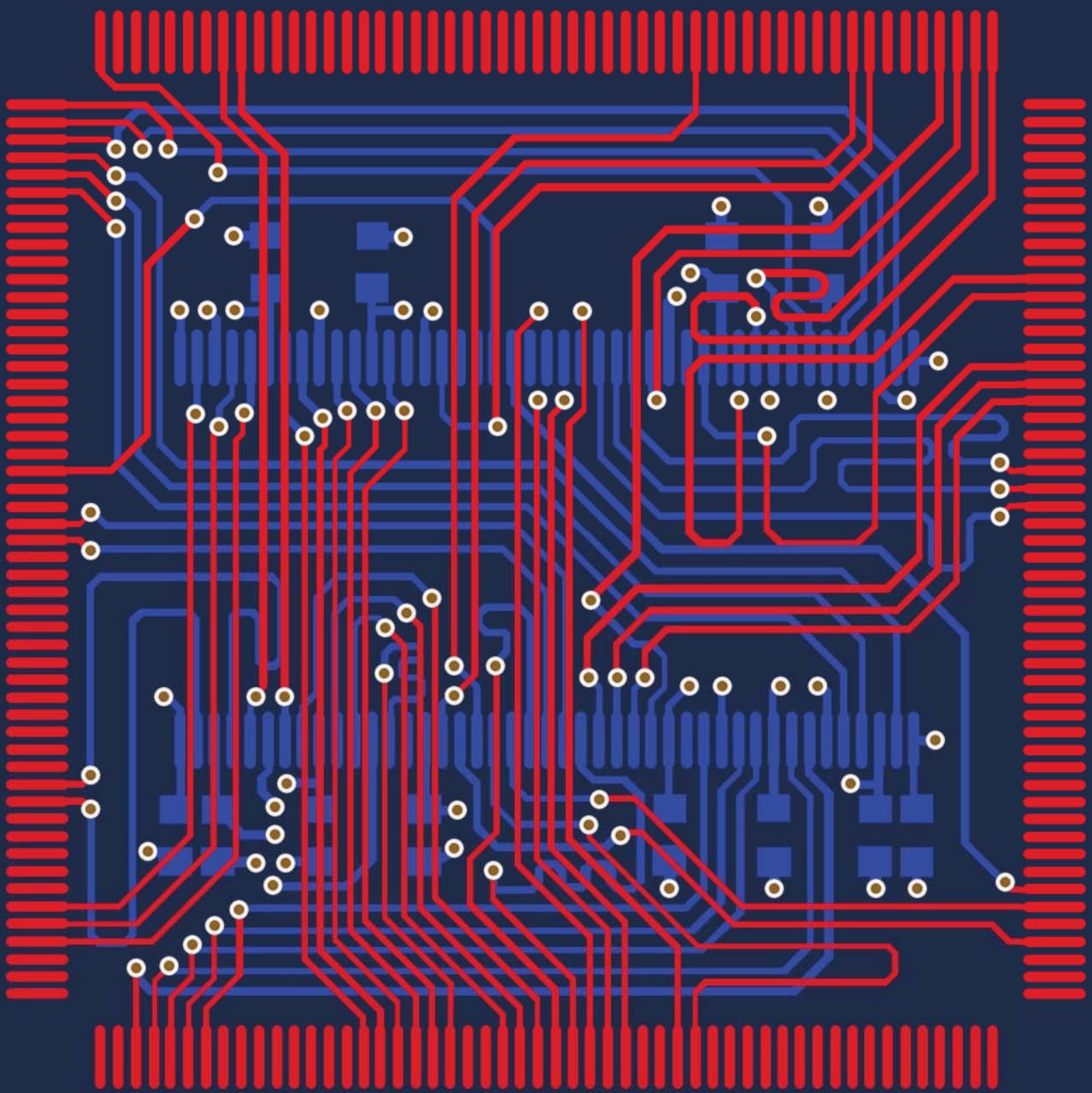


Altium[®]

Queries – How to Utilize Rules to their Full Potential



Jesse Lai

Applications Engineer

QUERIES – HOW TO UTILIZE RULES TO THEIR FULL POTENTIAL

Queries allow you to find, isolate, and operate on objects in a PCB design. Altium Designer® provides multiple methods of utilizing queries prioritized by particular rules. This will help you use the Query Builder and Query Finder, which are two essential tools for implementing the definitions of queries in different areas of the PCB documents.

INTRODUCTION

A query is a string composed of various, yet specific, keywords and syntax, which will isolate and return the inputted objects. In Altium Designer, there are different functional areas in which queries are used and proven to be very useful towards project item classification, isolation, and organization. Together, the queries become a computer language, in which the software uses Delphi to translate. Delphi is the preferred language in Altium Designer for its advantageously fast compiling, such as compiling millions of lines of code within a few seconds.

“AND” AND “OR” OPERATORS

When isolating different objects using “AND” and “OR” operators, they are used to filter objects that are focused on more than one condition. For example, in a two condition rule, the “AND” operator applies to the rule in Altium Designer if **both** conditions are applied, while the “OR” operator applies to **either** condition, **but not both**. This is of great importance and it will help differentiate between the two inputs to behave either independently or in the union.

HELP YOURSELF WITH RULES

Altium Designer currently provides two types of design rules: unary and binary. Unary rules are different to binary as they contain one rule scope and they’re applied to one object or each individual object classified in a specific type. Binary rules contain two rule scopes and are applied to both “two objects or between any object in one set to any object in a second set”^[1]. The rules are broken down into general classifications of objects: All objects, Net, Net Class, Layer, Net and Layer, and Query. Besides from the query, the other classes can only function based on their respective types. However, what if you want to combine multiple rule entities into one command line? This is where a custom query comes into play.

Let’s say you want to isolate a particular ground via to have a minimum hole size of 0.4 mm and a maximum of 0.6 mm. Currently in the rule generator, the “Via” class available in the PCB rules and constraints editor can only be applied to all vias in the project. The editor can be opened under **Design >> Rules** while the PCB document is focused. However, the rule to be applied to specific vias is achieved with an isolated rule through a custom query under the **Manufacturing Hole Size** rule section. In the following example, a user can input a custom query, such as “**IsVia**”, then include the “AND” operator, followed by GND for the ground net as the first ‘**Where the Object Matches**’ entry. The second ‘**Where the Object Matches**’ entry is then dependent on the rule type, which in this case is the minimum and maximum hole size rule. As an end result shown Figures 1 and 2, when running either online or offline Design Rule Checker (DRC), the program will focus on that rule and locate any potential vias that are violating it. The benefit of using queries is that the available attributes, such as designators, nets, layers, sizing, and shape can be utilized to isolate different PCB object types very easily, preventing many users from “pulling their hair out”.

QUERIES – HOW TO UTILIZE RULES TO THEIR FULL POTENTIAL

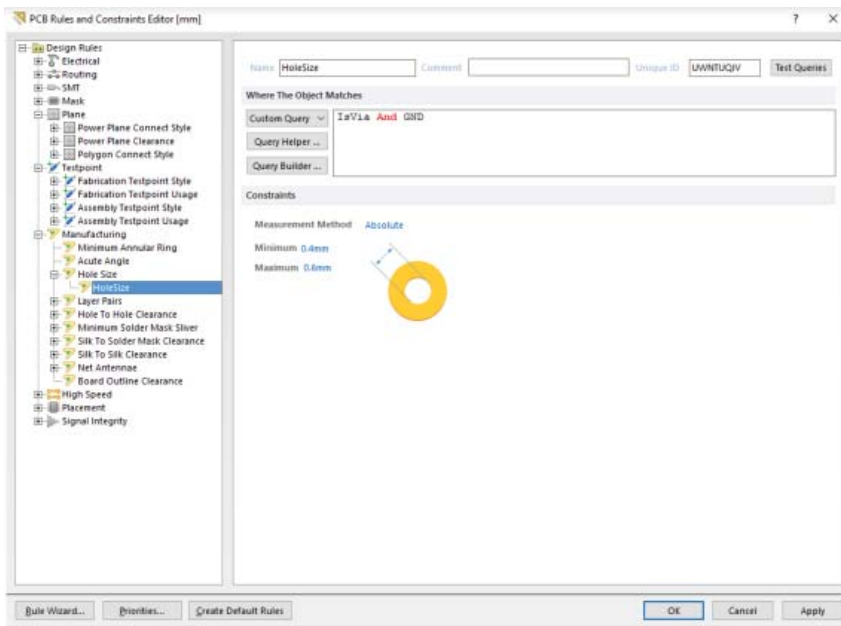


Figure 1: Rules set the maximum and minimum via hole diameter.



Figure 2: A resulting via pad.

CUSTOM QUERY HELPER AND BUILDER

The Custom Query Helper and the Custom Query Builder are located under the **Place Rules and Constraints Editor** (**Design >> Rules** or through shortcuts **D >> R**) in the software.

Query Helper

The query helper provides a list on the right section of the window that lists different functions with respective descriptions available within the software. The available functions to create queries are divided into three sections (PCB Functions, PCB Object Lists, and System Functions) and their subcategories. It can be used to train users in learning how to write customized queries and apply them directly to their projects.

Figure 3 shows how the query helper appears in Altium Designer. It is organized into several different sections: 1) the resulting output can be viewed; 2) the categories of different query inputs are available; 3.) the respective subcategories can be branched out with their respective descriptions; 4) the available operators showed beneath the query viewer, and 5) the masking feature. The mask feature is treated similarly to a filtered search option; the search results become isolated based on the mask input, allowing users to locate a particular solution much easier.

The most valuable feature of this tool is the ability to verify whether the query is valid or invalid based on the project, which is implemented in the button **Check Syntax** on the bottom left. This behavior is similar to inputting a custom query directly in the rules window, and when invalid the design rule type will be highlighted in red and a popup notification will appear. It is useful for the user to locate any area of the query that is improperly expressed.

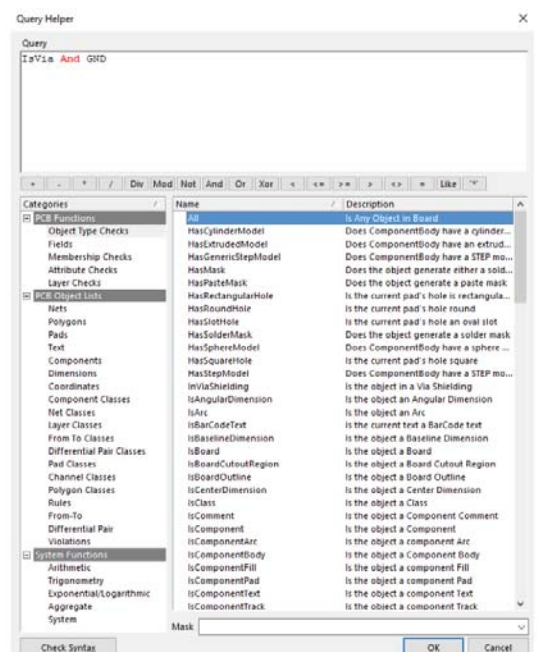


Figure 3: The Query Helper.

QUERIES – HOW TO UTILIZE RULES TO THEIR FULL POTENTIAL

Query Builder

The query finder is treated as a more simplified, sophisticated tool in comparison with the query helper. The available condition types and operators are sensitive to wherever the dialog was launched from. If the finder was accessed from a specific component, the condition types will only be relevant to the particular component object. This behavior is similar when the builder is accessed directly from the rules, in which the result is the conditions list, where multiple conditions from their respective types can be applied, is dependent only on condition types available for the specific rule (Figure 4).

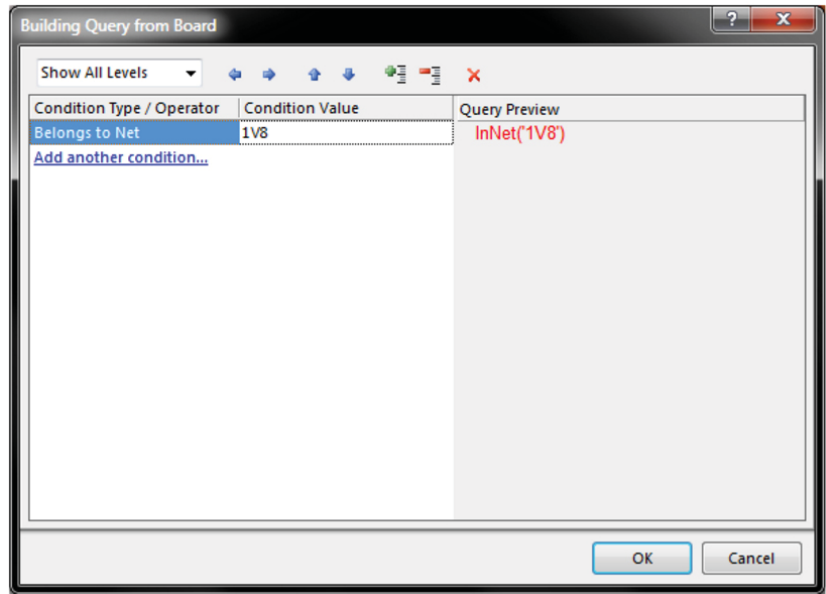


Figure 4: The Query Builder

Test Query button within the DRC rules

In Altium Designer in its respective subversions, a feature labeled **Test Queries** is implemented to make sure that the rules with queries isolating specific objects are actually applied to the areas of the design when the DRC is run. This will be very convenient for project users as it reduces the amount of time it takes to verify that the specific objects rule is actually applied or not.

In the test query result, it will indicate the specific rule being applied with the number of objects being applied as a result. The “Expression and rule priority” will indicate some areas of the rule not being applied due to another rule that is not necessary being conflicted, but having a higher priority. This feature can be achieved under **Design >> Rules** of the PCB editor where the button will be located on the top right section of each individual rule. This feature can be applied to unary and binary rules.

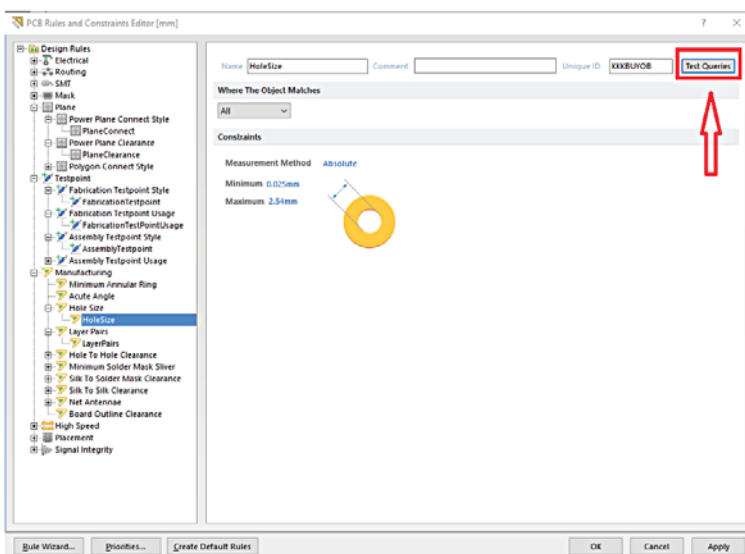


Figure 5: New in Altium Designer 16 is the Test Queries function.

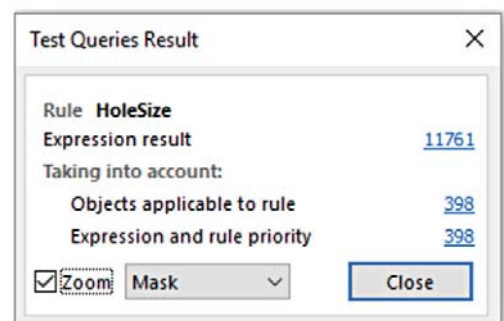


Figure 6: Results of the Test Query

QUERIES – HOW TO UTILIZE RULES TO THEIR FULL POTENTIAL

CONCLUSION

In general, users that only resort to pre-defined rule classes to generate rules for their projects will not realize the query capabilities of helping them save time, money, and limit errors when released to the manufacturers for assembly. Queries, in general, provide more room to utilize multiple object classes and their respective subclasses. This will allow a specific object in the design to be applied to a specific rule without having all objects be applied to that rule. So the rule of thumb is this when creating rules, use more queries and when necessary!

REFERENCES AND LINKS

[1] Loughhead, Phil. "Unary and Binary Rule Types." Unary and Binary Rule Types. Altium, Inc., 13 Aug. 2015. Web. 28 June 2016).

<http://techdocs.altium.com/display/ADOH/Introduction+to+the+Query+Language>

<http://techdocs.altium.com/display/ADOH/An+Insiders+Guide+to+the+Query+Language>

http://www.w3schools.com/sql/sql_and_or.asp

[http://techdocs.altium.com/display/ADRR/PCB_Dlg-QueryBuilderFromBoard\(\(Query+Builder\)\)_AD](http://techdocs.altium.com/display/ADRR/PCB_Dlg-QueryBuilderFromBoard((Query+Builder))_AD)

<http://techdocs.altium.com/display/ADOH/Unary+and+Binary+Rule+Types>

[https://en.wikipedia.org/wiki/Delphi_\(programming_language\)](https://en.wikipedia.org/wiki/Delphi_(programming_language))