

CODE BUILDER

Tags and Expressions are a powerful part of CPQ. To help reduce errors and simplify the process, the **Code Builder** is available to create expressions. To access the Code Builder, click the **Check**, **Assist**, or “**v**” button next to the field where the expression will go. This button is available any time an expression can be used.



Clicking a Code Builder button will bring up a screen similar to Figure 23-1. Code Builder changes according to the settings the expression is being created for. For example, if an expression is being added for a **Rule** (Chapter 34) or **Trigger** (Chapter 33), then an additional box (Figure 23-2) will be displayed with the Code Builder. The basic principles for using the Code Builder are the same despite what setting the expression is being created for.

Code Builder	
Select an attribute: CD or DVD Drive #1	Select sequence: 1-20 A-Z
Insert a parseable macro: attribute quantity attribute price attribute value	Insert a macro for sequence: PSEQ
Select attribute values: 64x DVD-ROM Drive 96x/64x/96x CD-RW Drive 48x DVD+RW/+R Drive	Insert a operator: IF ENDIF AND
<input type="button" value="Insert"/>	<input type="button" value="Insert sequence"/>
<input type="button" value="Insert operator"/>	
<div style="border: 1px solid gray; height: 80px; width: 100%;"></div>	
<input type="button" value="Update"/>	<input type="button" value="Check Syntax"/>

Figure 23-1 – Code Builder Screen

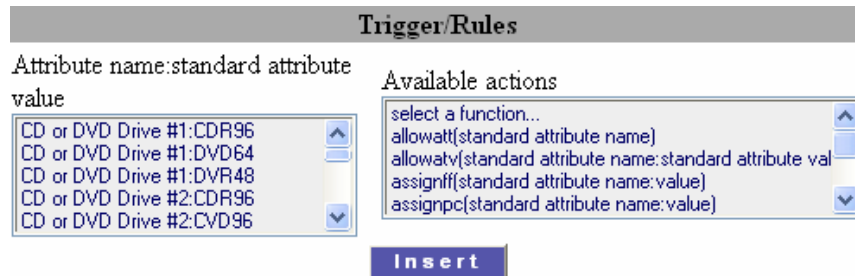


Figure 23-2 – Additional Trigger/Rules Section

The Code Builder window is divided into two sections. The main box at the bottom is where the expression is formed. Expressions can be typed in manually if necessary and in certain instances, phrases can be entered into the box to be displayed to the user.

The **Check Syntax** button checks the expression to ensure that the Tags have been joined correctly and that all punctuation is correct. Any errors will be displayed at the bottom of the window. It is a good idea to check an expression before inserting it to ensure it functions correctly and that it does not produce an error for the user. When finished, the **Update** button inserts the created expression into the field.

The top section of the Code Builder windows contains boxes to build the expression. In most cases, to build an expression, select the attribute, value, sequence, etc that an action will be performed on then select the action.

An **Insert a Operator** box is available in every Code Builder window to create conditional statements and Boolean logic. Click the Operator in the box and then click **Insert** to insert the Operator into the expression. In most cases the Operator is added first then the rest of the expression is included inside parenthesis next to the Operator. These parenthesis are not added automatically, so be sure to add them. Figure 23-3 shows an example of an expression using an Operator. This example states that the Hard Dive Attribute Value HD200 AND the Memory Attribute Value 2GB need to be selected in order for this expression to be true.

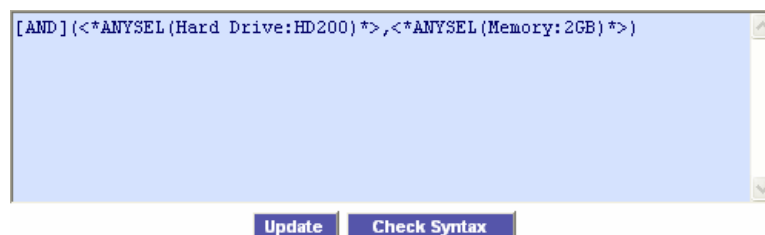


Figure 23-3 – Operator Example

Figure 23-1 is a common Code Builder window. To create an expression in this window, select an attribute from the **Select an Attribute** box, then select a Parseable Macro from the **Insert a Parseable Macro** box (the action). Attribute Value(s) can also be selected for the expression if necessary. Click the **Insert** button to add the new Tag to the expression. Sequences can also be used in expressions by selecting the sequence from

the **Select a Sequence** box then selecting the Macro for it in the **Insert a Macro for Sequence** box (see Chapter 29). Click the **Insert** button to add the new Tag to the expression.

The Attributes displayed in the box will only be those selected for that product or category, depending on what you are creating the expression for. This is the case for every Code Builder window. If a certain object or attribute is not available, make sure it is checked, selected, or even created for the settings the expression is being created for. This should also be considered when selecting Attribute Values.

The actions, Macros in this case, will also be different for each Code Builder window. For information on what an action does, check Appendix A for a list and description of all Tags. There is also information there on the proper way to join Tags and create expressions. While the Code Builder aids in the process of creating expressions, Tags need to be joined and built correctly in order for the expression to function accurately.