

Doc Type	Tech Notes
Doc Id	TN3011
Last Modified Date	06/25/2018

Multiplexing InBatch Phase

SUMMARY

This document describes a technique for multiplexing (mapping many to one or at least many to few) InBatch phases to controller phases. Multiplexing can limit the amount of PLC programming needed for Connections in Transfer class Phases by re-using the PLC code within another Phase. Phase multiplexing allows the control system to have complete InBatch functionality with minimal memory and programming required in the controller.

SITUATION

Two scenarios which can take advantage of this approach. This approach can be used with other Phases as well.

Example 1: Demonstrated a large tank farm for bulk ingredients and a common header line to the processing units.

Example 2: Mobile vessels used for minor ingredient preparation.

By Using this approach:

- The ability to assign multiple InBatch tags with the same address in your controller.
- Multiple InBatch tags can share memory in the PLC.
- Phase multiplexing involves the assigning of multiple phase control, status.
- Parameter tags to the same addresses in the controller.

Phase Multiplexing:

- Reduces the memory burden on the controller when implementing a Comprehensive model in InBatch.
- Takes full benefit of the InBatch model which uses connections.

InBatch Rules that need to be followed:

- InBatch can never run the same as the same time.
- Segments can be used to prevent multiple transfers from occurring at the same time.

ACTION

Example 1

This example is a tank farm which consists of 20 bulk ingredient tanks (Process Class: BULK), connected by a common header line to our processing facility which includes 10 processing vessels (Process Class: REACTOR).

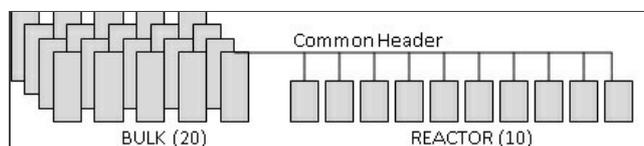


Figure 1: Tank Farm

- The InBatch model requires a transfer phase on this class to move material from a BULK unit to a REACTOR unit.
- Based on the example the math reveals 10 Reactors x 20 bulk tank = 200 connections for the transfer class from BULK to REACTOR.
- Without using Multiplexing the InBatch Model would require 200 phases that need to be defined in the PLC. Each of these phases may require eleven (11) discrete tags for control/status and at least two (2) analog tags for target and actual parameters.
- Phase Multiplexing allows the user to model all connections and use one phase in the controller instead of 200 phases one phase will have 11 bits for control/status and 2 analog registers for target and actual data.
- A third register must be added to provide the source/destination code so the controller phase knows which valves to open and which pumps to run, etc. in order to complete the transfer. The source/destination code will be unique to each connection in the transfer class.
- A segment is required in the model that is common to all the connections to ensure only one can be allocated at once.

The Schneider Electric industrial software business and AVEVA have merged to trade as AVEVA Group plc, a UK listed company. The Schneider Electric and Life Is On trademarks are owned by Schneider Electric and are being licensed to AVEVA by Schneider Electric.

[Terms of Use](#) | [Privacy Policy](#)

- TagLinker is used to assign all InBatch phase START tags to the address of the start command in the controller and all of the InBatch READY tags to the ready state bit in the controller, etc. The target and actual parameter extension tags in InBatch will be assigned to the controller target and actual registers.
- InBatch phase Preact parameters extension tags will be assigned to our third controller register - source/destination code.
- Preact is a constant value assigned to each connection in the model. It is intended to be used as a value that indicates how much material will “trickle” into the destination unit after the valves have closed. This value can be used by controller logic as an early warning to shut the valves in advance to presumably hit the target value every time. InBatch sends this constant through a tag to the controller every time the phase is run.

Example 2

This example involves mobile tanks (totes) supplying our REACTORs with minor ingredients that have perhaps been premixed. Mobile tanks will be modeled as Units in the InBatch Model.

- The system required 80 totes to supply 10 Reactors this translates into 800 connections in the InBatch Model.
- The solution is to provide 1 controller phase for each REACTOR to result in multiplexing 800 phases in InBatch into 10 phases in the controller. The number of phases you need in the controller is always dictated by how many phases can be simultaneously active.
- In this example it is not necessary for our Preact (aka Source/Destination code) to provide the destination. The destination is known by the phase because each phase is dedicated to a REACTOR destination. One phase will handle all transfers to REACTOR1, another will handle all transfers to REACTOR 2 and so on. Therefore, we can use the Preact tags to identify the source unit - the mobile vessel. In this example we use the Preact as a source code and all connections from the same source unit will have the same code. This will allow phases in the controller to know which source tank is providing the material.
- All transfer phases on all connections with the same destination will have common addressing in the controller. The phase in the controller will be responsible for all phases with it's REACTOR as the destination. This means that all transfer phases going to REACTOR1 will have the same address for their START tags but this address will be different for those phases with a destination of REACTOR2. REACTOR2 has it's own phase for it's transfers. We will model a segment on each REACTOR which is common to all connections leading to the REACTOR to make sure only one is active.
- In this example 11,200 InBatch tags have been multiplexed into 140 controller addresses.
- Preact value labels can be changed as well as all text in the InBatch configuration and runtime applications. Refer to the InBatch user guide for more information.