the APL-Library V7.1 SP5)
  alternatively:
  SIMATIC PCS 7 Version 8.0 Upd1 (includes the APL-Library V8.0 Upd1)
- SIMATIC Route Control V7.1 SP2 / V8.0 Upd1
- SIMATIC BATCH Version 7.1 SP2 / V8.0 Upd1
- PCS 7 BRAUMAT Library V7.1 Upd1

Furthermore the following Hardware-Components are needed:

- PG/PC (≥ 2 GHz CPU; ≥ 1024 MByte RAM; ≥ 120 GB Harddisk).
- One Automation System with CPU 416 or CPU 417 and at least with 8 MB Memory .
- One Ethernet-LAN-Port
- Single Monitor with a Resolution of 1280 x 1024 or higher

3.2 Project Structure

The main characteristics of the example project are:

- Single Station project with ES/OS PC-Station and one SIMATIC 400-Station
- The projekt is delivered as ZIP-Archive . The STEP 7 Multiproject contains the components:
  - ‚BL71_PRJ' – Example project with
    1 Simatic 400 Station
    1 PG/PC Station with BATCH-, RC- and WinCC Applications
  - ‚BL71_LIB' – Master Data Library with process tag types for the PCS 7 BRAUMAT Library blocks

3.3 Adapting the Example project

3.3.1 Retrieving the example project

Procedure

1. Open the SIMATIC Manager, go to the "File" menu and select the "Retrieve" command.
2. After installation of the PCS 7 BRAUMAT Library, the archived example project can be found at **..\SIEMENS\STEP7\EXAMPLES_MP\BL71_EXA.zip.**
3. Select the archived project and hit "open".
4. Specify the storage location for the archive.
5. Confirm the summary by hitting "open".
6. Open the retrieved project.

3.3.2 Adapting the project

Testing and adapting the project settings

After installing the project, the following project settings should be tested and adjusted where necessary.

Procedure

1. In SIMATIC Manager, change the display name and computer name in the SIMATIC PC Station properties.
2. Open the WinCC OS and replace the given computer name with the computer name of your Windows PC. (When the message regarding the non-existent server appears, confirm this by hitting "Yes").
3. In the Hardware configuration, replace the configured PS (Power Supply) with the actual Hardware available:
   - Delete the configured power supply module from the configuration.
   - Replace it with the actual module fitted.
4. In the Hardware configuration, replace the configured CPU with the actual Hardware fitted:
   - Save the S7 program, which is located in the project below the CPU.
   - Delete the configured CPU from the configuration.
   - Replace the CPU with the one that is actually in place.
   - Copy the S7 program back to the CPU. Do not leave any other copies of the S7 program in the project.
5. In AS hardware configuration, replace the configured communication card with the actual hardware fitted:
   - Delete the configured communication card (CP) from the configuration.
   - Replace the CP with the one actually fitted.
   - Replace the configured IP and MAC addresses with the actual addresses.
6. Compile the AS in hardware configuration (Menu: Station → Save and compile).
7. Open the PC station's hardware configuration and replace the communication card and/or the IP and MAC addresses configured in the project for the communication card with the actual addresses. Connect the CP with a configured net.
8. Compile the PC station in the hardware configuration.
9. Start the component configurator (Start → Component configurator) and import the *.xdb project file. This can be found in the project folder under "...\XDBs*.xdb".

3.3.3 Adjusting NetPro settings

When changing a CPU …

If you have changed the CPU in the example project, please execute the following procedure.

1. Open NetPro via the symbol in the SIMATIC Manager toolbar.
2. Right-click on CPU in the SIMATIC 400 Station and select "Add new connection"in the context menu.
3. In the next dialog, select the “WinCC Application”, click on "OK" and accept the connection properties displayed.
4. Repeat this action with “RC Application” and “BATCH Appl.”
5. Save, compile and load the settings.

3.3.4 Compiling and loading the SIMATIC 400 Station

Procedure

1. Open the CFC-Editor via double click on any CFC-chart
2. Open the ‘Compile program’ dialog and select the ‘Entire Program’ box
3. Open the ‘Download’ dialog, select the ‘Entire Program’ box and start the download process
4. Start the AS.

3.3.5 Transferring data to the PCS 7 OS

Procedure

1. Right-click on the OS object of the WinCC Application and select "Compile" in the context menu.
2. Confirm the following dialogs with ‘Next’ and ‘Finish’ Buttons.
3. The WinCC-application is to be compiled
   (Note: OS downloading is not necessary in this Single Station configuration)

3.3.6 Setting up the SIMATIC BATCH Servers

Introduction

SIMATIC BATCH Server setup is performed in two stages. The first part is carried out in the batch configuration dialog and the second part in the Batch Control Center.

Transferring the data to the SIMATIC BATCH Server

1. Right-click on an element in the plant view of the project and select the "SIMATIC BATCH" command in the context menu.
2. In the SIMATIC BATCH configuration dialog, perform the following steps one after the other:
   - Adjust the memory expansion and the allocation to the actual CPU memory available. Navigate to the SIMATIC 400 → CPU 417-4 station in the tree topology.
   - Generate the "batch types"
   - Compile the "batch instances"
   - Transfer the batch messages to the OS.
   - Load the "Batch process cell" in the OS.
3. Compile in CFC the AS program again and download it into the AS

Batch process cell (PCell) setup

Note:

The Start coordinator must be running (START → SIMATIC → BATCH → BATCH Launch Coordinator) in order to open the BCC. The Startup may take a few minutes.

1. Open the Batch Control Center (START → SIMATIC → BATCH → BATCH Control Center)
2. From the menu, select the Extras → Restore command.
3. Open the batch backup file **"BL_EXA.sbb"**, which is supplied with the project. The file is located in the installation path under ..\SIEMENS\STEP7\EXAMPLES_MP\....
4. Role - Definition after Restore
   Please replace the user displayed in the project under "Super user" with a user you have allocated as a "Super user". Then log in to the BATCH Control Center with the login data for the newly created user
   Please consider the notes in chapter 3.6.1 ‘Starting the examples project’
5. Right-click on your batch process cell in the project structure and select the "Update process cell" command in the context menu.

3.3.7 Setting up the SIMATIC RCS Server

Introduction

Use the RCS-Wizard to perform all possible actions. The condition is that the project including a valid CFC chart with one RCS configuration block (RC_IF_CFG)

Procedure

1. Select the menu item "Options → SIMATIC Route Control → Wizard" menu command. The Route Control Wizard opens.
2. The "Route Control Configuration" dialog box is opened if the Route Control does not yet contain a configuration for this S7 project. Click "Yes" → The database is created. The data is read from the S7 project. The "Introduction" step of the Route Control Wizard is opened.
3. Click "Next". The step for selecting the actions opens. Activate **all** check boxes.
4. Click "Next". The step for selecting the message server opens. Activate the check box for the message server
5. Click "Next". The step for selecting the RC-objects opens. Activate **all** check boxes.
6. Click "Next". The list of objects found opens.
7. Click "Next". The step for displaying the selected options opens.
8. Click "Finish". The selected actions are executed. The dialog now views the “Modifications made” as well as the “Following actions to apply the changes” is displayed.
9. Click on button "Display logs" to view the log file in a text editor.
10. Please examine the protocol. You may find the protocol also in SIMATIC Manager via the "Options → SIMATIC Route Control → Show wizard log…" menu item
11. Close the logfile.
12. Click the ‘Close’ button

At this point, please execute the command(s) given by the last Route Control Wizard dialog in the former step, for example:

- Compile and download CFC changes
- Compile and download NetPro
- PCS 7 OS download changes
- Download and update Route Control server …

For the last item please perform the following steps:

1. Select the "Start → SIMATIC → ROUTE CONTROL → Server" menu command. The Route Control Server is started and displayed as icon in the task bar. Optionally WinCC Runtime must be started in order to Start Up the RCS Server
2. In the SIMATIC manager select the "Options → SIMATIC Route Control → Engineering" menu command. The Route Control engineering tool opens:
   - Select the "Options → Download" menu command in Route Control Engineering. The "Load Server" dialog box opens.
   - Click "Start". The server is loaded.
   - Click "OK" in the next dialog box. The dialog box is closed.
3. Select the "Start → SIMATIC → ROUTE CONTROL → Center" menu command. The Route Control Center opens.
   - Click the Server Status icon in the Route Control Center. The "SIMATIC RC Server" dialog box opens.

- Click "Update". The "Update SIMATIC RC Server" dialog box opens.
- Click "OK". The Route Control Server is updated.
- Click "Close".
- Click “Restart” in the "SIMATIC RC Server" dialog box. The Route Control Server should establish a connection and go to status RUN

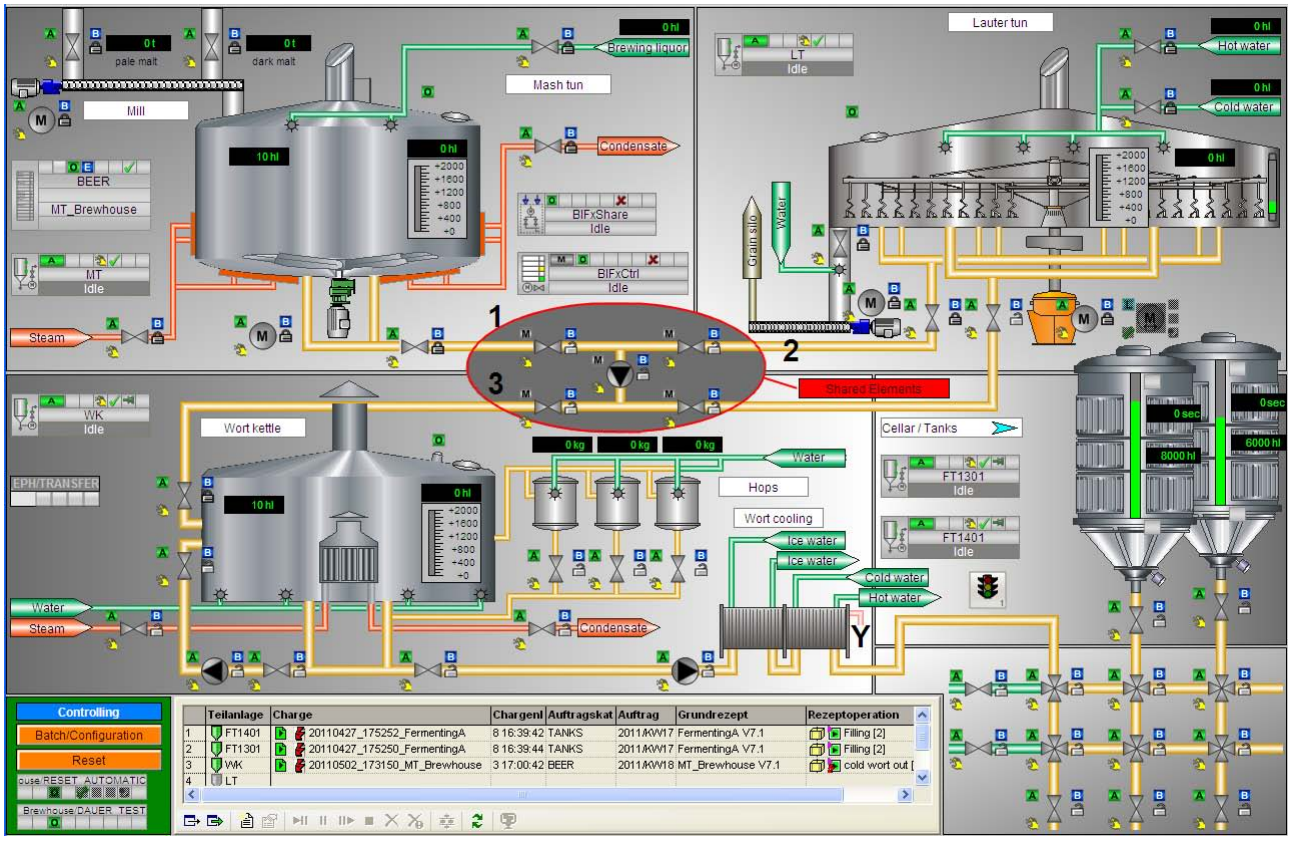
Maybe you have to restart the AS again, if server get now connection

3.4 Physical model and technological function

Besides the resources for testing and diagnostics of the PCS 7 BRAUMAT Library, the example project contains a Brewery application sample with some typical Units. These are leaned against real plants as far as possible concerning the equipment engineering as well as process execution, however the detailed degree was constrained due to the test- and demonstration nature as well as HW-configuration available for the project.

The underlying physical model and the technological function shall be outlined based on the process pictures delivered with the project.

3.4.1 Process picture ‘Brewhouse/BrewProcess’

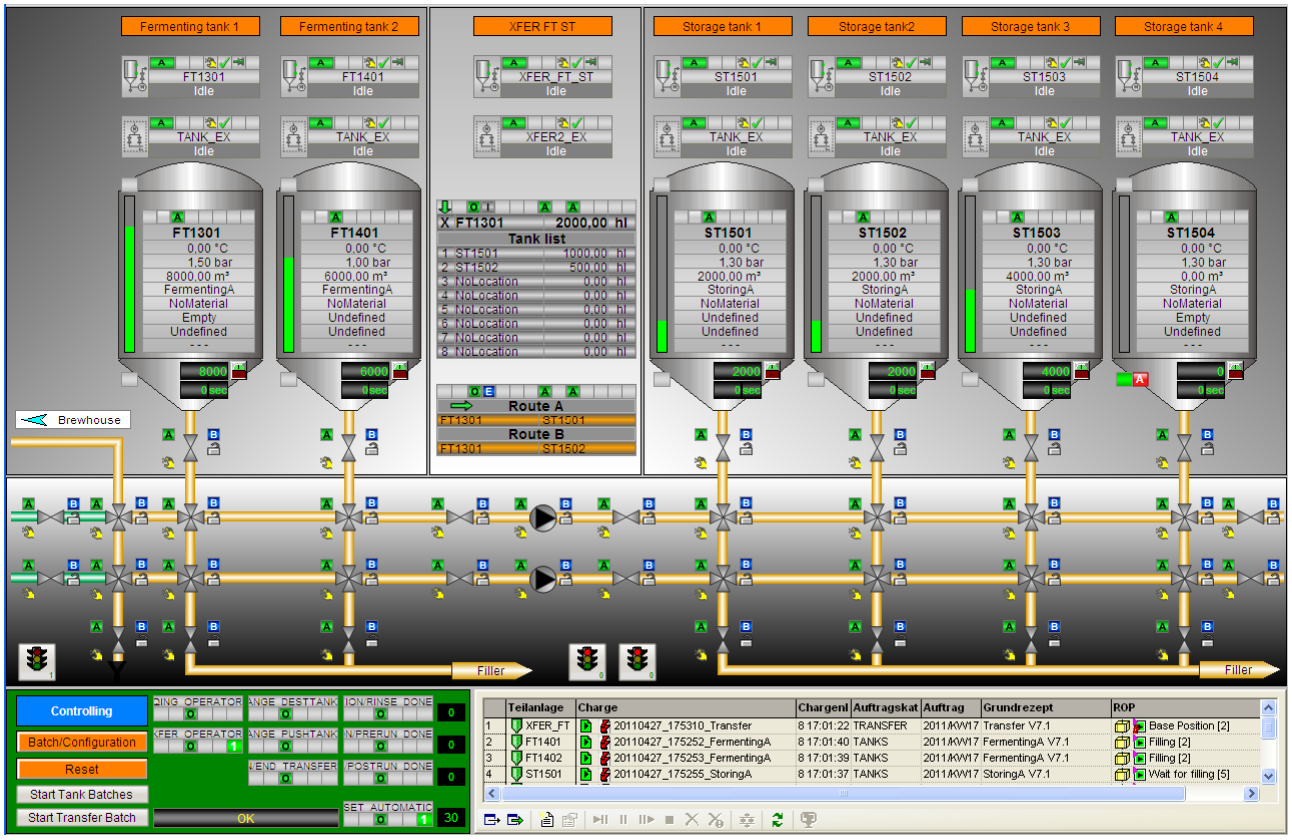


The wort production begins with the unit ‘mash vessel’ with valves MT_INLET_V02, V03 for dosing the pale and dark malt grist as well as water MT_INLET_V01. Steam is fed with the valves MT_HEAT_V04, V05 for heating. A motor MT_AGITATE_M01 agitates the mash during heating. The mash vessel is emptied into the ‘lauter vessel’ via MT_OUTLET_V06, V07, the Shared Elements like pump SH_OUTLET_P01 and the valves SH_OUTLET_V01, V04. After lautering, the product is pumped into the ‘wort cooker’ via LT_OUTLET_V04, V05, the Shared Elements Pump SH_OUTLET_P01 and the valves SH_OUTLET_V02, V03. After boiling the wort (WK_HEAT_V05, V06) hops are added (WK_HOPS_V01, V02, V03) in several stages and afterwards the wort clarification takes place via WK_WHIRLPOOL_V08, P01, V04. Then the finished wort is transferred into one of two Fermentation Tanks via the wort-cooler (WK_OUTLET_V07, P01). The participating aggregates (several valves and one pump) are controlled here with a requested Route in ‚SIMATIC Route Control’. The Fermenting Tank selection takes place at recipe start via recipe parameters.

Depending on pre selection (see process picture ‘BrewhouseBatch’ Select Box ‘Create and start automatic Fermenter-Batch ?’) the appropriate Fermentation-Batch is created and started automatically within the transfer-step or it has to be applied manually by the operator in ‘Cellar/Tank’ area.

The capacity of the vessels amounts 2.000 hl each.

3.4.2 Process picture ‘Cellar/CellarProcess’

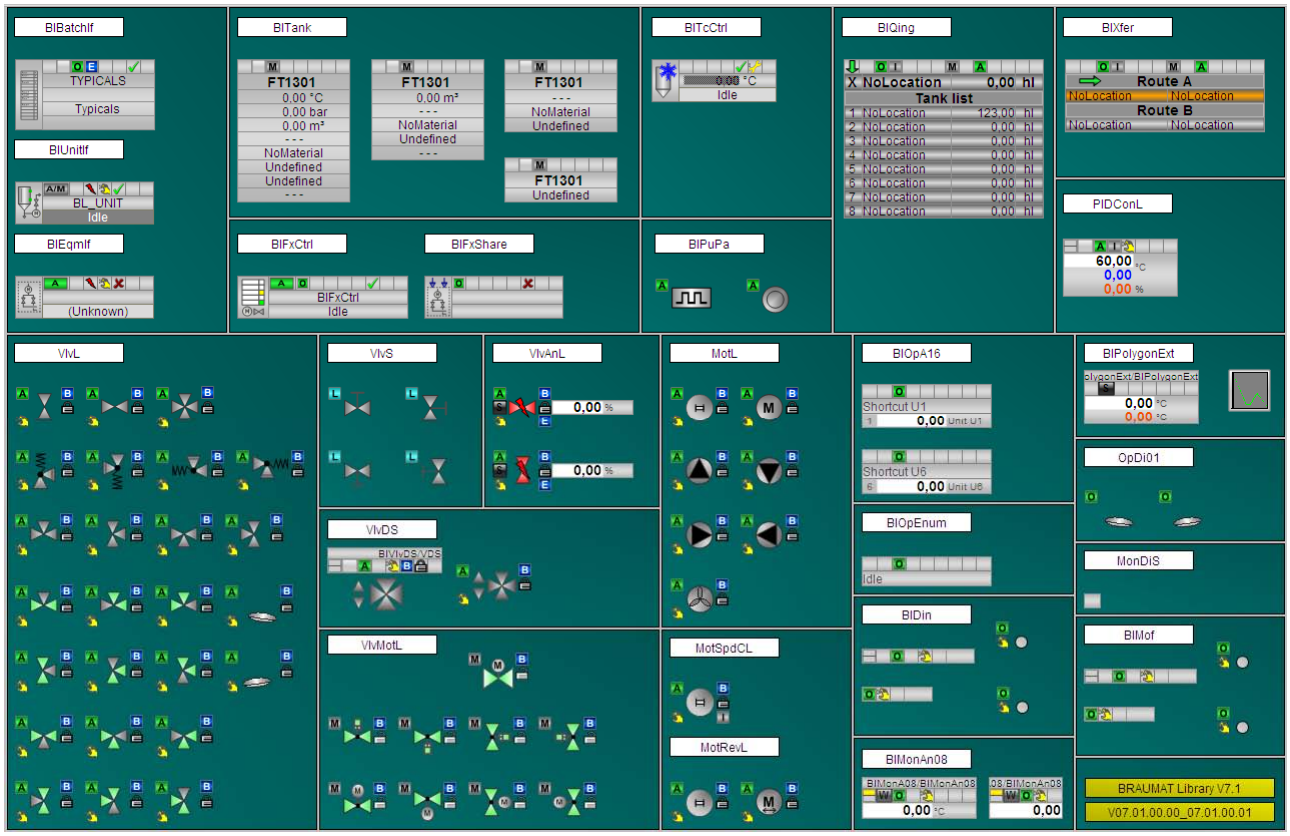


As a continuation to the Brewhouse, this process picture shows the Fermenting- and Storage-Cellar in form of a transfer picture. After fermenting the young beer is transferred from one of two Fermenting Tank units (FT 1301/1401) into one of four Storage Tank units (ST1501…1504) via several valves and pumps. The transfer unit (XFER FT ST – center of screen) is equipped with a BlQing block, which is operated in the 1:n constellation. The transfer list (1 source / up to 8 destinations and the according transfer amounts) is normally filled by recipe parameters, but may be changed in the OS-faceplate after switching into “Internal mode”. While the transfer batch is running, the routes from the source to destination tanks are requested and switched over in a bumpless manner via the BlXfer block and SIMATIC Route control.

Further details of the process picture’s display and operating elements you will find in later chapters.

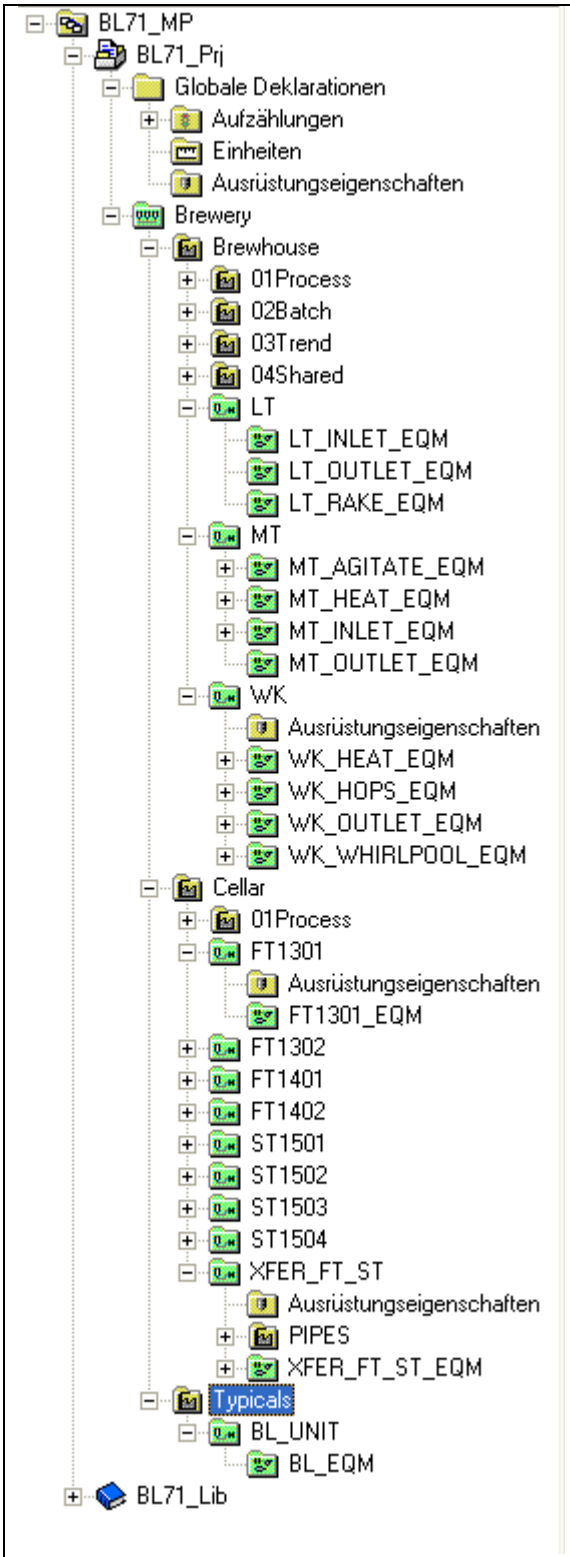
3.4.3 Process picture 'TestInstances'

This process picture serves for demonstration and test of the PCS 7 BRAUMAT Library and APL blocks. It contains the block symbols where more than one symbol could be present –according the block type- for different representations (e.g. for valves, motors, …). All symbols of same block type are linked to the same AS block instance. With click on a symbol the according OS-faceplate is opened.



3.5 Structure overview in the technological hierarchy

3.5.1 Technological hierarchy of the example project



Plant

In SIMATIC BATCH terminology the Area "Brewery" consists of the Process Cell „Brewhouse" with the units „MT" (Mashtun), "LT" (Lautertun) and "WK" (Wortkooker) and the Process Cell "Cellar" with 4 fermentation units "FTxxxx", 4 storage units „STxxxx" as well as one transfer unit "XFER_FT_ST". In the neutral folder "04Shared" there are the Shared Elements for the transfer from the "Mashtun" into the "Lautertun" and from the "Lautertun" into the "Wortcooker".

The hierarchy folders of the Process Cells have no ISA-88-Type property, they are displayed as neutral folders. The same applies to some folders containing OS process picture only (see note below).

Units

Each Unit folder contains the following components in the technological tree-view:

- CFC chart with the Unit specific blocks UNIT_PLC from SIMATIC BATCH Blocks (General\Blocks > BATCH) and BlUnitIf, BlBatchIf from PCS 7 BRAUMAT Library.
- CFC chart with blocks for Simulation
- One or more subfolders of ISA-88-Type "Equipment Module /EMOD".

Equipment Module / EQM

Each EQM folder with ISA-88-Type "EMOD" contains the CFC-charts with the control module blocks (motors, valves and Monitoring Functions) and a CFC-chart which implements the phase logic of the EQM as SFC-Type.

Note:

Deriving the Picture Hierarchy and OS Areas from the Plant Hierarchy:

The OS picture hierarchy for the operator on the OS were derived completely from the configured data for the plant hierarchy. This involves inserting the pictures in accordance with the plan structure into the plant hierarchy (PH). One picture for each hierarchy folder in the PH and per OS may be inserted for creating a picture hierarchy. Once the OS is compiled, the Picture Tree Manager has the same hierarchy for further editing.

3.5.2 Process Cell 'Brewhouse'

The following table shows the list of CFC-charts and the hierarchy levels.

3.5.2.1 CFC-charts of the technological hierarchy

UNIT-folder	UNIT-charts	EQM-folder	EQM-charts	Function
LT	LT LT_SIM…	LT_INLET_EQM	LT_INLET_EPH	Phase logic with SFC INLET_EX
			LT_INLET_V01	Inlet valve
			LT_INLET_V02	Outlet valve
		LT_OUTLET_EQM	LT_OUTLET_EPH	Phase logic with SFC OUTLET_EX
			LT_OUTLET_V04	Inlet valve
			LT_OUTLET_V05	Outlet valve
		LT_RAKE_EQM	LT_RAKE_EPH	Phase logic with SFC RAKE_EX
			LT_RAKE_M01	Drive motor
			LT_RAKE_M02	Drive motor
			LT_RAKE_M03	Drive motor
			LT_RAKE_V03	Outlet valve
MT	MT MT_SIM…	MT_AGITATE_EQM	MT_AGITATE_EPH	Phase logic with SFC AGITATE_EX
			MT_AGITATE_M01	Drive motor
		MT_HEAT_EQM	MT_HEAT_EPH	Phase logic with SFC HEAT_EX
			MT_HEAT_V04	Steam valve
			MT_HEAT_V05	Steam valve
		MT_INLET_EQM	MT_INLET_EPH	Phase logic
			MT_INLET_M01	Drive motor
			MT_INLET_V01	Inlet valve
			MT_INLET_V02	Inlet valve
			MT_INLET_V03	Inlet valve
		MT_OUTLET_EQM	MT_OUTLET_EPH	Phase logic
			MT_OUTLET_V06	Outlet valve
			MT_OUTLET_V07	Outlet valve

WK	WK WK_SIM	WK_HEAT_EQM	WK_HEAT_EPH WK_HEAT_V05 WK_HEAT_V06	Phase logic Steam valve Steam valve
		WK_HOPS_EQM	WK_HOP_EPH WK_HOPS_V01 WK_HOPS_V02 WK_HOPS_V03	Phase logic with SFC HOPS_EX Inlet valve Hops 1 Inlet valve Hops 2 Inlet valve Hops 3
		WK_OUTLET_EQM	WK_OUTLET_EPH WK_OUTLET_P01 WK_OUTLET_V07	Phase logic with SFC OUTLET_EX Pump Outlet valve
		WK_WHIRLPOOL _EQM	WK_WHIRLPOOL_EPH WK_WHIRLPOOL_P01 WK_WHIRLPOOL_V04 WK_WHIRLPOOL_V08	Phase logic with SFC WHIRLPOOL_EX Pump Outlet valve Outlet valve
04Shared		Shared Equipment for the UNITs MT and LT	SH_OUTLET_EQM SH_OUTLET_P01 SH_OUTLET_V01 SH_OUTLET_V02 SH_OUTLET_V03 SH_OUTLET_V04	Shared Equipment with BlFxShare und BlFxCtrl Pump Selector valve Selector valve Selector valve Selector valve

3.5.2.2 Tank Simulation

The actual filling levels and temperatures of the MT, LT and WK vessels are generated in Unit-specific Simulation-Charts. The input value of the respective integrator block is built based on the valve-control signals over selection-blocks. The calculated values are passed to the SFC-types as process values and create the transition conditions in relation with the recipe-setpoints .

3.5.2.3 SFC-Types

The following SFC-Types are used in the phase logic CFC-charts (xxx_EPH in the tables above) of the Warmblock PCell:

SFC-Type	Control Strategy	Setpoint (SP_) Processvalue (PV_) Controlvalue (CV_) Timer (T_)	Block contacts
AGITATE_EX	**Sequence Holding/Aborting** Reset 'StartAut.value' for M1 **Sequence RUNNING** CS_AGITATING: If PV_LEVEL > 0 activate M1 with duration T_AGITATE	PV_LEVEL T_AGITATE [60.0 s]	M1 – BlMotL
HEAT_EX	**Sequence Holding/Aborting** Reset 'OpenAut.value' for V1/V2 **Sequence RUNNING** CS_HEATING: Activate valves until PV_Temp >= SP_Temp	SP_Temp [10.0 °C] PV_Temp	V1 – BlVlvL V2 – BlVlvL
HOPS_EX	**Sequence Holding/Aborting** Reset 'OpenAut.value' for V1, V2, V3 **Sequence RUNNING** CS_HOP_ADD1 CS_HOP_ADD2 CS_HOP_ADD3 Activate respective valve until setpoint is reached	SP_KILO PV_HOPS_01 PV_HOPS_02 PV_HOPS_03	V1 – BlVlvL V2 – BlVlvL V3 – BlVlvL
INLET_EX	**Sequence Holding/Aborting** Reset 'StartAut.value'/ 'OpenAut.value' **Sequence RUNNING** CS_WATER CS_MALT_PALE CS_MALT_DARK Activate respective valve until setpoint is reached	SP_Liter SP_KILO PV_WATER PV_MALT_PALE PV_MALT_DARK CV_AGITATING → control value switches T_AGITATE in Phase AGITATE_EX between 30.0 / 120.0 s	V1 – BlVlvL V2 – BlVlvL V3 – BlVlvL M1 - BlMotL
OUTLET_EX	**Sequence Holding/Aborting** Reset 'OpenAut.value' and Function selection for BlFxCtrl block **Sequence RUNNING** The starting condition requests the Shared Elements via the	PV_VOLUME PV_LEVEL PV_WORT PV_SHARED_OCCUPIED PV_OPEN_SOURCE_SH_VAL PV_OPEN_DEST_SH_VAL PV_PUMP_ON_SH_MOT PV_SV01_FB_OPENED PV_SV02_FB_OPENED	V1 – BlVlvL V2 – BlVlvL

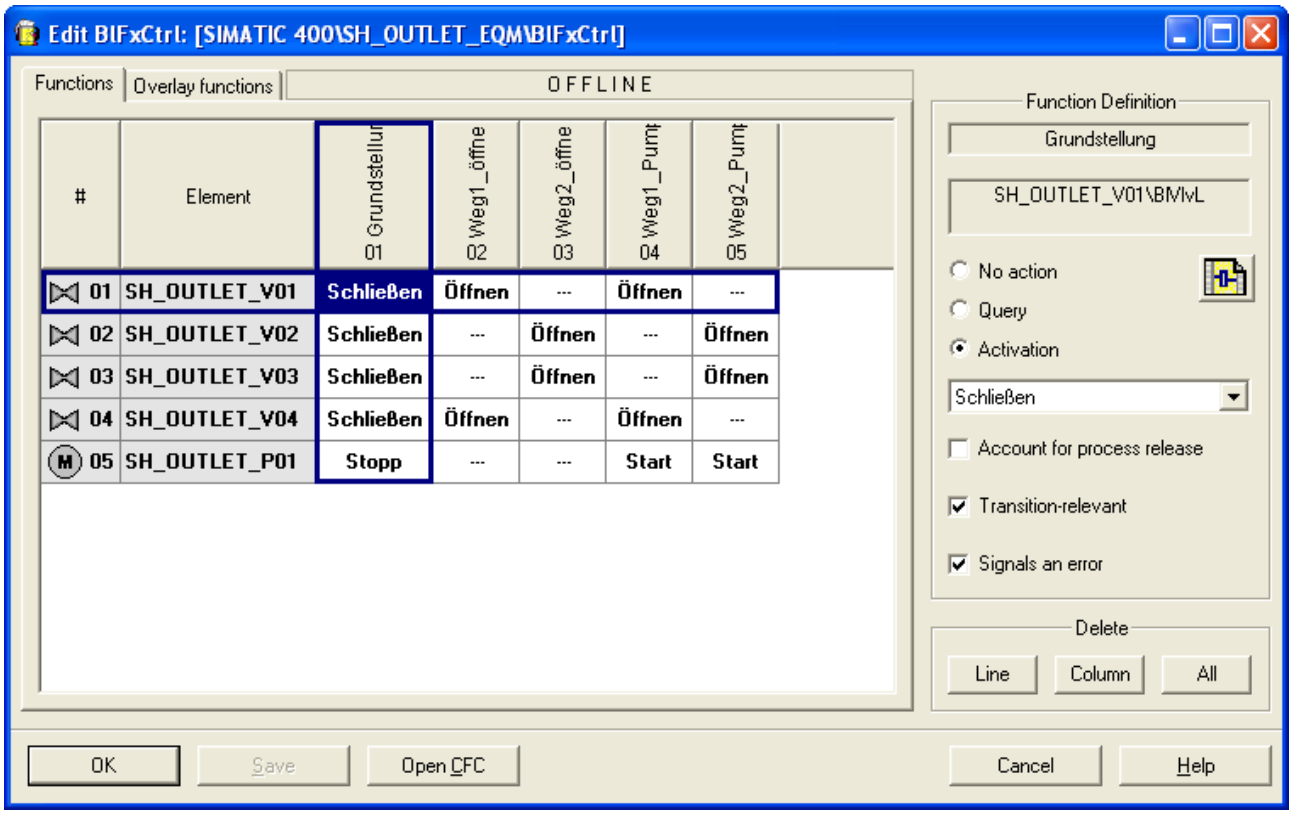
	BIFxShare block for this unit. CS_EMPTYING Activate the function for Shared Elements pump and valves through BIFxShare/BIFxCtrl blocks until setpoint is reached CS_WORT Set CV_1WORTOUT = true for 1st Wort, and = false for 2nd Wort	PV_SV03_FB_OPENED PV_SV04_FB_OPENED PV_SP01_FB_ON CV_1WORTOUT CV_SHARED_REQ CV_OPEN_SOURCE_SH_VAL CV_OPEN_DEST_SH_VAL CV_PUMP_ON_SH_MOT CV_SV01_CMD CV_SV02_CMD CV_SV03_CMD CV_SV04_CMD CV_SP01_CMD	
RAKE_EX	**Sequence Holding/Aborting** Reset 'StartAut.value' / 'OpenAut.value' **Sequence RUNNING** CS_ROTATE Start Motor M1 CS_OFF Switch Off all valves and motors CS_HOOK Hook rake via Motor M2 to P_HOOK_POS (defined in CFC-Chart instead of Recipe setpoint SP_RAKE_POS) CS_UNHOOK Unhook rake via Motor M2 to SP_RAKE_POS CS_DEEPCUT_RAKE Hook rake via Motor M2 to deepcut position SP_RAKE_POS CS_SPEND_GRAIN _REMOVAL Activate valve V1 for spent grain CS_POSITION Position rake to SP_RAKE_POS	SP_RAKE_POS PV_LEVEL PV_RAKE_POS PV_TOP PV_BOTTOM P_HOOK_POS	M1 - BIMotL M2 – BIMotRev M3 - BIMotL V1 – BIVlvL
TRANS_EX	**Sequence STARTING** Clear Route Requests; Request route from SP_SOURCE → SP_DEST; Repeat if request error; Reset the modes **Sequence RUNNING** CS_TRANSFER Base position Set route for pipe Activate pump Open Source-valve Open Destination-valve Wait until WK empty or FTx full	SP_SOURCE SP_DEST SP_VIA SP_FT_STATUS SP_UNIT_STATUS PV_AUTO_BATCH PV_FT_FULL PV_FT_EMPTY PV_FT_VOLUME PV_WK_FULL PV_WK_EMPTY PV_WK_VOLUME CV_START_FT1_BATCH CV_START_FT2_BATCH CV_SET_HOLD	RC – RC_IF_ROUTE SP – RC_IF_Encoder AC – RC_IF_Decoder

	The preselected Fermenting Batch is started automatically. A handshake via Route Control takes place, if the Tank status is in the correct state “Wait for Filling” or ‘Filling’		
WHIRLPOOL_EX	**Sequence Holding/Aborting** Reset ‘StartAut.value’ / ‘OpenAut.value’ **Sequence RUNNING** CS_WHIRLPOOL Open valves Start pump After T_Whirlpool stop pump Close valves	PV_LEVEL T_WHIRLPOOL [60.0s]	P1 - BlMotL V1 - BlVlvL V2 - BlVlvL

3.5.2.4 Shared Equipment with BlFxCtrl and BlFxShare

The control of four valves and the pump of the shared equipment “SH_OUTLET_EQM” for the mash transfer in the OUTLET_EX phase is implemented with the block BlFxCtrl in CFC chart „04Shared\SH_OUTLET_EQM“. The access from the participating units LT and MT of the shared equipment is coordinated with the BlFxShare block

The BlFxCtrl Engineering is shown in the following screenshot:



The possible routes are defined as functions of the BlFCtrl block. In the function definitions (columns) the commands of the involved CMs (V01 – V04, P01) are configured. At runtime the function number is forced in the SFC OUTLET_EX and the control and monitoring of the control modules is processed independently in the BlFxCtrl block.

3.5.3 PCell ‚Cellar'

The following table shows the list of CFC-charts in the according hierarchy levels.

3.5.3.1 CFC-charts of the technological hierarchy

In the cellar area, 8 tank units (4 x fermenting, 4 x storage) with identical structure are used.

UNIT-folder	UNIT-charts	EQM-folder	EQM-charts	Function
FTxxxx	FTxxxx FTxxxx_SIM	FTxxxx_TANK_EQM	FTxxxx_EPH	Phase logic with SFC - TANK_EX
			FTxxxx_LEVEL	Filling level
			FTxxxx_MOF	Grenzwerte (RC-PE-SE-IF)
			FTxxxx_PRESS	Pressure
			FTxxxx_TANK	Tankstatus (BlTank)
			FTxxxx_TEMPERATURE	Temperature
			FTxxxx_Vxx	Valve emptying
STxxxx	ST1 ST1_SIM	STxxxx_EQM	STxxxx_EPH	Phase logic with SFC - TANK_EX
			STxxxx_LEVEL	Filling level
			STxxxx_MOF	Grenzwerte (RC-PE-SE-IF)
			STxxxx_PRESS	Pressure
			STxxxx_TANK	Tankstatus (BlTank)
			STxxxx_TEMPERATURE	Temperature
			STxxxx_Vxx	Valve filling
XFER_FT_ST	XFER_FT_ST_UNIT	PIPES	FT_ST_Pxx FT_ST_Vxx	'Young-beer' lines Pumps /-Valves
		XFER_FT_ST_EQM	QING_XFER_EPH	Phase logic with SFC – XFER2_EX
			XFER_FT_ST_MOF	OP_D blocks for viewing active Modes of both routes

3.5.3.2 Tanksimulation

The actual filling levels and temperatures of the Fermenting and Storage-Vessels are generated in Unit-specific Simulation charts. The release signals from the central logic chart controls extra Simulation- and Compare blocks (AWL-Sources see S7-program/Source folder). The calculated values are passed to the SFC-Types as process values and create the transition conditions in relation with the recipe-setpoints .

3.5.3.3 SFC-Types

The SFC-Types TANK_EX and XFER2_EX are used in the EQM-CFC-charts (xxx_EPH in the table above) of the Cellar PCell.

The SFC-types contains several sequences for the different mode states, which in turn are partitioned into several control strategies. These are described below. For better understanding the SFC-Type TANK_EX or XFER2_EX in example project → S7-Program / charts should be opened with the SFC-Editor.

Note:
Only the functions which are used in the example project, are described here. As this SFC-Types originates from real projects, the implemented parameters, setpoints, control values and control strategies provides much more detailed control functions too.

3.5.3.3.1 TANK_EX

With that SFC-Type the Fermenting and Storage-Tanks can be controlled. The different states and the according Control-logics are implemented here via the following control strategies.

Sequence „Running“

CS_BASE_POSITION
wait for Tankstatus = „cleaned“ .OR. „sterile“

CS_OFF
actually no function…

CS_CHECK_PRESS
Wait 2 sec
Wait for Tank pressure is in range (0.0, 2.5)

CS_FILLING
Set Tankstatus =’Filling’
Wait for Tankstatus = ‘filled’ .AND. filling level >= SP_LEVEL

CS_STORAGE
Set Tankstatus =’filled’
Wait for PV_ACK_OP=1 (Operator-Switch ‚STORAGE_END’ in Process Picture) .OR. Q_OPRQ=0 from Tankstatus

CS_EMPTYING
Set Tankstatus = ‚Emptying’
Wait for Tankstatus = ‘empty’ .OR: filling level <= 0.0
Set Tankstatus = ‚Empty’

CS_CIP
Set Tankstatus = ‚Cleaning’
Wait 30sec
Set Tankstatus = ‚Cleaned’

CS_STATUS
actually no function

CS_BASEPOSITION_CIP
Wait for Tankstatus = ‚Filling’ .OR. ‚filled’ .OR. ‚Emptying’

CS_WAITFILLING
Set Tankstatus = ‚Wait_Filling’
Wait for Tankstatus = ‚Filling’

CS_WAITEMTYING
Set Tankstatus = ‚Wait_Emptying’
Wait for Tankstatus = ‚Emptying’

Sequences „Holding“ and „Aborting“

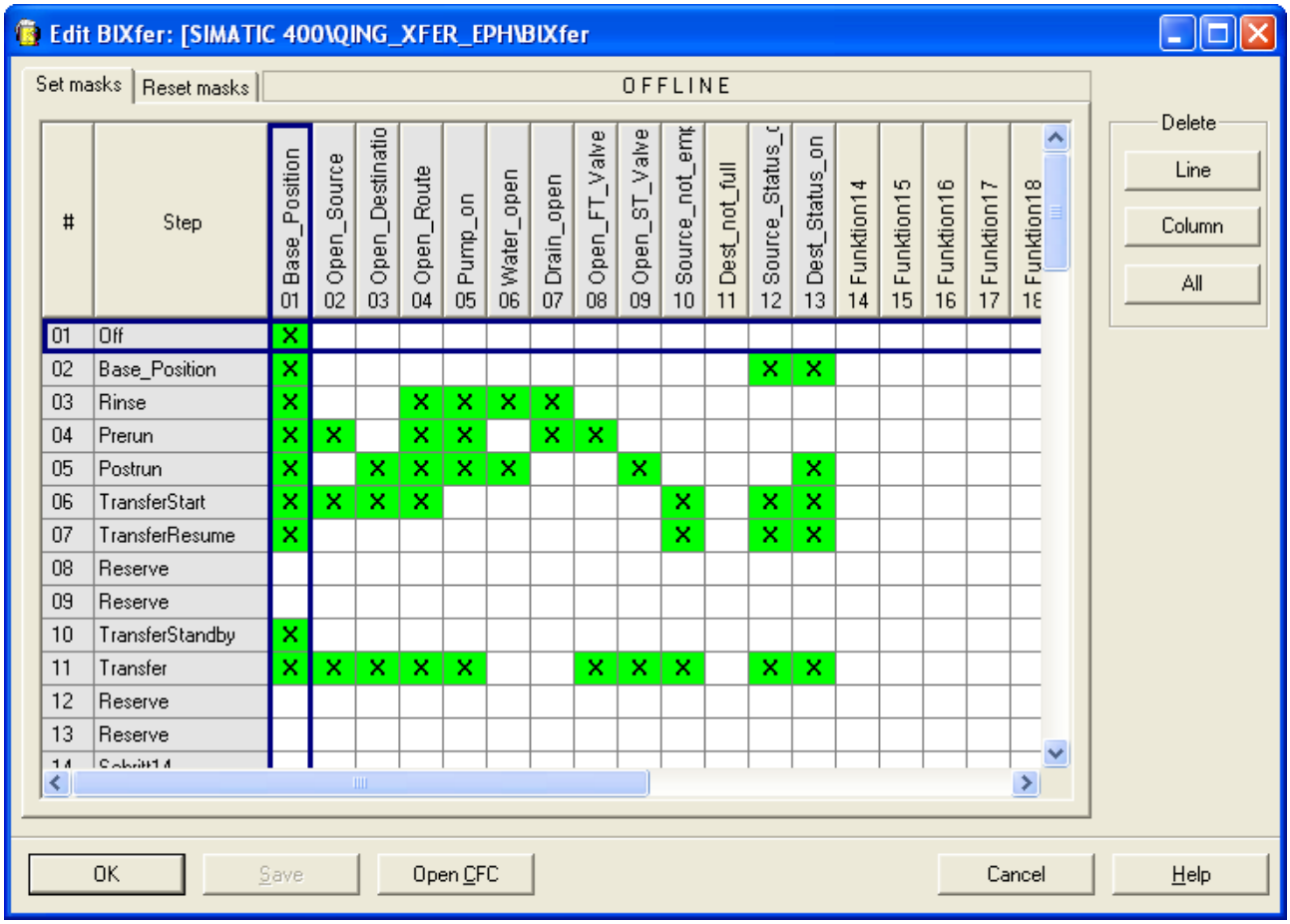
Reset all requests and controls

3.5.3.3.2 XFER2_EX

The SFC-type is responsible for bumpless Route switchover via the transfer master block BIXfer in association with SIMATIC Route Control. All necessary blocks for a universal transfer application in the fermenting and storage area are located and interconnected in CFC-chart ‚QING_XFER_EPH'. Furthermore the block BIQing was integrated in the transfer-example. With that a list of material movements (Transfer-Queue) with up to 8 Source-/Destination-Pairs in 1:n or n:1 constellation may be processed over the BIXfer block. With appropriate signal edges on the inputs the Source- /Destination – ID's are delivered consecutively and switched over in a bumpless manner. In doing so, each next route is prepared for RCS in Standby mode, while the actual is running.

Note:

The step programming of the BIXfer block is carried out in this version of the PCS 7 BRAUMAT Library with a new engineering dialog (SIMATIC Manager Menue: "Options → SIMATIC BRAUMAT → Block engineering"). The BIXfer configuration for this example project looks like this picture:



#	Step	01 Base_Position	02 Open_Source	03 Open_Destinatio	04 Open_Route	05 Pump_on	06 Water_open	07 Drain_open	08 Open_FT_Valve	09 Open_ST_Valve	10 Source_not_emp	11 Dest_not_full	12 Source_Status_c	13 Dest_Status_on	14 Funktion14	15 Funktion15	16 Funktion16	17 Funktion17	18 Funktion18
01	Off	X																	
02	Base_Position	X											X	X					
03	Rinse	X			X	X	X	X											
04	Prerun	X	X		X	X		X	X										
05	Postrun	X		X	X	X	X			X				X					
06	TransferStart	X	X	X	X						X		X	X					
07	TransferResume	X									X		X	X					
08	Reserve																		
09	Reserve																		
10	TransferStandby	X																	
11	Transfer	X	X	X	X	X			X	X	X		X	X					
12	Reserve																		
13	Reserve																		

Sequence „Running“

CS_BASE_POSITION

- Switch BlQing, BlXfer into „External“-Mode
- Switch BlQing into Transfer mode
- Query if routes are in Auto-Mode
- Request Master- and Standby-routes
- Output modes for base position (Step “Base position”)
- The signal ‘Ready to complete’ is applied if the elements are in automatic and the feedback „Base Position OK“ is signaled from the route.

CS_OFF

- Treminate Transfer mode in BlQing
- Shut down both routes:
- The signal ‘Ready to complete’ is applied, if feedback „Route off“ is active.

CS_RINSE

- Output modes for rinse (Step “Rinse”)
  The valves and the pump for push out with water to drain are activated.
- The signal ‘Ready to complete’ is applied, if the push out time (5 sec in this case) has expired.

CS_PRERUN

- Output modes for pushout (Step “Prerun”)
  The valves and the pump for push out the water to drain with material are activated.
- The signal ‘Ready to complete’ is applied, if the amount and the push out time (5 sec in this case) has expired.

CS_TRANSFER

- Output modes for transfer
  The valves and the pump for material transfer are activated.
- In Step WAIT_SWITCH the SFC waits for a signal for Tank switchover or the signal for terminating the Transfer. In case of a tank switchover via input PV_SWITCH, the standby route is activated and the former master route is terminated.
- In case of more Source – Destination transfers are in the queue of the upstream BlQing block the control jumps back to Step ACT_TRANSFER and the next transfer will be processed.
- The signal ‘Ready to complete’ is applied, if the lower limit value for the source tank is under-cut or the upper limit value for the Destination tank is exceeded and the actual values are > 0.0.

CS_POSTRUN

- Output the modes for pushout.
  The valves and the pump for push out material to the tank are activated.
- In Step WAIT_SWITCH the SFC waits for a signal for Tank switchover or the signal for terminating the Transfer. In case of a tank switchover via input PV_SWITCH_PUSHOUT, the standby route is activated and the former master route is terminated.
- In case of more Source – Destination transfers are in the queue of the upstream BlQing block the control jumps back to Step ACT_PUSHOUT and the next transfer will be processed.
- With the recipe setpoint SP_SEL_PUSHOUT it may be selected, if the pushout is done with a list of tanks or only into the last destination tank.
- The signal ‘Ready to complete’ is applied, when the signal PV_PUSH_DONE becomes active.

CS_STANDBY
actually not used

Sequence „Trouble Handling“

This sequence is called, if the Master route has raised an error and output „CV_SET_HOLD“ is activated on the SFC. If the SFC for itself should be held too, then output „CV_SET_HOLD“ must be connected with input „LOCKHOLD“. Via command RESUME the sequence and output CV_SET_HOLD will be reset..

Sequence „Completing“

R_RC1_Ack/R_RC2_Ack/R_RC1_Hold/R_RC2_Hold are reset

Sequence „Holding“

All controls and requests are reset and the message ‘Request Hold’ is applied. The pump is switched off at first via a Mask_ID

Holding is terminated, when the route is switched off

Sequence „Resuming“

„Request Hold“ is cancelled and the acknowledge for the route is applied.

Resuming is terminated, when the route is ok or a new error has occurred or feedback ‚Route is loaded’ is missing.

Sequence „Aborting“

All controls and requests are reset

Aborting is terminated when the route is switched off.

3.6 Working with the project

3.6.1 Starting the examples project

Prerequisite for starting the example project

- The actual Windows-User must be a member in the ‚Logon_Administrator' Group
- In WinCC User-Administrator the 'SIMATIC Logon' option is activated and the Group 'Logon_Administrator' is configured with all rights for all Units .
- Start WinCC Runtime. So the RCS server will be started automatically
- Start the SIMATIC BATCH Launch-Coordinator in case it is not started automatically with WinCC Runtime or when the PC is started up
- The Windows-User must be a member in the SIMATIC BATCH Role "Superuser". Please Log on with that user onto BATCH Control Center

Release the Recipes for Production in SIMATIC BATCH Control Center

If you have not already released the master recipe, right-click on the master recipe and select "release". The recipe will be tested for plausibility and this will be acknowledged. Following a successful plausibility test, an appropriate message will be displayed.

3.6.2 PCell ‚Brewhouse'

3.6.2.1 Ablaufschema

Die nachfolgende Liste zeigt die in diesem Beispiel verwendeten technologischen Schritte beim Bierbrauen.

Procedure for brewing

The following list shows the technological Steps in the example project for brewing beer

1. Fill 1.400 hl of water into the Mashvessel.
2. Start agitation in the Mashvessel.
3. Heat the Mashvessel to 50°C.
4. Fill 32 t of crushed pale malt or 20t dark malt to the Mashvessel.
5. Heat the Mashvessel to 70°C.
6. Agitate for 30 seconds
7. Empty the contents of the Mashvessel into the Lautervessel.
8. Wort circulation in Lautervessel (120 sec)
9. Empty the contents of the Lautervessel into the Wortkettle – 1st Wort
10. Lautering and Deepcut
11. Empty the contents of the Lautervessel into the Wortkettle – 2nd Wort
12. Remove Spent grains in Lautervessel
13. Heat the Wortkettle to 100°C and add 60kg of hops to the Wortkettle (hops addition 1 and 2).
14. Leave to rest for 30 seconds.
15. Wort clarification for 120 sec in the Wortkettle
16. Switch on cooling.

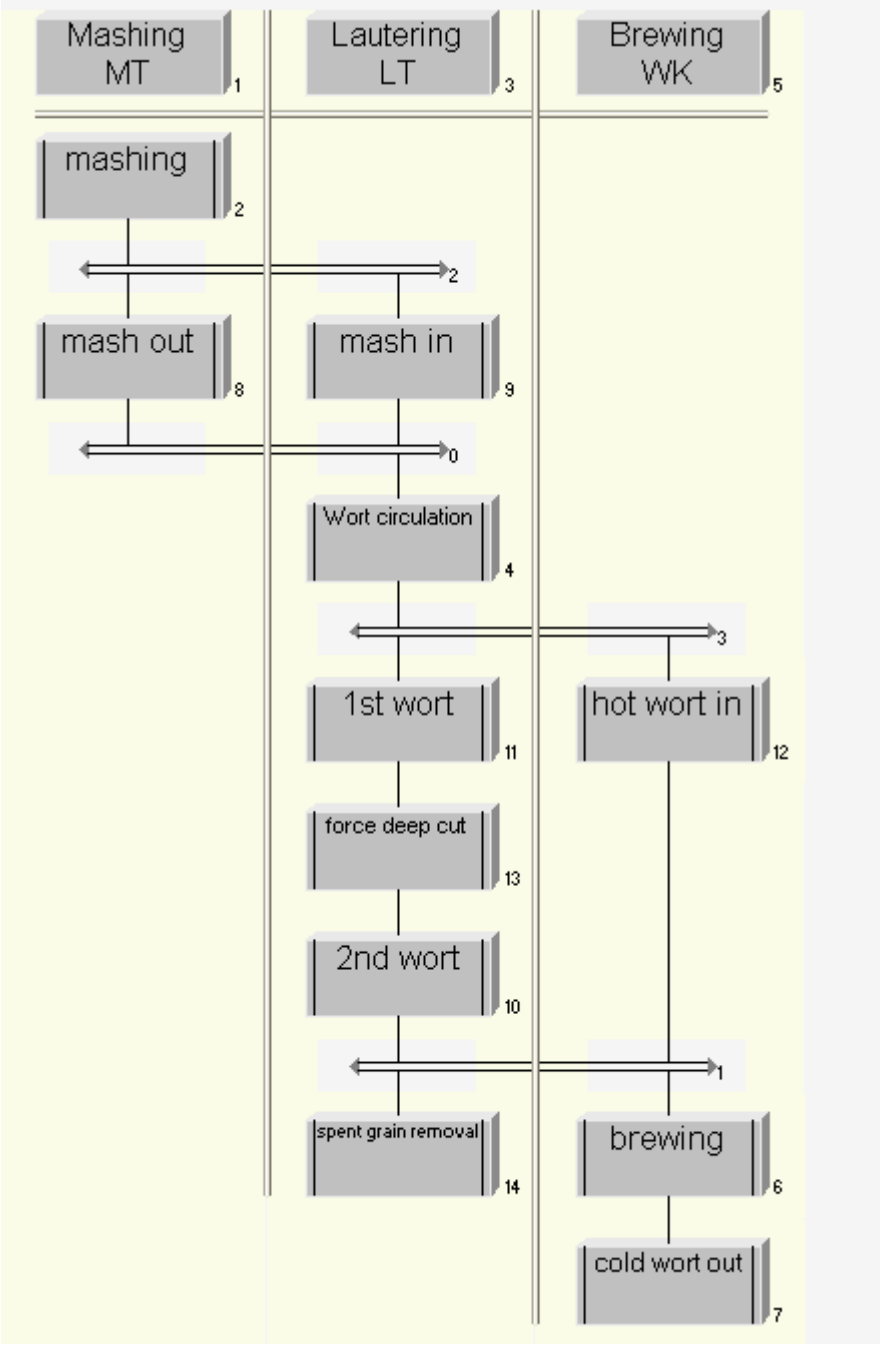
17. Empty the contents of the Wortkettle into the Fermenting vessel.
    Note: In the last Recipe Operation of the Wortcooker a new Batch for Fermeter 1 or 2 is created and started automatically – if a special checkbox is selected

3.6.2.2 Structure of the recipes

The master recipe „MT_BREWHOUSE V7.1“ of the BREWERY plant is a hierarchical recipe with the following recipe procedures :

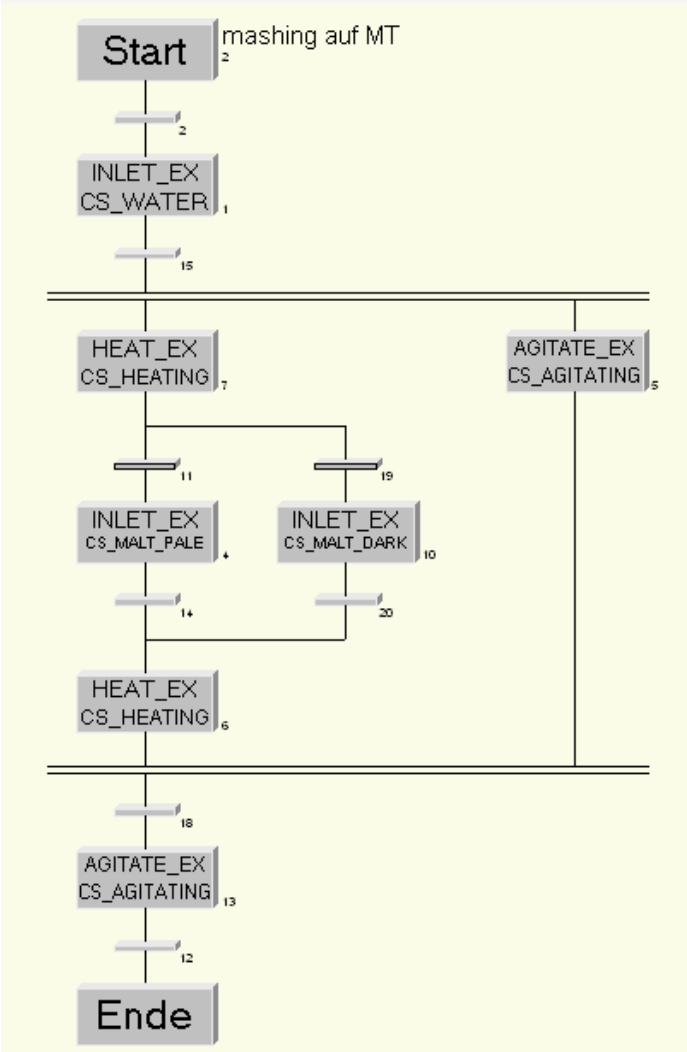
- Mashing (MT)
- Lautering (LT)
- Brewing (WK)

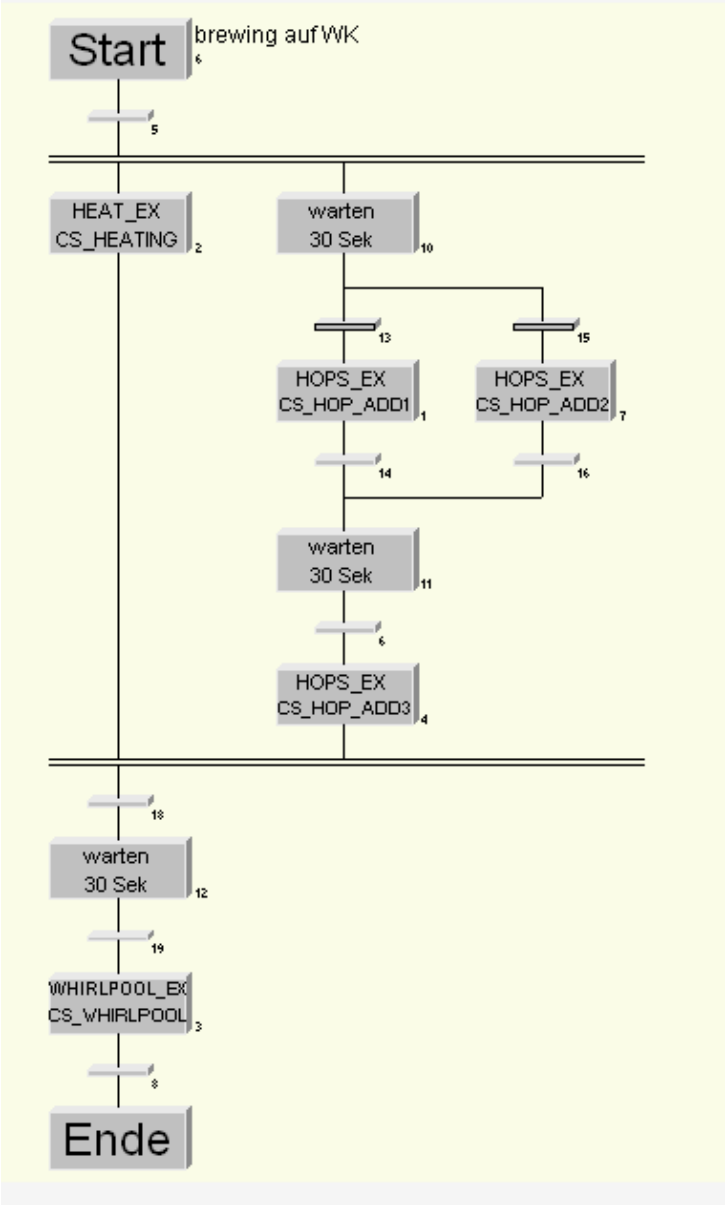
Each RP contains at least one recipe operation (ROP). The ROPs contain the recipe functions (RF) or -phases. The following display shows the recipe with the units and ROPs.



The activity of the recipe functions (RFs) to the according ROP is shown in the second recipe level. In the recipe functions the EQM-Phases (SFC-Types) with the control strategies and setpoints are called

Examples of ROPs

ROP_MASHING	
	In the mashing ROP after filling of water and malt the wort is heated. In parallel to that after the first filling the wort is stirred.

ROP_BREWING	
	In the ROP of wort cooker the hops addition takes place in several steps and in parallel the wort is heated After waiting for 30 sec the wort is rotated in phase WHIRLPOOL_EX for wort clarification.

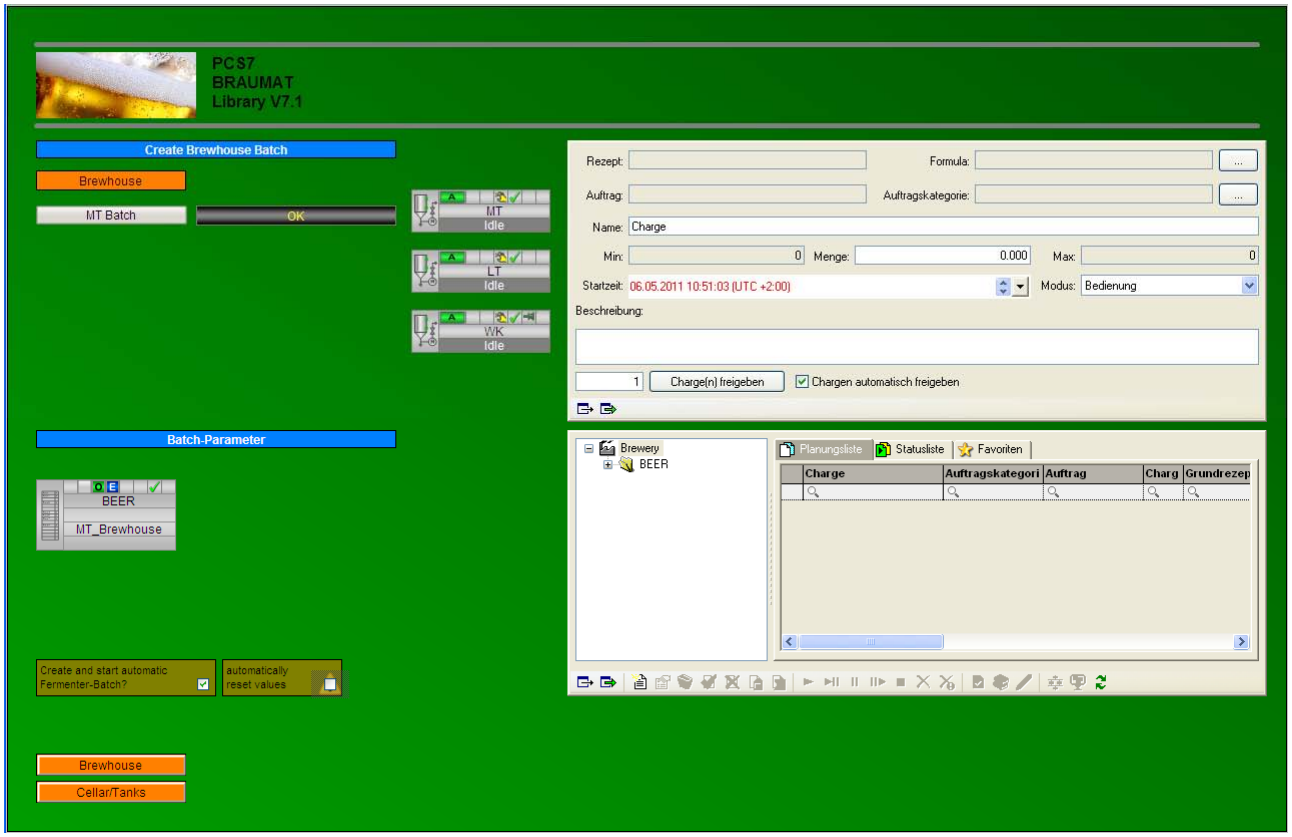
General:

Free running processes, as for example protein or saccharification rest are implemented as wait times in the recipe (NOP with configured runtime).

3.6.2.3 Process picture “BrewhouseBatch”

Batches can be created and started directly in the process picture „Brewhouse Batch“ alternatively to the SIMATIC BATCH Control Center. It may be called both via the Picture Tree and via Button „Batch/Configuration“ of the Brewhouse-overview picture (left below). Prior to starting a batch the filling levels in brewhouse area should be reset by pressing the button „Reset“. With selecting the checkbox ‘automatically reset values’ this can be done automatically on each batch start.

In the process picture the following Select-boxes and Buttons for Creating and Starting of Batches are available. The batch creation and start is processed in the AS via the ‘BIBatchIf’ block in unit ‘MT’. The created Batches are displayed in the right field in the lower window ‚BATCH OS Process Cell’ and may be started there.



Left field

- „Create Brewhouse Batch“
  Button „MT Batch“ → Create and Start Batch
  Symbols for units MT, LT, WK → Open BIUnitIf OS-Faceplate
- ‘Batch Parameter’
  With click on the symbol the BatchIf Faceplate may be opened. Here the batch data and the recipe parameters (preset from block parameters) are displayed. After switching into “Internal mode”, all parameters are changeable in the according fields
  - Destination tank/SP_DESTINATION (1301)
    The selected Tank should contain enough room for the batch volume. Before the Fermenting batch is started, the filling level of the selected tank is checked. An operator message is generated in case of insufficient room is detected. Before the Message is acknowledged, the destination tank may be emptied by manual adjustment of the filling level or by transfer into the Storage tank
  - Water (max. 2.000 hl)
  - Malztype (pale, dark)
  - Malt amount (max. 50 t)

- Hop (hop tank 1 or 2)
- Hop amount (max. 70 kg)
- Hop dosing (max. 20 kg)
- Hot water (max. 2.000 hl)

- Further batch parameters needed for Batch start are preset at the BlBatchIf block as follows (delivered state):

Parameter	**Block input**	**Value**
Order category	OrderCat	BEER
Recipe name	RecipeName	MT_Brewhouse
Recipe Version	RecipeVersion	„V7.1“
Batch Size	BatchSize	1

This parameters may certainly be changed and adapted to other recipes in CFC chart (activate Online-mode)

- ‘Create and start automatic Fermenter Batch ?’
  In the Brewhouse batch a new Fermenting-Batch is created and started automatically
- “Automatically reset values”
  The filling levels of the unit vessels are reset to zero, after the Recipe procedure has finished.

Right field

- Upper window
  shows the Batch Control “BATCH OS Batch Creation” with FilterRecipeName = „BEER*“
- Lower window
  shows the Batch control ‚BATCH OS Process Cell’with FilterOrderCategoryName = „BEER*“

See chapter 3.6.4 “Process picture ‘Batch controlling’ - Tracking batch action via BCC-OCX” for more details about the windows.

3.6.2.4 Process picture „BrewhouseTrend“

This process picture with an application example for the batch curves function of the PCS 7 BRAUMAT Library may be opened via the Picture Tree. In the according picture file „Brewhouse_Trend.pdl“ the following archive TAGs are pre-engineered, that means Step 1 in the engineering legend is completed:

Trend	**Color**	**Measured Value**	**Archive TAG**
MT-Level	Black	Mashtun Filling level	MT_SIMULATION/LEVEL.PV_Out
MT-Temp	Red	Mashtun Temperature	MT_SIMULATION/TEMPERATURE.PV_Out
LT-Level	Blue	Lautertun Filling level	LT_SIMULATION/LEVEL.PV_Out
WK-Level	Green	Wort cooker Filling level	WK_SIMULATION/LEVEL.PV_Out
WK-Temp	Orange	Wort cooker Temperature	WK_SIMULATION/TEMPERATURE.PV_Out

Step 2 Select Unit(s)

Select the desired Unit from the available Units in the list. Optionally a 2nd Unit may be selected.

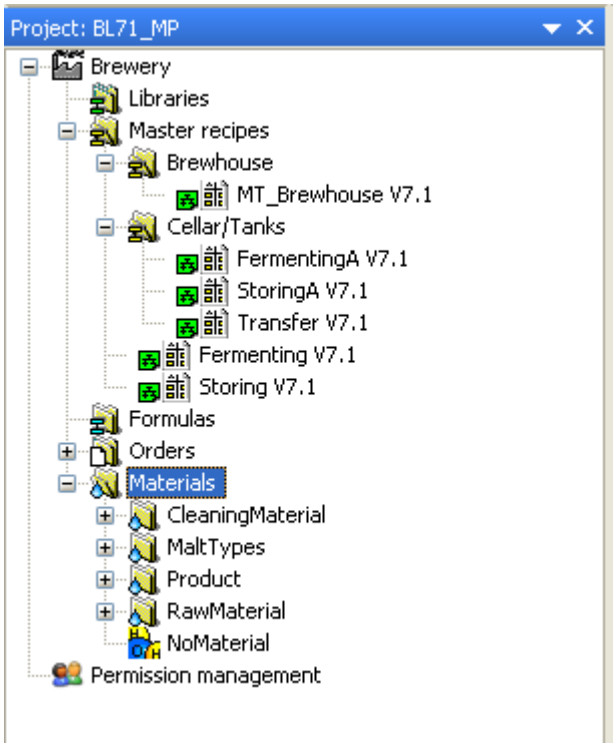
Step 3 Select Batch

Based on the selected Units and the Start-time the available Batches are displayed. After selecting a Batch the curve window is updatet. Thereby only the measured values of the selected Units are displayed.

3.6.3 PCell ‚Coldblock'

3.6.3.1 Recipe structure

Overview:



In the Fermenting- and Storage cellar a batch is started for every vessel. In the present example only the technological Steps for filling or emptying respectively are implemented. This Steps are located in the following recipes:

- Master recipes\Cellar/Tanks\FermentingA V7.1 → Fermenting Tanks
- Master recipes\Cellar/Tanks\StoringA V7.1 → Storage Tanks

3.6.3.2 Material-Transfer via SIMATIC Route Control:

The actual material movement from the Fermenting - into the Storage-Tanks is performed on AS-level in the Unit 'XFER_FT_ST' via requesting and activating a route in SIMATIC Route Control. For this an extra Transfer-Batch with recipe

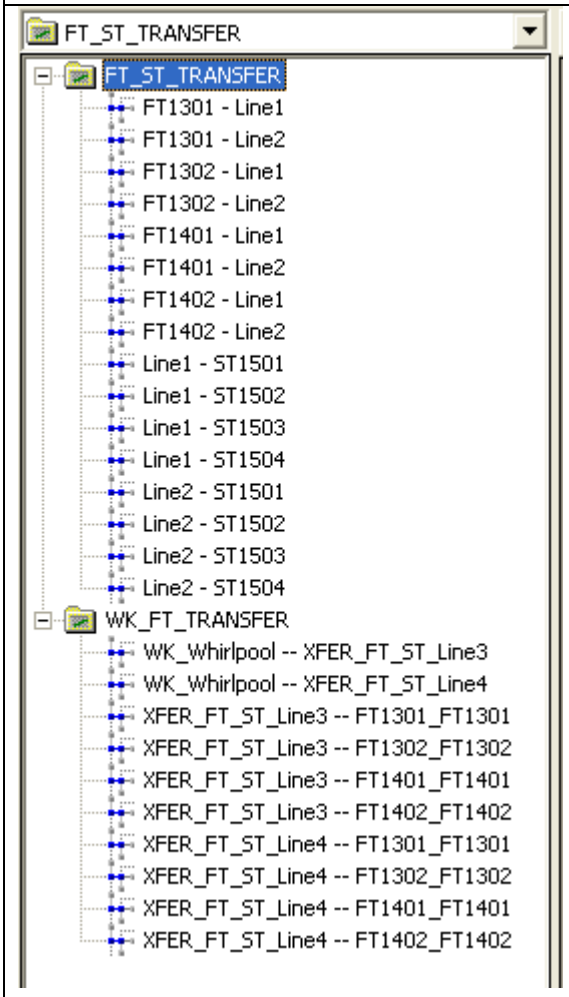
- Master recipes\Cellar/Tanks\Transfer V7.1

is to be started in this demo example. For the course of the batch this implies, that the ROPs "Filling" and "Emptying" respectively of the Tank-Batches are waiting until the Transfer-Batch is completed.

The transfer takes place from every Fermenting tanks into every Storage tanks according the selected recipe parameters. The required states of the participating aggregates (valves, pumps) are defined in terms of 'Partial Routes' and 'Function catalogs' in SIMATIC Route Control.

RCS-Engineering:

The RCS engineering used in the example is shown here in extracts:

Partial Route	Description
	Under Function Catalog **FT_ST_TRANSFER** the configured partial routes from the Fermenting-tanks to the Storage tanks are listed. The Partial route **FT1301-Line1** for example describes the Source/Destination-relation: • FT1301 = Fermenting tank 1 • Line1 = Transport line for material transfer. In the second Function Catalog **WK_FT_TRANSFER** the Partial Routes fort the material transfer from brewhouse into Fermenting/Storage cellar are listed. The same syntax applies here for the possible Source/Destination pairs WK → Wort cooker Linex → Transport line FTx → Fermenting tanks

The associated function definitions of Partial Route **FT1301-Line1** are shown in the following picture. The Control-, Sensor-, Parameter- and Link-elements are assigned to the partial route and the control function is selected and checked:

FT_ST_TRANSFER\FT1301 - Line1

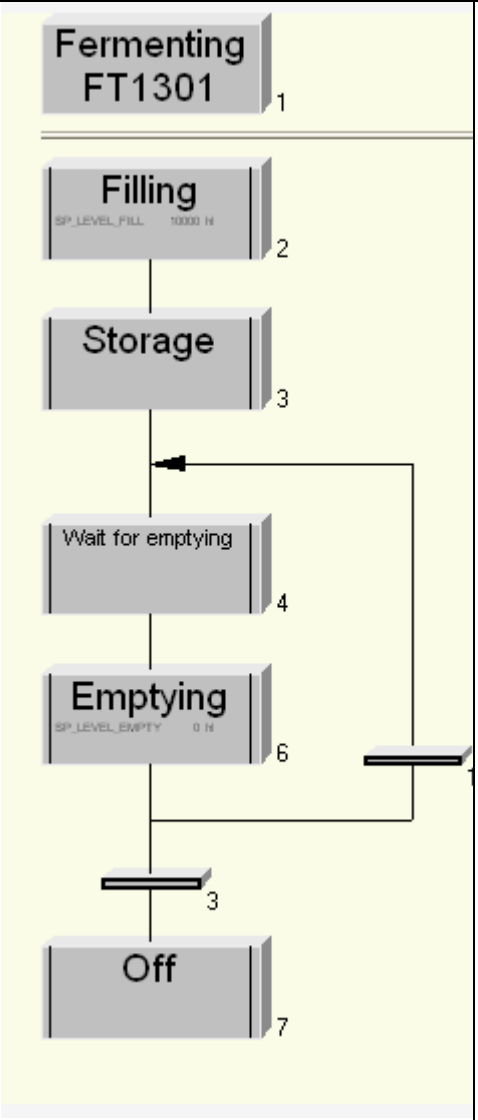
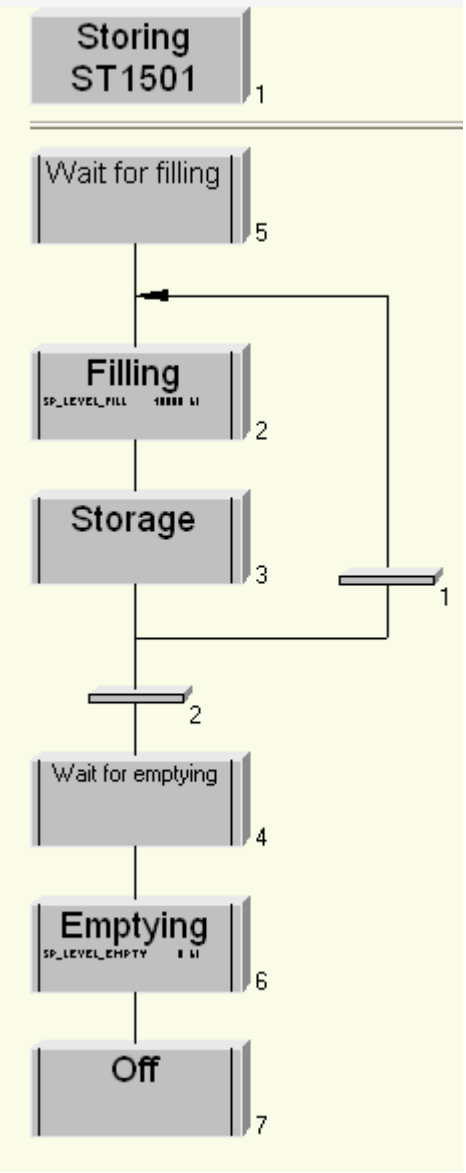
Modes | Elements | Deactivation sequence

Element	Type	No.	AS	Base_Position 01	Open_Source 02	Open_Destination 03	Open_Route 04	Pump_On 05	Water_open 06	Drain_open 07	Open_FT_Valve 08	Open_ST_Valve 09	Source_not_empty 10	Dest_not_full 11	Source_Status_on 12	Dest_Status_on 13
S7-Programm/FT1301_LEVEL/FT1301_LEVEL	PE	17	SIMATIC 400												1	
S7-Programm/FT1301_MOF/FT1301_ERROR	PE	19	SIMATIC 400												4	
S7-Programm/FT1301_MOF/FT1301_MAX	PE	16	SIMATIC 400												2	
S7-Programm/FT1301_MOF/FT1301_MIN	PE	15	SIMATIC 400												3	
S7-Programm/FT1301_MOF/FT1301_STATUS	PE	14	SIMATIC 400												10	
S7-Programm/FT1301_V11/CE_V11	CE	11	SIMATIC 400	d?	a											
S7-Programm/FT1302_V12/CE_V12	CE	12	SIMATIC 400	d?												
S7-Programm/FT_ST_P81/CE_P81	CE	81	SIMATIC 400	d?				a								
S7-Programm/FT_ST_V40/CE_V40	CE	40	SIMATIC 400	d?					a							
S7-Programm/FT_ST_V42/CE_V42	CE	42	SIMATIC 400	d?							a					
S7-Programm/FT_ST_V43/CE_V43	CE	43	SIMATIC 400	d?												
S7-Programm/FT_ST_V70/CE_V70	CE	70	SIMATIC 400	d?			a									

As an example - the state of tank valve FT1301_V11/CE_V11 is checked passive (d?) in function “01-Base_Position” and is activated (a) in function “02-Open_Source”.

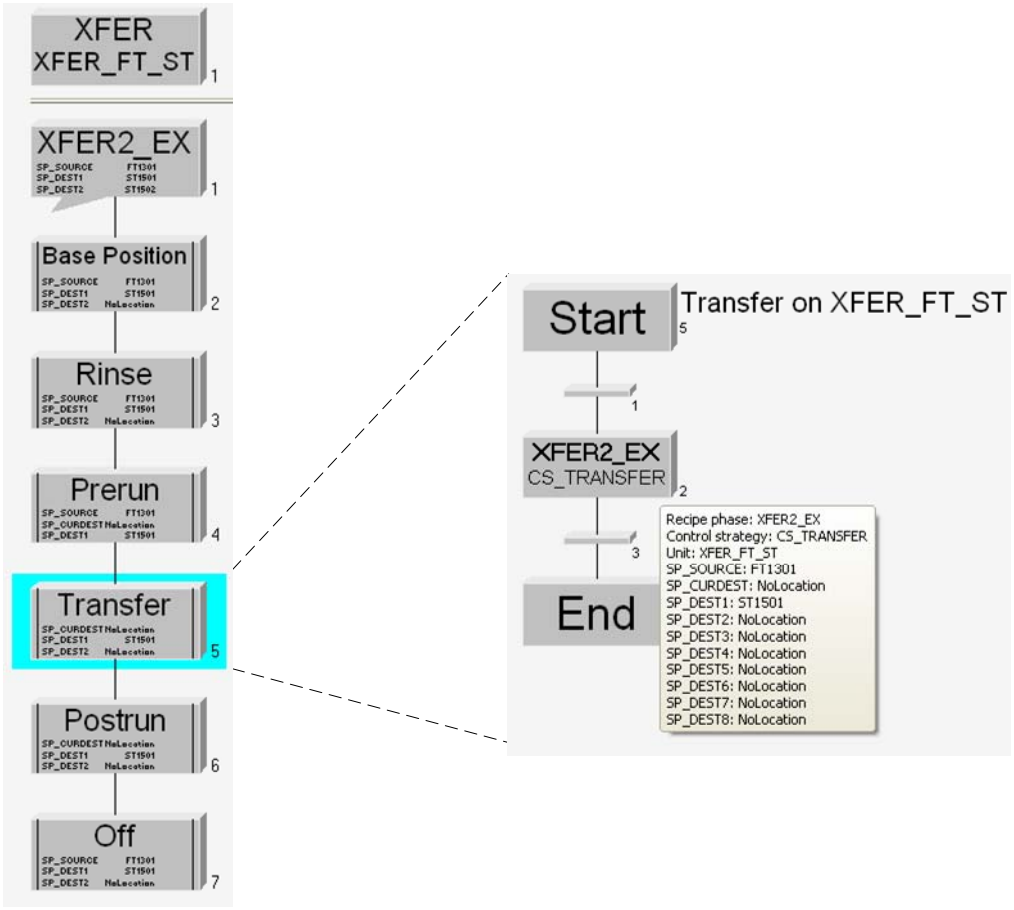
Further details can be found in the Manual of the RCS documentation.

3.6.3.3 TANK-Recipes

Fermenting		Storage
	← Fermenting activity 1. Unit 2. Filling from Wortcooker into Fermenting vessel 3. Storage (180 sec resting). 4. Wait for emptying 5. <not present> 6. Emptying Fermenting vessel into Storage vessel → activity occurs in Transfer-batch Jump back if additional transfers are requested (JUMP_TO_WEMPTY = True) 7. Tank inactive Storage activity → 1. Unit 2. Wait for filling 3. Filling from Fermenting vessel into Storage vessel → activity occurs in Transfer-batch 4. Storage Jump back if additional transfers are requested (JUMP_TO_FILLING = True) 5. Wait for emptying 6. Emptying into the Filler-Line 7. Tank inactive	

In the second recipe level always the EQM-Phase “TANK_EX” with the respective control strategies and setpoints is called

3.6.3.4 TRANSFER-Recipe XFER2_FT_ST



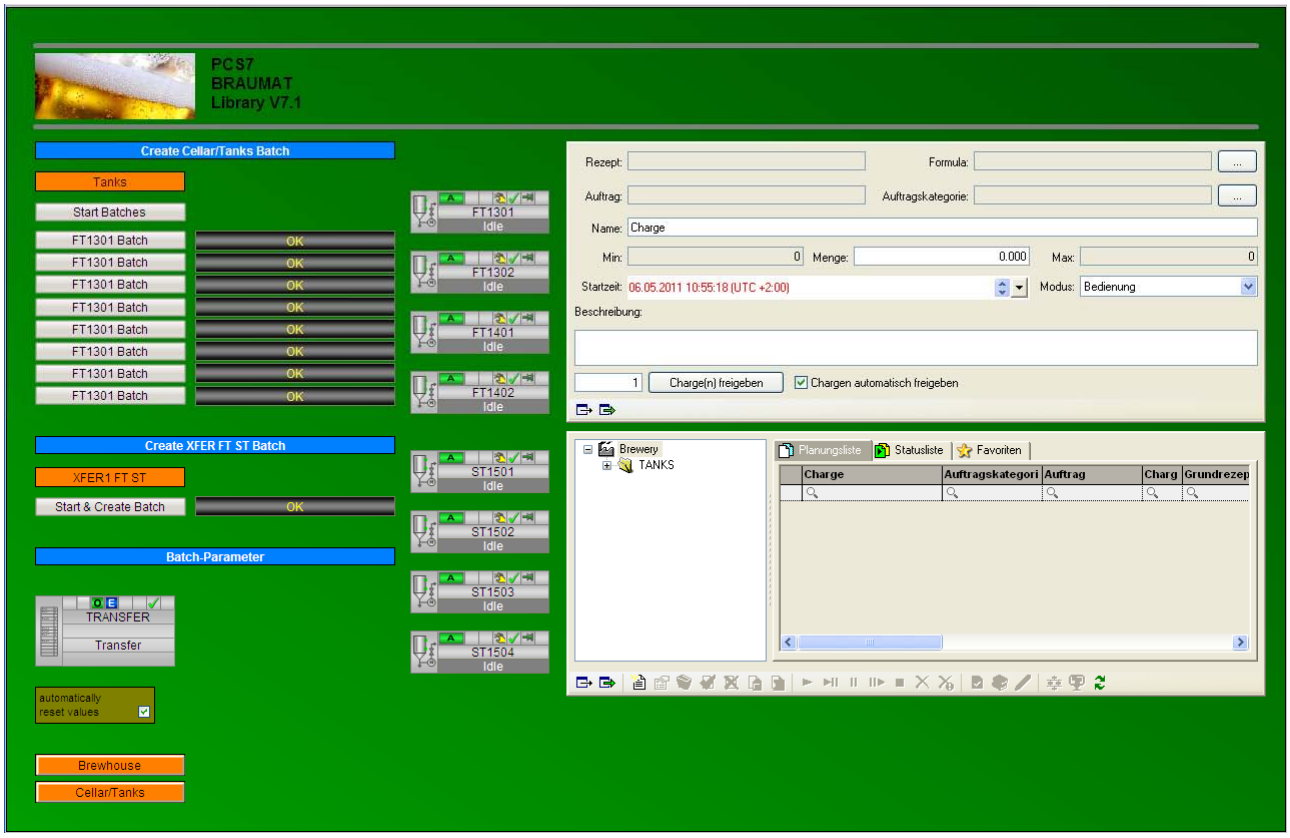
The first recipe operation “OPRequest” includes an operator request with acknowledge. This stops the batch until the operator confirms it. The recipe setpoints (Source-, Destination tanks and transfer amounts) may be changed in doing so.

Always the same EQM-Phase „XFER2_EX“ is called in the second recipe level also in this recipe with the control strategy and according setpoints. In the shown example the control strategy CS_Transfer is activated with the setpoints for source-, destination and transfer amount.

In control strategy CS_Transfer the complete transfer list from block BIQing is processed. The Route Requests at SIMATIC Route Control as well as the bumpless switchover between the routes are carried out via the block BIXfer of the PCS 7 BRAUMAT Library

3.6.3.5 Process picture „CellarBatch“

Here the following Select-boxes and Buttons for Creating and Starting of Batches are available. The batch creation and start is processed in the AS via the unit specific ‘BIBatchIf’ blocks.



Left field

- Create Cellar/Tanks Batch
  Button “Start Batches” → Create and Start Batches for all vessels
  Buttons “FTxxx/STxxx Batch” → Create and Start Batches for single vessels
  Symbols for units → Open BIUnitIf OS-Faceplate
- Create XFER_FT_ST Batch
  Button „Create & Start Batch“ → Create and Start Transfer-Batch
- Batch Parameter
  With click on the symbol the BatchIf Faceplate for the transfer batch may be opened. Here the batch data and the recipe parameters (preset from block inputs) are displayed. After switching the process parameters into “Internal mode” all parameters are changeable in the according fields
  - Source vessel (FTxxxx)
  - Destination vessel (STxxxx)
  - NEW Destination (STxxxx)
  - Further parameters for Batch creation are preset by default on the according block inputs as follows (delivered state):

Parameter	**Block input**	**Value**
Order Category	OrderCat	„XFER“
Recipe	RecipeName	„Transfer“
Recipe Version	RecipeVersion	„V7.1“

Batch Size	BatchSize	1

This parameters may certainly be changed and adapted to other recipes in CFC chart (activate Online-mode)

- "Automatically reset values"
  The tank filling levels are reset to zero approx. 30sec after the transfer batch has finished.

Right field

- Upper window
  shows the Batch Control "BATCH OS Batch Creation"
- Lower window
  shows the Batch control ‚BATCH OS Process Cell'
  with FilterOrderCategoryName = „FT*/ST*/TANK*"

See chapter 3.6.4 "Process picture 'Batch controlling' - Tracking batch action via BCC-OCX" for more details about the windows.

3.6.3.6 Process picture „CellarTrend"

Again, the picture is called via the Picture Tree. In the according picture file „CellarTrend.pdl" the following archive TAGs are pre-engineered, that means Step 1 in the engineering legend is completed:

Trend	**Color**	**Measured Value**	**Archive TAG**
FT1301-Level	Schwarz	Fermenting vessel 1 Filling level	FT1301_LEVEL/BlMonAn08.PV_OUT
FT1401-Level	Rot	Fermenting vessel 2 Filling level	FT1401_LEVEL/BlMonAn08.PV_OUT
ST1501-Level	Blau	Storage vessel 1 Filling level	FT1501_LEVEL/BlMonAn08.PV_OUT
ST1502-Level	Grün	Storage vessel 2 Filling level	FT1502_LEVEL/BlMonAn08.PV_OUT
ST1503-Level	Gelb	Storage vessel 3 Filling level	FT1503_LEVEL/BlMonAn08.PV_OUT
ST1504-Level	Cyan	Storage vessel 4 Filling level	FT1504_LEVEL/BlMonAn08.PV_OUT

Step 2 Select Unit(s)

Select the desired Unit from the available Units in the list. Optionally a 2nd Unit may be selected.

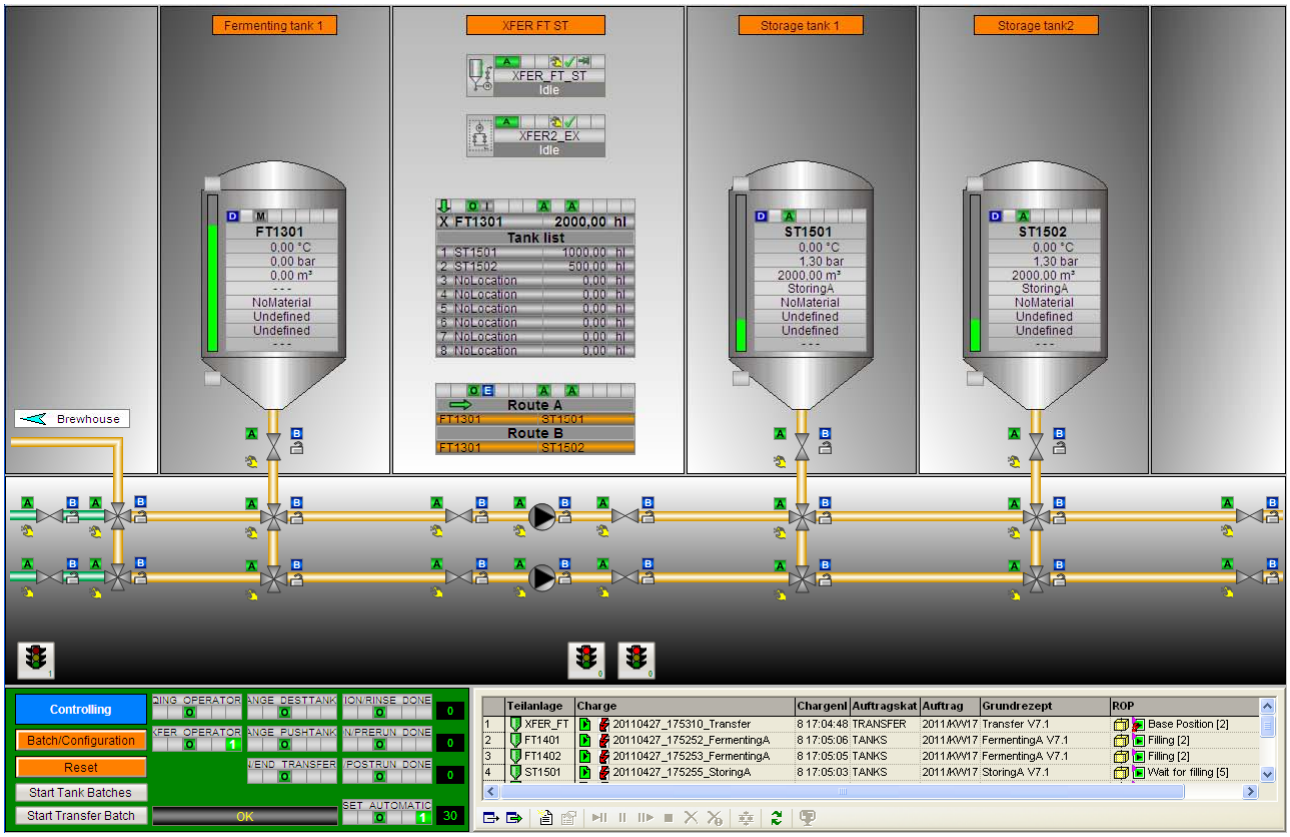
Step 3 Select Batch

Based on the selected Units and the Start-time the available Batches are displayed. After selecting a Batch the curve window is updated. Thereby only the measured values of the selected Units are displayed.

3.6.3.7 Process picture „CellarDynamic“

This process picture gives an example, how to realize a dynamic tank visualization with the new symbol of the transfer management block ‘BlTank’. The screenshot shows the following symbols:

- Center: transfer unit with the main control instance – consisting of the Queue transfer block ‘BlQing’ (Transfer list) and the transfer route control block ‘BlXfer’
- Left hand: Fermenting tank 1 – block ‘BlTank’ identified by the present source row of the transfer list
- Right hand: Storage tank 1 – block ‘BlTank’ identified by the present source row of the transfer list
- Right hand: Storage tank 2 – block ‘BlTank’ identified by the next source row of the transfer list



The ‘BlTank’ block symbol allows with the property „Configurations\EnableDynamicTankID = YES“ a dynamic association of the information display inside the symbol itself as well as for the calling OS-faceplate toward the underlying AS block ‘BlTank’. When the dynamic is activated (display **D** in the status line) the tank is selected through the property „Configurations\DynamicTankID“. This property have to be configured with a block parameter of type ‘LocationID’ by means if the ‘dynamic dialog’ of the WinCC Graphics Designer. In the present example, the tank symbols are configured as follows:

- Left hand: Fermenting tank 1 – DynamicTankID = QING_XFER_EPH/BlQing.CurSource
- Right hand: Storage tank 1 – DynamicTankID = QING_XFER_EPH/BlQing.CurDest
- Right hand: Storage tank 2 – DynamicTankID = QING_XFER_EPH/BlQing.NxtDest

Hence, when the transfer list is started, always the tanks which are actually joining the transfer and the next destination tank will be viewed in the process picture.

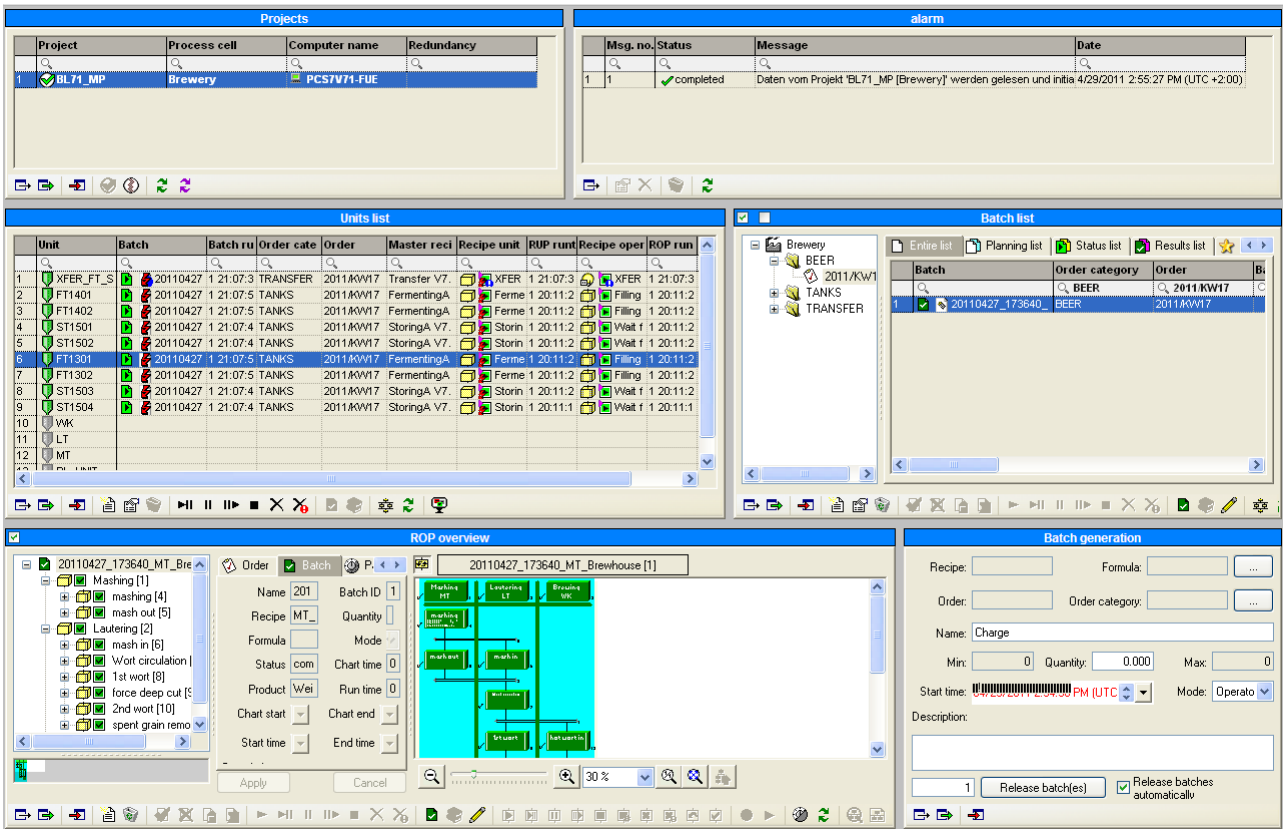
3.6.3.8 Operations in process picture „CellarDynamic“ and „CellarProcess“

The following symbols and buttons are available in the controlling section of the process picture:

Button, Operating symbol	***Function***
Batch/Configuration	Jump to process picture „CellarBatch“
Reset	Reset filling levels of the tanks
Start Tank Batches	Creates a batch for each tank and starts it
Start Transfer Batch	Creates a transfer batch and starts it
OpDi BLQING_OPERATOR	Mode for BlQing (On=Extern, Off=Intern)
OpDi BLXFER_OPERATOR	Mode for BlXfer (On=Extern, Off=Intern)
OpDi CHANGE_DESTTANK	BlQing manual switchover in transfer list
OpDi CHANGE_PUSHTANK	BlQing manual switchover in pushout list
OpDi END_TRANSFER	BlQing terminate transfer list
OpDi RINSE_DONE	Delay time in control strategy CS_Rinse in XFER2_EX is terminated
OpDi PRERUN_DONE	Delay time in control strategy CS_Prerun in XFER2_EX is terminated
OpDi POSTRUN_DONE	Delay time in control strategy CS_Postrun in XFER2_EX is terminated
OpDi RESET_AUTOMATIC	The tank filling levels are reset to zero approx. 30sec after the transfer batch has finished.

3.6.4 Process picture ‘Batch controlling’ - Tracking batch action via BCC-OCX

With this picture the batch activity in the Units may be monitored. New SIMATIC BATCH OS controls are available for operating and monitoring batch processes on a PCS 7 OS client as a graphic user interface (GUI). They are configured in the WinCC Graphics Designer. All the required operator controls for SIMATIC BATCH can be used in process mode (Runtime) using the controls.



The process picture “Batch-Controlling” contains sub-windows for the following Batch OS-Controls (See SIMATIC BATCH Manual Chapter BATCH OS Controls for further details):

- Projects – Select and connect Batch Project: Control SIMATIC BATCH OS Master
  In this control, select the BATCH project that you want to operate and monitor. All BATCH projects of available PCs in the network will be made available for selection if no filter or a fixed project was defined in the control itself. To activate a BATCH project, the Launch Coordinator of the BATCH server involved needs to be in the "Running" status. All other BATCH controls obtain their data from the activated BATCH project in this control, if they communicate on the same channel.
- Alarm - Control SIMATIC BATCH OS Monitor
  All BATCH process messages are displayed in this control. Always insert this control in your process picture. This will allow you to follow all the actions performed in the process mode based on the messages.
- Units list – Unit Overview: Control SIMATIC BATCH OS Allocation
  This control provides an overview of your units defined within your Batch process cells. In addition to the visualization of the units / unit allocation, the process conditions for the batch runtime can also be shown in the shortcut menu and/or toolbar.
- Batch list – Create Batch Orders: Control SIMATIC BATCH OS Process Cell
  The control shows the Batch process cell for the selected BATCH project. It is used for batch planning, batch creation and even batch operation.

- ROP overview – Control and Monitoring: Control SIMATIC BATCH OS Properties
  The control should be your central control for the operation and monitoring of batches in process mode. Here, a batch is brought to view in detail. In addition to visualization of the current batch, operator control can also be performed here.
- Batch generation - Control SIMATIC BATCH OS Batch Creation
  The control is used to create batches. There are two ways to create batches. You either create a new batch from scratch or you use the data from an existing batch as a template.

3.6.5 Prepare and start Batches in Batch Control Center

3.6.5.1 Create Batches

To create a batch, open the BCC with an activated tree view and proceed as follows:

1. Release the master recipe (f.e. .„MT_BREWHOUSE V7.1“) via the context menu for production.
2. Right-click on the order, set up a new order category and give it a name.
3. Set up a new order in the order category you have created. Proceed in the same way as point 2.
4. Select the .„MT_BREWHOUSE V7.1“ master recipe and click "OK".
5. Right-click on the newly created batch and select the "Release batch" command.
6. Start up the batch using the Start button in the context menu or the corresponding symbol in the toolbar.

3.6.5.2 Monitoring Batch Workflow

Open the control recipe in the main BCC window by double-clicking on the appropriate batch. This display corresponds to the display in the recipe editor. Double-clicking on ROPs will open them in a new window, but they cannot be changed. After the batch has been released, you will be able to access the project overview and start command toolbar, used to start the batch, via the context menu. The recipe starts with all ROPs in the first line and processes these one after another in the defined sequence. ROPs that are currently being processed are light green, ROPs that are closed are dark green. The symbols on or next to the ROPs provide information about the current status during and after processing. Additional information is contained in the online help for SIMATIC BATCH. Help on SIMATIC BATCH > Batch Control Center (BatchCC) > Batch control > Operation during batch control > Status of the batch steps.

Upon batch startup, the batch symbol in the batch list will change to "Running" . When the batch is complete, this will be closed and marked with "Completed" . To end the batch, select "Terminate" in the context menu.