Controller Parameters User Guide

Table of Contents

Introduction

The *Controller Parameters* utility was written to allow viewing or editing the contents of parameter files used by Common Controllers (FBPC, FB2, B3C-LC, …) or DC controllers (DC800, DC240, DC241, RC1, ...). The modified parameters can be exported to a text file then manually merged into the original parameter file.

Parameters can be validated using a set of parameter rules. The parameter rules can be used to validate any parameter against a constant or expression product from any other parameter or combination of parameters.

Overview

Illustration 1 shows the main view of the *Controller Parameters* utility. The *Controller Parameters* utility consists of three main sections: the rule editor, the list of parameter names and variables used by the rule editor, and the parameter editor.

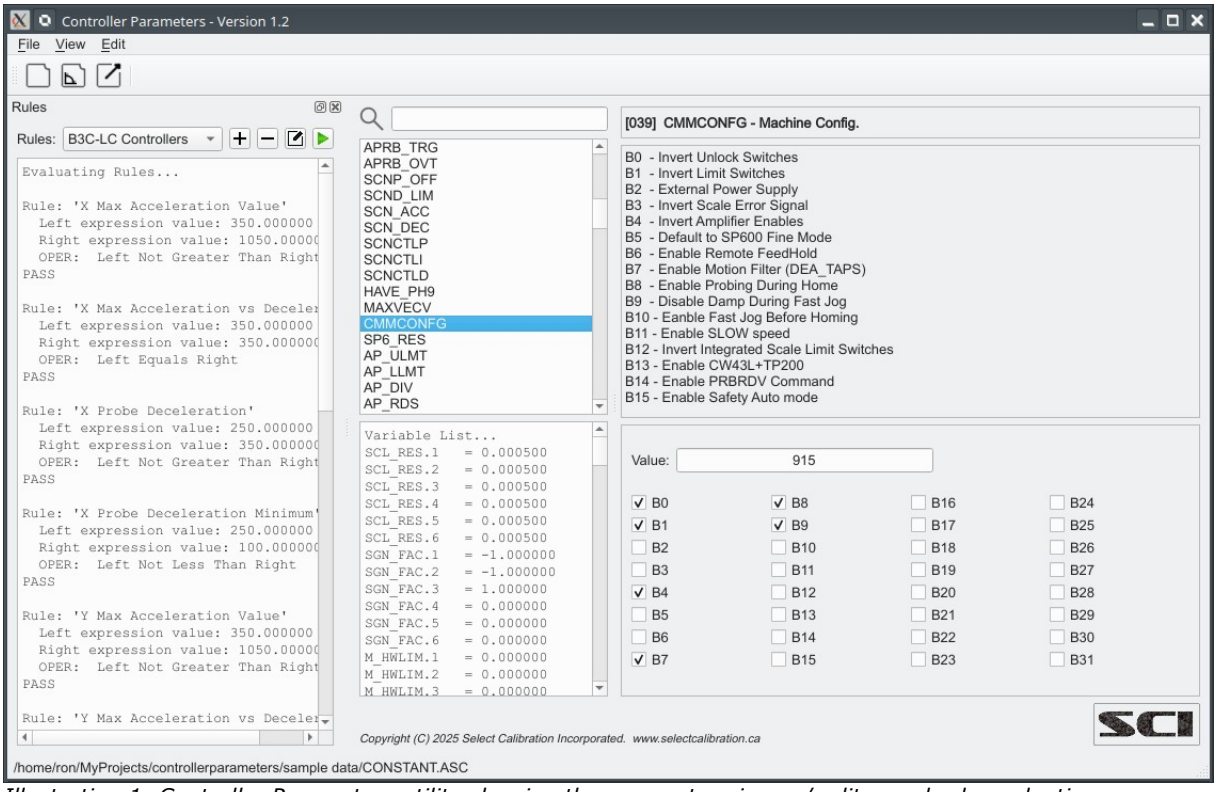

Illustration 1: Controller Parameters utility showing the parameter viewer / editor and rule evaluation.

Importing Parameters

Parameter data that is supported by the *Controller Parameters* utility can be loaded by one of two methods:

- Select *File – Open* from the main menu or from the toolbar shortcut.
- Drag and drop the parameter file onto the Controller Parameter utility.

The currently supported parameter files is CONSTANTS.ASC from Common Controllers (FBPC, FB2, B3C-LC, …) or any of the XML files from the DC controllers (DC800, DC240, DC241, RC1, …).

Parameter List

When a parameter file is loaded the parameter list section shows all the entries found in the file. Illustration 2 shows an example of the *servoAmpVal* parameter from a DC controllers X axis XML file.

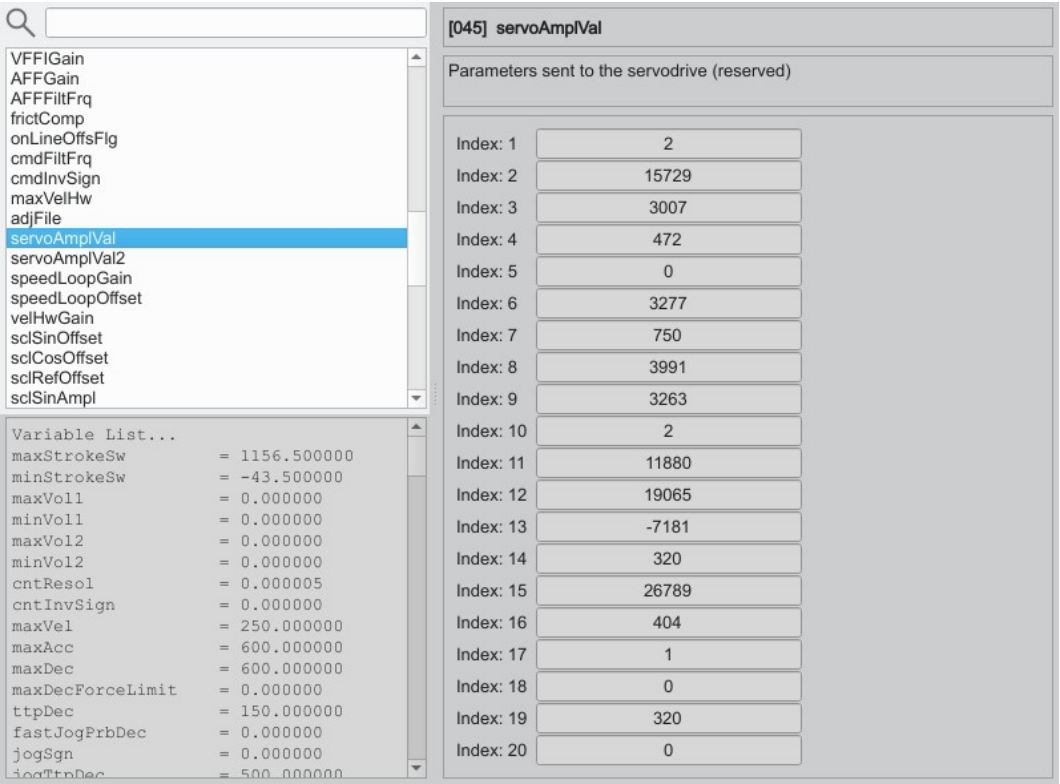

Illustration 2: Parameter List display section.

Parameter entries can be searched by entering all or part of a parameter name in the search bar. The search will hide all entries that do not contain the search text.

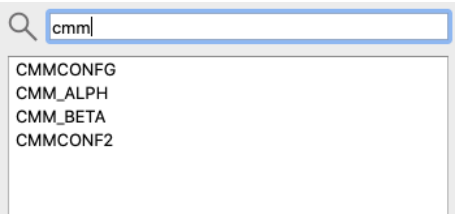

Illustration 3: Searching for all parameter names containing the word 'cmm'.

Selected entries will activate an editor for viewing or modification of the selected parameter value. Illustration 4 shows an example of editing the bitfield parameter CMMCONFG from a typical

Common Controller.

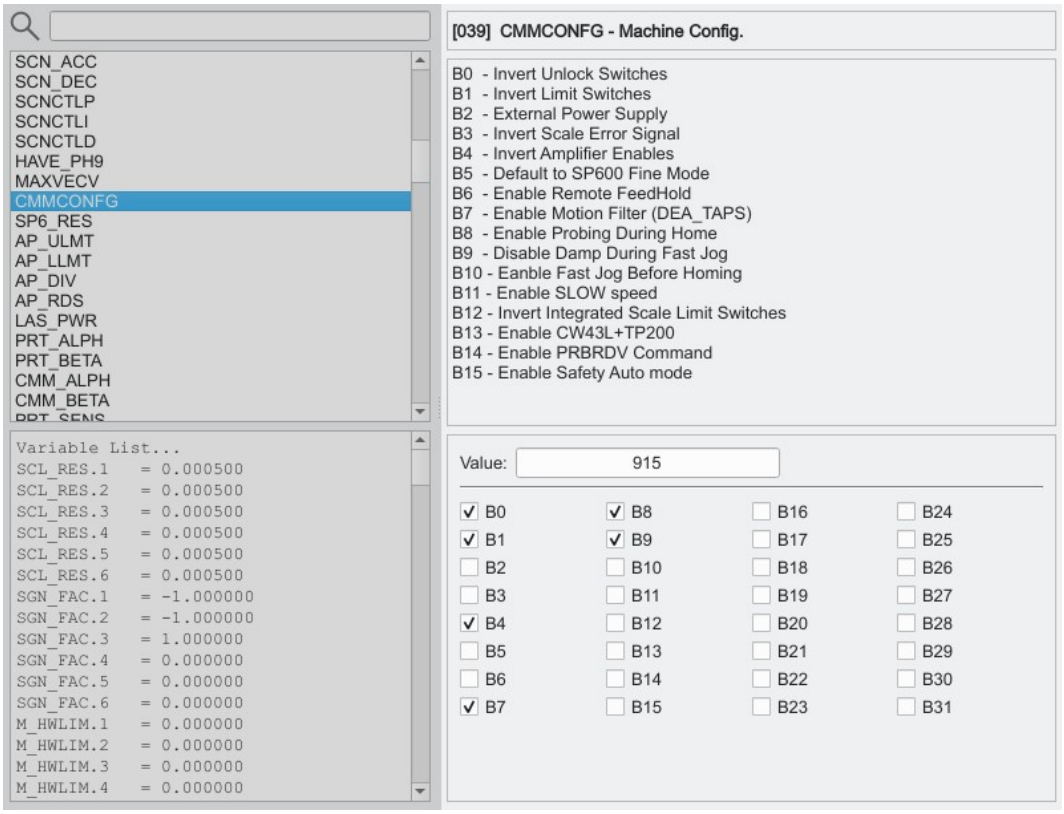

Illustration 4: Editor for parameter CMMCONFG.

The variable list below the parameter list shows all items that can be used in rule expressions. Illustration 5 shows an example of the variables created from the parameter VMAX.

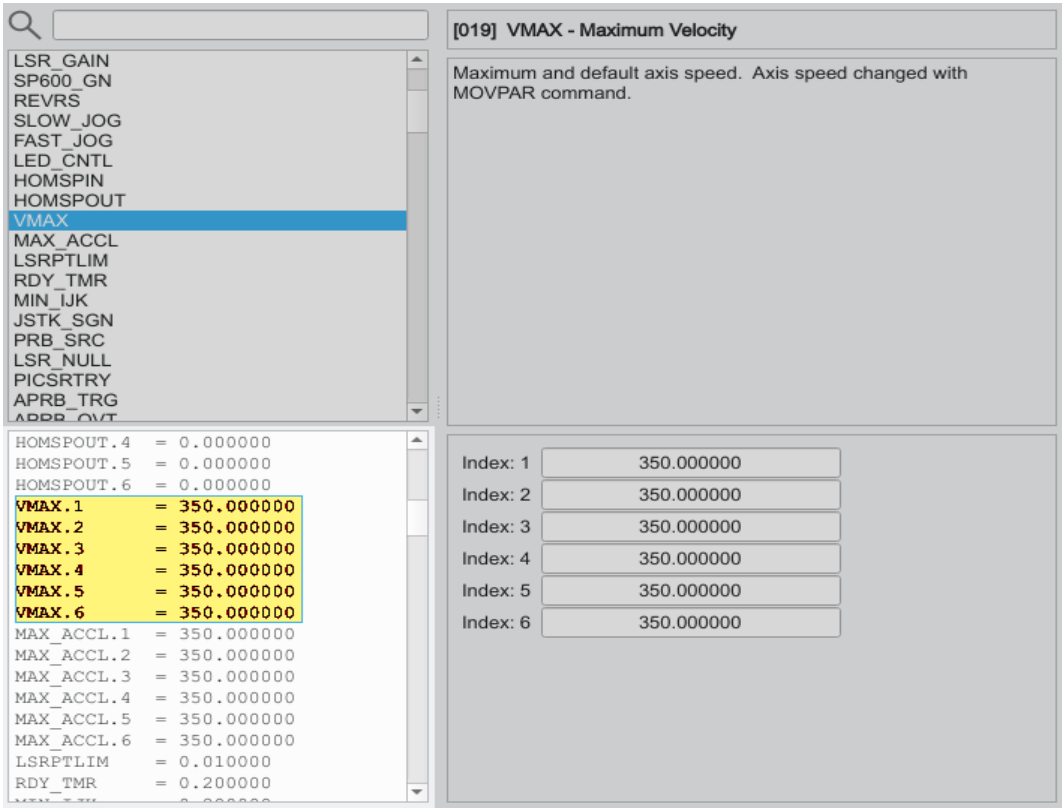

Illustration 5: Variables from the parameter VMAX that can be used for rule expressions.

Parameter types can be a single entry or an array of data. Single entry parameters will have a corresponding variable with the same name as the parameter while parameter arrays will have variables created with an extension of *<dot><index>* where index starts at 1 for the first entry. For example, the variable for the third index value of *VMAX* is *VMAX.3* representing the Z axis.

Exporting Parameters

Parameters that have been modified can be exported to a text file by the *Controller Parameters* utility. The formatting of the original input file is too quirky to re-create reliably so the changes are saved to a text file instead of over-writing the original input file. The changes must be manually transferred to the original parameter file and sent to the controller or manipulated in the controller directly.

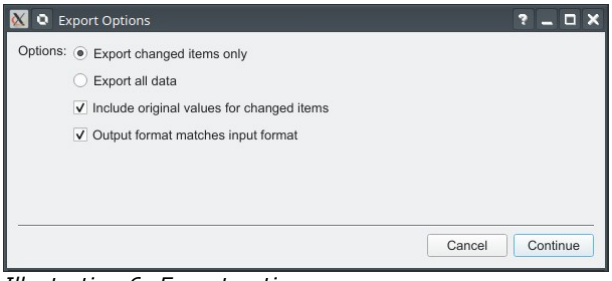

Illustration 6: Export options.

The format of the exported file can be similar to the format of the input file allowing for easier transfer to the source parameter file. The exported file can be configured to only contain changed parameters or all of the data.

Although the formatting of the input parameter files are considered quirky the formatting of the output file, with the option to match the input file checked, will be close to that of the input file.

Table 1: Export Options:

Option	*Description*
Export changed items only	The output file will only contain entries that were changed.
Export all data	The output file will contain all parameter entries.
Include original values for the changed items	This applies only to parameters that have been changed. The output with this option checked will show the original value and the new value.
Output format matches input format	The format of the output file will mimic that of the input file otherwise a generic format is used.

Exporting Parameters Example

The following is an example of changing the VMAX parameter from a Common Controller where the original value was 350 mm/sec for all axis but changed to 300 mm/sec for the X,Y, and Z axis and zero for the rotary table and wrist AB angles. When exported, the option to show the original value was checked along with the option to match the input format.

Portion of the input parameters file showing VMAX along with entries above and below it:

```
...
#18 Home Spd Outof Lim  : Double  : millimeters / Sec           HOMSPOUT
 3.00000000000000E+0000
 3.00000000000000E+0000
 3.00000000000000E+0000
 0.00000000000000E+0000
 0.00000000000000E+0000
 0.00000000000000E+0000
#19 Maximum Velocity    : Double  : millimeters / Sec           VMAX
 3.50000000000000E+0002
 3.50000000000000E+0002
 3.50000000000000E+0002
 3.50000000000000E+0002
 3.50000000000000E+0002
 3.50000000000000E+0002
#20 Maximum Accel       : Double  : millimeters / Sec /Sec      MAX_ACCL
 3.50000000000000E+0002
 3.50000000000000E+0002
 3.50000000000000E+0002
 3.50000000000000E+0002
 3.50000000000000E+0002
 3.50000000000000E+0002
```

```
...
```

Exported output file showing VMAX with the option to show the original value checked:

```
WARNINGS:
- Contents of this file does not included Reserved entries.
- File may contain additional data with alternate formatting.
- DO NOT SEND THIS FILE TO A CONTROLLER.
-------------------------------------------------------------
#19 Maximum Velocity    : Double                                VMAX
  Original:
+3.50000000000000E+0002
+3.50000000000000E+0002
+3.50000000000000E+0002
+3.50000000000000E+0002
+3.50000000000000E+0002
+3.50000000000000E+0002
  Updated:
+3.00000000000000E+0002
+3.00000000000000E+0002
+3.00000000000000E+0002
+0.00000000000000E+0000
+0.00000000000000E+0000
+0.00000000000000E+0000
```

Note the entry for VMAX in the exported file does not contain the text ': millimeters / Sec' following the declaration of the data type (Double). Showing the original and updated values (which is optional) allows for changes to be restored to original values if necessary.

Parameter Rules

The parameter data contains many entries and is therefore subject to many potential problems or conflicts. There has been numerous examples of parameter values that have caused subtle problems with machines observed over the years particularly with retrofit machines. Testing each parameter configuration manually is difficult and time consuming where automated testing, using the parameter rules, allows many of the mundane and easily overlooked entries to be validated.

Illustration 7 shows the parameter rule section which can be detached from the main program for convenience.

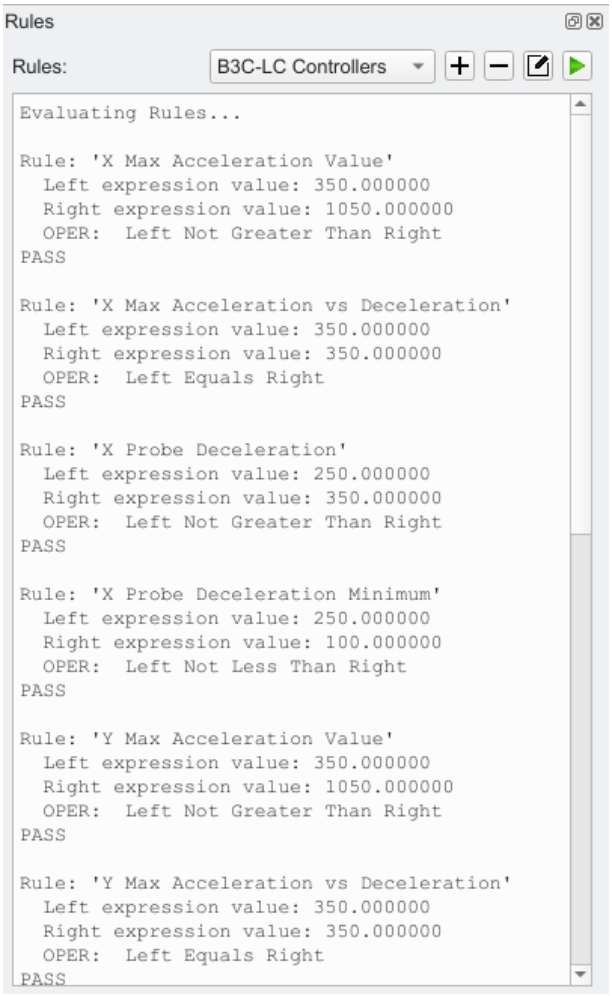

Illustration 7: Parameter rules.

Table 2: Rule Options:

Option	*Description*
Rules: B3C-LC Controllers	Selection list for rule groups. In this example the rule group is named based on the type of controller it is intended for.
+	Create a new rule group.
−	Remove the selected rule group. This also removes all rules associated to the rule group.
✎	Edit the rules for the selected rule group.
▶	Execute testing of the currently loaded parameter data using the rules in the selected rule group.

Rule Editor

Illustration 8 shows an example of the rule editor and all rules assigned to the selected rule group. Each individual rule entry is a specific test that is performed on the parameter data.

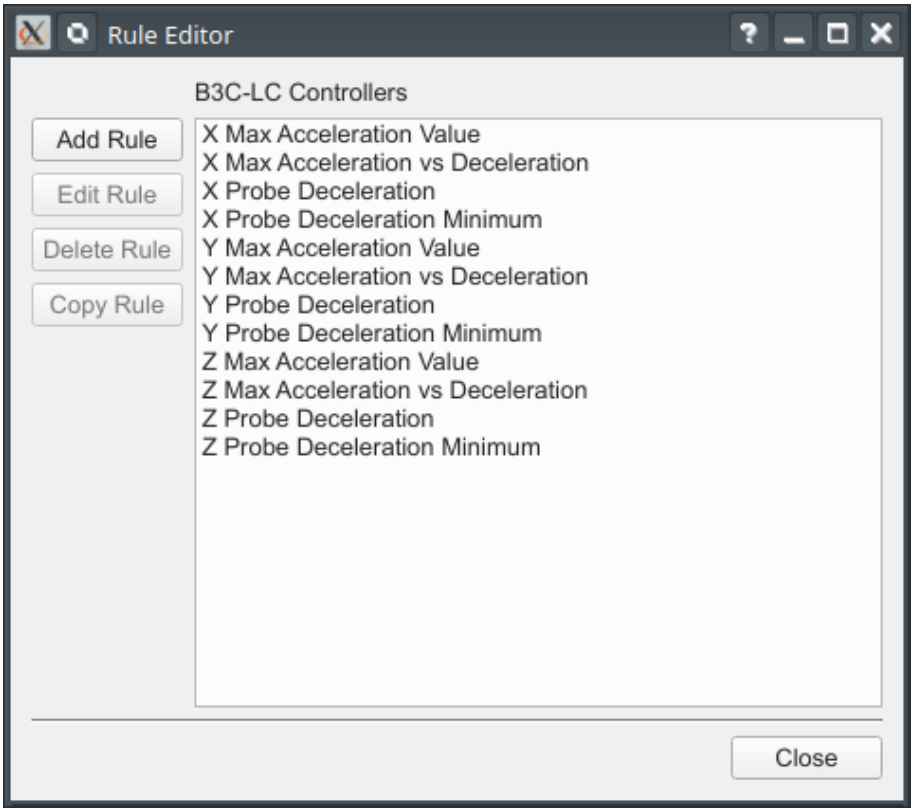

Illustration 8: Rule editor.

Table 3: Rule Editor Options:

Option	*Description*
Add Rule	Create a new rule entry.
Edit Rule	Modify the selected rule entry. Double clicking on a rule entry will automatically open the rule for editing.
Delete Rule	Delete the selected rule(s).
Copy Rule	Copy the selected rule(s) to another rule group. Rules that exist in the target group are replaced.

Rule Item Editor

Rule items are created or edited using the rule item editor. Illustration 9 shows an example of the rule item editor.

The data in the rule item editor is not cleared from the previous state when adding new rule entries. This is useful when creating several rules that are similar but only differ by minor

changes (axis entry for example) as all the previous entries are left intact.

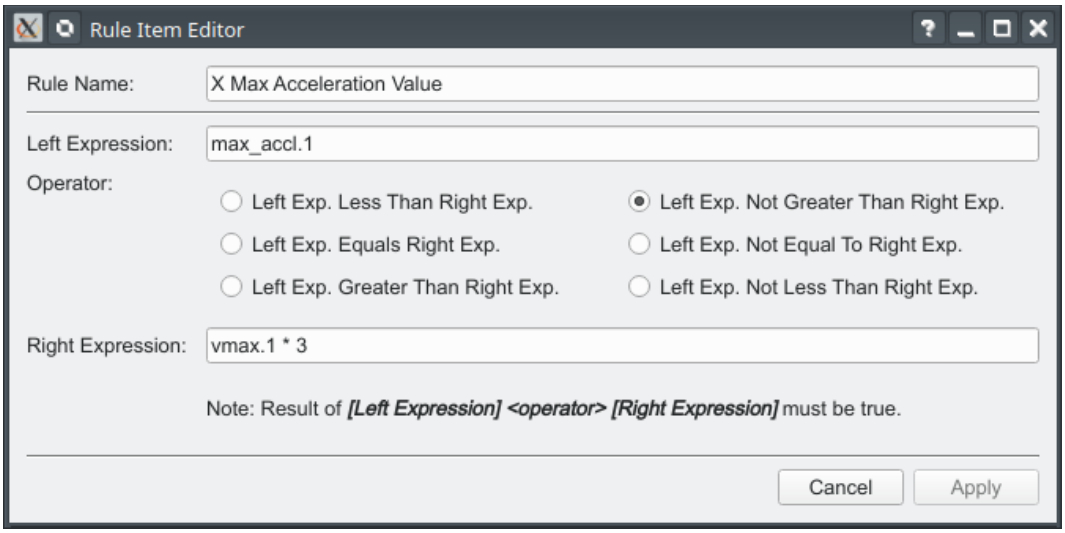

Illustration 9: Rule item editor

Table 4: Rule Item Editor Options:

Option	*Description*
Rule Name	Name of the rule entry. The name can be between 1 and 64 characters in length. The rule name must be unique for any particular rule group.
Left Expression	Left side of the expression. The expression can contain constants, variables, and include standard math operators (+-*/).
Operator	Operator for comparison between the left and right expression values.
Right Expression	Right side of the expression. The expression can contain constants, variables, and include standard math operators (+-*/).

Each rule item is evaluated as a logical test that produces a true or false result. In order for the rule item to pass evaluation the logical result must be true.

```
Result = [Left_Expression] <Operator> [Right_Expression]
```

Table 5: Expression Operators:

Operator	*Description*
Left Exp. Less Than Right Exp.	The value of the left expression must be less than the right expression.
Left Exp. Equals Right Exp.	The left and right expressions must be the same.
Left Exp. Greater Than Right Exp.	The value of the left expression must be greater than the right expression.
Left Exp. Not Greater Than Right Exp.	The value of the left expression must not be greater

Operator	*Description*
	than the right expression.
Left Exp. Not Equal To Right Exp.	The left and right expressions must not be the same.
Left Exp. Not Less Than Right Exp.	The value of the left expression must not be less than the right expression.

The operators with *NOT* are the logical reverse of the corresponding operators without *NOT*. The following scenarios shows the logic behind the different kinds of operators:

```
Left Input Expression : V1
Right Input Expression: V2

Goal    :  V1 must not be greater than V2.  V1 can equal V2.
Operator:  Left Exp. Not Greater Than Right Exp.

Goal    :  V1 must be greater than V2.  V1 cannot equal V2.
Operator:  Left Exp. Greater Than Right Exp.

Goal    :  V1 must be the same as V2.
Operator:  Left Exp. Equals Right Exp.

Goal    :  V1 cannot be the same as V2.
Operator:  Left Exp. Not Equal To Right Exp.
```

Rule Validation

The rules testing is performed by pressing the *Execute* button. The result from each rule item tested is displayed as text in the area below the rules toolbar.

Example from running a set of rule tests:

```
Evaluating Rules...

Rule: 'Max Acceleration Value'
Left expression value: 600.000000
Right expression value: 750.000000
OPER: Left Not Greater Than Right
PASS

Rule: 'Max Acceleration vs Deceleration'
Left expression value: 600.000000
Right expression value: 600.000000
OPER: Left Equals Right
PASS
...
```

Example of a failed rule:

```
Rule: 'Probe Deceleration Minimum'
Left expression value: 50.000000
Right expression value: 100.000000
OPER: Left Not Less Than Right
* FAIL *
```

There are five lines generated from each rule item that is executed. The five lines contain the information described in the Rule Evaluation Summary table.

Table 6: Rule Evaluation Summary:

Option	*Description*
Rule:	The name of the rule.
Left expression value:	The value of the left side expression. Expressions are evaluated to a value and displayed following the colon.
Right expression value:	The value of the right side expression. Expressions are evaluated to a value and displayed following the colon.
OPER:	Description of the operator used for comparison of the left and right expressions.
PASS / FAIL	Indicator if the rule passed (result was true) or failed (result was false).

Revision History

Date	*Version*	*Changes*
Aug 9, 2019	1.0	New Program
Feb 12, 2022	1.1	Update of depreciated functions. Allow Rule dock to be hidden. Added View menu for rule dock.
Aug 21, 2025	1.2	[bug fix] – Block direct editing of value for bit-field entries. Updated icons and signal/slot methods.