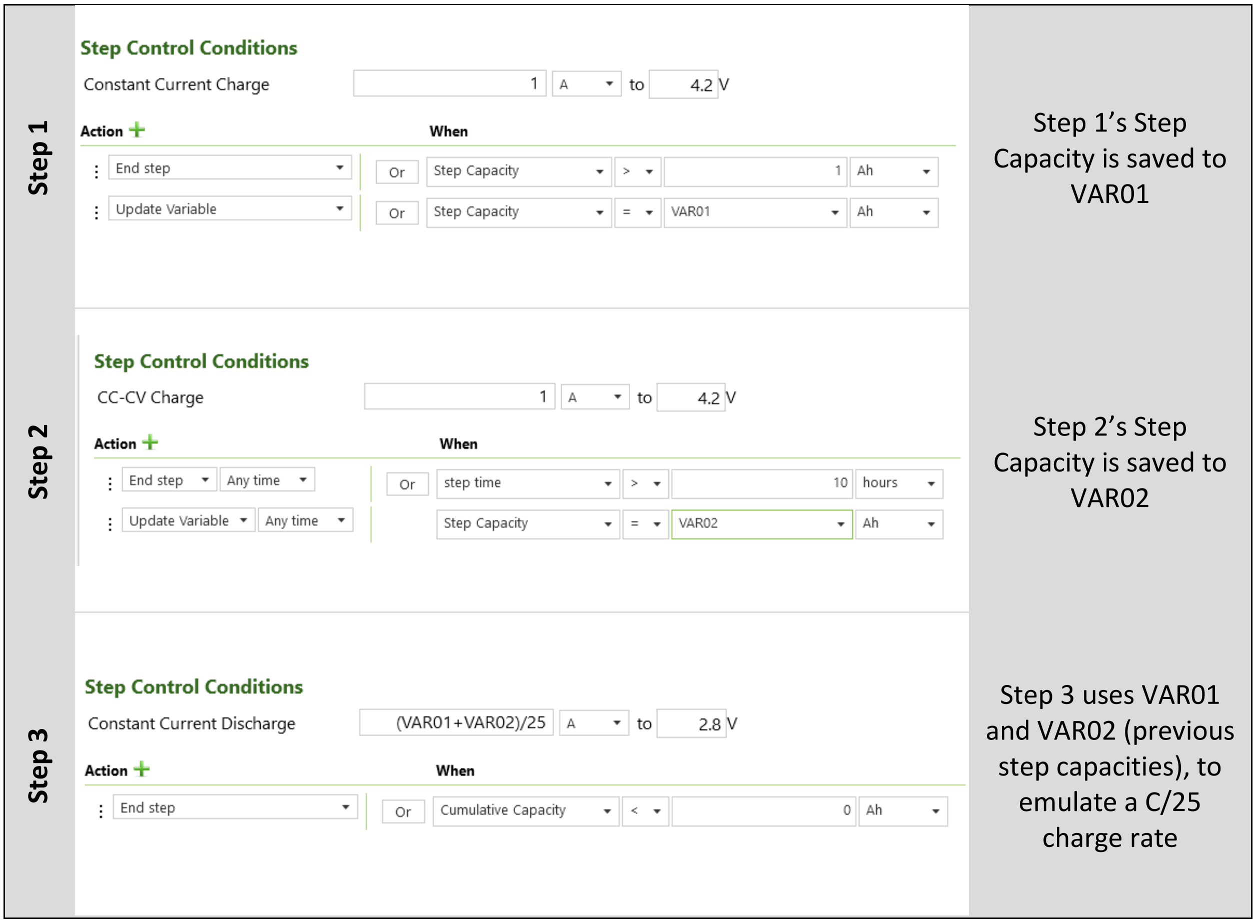
Getting started with UHPC Protocol’s new custom variables and functions

As of v2.6.0, UHPC Protocol supports creating and updating variables to trigger conditions within the ‘Step Control Conditions’ text boxes, as well as the ‘Current’ text box.

Example use

Users may wish to define Step Capacity as a variable rather than a static definition. In the example below, a 3-step protocol runs a CC charge, CC-CV charge and a CC discharge with variable step capacities:



Step 1

Step Control Conditions

Constant Current Charge: 1 A to 4.2 V

Action		When			
End step	Or	Step Capacity	>	1	Ah
Update Variable	Or	Step Capacity	=	VAR01	Ah

Step 1’s Step Capacity is saved to VAR01

Step 2

Step Control Conditions

CC-CV Charge: 1 A to 4.2 V

Action			When			
End step	Any time	Or	step time	>	10	hours
Update Variable	Any time		Step Capacity	=	VAR02	Ah

Step 2’s Step Capacity is saved to VAR02

Step 3

Step Control Conditions

Constant Current Discharge: (VAR01+VAR02)/25 A to 2.8 V

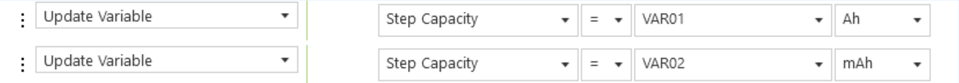
Action		When			
End step	Or	Cumulative Capacity	<	0	Ah

Step 3 uses VAR01 and VAR02 (previous step capacities), to emulate a C/25 charge rate

Throughout Step 1 and Step 2, Step Capacity is saved to **`VAR01`** and **`VAR02`** respectively. When Step 3 is executed, the Current equation uses custom variables to mimic “C/XX” where XX is 25, and C is the step capacity saved in step 1 and 2 as **`VAR01 + VAR02`**.

Note

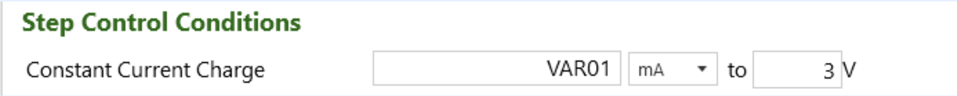
Keep an eye on your units. UHPC Protocol will convert all values to root units *before* saving. For example, in the example below for a 1Ah charge. **VAR01** would be saved as 1 and **VAR02** would be saved as 1000.



Update Variable | Step Capacity = VAR01 Ah

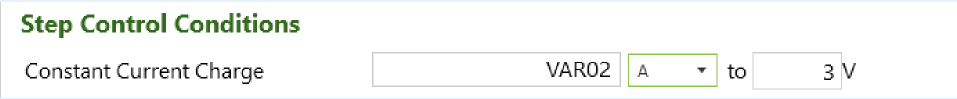
Update Variable | Step Capacity = VAR02 mAh

Which could result in a 1mA charge in the first image below, but an attempt to charge at 1000A in the second image below!



Step Control Conditions

Constant Current Charge VAR01 mA to 3 V

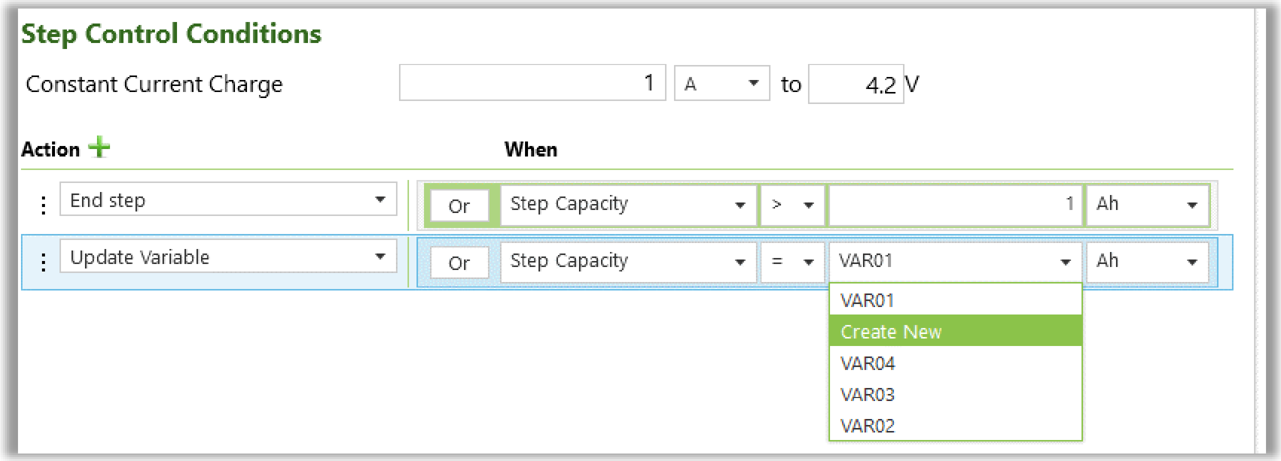


Step Control Conditions

Constant Current Charge VAR02 A to 3 V

New Custom Variables

To create new variables, or overwrite an existing variable, select “Create New”, or select an existing variable from the “Update Variable” dropdown menu:



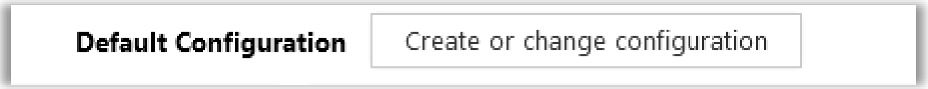
Step Control Conditions

Constant Current Charge 1 A to 4.2 V

Action	When
End step	Or Step Capacity > 1 Ah
Update Variable	Or Step Capacity = VAR01 Ah

When “Create New” is selected, a textbox appears. **All variables must contain at least one letter and no spaces**. They should not contain other variable names.

The list of newly created Custom Variables will be erased upon closing UHPC Protocol. **To persist Custom Variables, create them as part of the Default Configuration.**



Default Configuration

Available Custom Variables will also be loaded into the session when Protocols with existing Custom Variables are loaded.

> **Note**
> Custom Variables do not need to follow the `VAR01, VAR02, VARXX` naming conventions.
> Users are free to define more intuitive names – `STEP1CAP, STEP2CAP` etc.

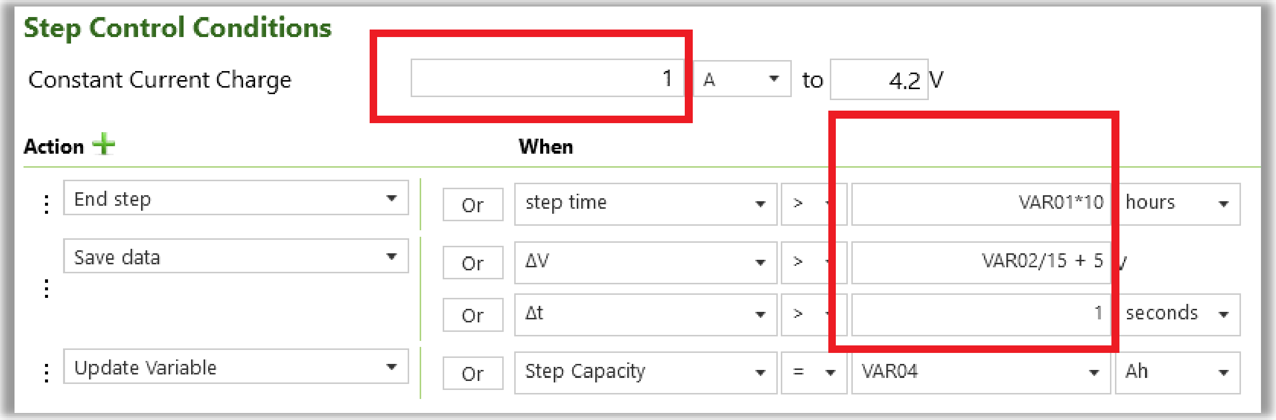
Custom Expressions

Additionally, right-side conditions and Current may be represented with a function containing variables to be executed in real time. **Voltage expressions are not valid at this time.**

Custom Expressions are intended to be used with Custom Variables but can also be used to execute simple algebra. The following are all valid expressions:

$2 + \frac{10}{15 + 5}$	`2+10/(15+5)`
$VAR_{01} + \frac{0.01 \ \times VAR_{02} \ \times 8}{12 + 7}$	`VAR01 +(0.01*VAR02*8)/(12 + 7)`
$\frac{VAR_{01}}{VAR_{03}} - 5$	`VAR01/VAR03 - 5`
$\frac{VAR_{01}}{VAR_{04} + 9}$	`VAR01/(VAR04 + 9)`

The expression below will be evaluated at the start of the step execution for Current, and throughout the step for End/Save Step Conditions:



Step Control Conditions

Constant Current Charge 1 A to 4.2 V

Action +		When			
End step	Or	step time	>	VAR01*10	hours
Save data	Or	ΔV	>	VAR02/15 + 5	V
	Or	Δt	>	1	seconds
Update Variable	Or	Step Capacity	=	VAR04	Ah

Note that in some cases, expressions cannot be evaluated. For example, some variables may grow to illegal values over the course of a test. Depending on where the expression is called, UHPC Protocol manages these exceptions.

In cases where expressions cannot be evaluated:

- In the case of current:
  - The test will halt in an error state
- In the case of a right-side condition:
  - The test may continue while logging errors of which condition could not be evaluated. **This may result in steps running past their end conditions**

💡 **Info**

Longer variable names or expressions require more width and may reduce readability in the program GUI. If you are struggling to read or write the full expression, you may copy/paste the expression with the help of Notepad. **If Protocol refuses to paste, your expression likely contains illegal characters.** Remember that only alphanumeric characters and basic math evaluators are allowed, `0-9A-Z,./*()+-^`.

Reducing complexity reduces chances of error. The logic in the code follows the BEDMAS order of operation, after replacing all variables with numbers.