Batch Control module

Design and usage overview

www.copadata.com
sales@copadata.com

Content

Author: Robert Harrison

1. Introduction

The Batch Control module has a wide scope of application, for entire facilities with multiple batch processes, at the production line, or in single unit operations. Batch Control incorporates different roles, such automation engineering, process engineering, operations management, quality management, and maintenance. zenon provides solutions for regulated environments such as Pharmaceutical industries, which are GMP and FDA 21 CFR Part 11 compliant, either in standalone or fully integrated applications.

This document aims to give an overview of the practicality of zenon’s Batch Control module without a deep technical knowledge, refining the benefits of automation, process control, operational use, and regulation compliance.

2. ISA 88 standard

Batch Control is defined in the ISA-88 standard. It defines the physical models, procedures, and recipe structure. Which through an ordered set of process operations, controls process equipment defined in these models.

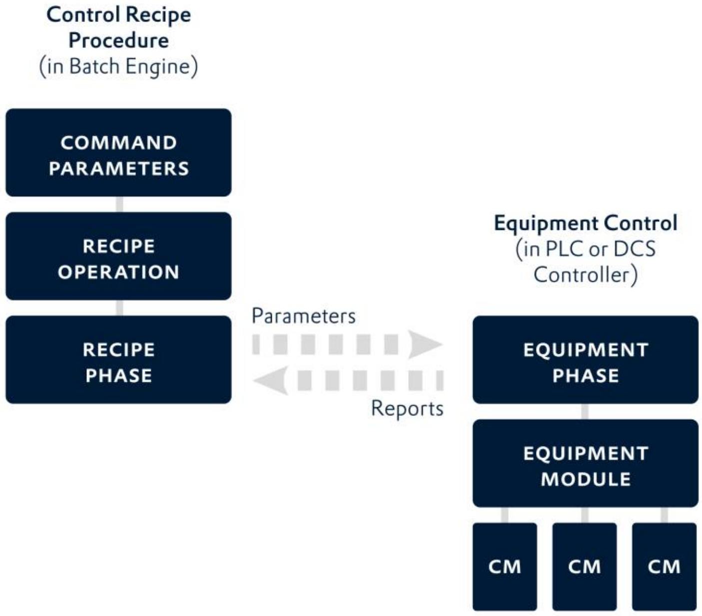
The ISA-88 was approved in 1995. Adhering to this standard is paramount for recognition and conformity in the process industries, the Batch Control module is therefore ISA-88 compliant. However, zenon goes beyond what the standard demands, providing the latest technology with a high end user interface.

Equipment control and process control

A principal element of Batch Control is to separate the process control from the equipment control. In this regard, the PLC controlling the physical equipment, knows how the equipment operates. For example a temperature control loop needs to have i) a temperature sensor input, ii) a control loop controlling to a set point, iii) an output to heat or cool the equipment. The same example can be applied to other control loops such as pressure or agitation speed. These equipment control loops know how to control their own hardware, however these control loops are not linked together into a process. The process control is achieved in the batch control module. The batch control recipe has phases which link to each equipment loop, these phases are constructed together into process flows, with a defined start sequence, process logic, and process parameters in a batch recipe.

The advantage here is that the individual equipment loops are designed to run at the equipment designed maximum. The equipment has no knowledge of how these individual loops are connected. Therefore the recipe can define any process configuration, within the maximum design limits of the equipment. Different sequences can be defined, as can different process conditions. This is a significant advantage in the pharmaceutical industries, as process change or optimization requires no change to the equipment design. Only the recipe needs to be validated, therefore saving significant effort, time, and cost on equipment requalification.

The diagram below, demonstrates the association between the PLC equipment control in the red box, and the batch process control in the blue box. The diagram also shows the batch control recipe has a hierarchy: 'Phases' provide the link to the equipment control, phases can be grouped in to 'Operations', 'Recipes' contain Phases and Operations. The whole batch process is contained in the Control Recipe.



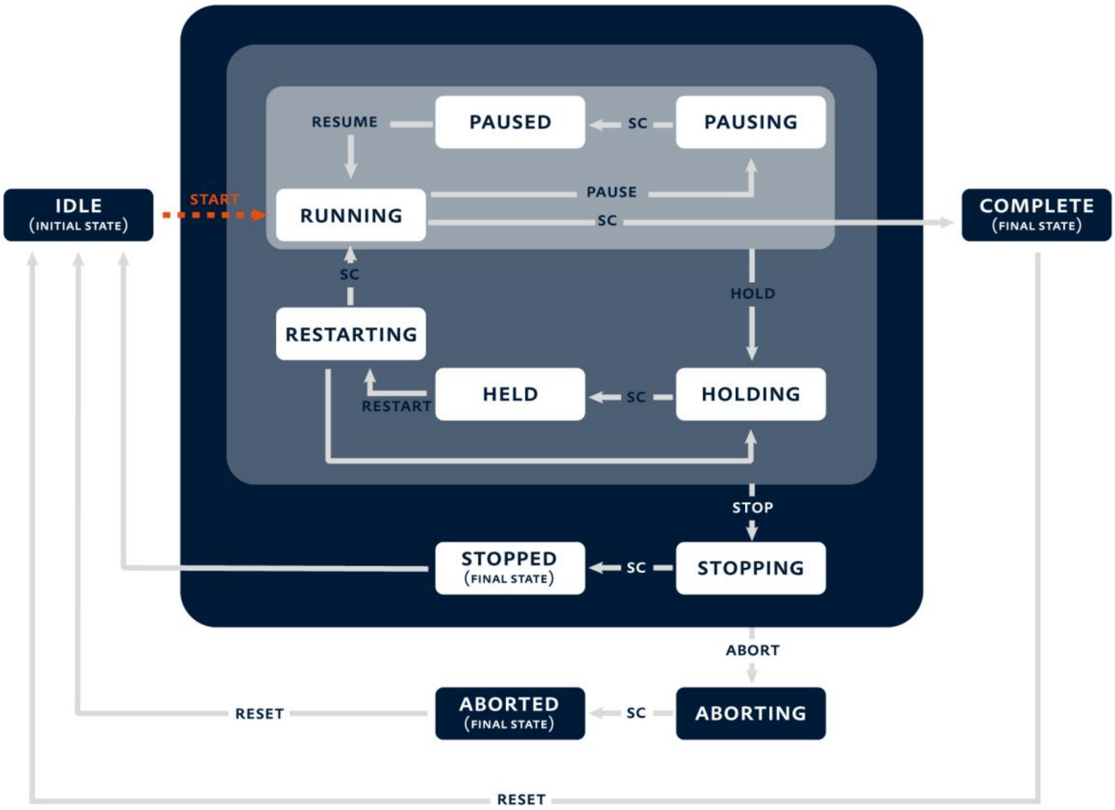
Master and Control Recipes

The ISA 88 standard describes a strict hierarchy for recipe management. The design of a recipe is carried out in the Master Recipe. Here the process flow is defined, with process decisions and the process parameters each phase will control to. Once the Master Recipe has been tested, it can be released for production.

A Control Recipe is generated from a released Master Recipe. Each Control recipe can be executed only once, so that each production batch is traceable to this unique instance.

Synchronization between control systems

The reality of two control systems, i.e. process control and equipment control, requires that both systems remain synchronized. The state model from the ISA 88 (shown below), describes how the two systems interact together.

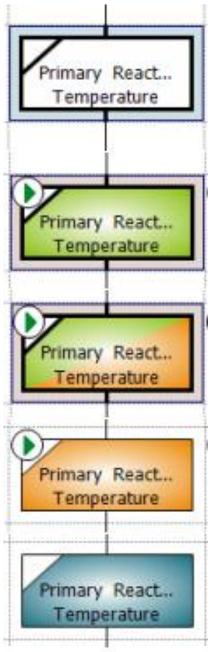


ISA 88 phase state model

For example when the phase is in a 'Running' state, the batch module would like to pause the recipe, the state model moves into the 'Pausing' state, this is communicated to the PLC. The PLC receives this state change, and carries out the necessary equipment logic. When this is complete the PLC changes the state to 'Paused'. The state model waits in this state until requested to 'Resume'.

The states ending in 'ING' are transient states in the batch engine, which request a state change to the equipment control PLC. The batch engine monitors the state of the equipment PLC logic.

The states are represented in zenon through different colours of the active Phase or Operation. Some examples are given below.



Idle.

Running. Command parameters have been sent to the PLC, and the PLC is executing the control process.

Pausing. The Batch engine has requested to the PLC a pause state.

Paused. The PLC has executed the required control sequence to place the equipment in a paused state.

Complete. The PLC has executed the required control, the done condition has been achieved, the phase has completed.

Additionally each phase has a symbol displaying the condition of the phase, below are some examples:

Symbol	Meaning
	Phase starts
	Waiting for unit allocation. The unit of the phase is already being used in another recipe.
	Whilst waiting for the input interlocking.
	▶ During the execution of a phase and the waiting for Reaction finished. ▶ With transitions: whilst running and waiting for transition condition. ▶ With end simultaneous sequence: Waiting for all branches combined.

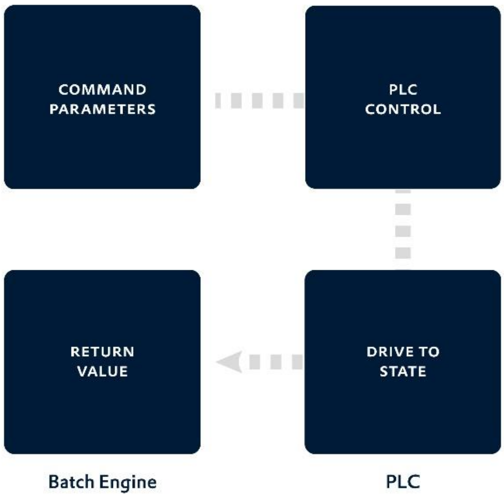
The communication mechanism between the Batch Control and the equipment PLC in zenon is not fixed. Each phase is open to configure specific logic to achieve the required synchronization. This flexibility offers the freedom to create the desired interface for each specific system, and allows for easy integration into existing control systems and infrastructure.

The batch phase sends synchronization information based on events within the phase. Each phase can use any tag within the phase, with a unique logic expression to define the condition. Therefore the communication variables are selected, and the condition for each state is defined. This very flexible approach has some significant advantages over competitors:

- zenon through its communication library can connect to many different control systems
- Any variable can be selected to command and receive information
- State conditions are free to design for each phase
- Each state decision can be equally in either the equipment control, the batch engine, or a mixture of both

A basic example to instruct the temperature control loop. The control loop needs:

- Temperature set point value
- Start signal
- Return status of the loop



When the phase is active in the batch recipe, the set point value is sent to the PLC, together with the start signal.

The PLC then begins to control the temperature to the requested set point value. When the set point is reached, the return status is updated to inform the batch engine. The phase is complete.

The command signal: This simple example only uses one command request to start the control, therefore one could imagine using a Boolean variable to communicate this. Equally an integer could be used, where a certain value is sent to the PLC to indicate a start condition. This facilitates the other states (pausing, aborting, etc.) to be communicated on the same variable. However, equally each state request could be individual Boolean variables. This is open for the designer to decide depending on the existing control or the desired protocol, and can use other variable types and conditions.

The same methods exist for the return status, an individual Boolean value can be defined for the done condition. Or an integer variable can be defined with different values determining different state conditions in the PLC. A logic formula is used to define the state conditions in the batch module, taking information from both batch engine and the equipment PLC.

The done condition decision can be defined in the PLC, for example when the temperature is at set point, the return status is updated to reflect this loop's condition. In this example the PLC is in full control of the batch decision. This can also be designed to give the batch engine full control. If for example the actual temperature is returned from the PLC, the batch engine knows the desired set point, and now has the actual temperature. The decision logic for phase done can be made on these two parameters in the batch engine.

This example demonstrates the clean separation between batch process control and the equipment logic control. The unit control maybe very basic and require the batch engine to control and decide on all loop executions. Or, the equipment PLC may be very intelligent, and require only the start and controlling parameters to sequence through the control process.

zenon provides an open mechanism to facilitate the control structure needed for each installation. This approach minimizes the design or change effort needed on the equipment PLC to provide efficient synchronization to zenon.

Tag hierarchy

Each tag defined in the phase has a scope of influence. If we take the previous temperature control loop example.

1. The synchronization between the equipment PLC and the batch engine is an engineering requirement. E.g. to start the phase a command integer variable sends a value of ‘2’ to the PLC. Once defined at the design stage, these values will not change, therefore this value is only available in the zenon editor at the design stage.

2. In contrast the temperature set point is a process variable, this may change from one product to another, and needs to be available in each recipe. The Master Recipe defines the process flow and control, therefore this variable would be selected to be ‘Changeable in the Master Recipe’.

3. Certain variables may need to be ‘Changeable in the Control Recipe’. Once a tag is available in the Master Recipe, it can be selected to be changed in the Control Recipe. An example would be for material quantities, the Master Recipe defines the process to produce a certain product, and only at the Control Recipe execution is the actual quantity know.

ISA-88 fast facts

- Batch Control separates the Process control from the Equipment logic control.
- Process change or optimization requires no change to the Equipment PLC.
- Significant saving on validation effort, time and cost.
- Master Recipes contain the process flow, and control parameters.
- Control Recipes can only be executed once, providing high traceability to this unique production instance.
- The State Model provides a tight synchronization between Process control and Equipment control.
- The synchronization and communication is flexible in zenon, configure your desired interface, for existing and new equipment. Bring different control systems and devices into the same batch control.
- zenon ergonomic interfaces provide a highly visible representation of the state model and equipment conditions.
- Flexibility and configuration minimizes the design effort or change needed to implement Batch Control.

3. Communication and integration

The Batch Control module is an integral module in zenon Supervisor, developed by COPA-DATA at our development centre in our headquarters. This seamless integration gives the Batch Control Module access to all variables defined in the project. Therefore through the extensive zenon communication library, a wide selection of equipment and industrial devices can be integrated into a common control system.

This open integration allows Batch Control systems to be integrated into existing process control environments, therefore bringing all necessary control systems into the same batch control environment. The different systems can contain different devices, PLC's and industrial networks.

The Batch Control module is in the same design environment as the other zenon control functions, it has full access to all functions contained in the project, and all variables in the project. Therefore the Batch Control module fits into the full SCADA environment with for example Historian, Reporting, SQL integration, and ERP integration.

Communication freedom extends vertically and facilitates higher level system integration. Interaction and control can take place, for example accessing functions which create batch recipes, execute and control batches. Batch data can be communicated to databases and MES or ERP systems.

Communication fast facts

- Communication freedom, connect to any PLC, device, or industrial network.
- Integrate into high level systems, SQL connectivity, OPC-UA, OPC-DA.
- Batch Control has full access to all variables in zenon, and all zenon functions.

4. The design environment

The zenon editor defines how each phase physically connects with the control equipment, how the synchronization mechanism and timing between the phases and the equipment takes place, and integrates each phase with zenon functions within the project and audit-trail.

Phases and tags

The Phase is the principal component in the batch engine. It links directly to the equipment control in the PLC, having command and return tags linked to real world variables, which can be from different communication drivers.

Phases are grouped together in Units. A Unit usually relates to a physical piece of process equipment, e.g. a reactor or a mixing vessel are Units. A Unit can have an unrestricted number of phases.

At the phase level the engineering and synchronization is defined, for example the logic for the phase done condition, or paused, held, and stopped conditions. Phase reactions, interlocking conditions, waiting time, minimum & maximum execution times, are defined. Different Control Strategies can be designed in a phase to change equipment functionality in the PLC.

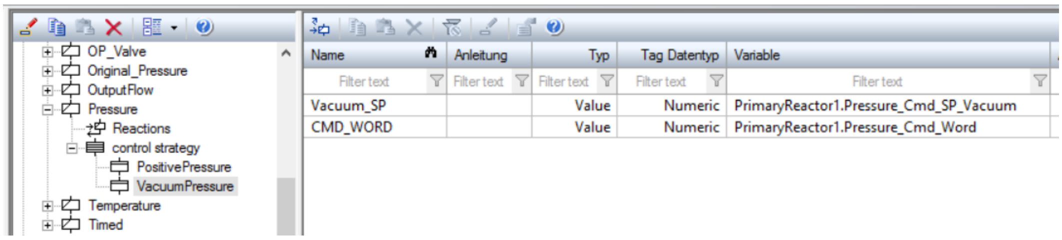
Control Strategies

Within the same phase, different Control Strategies can be defined to select different functionalities in the equipment PLC. A reactor pressure control loop for example, has the control possibility to either pressurize or depressurize the reactor. Using Control Strategies only one phase would be defined, with two individual control strategies for positive and negative pressures.

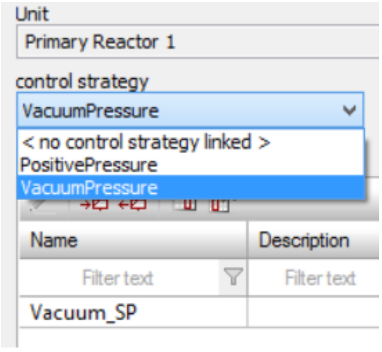
When activated, each control strategy can select different Command Tags, and different values within these command tags, therefore communicating different functionality requests to the equipment PLC. In the PLC it could be anticipated that the same pressure sensor and control loop is engaged, with different parameters selecting a different physical output.

The advantage of Control Strategies is to reduce the number of phases, simplify the logic and minimize potential errors. In this example there is only one pressure control for the reactor, all the intelligence for this control is contained in one phase. Therefore the control and monitoring is directed around this common control.

The control strategy is configured in the editor, and its use is defined in the Master Recipe. The actual selection of control strategies cannot be changed in the Control Recipe, however tags within the control strategy can be selected to be changeable in the control recipe.



Engineering editor phase configuration, showing the possibilities of two control strategies, with the Vacuum strategy properties being displayed.



Runtime Master Recipe selection, showing the possibilities to select either Positive or Vacuum pressure control. Here the Vacuum strategy has been selected.

Reactions

All Batch events can trigger a reaction. The firing of a reaction can either

- Send a value to any command tag defined within the phase.
- Write an entry in the Audit-Trail, to record specific information.
- Call a zenon function.
- Copy one phase tag value into another phase tag.
- Influence the recipe engine, e.g. change to automatic, semi-automatic, or manual.
- Influence the recipe state, e.g. change to any state within the ISA-88 state model.

The batch events which trigger a reaction are grouped as follows:

- Phase events. E.g. Phase started, interlocking blocked, finished writing command tags, done condition completed.
- Phase warning/errors. E.g. waiting periods exceeded, phase started multiple times.
- Recipe state change. Following the ISA-88 state model: Paused, holding, aborted, etc.
- Recipe mode change, to manual, semi-automatic, or automatic.
- Loss of communication
- PLC error.

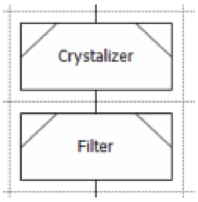
Through reactions many things are realized. Overall control of the project is established. The synchronization between the equipment logic and the batch engine is constructed. Specific GMP events are recorded. And a wider integration into the zenon project is established through the function calls.

Similarly function calls within zenon can influence the batch recipe, to create Control Recipes, start Control Recipes, change the recipe mode (manual, automatic, etc.), or change the recipe state (pausing, aborting, etc.). Through this mechanism external project events can command overall control, for example MES or ERP integration.

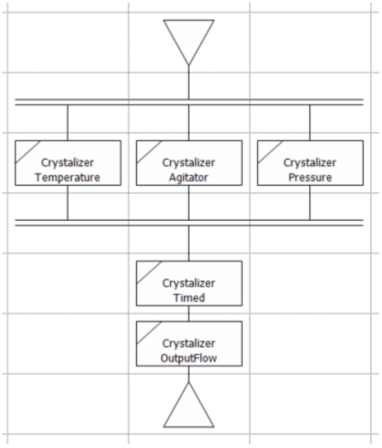
Operations

Recipes can provide the process flow entirely constructed out of phases. For small process executions with one or two units this method is more than adequate. However with large processes incorporating several units, this method become very involved and complex. 'Operations' contain phases and their process flow, and act like small recipes within the recipe. The Operation displays summary information of the phases contain within, therefore in the Runtime a high-level view of the process can be generated, and if needed the details drilled down for more information.

Below are two Operations for the units 'Filter' and 'Crystallizer', this is the high level view from the Recipe. The two diagrams that follow, display the phases contained within the Operations. The Crystallizer has a PFC diagram, and the Filter has a Matrix diagram, these are described in more detail later in this document.



Operation high level view from the recipe



Crystallizer Operation content

	Filter Pressure	Filter Timed	Filter DrainFlow	Filter OutputFlow
1 Pressurize	Active	Inactive	Inactive	Inactive
2 Filter time	Inactive	Active	Inactive	Inactive
3 Drain	Inactive	Inactive	Active	Inactive
4 Output Flow	Inactive	Inactive	Inactive	Active

Filter Operation content

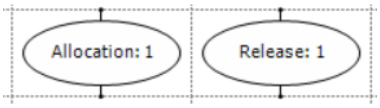
Operations are stored in the run-time project, and can be included in each Master Recipe. Each use of the Operation is an individual instance, and holds its own parameters.

Allocations

The Batch Control module can execute multiple recipes simultaneously. The Batch Control module can command single processes, or plant wide installations with multiple production processes being executed. Therefore some level of reservation of equipment must take place, for example a single reactor vessel cannot be used by two different recipes in at the same time.

Without using allocations in the recipe, when a phase is active, the Unit this phase belongs to is reserved entirely for that recipe. When the phase terminates the Unit is then released automatically.

In batch installations where different production processes are executed simultaneously, security needs to be established to reserve specific equipment needed for each recipe. Reservation of Units and their release can be instructed dynamically in the recipe through Allocations:



These are placed in the recipe, and specifically reserve or release any equipment at the defined point in the recipe.

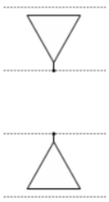
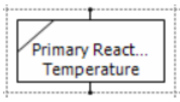
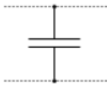
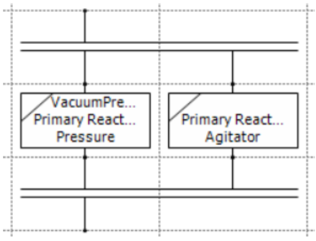
Design environment fast facts

- Flexible integration to the equipment control PLC, with an open interface, configure specific synchronization.
- Control Strategies reduce the number of phases, simplify the equipment logic, reduce tag count, and minimize possible errors.
- Reactions respond to events, within the batch control, and external in the SCADA project.
- Wider integration is bi-directional, Batch Control can react to events, external events can influence the Batch Control.
- Batch functions aid the integration into the supply chain, MES, and ERP systems.
- Tags are restricted to their scope of influence, i.e. Engineering, Process, Production Operations.
- Multiple recipes can be executed simultaneously.
- Equipment can be reserved and released dynamically within the recipe.
- Operations relieve complexity, and provide a high level view of the process.
- zenon's Batch Control is ISA 88 compliant.
- Batch Control has full use of zenon functionality and all project variables.

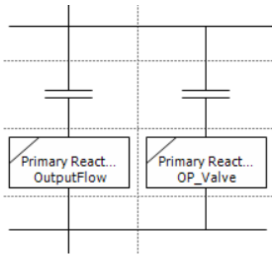
5. Master Recipe design

The Master Recipe holds the process knowledge of the batch. In the design environment we have seen how there is a separation between equipment logic control and process control. The Master Recipe prescribes how the process should be executed and the parameters which these process elements are controlled.

The PFC editor in zenon follows the ISA-88 requirements, it is a process flow chart description of how the process sequence is executed, and contains the process parameters to control within each phase. ISA-88 states that the following control elements are available in the recipe:

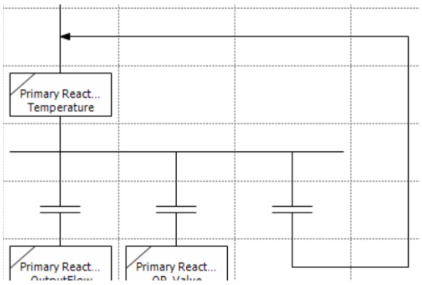
Start and End points		These define the absolute start and end of the recipe. All processes are contained between these two elements.
Phases		Execute the equipment control with the defined parameters.
Transitions		Hold the flow of the recipe until the transition condition becomes true.
Parallel Branch		Independent simultaneous sequences can be executed. Separate Units or multiple processes on the same Unit.

Decision Branch



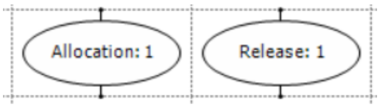
Decisions within the process to direct the process flow depending on conditions.

Jump



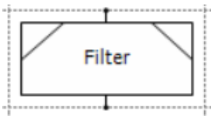
Jump forward or backwards within the recipe.

Allocation / Release



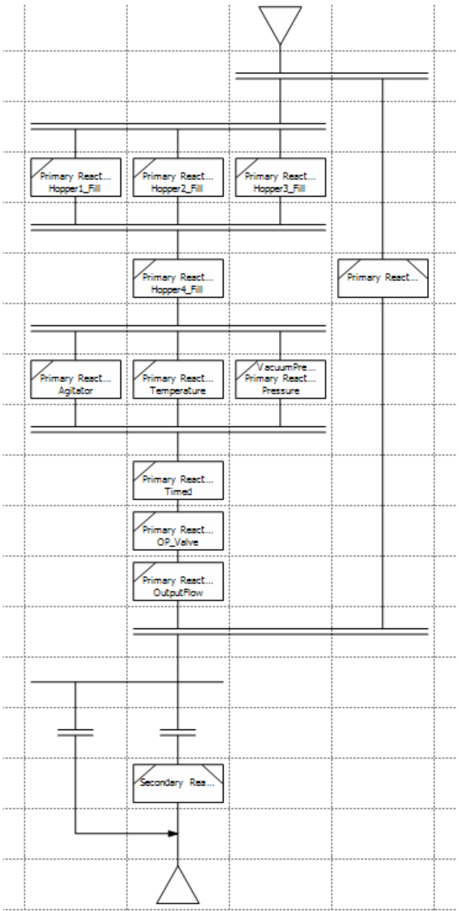
Dynamic Reservation & Release of the Unit equipment.

Operations



Contain groups of phases.

An example PFC batch recipe, showing the above functionalities in a process flow:



There is a second recipe editor in zenon. The Matrix editor uses the same structure with phases and operations separating the equipment control from the process control in the recipe.

	Global StartBatch	Primary Reactor...	Primary Reactor...	Secondary React...	Crystalizer	Filter	ConicalMixer Temperature	ConicalMixer Pressure	ConicalMixer Agitator	ConicalMixer Timed	ConicalMixer OutputFlow	Containers Fill	Global AcceptBatch
1 Operator Start	Active	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive
2 PR 1 & 2	Inactive	Active	Active	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive
3 PR 1 & 2	Inactive	Inactive	Inactive	Active	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive
4 PR 1 & 2	Inactive	Inactive	Inactive	Inactive	Active	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive
5 PR 1 & 2	Inactive	Inactive	Inactive	Inactive	Inactive	Active	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive
6 Conical Mixer	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Active	Active	Active	Inactive	Inactive	Inactive	Inactive
7 Conical Mixer	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Active	Inactive	Inactive	Inactive
8 Fill Containers	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Active	Active	Inactive
9 Operator Accept	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Active

Using a sequencer like structure to initiate the phases and operations. When each Phase or Operation for a certain step is complete, the Matrix moves on to the next step in the sequence.

Test mode

During normal operation the Control Recipe executes the batch for each production run. However pre-release of a Master Recipe may involve unrestricted testing, therefore a test mode has been implemented to be able to execute a Master Recipe on live equipment.

The Master Recipe needs to be complete and error free before test mode is permitted. Then the recipe executes in the same manner as the Control Recipe.

Master Recipe fast facts

- The Master Recipe holds the process knowledge of the batch.
- Master Recipes prescribe the process flow, and process parameters.
- ISA-88 functionality with: Phases, Operations, Parallel branch, Decision branch, Transitions, Jumps, Allocation and Release.
- Certain process tags can be selected to be changeable in the Master Recipe.
- The PFC editor is the ISA-88 process flow chart recipe environment.
- The Matrix editor provides a simpler sequence control of batch Phases and Operations.
- Test mode facilities the commissioning of Master Recipe, test live on equipment without the creation of Control Recipes.
- When enabled Master Recipe versioning allows only one recipe to be released, secure inbuilt Quality Management System.

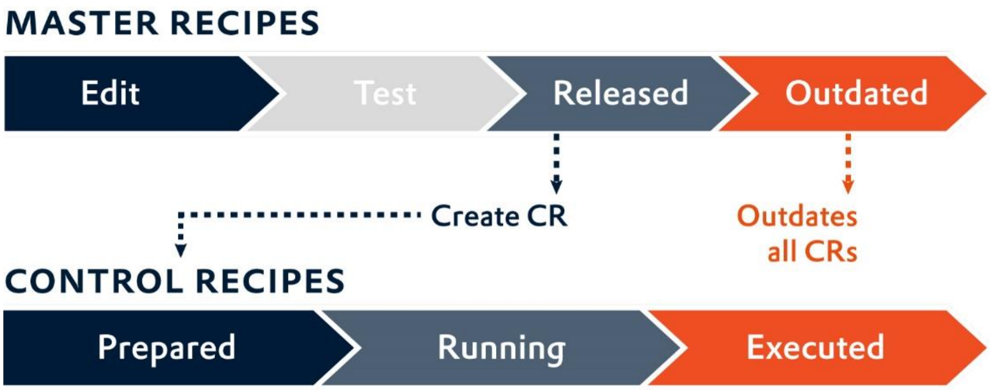
6. Control Recipes

Control Recipes are generated from a released Master Recipe, taking their process flow and defined parameters. In the Master Recipe, certain parameters can be selected to be changeable at the Control Recipe stage, this is to facilitate unique production batches of the same product type.

Control Recipes can only be executed once, and keep a specific execution recording of this batch production. This offers a high degree of traceability for each batch production.

The Control recipe can be generated manually, or via zenon functions. The manual method can use specific screen elements in any batch recipe editor screen, or use functions via buttons etc. to create the Control Recipes. The creation of Control Recipes and execution of the recipe can be implemented in one set of functions executed in a sequence, so that from a master recipe selection, a control recipe can be created and started automatically, without the need for an operator to do this manually.

Below is the life-cycle of a recipe from Master Recipe conception, through Control Recipe creation, and obsolescence.



The use of zenon functions allows for external systems such as a MES or ERP system to create, start, and control production Control Recipes from known Master recipes.

Job ID

Each Control Recipe is unique, with a unique name within the scope of the Master Recipe. When the production operation involves more than just the batch control area, for example when the batch is incorporated into a supply chain, a common identification is needed. This is called the 'Job ID', with this the MES or ERP system can trace all aspects of the batch from start to finish. In zenon, Job ID's can be implemented in to the work flow, and a selection is made to request a Job ID when either the Control Recipe is created or when the Control Recipe is started. When enabled the Job ID cannot be empty.

Control Recipe fast facts

- Control Recipes can only be executed once, providing full traceability on each unique production batch.
- Certain tags can be selected to be changeable in the Control Recipe, e.g. for production quantities.
- Control Recipes can be generated manually or via zenon functions. Facilitating user control, SCADA system control, and integration into wider control with MES or ERP.
- 'Job ID' incorporates the Batch Control into the wider scope of batch production, supply chain, MES or ERP integration.
- Control Recipes can only be generated from released Master Recipes.

7. Historian and Reporting

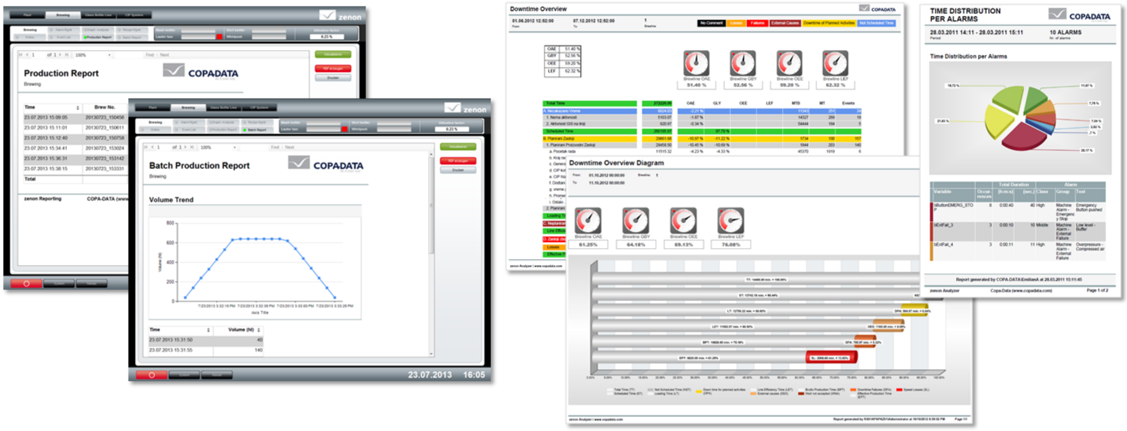
Although not directly enabled through the Batch Module, the Historian and Reporting functionality complements the Batch Module to offer a complete solution in GMP regulated environments.

The Historian archive for a specific batch can be started automatically through using reactions in the Batch recipe, taking the batch name or Job ID as the lot variable for the archive. Historian archives and other data can also be evacuated to external databases such as SQL for example, or use protocols such as OPC-UA, to integrate directly into higher level systems.

The report viewer functionality can automatically produce batch reports taken from the executed batch, or from an historic batch.

The Report Viewer module can produce different reports in relation to batch operation. Used in conjunction with the Historian, regulated batch reports can be executed to present all batch data such as Process variables, CQA's, Audit-Trail, Alarms, Media, Equipment, etc.

Master Recipes can be documented in a report, displaying a graphical representation of the recipe, and include a breakdown of the recipe contents such as real-world variables, tag values, phase done conditions, etc. Control Recipes contain the same content as Master Recipe reports, and include data of the actual recipe execution.



Historian and Reporting fast facts

- zenon functionality provides a complete GMP solution for regulated environments.
- Standalone and Integrated solutions are FDA 21 CFR Part 11 compliant.
- The Historian records selectable batch data, as standalone or integrate solutions.
- The Report Viewer can provide different reports, e.g. Batch Production, Regulation compliance, Quality documentation, Engineering and OEE.
- Link the Historian, Audit-Trail, Alarms, Materials, Equipment, and Users in to the report.
- Provide a graphical representation of the Master Recipe, including the construction of the phases, real-world connections, and process values.
- Provide a graphical representation of the Control Recipe with execution data.

8. FDA 21 CFR Part 11

GMP regulations place strict demands on automation in pharmaceutical production areas. zenon provides an efficient platform in GMP environments, providing FDA 21 CFR Part 11 compliant solutions in a configurable product. With integral Audit-Trail, Alarm management, Historian, User Administration (including Active Directory), Reporting and SQL interfacing. Standalone solutions are fully compliant, and integrated solutions are implemented with the minimum impact on validation.

A GAMP Software Category 4 product, zenon offers the most efficient automation solution to qualify for GMP areas, and validate for Part 11 compliance. Full solutions can be developed without any code, using only zenon configured functions to create for example Batch Control, Data Acquisition, Reporting or SCADA solutions.

Individual batch tasks can be attributed a specific user authorization level. For example creating a Master Recipe may involve a different user authorization to starting a Control Recipe. Therefore safeguarding any activity, forcing the correct workflow defined by the quality management system in place, with all activity logged in the audit-trail.

Part 11 fast facts

- zenon provides FDA 21 CFR Part 11 compliant solutions.
- Standalone applications or integrated solutions are fully Part 11 compliant.
- GAMP software category 4, zenon is a configurable product making validation as efficient as possible.
- Electronic Records & Signatures, Audit-Trail, Alarm management, Historian, User Administration, Report solutions in one application out of the box.

9. User interface

The batch user interface is configurable. Each process or machine is different, with different functionalities needed depending on the user's focus. The batch module fits into the zenon Runtime with elements which can be configured to match the desired functionality. For example:

- A small information screen for the operator, with limited functionality. Filter the recipes to the specific need of that user.
- Quality Management may need more detailed information or historic information on specific processes, and access to recipes in development.
- Operations Management may need a wider view of the production processes currently operating.

The visual environment can be designed around each functionality.

Multi-touch gives the maximum usability to plant or machine control. Incorporate multi-touch gestures, swipe, tap, zoom, two finger swipe, drag & drop, panning, or two handed operation into the project.

An efficient user interface allows for the possibility to merge manual tasks with automated processes. User phases can request that an operator carries out certain tasks via the workstation screen, the phase could also request information from the operator which can then be stored alongside fully automated data. Using such a method would create greater accuracy of data and time stamping. Forcing the phase to wait for the operator to execute the task and return the process values, would eliminate missing entries from a batch. Increasing accuracy and overall quality of the batch.

Use the batch element functionalities and standard screen elements to create the most suitable user experience. To facilitate international projects all texts and variable units (excluding recipe names) are language switchable.

User interface fast facts

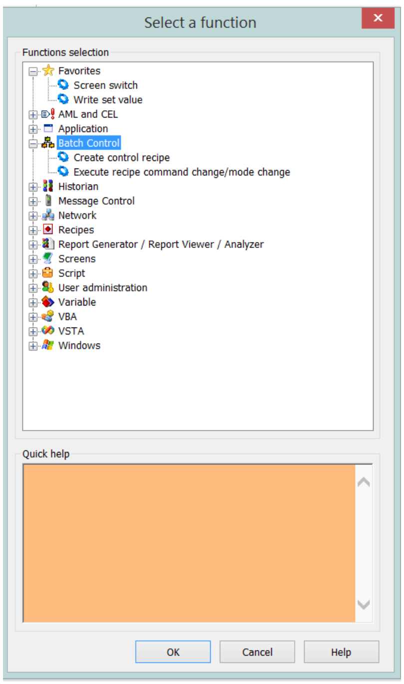
- Each user interface is configurable, create specific batch screens to each user function.
- Multi-Touch technology provides the maximum usability, intuitive interfaces giving maximum control and visualization.
- Soft user phases instruct the operator on manual tasks. Integrate automated and manual tasks into one control system, one audit-trail, and one batch report.
- Eliminate missing batch entries, increase accuracy and quality.
- Match the solution to the local user. All texts and process units are language switchable.

Appendix

Automated creation and start of Control Recipes

The creation and control of Control Recipes can be achieved using zenon functions, this allows for other controls in zenon to command the batch process, and allows for external systems such as MES or ERP systems to command production batch processes.

In the zenon editor there is a function called 'Create control recipe', under the Batch Control node.



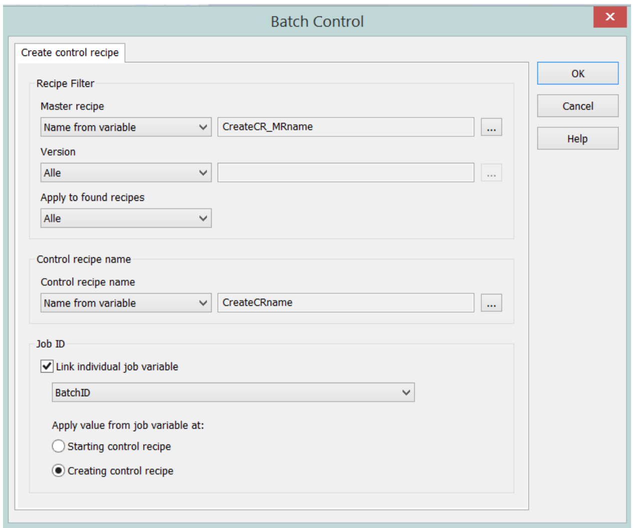
To create a Control Recipe a minimum of two pieces of information are required, the 'Master Recipe name' from which the Control Recipe will be created, and the actual name of the Control Recipe to be created.

The Master Recipe name can be a static value, therefore always the same MR will be used, which is specific to this function call. In this manner different functions are used to create recipes from different Master Recipes. The Master Recipe name can also be contained in a variable, providing a more flexible solution using only one function, this is the function utilized in this example.

Different versions of the Master Recipe can also be selected, either a fixed version, the oldest version, newest version, or take the version from a variable.

The Control recipe name can be automatically generated or taken from a variable. In this example the variable option is used.

In the same function, a Job ID can be requested. When enabled, this requires 'Job ID' (or BatchID) information to be provided, either at the creation of a Control Recipe, or when starting a Control Recipe.



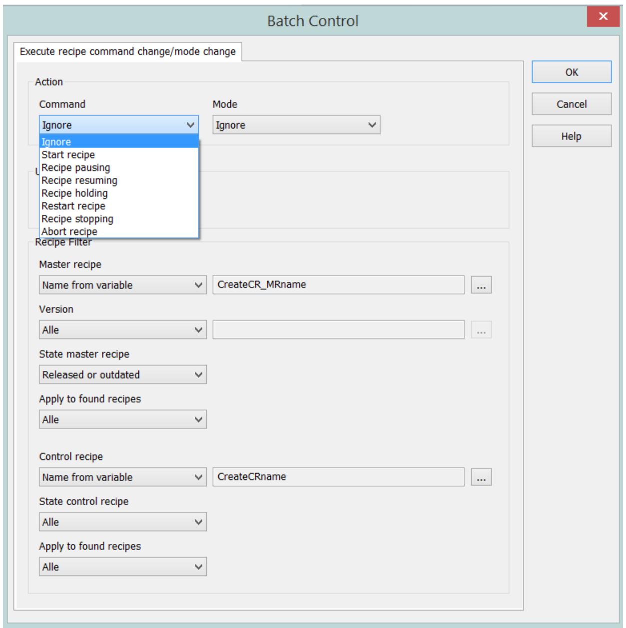
The above dialog provides an example function to create a new Control Recipe, with variables containing the Master and Control recipe name. In this example the Job ID is also requested, which is also contained in a specific variable. Therefore if this function call was to be used by a MES/ERP system, the call would require:

- Master Recipe name, contained in the variable 'CreateCR_MRname'
- Control Recipe name, contained in the variable 'CreateCRname'
- Job ID, contained in the variable 'BatchID'

Therefore this function creates a new Control Recipe 'CreateCRname', from the Master Recipe 'CreateCR_MRname'. At the point of creation, a Job ID is provided from the variable 'BatchID'.

The JobID can also be specified to be defined at the Control Recipe start. In this manner, the Control Recipes can be created with only the Master Recipe and Control Recipe information.

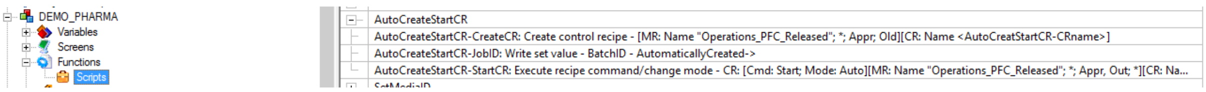
To complement the external creation of Control Recipes, there is a function to start and control a Control Recipe. This uses the same information as the Create Control Recipe function. Therefore full execution of recipes can be commanded from an external system.



The figure above displays the 'execute recipe command' function to start and control an existing Control Recipe.

Together the two functions used together in sequence, can, from a Master Recipe create and start a new Control Recipe. Therefore a simple operation of specifying a Master Recipe to execute can be designed, where in the background a unique Control Recipe has been created, and executed.

To execute a sequence of function calls in zenon, function scripts are used. Here a specific script can be created, then the required functions inserted in the specific sequence needed. The figure below displays a Script 'AutoCreateStartCR', which uses three functions in a sequence to create a control recipe, populate the BatchID variable, then start the newly created Control Recipe. The script is executed using a single zenon function call.



To create a function that from a known Master Recipe, will create a new Control Recipe, then start the Control Recipe; the following sequence must be utilized with the given variables:

Variables:

- Master Recipe name, contained in the variable 'CreateCR_MRname'
- Control Recipe name, contained in the variable 'CreateCRname'
- Job ID, contained in the variable 'BatchID'

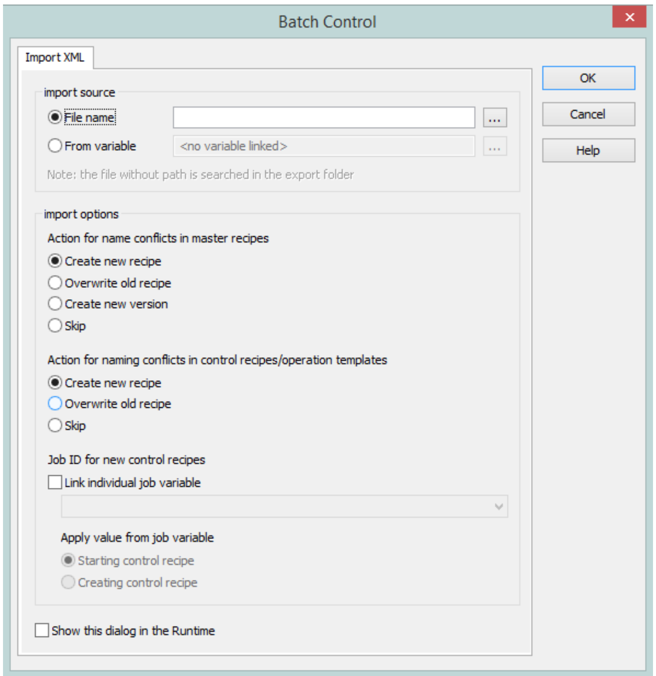
Sequence:

1. Load the variables with the desired batch recipe information:
   a. Known Master Recipe name - *'CreateCR_MRname'*
   b. Specific Control Recipe name, this name must be unique within the Master Recipe context - *'CreateCRname'*
   c. If required the Job ID from the supply chain can be utilized - *'BatchID'*
2. Use the function 'Create control recipe', selecting the variable information given above. This function is found in the Batch Control node of the function selection dialog.
3. Use the function 'Execute recipe command/change mode', selecting the Control Recipe and JobID variables. This function is found in the Batch Control node of the function selection dialog.
4. Create a function Script to execute the sequence of function calls: 1) create control recipe, and 2) start control recipe.
5. Use the 'Execute script' function to call the sequence.

Master Recipe Export & Import

From zenon version 7.50 onwards, it is possible to export and import recipes from the zenon Runtime. Therefore the Master Recipes can be held in an external system, either for security in quality management practices. Or to have for example the MES or ERP system use a specific recipe which is downloaded to the required runtime, then Control Recipes are created and executed from this downloaded recipe.

Master Recipe import



The figure above displays the zenon function call 'Import batch recipes' which will import a recipe into the zenon Runtime system. To import a Master Recipe the source information where the actual Master Recipe is stored, and how to handle conflicts of the same Master Recipe name must be specified.

This function call could then be added to the beginning of the sequence specified in the appendix 'Automated creation and start of Control Recipes'. Therefore a complete Control Recipe creation, can be commanded from an external system such as MES or ERP. The External system can hold the Master Recipes in xml format, then download to the Runtime, create a control recipe, and start a control recipe. The following describes the sequence:

Variables:

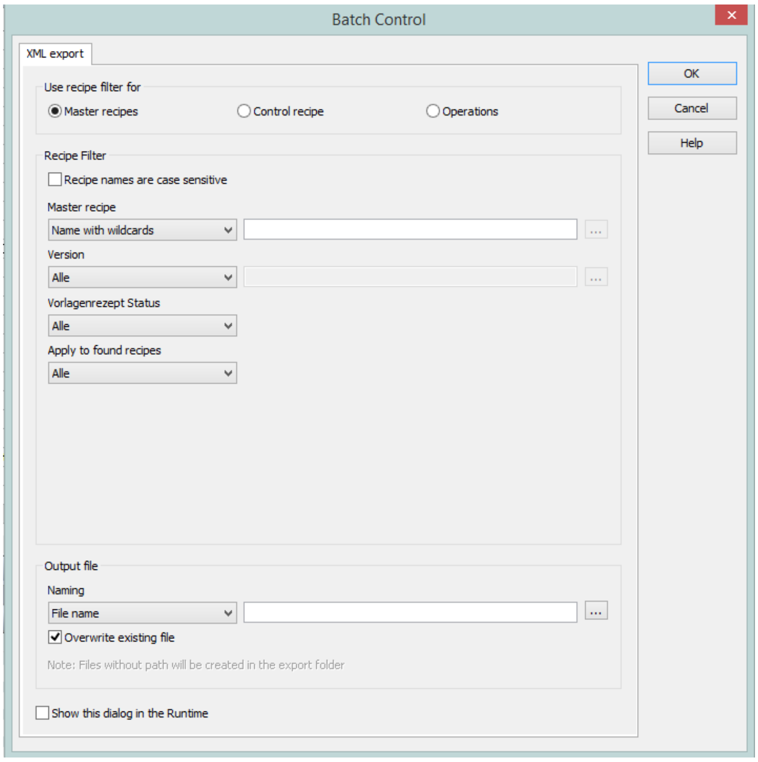
- Master Recipe file name & location.
- Master Recipe name
- Control Recipe name
- Job ID

Sequence:

1. Load the variables with batch information:
    a. Master Recipe name and location.
    b. Known Master Recipe name.
    c. Specific Control Recipe name.
    d. If required the Job ID from the supply chain can be utilized - *'BatchID'*
2. Import the Master Recipe into the required Runtime system, using the name and location specified.
3. Use the function *'Create control recipe'* selecting the variable information. This function is found in the Batch Control node of the function selection dialog.
4. Use the function 'Execute recipe command/change mode', selecting the Control Recipe and JobID variables. This function is found in the Batch Control node of the function selection dialog.
5. Create a function Script to execute the sequence of function calls.
6. Use the 'Execute script' function to call the sequence.

Master Recipe export

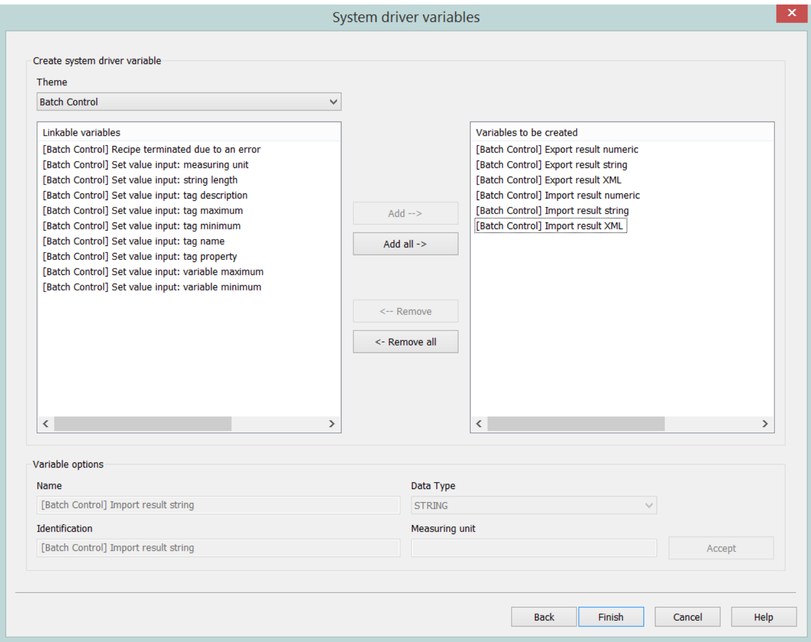
Master Recipes can be exported form the Runtime system using the zenon function 'Export batch recipes' shown in the dialog below.



The function requires the specific Master Recipe name, together with version and status information if needed. The Master Recipe is exported to a static file location, or a location provided in a variable. Similar to the import function previously defined. This function can be specified and commanded by an external system such as a MES or ERP for example.

Batch import / export system variables

To verify the Master Recipe was imported or exported successfully, system variables in the zenon runtime provide feedback on the process. In the zenon editor project, create a new variable. In the create variable dialog box select 'SYSDRV' for the driver type, and 'Systemvariable' as the driver object type. The button 'Next' is then enabled, click this to open the dialog box below, and select the 'Batch Control' Theme.



The two variables of interest are '[Batch Control] Export result numeric' and [Batch Control] Import result numeric'. These provide a status on the recipe transfer, thus handling transitions and transfer errors, the help description is given below.

[Batch Control] Export result numeric	DINT	The numeric variable is filled with the number of errors that occurred. Example: ▪ *-1*: is being executed ▪ *0*: Initialization value read successfully ▪ from 1: Number of errors that occurred

[Batch Control] Import result numeric	DINT	The numeric variable is filled with the number of errors that occurred. Example: ▪ *-1*: is being executed ▪ *0*: Initialization value read successfully ▪ from 1: Number of errors that occurred