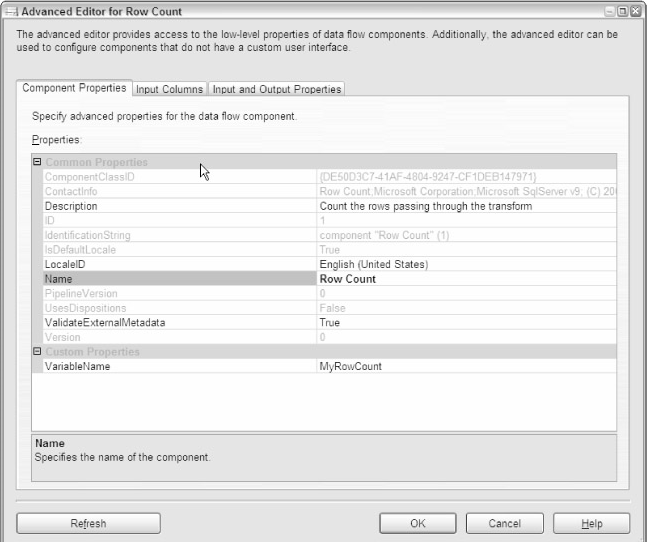
**Row Count**

The Row Count transformation task provides the ability to count rows in a stream that is directed to its input source. This transformation must place that count into a variable. This transformation is useful for tasks that require knowing "How many?" It is especially valuable since you don't physically have to commit stream data to a physical table to retrieve the count. If you need to know how many rows are split during the Conditional Split transformation, direct the output of each side of the split to a separate Row Count transformation. Each Row Count transformation is designed for an input stream and will output a row count into a Long or compatible data type. You can then use this variable to log information into storage, to build e-mail messages, or to conditionally run steps in your packages.

This transformation uses the Advanced Editor. As you recall, you used this editor in the Import Columns section of this chapter. Configuring this transformation is much easier. All you really need to provide in terms of configuration is the name of the variable to store the count of the input stream.

You will now simulate a situation where you have a conditional step in a package that should run only if an input stream row count is evaluated to have a row count greater than zero. You could use this type of logic to implement conditional execution of any task, but for simplicity, you'll conditionally execute a SQL Statement.

1. Create an SSIS package named Row Count Example. Add a Data Flow task to the Control Flow design surface.
2. Add a Variable named MyRowCount. Ensure that the variable is package scoped and of type Int32.
3. Add an OLE DB Data Source to the Data Flow design surface. Configure the source to point to your AdventureWorks database and the table [ErrorLog].
4. Add a Row Count transformation task to the Data Flow design surface. Open the Advanced Editor. Provide the variable name MyRowCount as the VariableName property.
5. Return to the Control Flow tab and add an Execute SQL task. Configure the connection to the same OLE DB connection and set the SQLStatement to SELECT 1. This task is not really going to perform any action. It will be used to show the conditional ability to perform steps based on the value returned by the Row Count transformation.
6. Connect the Data Flow task to the Execute SQL task.
7. Right-click the arrow connecting the Data Flow and Execute SQL tasks. Select the Edit menu. In the Precedence Constraint Editor, change the Evaluation Operation to Expression. Set the Expression to @MyRowCount>0. 

When you run the package, you'll see that the Execute SQL task is not executed. If you are curious, insert a row into the [ErrorLog] table and rerun the package. You'll see that the Execute SQL task will turn green, indicating that it was executed.