Introduction to ProAction PCEE

This document will assist the beginning user in setting up a process using the Process Control Engine and Editor (PCEE). There is a list of points that will help make the building of a process easier. Once the editor is open, there will be many figures and descriptions of all the conditions and actions in the help file.

PCEE is developed around a step, condition, and action philosophy.

A step contains conditions and actions (see example below). A condition is a situation that must be met to accomplish the true actions. If the conditions are not met, the false actions are executed. When a transition happens to the condition, the corresponding true or false actions will be executed. All actions will all be executed even after a pause.

Example:

The condition could be a Weight Compare (Gross is trip higher than 100).

The false action could be to turn on a digital output (Digital I/O 1 is On, since weight is below 1000).

The true action could be to turn off the digital output (Digital I/O 1 is Off, since the weight is above 1000).

Steps are typically sequenced one after another. This is accomplished by actions that allow a step to deactivate itself and to activate other steps. The true action list for the above example is:

- Digital I/O is Off
- Deactivate Step 1
- Activate Step 2

When a step is enabled, the actions will only be executed once until a condition transitions. This transition can be from true to false, or from false to true. Deactivated steps are not evaluated until activated by another step. When a step is activated (enabled), either the true or false actions will be executed.

Writing a PCE Process

The first step to writing a PCE Process is to decide what small routines are to be built into the total process. This could include steps for:

- The batch process
- Entering a new target weight
- Printing the weight
- Other functions relating to the process

Before the steps are created, you should write out the sequence of the process/functions on a worksheet. Then, consider the following in working through the details:

- The softkeys or inputs/outputs required
- The steps "enabled" at power-up (at least one is required to start the process)
- Determine the condition to evaluate for each step, and the actions
- What the print formats are going to be
- What database tables need to be created and the data stored in each field

As each step is built, keep in mind the True, False, and Pause lists execute the actions in the order they are listed, meaning an action such as "print" should precede an action such as "tare."

Steps for a particular routine can be built and tested using just that routine, and when all of the smaller routines are complete, a final merge can be done to finalize the process.

Within the PCE, there are four temporary variables called user data. They are used as an interface between:

- The steps and the database (such as reading or writing data)
- The user and the database (such as searching by an ID)
- The user and the step compares (such as setting a target value)

All data goes through these temporary variables as a "Get" or "Set." A "Get" is used to get data from a user or from a condition in a step. A "Set" is used to put data into a condition in a step.

Each variable supports a different type of data: integer (whole numbers without a decimal point); real (numbers that can contain a decimal point); string (alphanumeric printable data); and datetime (preset format for the date and time to be stored and printed). There is one location for each type of user data. The User Clear action will clear out one of the four variables based on its type.

Performing a Merge

There are some checks to perform whenever merging is used because the editor assigns condition numbers to each step:

1. Re-Check each step's Activate and Deactivate to ensure they are still aimed at the correct step.
2. If any Set or Get Compare actions are used, verify the condition number, as the editor will change it as it builds a merged process.
3. If a Step Compare is used, verify the correct step is being evaluated.

NOTE: *The above checks also apply if a step is moved up or down. If you move a step, ensure condition numbers are updated accordingly.*

Pause, Reset, and Restore Process

There are three unique actions that need a bit of clarification: the Pause Process, Reset Process, and Restore Process. These actions are used to pause the process and then either restore (continue) or reset (abort) the process. These actions are typically accomplished with a softkey or a digital input.

These actions will only affect the steps that have the Pausable flag set. Setting this flag in any of the actual process steps means only those steps will be affected. When a Pause Process is initiated, only the pause actions for the current step (or other enabled, pausable steps) are run. Those steps without the pausable flag set will still operate as normal. For example, a free-running independent weight compare or any non-pausable, enabled step can continue to activate/deactivate other non-pausable steps.

A Pause will stop the pausable steps. The state of each step is then saved. If a Restore is then executed, the steps' states are restored and the process continues where it left off. If the Reset action is executed, all pausable steps go back to the default state and the process will need to be manually restarted based on the setup of the process.

Along with the PCE Editor, there is also a Database Editor to build and download a database that can be used by PCEE. First a Schema is built. This is the framework of the database. This will include the name, number of fields, and the type of data each field will contain. This data can be used for displaying prompts, gathering data during a batch process, and as a warehouse for any data that the PCE needs to use.

Use of the database is addressed using the database commands discussed later in this document.

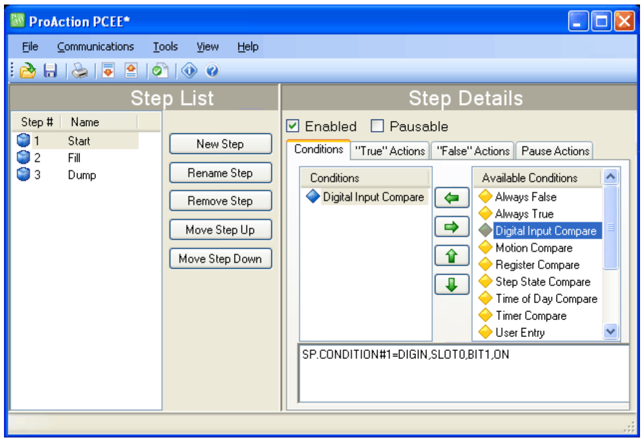
Revolution III

Whenever a PCE process is built, be sure to build a Revolution III file using the 720i module so that all configuration parameters match the process. The Revolution III file will define any required digital i/o, softkeys, and the print formats (AuxFmt) used.

In Revolution III, there is a Monitor Mode in the *Tools* tab of the editor, which can be used to set individual parameters such as a DigOut to Output or Input.

Setting Up a Simple Three-Step Fill

The following example will set up three steps to start the process with DigIn 1, turn on DigOut 2 until the target is reached (100 lbs), and wait for the scale to empty before allowing a restart.



Step #	Name
1	Start
2	Fill
3	Dump

☑ Enabled ☐ Pausable

SP.CONDITION#1=DIGIN,SLOT0,BIT1,ON

To create this process:

1. Install both the PCEE program and the DBE program.
2. Open the PCEE program.
3. Click New Step button.
   The *Step* dialog box appears.
4. In the *Step Name* text box, type the desired name for this step.
5. Ensure Auto-Sequence is checked.
6. Click OK.
   The step will appear in the left column and the Auto Sequence will add the Deactivate for this step and an Activate for the next step.
7. Click on Enable to set the default to *Enabled* for this step.
8. Under *Available Conditions*, click on Digital Input Compare and click the Left Arrow button. This will use a DigIn to start this process.
9. Select bit 1 in the parameter section and ON for Target State.
10. Using Rev III, ensure that DIO bit 1 on Slot 0 is set to INPUT.
11. Click New Step to create the second step and name it.

12. Under *Available Conditions*, click on Weight Compare and click the Left Arrow button.
13. Set the parameters to Higher and a value of 100. We won't use a Preact for now. This is our fill step.
14. Select the *"True Actions"* tab.
15. Click Digital Output and click the Left Arrow button.
16. In the parameters, set to bit 2 and OFF. This will turn our relay off when the target is reached.
17. Select the *"False Actions"* tab.
18. Click Digital Output and click the Left Arrow button.
19. Set the parameters to bit 2 and select ON. This will turn our relay on until the target is reached.
20. Using Rev III, ensure that *DIO bit 2* is set to OUTPUT.
21. Click the New Step button for the final step and name it.
22. Under *Available Conditions*, click on Weight Compare and click the Left Arrow button.
23. Set the parameters to Lower and a value of 5. This is our Dump step.
24. Under Step 3's true actions, adjust the Activate to Step 1 instead of Step 4. Auto Sequence will continue to enable a next step; however, if there is no next step, we need to change the activate back to our first step.
25. From the *Tools* menu, select Validate to check the process steps for any basic errors.
26. From the *File* menu, select Save As... and select a file name and save location.
27. Open Revolution III and Connect to 720i.
28. Download the matching Rev III file.
29. Download the file you saved in Step 26.
30. Press SaveExit softkey on the 720i to open and execute these files.
31. Verify all operations of your files.

For ease in debugging your process, there is an EDP command that will dump the raw list of commands you have sent to the 720i, "SPSUMP." The first half of the list is all the conditions and actions. The second half lists the steps and associated conditions and actions by numeric reference.

NOTE: The Enable flag will be True for the steps that are presently active. If they were set to ENABLED on download and no other step ever turns it off, it should still be ENABLED. If it was off at download and is part of a process sequence, then it may be on now to represent where the process is at this time. The EDP command can be used every step of the process to watch the progress. It is very useful.

Helpful Hints

PCEE has a special feature, called auto-sequence, that when enabled, will automatically add an Activate and Deactivate action to the current step. This is used to consecutively enable and disable steps in a routine to form a process flow.

The Enable is a preset flag that you set to activate a step on a download or reset. That is its default state. A Restore will always put this flag back to a default state when processed. It can also be turned ON or OFF from any other step to form a sequence of steps. If it stays on and no other step turns it off, it can be considered free-running and is always examined by the PCE.

The Pausable flag is used for any step to be included in a pause action. Any time the process needs to be paused, this pause list in each pausable step will be executed. This flag also marks steps to be included in a Reset action to put them back to its default state, particularly to put the enable flag back to its default state.

The Reset Process with reset all pausable steps back to the downloaded default settings. If a pause was not performed prior to this action, it will be performed and then the reset will be done.

The Restore action sets all paused steps back to where they were when the pause was executed so the process can continue where it left off.

There are 32 timers that are used as a "one-shot timer." They can be started inside a step and then another step can use a Timer Compare to check if the timer is still running or stopped, and perform the appropriate actions based on the timer's state.

A simple hint for naming the steps is to insert a character or number before the step name, i.e. "A. Start." The next step would then be "A. Setup," for example. This routine is then saved as group of steps and the next routine can be built with a prefix of "B." When the merge is done, this simplified identifying which steps are grouped together in the total process.

A simple Weight Compare step would have a Weight Compare condition of Higher than a Target with the false action setting a digital output to ON, and the true action would turn the digital output OFF when the Compare is true.

NOTE: Once a negative value is used for a Weight Compare, any new value set from a Set Compare will retain the negative sign.

Within the Weight Compare steps we can add actions that are the same as the old setpoints used, such as Tare Scale, Clear Tare, Accumulate, Clear Accumulator, and Turn a Digital Output On or OFF.

Whenever a User Get action is used, it should be used with a User Entry condition as the next step. The User Get will prompt the user for a new value. At this time, an Activate should start a new step with the User Entry condition that will pause processing of this stream of steps until the operator presses either the ENTER key or a Cancel key. This ensures the process has the requested data before it continues.

Printing can be done by using a Print action and selecting one of 20 AuxFmts. These formats can be configured using Rev III to print anything required. The <USx> strings can be added to any or all formats and there are 99 of these to use. A Set User Print Text will fill these fields. They are strings but any type of data, (real, integer, string or DT) can be inserted and the 720i will convert it to a printable string.

Registers are a group of 256 variable registers used to hold a number that can be incremented or decremented as a counter. There is also a Clear Register (only 1) and a Clear Register All (all of them).

Math functions of Add, Subtract, Multiply and Divide are supplied and used as follows:

- There are two sources for the data to be worked with. *Immediate* is the value within the step to be used. *Database* is the field of the data to be used. This source is then added, subtracted, etc. from the user variables. The result is then left in the user variable so it can be either printed or written to the database. Also only *Integers* and *Real* variables can be used.

Database Information

A Database consists of a group of fields that comprise a record; Record 1 = F1, F2, F3, F4, F5 etc.

Record 2 = F1, F2, F3, F4, F5 etc.

To ensure a simple search, try to make *Field 1* an integer to be used as an index number. This will allow you to search for an index and then get the other fields as you need them, or insert data to them after a successful search.

A *Sort* of the database by index will sort all records by the index and place them in order. This is not mandatory, but makes working with a database much easier.

The database index is the database number assigned from the Database Editor. DB 1 is **always** the embedded Truck Database for Weigh In / Weigh out functions. The next is DB 2 which is the first PCE database that can be used. Eight databases are allowed, but as the number increases the size of each is reduced. Slot is usually 0 as memory card expansion options are not allowed in the 720i.

There are two actions that are similar but need further definition. They are the DB Write and DB Capture. The DB Write will put a user variable into a field specified by the step. The DB Capture will put a captured value specified by the step into a database field. Captured data can be trip weight from a Weight Compare condition, Weight Compare preact value, motion status, timer state, DigIn or step state, current time and date, or a current register value.

Add Record, Delete Record, and Copy Record pertains to the complete record with all fields involved.

DB Write, DB Read, and DB Clear only affect the specified field of the current record.

A Set Compare or Get Compare is the method used to either retrieve the Compare value or set it to a new value. These values would be such as a Weight Compare (Trip) value. This is assigned to a condition number, not a step number. This is because a step could have multiple conditions of similar type so we need to aim at the exact condition.

From the *Tools* menu, there is a selection for Text View. There is also a keyboard shortcut of CTRL+T.

We can go look at the condition we want to Set or Get and then switch to *Text View* to see the exact condition number to assign. If you hover your mouse over any condition, it will show an ID number which is also the condition number.

An example of a Text View would be:

SP.CONDITION#5=WEIGHT,SCALE1,GROSS,0,HIGHER,OFF,0.

With this example, we would assign our Set Compare to condition 5. Also a Set or Get Compare can come from a user variable after a User Get or from the Database from a specified field.

Examples

The following are examples of steps that can be easily setup to do a particular function. Then these can be merged to form a larger process for a final task to be performed.

Keys

There are four keys routinely used in a batch process:

Start

This action is performed from a step that has the "Enabled" flag set, to start the process flow. Typically its function is to activate the next step

Condition:
Digital Input Compare (this could also be a User Key Pressed)

Pause

Any step that should pause must have the "Pausable" flag checked, which allows that step to be paused and will run the list of Pausable actions for that step. Specific actions can be defined in this pause action list, such as Digital Output = Off)

Action:
Pause Process

Restore

This action will resume the process from where the pause process was performed.

Action:
Restore Process

Reset

Aborts the process and defaults to the original enabled steps. This action will only deactivate any steps that are set as "Pausable." Use the pause action list to perform any required actions, such as Digital Output = Off.

Action:
Restore Process

Weight Compare Gross/Net

A Free Running setpoint would have the Enabled flag set, and never be deactivated
A Latched setpoint would be Activated/Deactivated)

Condition:
Weight Compare (Sets the mode, value, trip state)
False Action:
Digout (Select and set the digital output on until the weight condition is satisfied)
True Action:
Digout (Select and set the digital output off when the weight condition is satisfied)
Activate Step/ Deactivate Step
Pause Action:
Digout (Select and set the digital output off if the process is paused or reset)

Weight Compare % Relative (requires 2 steps, first to calculate & set the value, then to batch the value. This can be done with a database or with an immediate value in the properties.)

Action: (Step 1)
User Multiply Value (Target Wt. x DB Value (fixed percentage) = Sum, this is a Real number)
User Set Compare (Used to put the value into the Weight Compare. Enter the condition number for the weight compare step.)

Condition: (Step 2)
Weight Compare

Accumulate Weight (to a data base)

Action:
DB Search (Sets the pointer to the Data Base)
DB Capture (Declare the condition to capture-Weight, and the field to put the value into)
User Add Value (Source is Data Base, Field specifies where to retrieve the weight to be added to the current captured value in the user variable)
DB Write (Field specifies where to store the accumulated sum)

Auto Jog (Make this 2 steps)

Condition:
Weight Compare
False Action:
Start Timer (duration will equal the jog time)
Digital Output (On)
True Action:
Deactivate step (move on in the process)

Condition:
Timer Compare
True Action:
Digital Output (Off)
Activate step (loop back to the Weight Compare step)

Concur (Make this 2 steps)

Condition:
Step State Compare (step to start on)
Action:
Digital Output (On)

Condition:
Step State Compare (step to stop on)
Action:
Digital Output (Off)

Counter (make this 2 steps)

Condition:
Always True
Action:
Increment Register

Condition:
Register Compare
Action:
Clear Register (true action)
Activate/Deactivate (false action)

DB Prompt (make this 2 steps)

Condition:
User Key Compare
Actions:
Get Database Record
Read DB (this will read the old value that is being used as a current target in a set compare later in this sequence)
DB Prompt (this will display a user string to tell the user what to do)
User Get (set for the data type required)

Condition:
User Entry
Actions:
User Set Compare (sets a new value for the target)
DB Write (writes the value to the database for the next change required)

Delay (make this 2 steps)

Condition:
Always True
Action:
Start Timer (duration)

Condition:
Timer Compare (checks timer for stopped or running)

Pause (This is just an indefinite delay, requiring operator intervention.)

Possible conditions: **Digital Input Compare, User Key Pressed, or User Entry**

PreAct Learn

True Action
DB Capture (Weight)
User Get Compare (Target)
User Subtract Value (Target–Weight =Sum)
User Multiply Value (Sum x Correction Value = PreAct)
DB Write (PreAct)
User Get Compare (Target)
User Subtract Value (Target – PreAct = New Target)
User Set Compare (New Target)

Print

Action:
Print (Select the Aux format, and using Rev III, configure the text and User Strings 1-99. These User Strings are called tags)

Print User Entry

Action:
DB Search (when printing data that has been stored, the pointer finds the record)
DB Read (reads the specific field in the data record)
User Set Print Text (sets the tag number for the User String 1-99)

Tare

Condition:
Motion Compare (set to check for standstill)
Action:
Tare
(To remove a Tare, use the Action **Clear Tare**)

Timer (Watchdog) (Make this two steps)

Condition:
Step State Compare
Action:
Start Timer

Condition:
Timer Compare (check to see if still running)
Step State Compare

WaitSS

Condition:
Motion Compare (Check for stand still, coming out of motion will satisfy this condition)

For the following functions, any database action must have the database pointer preset to the right database.

Add a new record to the Data Base

Action:
User Add

Condition:
User Entry
Action:
DB Write

Capture Weight to the Data Base

Action:
DB Search
Database Capture (data type, index, field)

Print Field from the Data Base

Action:
DB Search (index & field)
DB Read (field)
User Set Print Text (tag for user string, & data)
Print (Aux format)

Print Report from the Data Base (make this a loop of 2 steps)

(First Step)
Action:
DB Search (index & field)
DB Read (field 1)
User Set Print Text (tag for user string, & data)
DB Read (field 2)
User Set Print Text (tag for user string, & data)
(Add more fields as necessary)
Print (Aux format)

(Second Step)
Condition:
DB Record (is there a record?)
True Action:
Activate 1st step
False Action:
Deactivate Step (and move on in the process)

Prompt from Data Base (User selection)

Condition:
User Key Compare
Action:
User Get (Integer number for search)

Condition:
User Entry
Action:
DB Search (search for field 1)
DB Prompt (display field 2)

Prompt from Data Base records

Condition:
Always True
Action:
User Clear (clears the integer value)
User Add Value (Integer number to add)
DB Search (search for field 1)
DB Prompt (display field 2)

Search the Data Base for a variable

Action:
User Get

Condition:
User Entry
Action:
DB Search

Write to the Data Base

Action:
User Get (data type)
Condition:
User Entry (keypress enter or cancel)
Action:
DB Write (enter the field)