About writing custom code for RBA

Experienced programmers can write code to customize the Rule Parameters Editor.
If you know how to write in a programming language that the Microsoft .NET Framework supports, such as Microsoft Visual Basic, C#, or C++, you can use it to assign values to parameters.
The Visual Basic language is supported natively in the RBA software. You can also use other languages, as long as you create the code libraries externally and then import the *managed* `.dll` file.
Note: If you use unmanaged `.dll` files, the .NET Framework does not provide feedback to the RBA system about exceptions. You must troubleshoot any code problem yourself.

Writing code is useful when you want do things that are not possible in Rule Parameters Editor. The basic constructs for storing code in an RBA rule are:

- **Parameter assignments**: Most RBA actions are parameter assignments. Custom coding allows you to specify how the action parameters obtain their values. The code is stored in a subroutine called `ConfigureAction()`.
- **Branches**: The Branch type of action uses simple Boolean expressions to get a true/false result. You can code your own condition logic. The code is stored in a subroutine called `ConfigureAction()`.
- **Filters**: The Filter type of action builds a list of objects that match a certain set of conditions. You can code your own selection criteria. The code is stored in a subroutine called `ConfigureAction()`.
- **User Defined**: The User Defined type of action is a simple container for any code. You control what code is executed and which resulting events are raised. The code is stored in a subroutine called `Run()`.

The Rule Engine only executes code in the `ConfigureAction()` and `Run()` subroutines. This code is wrapped invisibly in the supporting code for the Rule Engine. You cannot add your own import statements, and you must refer explicitly to object types by namespace.
You can create and load your own code libraries in your RBA code. This is useful for re-using code in more than one place and allows you to work in programming languages other than Visual Basic.

Writing code

Parameter Code Editor includes tools to help you write code. If you are more comfortable using another tool, such as Microsoft Visual Studio or a text editor, you can write code in it and then copy and paste the code into Parameter Code Editor.

Removing code

If you later decide to stop using code for a particular event-action pair, right-click the line between the event and action, and select **Reset to Default**.