This reference details the following in Altium Designer:

- Query Helper
- Library Functions
- Schematic Functions
- PCB Functions
- System Functions.

What is a Query?

Underlying Altium Designer is a powerful data filtering and editing system that lets you instruct the software to return you a specified set of objects. This instruction is entered in the form of a Query.

A query is a string you enter using specific keywords and syntax, which will return the targeted objects. There are many keywords available, allowing you to target objects by their type, their properties, or both.

For help on a specific query keyword, press F1 when the cursor is in the keyword. Use the links below for more information on the query system, operators, and tips on writing queries.

Use the Query Helper dialog for a list of keywords. To easily re-use popular queries add them to your Favorites (copy them from the History list). Favorites can be accessed from the Favorites dialog, or by pressing the Y shortcut in the schematic or PCB editors.

Links

- Introduction to the Query Language
- An Insider's Guide to the Query Language
Query Operators

The following is a list of the operator buttons available from the Query Helper dialog:

**Arithmetic Operators**

**Addition Operator, +**

Example:
NetPinCount + NetViaCount

**Subtraction Operator, -**

Example:
ArcStopAngle − ArcStartAngle

**Multiplication operator, ***

Example:
PadXSize_BottomLayer * PadYSize_BottomLayer

**Division Operator, /**

Example:
HoleDiameter / ViaDiameter

**Integral Division operator, Div**

Example:
Color Div 65536
This calculates Color divided by 65536 and the fractional part of the result is discarded.

**Mod**

This is used to obtain the remainder from an integral division operation, equivalent to determining a mathematical modulus.

Example:
Color Mod 256
This calculates the remainder when Color is divided by 256 without determining the fractional part of the result.

**Logic Operators**

**Logical AND operator, And**

**Logical AND operator, &&**

This is also used to implement Logical AND, but has a lower order of precedence.

Examples:
IsPad And OnMultiLayer
IsPad && OnMultiLayer

**Logical Or operator, Or**

|| This is also used to implement Logical OR, but has a lower order of precedence.

Example:
IsPad Or IsVia
IsPad || IsVia

**EXCLUSIVE OR operator, Xor**

This is used to implement Logical EXCLUSIVE OR.

Example:
OnMultiLayer Xor (HoleDiameter <> 0)
To be returned, an object has to either be on the Multi-Layer layer and have a Hole Diameter that is zero, or not be on the Multi-
Layer layer and have a Hole Diameter that is not zero.

Not
This is used to implement Logical NOT.
Example:
Not OnMultiLayer
To be returned, an object has to not be on the Multi-Layer layer.

Comparison Operators

< This means Less Than.
Example:
HoleDiameter < 40

<= This means Less Than Or Equal To.
Example:
HoleDiameter <= 40

>= This means Greater Than Or Equal To.
Example:
HoleDiameter >= 40

> This means Greater Than.
Example:
HoleDiameter > 40

<> This means Not Equal To.
Example:
HoleDiameter <> 40
To be returned, an object has to have a Hole Diameter which is not equal to 40.

= This means Equal To.
Example:
HoleDiameter = 40
To be returned, an object has to have a Hole Diameter which is equal to 40.

Between ... And ... This means Greater Than Or Equal To the first number, and Less Than Or Equal To the second number.
Example:
HoleDiameter Between 30 And 50
To be returned, an object has to have a Hole Diameter that is greater than or equal to 30, and less than or equal to 50.

Like Keywords that are used in conjunction with user-specified strings require that these strings be exactly specified whenever
these strings are being compared by the = operator. The use of the Like operator permits the provision of strings which contain
one or more Wild Card characters, thus supporting the comparison of strings which are not exactly specified. (See the
immediately following Section for a description of Wild Card Characters.)
Example:
Name Like 'ADDR?*' 
This returns objects having a Name property whose associated (text) string begins with 'ADDR' and which contains at least one
more character; the string property required has thus not been exactly specified, but has been partially specified.
Wild Card Characters

Wild Card characters permit the provision of strings which are not exactly specified. These characters are typically used in conjunction with other characters, resulting in the provision of strings which are partially specified. A few exceptional keywords can accept string parameters that are not exactly specified, but for the most part, strings can only contain Wild Card characters when these are being compared by the Like operator.

? This can be replaced by a single character of any type.

Example:

Footprint Like 'DIP1?'

This returns objects which have a Footprint property of 'DIP10', or 'DIP12', or 'DIP14', etc.

* This can be replaced by any number of characters, each of which can be of any type.

Example:

Footprint Like 'SIP*' 

This returns objects which have a Footprint property of 'SIP1', or 'SIP12', or 'SIP216', etc. (Any objects having a Footprint property of 'SIP' are also returned, because * can also be replaced by no characters.)

Boolean Strings

True This affirms the meaning of a Keyword.

Example:

IsPad = True

To be returned, an object has to be a pad.

False This negates the meaning of a Keyword.

Example:

IsVia = False

To be returned, an object has to not be a via.

Brackets and Order of Precedence

It is highly advisable to use brackets within a query whenever there is any possibility whatsoever that the query might not be correctly interpreted.

The following two queries provide a demonstration of how brackets are used in order to eliminate ambiguity:

((IsPad And OnMultiLayer) Or IsVia) And (HoleDiameter < 16)

(IsPad And OnMultiLayer) Or (IsVia And (HoleDiameter < 16))

Both of those queries return all vias whose hole diameter is less than 16mil, but each of them differs in which pads are also returned. The first query returns pads on the Multi-Layer layer whose hole diameter is less than 16mil, but the second query returns all pads on the Multi-Layer layer, regardless of their hole diameters.

Brackets have the highest precedence within an order of precedence that has been defined for the various operators provided, and which determines how queries are interpreted by the software (whenever the user has not provided brackets). The sequence of this order is as follows:

Brackets

Not

^, *, /, Div, Mod, And

+, -, Or, Xor

=, <>, <, >, <=, >=

&&, ||

Ambiguities are resolved by working from left to right. Parentheses are evaluated from inside to outside and equal levels are done left to right.
Examples:
A And B Or C becomes (A And B) Or C
A * B + C becomes (A * B) + C
This order of precedence is similar to that used in Pascal type languages. However, generous usage of brackets removes doubt, and makes the resulting queries easier to read by others.

Library Functions

Comment Field

Description
Returns all Library Component objects having a Comment property that complies with the Query.

Note: The child objects of these Component objects are not returned. The Comment property is only defined for Component objects.

Syntax
Comment : String

Examples
Comment = '100nF'
Returns all Components that have a Comment property of '100nF'.

Comment <> '100nF'
Returns all objects except these Component objects that have a Comment property of '100nF'. (Only Component objects have a Comment property, thus the remaining objects that do not have a Comment property of '100nF' are returned by this Query.)

ComponentType Field

Description
Returns all Schematic Library Component objects having a ComponentType property that complies with the Query. Note, a component type can be one of the following types; Standard, Mechanical, Graphical, Net Tie, Net Tie (In BOM), Standard (No BOM).

Syntax
ComponentType : String

Examples
ComponentType = 'Graphical'
Returns all Part objects that have a 'Graphical' Component Type property.

ComponentType <> 'Standard'
Returns all objects except Part objects that have a 'Standard' Component Type property. (Only Part objects have a ComponentType property, so all remaining types of objects do not have a 'Standard' ComponentType property, and are thus also returned by this Query.)

IsPart && ComponentType <> 'Standard'
Returns all Part objects that do not have a 'Standard' Component Type property.

Designator Field

Description
Returns all Schematic Library Component objects having a designator property that complies with the Query.

Syntax
Designator : String

Examples
Designator Like 'C*'  
Returns all Components that have a Designator property of 'C*'.

Designator <> 'C*'  
Returns all objects except these Component objects that have a Designator property of 'C*'.

Description Field

Description  
Returns all Library Component objects having a Description property that complies with the Query.

Syntax  
Description: String

Examples  
Description = 'F*'  
Returns all components or footprints that have a Description property with the first character of the string starting with F.

HasModel Field

Description  
Returns schematic components which meet the query criteria that has the HasModel keyword with the specified model type, model name and current model.

Syntax  
HasModel(ModelType : String , ModelName : String , CurrentModelOnly : Boolean) : Boolean

The ModelType string specifies the Type property of a linked Model, and must be one of the strings from the following list: 'PCB3DLIB', 'PCBLIB', 'SI', 'SIM'

Those strings respectively select linked Models having Type properties of PCB3D, Footprint, Signal Integrity, and Simulation.

The ModelName string specifies the Name property of a linked Model.

The CurrentModelOnly parameter specifies whether or not a linked Model (also) has to be a Current Model of each Part. When this is False, a linked Model does not have to be a Current Model (but it still can be); when this is True, a linked Model does have to be a Current Model.

Examples  
HasModel('PCBLIB','SOIC14',False)  
HasModel('PCBLIB','SOIC14',False) = True  
Returns all Part objects that are linked to a Footprint Model which has a Name property of 'SOIC14'; that Model does not have to be each Part's Current Footprint Model (but it can be).

HasModel('PCBLIB','DIP14',True)  
HasModel('PCBLIB','DIP14',True) = True  
Returns all Part objects that are linked to a Footprint Model which has a Name property of 'DIP14'; for each Part returned, that Model also has to be its Current Footprint Model.

HasModel('SI','RES1',False)  
HasModel('SI','RES1',False) = True  
Returns all Part objects that are linked to a Signal Integrity Model which has a Name property of 'RES1'; that Model does not have to be each Part's Current Signal Integrity Model (but it can be).

HasModel('SIM','RESISTOR',True)  
HasModel('SIM','RESISTOR',True) = True  
Returns all Part objects that are linked to a Simulation Model which has a Name property of 'RESISTOR'; for each Part returned, that Model also has to be its Current Simulation Model.
HasModelParameter Field

Description
The HasModelParameter keyword permits users to specify the Name and Value properties of a Parameter which is contained within a linked Model, and optionally whether that Model is a Current Model of each Part. (Each Part can be linked to Models of Footprint, Simulation, PCB3D, and Signal Integrity types. It is possible to link a Part to more than one Model of the same type, but only one Model of each type can be selected as a Part's Current Model.)

Syntax
HasModelParameter(ParameterName : String, ParameterValue : String, CurrentModelOnly : Boolean) : Boolean

The ParameterName string specifies the Parameter Name property of the Parameter contained in the linked Model.
The ParameterValue string specifies the (Parameter) Value property of the Parameter contained in the linked Model.
The CurrentModelOnly parameter specifies whether or not the linked Model (which contains a compliant Parameter) has to be a Current Model of each Part. When this is False, the linked Model does not have to be a Current Model (but it still can be); when this is True, the linked Model does have to be a Current Model.

Examples
HasModelParameter('Inductance A','1mH',False)
HasModelParameter('Inductance A','1mH',False) = True

Returns all Part objects that are linked to a Model which contains a Parameter that has a Parameter Name property of 'Inductance A' and a (Parameter) Value property of '1mH'; that Model does not have to be a Current Model of each Part (but it can be).

HasModelParameter('Coupling Factor','0.999',True)
HasModelParameter('Coupling Factor','0.999',True) = True

Returns all Part objects that are linked to a Model which contains a Parameter that has a Parameter Name property of 'Coupling Factor' and a (Parameter) Value property of '0.999', when that Model is also a Current Model of the Part.

Height Field

Description
Returns all footprint objects within PCB libraries having a Height property that complies with the Query that has the Height keyword.

Syntax
Height : Number

Example
(Height > 10) and (Height < 25)

Returns all the footprints with a height property of between 10 and 25 mils.

LibraryName field

Description
Returns all Library Components having a LibraryName property that complies with the Query. You need to specify the search type on the Libraries Search dialog when searching for PCB Libraries, Schematic Libraries etc. There can be multiple libraries that have the same name living in different locations on the hard disk for example and those components will be returned from these libraries.

Syntax
LibraryName : String

Examples
LibraryName = '4 Port Serial Interface.PcbLib'

Returns all footprints that live in a library specified by the LibraryName property as long as the Search type is set to Protel Footprints.
LibraryName = '3M Card CompactFlash.IntLib'
Returns all footprints that live in an integrated library specified by the LibraryName property if the Search type is set to Protel Footprints or returns all components if the Search type is set to Components etc.

**LibraryPath field**

**Description**
Returns all Library Component objects having a LibraryPath property that references that specific library and complies with the Query. You need to specify the search type on the Libraries Search dialog when searching for PCB libraries, Schematic Libraries etc.

**Syntax**
LibraryPath: String

**Examples**
LibraryPath = 'C:\Program Files\Altium Designer\Examples\Reference Designs\4 Port Serial Interface\Libraries\4 Port Serial Interface.PcbLib'
Returns all Footprints etc that points to a specific library.
LibraryPath = 'C:\Program Files\Altium Designer\Library\3m\3M Card CompactFlash.IntLib'
Returns all Components or Footprints etc that have a LibraryPath property of the full path to the specified library depending on which Search type is set in the Libraries Search dialog.

**LibReference Field**

**Description**
Returns all Library Component objects having a LibReference property which meet the query criteria.

**Syntax**
LibReference : String

**Examples**
LibReference Like 'C'*
Returns all Components that have a LibReference property of strings that start with C.

**Name field**

**Description**
Returns all Footprints or Schematic Library Component objects having a Name property that complies with the Query. The Libraries Search dialog needs to be set to the search type before you can do a search with a specified query.

**Syntax**
Name : String

**Examples**
Name = 'DIP14'
Returns all Footprints/Components that have a Name property of 'DIP14' as long the appropriate library exists.

**PadCount Field**

**Description**
Returns all footprints having a Pad Count property that complies with the Query.

**Syntax**
PadCount : Number

**Examples**
PadCount > 60
Returns all footprints that have a pad count of greater than 60

**PartCount Field**

**Description**
Returns all Schematic Library Component objects having a Part Count property that complies with the Query. Note, a component can have more than one part. For example a 74HC32 integrated circuit has 4 OR gates and thus there are four parts for the 74HC32 multi-part component.

**Syntax**
```
PartCount : Number
```

**Examples**
```
PartCount > 4
```
returns all components that have more than 4 parts.

**PinCount Field**

**Description**
Returns all Schematic Library Component objects having a Pin Count property that complies with the Query.

**Syntax**
```
PinCount : Number
```

**Examples**
```
PinCount < 2
```
returns the components in the installed libraries that have less than 2 pins.

**SourceLibraryName field**

**Description**
Returns all Library Component or Footprint objects having a Source Library Name property that represents the source library which complies with the Query. The Source Library Name property represents the name of the source library such as the Schematic Library or the PCB library. Integrated Libraries do not work in this case.

**Syntax**
```
SourceLibraryName : String
```

**Examples**
```
SourceLibraryName = '4 Port Serial Interface.PcbLib'
```
Returns all footprints that live in a library specified by the SourceLibraryName property as long as the Search type is set to Protel Footprints.

**SourceLibraryPath field**

**Description**
Returns all Library Component objects having a SourceLibraryPath property that references that specific library (PCBLib or SchLIB) and complies with the Query. You need to specify the search type on the Libraries Search dialog when searching for PCB libraries, Schematic Libraries etc. The Source Library path property represents the full path and name of the source library such as the Schematic Library or the PCB library. Integrated Libraries do not work in this case.

**Syntax**
```
SourceLibraryPath: String
```

**Examples**
```
SourceLibraryPath = 'C:\Program Files\Altium Designer\Examples\Reference Designs\4 Port Serial Interface\Libraries\4 Port Serial Interface.PcbLib'
```
Returns all Footprints etc that points to a specific library.
Schematic Functions

Alignment Field

Description
Returns all Note, Port, and Text Frame objects having an Alignment property that complies with the Query.
Note: The Alignment property is only defined for Note, Port, and Text Frame objects.

Syntax
Alignment = Alignment_String
Alignment <> Alignment_String

Alignment_String must be one of the strings from the following list:
'Bottom', 'Center', 'Left', 'Right', 'Top'

Note: The single quote characters ('') shown at the start and end of each Alignment_String are both mandatory.

Examples
Alignment = 'Center'
Returns all Note, Port, and Text Frame objects that have a 'Center' Alignment property.

Alignment <> 'Right'
Returns all objects except Note, Port, and Text Frame objects that have a 'Right' Alignment property. (Only Note, Port, and Text Frame objects have an Alignment property, so all remaining types of objects do not have a 'Right' Alignment property, and are thus also returned by this Query.)

Alignment <> '' && Alignment <> 'Right'
Alignment > '' && Alignment <> 'Right'
Returns all Note, Port, and Text Frame objects that do not have a 'Right' Alignment property.

Author Field

Description
Returns all Note objects having an Author property that complies with the Query.
Note: The Author property is only defined for Note objects.

Syntax
Author : String

Examples
Author = 'George'
Returns all Note objects that have an Author property of 'George'.

Author Like 'Ge*'
Returns all Note objects that have an Author property whose associated string starts with 'Ge'.

Author <> 'Robert'
Returns all objects except Note objects that have an Author property of 'Robert'. (Only Note objects have an Author property, so all remaining types of objects do not have an Author property of 'Robert', and are thus also returned by this Query.)

Author <> '' && Author <> 'Robert'
Author > '' && Author <> 'Robert'
IsNote && Author <> 'Robert'
Returns all Note objects that do not have an Author property of 'Robert'.

BorderWidth Field

Description
Returns all Image, Pie, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects having a Border Width property that complies with the Query.

Note: The BorderWidth property is only defined for Image, Pie, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects.

Syntax

```plaintext
BorderWidth = Width_String
BorderWidth <> Width_String
```

Width_String must be one of the strings from the following list:

'Medium', 'Large', 'Small', 'Smallest'

Note: The single quote characters (') shown at the start and end of each Width_String are both mandatory.

Examples

```plaintext
BorderWidth = 'Medium'

Returns all Image, Pie, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects that have a 'Medium' Border Width property.

BorderWidth <> 'Smallest'

Returns all objects except Image, Pie, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects that have a 'Smallest' Border Width property. (Only Image, Pie, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects have a BorderWidth property, so all remaining types of objects do not have a 'Smallest' BorderWidth property, and are thus also returned by this Query.)

BorderWidth <> '' && BorderWidth <> 'Smallest'
BorderWidth > '' && BorderWidth <> 'Smallest'

Returns all Image, Pie, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects that do not have a 'Smallest' Border Width property.
```

Collapsed Field

Description

Returns all Note objects having a Collapsed property that complies with the Query.

Note: The Collapsed property is only defined for Note objects.

Syntax

```plaintext
Collapsed : Boolean_String
```

Boolean_String must be either 'True' or 'False'.

Examples

```plaintext
Collapsed = 'True'

Returns all Note objects that have a 'True' Collapsed property.

Collapsed = 'False'

Returns all Note objects that have a 'False' Collapsed property.

Collapsed <> 'True'
Not (Collapsed = 'True')

Returns all objects except Note objects that have a 'True' Collapsed property.

Collapsed <> 'False'
Not (Collapsed = 'False')

Returns all objects except Note objects that have a 'False' Collapsed property.
```

Color Field (Schematic)

Description

Returns all objects having a Color property that complies with the Query.
Syntax

Color : Numeric_String

Color : Number {If all Part objects are totally excluded by one or more preceding tokens within the Query.}

The Color keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all Part and Probe objects (which do not have a Color property) have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:

Color <> ''
Color > ''
ObjectKind <> 'Part' && ObjectKind <> 'Probe'
Not IsPart && Not IsProbe
IsPie
IsArc Or IsBezier

The usage of such tokens is demonstrated in the examples that follow.

The color corresponding to a particular number can be deduced from the following relationship:

65536 * Blue + 256 * Green + Red

where the Blue, Green, and Red components each have a value between 0 and 255 (inclusive).

When the Color keyword can be used in conjunction with a Number (as previously described), it is additionally possible to individually evaluate the values of each of the Color property's Blue, Green, and Red components, by using the following constructs:

Color Div 65536 {Blue component}
Color Div 256 Mod 256 {Green component}
Color Mod 256 {Red component}

The usage of these constructs is also demonstrated in the examples that follow.

Examples

Color = '0'
Color <> '' && Color = 0

Returns all objects that have a black Color property. (A Color property of 0, as defined above, corresponds to values of Blue = 0, Green = 0, and Red = 0).

Color = '16711680'
Color > '' && Color = 16711680

Returns all objects that have a Color property of 16711680 (as defined above, which corresponds to values of Blue = 255, Green = 0, and Red = 0).

Color <> '16777215'

Returns all objects except those that have a (maximum intensity) white Color property. (A Color property of 16777215, as defined above, corresponds to values of Blue = 255, Green = 255, and Red = 255).

ObjectKind <> 'Part' && ObjectKind <> 'Probe' && Color <> 16777215

Returns all objects that have a Color property, and whose Color property is not (maximum intensity) white.

Not IsPart && Not IsProbe && Color Div 65536 >= 96 && Color Div 256 Mod 256 Between 64 And 192 && Color Mod 256 <= 128

Returns all objects that have a Color property, and whose Blue component has a value which is greater than or equal to 96, and whose Green component has a value which is greater than or equal to 64 and less than or equal to 192, and whose Red component has a value which is less than or equal to 128.

IsPie && Color Div 65536 <= 192

Returns all Pie objects whose Color property's Blue component has a value which is less than or equal to 192.

IsArc Or IsBezier && Color Mod 256 Between 64 And 192

Returns all Arc and Bezier objects whose Color property's Red component has a value which is greater than or equal to 64 and less than or equal to 192.
ComponentType Field (Schematic)

Description
Returns all Part objects having a Component Type property that complies with the Query.
Note: The ComponentType property is only defined for Part objects.

Syntax
ComponentType = ComponentType_String
ComponentType <> ComponentType_String

ComponentType_String must be one of the strings from the following list:
'Graphical' , 'Mechanical', 'Net Tie', 'Net Tie (In BOM)' , 'Standard' , 'Standard (No BOM)'
Note: The single quote characters (') shown at the start and end of each ComponentType_String are both mandatory.

Examples
ComponentType = 'Graphical'
Returns all Part objects that have a 'Graphical' Component Type property.
ComponentType <> 'Standard'
Returns all objects except Part objects that have a 'Standard' Component Type property. (Only Part objects have a ComponentType property, so all remaining types of objects do not have a 'Standard' ComponentType property, and are thus also returned by this Query.)
IsPart && ComponentType <> 'Standard'
Returns all Part objects that do not have a 'Standard' Component Type property.

Configuration Field

Description
Returns all Part objects having a Configuration property that complies with the Query.
Note: The Configuration property is only defined for Part objects.

Syntax
Configuration : String

Example
Configuration = 'PortSwitch_PCB'
Returns all Part objects that have a Configuration property of 'PortSwitch_PCB'.

CornerLocationX Field

Description
Returns all Bus Entry, Image, (non-Polyline) Line, Note, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects having a X2 property that complies with the Query.
Note: The CornerLocationX property is only defined for Bus Entry, Image, (non-Polyline) Line, Note, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects.
Note: Line objects can be of either a Line or a Polyline nature; the former have just two vertices, and only exist in files which were created using earlier versions of Protel. The latter can have more than two vertices, and are always created whenever the 'Place Line' command is used in Altium Designer.

Syntax
CornerLocationX : Numeric_String
CornerLocationX : Number {If all objects which do not have a X2 property are totally excluded by one or more preceding tokens within the Query.}

The CornerLocationX keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all objects which do not have a X2 property have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:
CornerLocationX <> ''
CornerLocationX > ''
IsRectangle Or IsRoundRectangle
IsSheetSymbol

The usage of such tokens is demonstrated in the examples that follow.

Examples

CornerLocationX = '320'
CornerLocationX <> '' && CornerLocationX = 320
Returns all Bus Entry, Image, (non-Polyline) Line, Note, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects that have a X2 property which is equal to 320.

CornerLocationX > '' && CornerLocationX Between 150 And 430
Returns all Bus Entry, Image, (non-Polyline) Line, Note, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects that have a X2 property which is greater than or equal to 150 and less than or equal to 430.

IsRectangle Or IsRoundRectangle && CornerLocationX >= 250
Returns all Rectangle and Round Rectangle objects that have a X2 property which is greater than or equal to 250.

IsSheetSymbol && CornerLocationX < 350
Returns all Sheet Symbol objects that have a X2 property which is less than 350.

CornerLocationX <> '210'
Returns all objects except Bus Entry, Image, (non-Polyline) Line, Note, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects that have a X2 property which is equal to 210. (Only Bus Entry, Image, (non-Polyline) Line, Note, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects have a CornerLocationX property, so all remaining types of objects do not have a CornerLocationX property of 210, and are thus also returned by this Query.)

CornerLocationX > '' && CornerLocationX <> 210
Returns all Bus Entry, Image, (non-Polyline) Line, Note, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects that have a X2 property which is not equal to 210.

CornerLocationY Field

Description
Returns all Bus Entry, Image, (non-Polyline) Line, Note, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects having a Y2 property that complies with the Query.

Note: The CornerLocationY property is only defined for Bus Entry, Image, (non-Polyline) Line, Note, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects.

Note: Line objects can be of either a Line or a Polyline nature; the former have just two vertices, and only exist in files which were created using earlier versions of Protei. The latter can have more than two vertices, and are always created whenever the 'Place Line' command is used in Altium Designer.

Syntax
CornerLocationY : Numeric_String
CornerLocationY : Number {If all objects which do not have a Y2 property are totally excluded by one or more preceding tokens within the Query.}

The CornerLocationY keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all objects which do not have a Y2 property have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:

CornerLocationY <> ''
CornerLocationY > ''
IsRectangle Or IsRoundRectangle
IsSheetSymbol

The usage of such tokens is demonstrated in the examples that follow.

Examples
CornerLocationY = '320'
CornerLocationY <> '' && CornerLocationY = 320
Returns all Bus Entry, Image, (non-Polyline) Line, Note, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects that have a Y2 property which is equal to 320.

CornerLocationY > '' && CornerLocationY Between 150 And 430
Returns all Bus Entry, Image, (non-Polyline) Line, Note, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects that have a Y2 property which is greater than or equal to 150 and less than or equal to 430.

IsRectangle Or IsRoundRectangle && CornerLocationY >= 250
Returns all Rectangle and Round Rectangle objects that have a Y2 property which is greater than or equal to 250.

IsSheetSymbol && CornerLocationY < 350
Returns all Sheet Symbol objects that have a Y2 property which is less than 350.

CornerLocationY <> '210'
Returns all objects except Bus Entry, Image, (non-Polyline) Line, Note, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects that have a Y2 property which is equal to 210. (Only Bus Entry, Image, (non-Polyline) Line, Note, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects have a CornerLocationY property, so all remaining types of objects do not have a CornerLocationY property of 210, and are thus also returned by this Query.)

CornerLocationY > '' && CornerLocationY <> 210
Returns all Bus Entry, Image, (non-Polyline) Line, Note, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects that have a Y2 property which is not equal to 210.

**CurrentFootprint Field**

**Description**
Returns all Part objects having a Current Footprint property that complies with the Query.

**Note:** The CurrentFootprint property is only defined for Part objects.

**Syntax**
CurrentFootprint : String

**Examples**
CurrentFootprint = 'DIP14'
Returns all Part objects that have a Current Footprint property of 'DIP14'.

CurrentFootprint Like 'SIP?*'
Returns all Part objects that have a Current Footprint property whose associated string starts with 'SIP', and which contains at least one more following character.

**DatabaseTableName Field**

**Description**
Returns all components objects having a DatabaseTableName property that complies with the Query.

**Syntax**
DatabaseTableName : String

**Examples**
DatabaseTableName = 'TableName'
Returns all objects that have a DatabaseTableName property of 'TableName'.

DatabaseTableName Like '*Table Name *
Returns all objects that have a DatabaseTableName property whose associated string includes a substring of 'Table Name'.

**Description Field**

**Description**
Returns all Part and Pin objects having a Description property that complies with the Query.
Note: The Description property is only defined for Part and Pin objects.

**Syntax**

```
Description : String
```

**Examples**

- `Description = 'TTL-RS232 DRIVER'`
  Returns all Part and Pin objects that have a Description property of 'TTL-RS232 DRIVER'.
- `Description Like '*RS485*'`
  Returns all Part and Pin objects that have a Description property whose associated string includes a substring of 'RS485'.

---

**DesignatorLocked Field**

**Description**

Returns all Part objects having a Lock Designator property that complies with the Query.
Note: The DesignatorLocked property is only defined for Part objects.

**Syntax**

```
DesignatorLocked : Boolean_String
```

Boolean_String must be either 'True' or 'False'.

**Examples**

- `DesignatorLocked = 'True'`
  Returns all Part objects that have a 'True' Lock Designator property.
- `DesignatorLocked = 'False'`
  Returns all Part objects that have a 'False' Lock Designator property.
- `DesignatorLocked <> 'True'`
  Not (DesignatorLocked = 'True')
  Returns all objects except Part objects that have a 'True' Lock Designator property.
- `DesignatorLocked <> 'False'`
  Not (DesignatorLocked = 'False')
  Returns all objects except Part objects that have a 'False' Lock Designator property.

---

**EndAngle Field**

**Description**

Returns all Arc, Elliptical Arc, and Pie objects having a Stop Angle property that complies with the Query.

Note: The EndAngle property is only defined for Arc, Elliptical Arc, and Pie objects.

**Syntax**

```
EndAngle : Numeric_String
```

The number returned specifies the Stop Angle in units of degrees.

The EndAngle keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all objects which do not have a Stop Angle property have been totally excluded by one or more preceding tokens within the Query.

Examples of thus-compliant tokens are as follows:

- `IsArc Or IsEllipticalArc Or IsPie`
- `IsPie`
- `IsArc Or IsEllipticalArc`

The usage of such tokens is demonstrated in the examples that follow.
Examples

EndAngle = '30'
IsArc Or IsEllipticalArc Or IsPie && EndAngle = 30
Returns all Arc, Elliptical Arc, and Pie objects that have a Stop Angle property which is equal to 30 degrees.

IsArc Or IsEllipticalArc Or IsPie && EndAngle >= 60
Returns all Arc, Elliptical Arc, and Pie objects that have a Stop Angle property which is greater than or equal to 60 degrees.

IsArc Or IsEllipticalArc Or IsPie && EndAngle Between 30 And 150
Returns all Arc, Elliptical Arc, and Pie objects that have a Stop Angle property which is greater than or equal to 30 degrees and less than or equal to 150 degrees.

IsPie && EndAngle < 45
Returns all Pie objects that have a Stop Angle property which is less than 45 degrees.

EndAngle <> '135'
Returns all objects except Arc, Elliptical Arc, and Pie objects that have a Stop Angle property which is equal to 135 degrees. (Only Arc, Elliptical Arc, and Pie objects have an EndAngle property, so all remaining types of objects do not have an EndAngle property of 135 degrees, and are thus also returned by this Query.)

IsArc Or IsEllipticalArc && EndAngle <> 135
Returns all Arc and Elliptical Arc objects that have a Stop Angle property which is not equal to 135 degrees.

IsArc Or IsEllipticalArc Or IsPie && (EndAngle Between 315 And 360 || EndAngle Between 0 And 45)
Returns all Arc, Elliptical Arc, and Pie objects that have a Stop Angle property which is greater than or equal to 315 degrees and less than or equal to 45 degrees.

See Also
StartAngle

FileName Field

Description
Returns all Image, Part, and Sheet File Name objects having a File Name property that complies with the Query.
Note: The FileName property is only defined for Image, Part, and Sheet File Name objects.

Syntax
FileName : String

Examples
FileName = '4 Port Serial Interface.SchDoc'
Returns all Image, Part, and Sheet File Name objects that have a File Name property of '4 Port Serial Interface.SchDoc'.

FileName Like '*Parallel*'
Returns all Image, Part, and Sheet File Name objects that have a File Name property whose associated string includes a substring of 'Parallel'.

FillColor Field

Description
Returns all Ellipse, Note, Pie, Polygon, Port, Rectangle, Round Rectangle, Sheet Entry, Sheet Symbol, and Text Frame objects having a Fill Color property that complies with the Query.
Note: The FillColor property is only defined for Ellipse, Note, Pie, Polygon, Port, Rectangle, Round Rectangle, Sheet Entry, Sheet Symbol, and Text Frame objects.

Syntax
FillColor : Numeric_String

Examples
FillColor : Number {If all objects which do not have a Fill Color property are totally excluded by one or more preceding tokens within the Query.}
The FillColor keyword can **always** be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but **only** when all objects which do **not** have a Fill Color property have been **totally** excluded by one or more **preceding** tokens within the Query. Examples of thus-compliant tokens are as follows:

```
FillColor <> ''
FillColor > ''
IsTextFrame
IsEllipse Or IsPolygon
```

The usage of such tokens is demonstrated in the examples that follow.

The color corresponding to a particular number can be deduced from the following relationship:

\[65536 \times \text{Blue} + 256 \times \text{Green} + \text{Red}\]

where the Blue, Green, and Red components each have a value between 0 and 255 (inclusive).

When the FillColor keyword can be used in conjunction with a Number (as previously described), it is additionally possible to individually evaluate the values of each of the Fill Color property’s Blue, Green, and Red components, by using the following constructs:

```
FillColor Div 65536  \{Blue component\}
FillColor Div 256 Mod 256  \{Green component\}
FillColor Mod 256  \{Red component\}
```

The usage of these constructs is also demonstrated in the examples that follow.

**Examples**

```
FillColor = '0'
FillColor <> '' && FillColor = 0
Returns all objects that have a black Fill Color property. (A Fill Color property of 0, as defined above, corresponds to values of Blue = 0, Green = 0, and Red = 0).

FillColor = '11599871'
FillColor > '' && FillColor = 11599871
Returns all objects that have a Fill Color property of 11599871 (as defined above, which corresponds to values of Blue = 176, Green = 255, and Red = 255).

FillColor <> '16777215'
Returns all objects except those that have a (maximum intensity) white Fill Color property. (A Fill Color property of 16777215, as defined above, corresponds to values of Blue = 255, Green = 255, and Red = 255).

FillColor > '' && FillColor <> 16777215
Returns all objects that have a Fill Color property, and whose Fill Color property is **not** (maximum intensity) white. (A Fill Color property of 16777215, as defined above, corresponds to values of Blue = 255, Green = 255, and Red = 255).

IsTextFrame && FillColor Div 65536 <= 192
Returns all Text Frame objects whose Fill Color property's Blue component has a value which is less than or equal to 192.

IsEllipse Or IsPolygon && FillColor Mod 256 Between 64 And 192
Returns all Ellipse and Polygon objects whose Fill Color property's Red component has a value which is greater than or equal to 64 and less than or equal to 192.
```

**FontId Field**

**Description**

Returns all Designator, Net Label, Note, Parameter, Sheet File Name, Sheet Name, Text Frame, and Text String objects having a FontId property that complies with the Query.
Note: The FontId property is only defined for Designator, Net Label, Note, Parameter, Sheet File Name, Sheet Name, Text Frame, and Text String objects.

Syntax
FontID : Numeric_String

HasError Field

Description
Returns all objects which caused at least one violation to be reported during compilation.

Syntax
HasError : Boolean

Examples
HasError
HasError = True
Returns all objects which caused at least one violation to be reported during compilation.

Not HasError
HasError = False
Returns all objects which caused no violations to be reported during compilation.

HasModel Membership Check

Description
Returns each Part object that is linked to a Model whose properties match those specified by the Query.
Each Part can be linked to Models of Footprint, Simulation, PCB3D, and Signal Integrity types. It is possible to link a Part to more than one Model of the same type, but only one Model of each type can be selected as a Part's Current Model.
The HasModel keyword permits users to specify the Type and Name properties of a linked Model, and optionally whether that Model is (also) a Current Model of each Part.

Syntax
HasModel(ModelType : String , ModelName : String , CurrentModelOnly : Boolean) : Boolean

The ModelType string specifies the Type property of a linked Model, and must be one of the strings from the following list:
'PCB3DLIB' , 'PCBLIB' , 'SI' , 'SIM'
Those strings respectively select linked Models having Type properties of PCB3D, Footprint, Signal Integrity, and Simulation.
The ModelName string specifies the Name property of a linked Model.
The CurrentModelOnly parameter specifies whether or not a linked Model (also) has to be a Current Model of each Part. When this is False, a linked Model does not have to be a Current Model (but it still can be); when this is True, a linked Model does have to be a Current Model.

Examples
HasModel('PCBLIB','SOIC14',False)
HasModel('PCBLIB','SOIC14',False) = True
Returns all Part objects that are linked to a Footprint Model which has a Name property of 'SOIC14'; that Model does not have to be each Part's Current Footprint Model (but it can be).

HasModel('PCBLIB','DIP14',True)
HasModel('PCBLIB','DIP14',True) = True
Returns all Part objects that are linked to a Footprint Model which has a Name property of 'DIP14'; for each Part returned, that Model also has to be its Current Footprint Model.

HasModel('SI','RES1',False)
HasModel('SI','RES1',False) = True
Returns all Part objects that are linked to a Signal Integrity Model which has a Name property of 'RES1'; that Model does not have to be each Part's Current Signal Integrity Model (but it can be).

```
HasModel('SIM','RESISTOR',True)
```

```
HasModel('SIM','RESISTOR',True) = True
```

Returns all Part objects that are linked to a Simulation Model which has a Name property of 'RESISTOR'; for each Part returned, that Model also has to be its Current Simulation Model.

### HasModelParameter Membership Check

**Description**

Returns each Part object that is linked to a Model which contains a Parameter whose properties match those specified by the Query.

The HasModelParameter keyword permits users to specify the Name and Value properties of a Parameter which is contained within a linked Model, and optionally whether that Model is a Current Model of each Part. (Each Part can be linked to Models of Footprint, Simulation, PCB3D, and Signal Integrity types. It is possible to link a Part to more than one Model of the same type, but only one Model of each type can be selected as a Part's Current Model.)

**Syntax**

```
HasModelParameter(ParameterName : String , ParameterValue : String , CurrentModelOnly : Boolean) : Boolean
```

The ParameterName string specifies the Parameter Name property of the Parameter contained in the linked Model. The ParameterValue string specifies the (Parameter) Value property of the Parameter contained in the linked Model. The CurrentModelOnly parameter specifies whether or not the linked Model (which contains a compliant Parameter) has to be a Current Model of each Part. When this is False, the linked Model does not have to be a Current Model (but it still can be); when this is True, the linked Model does have to be a Current Model.

**Examples**

```
HasModelParameter('Inductance A','1mH',False)
```

```
HasModelParameter('Inductance A','1mH',False) = True
```

Returns all Part objects that are linked to a Model which contains a Parameter that has a Parameter Name property of 'Inductance A' and a (Parameter) Value property of '1mH'; that Model does not have to be a Current Model of each Part (but it can be).

```
HasModelParameter('Coupling Factor','0.999',True)
```

```
HasModelParameter('Coupling Factor','0.999',True) = True
```

Returns all Part objects that are linked to a Model which contains a Parameter that has a Parameter Name property of 'Coupling Factor' and a (Parameter) Value property of '0.999', when that Model is also a Current Model of the Part.

### HasParameter Membership Check

**Description**

Returns all Parameter Set, Part, Pin, Port, Probe, and Sheet Symbol objects incorporating a Parameter object having a Parameter Name property and (Parameter) Value property that complies with the Query.

**Syntax**

```
HasParameter(ParameterName : String , ParameterValue : String) : Boolean
```

The ParameterName string specifies the Parameter Name property of the Parameter. The ParameterValue string specifies the (Parameter) Value property of the Parameter.

**Examples**

```
HasParameter('Comment','1k')
```

```
HasParameter('Comment','1k') = True
```

Returns all Parameter Set, Part, Pin, Port, Probe, and Sheet Symbol objects incorporating a Parameter object that has a Parameter Name property of 'Comment' and a (Parameter) Value property of '1k'.

```
HasParameter('Text Field1','CAPACITOR')
```

```
HasParameter('Text Field1','CAPACITOR')
```
HasParameter('Text Field1','CAPACITOR') = True

Returns all Parameter Set, Part, Pin, Port, Probe, and Sheet Symbol objects incorporating a Parameter object that has a Parameter Name property of 'Text Field1' and a (Parameter) Value property of 'CAPACITOR'.

**HasPin Membership Check**

**Description**

Returns all Part objects incorporating a Pin object having a Pin Designator property and Name property that complies with the Query.

**Syntax**

HasPin(Pin Designator : String , Name : String) : Boolean

**Examples**

HasPin('7','GND')
HasPin('7','GND') = True

Returns all Part objects that contain a Pin object which has a Pin Designator property of '7' and a Name property of 'GND'.

HasPin('?','IN*')
HasPin('?','IN*') = True

Returns all Part objects that contain a Pin object which has a Pin Designator property whose associated string consists of one character, and a Name property whose associated string commences with 'IN'.
HasSheetEntry Membership Check

Description
Returns all Sheet Symbol objects incorporating a Sheet Entry object having a Name property that complies with the Query.

Syntax
HasSheetEntry(Sheet Entry Name : String) : Boolean

Example
HasSheetEntry('BUS[0..7]') = True

HorizontalJustification Field

Description
Returns all Designator, Parameter, Sheet File Name, Sheet Name, and Text String objects having a Horizontal Justification property that complies with the Query.

Note: The HorizontalJustification property is only defined for Designator, Parameter, Sheet File Name, Sheet Name, and Text String objects.

Syntax
HorizontalJustification = HorizontalJustification_String
HorizontalJustification <> HorizontalJustification_String

HorizontalJustification_String must be one of the strings from the following list:
'Center', 'Left', 'Right'

Note: The single quote characters (') shown at the start and end of each HorizontalJustification_String are both mandatory.

Examples
HorizontalJustification = 'Right'

IeeeSymbol Field

Description
Returns all IEEE Symbol objects having a Symbol property that complies with the Query.

Note: The IeeeSymbol property is only defined for IEEE Symbol objects.

Syntax
IeeeSymbol = IeeeSymbol_String
IeeeSymbol <> IeeeSymbol_String

IeeeSymbol_String must be one of the strings from the following list:

Note: The single quote characters (') shown at the start and end of each IeeeSymbol_String are both mandatory.

**Examples**

IeeeSymbol = 'Group Bin'

Returns all IEEE Symbol objects that have a 'Group Bin' Symbol property.

IeeeSymbol <> 'Open Output'

Returns all objects except IEEE Symbol objects that have an 'Open Output' Symbol property. (Only IEEE Symbol objects have an IeeeSymbol property, so all remaining types of objects do not have an 'Open Output' IeeeSymbol property, and are thus also returned by this Query.)

IeeeSymbol <> '' && IeeeSymbol <> 'Open Output'

IsSymbol && IeeeSymbol <> 'Open Output'

Returns all IEEE Symbol objects that do not have an 'Open Output' Symbol property.

**ImageEmbedded Field**

**Description**

Returns all Image objects having an Embedded property that complies with the Query.

Note: The ImageEmbedded property is only defined for Image objects.

**Syntax**

ImageEmbedded : Boolean_String

Boolean_String must be either 'True' or 'False'.

**Examples**

ImageEmbedded = 'True'

Returns all Image objects that have a 'True' Embedded property.

ImageEmbedded = 'False'

Returns all Image objects that have a 'False' Embedded property.

ImageEmbedded <> 'True'

Not (ImageEmbedded = 'True')

Returns all objects except Image objects that have a 'True' Embedded property.

ImageEmbedded <> 'False'

Not (ImageEmbedded = 'False')

Returns all objects except Image objects that have a 'False' Embedded property.

**ImageKeepAspectRatio Field**

**Description**

Returns all Image objects having a XY Ratio 1:1 property that complies with the Query.

Note: The ImageKeepAspectRatio property is only defined for Image objects.

**Syntax**

ImageKeepAspectRatio : Boolean_String

Boolean_String must be either 'True' or 'False'.

**Examples**

ImageKeepAspectRatio = 'True'
Returns all Image objects that have a ‘True’ XY Ratio 1:1 property.
```
ImageKeepAspect = 'False'
```

Returns all Image objects that have a ‘False’ XY Ratio 1:1 property.
```
ImageKeepAspect <> 'True'
```
```
Not (ImageKeepAspect = 'True')
```

Returns all objects except Image objects that have a ‘True’ XY Ratio 1:1 property.
```
ImageKeepAspect <> 'False'
```
```
Not (ImageKeepAspect = 'False')
```

Returns all objects except Image objects that have a ‘False’ XY Ratio 1:1 property.

**InComponent Membership Check (Schematic)**

**Description**
Returns each Part object having a Component Designator property matching that specified by the Query; all of the child objects of each such Part object are also returned.

This is an alias for **InPart (Membership Check)**.

**Syntax**
```
InComponent(Component Designator : String) : Boolean
InComponent(Component Designator : String , Component Designator : String) : Boolean
InComponent(Component Designator : String , Component Designator : String , Component Designator : String) : Boolean
```

**Examples**
```
InComponent('U4')
InComponent('U4') = True
```

Returns any Part that has a Component Designator property of ‘U4’, along with all of the child objects of each such Part object.
```
InComponent('R19','R42','J1')
InComponent('R19','R42','J1') = True
```

Returns any Part that has a Component Designator property of ‘R19’ or ‘R42’ or ‘J1’, along with all of the child objects of each such Part object.

**InPart Membership Check**

**Description**
Returns each Part object having a Component Designator property matching that specified by the Query; all of the child objects of each such Part object are also returned.

This is an alias for **InComponent (Membership Check)**.

**Syntax**
```
InPart(Component Designator : String) : Boolean
InPart(Component Designator : String , Component Designator : String) : Boolean
InPart(Component Designator : String , Component Designator : String , Component Designator : String) : Boolean
```

**Examples**
```
InPart('U4')
InPart('U4') = True
```

Returns any Part that has a Component Designator property of ‘U4’, along with all of the child objects of each such Part object.
```
InPart('R19','R42','J1')
InPart('R19','R42','J1') = True
```

InComponent Membership Check (Schematic)

**Description**
Returns each Part object having a Component Designator property matching that specified by the Query; all of the child objects of each such Part object are also returned.

This is an alias for **InPart (Membership Check)**.

**Syntax**
```
InComponent(Component Designator : String) : Boolean
InComponent(Component Designator : String , Component Designator : String) : Boolean
InComponent(Component Designator : String , Component Designator : String , Component Designator : String) : Boolean
```

**Examples**
```
InComponent('U4')
InComponent('U4') = True
```

Returns any Part that has a Component Designator property of ‘U4’, along with all of the child objects of each such Part object.
```
InComponent('R19','R42','J1')
InComponent('R19','R42','J1') = True
```

Returns any Part that has a Component Designator property of ‘R19’ or ‘R42’ or ‘J1’, along with all of the child objects of each such Part object.
Returns any Part that has a Component Designator property of 'R19' or 'R42' or 'J1', along with all of the child objects of each such Part object.

**InSheetSymbol Membership Check**

**Description**
Returns the Sheet Symbol object having a Sheet Name property matching that specified by the Query; all of its child objects are also returned.

**Syntax**
InSheetSymbol(Sheet Name : String) : Boolean

**Example**
InSheetSymbol('LCD Processor')
InSheetSymbol('LCD Processor') = True
Returns any Sheet Symbol object that has a Sheet Name property of 'LCD Processor', along with all of the child objects of each such Sheet Symbol object.

**IsArc Object Type Check (Schematic)**

**Description**
Returns Arc objects only.

**Syntax**
IsArc : Boolean

**Examples**
IsArc
IsArc = True
Returns all Arc objects.
Not IsArc
IsArc = False
Returns all objects except Arc objects.

**IsBezier Object Type Check**

**Description**
Returns Bezier objects only.

**Syntax**
IsBezier : Boolean

**Examples**
IsBezier
IsBezier = True
Returns all Bezier objects.
Not IsBezier
IsBezier = False
Returns all objects except Bezier objects.

**IsBus Object Type Check**

**Description**
Returns Bus objects only.
**Syntax**

IsBus : Boolean

**Examples**

IsBus

IsBus = True

Returns all Bus objects.

Not IsBus

IsBus = False

Returns all objects except Bus objects.

---

**IsBusEntry Object Type Check**

**Description**

Returns Bus Entry objects only.

**Syntax**

IsBusEntry : Boolean

**Examples**

IsBusEntry

IsBusEntry = True

Returns all Bus Entry objects.

Not IsBusEntry

IsBusEntry = False

Returns all objects except Bus Entry objects.

---

**IsComment Object Type Check (Schematic)**

**Description**

Returns Parameter objects which are child objects of Part objects and which have a 'Comment' Parameter Name property; these display the Part Comment property of their parent objects.

**Syntax**

IsComment : Boolean

**Examples**

IsComment

IsComment = True

Returns all Parameter objects which are child objects of Part objects and which have a 'Comment' Parameter Name property.

Not IsComment

IsComment = False

Returns all objects except Parameter objects which are child objects of Part objects and which have a 'Comment' Parameter Name property.

---

**IsDesignator Object Type Check (Schematic)**

**Description**

Returns Designator objects; these are child objects of Part objects, and display the Component Designator property of their parent objects.

**Syntax**

IsDesignator : Boolean

**Examples**
IsDesignator
IsDesignator = True
Returns all Designator objects.
Not IsDesignator
IsDesignator = False
Returns all objects except Designator objects.

IsEllipse Object Type Check

Description
Returns Ellipse objects only.
Syntax
IsEllipse : Boolean
Examples
IsEllipse
IsEllipse = True
Returns all Ellipse objects.
Not IsEllipse
IsEllipse = False
Returns all objects except Ellipse objects.

IsEllipticalArc Object Type Check

Description
Returns Elliptical Arc objects only.
Syntax
IsEllipticalArc : Boolean
Examples
IsEllipticalArc
IsEllipticalArc = True
Returns all Elliptical Arc objects.
Not IsEllipticalArc
IsEllipticalArc = False
Returns all objects except Elliptical Arc objects.

IsErrorMarker Object Type Check

Description
Returns Error Marker objects only.
Syntax
IsErrorMarker : Boolean
Examples
IsErrorMarker
IsErrorMarker = True
Returns all Error Marker objects.
Not IsErrorMarker
IsErrorMarker = False
Returns all objects except Error Marker objects.

**IsHidden Field**

**Description**
Returns all Parameter, Pin, Sheet File Name, and Sheet Name objects having a Hide property that complies with the Query.

Note: The IsHidden property is only defined for Parameter, Pin, Sheet File Name, and Sheet Name objects.

**Syntax**
IsHidden : Boolean

**Examples**
IsHidden
IsHidden = True
Returns all Parameter, Pin, Sheet File Name, and Sheet Name objects that have a True Hide property.

Not IsHidden
IsHidden = False
Returns all objects except Parameter, Pin, Sheet File Name, and Sheet Name objects that have a True Hide property.

**IsImage Object Type Check**

**Description**
Returns Image objects only.

**Syntax**
IsImage : Boolean

**Examples**
IsImage
IsImage = True
Returns all Image objects.

Not IsImage
IsImage = False
Returns all objects except Image objects.

**IsJunction Object Type Check**

**Description**
Returns Junction objects only.

**Syntax**
IsJunction : Boolean

**Examples**
IsJunction
IsJunction = True
Returns all Junction objects.

Not IsJunction
IsJunction = False
Returns all objects except Junction objects.

**IsLabel Object Type Check**

**Description**
Returns Text String objects.

**Syntax**

IsLabel : Boolean

**Examples**

IsLabel = True

Returns all Text String objects.

Not IsLabel

IsLabel = False

Returns all objects except Text String objects.

---

### IsLine Object Type Check

**Description**

Returns Line objects that are of a Line nature.

(Line objects can be of either a Line nature or a Polyline nature; the former have just two vertices, and only exist in files which were created using earlier versions of Protel. The latter can have more than two vertices, and are always created whenever the 'Place Line' command is used in Altium Designer.)

**Syntax**

IsLine : Boolean

**Examples**

IsLine = True

Returns all Line objects that are of a Line nature.

Not IsLine

IsLine = False

Returns all objects except Line objects that are of a Line nature.

---

### IsMirrored Field

**Description**

Returns all IEEE Symbol, Part, and Text String objects having a Mirrored property that complies with the Query.

Note: The IsMirrored property is only defined for IEEE Symbol, Part, and Text String objects.

**Syntax**

IsMirrored : Boolean_String

Boolean_String must be either 'True' or 'False'.

**Examples**

IsMirrored = 'True'

Returns all IEEE Symbol, Part, and Text String objects that have a 'True' Mirrored property.

IsMirrored = 'False'

Returns all IEEE Symbol, Part, and Text String objects that have a 'False' Mirrored property.

IsMirrored <> 'True'

Not (IsMirrored = 'True')

Returns all objects except IEEE Symbol, Part, and Text String objects that have a 'True' Mirrored property.

IsMirrored <> 'False'

Not (IsMirrored = 'False')

Returns all objects except IEEE Symbol, Part, and Text String objects that have a 'False' Mirrored property.
**IsNetLabel Object Type Check**

**Description**
Returns Net Label objects.

**Syntax**
IsNetLabel : Boolean

**Examples**
IsNetLabel
IsNetLabel = True
Returns all Net Label objects.
Not IsNetLabel
IsNetLabel = False
Returns all objects except Net Label objects.

**IsNoERC Object Type Check**

**Description**
Returns No ERC objects.

**Syntax**
IsNoERC : Boolean

**Examples**
IsNoERC
IsNoERC = True
Returns all No ERC objects.
Not IsNoERC
IsNoERC = False
Returns all objects except No ERC objects.

**IsNote Object Type Check**

**Description**
Returns Note objects.

**Syntax**
IsNote : Boolean

**Examples**
IsNote
IsNote = True
Returns all Note objects.
Not IsNote
IsNote = False
Returns all objects except Note objects.

**IsOffSheetConnector Object Type Check**

**Description**
Returns Off Sheet Connector objects.

**Syntax**
IsOffSheetConnector : Boolean
Examples
IsOffSheetConnector
IsOffSheetConnector = True
Returns all Off Sheet Connector objects.
Not IsOffSheetConnector
IsOffSheetConnector = False
Returns all objects except Off Sheet Connector objects.

IsParameter Object Type Check
Description
Returns Parameter objects.
Syntax
IsParameter : Boolean
Examples
IsParameter
IsParameter = True
Returns all Parameter objects.
Not IsParameter
IsParameter = False
Returns all objects except Parameter objects.

IsParameterSet Object Type Check
Description
Returns Parameter Set objects.
Note, a parameter set is is a group of parameters as a design parameter set directive for a wire or a net on the schematic document that can be transferred to its corresponding PCB document.
Syntax
IsParameterSet : Boolean
Examples
IsParameterSet
IsParameterSet = True
Returns all Parameter Set objects.
Not IsParameterSet
IsParameterSet = False
Returns all objects except Parameter Set objects.

IsPart Object Type Check
Description
Returns Part objects.
Note: This is an alias for IsSchComponent (Object Type Check).
Syntax
IsPart : Boolean
Examples
IsPart
IsPart = True
Returns all Part objects.
Not IsPart
IsPart = False
Returns all objects except Part objects.

**IsPie Object Type Check**

**Description**
Returns Pie objects.

**Syntax**
IsPie : Boolean

**Examples**
IsPie
IsPie = True
Returns all Pie objects.
Not IsPie
IsPie = False
Returns all objects except Pie objects.

**IsPin Object Type Check**

**Description**
Returns Pin objects.

**Syntax**
IsPin : Boolean

**Examples**
IsPin
IsPin = True
Returns all Pin objects.
Not IsPin
IsPin = False
Returns all objects except Pin objects.

**IsPolygon Object Type Check (Schematic)**

**Description**
Returns Polygon objects.

**Syntax**
IsPolygon : Boolean

**Examples**
IsPolygon
IsPolygon = True
Returns all Polygon objects.
Not IsPolygon
IsPolygon = False
Returns all objects except Polygon objects.
IsPolyline Object Type Check

Description
Returns Line objects that are of a Polyline nature.
(Line objects can be of either a Line nature or a Polyline nature; the former have just two vertices, and only exist in files which were created using earlier versions of Protel. The latter can have more than two vertices, and are always created whenever the 'Place Line' command is used in Altium Designer.)

Syntax
IsPolyline : Boolean

Examples
IsPolyline
IsPolyline = True
Returns all Line objects that are of a Polyline nature.
Not IsPolyline
IsPolyline = False
Returns all objects except Line objects that are of a Polyline nature.

IsPort Object Type Check

Description
Returns Port objects.

Syntax
IsPort : Boolean

Examples
IsPort
IsPort = True
Returns all Port objects.
Not IsPort
IsPort = False
Returns all objects except Port objects.

IsPowerObject Object Type Check

Description
Returns Power Object objects.

Syntax
IsPowerObject : Boolean

Examples
IsPowerObject
IsPowerObject = True
Returns all Power Object objects.
Not IsPowerObject
IsPowerObject = False
Returns all objects except Power Object objects.

IsProbe Object Type Check

Description
Returns Probe objects.

Syntax

IsProbe : Boolean

Examples

IsProbe

IsProbe = True

Returns all Probe objects.

Not IsProbe

IsProbe = False

Returns all objects except Probe objects.

IsRectangle Object Type Check

Description

Returns Rectangle objects.

Syntax

IsRectangle : Boolean

Examples

IsRectangle

IsRectangle = True

Returns all Rectangle objects.

Not IsRectangle

IsRectangle = False

Returns all objects except Rectangle objects.

IsRoundRectangle Object Type Check

Description

Returns Round Rectangle objects.

Syntax

IsRoundRectangle : Boolean

Examples

IsRoundRectangle

IsRoundRectangle = True

Returns all Round Rectangle objects.

Not IsRoundRectangle

IsRoundRectangle = False

Returns all objects except Round Rectangle objects.

IsSchComponent Object Type Check

Description

Returns Part objects.

Note: This is an alias for IsPart (Object Type Check).

Syntax

IsSchComponent : Boolean

Examples
IsSchComponent
IsSchComponent = True
Returns all Part objects.
Not IsSchComponent
IsSchComponent = False
Returns all objects except Part objects.

IsSelected Object Type Check (Schematic)

Description
Returns objects that are currently in a selected state. This permits the user to control specifically which objects are returned, as the user is always able to control (from the List Workspace Panel) which objects are currently selected.

Syntax
IsSelected : Boolean

Examples
IsSelected
IsSelected = True
Returns all objects that are currently in a selected state.
Not IsSelected
IsSelected = False
Returns all objects except those that are currently in a selected state.

IsSheetEntry Object Type Check

Description
Returns Sheet Entry objects.

Syntax
IsSheetEntry : Boolean

Examples
IsSheetEntry
IsSheetEntry = True
Returns all Sheet Entry objects.
Not IsSheetEntry
IsSheetEntry = False
Returns all objects except Sheet Entry objects.

IsSheetFileName Object Type Check

Description
Returns Sheet File Name objects.

Syntax
IsSheetFileName : Boolean

Examples
IsSheetFileName
IsSheetFileName = True
Returns all Sheet File Name objects.
Not IsSheetFileName
IsSheetFileName = False
Returns all objects except Sheet File Name objects.
Returns all objects except Sheet File Name objects.

IsSheetName Object Type Check

Description
Returns Sheet Name objects.

Syntax
IsSheetName : Boolean

Examples
IsSheetName
IsSheetName = True
Returns all Sheet Name objects.
Not IsSheetName
IsSheetName = False
Returns all objects except Sheet Name objects.

IsSheetSymbol Object Type Check

Description
Returns Sheet Symbol objects.

Syntax
IsSheetSymbol : Boolean

Examples
IsSheetSymbol
IsSheetSymbol = True
Returns all Sheet Symbol objects.
Not IsSheetSymbol
IsSheetSymbol = False
Returns all objects except Sheet Symbol objects.

IsSymbol Object Type Check

Description
Returns IEEE Symbol objects.

Syntax
IsSymbol : Boolean

Examples
IsSymbol
IsSymbol = True
Returns all IEEE Symbol objects.
Not IsSymbol
IsSymbol = False
Returns all objects except IEEE Symbol objects.

IsTaskHolder Object Type Check

Description
Returns Process Container objects.
Syntax
IsTaskHolder : Boolean

Examples
IsTaskHolder
IsTaskHolder = True
Returns all Process Container objects.
Not IsTaskHolder
IsTaskHolder = False
Returns all objects except Process Container objects.

IsTextFrame Object Type Check

Description
Returns Text Frame objects.

Syntax
IsTextFrame : Boolean

Examples
IsTextFrame
IsTextFrame = True
Returns all Text Frame objects.
Not IsTextFrame
IsTextFrame = False
Returns all objects except Text Frame objects.

IsWire Object Type Check (Schematic)

Description
Returns Wire objects.

Syntax
IsWire : Boolean

Examples
IsWire
IsWire = True
Returns all Wire objects.
Not IsWire
IsWire = False
Returns all objects except Wire objects.

JunctionSize Field

Description
Returns all Junction objects having a Size property that complies with the Query. Note: The JunctionSize property is only defined for Junction objects.

Syntax
JunctionSize = Width_String
JunctionSize <> Width_String

Width_String must be one of the strings from the following list:
'Medium', 'Large', 'Small', 'Smallest'
Note: The single quote characters (') shown at the start and end of each Width_String are both mandatory.

Examples
JunctionSize = 'Large'
Returns all Junction objects that have a 'Large' Size property.
JunctionSize <> 'Medium'
Returns all objects except Junction objects that have a 'Medium' Size property. (Only Junction objects have a JunctionSize property, so all remaining types of objects do not have a 'Medium' JunctionSize property, and are thus also returned by this Query.)
IsJunction && JunctionSize <> 'Medium'
Returns all Junction objects that do not have a 'Medium' Size property.

LineStyle Field
Description
Returns all Line objects having a Line Style property that complies with the Query. Note: The LineStyle property is only defined for Line objects.
Syntax
LineStyle = LineStyle_String
LineStyle <> LineStyle_String
LineStyle_String must be one of the strings from the following list: 'Dashed', 'Dotted', 'Solid'
Note: The single quote characters (') shown at the start and end of each LineStyle_String are both mandatory.
Examples
LineStyle = 'Dotted'
Returns all Line objects that have a 'Dotted' Line Style property.
LineStyle <> 'Dashed'
Returns all objects except Line objects that have a 'Dashed' Line Style property. (Only Line objects have a LineStyle property, so all remaining types of objects do not have a 'Dashed' LineStyle property, and are thus also returned by this Query.)
ObjectKind = 'Line' && LineStyle <> 'Dashed'
IsLine Or IsPolyLine && LineStyle <> 'Dashed'
LineStyle = 'Dotted' || LineStyle = 'Solid'
Returns all Line objects that do not have a 'Dashed' Line Style property.

LineWidth Field
Description
Returns all Arc, Bezier, Bus, Bus Entry, Ellipse, Elliptical Arc, IEEE Symbol, Line, Polygon, and Wire objects having a Line Width property that complies with the Query.
Note: The LineWidth property is only defined for Arc, Bezier, Bus, Bus Entry, Ellipse, Elliptical Arc, IEEE Symbol, Line, Polygon, and Wire objects.
Syntax
LineWidth = Width_String
LineWidth <> Width_String
Width_String must be one of the strings from the following list: 'Medium', 'Large', 'Small', 'Smallest'
Note: The single quote characters (') shown at the start and end of each Width_String are both mandatory.
Examples
LineWidth = 'Large'
Returns all Arc, Bezier, Bus, Bus Entry, Ellipse, Elliptical Arc, IEEE Symbol, Line, Polygon, and Wire objects that have a 'Large' Line Width property.

LineWidth <> 'Medium'

Returns all objects except Arc, Bezier, Bus, Bus Entry, Ellipse, Elliptical Arc, IEEE Symbol, Line, Polygon, and Wire objects that have a 'Medium' Line Width property. (Only Arc, Bezier, Bus, Bus Entry, Ellipse, Elliptical Arc, IEEE Symbol, Line, Polygon, and Wire objects have a LineWidth property, so all remaining types of objects do not have a 'Medium' LineWidth property, and are thus also returned by this Query.)

LineWidth <> '' && LineWidth <> 'Medium'
LineWidth > '' && LineWidth <> 'Medium'

Returns all Arc, Bezier, Bus, Bus Entry, Ellipse, Elliptical Arc, IEEE Symbol, Line, Polygon, and Wire objects that do not have a 'Medium' Size property.

LocationX Field

Description
Returns all objects having a X1 property that complies with the Query.

Syntax
LocationX : Numeric_String
LocationX : Number {If all Sheet Entry objects are totally excluded by one or more preceding tokens within the Query.}

The LocationX keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all Sheet Entry objects (which do not have a X1 property) have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:

LocationX <> ''
LocationX > ''
ObjectKind <> 'Sheet Entry'
Not IsSheetEntry

The usage of such tokens is demonstrated in the examples that follow.

Examples
LocationX = '320'
LocationX <> '' && LocationX = 320
Returns all objects that have a X1 property which is equal to 320.

LocationX > '' && LocationX Between '150' And '430'
Returns all objects that have a X1 property which is greater than or equal to 150 and less than or equal to 430.

ObjectKind <> 'Sheet Entry' && LocationX >= '250'
Returns all objects that have a X1 property which is greater than or equal to 250.

Not IsSheetEntry && LocationX < 350
Returns all objects that have a X1 property which is less than 350.

LocationX <> '210'
Returns all objects that have a X1 property that is not equal to 210, and all Sheet Entry objects. (Sheet Entry objects do not have a LocationX property, so do not have a LocationX property of 210, and are thus also returned by this Query.)

LocationX > '' && LocationX <> '210'
Returns all objects that have a X1 property which is not equal to 210.
**LocationY Field**

**Description**
Returns all objects having a Y1 property that complies with the Query.

**Syntax**

```
LocationY : Numeric_String
```

The LocationY keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all Sheet Entry objects (which do not have a Y1 property) have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:

- `LocationY <> ''`
- `LocationY > ''`
- `ObjectKind <> 'Sheet Entry'`
- `Not IsSheetEntry`

The usage of such tokens is demonstrated in the examples that follow.

**Examples**

```
LocationY = '320'
```

Returns all objects that have a Y1 property which is equal to 320.

```
LocationY > '' && LocationY = 320
```

```
Returns all objects that have a Y1 property which is greater than or equal to 320.
```

```
LocationY > '' && LocationY Between 150 And 430
```

Returns all objects that have a Y1 property which is greater than or equal to 150 and less than or equal to 430.

```
ObjectKind <> 'Sheet Entry' && LocationY >= 250
```

Returns all objects that have a Y1 property which is greater than or equal to 250.

```
Not IsSheetEntry && LocationY < 350
```

Returns all objects that have a Y1 property which is less than 350.

```
LocationY <> '210'
```

Returns all objects that have a Y1 property that is not equal to 210, and all Sheet Entry objects. (Sheet Entry objects do not have a LocationY property, so do not have a LocationY property of 210, and are thus also returned by this Query.)

```
LocationY > '' && LocationY <> 210
```

Returns all objects that have a Y1 property which is not equal to 210.

**Locked Field (Schematic)**

**Description**
Returns all Junction objects having a Locked property that complies with the Query. Note: The Locked property is only defined for Junction objects.

**Syntax**

```
Locked : Boolean_String
```

Boolean_String must be either 'True' or 'False'.

**Examples**

```
Locked = 'True'
```

Returns all Junction objects that have a 'True' Locked property.

```
Locked = 'False'
```

Returns all Junction objects that have a 'False' Locked property.

```
Locked <> 'True'
```

Not (Locked = 'True')

Returns all objects except Junction objects that have a 'True' Locked property.
Locked <> 'False'
Not (Locked = 'False')
Returns all objects except Junction objects that have a 'False' Locked property.

**LongStringText Field**

**Description**
Returns all Note and Text Frame objects having a Long Text property that complies with the Query.

*Note: The LongStringText property is only defined for Note and Text Frame objects.*

**Syntax**
LongStringText : String

**Examples**
LongStringText = 'NOTE: Fit battery after all other components.'
Returns all Note and Text Frame objects that have a Long Text property of 'NOTE: Fit battery after all other components.'.
LongStringText Like 'NOTE:*'
Returns all Note and Text Frame objects that have a Long Text property whose associated string starts with 'NOTE:'.

**Name Field (Schematic)**

**Description**
Returns all Parameter Set, Pin, Port, Probe, and Sheet Entry objects having a Name property that complies with the Query.

*Note: The Name property is only defined for Parameter Set, Pin, Port, Probe, and Sheet Entry objects.*

**Syntax**
Name : String

**Examples**
Name = 'W\R\'
Returns all Parameter Set, Pin, Port, Probe, and Sheet Entry objects that have a Name property of 'W\R\'.
Name Like 'INT?'
Returns all Parameter Set, Pin, Port, Probe, and Sheet Entry objects that have a Name property whose associated string commences with 'INT' and which contains one more following character; 'INTA' and 'INTB' are examples of thus compliant strings.

**Object_AreaColor Membership Check**

**Description**
Returns all child objects of all Port and Sheet Symbol objects having a Fill Color property that complies with the Query.

**Syntax**
Object_AreaColor(Parent) : Numeric_String

Object_AreaColor(Parent) : Number {If all parent objects which do not have a Fill Color property are totally excluded by one or more preceding tokens within the Query.}

The Object_AreaColor keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all parent objects which do not have a Fill Color property have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:

Object_AreaColor(Parent) <> ''
Object_AreaColor(Parent) > ''
Object_ObjectKind(Parent) = 'Port'
Object_ObjectKind(Parent) = 'Sheet Symbol'

The usage of such tokens is demonstrated in the examples that follow.
The color corresponding to a particular number can be deduced from the following relationship:
where the Blue, Green, and Red components each have a value between 0 and 255 (inclusive).

**When** the `Object_AreaColor` keyword can be used in conjunction with a Number (as previously described), it is additionally possible to individually evaluate the values of each of the Fill Color property's Blue, Green, and Red components, by using the following constructs:

- `Object_AreaColor(Parent) Div 65536 {Blue component}
- `Object_AreaColor(Parent) Div 256 Mod 256 {Green component}
- `Object_AreaColor(Parent) Mod 256 {Red component}

The usage of these constructs is also demonstrated in the examples that follow.

**Examples**

- `Object_AreaColor(Parent) = '0'
- `Object_AreaColor(Parent) <> '' && Object_AreaColor(Parent) = 0
  Returns all child objects of all Port and Sheet Symbol objects that have a black Fill Color property. (A Fill Color property of 0, as defined above, corresponds to values of Blue = 0, Green = 0, and Red = 0).

- `Object_AreaColor(Parent) = '8454016'
- `Object_AreaColor(Parent) > '' && Object_AreaColor(Parent) = 8454016
  Returns all child objects of all Port and Sheet Symbol objects that have a Fill Color property of 8454016 (as defined above, which corresponds to values of Blue = 128, Green = 255, and Red = 128).

- `Object_AreaColor(Parent) <> '16777215'
- `Object_AreaColor(Parent) > '' && Object_AreaColor(Parent) <> 16777215
  Returns all objects except child objects of Port and Sheet Symbol objects that have a (maximum intensity) white Fill Color property. (A Fill Color property of 16777215, as defined above, corresponds to values of Blue = 255, Green = 255, and Red = 255).

- `Object_AreaColor(Parent) > '' && Object_AreaColor(Parent) <> 16777215
  Returns all child objects of all Port and Sheet Symbol objects that have a Fill Color property which is not (maximum intensity) white.

- `Object_AreaColor(Parent) > '' && Object_AreaColor(Parent) Div 65536 >= 96 && Object_AreaColor(Parent) Div 256 Mod 256 Between 64 And 192 && Object_AreaColor(Parent) Mod 256 <= 128
  Returns all child objects of all Port and Sheet Symbol objects that have a Fill Color property whose Blue component has a value which is greater than or equal to 96, and whose Green component has a value which is greater than or equal to 64 and less than or equal to 192, and whose Red component has a value which is less than or equal to 128.

- `Object_ObjectKind(Parent) = 'Port' && Object_AreaColor(Parent) Div 65536 <= 192
  Returns all child objects of all Port objects whose Fill Color property's Blue component has a value which is less than or equal to 192.

- `Object_ObjectKind(Parent) = 'Sheet Symbol' && Object_AreaColor(Parent) Mod 256 Between 64 And 192
  Returns all child objects of all Sheet Symbol objects whose Fill Color property's Red component has a value which is greater than or equal to 64 and less than or equal to 192.
**Object_ArrowStyle Membership Check**

**Description**
Returns all child objects of all Port objects having a Port Style property that complies with the Query.

**Syntax**
- `Object_ArrowStyle(Parent) = PortStyle_String`
- `Object_ArrowStyle(Parent) <> PortStyle_String`

*PortStyle_String* must be one of the strings from the following list:
- 'Bottom'
- 'Left'
- 'Left & Right'
- 'None (Horizontal)'
- 'None (Vertical)'
- 'Right'
- 'Top'
- 'Top & Bottom'

Note: The single quote characters ('') shown at the start and end of each *PortStyle_String* are both mandatory.

**Examples**
- `Object_ArrowStyle(Parent) = 'Left'`
  Returns all child objects of all Port objects that have a 'Left' Port Style property.
- `Object_ArrowStyle(Parent) <> 'Bottom'`
  Returns all objects except child objects of Port objects that have a 'Bottom' Port Style property.
- `Object_ArrowStyle(Parent) <> '' && Object_ArrowStyle(Parent) <> 'Bottom'`
  Returns all child objects of all Port objects that do not have a 'Bottom' Port Style property.

**Object_Color Membership Check**

**Description**
Returns all child objects of all Parameter Set, Pin, Port, and Sheet Symbol objects having a Color property that complies with the Query.

**Syntax**
- `Object_Color(Parent) : Numeric_String`
- `Object_Color(Parent) : Number` (if all parent objects which do not have a Color property are totally excluded by one or more preceding tokens within the Query.)

The *Object_Color* keyword can always be used in conjunction with a *Numeric_String*; it can alternatively be used in conjunction with a *Number*, but only when all parent objects which do not have a Color property have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:
- `Object_Color(Parent) <> ''`
- `Object_Color(Parent) > ''`
- `Object_ObjectKind(Parent) = 'Parameter Set'`
- `Object_ObjectKind(Parent) = 'Port'`

The usage of such tokens is demonstrated in the examples that follow.

The color corresponding to a particular number can be deduced from the following relationship:

\[ 65536 \times \text{Blue} + 256 \times \text{Green} + \text{Red} \]

where the Blue, Green, and Red components each have a value between 0 and 255 (inclusive).

When the *Object_Color* keyword can be used in conjunction with a *Number* (as previously described), it is additionally possible to individually evaluate the values of each of the Color property's Blue, Green, and Red components, by using the following constructs:
- `Object_Color(Parent) Div 65536 {Blue component}`
- `Object_Color(Parent) Div 256 Mod 256 {Green component}`
- `Object_Color(Parent) Mod 256 {Red component}`

The usage of these constructs is also demonstrated in the examples that follow.
Examples

Object_Color(Parent) = '0'
Object_Color(Parent) <> '' && Object_Color(Parent) = 0

Returns all child objects of all Parameter Set, Pin, Port, and Sheet Symbol objects that have a black Color property. (A Color property of 0, as defined above, corresponds to values of Blue = 0, Green = 0, and Red = 0).

Object_Color(Parent) = '8388608'
Object_Color(Parent) > '' && Object_Color(Parent) = 8388608

Returns all child objects of all Parameter Set, Pin, Port, and Sheet Symbol objects that have a Color property of 8388608 (as defined above, which corresponds to values of Blue = 128, Green = 0, and Red = 0).

Object_Color(Parent) <> '16777215'

Returns all objects except child objects of Parameter Set, Pin, Port, and Sheet Symbol objects that have a (maximum intensity) white Color property. (A Color property of 16777215, as defined above, corresponds to values of Blue = 255, Green = 255, and Red = 255).

Object_Color(Parent) > '' && Object_Color(Parent) <> 16777215

Returns all child objects of all Parameter Set, Pin, Port, and Sheet Symbol objects that have a Color property which is not (maximum intensity) white.

Object_Color(Parent) > '' && Object_Color(Parent) Div 65536 >= 96 && Object_Color(Parent) Div 256 Mod 256 Between 64 And 192 && Object_Color(Parent) Mod 256 <= 128

Returns all child objects of all Parameter Set, Pin, Port, and Sheet Symbol objects that have a Color property whose Blue component has a value which is greater than or equal to 96, and whose Green component has a value which is greater than or equal to 64 and less than or equal to 192, and whose Red component has a value which is less than or equal to 128.

Object_ObjectKind(Parent) = 'Parameter Set' && Object_Color(Parent) Div 65536 <= 192

Returns all child objects of all Parameter Set objects whose Color property's Blue component has a value which is less than or equal to 192.

Object_ObjectKind(Parent) = 'Port' && Object_Color(Parent) Mod 256 Between 64 And 192

Returns all child objects of all Port objects whose Color property's Red component has a value which is greater than or equal to 64 and less than or equal to 192.

Object_Comment Membership Check

Description
Returns all child objects of all Part objects having a Part Comment property that complies with the Query.

Syntax
Object_Comment(Parent) : String

Example
Object_Comment(Parent) = '1488'

Returns all child objects of all Part objects that have a Part Comment property of '1488'.

Object_ComponentType Membership Check

Description
Returns all child objects of all Part objects having a Component Type property that complies with the Query.

Syntax
Object_ComponentType(Parent) = ComponentType_String
Object_ComponentType(Parent) <> ComponentType_String

ComponentType_String must be one of the strings from the following list:
'Graphical', 'Mechanical', 'Net Tie', 'Net Tie (In BOM)', 'Standard', 'Standard (No BOM)'

Note: The single quote characters ('') shown at the start and end of each ComponentType_String are both mandatory.

Examples
Object_ComponentType(Parent) = 'Net Tie (In BOM)'
Returns all child objects of all Part objects that have a 'Net Tie (In BOM)' Component Type property.

Object_ComponentType(Parent) <> 'Standard'
Returns all objects except child objects of Part objects that have a 'Standard' Component Type property.

Object_ComponentType(Parent) <> '' && Object_ComponentType(Parent) <> 'Standard'
Object_ComponentType(Parent) > '' && Object_ComponentType(Parent) <> 'Standard'
Object_ObjectKind(Parent) = 'Part' && Object_ComponentType(Parent) <> 'Standard'
Returns all child objects of all Part objects that do not have a 'Standard' Component Type property.

Object_CurrentFootprint Membership Check

Description
Returns all child objects of all Part objects having a Current Footprint property that complies with the Query.

Syntax
Object_CurrentFootprint(Parent) : String

Examples
Object_CurrentFootprint(Parent) = 'DIP14'
Returns all child objects of all Part objects that have a Current Footprint property of 'DIP14'.
Object_CurrentFootprint(Parent) Like 'SIP?*
Returns all child objects of all Part objects that have a Current Footprint property whose associated string starts with 'SIP', and which contains at least one more following character.

Object_Designator Membership Check

Description
Returns all child objects of all Part objects having a Component Designator property that complies with the Query.

Syntax
Object_Designator(Parent) : String

Examples
Object_Designator(Parent) = 'U4'
Returns all child objects of all Part objects that have a Component Designator property of 'U4'.
Object_Designator(Parent) Like 'R*'
Returns all child objects of all Part objects that have a Component Designator property whose associated string starts with 'R'; compliant examples are 'R1', 'R2', 'RA1', 'RV12', etc.

Object_DesignatorLocked Membership Check

Description
Returns all child objects of all Part objects having a Lock Designator property that complies with the Query.

Syntax
Object_DesignatorLocked(Parent) : Boolean_String

Boolean_String must be either 'True' or 'False'.

Examples
Object_DesignatorLocked(Parent) = 'True'
Returns all child objects of all Part objects that have a 'True' Lock Designator property.
Object_DesignatorLocked(Parent) = 'False'
Returns all child objects of all Part objects that have a 'False' Lock Designator property.
Object_DesignatorLocked(Parent) <> 'True'
Not (Object_DesignatorLocked(Parent) = 'True')

Returns all objects except child objects of Part objects that have a 'True' Lock Designator property.

Object_DesignatorLocked(Parent) <> 'False'
Not (Object_DesignatorLocked(Parent) = 'False')

Returns all objects except child objects of Part objects that have a 'False' Lock Designator property.

Object_ElectricalType Membership Check

Description
Returns all child objects of all Pin objects having an Electrical Type property that complies with the Query.

Syntax
Object_ElectricalType(Parent) = ElectricalType_String
Object_ElectricalType(Parent) <> ElectricalType_String

ElectricalType_String must be one of the strings from the following list:
'Emitter', 'HiZ', 'Input', 'IO', 'OpenCollector', 'Output', 'Passive', 'Power', 'VHDL-Buffer', 'VHDL-Port'

Note: The single quote characters (') shown at the start and end of each ElectricalType_String are both mandatory.

Examples
Object_ElectricalType(Parent) = 'Power'

Returns all child objects of all Pin objects that have a 'Power' Electrical Type property.

Object_ElectricalType(Parent) <> 'HiZ'

Returns all objects except child objects of Pin objects that have a 'HiZ' Electrical Type property.

Object_ElectricalType(Parent) <> '' && Object_ElectricalType(Parent) <> 'HiZ'
Object_ElectricalType(Parent) > '' && Object_ElectricalType(Parent) <> 'HiZ'
Object_ObjectKind(Parent) = 'Pin' && Object_ElectricalType(Parent) <> 'HiZ'

Returns all child objects of all Pin objects that do not have a 'HiZ' Electrical Type property.

Object_FileName Membership Check

Description
Returns all child objects of all Part objects having a File Name property that complies with the Query.

Syntax
Object_FileName(Parent) : String

Example
Object_FileName(Parent) = 'Address Decoder.pld'

Returns all child objects of all Part objects that have a File Name property of 'Address Decoder.pld'.
object_has_model membership check

description
returns all child objects of each part object that is linked to a model whose properties match those specified by the query.

each part can be linked to models of footprint, simulation, pcb3d, and signal integrity types. it is possible to link a part to more than one model of the same type, but only one model of each type can be selected as a part's current model.

the object_has_model keyword permits users to specify the type and name properties of a linked model, and optionally whether that model is (also) a current model of each part.

syntax
object_has_model(parent , modeltype : string , modelname : string , currentmodelonly : boolean) : boolean

the modeltype string specifies the type property of a linked model, and must be one of the strings from the following list:
'pcb3dlib' , 'pcblib' , 'si' , 'sim'

those strings respectively select linked models having type properties of pcb3d, footprint, signal integrity, and simulation.

the modelname string specifies the name property of a linked model.

the currentmodelonly parameter specifies whether or not a linked model (also) has to be a current model of each part. when this is false, a linked model does not have to be a current model (but it still can be); when this is true, a linked model does have to be a current model.

examples
object_has_model(parent , 'pcblib' , 'soic14' , false)
object_has_model(parent , 'pcblib' , 'soic14' , false) = true

returns all child objects of part objects that are linked to a footprint model which has a name property of 'soic14'; that model does not have to be each part's current footprint model (but it can be).

object_has_model(parent , 'pcblib' , 'dip14' , true)
object_has_model(parent , 'pcblib' , 'dip14' , true) = true

returns all child objects of part objects that are linked to a footprint model which has a name property of 'dip14', when that model is also the current footprint model of the part.

object_has_model(parent , 'si' , 'resi' , false)
object_has_model(parent , 'si' , 'resi' , false) = true

returns all child objects of part objects that are linked to a signal integrity model which has a name property of 'resi'; that model does not have to be each part's current signal integrity model (but it can be).

object_has_model(parent , 'sim' , 'resistor' , true)
object_has_model(parent , 'sim' , 'resistor' , true) = true

returns all child objects of part objects that are linked to a simulation model which has a name property of 'resistor', when that model is also the current simulation model of the part.

object_has_modelparameter membership check

description
returns all child objects of each part object that is linked to a model which contains a parameter whose properties match those specified by the query.

the object_has_modelparameter keyword permits users to specify the name and value properties of a parameter which is contained within a linked model, and optionally whether that model is a current model of each part. (each part can be linked to models of footprint, simulation, pcb3d, and signal integrity types. it is possible to link a part to more than one model of the same type, but only one model of each type can be selected as a part's current model.)

syntax
object_has_modelparameter(parent , parametername : string , parametervalue : string , currentmodelonly : boolean) : boolean

the parametername string specifies the parameter name property of the parameter contained in the linked model.
The ParameterValue string specifies the (Parameter) Value property of the Parameter contained in the linked Model. The CurrentModelOnly parameter specifies whether or not the linked Model (which contains a compliant Parameter) has to be a Current Model of each Part. When this is False, the linked Model does not have to be a Current Model (but it still can be); when this is True, the linked Model does have to be a Current Model.

**Examples**

Object_HasModelParameter(Parent,'Inductance A','1mH',False)
Object_HasModelParameter(Parent,'Inductance A','1mH',False) = True

Returns all child objects of Part objects that are linked to a Model which contains a Parameter that has a Parameter Name property of 'Inductance A' and a (Parameter) Value property of '1mH'; that Model does not have to be a Current Model of each Part (but it can be).

Object_HasModelParameter(Parent,'Coupling Factor','0.999',True)
Object_HasModelParameter(Parent,'Coupling Factor','0.999',True) = True

Returns all child objects of Part objects that are linked to a Model which contains a Parameter that has a Parameter Name property of 'Coupling Factor' and a (Parameter) Value property of '0.999', when that Model is also a Current Model of the Part.

### Object_HasParameter Membership Check

**Description**

Returns all child objects of all Parameter Set, Part, Pin, Port, Probe, and Sheet Symbol objects incorporating a Parameter object having a Parameter Name property and (Parameter) Value property that complies with the Query.

**Syntax**

Object_HasParameter(Parent , ParameterName : String , ParameterValue : String) : Boolean

The ParameterName string specifies the Parameter Name property of the Parameter.

The ParameterValue string specifies the (Parameter) Value property of the Parameter.

**Examples**

Object_HasParameter(Parent,'Comment','1k')
Object_HasParameter(Parent,'Comment','1k') = True

Returns all child objects of all Parameter Set, Part, Pin, Port, Probe, and Sheet Symbol objects incorporating a Parameter object that has a Parameter Name property of 'Comment' and a (Parameter) Value property of '1k'.

Object_HasParameter(Parent,'Text Field1','CAPACITOR')
Object_HasParameter(Parent,'Text Field1','CAPACITOR') = True

Returns all child objects of all Parameter Set, Part, Pin, Port, Probe, and Sheet Symbol objects incorporating a Parameter object that has a Parameter Name property of 'Text Field1' and a (Parameter) Value property of 'CAPACITOR'.

### Object_HasPin Membership Check

**Description**

Returns all child objects of all Part objects incorporating a Pin object having a Pin Designator property and Name property that complies with the Query.

**Syntax**

Object_HasPin(Parent , Pin Designator : String , Name : String) : Boolean

**Example**

Object_HasPin(Parent,'7','GND')
Object_HasPin(Parent,'7','GND') = True

Returns all child objects of all Part objects that contain a Pin object which has a Pin Designator property of '7' and a Name property of 'GND'.

Version (v1.7) Apr 21, 2008
Object_HasSheetEntry Membership Check

Description
Returns all child objects of all Sheet Symbol objects incorporating a Sheet Entry object having a Name property that complies with the Query.

Syntax
Object_HasSheetEntry(Parent, Sheet Entry Name : String) : Boolean

Example
Object_HasSheetEntry(Parent, 'BUS[0..7]')
Object_HasSheetEntry(Parent, 'BUS[0..7]') = True

Returns all child objects of all Sheet Symbol objects that contain a Sheet Entry object which has a Name property of 'BUS[0..7]'.

Object_IeeeSymbolInner Membership Check

Description
Returns all child objects of all Pin objects having an Ieee Symbol Inside property that complies with the Query.

Syntax
Object_IeeeSymbolInner(Parent) = IeeeSymbolInnerType_String
Object_IeeeSymbolInner(Parent) <> IeeeSymbolInnerType_String

IeeeSymbolInnerType_String must be one of the strings from the following list:

Note: The single quote characters ('') shown at the start and end of each IeeeSymbolInnerType_String are both mandatory.

Examples
Object_IeeeSymbolInner(Parent) = 'Schmitt'
Returns all child objects of all Pin objects that have a 'Schmitt' Ieee Symbol Inside property.
Object_IeeeSymbolInner(Parent) <> 'Pulse'
Returns all objects except child objects of Pin objects that have a 'Pulse' Ieee Symbol Inside property.
Object_IeeeSymbolInner(Parent) = '' && Object_IeeeSymbolInner(Parent) <> 'Pulse'
Object_IeeeSymbolInner(Parent) > '' && Object_IeeeSymbolInner(Parent) <> 'Pulse'
Object_ObjectKind(Parent) = 'Pin' && Object_IeeeSymbolInner(Parent) <> 'Pulse'

Returns all child objects of all Pin objects that do not have a 'Pulse' Ieee Symbol Inside property.

Object_IeeeSymbolInnerEdge Membership Check

Description
Returns all child objects of all Pin objects having an Ieee Symbol Inside Edge property that complies with the Query.

Syntax
Object_IeeeSymbolInnerEdge(Parent) = IeeeSymbolInnerEdgeType_String
Object_IeeeSymbolInnerEdge(Parent) <> IeeeSymbolInnerEdgeType_String

IeeeSymbolInnerEdgeType_String must be one of the strings from the following list:
'Clock', 'No Symbol'

Note: The single quote characters ('') shown at the start and end of each IeeeSymbolInnerEdgeType_String are both mandatory.

Examples
Object_IeeeSymbolInnerEdge(Parent) = 'Clock'
Returns all child objects of all Pin objects that have a 'Clock' Ieee Symbol Inside Edge property.
Object_IeeeSymbolInnerEdge(Parent) <> 'No Symbol'

Returns all objects except child objects of Pin objects that have a 'No Symbol' Ieee Symbol Inside Edge property.
Object_IeeeSymbolInnerEdge(Parent) <> '' && Object_IeeeSymbolInnerEdge(Parent) <> 'No Symbol'
Object_IeeeSymbolInnerEdge(Parent) > '' && Object_IeeeSymbolInnerEdge(Parent) <> 'No Symbol'
Object_ObjectKind(Parent) = 'Pin' && Object_IeeeSymbolInnerEdge(Parent) <> 'No Symbol'

Returns all child objects of all Pin objects that do not have a 'No Symbol' Ieee Symbol Inside Edge property.

Object_IeeeSymbolOuter Membership Check

Description
Returns all child objects of all Pin objects having an Ieee Symbol Outside property that complies with the Query.

Syntax
Object_IeeeSymbolOuter(Parent) = IeeeSymbolOuterType_String
Object_IeeeSymbolOuter(Parent) <> IeeeSymbolOuterType_String

IeeeSymbolOuterType_String must be one of the strings from the following list:

Note: The single quote characters (') shown at the start and end of each IeeeSymbolOuterType_String are both mandatory.

Examples
Object_IeeeSymbolOuter(Parent) = 'Bidirectional Signal Flow'
Returns all child objects of all Pin objects that have a 'Bidirectional Signal Flow' Ieee Symbol Outside property.
Object_IeeeSymbolOuter(Parent) <> 'Right Left Signal Flow'
Returns all objects except child objects of Pin objects that have a 'Right Left Signal Flow' Ieee Symbol Outside property.
Object_IeeeSymbolOuter(Parent) <> '' && Object_IeeeSymbolOuter(Parent) <> 'Right Left Signal Flow'
Object_IeeeSymbolOuter(Parent) > '' && Object_IeeeSymbolOuter(Parent) <> 'Right Left Signal Flow'
Object_ObjectKind(Parent) = 'Pin' && Object_IeeeSymbolOuter(Parent) <> 'Right Left Signal Flow'

Returns all child objects of all Pin objects that do not have a 'Right Left Signal Flow' Ieee Symbol Outside property.

Object_IeeeSymbolOuterEdge Membership Check

Description
Returns all child objects of all Pin objects having an Ieee Symbol Outside Edge property that complies with the Query.

Syntax
Object_IeeeSymbolOuterEdge(Parent) = IeeeSymbolOuterEdgeType_String
Object_IeeeSymbolOuterEdge(Parent) <> IeeeSymbolOuterEdgeType_String

IeeeSymbolOuterEdgeType_String must be one of the strings from the following list:
'Active Low Input' , 'Active Low Output' , 'Dot' , 'No Symbol'

Note: The single quote characters (') shown at the start and end of each IeeeSymbolOuterEdgeType_String are both mandatory.

Examples
Object_IeeeSymbolOuterEdge(Parent) = 'Dot'
Returns all child objects of all Pin objects that have a 'Dot' Ieee Symbol Outside Edge property.
Object_IeeeSymbolOuterEdge(Parent) <> 'Active Low Output'
Returns all objects except child objects of Pin objects that have an 'Active Low Output' Ieee Symbol Outside Edge property.
Object_IeeeSymbolOuterEdge(Parent) <> '' && Object_IeeeSymbolOuterEdge(Parent) <> 'Active Low Output'
Object_IeeeSymbolOuterEdge(Parent) > '' && Object_IeeeSymbolOuterEdge(Parent) <> 'Active Low Output'
Object_ObjectKind(Parent) = 'Pin' && Object_IeeeSymbolOuterEdge(Parent) <> 'Active Low Output'

Returns all child objects of all Pin objects that do not have an 'Active Low Output' Ieee Symbol Outside Edge property.

Object_IOType Membership Check

Description
Returns all child objects of all Port objects having an IO Type property that complies with the Query.

Syntax
Object_IOType(Parent) = IOType_String
Object_IOType(Parent) <> IOType_String

IOType_String must be one of the strings from the following list:
'Bidirectional', 'Input', 'Output', 'Unspecified'

Note: The single quote characters ('') shown at the start and end of each IOType_String are both mandatory.

Examples
Object_IOType(Parent) = 'Input'
Returns all child objects of all Port objects that have an 'Input' IO Type property.
Object_IOType(Parent) <> 'Output'
Returns all objects except child objects of Port objects that have an 'Output' IO Type property.
Object_IOType(Parent) > '' && Object_IOType(Parent) <> 'Output'
Object_IOType(Parent) > '' && Object_IOType(Parent) <> 'Output'
Object_ObjectKind(Parent) = 'Port' && Object_IOType(Parent) <> 'Output'

Returns all child objects of all Port objects that do not have an 'Output' IO Type property.

Object_IsHidden Membership Check

Description
Returns all child objects of all Pin objects having a Hide property that complies with the Query.

Syntax
Object_IsHidden(Parent) : Boolean

Examples
Object_IsHidden(Parent)
Object_IsHidden(Parent) = True
Returns all child objects of all Pin objects that have a True Hide property.
Not Object_IsHidden(Parent)
Object_IsHidden(Parent) = False
Returns all objects except child objects of Pin objects that have a True Hide property.
Object_ObjectKind(Parent) = 'Pin' && Not Object_IsHidden(Parent)
Returns all child objects of all Pin objects that have a False Hide property.

Object_IsMirrored Membership Check

Description
Returns all child objects of all Part objects having a Mirrored property that complies with the Query.

Syntax
Object_IsMirrored(Parent) : Boolean_String

Boolean_String must be either 'True' or 'False'.
Examples

Object_IsMirrored(Parent) = 'True'
Returns all child objects of all Part objects that have a 'True' Mirrored property.

Object_IsMirrored(Parent) = 'False'
Returns all child objects of all Part objects that have a 'False' Mirrored property.

Object_IsMirrored(Parent) <> 'True'
Not (Object_IsMirrored(Parent) = 'True')
Returns all objects except child objects of Part objects that have a 'True' Mirrored property.

Object_IsMirrored(Parent) <> 'False'
Not (Object_IsMirrored(Parent) = 'False')
Returns all objects except child objects of Part objects that have a 'False' Mirrored property.

Object_Length Membership Check

Description
Returns all child objects of all Pin objects having a Length property that complies with the Query.

Syntax
Object_Length(Parent) : Numeric_String
Object_Length(Parent) : Number {If all parent objects which do not have a Length property are totally excluded by one or more preceding tokens within the Query.}

The Object_Length keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all parent objects which do not have a Length property have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:

Object_Length(Parent) <> ''
Object_Length(Parent) > ''
Object_ObjectKind(Parent) = 'Pin'

The usage of such tokens is demonstrated in the examples that follow.

Examples

Object_Length(Parent) = '30'
Object_Length(Parent) <> '' && Object_Length(Parent) = 30
Returns all child objects of all Pin objects that have a Length property which is equal to 30.

Object_Length(Parent) > '' && Object_Length(Parent) Between 20 And 40
Returns all child objects of all Pin objects that have a Length property which is greater than or equal to 20 and less than or equal to 40.

Object_ObjectKind(Parent) = 'Pin' && Object_Length(Parent) >= 10
Returns all child objects of all Pin objects that have a Length property which is greater than or equal to 10.

Object_Length(Parent) > '' && Object_Length(Parent) < 30
Returns all child objects of all Pin objects that have a Length property which is less than 30.

Object_Length(Parent) <> '40'
Returns all objects except child objects of Pin objects that have a Length property which is equal to 40.

Object_Length(Parent) > '' && Object_Length(Parent) <> 40
Returns all child objects of all Pin objects that have a Length property which is not equal to 40.
Object_LibraryName Membership Check

Description
Returns all child objects of all Part objects having a Library Name property that complies with the Query.

Syntax
Object_LibraryName(Parent) : String

Example
Object_LibraryName(Parent) = 'Miscellaneous Devices.IntLib'
Returns all child objects of all Part objects that have a Library Name property of 'Miscellaneous Devices.IntLib'.

Object_LibReference Membership Check

Description
Returns all child objects of all Part objects having a Library Reference property that complies with the Query.

Syntax
Object_LibReference(Parent) : String

Example
Object_LibReference(Parent) = '1488_1'
Returns all child objects of all Part objects that have a Library Reference property of '1488_1'.

Object_LocationX Membership Check

Description
Returns all child objects of all Parameter Set, Part, Pin, Port, Probe, and Sheet Symbol objects having a X1 property that complies with the Query.

Syntax
Object_LocationX(Parent) : Numeric_String
Object_LocationX(Parent) : Number {If all parent objects which do not have a X1 property are totally excluded by one or more preceding tokens within the Query.}
The Object_LocationX keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all parent objects which do not have a X1 property have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:
Object_LocationX(Parent) <> ''
Object_LocationX(Parent) > ''
Object_ObjectKind(Parent) <> 'Sheet Entry'
Object_ObjectKind(Parent) = 'Part'
The usage of such tokens is demonstrated in the examples that follow.

Examples
Object_LocationX(Parent) = '320'
Object_LocationX(Parent) <> '' && Object_LocationX(Parent) = 320
Returns all child objects of all Parameter Set, Part, Pin, Port, Probe, and Sheet Symbol objects that have a X1 property which is equal to 320.
Object_LocationX(Parent) > '' && Object_LocationX(Parent) Between 150 And 430
Returns all child objects of all Parameter Set, Part, Pin, Port, Probe, and Sheet Symbol objects that have a X1 property which is greater than or equal to 150 and less than or equal to 430.
Object_ObjectKind(Parent) <> 'Sheet Entry' && Object_LocationX(Parent) >= 250
Returns all child objects of all Parameter Set, Part, Pin, Port, Probe, and Sheet Symbol objects that have a X1 property which is greater than or equal to 250.
Object_ObjectKind(Parent) = 'Part' & Object_LocationX(Parent) < 350
Returns all child objects of all Part objects that have a X1 property which is less than 350.

Object_LocationX(Parent) <> '210'
Returns all objects except child objects of Parameter Set, Part, Pin, Port, Probe, and Sheet Symbol objects that have a X1 property which is equal to 210.

Object_LocationX(Parent) > '' && Object_LocationX(Parent) <> 210
Returns all child objects of all Parameter Set, Part, Pin, Port, Probe, and Sheet Symbol objects that have a X1 property which is not equal to 210.

Object_LocationY Membership Check

Description
Returns all child objects of all Parameter Set, Part, Pin, Port, Probe, and Sheet Symbol objects having a Y1 property that complies with the Query.

Syntax
Object_LocationY(Parent) : Numeric_String
Object_LocationY(Parent) : Number (If all parent objects which do not have a Y1 property are totally excluded by one or more preceding tokens within the Query.)
The Object_LocationY keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all parent objects which do not have a Y1 property have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:
Object_LocationY(Parent) <> ''
Object_LocationY(Parent) > ''
Object_ObjectKind(Parent) <> 'Sheet Entry'
Object_ObjectKind(Parent) = 'Part'
The usage of such tokens is demonstrated in the examples that follow.

Examples
Object_LocationY(Parent) = '320'
Object_LocationY(Parent) <> '' && Object_LocationY(Parent) = 320
Returns all child objects of all Parameter Set, Part, Pin, Port, Probe, and Sheet Symbol objects that have a Y1 property which is equal to 320.

Object_LocationY(Parent) > '' && Object_LocationY(Parent) Between 150 And 430
Returns all child objects of all Parameter Set, Part, Pin, Port, Probe, and Sheet Symbol objects that have a Y1 property which is greater than or equal to 150 and less than or equal to 430.

Object_ObjectKind(Parent) <> 'Sheet Entry' && Object_LocationY(Parent) >= 250
Returns all child objects of all Parameter Set, Part, Pin, Port, Probe, and Sheet Symbol objects that have a Y1 property which is greater than or equal to 250.

Object_ObjectKind(Parent) = 'Part' && Object_LocationY(Parent) < 350
Returns all child objects of all Part objects that have a Y1 property which is less than 350.

Object_LocationY(Parent) <> '210'
Returns all objects except child objects of Parameter Set, Part, Pin, Port, Probe, and Sheet Symbol objects that have a Y1 property which is equal to 210.

Object_LocationY(Parent) > '' && Object_LocationY(Parent) <> 210
Returns all child objects of all Parameter Set, Part, Pin, Port, Probe, and Sheet Symbol objects that have a Y1 property which is not equal to 210.
Object_ObjectKind Membership Check

Description
Returns all child objects of all objects having an Object Kind property matching that specified by the Query.

Syntax
\[
\text{Object\_ObjectKind(Parent) = ObjectKind\_String}\\
\text{Object\_ObjectKind(Parent) <> ObjectKind\_String}
\]

ObjectKind\_String must be one of the strings from the following list:
'Parameter Set', 'Part', 'Pin', 'Port', 'Probe', 'Sheet Symbol', ''

Note: The single quote characters (') shown at the start and end of each ObjectKind\_String are both mandatory; the last string listed is a "null" string, with no intermediate characters between the (opening and closing) single quote characters.

(To return any objects, the type of object specified needs to be one that can have child objects; if a "null" string is specified, all objects which do not have a parent object are returned.)

Examples
\[
\text{Object\_ObjectKind(Parent) = 'Sheet Symbol'}\\
\text{Returns all child objects of all Sheet Symbol objects.}\\
\text{Object\_ObjectKind(Parent) = '}'\\
\text{Returns all objects which do not have a parent object; those objects include (but are not confined to) Document Parameters.}\\
\text{Object\_ObjectKind(Parent) <> 'Port'}\\
\text{Returns all objects except child objects of Port objects.}
\]

Object_Orientation Membership Check

Description
Returns all child objects of all Parameter Set, Part, Pin, and Probe objects having an Orientation property that complies with the Query.

Syntax
\[
\text{Object\_Orientation(Parent) = Orientation\_String}\\
\text{Object\_Orientation(Parent) <> Orientation\_String}
\]

Orientation\_String must be one of the strings from the following list:
'0 Degrees', '90 Degrees', '180 Degrees', '270 Degrees'

Note: The single quote characters (') shown at the start and end of each Orientation\_String are both mandatory.

Examples
\[
\text{Object\_Orientation(Parent) = '90 Degrees'}\\
\text{Returns all child objects of all Parameter Set, Part, Pin, and Probe objects that have a '90 Degrees' Orientation property.}\\
\text{Object\_Orientation(Parent) <> '180 Degrees'}\\
\text{Returns all objects except child objects of Parameter Set, Part, Pin, and Probe objects that have a '180 Degrees' Orientation property.}\\
\text{Object\_Orientation(Parent) <> '' && Object\_Orientation(Paren}\\
\text{t) <> '180 Degrees'}\\
\text{Returns all child objects of all Parameter Set, Part, Pin, and Probe objects that do not have a '180 Degrees' Orientation property.}
\]
Object_PartId Membership Check

Description
Returns all child objects of all (multiple-part) Part objects having a Current Part property that complies with the Query.

Note: Part objects are either of a "single-part" or "multiple-part" nature. The Current Part property is only defined for Part objects of a multiple-part nature, and is not defined for Part objects of a single-part nature.

Syntax
Object_PartId(Parent) : String

Examples
Object_PartId(Parent) = 'A'
Returns all child objects of all (multiple-part) Part objects that have a Current Part property of 'A'.

Object_PartId(Parent) <> 'B'
Returns all objects except child objects of (multiple-part) Part objects that have a Current Part property of 'B'.

Object_PartId(Parent) <> '' && Object_PartId(Parent) <> 'B'
Returns all child objects of all multiple-part Part objects that do not have a Current Part property of 'B'.

Object_PartId(Parent) <> ''
Object_PartId(Parent) Like '?*'
Returns all child objects of all multiple-part Part objects. (Such Part objects do not have a Current Part property of ".")

Object_PinDefaultValue Membership Check

Description
Returns all Pin objects having a Pin Default Value property that complies with the Query.

Syntax
Object_PinDefaultValue = Value_String

Note: The single quote characters (') shown at the start and end of each Alignment_String are both mandatory.

Examples
Object_PinDefaultValue = Value_String
Returns all Pin objects that have a Pin Default Value property.

Object_PinShowDesignator Membership Check

Description
Returns all child objects of all Pin objects having a Show Designator property that complies with the Query.

Syntax
Object_PinShowDesignator(Parent) : Boolean_String

Boolean_String must be either 'True' or 'False'.

Examples
Object_PinShowDesignator(Parent) = 'True'
Returns all child objects of all Pin objects that have a 'True' Show Designator property.

Object_PinShowDesignator(Parent) = 'False'
Returns all child objects of all Pin objects that have a 'False' Show Designator property.

Object_PinShowDesignator(Parent) <> 'True'
Not (Object_PinShowDesignator(Parent) = 'True')
Returns all objects except child objects of Pin objects that have a 'True' Show Designator property.

Object_PinShowDesignator(Parent) <> 'False'
Not (Object_PinShowDesignator(Parent) = 'False')
Returns all objects except child objects of Pin objects that have a 'False' Show Designator property.
Returns all objects except child objects of Pin objects that have a 'False' Show Designator property.

### Object_PinsLocked Membership Check

**Description**
Returns all child objects of all Part objects having a Pins Locked property that complies with the Query.

**Syntax**

```
Object_PinsLocked(Parent) : Boolean_String
```

Boolean_String must be either 'True' or 'False'.

**Examples**

- `Object_PinsLocked(Parent) = 'True'`
  Returns all child objects of all Part objects that have a 'True' Pins Locked property.

- `Object_PinsLocked(Parent) = 'False'`
  Returns all child objects of all Part objects that have a 'False' Pins Locked property.

- `Object_PinsLocked(Parent) <> 'True'`
  `Not (Object_PinsLocked(Parent) = 'True')`
  Returns all objects except child objects of Part objects that have a 'True' Pins Locked property.

- `Object_PinsLocked(Parent) <> 'False'`
  `Not (Object_PinsLocked(Parent) = 'False')`
  Returns all objects except child objects of Part objects that have a 'False' Pins Locked property.

### Object_SheetFileName Membership Check

**Description**
Returns all child objects of all Sheet Symbol objects having a Sheet File Name property that complies with the Query.

**Syntax**

```
Object_SheetFileName(Parent) : String
```

**Example**

```
Object_SheetFileName(Parent) = 'Screen Controller Interface.SchDoc'
```

Returns all child objects of all Sheet Symbol objects that have a Sheet File Name property of 'Screen Controller Interface.SchDoc'.

### Object_SheetName Membership Check

**Description**
Returns all child objects of all Sheet Symbol objects having a Sheet Name property that complies with the Query.

**Syntax**

```
Object_SheetName(Parent) : String
```

**Example**

```
Object_SheetName(Parent) = 'Screen Controller Interface'
```

Returns all child objects of all Sheet Symbol objects that have a Sheet Name property of 'Screen Controller Interface'.

### Object_ShowHiddenPins Membership Check

**Description**
Returns all child objects of all Part objects having a Show Hidden Pins property that complies with the Query.

**Syntax**

```
Object_ShowHiddenPins(Parent) : Boolean_String
```
Boolean_String must be either 'True' or 'False'.

Examples

Object_ShowHiddenPins(Parent) = 'True'
Returns all child objects of all Part objects that have a 'True' Show Hidden Pins property.

Object_ShowHiddenPins(Parent) = 'False'
Returns all child objects of all Part objects that have a 'False' Show Hidden Pins property.

Object_ShowHiddenPins(Parent) <> 'True'
Not (Object_ShowHiddenPins(Parent) = 'True')
Returns all objects except child objects of Part objects that have a 'True' Show Hidden Pins property.

Object_ShowHiddenPins(Parent) <> 'False'
Not (Object_ShowHiddenPins(Parent) = 'False')
Returns all objects except child objects of Part objects that have a 'False' Show Hidden Pins property.

Object_ShowName Membership Check

Description
Returns all child objects of all Pin objects having a Show Name property that complies with the Query.

Syntax

Object_ShowName(Parent) : Boolean_String

Boolean_String must be either 'True' or 'False'.

Examples

Object_ShowName(Parent) = 'True'
Returns all child objects of all Pin objects that have a 'True' Show Name property.

Object_ShowName(Parent) = 'False'
Returns all child objects of all Pin objects that have a 'False' Show Name property.

Object_ShowName(Parent) <> 'True'
Not (Object_ShowName(Parent) = 'True')
Returns all objects except child objects of Pin objects that have a 'True' Show Name property.

Object_ShowName(Parent) <> 'False'
Not (Object_ShowName(Parent) = 'False')
Returns all objects except child objects of Pin objects that have a 'False' Show Name property.

Object_TargetFileName Membership Check

Description
Returns all component objects having a Target File Name property that complies with the Query.

Syntax

Object_TargetFileName(Parent) : String

Example

Object_TargetFileName(Parent) = '8051'
Returns all component objects that have a Target File Name property of '8051'.

Object_TextColor Membership Check

Description
Returns all child objects of all Port objects having a Text Color property that complies with the Query.

Syntax
Object_TextColor(Parent) : Numeric_String

Object_TextColor(Parent) : Number (If all parent objects which do not have a Text Color property are totally excluded by one or more preceding tokens within the Query.)

The Object_TextColor keyword can **always** be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but **only** when all parent objects which do not have a Text Color property have been **totally** excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:

Object_TextColor(Parent) <> ''
Object_TextColor(Parent) > ''
Object_ObjectKind(Parent) = 'Port'

The usage of such tokens is demonstrated in the examples that follow.

The color corresponding to a particular number can be deduced from the following relationship:

\[ 65536 \times \text{Blue} + 256 \times \text{Green} + \text{Red} \]

where the Blue, Green, and Red components each have a value between 0 and 255 (inclusive).

**When** the Object_TextColor keyword can be used in conjunction with a Number (as previously described), it is additionally possible to individually evaluate the values of each of the Text Color property's Blue, Green, and Red components, by using the following constructs:

Object_TextColor(Parent) Div 65536 \{Blue component\}
Object_TextColor(Parent) Div 256 Mod 256 \{Green component\}
Object_TextColor(Parent) Mod 256 \{Red component\}

The usage of these constructs is also demonstrated in the examples that follow.

**Examples**

Object_TextColor(Parent) = '0'
Object_TextColor(Parent) <> '' && Object_TextColor(Parent) = 0

Returns all child objects of all Port objects that have a black Text Color property. (A Text Color property of 0, as defined above, corresponds to values of Blue = 0, Green = 0, and Red = 0).

Object_TextColor(Parent) = '128'
Object_TextColor(Parent) > '' && Object_TextColor(Parent) = 128

Returns all child objects of all Port objects that have a Text Color property of 128 (as defined above, which corresponds to values of Blue = 0, Green = 0, and Red = 128).

Object_TextColor(Parent) <> '16777215'

Returns all objects except child objects of Port objects that have a (maximum intensity) white Text Color property. (A Text Color property of 16777215, as defined above, corresponds to values of Blue = 255, Green = 255, and Red = 255).

Object_TextColor(Parent) > '' && Object_TextColor(Parent) <> 16777215

Returns all child objects of all Port objects that have a Text Color property which is **not** (maximum intensity) white.

Object_TextColor(Parent) > '' && Object_TextColor(Parent) Div 65536 >= 96 &&
Object_TextColor(Parent) Div 256 Mod 256 Between 64 And 192 && Object_TextColor(Parent) Mod 256 <= 128

Returns all child objects of all Port objects that have a Text Color property whose Blue component has a value which is greater than or equal to 96, and whose Green component has a value which is greater than or equal to 64 and less than or equal to 192, and whose Red component has a value which is less than or equal to 128.

Object_ObjectKind(Parent) = 'Port' && Object_TextColor(Parent) Div 65536 <= 192

Returns all child objects of all Port objects whose Text Color property's Blue component has a value which is less than or equal to 96.

Object_ObjectKind(Parent) = 'Port' && Object_TextColor(Parent) Mod 256 Between 64 And 192

Returns all child objects of all Port objects whose Text Color property's Red component has a value which is greater than or equal to 64 and less than or equal to 128.
**ObjectKind Field (Schematic)**

**Description**
Returns all objects having an Object Kind property matching that specified by the Query.

**Syntax**

```
ObjectKind = ObjectKind_String  
ObjectKind <> ObjectKind_String  
```

ObjectKind_String must be one of the strings from the following list:


Note: The single quote characters (') shown at the start and end of each ObjectKind_String are both mandatory.

**Examples**

```
ObjectKind = 'Bus'  
Returns all Bus objects.  
ObjectKind <> 'Pie'  
Returns all objects except for Pies.  
ObjectKind = 'Line'  
Returns all Line objects, of both a Line nature and a Polyline nature.  
(Line objects can be of either a Line nature or a Polyline nature; the former have just two vertices, and only exist in files which were created using earlier versions of Protel. The latter can have more than two vertices, and are always created whenever the 'Place Line' command is used in Altium Designer.)
```

**OffSheetStyle Field**

**Description**
Returns all Off Sheet Connector objects having an Off Sheet Style property that complies with the Query.

Note: The OffSheetStyle property is only defined for Off Sheet Connector objects.

**Syntax**

```
OffSheetStyle = OffSheetStyle_String  
OffSheetStyle <> OffSheetStyle_String  
```

OffSheetStyle_String must be one of the strings from the following list:

- 'Left', 'Right'

Note: The single quote characters (') shown at the start and end of each OffSheetStyle_String are both mandatory.

**Examples**

```
OffSheetStyle = 'Right'  
Returns all Off Sheet Connector objects that have a 'Right' Off Sheet Style property.  
OffSheetStyle <> 'Left'  
Returns all objects except Off Sheet Connector objects that have a 'Left' Off Sheet Style property. (Only Off Sheet Connector objects have an OffSheetStyle property, so all remaining types of objects do not have a 'Left' OffSheetStyle property, and are thus also returned by this Query.)
```
Orientation Field

Description
Returns all Designator, IEEE Symbol, Net Label, Off Sheet Connector, Parameter, Parameter Set, Part, Pin, Power Object, Probe, Sheet File Name, Sheet Name, and Text String objects having an Orientation property that complies with the Query.
Note: The Orientation property is only defined for Designator, IEEE Symbol, Net Label, Off Sheet Connector, Parameter, Parameter Set, Part, Pin, Power Object, Probe, Sheet File Name, Sheet Name, and Text String objects.

Syntax
Orientation = Orientation_String
Orientation <> Orientation_String

Orientation_String must be one of the strings from the following list:
'0 Degrees', '90 Degrees', '180 Degrees', '270 Degrees'
Note: The single quote characters (') shown at the start and end of each Orientation_String are both mandatory.

Examples
Orientation = '90 Degrees'
Returns all Designator, IEEE Symbol, Net Label, Off Sheet Connector, Parameter, Parameter Set, Part, Pin, Power Object, Probe, Sheet File Name, Sheet Name, and Text String objects that have a '90 Degrees' Orientation property.
Orientation <> '180 Degrees'
Returns all objects except Designator, IEEE Symbol, Net Label, Off Sheet Connector, Parameter, Parameter Set, Part, Pin, Power Object, Probe, Sheet File Name, Sheet Name, and Text String objects that have a '180 Degrees' Orientation property.
(Only Designator, IEEE Symbol, Net Label, Off Sheet Connector, Parameter, Parameter Set, Part, Pin, Power Object, Probe, Sheet File Name, Sheet Name, and Text String objects have an Orientation property, so all remaining types of objects do not have a '180 Degrees' Orientation property, and are thus also returned by this Query.)
Orientation <> ' ' && Orientation <> '180 Degrees'
Orientation > ' ' && Orientation <> '180 Degrees'
Returns all Designator, IEEE Symbol, Net Label, Off Sheet Connector, Parameter, Parameter Set, Part, Pin, Power Object, Probe, Sheet File Name, Sheet Name, and Text String objects that do not have a '180 Degrees' Orientation property.

OwnerDocument Field

Description
Returns all objects which are contained in the file (the Owner Document) that is specified by the Query.

Syntax
OwnerDocument : String

Example
OwnerDocument = 'Screen Memory.SchDoc'
Returns all objects which are contained in the file that has a name of 'Screen Memory.SchDoc'.
OwnerName Field

Description
Returns all Designator, Parameter, Pin, Sheet Entry, Sheet File Name, and Sheet Name objects having an Owner property that complies with the Query.

Note: The OwnerName property is only defined for Designator, Parameter, Pin, Sheet Entry, Sheet File Name, and Sheet Name objects.

Syntax
OwnerName : String

Examples
OwnerName = 'U3'
Returns all Designator, Parameter, Pin, Sheet Entry, Sheet File Name, and Sheet Name objects that have an Owner property of 'U3'.

OwnerName Like 'U?*'
Returns all Designator, Parameter, Pin, Sheet Entry, Sheet File Name, and Sheet Name objects that have an Owner property whose associated string starts with 'U', and which also contains at least one more following character.

OwnerPartDisplayMode Field

Description
Returns all objects having an OwnerPartDisplayMode property that complies with the Query. Components have up to 255 display modes and the mode 0 is the normal mode and the rest are alternative display modes.

Syntax
OwnerPartDisplayMode = Mode_String

Where Mode_String is a string enclosed with single quotes.

Example
OwnerPartDisplayMode = 'Normal'
Fetches only those components that are in Normal mode.

OwnerPartDisplayMode = 'Alternate 4'
Fetches only those components that are in the Alternate 4 mode.

OwnerPartId Field

Description
Returns all objects having an OwnerPartId property that complies with the Query.

Syntax
OwnerPartId = Id_String

Where Id_String is a numerical string enclosed with single quotes. For example '3'.

Example
OwnerPartId = '3'
Fetches only those multi-part components that have the Part Id of 3.

ParameterAllowDatabaseSynchronize Field

Description
Returns all Parameter objects having an Allow Database Synchronize property that complies with the Query.

Note: The ParameterAllowDatabaseSynchronize property is only defined for Parameter objects.

Syntax
ParameterAllowDatabaseSynchronize : Boolean_String

Boolean_String must be either 'True' or 'False'.

Examples
ParameterAllowDatabaseSynchronize = 'True'
Returns all Parameter objects that have a 'True' Allow Database Synchronize property.
ParameterAllowDatabaseSynchronize = 'False'
Returns all Parameter objects that have a 'False' Allow Database Synchronize property.
ParameterAllowDatabaseSynchronize <> 'True'
Not (ParameterAllowDatabaseSynchronize = 'True')
Returns all objects except Parameter objects that have a 'True' Allow Database Synchronize property.
ParameterAllowDatabaseSynchronize <> 'False'
Not (ParameterAllowDatabaseSynchronize = 'False')
Returns all objects except Parameter objects that have a 'False' Allow Database Synchronize property.

ParameterAllowLibrarySynchronize Field

Description
Returns all Parameter objects having an Allow Library Synchronize property that complies with the Query.
Note: The ParameterAllowLibrarySynchronize property is only defined for Parameter objects.

Syntax
ParameterAllowLibrarySynchronize : Boolean_String

Boolean_String must be either 'True' or 'False'.

Examples
ParameterAllowLibrarySynchronize = 'True'
Returns all Parameter objects that have a 'True' Allow Library Synchronize property.
ParameterAllowLibrarySynchronize = 'False'
Returns all Parameter objects that have a 'False' Allow Library Synchronize property.
ParameterAllowLibrarySynchronize <> 'True'
Not (ParameterAllowLibrarySynchronize = 'True')
Returns all objects except Parameter objects that have a 'True' Allow Library Synchronize property.
ParameterAllowLibrarySynchronize <> 'False'
Not (ParameterAllowLibrarySynchronize = 'False')
Returns all objects except Parameter objects that have a 'False' Allow Library Synchronize property.

ParameterAutoposition Field

Description
Returns all Parameter objects having an Autoposition property that complies with the Query.
Note: The ParameterAutoposition property is only defined for Parameter objects.

Syntax
ParameterAutoposition : Boolean_String

Boolean_String must be either 'True' or 'False'.

Examples
ParameterAutoposition = 'True'
Returns all Parameter objects that have a 'True' Autoposition property.
ParameterAutoposition = 'False'
Returns all Parameter objects that have a 'False' Autoposition property.
ParameterAutoposition <> 'True'
Not (ParameterAutoposition = 'True')
Returns all objects except Parameter objects that have a 'True' Autoposition property.

ParameterAutoposition <> 'False'
Not (ParameterAutoposition = 'False')
Returns all objects except Parameter objects that have a 'False' Autoposition property.

ParameterName Field

Description
Returns all Parameter objects having a Parameter Name property that complies with the Query.
Note: The ParameterName property is only defined for Parameter objects.

Syntax
ParameterName : String

Examples
ParameterName = 'Comment'
Returns all Parameter objects that have a Parameter Name property of 'Comment'.

ParameterReadOnlyState Field

Description
Returns all Parameter objects having a ParameterReadOnlyState property that complies with the Query.
Note: The ParameterReadOnlyState property is only defined for Parameter objects.

Syntax
This is not implemented.

Examples
N/A

ParametersList Field

Description
Returns all Parameter Set, Pin, Port, Probe, and Sheet Symbol objects having a Parameters List property that complies with the Query.
Note: The ParametersList property is only defined for Parameter Set, Pin, Port, Probe, and Sheet Symbol objects.

Syntax
ParametersList : String

Example
ParametersList = 'Manufacturer=Motorola, Supplier=Dick Smith Electronics'
Returns all Parameter Set, Pin, Port, Probe, and Sheet Symbol objects that have two parameters, when the first parameter has a Name property of 'Manufacturer' and a Value property of 'Motorola', and the second parameter has a Name property of 'Supplier' and a Value property of 'Dick Smith Electronics'.

ParameterType Field

Description
Returns all Parameter objects having a Type property that complies with the Query.
Note: The ParameterType property is only defined for Parameter objects.

**Syntax**

ParameterType = ParameterType_String
ParameterType <> ParameterType_String

ParameterType_String must be one of the strings from the following list:
'BOOLEAN', 'FLOAT', 'INTEGER', 'STRING'

Note: The single quote characters (') shown at the start and end of each ParameterType_String are both mandatory.

**Examples**

ParameterType = 'STRING'

Returns all Parameter objects that have a 'STRING' Type property.

ParameterType <> 'FLOAT'

Returns all objects except Parameter objects that have a 'FLOAT' Type property. (Only Parameter objects have a ParameterType property, so all remaining types of objects do not have a 'FLOAT' ParameterType property, and are thus also returned by this Query.)

**ParameterValue Field**

**Description**

Returns all Parameter objects having a Value property that complies with the Query.

Note: The ParameterValue property is only defined for Parameter objects.

**Syntax**

ParameterValue : String

**Examples**

ParameterValue = '100nF'

Returns all Parameter objects that have a Value property of '100nF'.

ParameterValue Like '*nF'

Returns all Parameter objects that have a Value property whose associated string concludes with 'nF'; '22nF' and '100nF' are examples of thus compliant strings.

**Parent Field**

**Description**

This keyword is used in conjunction with the majority of the "Membership Check" keywords; it then signifies that each object which is returned by the associated Query is returned because of a property which is held by its parent object.

**Syntax**

This depends upon which "Membership Check" keyword that the Parent keyword is used in conjunction with, but it is always the first (and often the only) parameter to be specified for the "Membership Check" keyword concerned.

**Examples**

Object_ObjectKind(Parent) = 'Part' && Object_PartId(Parent) = ''

Returns all child objects of all single-part Part objects. (All such Part objects have a Current Part property of ".")

Object_HasModel(Parent,'PCBLib','DIP14',False) && Object_HasPin(Parent,'14','VCC')

Returns all child objects of all Part objects that are linked to a Footprint Model which has a Name property of 'DIP14', when the Part object also contains a Pin object which has a Pin Designator property of '14' and a Name property of 'VCC'.

Object_Color(Parent) > '' && Object_Color(Parent) Div 65536 >= 96 && Object_Color(Parent) Div 256 Mod 256 Between 64 And 192 && Object_Color(Parent) Mod 256 <= 128

Returns all child objects of all Parameter Set, Pin, Port, and Sheet Symbol objects that have a Color property whose Blue component has a value which is greater than or equal to 96, and whose Green component has a value which is greater than or equal to 64 and less than or equal to 192, and whose Red component has a value which is less than or equal to 128.
PartComment Field

Description
Returns all Part objects having a Part Comment property that complies with the Query.
Note: The PartComment property is only defined for Part objects.

Syntax
PartComment : String

Example
PartComment = '1488'
Returns all Part objects that have a Part Comment property of '1488'.

PartDesignator Field

Description
Returns all Part objects having a Component Designator property that complies with the Query.
Note: The PartDesignator property is only defined for Part objects.

Syntax
PartDesignator : String

Example
PartDesignator = 'U4'
Returns all Part objects that have a Component Designator property of 'U4'.

PartDisplayMode Field

Description
Returns all Part objects having a Display Mode property that complies with the Query.
Note: The PartDisplayMode property is only defined for Part objects.

Syntax
PartDisplayMode : PartDisplayMode_String
PartDisplayMode_String must be one of the strings from the following list:
'Normal' , 'Alternate 1' , 'Alternate 2' , ... , 'Alternate 255'
Note: The single quote characters (') shown at the start and end of each PartDisplayMode_String are both mandatory.

Examples
PartDisplayMode = 'Normal'
Returns all Part objects that have a 'Normal' Display Mode property.
PartDisplayMode = 'Alternate 1'
Returns all Part objects that have an 'Alternate 1' Display Mode property.
PartDisplayMode Like 'Alternate *'
Returns all Part objects that do not have a 'Normal' Display Mode property.
PartDisplayMode <> 'Alternate 2'
Returns all objects except Part objects that have an 'Alternate 2' Display Mode property. (Only Part objects have a PartDisplayMode property, so all remaining types of objects do not have an 'Alternate 2' PartDisplayMode property, and are thus also returned by this Query.)

PartId Field

Description
Returns all Part objects having a Current Part property that complies with the Query.
Note: The PartId property is only defined for Part objects.

**Syntax**

PartId : String

**Example**

PartId = 'A'

Returns all Part objects that have a Current Part property of 'A'.

---

**PartIDLocked Field**

**Description**

Returns all Part objects having a Lock Part ID property that complies with the Query.

Note: The PartIDLocked property is only defined for Part objects.

**Syntax**

PartIDLocked : Boolean_String

Boolean_String must be either 'True' or 'False'.

**Examples**

PartIDLocked = 'True'

Returns all Part objects that have a 'True' Lock Part ID property.

PartIDLocked = 'False'

Returns all Part objects that have a 'False' Lock Part ID property.

PartIDLocked <> 'True'

Not (PartIDLocked = 'True')

Returns all objects except Part objects that have a 'True' Lock Part ID property.

PartIDLocked <> 'False'

Not (PartIDLocked = 'False')

Returns all objects except Part objects that have a 'False' Lock Part ID property.

---

**PartLibrary Field**

**Description**

Returns all Part objects having a Library property that complies with the Query.

Note: The PartLibrary property is only defined for Part objects.

**Syntax**

PartLibrary : String

**Example**

PartLibrary = 'Miscellaneous Devices.IntLib'

Returns all Part objects that have a Library property of 'Miscellaneous Devices.IntLib'.

PartLibrary Like 'Motorola *.IntLib'

Returns all Part objects that have a Library property whose associated string starts with 'Motorola ' and which ends with '.IntLib'.

---

**PartLibReference Field**

**Description**

Returns all Part objects having a Library Reference property that complies with the Query.

Note: The PartLibReference property is only defined for Part objects.

**Syntax**

PartLibReference : String
Example
PartLibReference = 'LM833'
Returns all Part objects that have a Library Reference property of 'LM833'.

PinDefaultValue Field
Description
Returns all Pin objects having a PinDefaultValue property that complies with the Query.

Syntax
PinDefaultValue = String
Note: The single quote characters ('') shown at the start and end of the String are both mandatory.

Examples
PinDefaultValue = '1'
Returns all Pin objects that have a PinDefaultValue property which is equal to 1.
PinDefaultValue <> '1'
Returns all objects except Pin objects that have a PinDefaultValue property which is equal to 1. (Only Pin objects have a PinDefaultValue property, so all remaining types of objects do not have a PinDefaultValue property of 1, and are thus also returned by this Query.)

PinDesignator Field
Description
Returns all Pin objects having a Pin Designator property that complies with the Query.
Note: The PinDesignator property is only defined for Pin objects.

Syntax
PinDesignator : String

Examples
PinDesignator = '1'
Returns all Pin objects that have a Pin Designator property of '1'.
PinDesignator = 'AA'
Returns all Pin objects that have a Pin Designator property of 'AA'.
PinDesignator Like 'Z?'
Returns all Pin objects that have a Pin Designator property whose associated string consists of two characters, with the first of those characters being 'Z'.

PinElectrical Field
Description
Returns all Pin objects having an Electrical Type property that complies with the Query.
Note: The PinElectrical property is only defined for Pin objects.

Syntax
PinElectrical = ElectricalType_String
PinElectrical <> ElectricalType_String
ElectricalType_String must be one of the strings from the following list:
'Emitter', 'HiZ', 'Input', 'IO', 'OpenCollector', 'Output', 'Passive', 'Power', 'VHDL-Buffer', 'VHDL-Port'
Note: The single quote characters ('') shown at the start and end of each ElectricalType_String are both mandatory.

Examples
PinElectrical = 'Passive'
Returns all Pin objects that have a 'Passive' Electrical Type property.

PinElectrical <> 'HiZ'
Returns all objects except Pin objects that have a 'HiZ' Electrical Type property. (Only Pin objects have a PinElectrical property, so all remaining types of objects do not have a 'HiZ' PinElectrical property, and are thus also returned by this Query.)

PinFormalType Field

Description
Returns all Pin objects having a PinFormalType property that complies with the Query.
Note: The PinFormalType property is only defined for Pin objects that possess VHDL parameters.

Syntax
Formal_String = FormalType_String
Formal_String must be one of the strings from the following list:
'Unitialized', 'ForcingUnknown', 'Forcing0', 'Forcing1', 'HiZ', 'WeakUnknown', 'Weak0', 'Weak1', 'DontCare'
Note: The single quote characters (') shown at the start and end of each Formal_String are both mandatory.

Examples
PinFormalType = 'Forcing0'
Returns all Pin objects that have a 'Forcing0' PinFormalType property.

PinFormalType <> 'ForcingUnknown'
Returns all objects except Pin objects that have a 'ForcingUnknown' PinFormalType property. (Only Pin objects have a PinFormalType property, so all remaining types of objects do not have a 'ForcingUnknown' PinFormalType property, and are thus also returned by this Query.)

PinFormalType <> '' && PinFormalType <> 'ForcingUnknown'
Returns all Pin objects that do not have a 'ForcingUnknown' PinFormalType property.

PinHiddenNetName Field

Description
Returns all Pin objects having a Hidden Net Name property that complies with the Query.
Note: The PinHiddenNetName property is only defined for Pin objects.

Syntax
PinHiddenNetName : String

Examples
PinHiddenNetName = 'GND'
Returns all Pin objects that have a Hidden Net Name property of 'GND'.

PinHiddenNetName Like '???'
Returns all Pin objects that have a Hidden Net Name property whose associated string consists of any three characters; 'GND' and 'Vcc' are examples of thus compliant strings.

PinIeeeSymbolInner Field

Description
Returns all Pin objects having an Ieee Symbol Inside property that complies with the Query.
Note: The PinIeeeSymbolInner property is only defined for Pin objects.

Syntax
PinIeeeSymbolInner = IeeeSymbolInnerType_String
PinIeeeSymbolInner <> IeeeSymbolInnerType_String

IeeeSymbolInnerType_String must be one of the strings from the following list:

Note: The single quote characters (') shown at the start and end of each IeeeSymbolInnerType_String are both mandatory.

Examples
PinIeeeSymbolInner = 'Schmitt'
Returns all Pin objects that have a 'Schmitt' Ieee Symbol Inside property.

PinIeeeSymbolInner <> 'Pulse'
Returns all objects except Pin objects that have a 'Pulse' Ieee Symbol Inside property. (Only Pin objects have a PinIeeeSymbolInner property, so all remaining types of objects do not have a 'Pulse' PinIeeeSymbolInner property, and are thus also returned by this Query.)

PinIeeeSymbolInnerEdge Field

Description
Returns all Pin objects having an Ieee Symbol Inside Edge property that complies with the Query.

Note: The PinIeeeSymbolInnerEdge property is only defined for Pin objects.

Syntax
PinIeeeSymbolInnerEdge = IeeeSymbolInnerEdgeType_String
PinIeeeSymbolInnerEdge <> IeeeSymbolInnerEdgeType_String

IeeeSymbolInnerEdgeType_String must be one of the strings from the following list:
'Clock', 'No Symbol'

Note: The single quote characters (') shown at the start and end of each IeeeSymbolInnerEdgeType_String are both mandatory.

Examples
PinIeeeSymbolInnerEdge = 'Clock'
Returns all Pin objects that have a 'Clock' Ieee Symbol Inside Edge property.

PinIeeeSymbolInnerEdge <> 'No Symbol'
Returns all objects except Pin objects that have a 'No Symbol' Ieee Symbol Inside Edge property. (Only Pin objects have a PinIeeeSymbolInnerEdge property, so all remaining types of objects do not have a 'No Symbol' PinIeeeSymbolInnerEdge property, and are thus also returned by this Query.)

PinIeeeSymbolOuter Field

Description
Returns all Pin objects having an Ieee Symbol Outside property that complies with the Query.

Note: The PinIeeeSymbolOuter property is only defined for Pin objects.

Syntax
PinIeeeSymbolOuter = IeeeSymbolOuterType_String
PinIeeeSymbolOuter <> IeeeSymbolOuterType_String

IeeeSymbolOuterType_String must be one of the strings from the following list:
'Analog Signal In', 'Bidirectional Signal Flow', 'Digital Signal In', 'Left Right Signal Flow', 'No Symbol', 'Not Logic Connection', 'Right Left Signal Flow'

Note: The single quote characters (') shown at the start and end of each IeeeSymbolOuterType_String are both mandatory.

Examples
PinIeeeSymbolOuter = 'Bidirectional Signal Flow'
Returns all Pin objects that have a 'Bidirectional Signal Flow' Ieee Symbol Outside property.

PinIeeeSymbolOuter <> 'Right Left Signal Flow'
Returns all objects except Pin objects that have a 'Right Left Signal Flow' Ieee Symbol Outside property. (Only Pin objects have a PinIeeeSymbolOuter property, so all remaining types of objects do not have a 'Right Left Signal Flow' PinIeeeSymbolOuter property, and are thus also returned by this Query.)

**PinIeeeSymbolOuterEdge Field**

**Description**

Returns all Pin objects having an Ieee Symbol Outside Edge property that complies with the Query.

**Note:** The PinIeeeSymbolOuterEdge property is only defined for Pin objects.

**Syntax**

PinIeeeSymbolOuterEdge = IeeeSymbolOuterEdgeType_String
PinIeeeSymbolOuterEdge <> IeeeSymbolOuterEdgeType_String

IeeeSymbolOuterEdgeType_String must be one of the strings from the following list:

'Active Low Input', 'Active Low Output', 'Dot', 'No Symbol'

**Note:** The single quote characters (') shown at the start and end of each IeeeSymbolOuterEdgeType_String are both mandatory.

**Examples**

PinIeeeSymbolOuterEdge = 'Dot'

Returns all Pin objects that have a 'Dot' Ieee Symbol Outside Edge property.

PinIeeeSymbolOuterEdge <> 'Active Low Output'

Returns all objects except Pin objects that have an 'Active Low Output' Ieee Symbol Outside Edge property. (Only Pin objects have a PinIeeeSymbolOuterEdge property, so all remaining types of objects do not have an 'Active Low Output' PinIeeeSymbolOuterEdge property, and are thus also returned by this Query.)

**PinLength Field**

**Description**

Returns all Pin objects having a Length property that complies with the Query.

**Note:** The PinLength property is only defined for Pin objects.

**Syntax**

PinLength : Numeric_String

PinLength : Number {If all non-Pin objects are totally excluded by one or more preceding tokens within the Query.}

The PinLength keyword can **always** be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but **only** when all non-Pin objects (which do not have a Length property) have been **totally** excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:

PinLength <> ''
PinLength > ''
ObjectKind = 'Pin'
IsPin

The usage of such tokens is demonstrated in the examples that follow.

**Examples**

PinLength = '30'
PinLength <> ' ' & PinLength = 30

Returns all Pin objects that have a Length property which is equal to 30.

PinLength > ' ' & PinLength Between 20 And 40

Returns all Pin objects that have a Length property which is greater than or equal to 20 and less than or equal to 40.

ObjectKind = 'Pin' & PinLength >= 10

Returns all Pin objects that have a Length property which is greater than or equal to 10.
IsPin && PinLength < 30  
Returns all Pin objects that have a Length property which is less than 30.

PinLength <> '40'
Returns all objects except Pin objects that have a Length property which is equal to 40. (Only Pin objects have a PinLength property, so all remaining types of objects do not have a PinLength property of 40, and are thus also returned by this Query.)

IsPin && PinLength <> 40
Returns all Pin objects that have a Length property which is not equal to 40.

**PinShowDesignator Field**

**Description**
Returns all Pin objects having a Show Designator property that complies with the Query.
Note: The PinShowDesignator property is only defined for Pin objects.

**Syntax**
PinShowDesignator : Boolean_String
Boolean_String must be either 'True' or 'False'.

**Examples**
PinShowDesignator = 'True'
Returns all Pin objects that have a 'True' Show Designator property.
PinShowDesignator = 'False'
Returns all Pin objects that have a 'False' Show Designator property.
PinShowDesignator <> 'True'
Not (PinShowDesignator = 'True')
Returns all objects except Pin objects that have a 'True' Show Designator property.
PinShowDesignator <> 'False'
Not (PinShowDesignator = 'False')
Returns all objects except Pin objects that have a 'False' Show Designator property.

**PinsLocked Field**

**Description**
Returns all Part objects having a Pins Locked property that complies with the Query.
Note: The PinsLocked property is only defined for Part objects.

**Syntax**
PinsLocked : Boolean_String
Boolean_String must be either 'True' or 'False'.

**Examples**
PinsLocked = 'True'
Returns all Part objects that have a 'True' Pins Locked property.
PinsLocked = 'False'
Returns all Part objects that have a 'False' Pins Locked property.
PinsLocked <> 'True'
Not (PinsLocked = 'True')
Returns all objects except Part objects that have a 'True' Pins Locked property.
PinsLocked <> 'False'
Not (PinsLocked = 'False')
Returns all objects except Part objects that have a 'False' Pins Locked property.
**PinSwapId_Part Field**

**Description**
Returns all Pin objects having a Part Swap Group property that complies with the Query.

*Note:* The PinSwapId_Part property is only defined for Pin objects.

**Syntax**

```
PinSwapId_Part : Numeric_String
```

The PinSwapId_Part keyword can *always* be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but *only* when all non-Pin objects (which do not have a Part Swap Group property) have been *totally* excluded by one or more *preceding* tokens within the Query. Examples of thus-compliant tokens are as follows:

```
PinSwapId_Part <> ''
PinSwapId_Part > ''
ObjectKind = 'Pin'
IsPin
```

The usage of such tokens is demonstrated in the examples that follow.

**Examples**

```
PinSwapId_Part = '4'
```

Returns all Pin objects that have a Part Swap Group property which is equal to 4.

```
PinSwapId_Part > '' && PinSwapId_Part = 4
```

Returns all Pin objects that have a Part Swap Group property which is greater than or equal to 1 and less than or equal to 5.

```
ObjectKind = 'Pin' && PinSwapId_Part Between 1 And 5
```

Returns all Pin objects that have a Part Swap Group property which is greater than or equal to 2.

```
IsPin && PinSwapId_Part < 3
```

Returns all Pin objects that have a Part Swap Group property which is less than 3.

```
PinSwapId_Part <> '5'
```

Returns all objects *except* Pin objects that have a Part Swap Group property which is equal to 5. (Only Pin objects have a PinSwapId_Part property, so all remaining types of objects do not have a PinSwapId_Part property of 5, and are thus also returned by this Query.)

```
IsPin && PinSwapId_Part <> 5
```

Returns all Pin objects that have a Part Swap Group property which is *not* equal to 5.

**PinSwapId_PartPin Field**

**Description**
Returns all Pin objects having a Pin Equivalent Across Parts property that complies with the Query.

*Note:* The PinSwapId_PartPin property is only defined for Pin objects.

**Syntax**

```
PinSwapId_PartPin : String
```

**Example**

```
PinSwapId_PartPin = ''
```

Returns all Pin objects that have a Pin Equivalent Across Parts property of "".

**PinSwapId_Pin Field**

**Description**
Returns all Pin objects having a Pin Swap Group property that complies with the Query.
Note: The PinSwapId_Pin property is only defined for Pin objects.

**Syntax**

PinSwapId_Pin : String

**Examples**

PinSwapId_Pin = 'A1'

Returns all Pin objects that have a Pin Swap Group property of 'A1'.

PinSwapId_Pin Like 'D?'

Returns all Pin objects that have a Pin Swap Group property whose associated string consists of two characters, with the first of those characters being 'D'.

**PinWidth Field**

**Description**

Returns all Pin objects having a Width property that complies with the Query.

Note: The PinWidth property is only defined for Pin objects.

**Syntax**

PinWidth : Numeric_String

**Examples**

PinWidth = '30'

PinWidth <> '' && PinWidth = 30

Returns all Pin objects that have a Length property which is equal to 30.

PinWidth > ' ' && PinWidth Between 20 And 40

Returns all Pin objects that have a Width property which is greater than or equal to 20 and less than or equal to 40.

ObjectKind = 'Pin' && PinWidth >= 10

Returns all Pin objects that have a Width property which is greater than or equal to 10.

IsPin && PinLength < 30

Returns all Pin objects that have a Length property which is less than 30.

PinWidth <> '40'

Returns all objects except Pin objects that have a Width property which is equal to 40. (Only Pin objects have a PinWidth property, so all remaining types of objects do not have a PinWidth property of 40, and are thus also returned by this Query.)

IsPin && PinLength <= 40

Returns all Pin objects that have a Width property which is not equal to 40.
PortArrowStyle Field

**Description**
Returns all Port and Sheet Entry objects having a Port Style property that complies with the Query.

Note: The PortArrowStyle property is only defined for Port and Sheet Entry objects.

**Syntax**

PortArrowStyle = PortStyle_String
PortArrowStyle <> PortStyle_String

PortStyle_String must be one of the strings from the following list:
'Bottom', 'Left', 'Left & Right', 'None (Horizontal)', 'None (Vertical)', 'Right', 'Top', 'Top & Bottom'

Note: The single quote characters (') shown at the start and end of each PortStyle_String are both mandatory.

**Examples**

PortArrowStyle = 'Top'
Returns all Port and Sheet Entry objects that have a 'Top' Port Style property.

PortArrowStyle <> 'Right'
Returns all objects except Port and Sheet Entry objects that have a 'Right' Port Style property. (Only Port and Sheet Entry objects have a PortArrowStyle property, so all remaining types of objects do not have a 'Right' PortArrowStyle property, and are thus also returned by this Query.)

PortIOType Field

**Description**
Returns all Port and Sheet Entry objects having an IO Type property that complies with the Query.

Note: The PortIOType property is only defined for Port and Sheet Entry objects.

**Syntax**

PortIOType = IOType_String
PortIOType <> IOType_String

IOType_String must be one of the strings from the following list:
'Bidirectional', 'Input', 'Output', 'Unspecified'

Note: The single quote characters (') shown at the start and end of each IOType_String are both mandatory.

**Examples**

PortIOType = 'Bidirectional'
Returns all Port and Sheet Entry objects that have a 'Bidirectional' IO Type property.

PortIOType <> 'Output'
Returns all objects except Port and Sheet Entry objects that have an 'Output' IO Type property. (Only Port and Sheet Entry objects have a PortIOType property, so all remaining types of objects do not have an 'Output' PortIOType property, and are thus also returned by this Query.)

PowerObjectStyle Field

**Description**
Returns all Power Object objects having a Power Object Style property that complies with the Query.

Note: The PowerObjectStyle property is only defined for Power Object objects.

**Syntax**

PowerObjectStyle = PowerObjectStyle_String
PowerObjectStyle <> PowerObjectStyle_String

PowerObjectStyle_String must be one of the strings from the following list:
'Arrow' , 'Bar' , 'Circle' , 'Earth' , 'Power Ground' , 'Signal Ground' , 'Wave'

Note: The single quote characters (') shown at the start and end of each PowerObjectStyle_String are both mandatory.

**Examples**

PowerObjectStyle = 'Circle'

Returns all Power Object objects that have a 'Circle' Power Object Style property.

PowerObjectStyle <> 'Wave'

Returns all objects except Power Object objects that have a 'Wave' Power Object Style property. (Only Power Object objects have a PowerObjectStyle property, so all remaining types of objects do not have a 'Wave' PowerObjectStyle property, and are thus also returned by this Query.)

PowerObjectStyle <> '' && PowerObjectStyle <> 'Wave'

IsPowerObject && PowerObjectStyle <> 'Wave'

Returns all Power Object objects that do not have a 'Wave' Power Object Style property.

---

**Radius Field**

**Description**

Returns all Arc, Ellipse, Elliptical Arc, and Pie objects having a Radius property that complies with the Query.

Note: The Radius property is only defined for Arc, Ellipse, Elliptical Arc, and Pie objects.

**Syntax**

Radius : Numeric_String

Radius : Number {If all objects which do not have a Radius property are totally excluded by one or more preceding tokens within the Query.}

The Radius keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all objects which do not have a Radius property have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:

Radius = '20'

Radius <> '' && Radius = 20

Returns all Arc, Ellipse, Elliptical Arc, and Pie objects that have a Radius property which is equal to 20.

Radius > '' && Radius Between 10 And 40

Returns all Arc, Ellipse, Elliptical Arc, and Pie objects that have a Radius property which is greater than or equal to 10 and less than or equal to 40.

IsArc Or IsEllipse Or IsEllipticalArc Or IsPie

IsPie

IsArc Or IsEllipticalArc

The usage of such tokens is demonstrated in the examples that follow.

**Examples**

Radius = '20'

Radius <> '' && Radius = 20

Returns all Arc, Ellipse, Elliptical Arc, and Pie objects that have a Radius property which is equal to 20.

Radius > '' && Radius Between 10 And 40

Returns all Arc, Ellipse, Elliptical Arc, and Pie objects that have a Radius property which is greater than or equal to 10 and less than or equal to 40.

IsArc Or IsEllipse Or IsEllipticalArc Or IsPie && Radius >= 30

Returns all Arc, Ellipse, Elliptical Arc, and Pie objects that have a Radius property which is greater than or equal to 30.

Radius > '' && Radius < 50

Returns all Arc, Ellipse, Elliptical Arc, and Pie objects that have a Radius property which is less than 50.

Radius <> '15'

Returns all objects except Arc, Ellipse, Elliptical Arc, and Pie objects that have a Radius property which is equal to 15. (Only Arc, Ellipse, Elliptical Arc, and Pie objects have a Radius property, so all remaining types of objects do not have a Radius property of 15, and are thus also returned by this Query.)

IsPie && Radius <> 15
Returns all Pie objects that have a Radius property which is **not** equal to 15.

IsArc Or IsEllipticalArc && Radius <= 35

Returns all Arc and Elliptical Arc objects that have a Radius property which is less than or equal to 35.

**RoundRectangleCornerRadiusX Field**

**Description**

Returns all Round Rectangle objects having a X Radius property that complies with the Query.

Note: The RoundRectangleCornerRadiusX property is only defined for Round Rectangle objects.

**Syntax**

RoundRectangleCornerRadiusX : Numeric_String

RoundRectangleCornerRadiusX : Number {If all non-Round Rectangle objects are totally excluded by one or more preceding tokens within the Query.}

The RoundRectangleCornerRadiusX keyword can **always** be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but **only** when all non-Round Rectangle objects (which do **not** have a X Radius property) have been **totally** excluded by one or more **preceding** tokens within the Query. Examples of thus-compliant tokens are as follows:

RoundRectangleCornerRadiusX <> ''
RoundRectangleCornerRadiusX > ''
ObjectKind = 'Round Rectangle'

The usage of such tokens is demonstrated in the examples that follow.

**Examples**

RoundRectangleCornerRadiusX = '20'
RoundRectangleCornerRadiusX <> '' && RoundRectangleCornerRadiusX = 20

Returns all Round Rectangle objects that have a X Radius property which is equal to 20.

RoundRectangleCornerRadiusX > '' && RoundRectangleCornerRadiusX Between 10 And 25

Returns all Round Rectangle objects that have a X Radius property which is greater than or equal to 10 and less than or equal to 25.

ObjectKind = 'Round Rectangle' && RoundRectangleCornerRadiusX >= 15

Returns all Round Rectangle objects that have a X Radius property which is greater than or equal to 15.

IsRoundRectangle && RoundRectangleCornerRadiusX < 20

Returns all Round Rectangle objects that have a X Radius property which is less than 20.

RoundRectangleCornerRadiusX <= '30'

Returns all objects **except** Round Rectangle objects that have a X Radius property which is equal to 30. (Only Round Rectangle objects have a RoundRectangleCornerRadiusX property, so all remaining types of objects do not have a RoundRectangleCornerRadiusX property of 30, and are thus also returned by this Query.)

IsRoundRectangle && RoundRectangleCornerRadiusX <= 30

Returns all Round Rectangle objects that have a X Radius property which is **not** equal to 30.
RoundRectangleCornerRadiusY Field

Description
Returns all Round Rectangle objects having a Y Radius property that complies with the Query.

Note: The RoundRectangleCornerRadiusY property is only defined for Round Rectangle objects.

Syntax
RoundRectangleCornerRadiusY : Numeric_String
RoundRectangleCornerRadiusY : Number {If all non-Round Rectangle objects are totally excluded by one or more preceding tokens within the Query.}

The RoundRectangleCornerRadiusY keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all non-Round Rectangle objects (which do not have a Y Radius property) have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:

RoundRectangleCornerRadiusY <> ''
RoundRectangleCornerRadiusY > ''
ObjectKind = 'Round Rectangle'
IsRoundRectangle

The usage of such tokens is demonstrated in the examples that follow.

Examples
RoundRectangleCornerRadiusY = '20'
RoundRectangleCornerRadiusY <> '' & RoundRectangleCornerRadiusY = 20
Returns all Round Rectangle objects that have a Y Radius property which is equal to 20.

RoundRectangleCornerRadiusY > '' & RoundRectangleCornerRadiusY Between 10 And 25
Returns all Round Rectangle objects that have a Y Radius property which is greater than or equal to 10 and less than or equal to 25.

ObjectKind = 'Round Rectangle' & RoundRectangleCornerRadiusY >= 15
Returns all Round Rectangle objects that have a Y Radius property which is greater than or equal to 15.

IsRoundRectangle & RoundRectangleCornerRadiusY < 20
Returns all Round Rectangle objects that have a Y Radius property which is less than 20.

RoundRectangleCornerRadiusY <> '30'
Returns all objects except Round Rectangle objects that have a Y Radius property which is equal to 30. (Only Round Rectangle objects have a RoundRectangleCornerRadiusY property, so all remaining types of objects do not have a RoundRectangleCornerRadiusY property of 30, and are thus also returned by this Query.)

IsRoundRectangle & RoundRectangleCornerRadiusY <> 30
Returns all Round Rectangle objects that have a Y Radius property which is not equal to 30.

SecondaryRadius Field

Description
Returns all Ellipse and Elliptical Arc objects having a Secondary Radius property that complies with the Query.

Note: The SecondaryRadius property is only defined for Ellipse and Elliptical Arc objects.

Syntax
SecondaryRadius : Numeric_String
SecondaryRadius : Number {If all objects which do not have a Secondary Radius property are totally excluded by one or more preceding tokens within the Query.}

The SecondaryRadius keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all objects which do not have a Secondary Radius property have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:
SecondaryRadius <> ''
SecondaryRadius > ''
IsEllipse Or IsEllipticalArc
IsEllipse

The usage of such tokens is demonstrated in the examples that follow.

Examples
SecondaryRadius = '20'
SecondaryRadius <> '' && SecondaryRadius = 20
Returns all Ellipse and Elliptical Arc objects that have a Secondary Radius property which is equal to 20.
SecondaryRadius > '' && SecondaryRadius Between 10 And 40
Returns all Ellipse and Elliptical Arc objects that have a Secondary Radius property which is greater than or equal to 10 and less than or equal to 40.
IsEllipse Or IsEllipticalArc && SecondaryRadius >= 30
Returns all Ellipse and Elliptical Arc objects that have a Secondary Radius property which is greater than or equal to 30.
IsEllipse && SecondaryRadius < 50
Returns all Ellipse objects that have a Secondary Radius property which is less than 50.
SecondaryRadius <> '25'
Returns all objects except Ellipse and Elliptical Arc objects that have a Secondary Radius property which is equal to 25. (Only Ellipse and Elliptical Arc objects have a SecondaryRadius property, so all remaining types of objects do not have a SecondaryRadius property of 25, and are thus also returned by this Query.)
IsEllipse Or IsEllipticalArc && SecondaryRadius <> 25
Returns all Ellipse and Elliptical Arc objects that have a Secondary Radius property which is not equal to 25.

SelectedVertex_Y Field

Description
Returns all polygon based objects (such as buses and polygons) having a selected vertex that complies with the Query.

Syntax
SelectedVertex_Y : String
Where string is a numerical string enclosed by single quotes.

Example
SelectedVertex_Y = '220'
Returns all objects that have a selected first vertex at Y coordinate with a value of 220 units.

SelectedVertex_X Field

Description
Returns all polygon based objects (such as buses and polygons) having a selected vertex that complies with the Query.

Syntax
SelectedVertex_X : String
Where string is a numerical string enclosed by single quotes.

Example
SelectedVertex_X = '220'
Returns all objects that have a selected first vertex at X coordinate with a value of 220 units.

SelectedVertex2_X Field

Description
Returns all polygon based objects (such as buses and polygons) having a selected vertex that complies with the Query.

**Syntax**

SelectedVertex2_X : String

Where string is a numerical string enclosed by single quotes.

**Example**

SelectedVertex2_X = '220'

Returns all objects that have a selected second vertex at X coordinate with a value of 220 units.

### SelectedVertex2_Y Field

**Description**

Returns all polygon based objects (such as buses and polygons) having a selected vertex that complies with the Query.

**Syntax**

SelectedVertex2_Y : String

Where string is a numerical string enclosed by single quotes.

**Example**

SelectedVertex2_Y = '220'

Returns all objects that have a selected second vertex at Y coordinate with a value of 220 units.

### SheetEntryDistanceFromTop Field

**Description**

Returns all Sheet Entry objects having a Position property that complies with the Query.

Note: The SheetEntryDistanceFromTop property is only defined for Sheet Entry objects.

**Syntax**

SheetEntryDistanceFromTop : Numeric_String

SheetEntryDistanceFromTop : Number {If all non-Sheet Entry objects are totally excluded by one or more preceding tokens within the Query.}

The SheetEntryDistanceFromTop keyword can *always* be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but *only* when all non-Sheet Entry objects (which do not have a Position property) have been *totally* excluded by one or more *preceding* tokens within the Query. Examples of thus-compliant tokens are as follows:

- SheetEntryDistanceFromTop <> ''
- SheetEntryDistanceFromTop > ''
- ObjectKind = 'Sheet Entry'
- IsSheetEntry

The usage of such tokens is demonstrated in the examples that follow.

**Examples**

SheetEntryDistanceFromTop = '4'

SheetEntryDistanceFromTop <> '' && SheetEntryDistanceFromTop = 4

Returns all Sheet Entry objects that have a Position property which is equal to 4.

SheetEntryDistanceFromTop > '' && SheetEntryDistanceFromTop Between 2 And 5

Returns all Sheet Entry objects that have a Position property which is greater than or equal to 2 and less than or equal to 5.

ObjectKind = 'Sheet Entry' && SheetEntryDistanceFromTop >= 3

Returns all Sheet Entry objects that have a Position property which is greater than or equal to 3.

IsSheetEntry && SheetEntryDistanceFromTop < 6

Returns all Sheet Entry objects that have a Position property which is less than 6.
Returns all objects except Sheet Entry objects that have a Position property which is equal to 1. (Only Sheet Entry objects have a SheetEntryDistanceFromTop property, so all remaining types of objects do not have a SheetEntryDistanceFromTop property of 1, and are thus also returned by this Query.)

\[
\text{IsSheetEntry} \land \text{SheetEntryDistanceFromTop} \neq 1
\]

Returns all Sheet Entry objects that have a Position property which is not equal to 1.

**SheetEntrySide Field**

**Description**

Returns all Sheet Entry objects having a Side property that complies with the Query.

Note: The SheetEntrySide property is only defined for Sheet Entry objects.

**Syntax**

\[
\text{SheetEntrySide} = \text{SheetSide\_String}
\]

\[
\text{SheetEntrySide} \neq \text{SheetSide\_String}
\]

SheetSide\_String must be one of the strings from the following list:

- 'Bottom'
- 'Left'
- 'Right'
- 'Top'

Note: The single quote characters (') shown at the start and end of each SheetSide\_String are both mandatory.

**Examples**

\[
\text{SheetEntrySide} = 'Right'
\]

Returns all Sheet Entry objects that have a 'Right' Side property.

\[
\text{SheetEntrySide} \neq 'Bottom'
\]

Returns all objects except Sheet Entry objects that have a 'Bottom' Side property. (Only Sheet Entry objects have a SheetEntrySide property, so all remaining types of objects do not have a 'Bottom' SheetEntrySide property, and are thus also returned by this Query.)

\[
\text{SheetEntrySide} \neq '' \land \text{SheetEntrySide} \neq 'Bottom'
\]

\[
\text{IsSheetEntry} \land \text{SheetEntrySide} \neq 'Bottom'
\]

Returns all Sheet Entry objects that do not have a 'Bottom' Side property.

**SheetFileName Field**

**Description**

Returns all Sheet Symbol objects having a Sheet File Name property that complies with the Query.

Note: The SheetFileName property is only defined for Sheet Symbol objects.

**Syntax**

\[
\text{SheetFileName} : \text{String}
\]

**Example**

\[
\text{SheetFileName} = 'ISA Bus and Address Decoding.SchDoc'
\]

Returns all Sheet Symbol objects that have a Sheet File Name property of 'ISA Bus and Address Decoding.SchDoc'.

**SheetName Field**

**Description**

Returns all Sheet Symbol objects having a Sheet Name property that complies with the Query.

Note: The SheetName property is only defined for Sheet Symbol objects.

**Syntax**

\[
\text{SheetName} : \text{String}
\]

**Example**

\[
\text{SheetName} = 'ISA Bus and Address Decoding.SchDoc'
\]
SheetName = 'ISA Bus and Address Decoding'
Returns all Sheet Symbol objects that have a Sheet Name property of 'ISA Bus and Address Decoding'.

**ShowDesignator Field**

**Description**
Returns all Part objects having a Show Designator property that complies with the Query.

*Note: The ShowDesignator property is only defined for Part objects.*

**Syntax**
ShowDesignator : Boolean_String

*Boolean_String must be either 'True' or 'False'.*

**Examples**
- ShowDesignator = 'True'
  Returns all Part objects that have a 'True' Show Designator property.
- ShowDesignator = 'False'
  Returns all Part objects that have a 'False' Show Designator property.
- ShowDesignator <> 'True'
  Not (ShowDesignator = 'True')
  Returns all objects except Part objects that have a 'True' Show Designator property.
- ShowDesignator <> 'False'
  Not (ShowDesignator = 'False')
  Returns all objects except Part objects that have a 'False' Show Designator property.

**ShowHiddenFields Field**

**Description**
Returns all Sheet Symbol objects having a Show Hidden Fields property that complies with the Query.

*Note: The ShowHiddenFields property is only defined for Sheet Symbol objects.*

**Syntax**
ShowHiddenFields : Boolean_String

*Boolean_String must be either 'True' or 'False'.*

**Examples**
- ShowHiddenFields = 'True'
  Returns all Sheet Symbol objects that have a 'True' Show Hidden Fields property.
- ShowHiddenFields = 'False'
  Returns all Sheet Symbol objects that have a 'False' Show Hidden Fields property.
- ShowHiddenFields <> 'True'
  Not (ShowHiddenFields = 'True')
  Returns all objects except Sheet Symbol objects that have a 'True' Show Hidden Fields property.
- ShowHiddenFields <> 'False'
  Not (ShowHiddenFields = 'False')
  Returns all objects except Sheet Symbol objects that have a 'False' Show Hidden Fields property.

**ShowHiddenPins Field**

**Description**
Returns all Part objects having a Show Hidden Pins property that complies with the Query.
Note: The ShowHiddenPins property is only defined for Part objects.

Syntax
ShowHiddenPins : Boolean_String

Boolean_String must be either 'True' or 'False'.

Examples
ShowHiddenPins = 'True'
Returns all Part objects that have a 'True' Show Hidden Pins property.
ShowHiddenPins = 'False'
Returns all Part objects that have a 'False' Show Hidden Pins property.
ShowHiddenPins <> 'True'
Not (ShowHiddenPins = 'True')
Returns all objects except Part objects that have a 'True' Show Hidden Pins property.
ShowHiddenPins <> 'False'
Not (ShowHiddenPins = 'False')
Returns all objects except Part objects that have a 'False' Show Hidden Pins property.

ShowName Field (Schematic)

Description
Returns all Parameter and Pin objects having a Show Name property that complies with the Query.

Note: The ShowName property is only defined for Parameter and Pin objects.

Syntax
ShowName : Boolean_String

Boolean_String must be either 'True' or 'False'.

Examples
ShowName = 'True'
Returns all Parameter and Pin objects that have a 'True' Show Name property.
ShowName = 'False'
Returns all Parameter and Pin objects that have a 'False' Show Name property.
ShowName <> 'True'
Not (ShowName = 'True')
Returns all objects except Parameter and Pin objects that have a 'True' Show Name property.
ShowName <> 'False'
Not (ShowName = 'False')
Returns all objects except Parameter and Pin objects that have a 'False' Show Name property.
Solid Field

Description
Returns all Ellipse, Pie, Polygon, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects having a Solid property that complies with the Query.

Note: The Solid property is only defined for Ellipse, Pie, Polygon, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects.

Syntax
Solid : Boolean_String

Boolean_String must be either 'True' or 'False'.

Examples
Solid = 'True'
Returns all Ellipse, Pie, Polygon, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects that have a 'True' Solid property.

Solid = 'False'
Returns all Ellipse, Pie, Polygon, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects that have a 'False' Solid property.

Solid <> 'True'
Not (Solid = 'True')
Returns all objects except Ellipse, Pie, Polygon, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects that have a 'True' Solid property.

Solid <> 'False'
Not (Solid = 'False')
Returns all objects except Ellipse, Pie, Polygon, Rectangle, Round Rectangle, Sheet Symbol, and Text Frame objects that have a 'False' Solid property.

StartAngle Field

Description
Returns all Arc, Elliptical Arc, and Pie objects having a Start Angle property that complies with the Query.

Note: The StartAngle property is only defined for Arc, Elliptical Arc, and Pie objects.

Syntax
StartAngle : Numeric_String

StartAngle : Number {If all objects which do not have a Start Angle property are totally excluded by one or more preceding tokens within the Query.}

The number returned specifies the Start Angle in units of degrees.

The StartAngle keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all objects which do not have a Start Angle property have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:

IsArc Or IsEllipticalArc Or IsPie
IsPie
IsArc Or IsEllipticalArc

The usage of such tokens is demonstrated in the examples that follow.

Examples
StartAngle = '30'
IsArc Or IsEllipticalArc Or IsPie && StartAngle = 30

Returns all Arc, Elliptical Arc, and Pie objects that have a Start Angle property which is equal to 30 degrees.

IsArc Or IsEllipticalArc Or IsPie && StartAngle >= 60
Returns all Arc, Elliptical Arc, and Pie objects that have a Start Angle property which is greater than or equal to 60 degrees.

IsArc Or IsEllipticalArc Or IsPie && StartAngle Between 30 And 150

Returns all Arc, Elliptical Arc, and Pie objects that have a Start Angle property which is greater than or equal to 30 degrees and less than or equal to 150 degrees.

IsPie && StartAngle < 45

Returns all Pie objects that have a Start Angle property which is less than 45 degrees.

StartAngle <> '135'

Returns all objects except Arc, Elliptical Arc, and Pie objects that have a Start Angle property which is equal to 135 degrees. (Only Arc, Elliptical Arc, and Pie objects have a StartAngle property, so all remaining types of objects do not have a StartAngle property of 135 degrees, and are thus also returned by this Query.)

IsArc Or IsEllipticalArc && StartAngle <> 135

Returns all Arc and Elliptical Arc objects that have a Start Angle property which is not equal to 135 degrees.

IsArc Or IsEllipticalArc Or IsPie && (StartAngle Between 315 And 360 || StartAngle Between 0 And 45)

Returns all Arc, Elliptical Arc, and Pie objects that have a Start Angle property which is greater than or equal to 315 degrees and less than or equal to 45 degrees.

See Also
EndAngle

StringText Field (Schematic)

Description
Returns all Designator, Error Marker, Net Label, Off Sheet Connector, Power Object, Sheet Name, and Text String objects having a Text property that complies with the Query.

Note: The StringText property is only defined for Designator, Error Marker, Net Label, Off Sheet Connector, Power Object, Sheet Name, and Text String objects.

Syntax
StringText : String

Examples
StringText = 'GND'

Returns all Designator, Error Marker, Net Label, Off Sheet Connector, Power Object, Sheet Name, and Text String objects that have a Text property of 'GND'.

StringText Like '???'

Returns all Designator, Error Marker, Net Label, Off Sheet Connector, Power Object, Sheet Name, and Text String objects that have a Text property whose associated string consists of exactly three characters; 'GND' and 'Vcc' are examples of thus compliant strings.

SymbolScaleFactor Field

Description
Returns all IEEE Symbol objects having a Scale property that complies with the Query.

Note: The SymbolScaleFactor property is only defined for IEEE Symbol objects.

Syntax
SymbolScaleFactor : Numeric_String

SymbolScaleFactor : Number {If all non-IEEE Symbol objects are totally excluded by one or more preceding tokens within the Query.}

The SymbolScaleFactor keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all non-IEEE Symbol objects (which do not have a Scale property) have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:
SymbolScaleFactor <> ''
SymbolScaleFactor > ''
ObjectKind = 'IEEE Symbol'
IsSymbol

The usage of such tokens is demonstrated in the examples that follow.

Examples
SymbolScaleFactor = '10'
SymbolScaleFactor <> '' && SymbolScaleFactor = 10
Returns all IEEE Symbol objects that have a Scale property which is equal to 10.
SymbolScaleFactor > '' && SymbolScaleFactor Between 8 And 12
Returns all IEEE Symbol objects that have a Scale property which is greater than or equal to 8 and less than or equal to 12.
Objectkind = 'IEEE Symbol' && SymbolScaleFactor >= 9
Returns all IEEE Symbol objects that have a Scale property which is greater than or equal to 9.
IsSymbol && SymbolScaleFactor < 11
Returns all IEEE Symbol objects that have a Scale property which is less than 11.
SymbolScaleFactor <> '10'
Returns all objects except IEEE Symbol objects that have a Scale property which is equal to 10. (Only IEEE Symbol objects have a SymbolScaleFactor property, so all remaining types of objects do not have a SymbolScaleFactor property of 10, and are thus also returned by this Query.)
IsSymbol && SymbolScaleFactor <> 10
Returns all IEEE Symbol objects that have a Scale property which is not equal to 10.

TaskHolderConfiguration Field

Description
Not Implemented
Syntax
Not Implemented

TaskHolderInstanceName Field

Description
Not Implemented
Syntax
Not Implemented

TaskHolderProcess Field

Description
Not Implemented
Syntax
Not Implemented

TextColor Field

Description
Returns all Note, Port, Sheet Entry, and Text Frame objects having a Text Color property that complies with the Query.
Note: The TextColor property is only defined for Note, Port, Sheet Entry, and Text Frame objects.
Syntax

TextColor : Numeric_String

TextColor : Number (If all objects which do not have a Text Color property are totally excluded by one or more preceding tokens within the Query.)

The TextColor keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all objects which do not have a Text Color property have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:

TextColor <> ''
TextColor > ''
IsNote Or IsPort Or IsSheetEntry Or IsTextFrame
IsTextFrame
IsPort Or IsSheetEntry

The usage of such tokens is demonstrated in the examples that follow.

The color corresponding to a particular number can be deduced from the following relationship:

$$65536 \times \text{Blue} + 256 \times \text{Green} + \text{Red}$$

where the Blue, Green, and Red components each have a value between 0 and 255 (inclusive).

When the TextColor keyword can be used in conjunction with a Number (as previously described), it is additionally possible to individually evaluate the values of each of the Text Color property's Blue, Green, and Red components, by using the following constructs:

TextColor Div 65536 \{Blue component\}
TextColor Div 256 Mod 256 \{Green component\}
TextColor Mod 256 \{Red component\}

The usage of these constructs is also demonstrated in the examples that follow.

Examples

TextColor = '0'
TextColor <> '' && TextColor = 0

Returns all objects that have a black Text Color property. (A Text Color property of 0, as defined above, corresponds to values of Blue = 0, Green = 0, and Red = 0).

TextColor = '32896'
TextColor > '' && TextColor = 32896

Returns all objects that have a Text Color property of 32896 (as defined above, which corresponds to values of Blue = 0, Green = 128, and Red = 128).

TextColor <> '16777215'

Returns all objects except those that have a (maximum intensity) white Text Color property. (A Text Color property of 16777215, as defined above, corresponds to values of Blue = 255, Green = 255, and Red = 255).

IsNote Or IsPort Or IsSheetEntry Or IsTextFrame && TextColor <> 16777215

Returns all objects that have a Text Color property, and whose Text Color property is not (maximum intensity) white.

TextColor > '' && TextColor Div 65536 >= 96 && TextColor Div 256 Mod 256 Between 64 And 192 && TextColor Mod 256 <= 128

Returns all objects that have a Text Color property, and whose Blue component has a value which is greater than or equal to 96, and whose Green component has a value which is greater than or equal to 64 and less than or equal to 192, and whose Red component has a value which is less than or equal to 128.

IsTextFrame && TextColor Div 65536 <= 192

Returns all Text Frame objects whose Text Color property's Blue component has a value which is less than or equal to 192.

IsPort Or IsSheetEntry && TextColor Mod 256 Between 64 And 192

Returns all Port and Sheet Entry objects whose Text Color property's Red component has a value which is greater than or equal to 64 and less than or equal to 192.
**TextFrameClipToRect Field**

**Description**
Returns all Note and Text Frame objects having a Clip To Area property that complies with the Query.

*Note:* The TextFrameClipToRect property is only defined for Note and Text Frame objects.

**Syntax**
```
TextFrameClipToRect : Boolean_String
```

Boolean_String must be either 'True' or 'False'.

**Examples**
- `TextFrameClipToRect = 'True'`
  Returns all Note and Text Frame objects that have a 'True' Clip To Area property.
- `TextFrameClipToRect <> 'True'`
  Not (TextFrameClipToRect = 'True')
  Returns all objects except Note and Text Frame objects that have a 'True' Clip To Area property.
- `TextFrameClipToRect <> 'False'`
  Not (TextFrameClipToRect = 'False')
  Returns all objects except Note and Text Frame objects that have a 'False' Clip To Area property.

**TextFrameShowBorder Field**

**Description**
Returns all Image and Text Frame objects having a Show Border property that complies with the Query.

*Note:* The TextFrameShowBorder property is only defined for Image and Text Frame objects.

**Syntax**
```
TextFrameShowBorder : Boolean_String
```

Boolean_String must be either 'True' or 'False'.

**Examples**
- `TextFrameShowBorder = 'True'`
  Returns all Image and Text Frame objects that have a 'True' Show Border property.
- `TextFrameShowBorder <> 'True'`
  Not (TextFrameShowBorder = 'True')
  Returns all objects except Image and Text Frame objects that have a 'True' Show Border property.
- `TextFrameShowBorder <> 'False'`
  Not (TextFrameShowBorder = 'False')
  Returns all objects except Image and Text Frame objects that have a 'False' Show Border property.

**TextFrameWordWrap Field**

**Description**
Returns all Note and Text Frame objects having a Word Wrap property that complies with the Query.

*Note:* The TextFrameWordWrap property is only defined for Note and Text Frame objects.

**Syntax**
```
TextFrameWordWrap : Boolean_String
```

Boolean_String must be either 'True' or 'False'.

**Examples**
- `TextFrameWordWrap = 'True'`
  Returns all Note and Text Frame objects that have a 'True' Word Wrap property.
TextFrameWordWrap = 'False'
Returns all Note and Text Frame objects that have a 'False' Word Wrap property.

TextFrameWordWrap <> 'True'
Not (TextFrameWordWrap = 'True')
Returns all objects except Note and Text Frame objects that have a 'True' Word Wrap property.

TextFrameWordWrap <> 'False'
Not (TextFrameWordWrap = 'False')
Returns all objects except Note and Text Frame objects that have a 'False' Word Wrap property.

**Transparent Field**

**Description**
Returns all Ellipse, Polygon, and Rectangle objects having a Transparent property that complies with the Query.

Note: The Transparent property is only defined for Ellipse, Polygon, and Rectangle objects.

**Syntax**
Transparent : Boolean_String

Boolean_String must be either 'True' or 'False'.

**Examples**
Transparent = 'True'
Returns all Ellipse, Polygon, and Rectangle objects that have a 'True' Transparent property.

Transparent = 'False'
Returns all Ellipse, Polygon, and Rectangle objects that have a 'False' Transparent property.

Transparent <> 'True'
Not (Transparent = 'True')
Returns all objects except Ellipse, Polygon, and Rectangle objects that have a 'True' Transparent property.

Transparent <> 'False'
Not (Transparent = 'False')
Returns all objects except Ellipse, Polygon, and Rectangle objects that have a 'False' Transparent property.

**VerticalJustification Field**

**Description**
Returns all Designator, Parameter, Sheet File Name, Sheet Name, and Text String objects having a Vertical Justification property that complies with the Query.

Note: The VerticalJustification property is only defined for Designator, Parameter, Sheet File Name, Sheet Name, and Text String objects.

**Syntax**
VerticalJustification = VerticalJustification_String
VerticalJustification <> VerticalJustification_String

VerticalJustification_String must be one of the strings from the following list:
'BOTTOM', 'CENTER', 'TOP'

Note: The single quote characters (') shown at the start and end of each VerticalJustification_String are both mandatory.

**Examples**
VerticalJustification = 'Center'
Returns all Designator, Parameter, Sheet File Name, Sheet Name, and Text String objects that have a 'Center' Vertical Justification property.

VerticalJustification <> 'Bottom'
**Returns all objects except** Designator, Parameter, Sheet File Name, Sheet Name, and Text String objects that have a 'Bottom' Vertical Justification property. (Only Designator, Parameter, Sheet File Name, Sheet Name, and Text String objects have a Vertical Justification property, so all remaining types of objects do not have a 'Bottom' Vertical Justification property, and are thus also returned by this Query.)

```
VerticalJustification = 'Center' || VerticalJustification = 'Top'
VerticalJustification <> '' && VerticalJustification <> 'Bottom'
```

Returns all Designator, Parameter, Sheet File Name, Sheet Name, and Text String objects that do not have a 'Bottom' Vertical Justification property.

**Width Field (Schematic)**

**Description**

Returns all Port objects having a Width property that complies with the Query. Note: The Width property is only defined for Port objects.

**Syntax**

```
Width : Numeric_String
Width : Number {If all non-Port objects are totally excluded by one or more preceding tokens within the Query.}
```

The Width keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all non-Port objects (which do not have a Width property) have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:

```
Width <> ''
Width > ''
ObjectKind = 'Port'
IsPort
```

The usage of such tokens is demonstrated in the examples that follow.

**Examples**

```
Width = '60'
Width <> '' && Width = 60
```

Returns all Port objects that have a Width property which is equal to 60.

```
Width > '' && Width Between 40 And 80
```

Returns all Port objects that have a Width property which is greater than or equal to 40 and less than or equal to 80.

```
ObjectKind = 'Port' && Width >= 50
```

Returns all Port objects that have a Width property which is greater than or equal to 50.

```
IsPort && Width < 70
```

Returns all Port objects that have a Width property which is less than 70.

```
Width <> '40'
```

Returns all objects except Port objects that have a Width property which is equal to 40. (Only Port objects have a Width property, so all remaining types of objects do not have a Width property of 40, and are thus also returned by this Query.)

```
IsPort && Width <> 40
```

Returns all Port objects that have a Width property which is not equal to 40.
PCB Functions

All Object Type Check

Description
Returns objects on the PCB document.

Syntax
All : Boolean

Examples
All
All = True
Returns all objects.
Not All
All = False
Returns no objects.

PlaneToPlaneDistance Impedance

Description
Specifies the copper-to-copper distance between two adjacent Internal Plane layers within a multi-layer PCB. That distance has an influence upon the characteristic impedance between a track on an intermediate Signal layer and the copper on those Internal Plane layers, or the width of such a track when a particular characteristic impedance is required.

Syntax
PlaneToPlaneDistance : Number
The distance is specified in units of mils. (1 mil = 0.001 inch)

Examples
(60/SQRT(Er))*LN((1.9*PlaneToPlaneDistance)/(0.8*TraceWidth + TraceHeight))
The default Query for specifying the characteristic impedance between a track on the intermediate Signal layer and the copper on the two surrounding Internal Plane layers, when the Stripline option is selected.

((1.9*PlaneToPlaneDistance/EXP(CharacteristicImpedance/(60/SQRT(Er))))- TraceHeight)*1.25
The default Query for specifying the required width of a track on the intermediate Signal layer, when the Stripline option is selected.

ArcRadius Field

Description
Returns all Arc objects having a Radius property that complies with the Query.

Note: The ArcRadius property is only defined for Arc objects.

Syntax
ArcRadius : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(ArcRadius) : Number (This specifies a distance with a unit of 1 mil.)
AsMM(ArcRadius) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**

ArcRadius = 300

Returns all Arc objects that have a Radius property which is equal to 300 current measurement units.

ArcRadius >= 15.24

Returns all Arc objects that have a Radius property which is greater than or equal to 15.24 current measurement units.

AsMils(ArcRadius) > 500

Returns all Arc objects that have a Radius property which is greater than 500 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(ArcRadius) Between 10.16 And 17.78

Returns all Arc objects that have a Radius property which is greater than or equal to 10.16 mm and less than or equal to 17.78 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

ArcRadius <> 550

Returns all Arc objects that have a Radius property which is not equal to 550 current measurement units.

AsMils(ArcRadius) <= 650

Returns all Arc objects that have a Radius property which is less than or equal to 650 mil.

AsMM(ArcRadius) < 11.43

Returns all Arc objects that have a Radius property which is less than 11.43 mm.

**See Also**

ArcStartAngle Field
ArcStopAngle Field

---

**ArcStartAngle Field**

**Description**

Returns all Arc objects having a Start Angle property that complies with the Query.

Note: The ArcStartAngle property is only defined for Arc objects.

**Syntax**

ArcStartAngle : Number

The number returned specifies the Start Angle in units of degrees.

**Examples**

ArcStartAngle = 30.0

Returns all Arc objects that have a Start Angle property which is 30.0 degrees.

ArcStartAngle > 60.0

Returns all Arc objects that have a Start Angle property which is greater than 60.0 degrees.

ArcStartAngle Between 30.0 And 150.0

Returns all Arc objects that have a Start Angle property which is greater than or equal to 30.0 degrees and less than or equal to 150.0 degrees.

ArcStartAngle <= 45.0

Returns all Arc objects that have a Start Angle property which is less than or equal to 45.0 degrees.

ArcStartAngle Between 315 And 360 || ArcStartAngle Between 0 And 45

Returns all Arc objects that have a Start Angle property which is between 315 degrees and 45 degrees.

**See Also**

ArcRadius Field
ArcStopAngle Field
**ArcStopAngle Field**

**Description**
Returns all Arc objects having a Stop Angle property that complies with the Query.

Note: The ArcStopAngle property is only defined for Arc objects.

**Syntax**

ArcStopAngle : Number

The number returned specifies the Stop Angle in units of degrees.

**Examples**

ArcStopAngle = 30.0
Returns all Arc objects that have a Stop Angle property which is 30.0 degrees.

ArcStopAngle > 60.0
Returns all Arc objects that have a Stop Angle property which is greater than 60.0 degrees.

ArcStopAngle Between 30.0 And 150.0
Returns all Arc objects that have a Stop Angle property which is greater than or equal to 30.0 degrees and less than or equal to 150.0 degrees.

ArcStopAngle <= 45.0
Returns all Arc objects that have a Stop Angle property which is less than or equal to 45.0 degrees.

ArcStopAngle Between 315 And 360 || ArcStopAngle Between 0 And 45
Returns all Arc objects that have a Stop Angle property which is between 315 degrees and 45 degrees.

**See Also**

ArcRadius Field
ArcStartAngle Field
AsMils Membership Check

Description
This converts a coordinate to mils. This keyword is used in conjunction with the keywords which are compared with a Number specifying a distance, and stipulates that Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

Syntax
AsMils(Number) : Number

Examples
AsMils(ArcRadius) = 250
Returns all Arc objects that have a Radius property which is 250 mil.
AsMils(Height) Between 300 And 500
Returns all Component objects that have a Height property which is greater than or equal to 300 mil and less than or equal to 500 mil.
AsMils(PolygonGridSize) >= 10
Returns all Polygon objects that have a Grid Size property which is greater than or equal to 10 mil.
AsMils(Width) < 50
Returns all Arc and Track objects that have a Width property which is less than 50 mil.

See Also
AsMM Membership Check

AsMM Membership Check

Description
This converts coordinate to millimetres and this keyword is used in conjunction with the keywords which are compared with a Number specifying a distance, and stipulates that a Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

Syntax
AsMM(Number) : Number

Examples
AsMM(ArcRadius) = 6.35
Returns all Arc objects that have a Radius property which is 6.35 mm.
AsMM(Height) Between 7.62 And 12.7
Returns all Component objects that have a Height property which is greater than or equal to 7.62 mm and less than or equal to 12.7 mm.
AsMM(PolygonGridSize) >= 0.254
Returns all Polygon objects that have a Grid Size property which is greater than or equal to 0.254 mm.
AsMM(Width) < 1.27
Returns all Arc and Track objects that have a Width property which is less than 1.27 mm.

See Also
AsMils Membership Check
AutoPosition Field

Description
Returns all Text objects having an Autoposition property that complies with the Query.
Note: The AutoPosition property is only defined for Designator and Comment Text objects (which are child objects of Component objects).

Syntax
AutoPosition = AutoPosition_String
AutoPosition <> AutoPosition_String

AutoPosition_String must be one of the strings from the following list:
'Center', 'Center-Above', 'Center-Below', 'Left-Above', 'Left-Below', 'Left-Center', 'Manual', 'Right-Above', 'Right-Below', 'Right-Center'

Note: The single quote characters ('') shown at the start and end of each AutoPosition_String are both mandatory.

Examples
AutoPosition = 'Center'
Returns all (Designator and Comment) Text objects that have a 'Center' Autoposition property.
AutoPosition <> 'Manual'
Returns all objects except (Designator and Comment) Text objects that have a 'Manual' Autoposition property. (Only Designator and Comment Text objects have an AutoPosition property, so all remaining types of objects do not have a 'Manual' AutoPosition property, and are thus also returned by this Query.)

BigColumnCount Attribute Check

Description
Returns Component objects whose count of "big" columns of pads complies with the Query; all of their child objects are also returned. In this context, each "big" column of pads contains at least four pads.

Syntax
BigColumnCount : Number

Examples
BigColumnCount = 8
Returns all Components that have exactly 8 "big" columns of pads; all of their child objects are also returned.
BigColumnCount Between 6 And 10
Returns all Components that have between 6 and 10 "big" columns of pads; all of their child objects are also returned.
BigColumnCount <= 4
Returns all Components that have up to 4 "big" columns of pads; all of their child objects are also returned. This Query also returns all objects which are not child objects of any Component object. (Such objects do not have an explicit BigColumnCount property, so that is assessed as being zero, which is less than 4.)

See Also
BigRowCount
ColumnCount
RowCount

BigRowCount Attribute Check

Description
Returns Component objects whose count of "big" rows of pads complies with the Query; all of their child objects are also returned. In this context, each "big" row of pads contains at least four pads.

Syntax
BigRowCount : Number

Examples
BigRowCount = 8
Returns all Components that have exactly 8 "big" rows of pads; all of their child objects are also returned.
BigRowCount Between 6 And 10
Returns all Components that have between 6 and 10 "big" rows of pads; all of their child objects are also returned.
BigRowCount <= 4
Returns all Components that have up to 4 "big" rows of pads; all of their child objects are also returned. This Query also returns all objects which are not child objects of any Component object. (Such objects do not have an explicit BigRowCount property, so that is assessed as being zero, which is less than 4.)

See Also
BigColumnCount
ColumnCount
RowCount

BodyProjection Field

Description
Returns all component objects having a BodyProjection property that complies with the Query. Note a body projection is the top or bottom layer the component is placed on.

Syntax
BodyProjection= BodyProjection_String
BodyProjection<> BodyProjection_String

BodyProjection_String must be one of the strings from the following list:
'Top Side', 'Bottom Side'

Note: The single quote characters (') shown at the start and end of each BodyProjection_String are both mandatory.

Examples
BodyProjection = 'Bottom Side'
Returns all component objects that have a 'Bottom Side' BodyProjection property only.

ChannelOffset Field

Description
Returns all Component objects having a Channel Offset property that complies with the Query.

Note: The child objects of these component objects are not also returned.
Note: The ChannelOffset property is only defined for Component objects.

Syntax
ChannelOffset : Numeric_String

ChannelOffset : Number {If all non-Component objects are totally excluded by one or more preceding tokens within the Query.}
The ChannelOffset keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all non-Component objects (which do not have a Channel Offset property) have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:
ChannelOffset <> ''
ChannelOffset > ''
ObjectKind = 'Component'
IsComponent

The usage of such tokens is demonstrated in the examples that follow.

Examples
ChannelOffset = '-1'
ChannelOffset <> '' && ChannelOffset = -1
Returns all Components that have a Channel Offset property of -1.
ChannelOffset > '' && ChannelOffset Between 2 And 10
Returns all Components that have a Channel Offset property which is greater than or equal to 2 and less than or equal to 10.
ObjectKind = 'Component' && ChannelOffset >= 3
Returns all Components that have a Channel Offset property which is greater than or equal to 3.
IsComponent && ChannelOffset < 12
Returns all Components that have a Channel Offset property which is less than 12.
ChannelOffset <> '-1'
Returns all objects except Components that have a Channel Offset property of -1. (Only Component objects have a Channel Offset property, so all remaining types of objects do not have a ChannelOffset property of 1, and are thus also returned by this Query.)
IsComponent && ChannelOffset <> -1
Returns all Components that do not have a Channel Offset property of 1.

**CharacteristicImpedance Impedance**

**Description**
Specifies the characteristic impedance required between a track on a Signal layer and the copper on an (adjacent) Internal Plane layer (or the copper on (adjacent) surrounding Internal Plane layers) within a multi-layer PCB. That characteristic impedance has an influence upon what width that track must subsequently have.

Note: This is an alias for Z0 (Impedance).

**Syntax**
CharacteristicImpedance : Number
The characteristic impedance is specified in units of ohms.

**Examples**

```
((5.98*TraceToPlaneDistance/EXP(CharacteristicImpedance*SQRT(Er+1.41)/87))- TraceHeight)*1.25
```
The default Query for specifying the required width of a track on the Signal layer, when the Microstrip option is selected.

```
((1.9*PlaneToPlaneDistance/EXP(CharacteristicImpedance/(60/SQRT(Er))))- TraceHeight)*1.25
```
The default Query for specifying the required width of a track on the intermediate Signal layer, when the Stripline option is selected.

**ClassMembers Field**

**Description**
Returns all Class objects having a ClassMembers property that complies with the Query.

Note: The ClassMembers property is only defined for Class objects.

**Syntax**
ClassMembers = ClassMembers_String
ClassMembers <> ClassMembers_String

ClassMember_String must be one of the members in a defined class for the PCB document:

Note: The single quote characters (') shown at the start and end of each ClassType_String are both mandatory.

**Examples**

ClassMembers = 'Gnd'
Returns all members of a class associated with a Gnd member.
**ClassType Field**

**Description**
Returns all Class objects having a Class Type property that complies with the Query.

*Note: The ClassType property is only defined for Class objects.*

**Syntax**

ClassType = ClassType_String
ClassType <> ClassType_String

ClassType_String must be one of the strings from the following list:
'Component Class', 'Design Channel', 'From To Class', 'Layer Class', 'Net Class', 'Pad Class'

*Note: The single quote characters (') shown at the start and end of each ClassType_String are both mandatory.*

**Examples**

ClassType = 'Component Class'
Returns all Component Class objects.

ClassType <> 'Layer Class'
Returns all objects *except* Layer Class objects. (Only Class objects have a Class Type property, so all remaining types of objects do not have a 'Layer Class' ClassType property, and are thus also returned by this Query.)

**Color Field (PCB)**

**Description**
Returns all Net objects having a Color property that complies with the Query.

*Note: The child objects of these Net objects are *not* also returned.*

*Note: The Color property is only defined for Net objects.*

**Syntax**

Color : Numeric_String
Color : Number {If all non-Net objects are totally excluded by one or more preceding tokens within the Query.}

The Color keyword can *always* be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but *only* when all non-Net objects (which do *not* have a Color property) have been *totally* excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:

Color <> ''
Color > ''
ObjectKind = 'Net'
IsNet

The usage of such tokens is demonstrated in the examples that follow.

The color corresponding to a particular number can be deduced from the following relationship:

65536 * Blue + 256 * Green + Red

where the Blue, Green, and Red components each have a value between 0 and 255 (inclusive).

*When* the Color keyword can be used in conjunction with a Number (as previously described), it is additionally possible to individually evaluate the values of each of the Color property's Blue, Green, and Red components, by using the following constructs:

Color Div 65536 {Blue component}
Color Div 256 Mod 256 {Green component}
Color Mod 256 {Red component}

The usage of these constructs is also demonstrated in the examples that follow.

**Examples**

Color = '0'
Color <> '' && Color = 0
Returns all Net objects that have a black Color property. (A Color property of 0, as defined above, corresponds to values of Blue = 0, Green = 0, and Red = 0).

\[
\text{Color} = '8421376'
\]

\[
\text{Color} > '0' \&\& \text{Color} = 8421376
\]

Returns all Net objects that have a Color property of 8421376 (as defined above, which corresponds to values of Blue = 128, Green = 128, and Red = 0).

\[
\text{Color} <> '16777215'
\]

Returns all objects except Net objects that have a (maximum intensity) white Color property. (A Color property of 16777215, as defined above, corresponds to values of Blue = 255, Green = 255, and Red = 255).

\[
\text{ObjectKind} = 'Net' \&\& \text{Color} <> 16777215
\]

Returns all Net objects that do not have a (maximum intensity) white Color property.

\[
\text{IsNet} \&\& \text{Color Div 65536} >= 96 \&\& \text{Color Div 256 Mod 256 Between 64 And 192} \&\& \text{Color Mod 256} <= 128
\]

Returns all Net objects that have a Color property whose Blue component has a value which is greater than or equal to 96, and whose Green component has a value which is greater than or equal to 64 and less than or equal to 192, and whose Red component has a value which is less than or equal to 128.

\[
\text{IsNet} \&\& \text{Color Div 65536} <= 192
\]

Returns all Net objects whose Color property's Blue component has a value which is less than or equal to 192.

\[
\text{IsNet} \&\& \text{Color Mod 256 Between 64 And 192}
\]

Returns all Net objects whose Color property's Red component has a value which is greater than or equal to 64 and less than or equal to 192.

---

**ColumnCount Attribute Check**

**Description**

Returns Component objects whose count of columns of pads complies with the Query; all of their child objects are also returned.

**Syntax**

\[
\text{ColumnCount} : \text{Number}
\]

**Examples**

\[
\text{ColumnCount} = 8
\]

Returns all Components that have exactly 8 columns of pads; all of their child objects are also returned.

\[
\text{ColumnCount Between 6 And 10}
\]

Returns all Components that have between 6 and 10 columns of pads; all of their child objects are also returned.

\[
\text{ColumnCount} <= 4
\]

Returns all Components that have up to 4 columns of pads; all of their child objects are also returned. This Query also returns all objects which are not child objects of any Component object. (Such objects do not have an explicit ColumnCount property, so that is assessed as being zero, which is less than 4.)

**See Also**

BigColumnCount

BigRowCount

RowCount
Comment Field

Description
Returns all Component objects having a Component Comment property that complies with the Query.
Note: The child objects of these Component objects are not also returned.
Note: The Comment property is only defined for Component objects.

Syntax
Comment : String

Examples
Comment = '100nF'
Returns all Components that have a Component Comment property of '100nF'.
Comment <> '100nF'
Returns all objects except Component objects that have a Component Comment property of '100nF'. (Only Component objects have a Comment property, so all remaining types of objects do not have a Comment property of '100nF', and are thus also returned by this Query.)

CompCenterOfPadsX Attribute Check

Description
Returns components when the X-Coordinate of the centroid of their pads complies with the Query; all of their child objects are also returned.
At present, all distances always use units of mils. (1 mil = 0.001 inch)

Syntax
CompCenterOfPadsX : Distance_String

Examples
CompCenterOfPadsX = 1200
Returns all Component objects (and all of their child objects) when the X-Coordinate of the centroid of their pads is equal to 1200 mil.
CompCenterOfPadsX >= 1500
Returns all Component objects (and all of their child objects) when the X-Coordinate of the centroid of their pads is greater than or equal to 1500 mil.
CompCenterOfPadsX Between 1000 And 1400
Returns all Component objects (and all of their child objects) when the X-Coordinate of the centroid of their pads is greater than or equal to 1000 mil and less than or equal to 1400 mil.
CompCenterOfPadsX < 800
Returns all objects except Component objects (and all of their child objects) when the X-Coordinate of the centroid of their pads is greater than or equal to 800 mil.

See Also
CompCenterOfPadsY
CompCenterX
CompCenterY
CompCenterOfPadsY Attribute Check

Description
Returns components when the Y-Coordinate of the centroid of their pads complies with the Query; all of their child objects are also returned.

At present, all distances always use units of mils. (1 mil = 0.001 inch)

Syntax
CompCenterOfPadsY : Distance_String

Examples
CompCenterOfPadsY = 1200
Returns all Component objects (and all of their child objects) when the Y-Coordinate of the centroid of their pads is equal to 1200 mil.

CompCenterOfPadsY >= 1500
Returns all Component objects (and all of their child objects) when the Y-Coordinate of the centroid of their pads is greater than or equal to 1500 mil.

CompCenterOfPadsY Between 1000 And 1400
Returns all Component objects (and all of their child objects) when the Y-Coordinate of the centroid of their pads is greater than or equal to 1000 mil and less than or equal to 1400 mil.

CompCenterOfPadsY < 800
Returns all objects except Component objects (and all of their child objects) when the Y-Coordinate of the centroid of their pads is greater than or equal to 800 mil.

See Also
CompCenterOfPadsX
CompCenterX
CompCenterY

CompCenterX Attribute Check

Description
Returns components when the X-Coordinate of the centroid of all of their child objects complies with the Query; all of their child objects are also returned.

At present, all distances always use units of mils. (1 mil = 0.001 inch)

Syntax
CompCenterX : Distance_String

Examples
CompCenterX = 1200
Returns all Component objects (and all of their child objects) when the X-Coordinate of the centroid of all of their child objects is equal to 1200 mil.

CompCenterX >= 1500
Returns all Component objects (and all of their child objects) when the X-Coordinate of the centroid of all of their child objects is greater than or equal to 1500 mil.

CompCenterX Between 1000 And 1400
Returns all Component objects (and all of their child objects) when the X-Coordinate of the centroid of all of their child objects is greater than or equal to 1000 mil and less than or equal to 1400 mil.

CompCenterX < 800
Returns all objects except Component objects (and all of their child objects) when the X-Coordinate of the centroid of all of their child objects is greater than or equal to 800 mil.

See Also
CompCenterOfPadsX
CompCenterOfPadsY
CompCenterY

**CompCenterY Attribute Check**

**Description**
Returns components when the Y-Coordinate of the centroid of all of their child objects complies with the Query; all of their child objects are also returned.

At present, all distances always use units of mils. (1 mil = 0.001 inch)

**Syntax**
CompCenterY : Distance_String

**Examples**
CompCenterY = 1200
Returns all Component objects (and all of their child objects) when the Y-Coordinate of the centroid of all of their child objects is equal to 1200 mil.

CompCenterY >= 1500
Returns all Component objects (and all of their child objects) when the Y-Coordinate of the centroid of all of their child objects is greater than or equal to 1500 mil.

CompCenterY Between 1000 And 1400
Returns all Component objects (and all of their child objects) when the Y-Coordinate of the centroid of all of their child objects is greater than or equal to 1000 mil and less than or equal to 1400 mil.

CompCenterY < 800
Returns all objects except Component objects (and all of their child objects) when the Y-Coordinate of the centroid of all of their child objects is greater than or equal to 800 mil.

**See Also**
CompCenterOfPadsX
CompCenterOfPadsY
CompCenterX

**Component Field**

**Description**
Returns all of the child objects of any Component object whose Name property complies with the Query.

Note: The parent object of each component is not also returned.

**Syntax**
Component : String

**Examples**
Component = 'C1'
Returns all of the child objects of any Component that has a Name property of 'C1'.

Component Like 'C?*'
Returns all of the child objects of any Component that has a Name property which starts with 'C', and which includes at least one additional character.

Component <> 'C1'
Returns all objects except the child objects of any Component object that has a Name property of 'C1'. (Only Component objects have a Component property, so all objects which are not part of some Component object do not have a Component property of 'C1', and are thus also returned by this Query.)

Component = 'Free'
Returns all objects except the child objects of Component objects. (Any object which is not a child object of a Component object has a Component property of 'Free', and is thus also returned by this Query.)

ComponentStringsLocked

Description
Returns all Component objects having a ComponentStringsLocked property that complies with the Query.
Note: The child objects of these Component objects are not also returned.
Note: The ComponentStringsLocked property is only defined for Component objects.

Syntax
ComponentStringsLocked : Boolean_String

Examples
ComponentStringsLocked
ComponentStringsLocked= 'True'
Returns all strings associated with components that are locked.
Not ComponentStringsLocked
ComponentStringsLocked= 'False'
Returns all strings associated with components which are not locked.

ComponentType Field (PCB)

Description
Returns all Component objects having a Component Type property that complies with the Query.
Note: The child objects of these Component objects are not also returned.
Note: The ComponentType property is only defined for Component objects.

Syntax
ComponentType = ComponentType_String
ComponentType <> ComponentType_String
ComponentType_String must be one of the strings from the following list:
'Graphical', 'Mechanical', 'Net Tie', 'Net Tie (In BOM)', 'Standard', 'Standard (No BOM)'
Note: The single quote characters (') shown at the start and end of each ComponentType_String are both mandatory.

Examples
ComponentType = 'Graphical'
Returns all Component objects that have a 'Graphical' Component Type property.
ComponentType <> 'Standard'
Returns all objects except Component objects that have a 'Standard' Component Type property. (Only Component objects have a Component Type property, so all remaining types of objects do not have a 'Standard' ComponentType property, and are thus also returned by this Query.)
CompPinCount Attribute Check

Description
Returns Component objects whose count of child Pad objects complies with the Query; all of their child objects are also returned.

Syntax
CompPinCount : Number

Examples
CompPinCount = 10
Returns all Components that have exactly 10 child Pad objects; all of their child objects are also returned.

CompPinCount Between 20 And 30
Returns all Components that have between 20 and 30 child Pad objects; all of their child objects are also returned.

CompPinCount <= 8
Returns all Components that have up to 8 Pad objects; all of their child objects are also returned. This Query also returns all objects which are not child objects of any Component object. (Such objects do not have an explicit CompPinCount property, so that is assessed as being zero, which is less than 8.)

See Also
CompPinCount_Bottom
CompPinCount_Internal
CompPinCount_Surface
CompPinCount_ThruHole
CompPinCount_Top

CompPinCount_Bottom Attribute Check

Description
Returns Component objects whose count of child Pad objects on the Bottom (copper) layer complies with the Query; all of their child objects are also returned.

Syntax
CompPinCount_Bottom : Number

Examples
CompPinCount_Bottom = 10
Returns all Components that have exactly 10 child Pad objects on the Bottom (copper) layer; all of their child objects are also returned.

CompPinCount_Bottom Between 20 And 30
Returns all Components that have between 20 and 30 child Pad objects on the Bottom (copper) layer; all of their child objects are also returned.

CompPinCount_Bottom <= 8
Returns all Components that have up to 8 Pad objects on the Bottom (copper) layer; all of their child objects are also returned. This Query also returns all objects which are not child objects of any Component object. (Such objects do not have an explicit CompPinCount_Bottom property, so that is assessed as being zero, which is less than 8.)

See Also
CompPinCount
CompPinCount_Internal
CompPinCount_Surface
CompPinCount_ThruHole
CompPinCount_Top
**CompPinCount_Internal Attribute Check**

**Description**
Returns Component objects whose count of child Pad objects on the internal (copper) layers complies with the Query; all of their child objects are also returned.

**Syntax**
CompPinCount_Internal : Number

**Examples**
CompPinCount_Internal = 10
Returns all Components that have exactly 10 child Pad objects on the internal (copper) layers; all of their child objects are also returned.

CompPinCount_Internal Between 20 And 30
Returns all Components that have between 20 and 30 child Pad objects on the internal (copper) layers; all of their child objects are also returned.

CompPinCount_Internal <= 8
Returns all Components that have up to 8 Pad objects on the internal (copper) layers; all of their child objects are also returned. This Query also returns all objects which are not child objects of any Component object. (Such objects do not have an explicit CompPinCount_Internal property, so that is assessed as being zero, which is less than 8.)

**See Also**
CompPinCount
CompPinCount_Bottom
CompPinCount_Surface
CompPinCount_ThruHole
CompPinCount_Top

**CompPinCount_Surface Attribute Check**

**Description**
Returns Component objects whose count of child Pad objects on the Top and Bottom (copper) layers complies with the Query; all of their child objects are also returned.

**Syntax**
CompPinCount_Surface : Number

**Examples**
CompPinCount_Surface = 10
Returns all Components that have exactly 10 child Pad objects on the Top and Bottom (copper) layers; all of their child objects are also returned.

CompPinCount_Surface Between 20 And 30
Returns all Components that have between 20 and 30 child Pad objects on the Top and Bottom (copper) layers; all of their child objects are also returned.

CompPinCount_Surface <= 8
Returns all Components that have up to 8 Pad objects on the Top and Bottom (copper) layers; all of their child objects are also returned. This Query also returns all objects which are not child objects of any Component object. (Such objects do not have an explicit CompPinCount_Surface property, so that is assessed as being zero, which is less than 8.)

**See Also**
CompPinCount
CompPinCount_Bottom
CompPinCount_Internal
CompPinCount_ThruHole
CompPinCount_Top
**CompPinCount_ThruHole Attribute Check**

**Description**
Returns Component objects whose count of child Pad objects on the Multi-Layer layer complies with the Query; all of their child objects are also returned.

**Syntax**

```plaintext
CompPinCount_ThruHole : Number
```

**Examples**

- **CompPinCount_ThruHole = 10**
  Returns all Components that have exactly 10 child Pad objects on the Multi-Layer layer; all of their child objects are also returned.

- **CompPinCount_ThruHole Between 20 And 30**
  Returns all Components that have between 20 and 30 child Pad objects on the Multi-Layer layer; all of their child objects are also returned.

- **CompPinCount_ThruHole <= 8**
  Returns all Components that have up to 8 Pad objects on the Multi-Layer layer; all of their child objects are also returned. This Query also returns all objects which are not child objects of any Component object. (Such objects do not have an explicit CompPinCount_ThruHole property, so that is assessed as being zero, which is less than 8.)

**See Also**

- CompPinCount
- CompPinCount_Bottom
- CompPinCount_Internal
- CompPinCount_Surface
- CompPinCount_Top

**CompPinCount_Top Attribute Check**

**Description**
Returns Component objects whose count of child Pad objects on the Top (copper) layer complies with the Query; all of their child objects are also returned.

**Syntax**

```plaintext
CompPinCount_Top : Number
```

**Examples**

- **CompPinCount_Top = 10**
  Returns all Components that have exactly 10 child Pad objects on the Top (copper) layer; all of their child objects are also returned.

- **CompPinCount_Top Between 20 And 30**
  Returns all Components that have between 20 and 30 child Pad objects on the Top (copper) layer; all of their child objects are also returned.

- **CompPinCount_Top <= 8**
  Returns all Components that have up to 8 Pad objects on the Top (copper) layer; all of their child objects are also returned. This Query also returns all objects which are not child objects of any Component object. (Such objects do not have an explicit CompPinCount_Top property, so that is assessed as being zero, which is less than 8.)

**See Also**

- CompPinCount
- CompPinCount_Bottom
- CompPinCount_Internal
- CompPinCount_Surface
- CompPinCount_ThruHole
CoordinateDimensionLineWidth Field

Description
Returns all Coordinate and Original Dimension objects having a Line Width (Coord/Dimen) property that complies with the Query.
Note: The child objects of these Coordinate and Original Dimension objects are not also returned.
Note: The CoordinateDimensionLineWidth property is only defined for Coordinate and Original Dimension objects.

Syntax
CoordinateDimensionLineWidth : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(CoordinateDimensionLineWidth) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(CoordinateDimensionLineWidth) : Number {This specifies a distance with a unit of 1 mm.}

Examples
CoordinateDimensionLineWidth = 10
Returns all Coordinate and Original Dimension objects that have a Line Width (Coord/Dimen) property which is 10 current measurement units.
CoordinateDimensionLineWidth >= 0.2286
Returns all Coordinate and Original Dimension objects that have a Line Width (Coord/Dimen) property which is greater than or equal to 0.2286 current measurement units.
AsMils(CoordinateDimensionLineWidth) Between 8 And 12
Returns all Coordinate and Original Dimension objects that have a Line Width (Coord/Dimen) property which is greater than or equal to 8 mil and less than or equal to 12 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(CoordinateDimensionLineWidth) < 0.2794
Returns all Coordinate and Original Dimension objects that have a Line Width (Coord/Dimen) property which is less than 0.2794 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

CoordinateDimensionSize Field

Description
Returns all Coordinate, Center Dimension, Leader Dimension, and Original Dimension objects having a Size (Coord/Dimen) property that complies with the Query.
Note: The child objects of these Coordinate, Center Dimension, Leader Dimension, and Original Dimension objects are not also returned.
Note: The CoordinateDimensionSize property is only defined for Coordinate, Center Dimension, Leader Dimension, and Original Dimension objects.

Syntax
CoordinateDimensionSize : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(CoordinateDimensionSize) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(CoordinateDimensionSize) : Number {This specifies a distance with a unit of 1 mm.}
Examples
CoordinateDimensionSize = 10
Returns all Coordinate, Center Dimension, Leader Dimension, and Original Dimension objects that have a Size (Coord/Dimen) property which is 10 current measurement units.

CoordinateDimensionSize >= 0.2286
Returns all Coordinate, Center Dimension, Leader Dimension, and Original Dimension objects that have a Size (Coord/Dimen) property which is greater than or equal to 0.2286 current measurement units.

AsMils(CoordinateDimensionSize) Between 8 And 12
Returns all Coordinate, Center Dimension, Leader Dimension, and Original Dimension objects that have a Size (Coord/Dimen) property which is greater than or equal to 8 mil and less than or equal to 12 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(CoordinateDimensionSize) < 0.2794
Returns all Coordinate, Center Dimension, Leader Dimension, and Original Dimension objects that have a Size (Coord/Dimen) property which is less than 0.2794 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

CoordinateDimensionUnitStyle Field

Description
Returns all Coordinate and Original Dimension objects having a Unit Style property that complies with the Query.

Note: The child objects of these Coordinate and Original Dimension objects are not also returned.

Note: The CoordinateDimensionUnitStyle property is only defined for Coordinate and Original Dimension objects.

Syntax
CoordinateDimensionUnitStyle = UnitStyle_String
CoordinateDimensionUnitStyle <> UnitStyle_String

UnitStyle_String must be one of the strings from the following list:
'Brackets', 'None', 'Normal'

Note: The single quote characters (') shown at the start and end of each UnitStyle_String are both mandatory.

Examples
CoordinateDimensionUnitStyle = 'Brackets'
Returns Coordinate and Original Dimension objects that have a 'Brackets' Unit Style property.

CoordinateDimensionUnitStyle <> 'Normal'
Returns all objects except Coordinate and Original Dimension objects that have a 'Normal' Unit Style property. (Only Coordinate and Original Dimension objects have a Unit Style property, so all remaining types of objects do not have a 'Normal' CoordinateDimensionUnitStyle property, and are thus also returned by this Query.)
DielectricConstant Impedance

Description
Specifies the relative dielectric constant of the core (non-copper) material which a PCB is composed of. That value has an influence upon the characteristic impedance between a track on a Signal layer and the copper on an (adjacent) Internal Plane layer (or the copper on (adjacent) surrounding Internal Plane layers), or the width of such a track when a particular characteristic impedance is required.

Note: This is an alias for Er (Impedance).

Syntax
DielectricConstant : Number
The relative dielectric constant is the ratio of the material's dielectric constant to the dielectric constant of empty space.

Examples
(87/SQRT(DielectricConstant+1.41))*LN(5.98*TraceToPlaneDistance/(0.8*TraceWidth + TraceHeight))
A modified form of the default Query (in which DielectricConstant has replaced Er) for specifying the characteristic impedance between a track on the Signal layer and the copper on the (adjacent) Internal Plane layer, when the Microstrip option is selected.

(60/SQRT(DielectricConstant))*LN((1.9*PlaneToPlaneDistance)/(0.8*TraceWidth + TraceHeight))
A modified form of the default Query (in which DielectricConstant has replaced Er) for specifying the characteristic impedance between a track on the intermediate Signal layer and the copper on the two surrounding Internal Plane layers, when the Stripline option is selected.

Description Field

Description
Returns all Library Component objects having a Description property that complies with the Query.

Syntax
Description: String

Examples
Description = 'F*'
Returns all components or footprints that have a Description property with the first character of the string starting with F.

DimensionArrowGap Field

Description
Returns all Leader Dimension objects having an Arrow Gap property that complies with the Query.
Note: The child objects of these Leader Dimension objects are not also returned.
Note: The DimensionArrowGap property is only defined for Leader Dimension objects.

Syntax
DimensionArrowGap : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(DimensionArrowGap) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(DimensionArrowGap) : Number {This specifies a distance with a unit of 1 mm.}

Examples
DimensionArrowGap = 10
Returns all Leader Dimension objects that have an Arrow Gap property which is 10 current measurement units.

DimensionArrowGap >= 0.2286
Returns all Leader Dimension objects that have an Arrow Gap property which is greater than or equal to 0.2286 current measurement units.

AsMils(DimensionArrowGap) Between 8 And 12
Returns all Leader Dimension objects that have an Arrow Gap property which is greater than or equal to 8 mil and less than or equal to 12 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(DimensionArrowGap) < 0.2794
Returns all Leader Dimension objects that have an Arrow Gap property which is less than 0.2794 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

DimensionArrowLength Field

Description
Returns all Angular Dimension, Baseline Dimension, Linear Diameter Dimension, Linear Dimension, and Radial Diameter Dimension objects having an Arrow Length property that complies with the Query.

Note: The child objects of these Angular Dimension, Baseline Dimension, Linear Diameter Dimension, Linear Dimension, and Radial Diameter Dimension objects are not also returned.

Note: The DimensionArrowLength property is only defined for Angular Dimension, Baseline Dimension, Linear Diameter Dimension, Linear Dimension, and Radial Diameter Dimension objects.

Syntax
DimensionArrowLength : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(DimensionArrowLength) : Number (This specifies a distance with a unit of 1 mil.)
AsMM(DimensionArrowLength) : Number (This specifies a distance with a unit of 1 mm.)

Examples
DimensionArrowLength = 100
Returns all Angular Dimension, Baseline Dimension, Linear Diameter Dimension, Linear Dimension, and Radial Diameter Dimension objects that have an Arrow Length property which is 100 current measurement units.

DimensionArrowLength >= 2.286
Returns all Angular Dimension, Baseline Dimension, Linear Diameter Dimension, Linear Dimension, and Radial Diameter Dimension objects that have an Arrow Length property which is greater than or equal to 2.286 current measurement units.

AsMils(DimensionArrowLength) Between 80 And 120
Returns all Angular Dimension, Baseline Dimension, Linear Diameter Dimension, Linear Dimension, and Radial Diameter Dimension objects that have an Arrow Length property which is greater than or equal to 80 mil and less than or equal to 120 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(DimensionArrowLength) < 2.794
Returns all Angular Dimension, Baseline Dimension, Linear Diameter Dimension, Linear Dimension, and Radial Diameter Dimension objects that have an Arrow Length property which is less than 2.794 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
DimensionArrowLineWidth Field

Description
Returns all Angular Dimension, Baseline Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects having an Arrow Line Width property that complies with the Query.

Note: The child objects of these Angular Dimension, Baseline Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects are not also returned.

Note: The DimensionArrowLineWidth property is only defined for Angular Dimension, Baseline Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects.

Syntax
DimensionArrowLineWidth : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(DimensionArrowLineWidth) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(DimensionArrowLineWidth) : Number {This specifies a distance with a unit of 1 mm.}

Examples
DimensionArrowLineWidth = 10
Returns all Angular Dimension, Baseline Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have an Arrow Line Width property which is 10 current measurement units.
DimensionArrowLineWidth >= 0.2286
Returns all Angular Dimension, Baseline Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have an Arrow Line Width property which is greater than or equal to 0.2286 current measurement units.
AsMils(DimensionArrowLineWidth) Between 8 And 12
Returns all Angular Dimension, Baseline Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have an Arrow Line Width property which is greater than or equal to 8 mil and less than or equal to 12 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(DimensionArrowLineWidth) < 0.2794
Returns all Angular Dimension, Baseline Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have an Arrow Line Width property which is less than 0.2794 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

DimensionArrowPosition Field

Description
Returns all Angular Dimension, Baseline Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects having an Arrow Position property that complies with the Query.

Note: The child objects of these Angular Dimension, Baseline Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects are not also returned.

Note: The DimensionArrowPosition property is only defined for Angular Dimension, Baseline Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects.

Syntax
DimensionArrowPosition = ArrowPosition_String
DimensionArrowPosition <> ArrowPosition_String
ArrowPosition_String must be one of the strings from the following list:

'Inside', 'Outside'

Note: The single quote characters (') shown at the start and end of each ArrowPosition_String are both mandatory.

**Examples**

```
DimensionArrowPosition = 'Inside'
```

Returns all Angular Dimension, Baseline Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have an 'Inside' Arrow Position property.

```
DimensionArrowPosition <> 'Outside'
```

Returns all objects except Angular Dimension, Baseline Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have an 'Outside' Arrow Position property. (Only Angular Dimension, Baseline Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects have an Arrow Position property, so all remaining types of objects do not have an 'Outside' DimensionArrowPosition property, and are thus also returned by this Query.)

**DimensionArrowSize Field**

**Description**

Returns all Angular Dimension, Baseline Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects having an Arrow Size property that complies with the Query.

Note: The child objects of these Angular Dimension, Baseline Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects are not also returned.

Note: The DimensionArrowSize property is only defined for Angular Dimension, Baseline Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects.

**Syntax**

```
DimensionArrowSize : Number
```

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

```
AsMils(DimensionArrowSize) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(DimensionArrowSize) : Number {This specifies a distance with a unit of 1 mm.}
```

**Examples**

```
DimensionArrowSize = 60
```

Returns all Angular Dimension, Baseline Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have an Arrow Size property which is 60 current measurement units.

```
DimensionArrowSize >= 1.016
```

Returns all Angular Dimension, Baseline Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have an Arrow Size property which is greater than or equal to 1.016 current measurement units.

```
AsMils(DimensionArrowSize) Between 50 And 70
```

Returns all Angular Dimension, Baseline Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have an Arrow Size property which is greater than or equal to 50 mil and less than or equal to 70 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

```
AsMM(DimensionArrowSize) < 2.032
```

Returns all Angular Dimension, Baseline Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have an Arrow Size property which is less than 2.032 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
**DimensionExtensionLineWidth Field**

**Description**

Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, and Linear Dimension objects having an Extension Line Width property that complies with the Query.

Note: The child objects of these Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, and Linear Dimension objects are **not** also returned.

Note: The DimensionExtensionLineWidth property is only defined for Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, and Linear Dimension objects.

**Syntax**

**DimensionExtensionLineWidth : Number**

- If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
- If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

- **AsMils(DimensionExtensionLineWidth) : Number** {This specifies a distance with a unit of 1 mil.}
- **AsMM(DimensionExtensionLineWidth) : Number** {This specifies a distance with a unit of 1 mm.}

**Examples**

- **DimensionExtensionLineWidth = 10**
  
  Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, and Linear Dimension objects that have an Extension Line Width property which is 10 current measurement units.

- **DimensionExtensionLineWidth >= 0.2286**
  
  Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, and Linear Dimension objects that have an Extension Line Width property which is greater than or equal to 0.2286 current measurement units.

- **AsMils(DimensionExtensionLineWidth) Between 8 And 12**
  
  Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, and Linear Dimension objects that have an Extension Line Width property which is greater than equal to 8 mil and less than or equal to 12 mil. (The current value of the target file’s Measurement Unit property has no impact upon this outcome.)

- **AsMM(DimensionExtensionLineWidth) < 0.2794**
  
  Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, and Linear Dimension objects that have an Extension Line Width property which is less than 0.2794 mm. (The current value of the target file’s Measurement Unit property has no impact upon this outcome.)

---

**DimensionExtensionOffset Field**

**Description**

Returns all Angular Dimension, Baseline Dimension, Linear Diameter Dimension, and Linear Dimension objects having an Extension Offset property that complies with the Query.

Note: The child objects of these Angular Dimension, Baseline Dimension, Linear Diameter Dimension, and Linear Dimension objects are **not** also returned.

Note: The DimensionExtensionOffset property is only defined for Angular Dimension, Baseline Dimension, Linear Diameter Dimension, and Linear Dimension objects.

**Syntax**

**DimensionExtensionOffset : Number**

- If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
- If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(DimensionExtensionOffset) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(DimensionExtensionOffset) : Number {This specifies a distance with a unit of 1 mm.}

Examples

DimensionExtensionOffset = 10
Returns all Angular Dimension, Baseline Dimension, Linear Diameter Dimension, and Linear Dimension objects that have an Extension Offset property which is 10 current measurement units.

DimensionExtensionOffset &ge; 0.2286
Returns all Angular Dimension, Baseline Dimension, Linear Diameter Dimension, and Linear Dimension objects that have an Extension Offset property which is greater than or equal to 0.2286 current measurement units.

AsMils(DimensionExtensionOffset) Between 8 And 12
Returns all Angular Dimension, Baseline Dimension, Linear Diameter Dimension, and Linear Dimension objects that have an Extension Offset property which is greater than or equal to 8 mil and less than or equal to 12 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(DimensionExtensionOffset) < 0.2794
Returns all Angular Dimension, Baseline Dimension, Linear Diameter Dimension, and Linear Dimension objects that have an Extension Offset property which is less than 0.2794 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

DimensionExtensionPickGap Field

Description

Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, and Linear Dimension objects having an Extension Pick Gap property that complies with the Query.

Note: The child objects of these Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, and Linear Dimension objects are not also returned.

Note: The DimensionExtensionPickGap property is only defined for Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, and Linear Dimension objects.

Syntax

DimensionExtensionPickGap : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(DimensionExtensionPickGap) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(DimensionExtensionPickGap) : Number {This specifies a distance with a unit of 1 mm.}

Examples

DimensionExtensionPickGap = 10
Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, and Linear Dimension objects that have an Extension Pick Gap property which is 10 current measurement units.

DimensionExtensionPickGap &ge; 0.2286
Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, and Linear Dimension objects that have an Extension Pick Gap property which is greater than or equal to 0.2286 current measurement units.

AsMils(DimensionExtensionPickGap) Between 8 And 12
Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, and Linear Dimension objects that have an Extension Pick Gap property which is greater than or equal to 8 mil and less than or equal to 12 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(DimensionExtensionPickGap) < 0.2794
Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, and Linear Dimension objects that have an Extension Pick Gap property which is less than 0.2794 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

**DimensionTextFont Field**

**Description**

Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Original Dimension, Radial Diameter Dimension, and Radial Dimension objects having a Dimension Font property that complies with the Query.

Note: The child objects of these Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Original Dimension, Radial Diameter Dimension, and Radial Dimension objects are not also returned.

Note: The DimensionTextFont property is only defined for Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Original Dimension, Radial Diameter Dimension, and Radial Dimension objects.

**Syntax**

DimensionTextFont = Font_String
DimensionTextFont <> Font_String

Font_String must be one of the strings from the following list:

'Default', 'Serif', 'Sans Serif'

Note: The single quote characters ('') shown at the start and end of each Font_String are both mandatory.

**Examples**

DimensionTextFont = 'Default'

Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Original Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a 'Default' Dimension Font property.

DimensionTextFont <> 'Serif'

Returns all objects except Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Original Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a 'Serif' Dimension Font property. (Only Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Original Dimension, Radial Diameter Dimension, and Radial Dimension objects have a Dimension Font property, so all remaining types of objects do not have a 'Serif' DimensionTextFont property, and are thus also returned by this Query.)
DimensionTextFormat Field

Description
Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects having a Dimension Text format property that complies with the Query.

Note: The child objects of these Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects are not also returned.

Syntax
DimensionTextFormat = Text_Format

Text_Format string must be from one of the following list:
'None', 'Value Only', 'Normal', 'Brackets'

Note: The single quote characters (') shown at the start and end of each Text_Format string are both mandatory.

Examples
DimensionTextFormat = 'None'
Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Original Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension format property that have no values and units displayed.
DimensionTextFormat <> 'None'

DimensionTextGap Field

Description
Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects having a Dimension Text Gap property that complies with the Query.

Note: The child objects of these Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects are not also returned.

Note: The DimensionTextGap property is only defined for Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects.

Syntax
DimensionTextGap : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(DimensionTextGap) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(DimensionTextGap) : Number {This specifies a distance with a unit of 1 mm.}

Examples
DimensionTextGap = 10
Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Text Gap property which is 10 current measurement units.
DimensionTextGap >= 0.2286
Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Text Gap property which is greater than or equal to 0.2286 current measurement units.
AsMils(DimensionTextGap) Between 8 And 12
Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Text Gap property which is greater than or equal to 8 mil and less than or equal to 12 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

\[ \text{AsMM(DimensionTextGap)} < 0.2794 \]

Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Text Gap property which is less than 0.2794 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

**DimensionTextHeight Field**

**Description**

Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects having a Dimension Text Height property that complies with the Query.

Note: The child objects of these Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects are not also returned.

Note: The DimensionTextHeight property is only defined for Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects.

**Syntax**

```plaintext
DimensionTextHeight : Number
```

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

- **AsMils(DimensionTextHeight)** : Number (This specifies a distance with a unit of 1 mil.)
- **AsMM(DimensionTextHeight)** : Number (This specifies a distance with a unit of 1 mm.)

**Examples**

- **DimensionTextHeight = 60**
  Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Text Height property which is 60 current measurement units.

- **DimensionTextHeight >= 1.27**
  Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Text Height property which is greater than or equal to 1.27 current measurement units.

- **AsMils(DimensionTextHeight) Between 40 And 80**
  Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Text Height property which is greater than or equal to 40 mil and less than or equal to 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

- **AsMM(DimensionTextHeight) < 1.778**
  Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Text Height property which is less than 1.778 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
**DimensionTextLineWidth Field**

**Description**

Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects having a Dimension Text Line Width property that complies with the Query.

Note: The child objects of these Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects are not also returned.

Note: The DimensionTextLineWidth property is only defined for Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects.

**Syntax**

DimensionTextLineWidth : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(DimensionTextLineWidth) : Number {This specifies a distance with a unit of 1 mil.}

AsMM(DimensionTextLineWidth) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**

DimensionTextLineWidth = 6

Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Text Line Width property which is 6 current measurement units.

DimensionTextLineWidth >= 0.127

Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Text Line Width property which is greater than or equal to 0.127 current measurement units.

AsMils(DimensionTextLineWidth) Between 4 And 8

Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Text Line Width property which is greater than or equal to 4 mil and less than or equal to 8 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(DimensionTextLineWidth) < 0.1778

Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Text Line Width property which is less than 0.1778 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

**DimensionTextPosition Field**

**Description**

Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects having a Dimension Text Position property that complies with the Query.

Note: The child objects of these Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects are not also returned.

Note: The DimensionTextPosition property is only defined for Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects.

**Syntax**

DimensionTextPosition = DimensionTextPosition_String
DimensionTextPosition <> DimensionTextPosition_String

DimensionTextPosition_String must be one of the strings from the following list:

Note: The single quote characters (') shown at the start and end of each DimensionTextPosition_String are both mandatory.

Examples

DimensionTextPosition = 'Automatic'
Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have an 'Automatic' Dimension Text Position property.

DimensionTextPosition <> 'Manual'
Returns all objects except Angular Dimension, Baseline Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a 'Manual' Dimension Text Position property. (Only Component objects have a Dimension Text Position property, so all remaining types of objects do not have a 'Manual' DimensionTextPosition property, and are thus also returned by this Query.)

DimensionTextPrecision Field

Description
Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects having a Dimension Precision property that complies with the Query.

Note: The child objects of these Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects are not also returned.

Note: The DimensionTextPrecision property is only defined for Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects.

Syntax
DimensionTextPrecision : Numeric_String
DimensionTextPrecision : Number {If all objects which do not have a Dimension Precision property are totally excluded by one or more preceding tokens within the Query.}

This number is an integral value between 0 and 5 (inclusive) for Angular Dimension objects, and between 0 and 6 (inclusive) for Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects.

The DimensionTextPrecision keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all objects which do not have a Dimension Precision property have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:
DimensionTextPrecision <> ''
DimensionTextPrecision > ''
IsDimension

The usage of such tokens is demonstrated in the examples that follow.

Examples

DimensionTextPrecision = '2'
DimensionTextPrecision <> '' && DimensionTextPrecision = 2
Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Precision property of 2.

DimensionTextPrecision > '' && DimensionTextPrecision Between 3 And 5
Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Precision property which is greater than or equal to 3 and less than or equal to 5.
IsDimension && DimensionTextPrecision < 4
Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Precision property which is less than 4.

```
DimensionTextPrecision <> '1'
```

Returns all objects except Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Precision property which is equal to 1. (Only Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects have a DimensionTextPrecision property, so all remaining types of objects do not have a DimensionTextPrecision property of 1, and are thus also returned by this Query.)

```
IsDimension && DimensionTextPrecision <> 1
```

Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Precision property that is **not** equal to 1.

### DimensionTextPrefix Field

**Description**

Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects having a Dimension Prefix property that complies with the Query.

Note: The child objects of these Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects are **not** also returned.

Note: The DimensionTextPrefix property is only defined for Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects.

**Syntax**

```
DimensionTextPrefix : String
```

**Example**

```
DimensionTextPrefix = 'ø'
```

Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Prefix property of 'ø'.

### DimensionTextSuffix Field

**Description**

Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects having a Dimension Suffix property that complies with the Query.

Note: The child objects of these Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects are **not** also returned.

Note: The DimensionTextSuffix property is only defined for Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects.

**Syntax**

```
DimensionTextSuffix : String
```

**Example**

```
DimensionTextSuffix = '(+/- 0.5%)'
```

Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects that have a Dimension Suffix property of '(+/- 0.5%)'.

### DimensionTextUnit Field

**Description**

Returns all Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects having a Dimension Unit property that complies with the Query.

Note: The child objects of these Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects are **not** also returned.
Note: The DimensionTextUnit property is only defined for Angular Dimension, Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects.

At present, all distances always use units of mils. (1 mil = 0.001 inch)

**Syntax**

```plaintext
DimensionTextUnit = DimensionUnit_String
DimensionTextUnit <> DimensionUnit_String
```

**For Angular Dimension objects**, DimensionUnit_String must be one of the strings from the following list:

'Degrees', 'Radians'

**For Baseline Dimension, Datum Dimension, Linear Diameter Dimension, Linear Dimension, Radial Diameter Dimension, and Radial Dimension objects**, DimensionUnit_String must be one of the strings from the following list:

'Automatic', 'Centimeters', 'Inches', 'Millimeters', 'Mils'

Note: The single quote characters (') shown at the start and end of each DimensionUnit_String are both mandatory.

**Examples**

```plaintext
DimensionTextUnit = 'Radians'
Returns all Angular Dimension objects that have a 'Radians' Dimension Unit property.
DimensionTextUnit <> 'Degrees'
Returns all objects except Angular Dimension objects that have a 'Degrees' Dimension Unit property.
```

**DrillType Field**

**Description**

Not Implemented

**EnablePartSwapping Field**

**Description**

Returns all PCB components objects having an EnablePartSwapping property that complies with the Query.

**Syntax**

```plaintext
EnablePartSwapping : Boolean
```

**Examples**

```plaintext
EnablePartSwapping
EnablePartSwapping = True
Returns all PCB components that have the Part Swapping setting set to true.
EnablePartSwapping = False
Returns all PCB components that have the Part Swapping setting set to false.
```

**EnablePinSwapping Field**

**Description**

Returns all PCB components objects having an EnablePinSwapping property that complies with the Query.

**Syntax**

```plaintext
EnablePinSwapping : Boolean
```
Examples
EnablePinSwapping
EnablePinSwapping = True

Returns all PCB components that have the Pin Swapping setting set to true.
EnablePartSwapping = False

Returns all PCB components that have the Pin Swapping setting set to false.

Er Impedance

Description
Specifies the relative dielectric constant of the core (non-copper) material which a PCB is composed of. That value has an influence upon the characteristic impedance between a track on a Signal layer and the copper on an (adjacent) Internal Plane layer (or the copper on (adjacent) surrounding Internal Plane layers), or the width of such a track when a particular characteristic impedance is required.

Note: This is an alias for DielectricConstant (Impedance).

Syntax
Er : Number

The relative dielectric constant is the ratio of the material's dielectric constant to the dielectric constant of empty space.

Examples
(87/SQRT(Er+1.41))*LN(5.98*TraceToPlaneDistance/(0.8*TraceWidth + TraceHeight))

The default Query for specifying the characteristic impedance between a track on the Signal layer and the copper on the (adjacent) Internal Plane layer, when the Microstrip option is selected.

(60/SQRT(Er))*LN((1.9*PlaneToPlaneDistance)/(0.8*TraceWidth + TraceHeight))

The default Query for specifying the characteristic impedance between a track on the intermediate Signal layer and the copper on the two surrounding Internal Plane layers, when the Stripline option is selected.

ExistsOnLayer Membership Check

Description
Returns objects that exist on the specific layer (including those pads that have a multiple layer property).

Syntax
ExistsOnLayer(LayerString) : Boolean

where LayerString is one of the following layer strings;

Note, the LayerString parameter does not have to be enclosed by inverted commas.

Examples
ExistsOnLayer(TopLayer)
ExistsOnLayer(TopBottom) = True
Returns objects that exist on the top layer only.

Not ExistsOnLayer(BottomLayer)

ExistsOnLayer(BottomLayer) = False

Returns all objects on other layers other than the bottom layer.

Font Field

Description

Returns all Coordinate, Original Dimension, and Text objects having a Font property that complies with the Query.

Note: The child objects of these Coordinate and Original Dimension objects are not also returned.

Note: The Font property is only defined for Coordinate, Original Dimension, and Text objects.

Syntax

Font = Font_String
Font <> Font_String

Font_String must be one of the strings from the following list:
'Default', 'Serif', 'Sans Serif'

Note: The single quote characters ('') shown at the start and end of each Font_String are both mandatory.

Examples

Font = 'Default'

Returns all Coordinate, Original Dimension, and Text objects that have a 'Default' Font property.

Font <> 'Serif'

Returns all objects except Coordinate, Original Dimension, and Text objects that have a 'Serif' Font property. (Only Coordinate, Original Dimension, and Text objects have a Font property, so all remaining types of objects do not have a 'Serif' Font property, and are thus also returned by this Query.)

FontName Field

Description

Returns all Coordinate, Dimension, and Text objects having a FontName property that complies with the Query.

Note: The child objects of these Coordinate and Dimension objects are not also returned.

Syntax

FontName = FontName_String
FontName <> FontName_String

FontName_String must be one of the strings from the following list:
'Default', 'Serif', 'Sans Serif' as well as one of the true type fonts used by the PCB document on a particular PC.

Note: The single quote characters ('') shown at the start and end of each FontName_String are both mandatory.

Examples

Font = 'Default'

Returns all Coordinate, Dimension, and Text objects that have a 'Default' Font property.

Font <> 'Serif'

Returns all objects except Coordinate, Original Dimension, and Text objects that have a 'Serif' Font property. (Only Coordinate, Original Dimension, and Text objects have a Font property, so all remaining types of objects do not have a 'Serif' Font property, and are thus also returned by this Query.)

Footprint Field

Description

Returns all Component objects having a Footprint property that complies with the Query.
Note: The child objects of these Component objects are not also returned.

Note: The Footprint property is only defined for Component objects.

Syntax
Footprint : String

Examples
Footprint = 'DIP14'
Returns all components that have a Footprint property of 'DIP14'.
Footprint Like 'DIP*'
Returns all components that have a Footprint property whose associated string commences with 'DIP'.

FromToPad1 Field

Description
Returns all From To objects having a From Pad property that complies with the Query.

Note: The FromToPad1 property is only defined for From To objects.

Syntax
FromToPad1 : PadName_String

The PadName_String specifies the full Name that a pad has to match; this consists of an initial substring which specifies the Name property of the pad's parent component, followed by a '-' character, followed by another substring which specifies the Name property of the pad itself. Pads which are of a "free" nature are identified by an initial substring of 'Free-' (followed by the substring that specifies the pad's own Name property).

Example
FromToPad1 = 'ZZ1-2'
Returns all From To objects that have a From Pad property of 'ZZ1-2'.

FromToPad2 Field

Description
Returns all From To objects having a To Pad property that complies with the Query.

Note: The FromToPad2 property is only defined for From To objects.

Syntax
FromToPad2 : PadName_String

The PadName_String specifies the full Name that a pad has to match; this consists of an initial substring which specifies the Name property of the pad's parent component, followed by a '-' character, followed by another substring which specifies the Name property of the pad itself. Pads which are of a "free" nature are identified by an initial substring of 'Free-' (followed by the substring that specifies the pad's own Name property).

Example
FromToPad2 = 'ZZ2-1'
Returns all From To objects that have a To Pad property of 'ZZ2-1'.

H Field

Description
Returns components whose Height property complies with the Query.

Note: The child objects of these Component objects are not also returned.

Note: The H property is only defined for Component objects.

Note: This is an alias for Height (Field).

Syntax
H : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(H) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(H) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**

\[
\begin{align*}
H &= 300 \\
\text{Returns all components that have a Height property which is equal to 300 current measurement units.} \\
H &\geq 5.08 \\
\text{Returns all components that have a Height property which is greater than or equal to 5.08 current measurement units.} \\
\text{AsMils}(H) &> 400 \\
\text{Returns all components that have a Height property which is greater than 400 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)} \\
\text{AsMM}(H) &\text{ Between 6.35 And 8.89} \\
\text{Returns all components that have a Height property which is greater than or equal to 6.35 mm and less than or equal to 8.89 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)} \\
H &\not= 550 \\
\text{Returns all components that have a Height property which is } \not= \text{ equal to 550 current measurement units.} \\
\text{AsMils}(H) &\leq 650 \\
\text{Returns all components that have a Height property which is less than or equal to 650 mil.} \\
\text{AsMM}(H) &< 11.43 \\
\text{Returns all components that have a Height property which is less than 11.43 mm.}
\end{align*}
\]

**HasFootprint Membership Check**

**Description**

Returns all objects associated with any Component object having a Footprint property that complies with the Query. For each such Component object, both the parent object and all of its child objects are returned.

**Syntax**

HasFootprint(Footprint : String) : Boolean

The String specifies the Footprint property that Component objects need to match.

The '?' and '*' characters are "wild card" in nature; '?' replaces any single character, while '*' replaces any number of any characters.

**Examples**

\[
\begin{align*}
\text{Not HasFootprint('IDC16') } &\quad \text{Returns all objects \textbf{except} those associated with Component objects that have a Footprint property of 'IDC16.'} \\
\text{HasFootprint('SIP?0')} &\quad \text{Returns all objects associated with Component objects that have a Footprint property whose associated string starts with 'SIP' and finishes with '0', and includes one other intermediate character; compliant strings include 'SIP10', 'SIP20', 'SIPQ0', etc.} \\
\text{HasFootprint('DIP**')} &\quad \text{Returns all objects associated with Component objects that have a Footprint property whose associated string starts with 'DIP'; compliant strings include 'DIP8', 'DIP14', 'DIP24N', etc.} \\
\end{align*}
\]
HasFootprint('*', '*2*') = True

Returns all objects associated with Component objects that have a Footprint property whose associated string includes a character of '2'; compliant strings include '2012', '2N5001', 'IDC20', 'HDR1X2', etc.

**HasFootprintPad Membership Check**

**Description**

Returns Pad objects having a Name property that complies with the Query, when these are child objects of any Component object having a Footprint property that also complies with the Query.

**Syntax**

HasFootprintPad(Footprint : String, Name : String) : Boolean

The first String specifies the Footprint property that each pad's parent component needs to match, while the second String specifies the Name property that each pad itself needs to match.

The '?' and '*' characters are "wild card" in nature; '?' replaces any single character, while '*' replaces any number of any characters.

**Examples**

Not HasFootprintPad('0805', '1')

HasFootprintPad('0805', '1') = False

Returns all objects except Pad objects that have a Name property of '1' and which are child objects of Component objects that have a Footprint property of '0805'.

HasFootprintPad('SIP?0', '1')

HasFootprintPad('SIP?0', '1') = True

Returns all Pad objects that have a Name property of '1', and which are child objects of Component objects that have a Footprint property whose associated string starts with 'SIP' and finishes with '0', and includes one other intermediate character; compliant strings include 'SIP10', 'SIP20', 'SIPQ0', etc.

HasFootprintPad('DB*', 'MNT')

HasFootprintPad('DB*', 'MNT') = True

Returns all Pad objects that have a Name property of 'MNT', and which are child objects of Component objects that have a Footprint property whose associated string starts with 'DB'; compliant strings include 'DB9S', 'DB15HD', 'DB25P', etc.

HasFootprintPad('*', '*')

HasFootprintPad('*', '*') = True

Returns all Pad objects that have a Name property whose associated string starts with 'MN', and which are child objects of any Component object. (Because a Footprint property of '*' is specified, all Footprint strings comply with the Query, and thus all Component objects comply with the Footprint part of the Query.)

**HasPad Membership Check**

**Description**

Returns Pad objects whose full Name property complies with the Query.

**Syntax**

HasPad(PadName_String) : Boolean

The PadName_String specifies the full Name property that each pad has to match; it consists of an initial substring which specifies the Name property of the pad's parent component, followed by a '-' character, followed by another substring which specifies the Name property of the pad itself. Pads which are of a "free" nature are specified by using an initial substring of 'Free-' (followed by a substring that specifies the pad's own Name property).

The '?' and '*' characters are "wild card" in nature; '?' replaces any single character, while '*' replaces any number of any characters.

**Examples**

Not HasPad('C4-1')

HasPad('C4-1') = False
Returns all objects except Pad objects that have a Name property of '1' and which are child objects of Component objects that have a Name property of 'C4'.

\[
\text{HasPad('J10-MN?')} \\
\text{HasPad('J10-MN?')} = \text{True}
\]

Returns all Pad objects which are child objects of Component objects having a Name property of 'J10', and which have a Name property whose associated string starts with 'MN' and consists of one more character; compliant (pad) Name properties include 'MN1', 'MN2', 'MNT', etc.

\[
\text{HasPad('Free-*')} \\
\text{HasPad('Free-*')} = \text{True}
\]

Returns all Pad objects which are of a "free" nature, i.e. which are not child objects of any Component objects.

\[
\text{HasPad('*-1')} \\
\text{HasPad('*-1')} = \text{True}
\]

Returns all Pad objects having a Name property of '1', and regardless of whether these are of a "free" nature or child objects of any Component object.

### HasParameter field

**Description**

Returns all Library Component or Footprints having a parameter property with two Name and Value parameters that complies with the Query.

**Syntax**

\[
\text{HasParameter(}\text{Parameter Name, Parameter Value}\text{)} : \text{Boolean}
\]

**Examples**

\[
\text{HasParameter('Text Field1','CAPACITOR (ELECTROLYTIC)')} \\
\text{HasParameter('Text Field1','CAPACITOR (ELECTROLYTIC)')} = \text{True}
\]

Returns objects that have a Parameter Name of Text Field 1 and a Parameter Value of CAPACITOR (ELECTROLYTIC).

\[
\text{Not HasParameter('Text','Value')} \\
\text{HasParameters('','')} = \text{False}
\]

Returns all objects except objects that have 'Text' Parameter Name and 'Value' Parameter Value.

### HasRoundHole Object Type Check

**Description**

Returns pad objects that have round holes.

**Syntax**

\[
\text{HasRoundHole} : \text{Boolean}
\]

**Examples**

\[
\text{HasRoundHole} \\
\text{HasRoundHole} = \text{True}
\]

Returns all pad objects that have round holes.

\[
\text{Not HasRoundHole} \\
\text{HasRoundHole} = \text{False}
\]

Returns all objects except those pad objects that have round holes.
HasSlotHole Object Type Check

Description
Returns pad objects that have slotted holes.

Syntax
HasSlotHole : Boolean

Examples
HasSlotHole
HasSlotHole = True
Returns all pad objects that have slotted holes.
Not HasSlotHole
HasSlotHole = False
Returns all objects except those pad objects that have slotted holes.

HasSquareHole Object Type Check

Description
Returns pad objects that have square holes.

Syntax
HasSquareHole : Boolean

Examples
HasSquareHole
HasSquareHole = True
Returns all pad objects that have squared holes.
Not HasSquareHole
HasSquareHole = False
Returns all objects except those pad objects that have squared holes.

HasViolations Attribute Check

Description
Returns objects whose Violation state is True; that occurs when a violation of one or more Design Rules has been detected that involves the object concerned.

Syntax
HasViolations : Boolean

Examples
HasViolations
HasViolations = True
Returns objects that have a True Violation state.
Not HasViolations
HasViolations = False
Returns all objects except objects that have a True Violation state.
**Height Field**

**Description**
Returns components whose Height property complies with the Query.

Note: The child objects of these Component objects are **not** also returned.

Note: The Height property is only defined for Component objects.

Note: This is an alias for H (Field).

**Syntax**

```
Height : Number
```

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

```
AsMils(Height) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(Height) : Number {This specifies a distance with a unit of 1 mm.}
```

**Examples**

```
Height = 300
Returns all components that have a Height property which is equal to 300 current measurement units.

Height >= 5.08
Returns all components that have a Height property which is greater than or equal to 5.08 current measurement units.

AsMils(Height) > 400
Returns all components that have a Height property which is greater than 400 mil. (The current value of the target file’s Measurement Unit property has no impact upon this outcome.)

AsMM(Height) Between 6.35 And 8.89
Returns all components that have a Height property which is greater than or equal to 6.35 mm and less than or equal to 8.89 mm. (The current value of the target file’s Measurement Unit property has no impact upon this outcome.)

Height <> 550
Returns all components that have a Height property which is **not** equal to 550 current measurement units.

AsMils(Height) <= 650
Returns all components that have a Height property which is less than or equal to 650 mil.

AsMM(Height) < 11.43
Returns all components that have a Height property which is less than 11.43 mm.
```

**Hide Field**

**Description**
Returns all Designator (Text), Comment (Text), and Net objects whose Hide property complies with the Query.

Note: The Hide property is only defined for Designator and Comment objects (which are child objects of Component objects) and Net objects.

**Syntax**

```
Hide : Boolean
```

**Examples**

```
Hide
Hide = True
Returns all Designator, Comment, and Net objects that have a True Hide property.

Not Hide
```
Hide = False
Returns all objects except Designator, Comment, and Net objects that have a True Hide property.

**Hole Field**

**Description**
Returns all Pad and Via objects having a Hole Size property that complies with the Query.

Note: The Hole property is only defined for Pad and Via objects.

Note: This is an alias for HoleSize (Field).

**Syntax**
Hole : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(Hole) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(Hole) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**
Hole = 16
Returns all Pad and Via objects that have a Hole Size property which is 16 current measurement units.

Hole >= 0.508
Returns all Pad and Via objects that have a Hole Size property which is greater than or equal to 0.508 current measurement units.

AsMils(Hole) Between 12 And 32
Returns all Pad and Via objects that have a Hole Size property which is greater than or equal to 12 mil and less than or equal to 32 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(Hole) < 1.016
Returns all Pad and Via objects that have a Hole Size property which is less than 1.016 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

**HoleDiameter Field**

**Description**
Returns all Pad and Via objects having a Hole Size property that complies with the Query.

Note: The HoleDiameter property is only defined for Pad and Via objects.

Note: This is an alias for HoleSize (Field).

**Syntax**
HoleDiameter : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(HoleDiameter) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(HoleDiameter) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**
HoleDiameter = 16
Returns all Pad and Via objects that have a Hole Size property which is 16 current measurement units.
HoleDiameter >= 0.508

Returns all Pad and Via objects that have a Hole Size property which is greater than or equal to 0.508 current measurement units.
AsMils(HoleDiameter) Between 12 And 32

Returns all Pad and Via objects that have a Hole Size property which is greater than or equal to 12 mil and less than or equal to 32 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(HoleDiameter) < 1.016

Returns all Pad and Via objects that have a Hole Size property which is less than 1.016 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

**HoleRotation Field**

**Description**

Returns all pad objects having a HoleRotation property that complies with the Query. All pad objects that are circular have 0 degrees. Those pad objects that have squared or slotted holes can have different hole rotation values.

**Syntax**

HoleRotation = RealDegreesValue
HoleRotation <> RealDegreesValue

RealDegreesValue is a real value in degrees.

**Examples**

HoleRotation = 0

Returns all pad objects with a hole rotation property of 0 degrees.

HoleRotation <> 45

Returns all pad objects except for those pad objects that have a hole rotation value of 45 degrees.

**HoleSize Field**

**Description**

Returns all Pad and Via objects having a Hole Size property that complies with the Query.

Note: The HoleSize property is only defined for Pad and Via objects.

**Syntax**

HoleSize : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(HoleSize) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(HoleSize) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**

HoleSize = 16

Returns all Pad and Via objects that have a Hole Size property which is 16 current measurement units.

HoleSize >= 0.508

Returns all Pad and Via objects that have a Hole Size property which is greater than or equal to 0.508 current measurement units.

AsMils(HoleSize) Between 12 And 32
Returns all Pad and Via objects that have a Hole Size property which is greater than or equal to 12 mil and less than or equal to 32 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

HoleType Field

Description
Returns all pad objects having a HoleType property that complies with the Query.

Syntax
HoleType = 'HoleTypeValue'

Where there are three different hole types; 'Round', 'Slot', 'Square' for the HoleTypeValue variable. When you specify the holetype string, make sure you have the inverted commas as well.

Examples
HoleType = 'Square'

Returns all pad objects with a square hole type.

HoleType <> 'Slot'

Returns all pads except those Pad objects that have the Slotted hole property.

HoleType = 'Square' || HoleType = 'Slot'
IsPad && HoleType <> 'Round'

Returns all Pad objects that do not have a round hole property.

Id Attribute Check

Description
Returns the object or objects that are specified by the Query.

Syntax
Id : ObjectIdentifying_String

ObjectIdentifying_String must be one of the strings that is listed within the PCB Object Lists Category of the Query Helper dialog, and it needs to be preceded and followed by single quote characters (''). What strings are listed there depends upon what objects currently exist within the file currently being edited, with corresponding strings being provided for all currently existing Net, Pad, Text, Component, Dimension, Coordinate, Component Class, Net Class, Layer Class, From-To Class, Pad Class, Channel Class, (Design) Rule, From-To, and Violation objects.

Although a string is listed for each of those objects, these strings do not necessarily identify one particular object. As one example, the string provided for each (Design) Rule object consists of the characters 'Rule[0mil,0mil]', so specifying that string will result in all of the Rule objects being returned.

Examples
Id = 'GND'

Returns the Net object whose Net property is 'GND'.

Id = 'C3-1'

Returns the Pad object(s) whose full Name property is 'C3-1'.

Id = 'Text[3510mil,3800mil]' 

Returns the Text object(s) whose point is at 3510mil,3800mil.

Id = 'D3'

Returns the Component object(s) whose Designator property is 'D3'.

Id = 'Dimension[4800mil,3800mil]'

Returns the Dimension object(s) whose insertion point is at 4800mil,3800mil.

Id = 'Coordinate[3500mil,3800mil]'
Returns the Coordinate object(s) whose insertion point is at 3500mil,3800mil.
Id = 'My Top Side Layers'

Returns the (Layer) Class object whose Name property is 'My Top Side Layers'.
Id = 'Rule[0mil,0mil]'

Returns all of the (Design) Rule objects (as these all have the same identifying string assigned to them).
Id = 'GND (ZZ4-2 : ZZ3-2)'

Returns the From-To object whose Net property is 'GND', From Pad property is 'ZZ4-2', and To Pad property is 'ZZ3-2'.
Id = 'Violation[0mil,0mil]'

Returns all of the Violation objects (as these all have the same identifying string assigned to them).

InBGA Attribute Check

Description
Returns child objects of components that are of a BGA nature. Such a component contains at least 10 rows of pads and at least 10 columns of pads, the (absolute) difference between the number of rows of pads and the number of columns of pads is less than 6, the total pad count is at least half of the product of the number of rows of pads and the number of columns of pads, and all of the pads are of a surface mount nature.
Note: The parent objects of these component objects are not also returned.

Syntax
InBGA : Boolean

Examples
InBGA
InBGA = True

Returns all child objects of all Component objects that are of a BGA nature (as defined above).
Not InBGA
InBGA = False

Returns all objects except child objects of Component objects that are of a BGA nature (as defined above).

See Also
InDIP
InEdgeConnector
InLCC
InPGA
InSIP
InSMSIP
InSMTComponent
InSOIC
InThruComponent

InAnyComponent Attribute Check

Description
Returns all the highlighted components only as specified by the InAnyComponent Query.

Syntax
InAnyComponent : Boolean

Examples
InAnyComponent
InAnyComponent = True
Returns all components only.
Not InAnyComponent

Returns all the objects that are not components.

InAnyDifferentialPair Attribute Check

Description
Returns all objects that possess a Differential Pair property as specified by the InAnyDifferentialPair Query.

Syntax
InAnyDifferentialPair : Boolean

Examples
InAnyDifferentialPair
InAnyDifferentialPair = True

Returns any differential pairs on the current PCB document.
Not InAnyDifferentialPair
Returns any objects that do not possess the differential pair property.

InAnyNet Attribute Check

Description
Returns all the highlighted objects that has a Net property as specified by the InAnyNet Query.

Syntax
InAnyNet : Boolean

Examples
InAnyNet
InAnyNet = True

Returns all Objects that have a Net property.
Not InAnyNet
Returns all the objects that don't have a Net property.

InChannelClass Membership Check

Description
Returns all objects associated with any Component object that is a member of any Channel Class having a Name property that complies with the Query. For each such Component object, both the parent object and all of its child objects are returned.

Syntax
InChannelClass(Name : String) : Boolean
InChannelClass(Name : String, Name : String) : Boolean
InChannelClass(Name : String, Name : String, Name : String) : Boolean
...

Each String specifies the Name property of a Channel Class.
The '?' and '*' characters are "wild card" in nature; '?' replaces any single character, while '*' replaces any number of any characters.

Examples
InChannelClass('Input channel')
InChannelClass('Input channel') = True

Returns all Component objects that are members of the Channel Class that has a Name property of 'Input channel', along with all of the child objects of those Component objects.
InChannelClass('Output channel', 'Input channel')
InChannelClass('Output channel', 'Input channel') = True

Returns all Component objects that are members of either of the Channel Classes that have Name properties of 'Output channel' and 'Input channel', along with all of the child objects of those Component objects.

InChannelClass('Input*')
InChannelClass('Input*') = True

Returns all Component objects that are members of any Channel Class that have a Name property whose associated string starts with 'Input', along with all of the child objects of those Component objects.

InComponent Membership Check (PCB)

Description
Returns each component having a Name property matching that specified by the Query; all of the child objects of each such Component object are also returned.

Syntax
InComponent(Name : String) : Boolean
InComponent(Name : String, Name : String) : Boolean
InComponent(Name : String, Name : String, Name : String) : Boolean

The '?' and '*' characters are "wild card" in nature; '?' replaces any single character, while '*' replaces any number of any characters.

Examples
InComponent('U1')
InComponent('U1') = True

Returns any component that has a Name property of 'U1', along with all of the child objects of each such component.

InComponent('U1','U2','J1')
InComponent('U1','U2','J1') = True

Returns any component that has a Name property of 'U1' or 'U2' or 'J1', along with all of the child objects of each such component.

InComponent('U*')
InComponent('U*') = True

Returns any component that has a Name property whose associated string starts with 'U', along with all of the child objects of each such component.

InComponentClass Membership Check

Description
Returns all Component objects that are members of the specified Component Class.

The user can define Component Classes as desired, and each component within the file can either be a member, or not, of each Component Class which is so created.

The 'All Components' Component Class exists in every PCB file by default; it includes all components in the file. It is not possible to change which components are members of that Component Class, but the user has full control over which components are members of any other Component Classes (which are created and named by the user).

Syntax
InComponentClass(Name : String) : Boolean

The String specifies the Name property of the Component Class.

Examples
InComponentClass('All Components')
InComponentClass('All Components') = True

Returns all Component objects. (All components are members of the (default) Component Class of 'All Components'.)
InComponentClass('Bottom Side SMD')
InComponentClass('Bottom Side SMD') = True

Returns all Component objects that are members of the user-defined Component Class with a Name property of 'Bottom Side SMD'. (If that Component Class has not been created, or if no components within the file are members of it, then no objects are returned.)

InDifferentialPair Membership Check

Description
Returns each Differential Pair having a Designator Name property matching that specified by the Query.

Syntax
InDifferentialPair(Name : String) : Boolean
InDifferentialPair(Name : String , Name : String) : Boolean
InDifferentialPair(Name : String , Name : String , Name : String) : Boolean

The '?' and '*' characters are "wild card" in nature; '?' replaces any single character, while '*' replaces any number of any characters.

Examples
InDifferentialPair('NewDifferentialPair1')
InDifferentialPair('NewDifferentialPair1') = True

Returns any differential pairs that has a Name property of 'NewDifferentialPair1'.
InDifferentialPair('NewDifferentialPair1','NewDifferentialPair2','NewDifferentialPair3')
Returns any differential pairs that has a Name property of 'NewDifferentialPair1' or 'NewDifferentialPair2' or 'NewDifferentialPair3'.

InDifferentialPairClass Membership Check

Description
Returns all Differential Pairs objects that are members of the specified Differential Pair Class.

The user can define Differential Pair Classes, and each component within the file can either be a member, or not, of each created Component Class.

The 'All Differential' Differential Pair Class exists in every PCB document by default and includes all differential pairs. It is not possible to change which differential pairs are members of the All Differential Pair Class, but the user has full control over which differential pairs are members of other Component Classes (which are created and named by the user).

Syntax
The String specifies the Name property of the Differential Pair Class.

Examples
InDifferentialPairClass('All Differential Pairs')
InDifferentialPairClass('All Differential Pairs') = True

Returns all differential pair objects. (All differential pairs are members of the (default) differential pair Class of 'All Differential Pairs'.)
InDifferentialPairClass('NewDifferentialPair1')
InDifferentialPairClass('NewDifferentialPair1') = True

Returns all differential pair objects that are members of the user-defined Differential Pair Class with a Name property of 'Bottom Side SMD'. (If that Differential Pair Class has not been created, or if no differential pairs within the file are members of it, then no objects are returned.)

InDIP Attribute Check

Description
Returns child objects of components that are of a DIP nature. At least one (and typically all) of the pads of such a component are of a through-hole nature, and the total pad count is equal to the product of the number of rows of pads and the number of columns of pads. Such a component also has either exactly two rows of pads and at least three columns of pads, or else has exactly two columns of pads and at least three rows of pads.

Note: The parent objects of these component objects are not also returned.

**Syntax**

`InDIP : Boolean`

**Examples**

`InDIP = True`

Returns all child objects of all Component objects that are of a DIP nature (as defined above).

`Not InDIP`

`InDIP = False`

Returns all objects except child objects of Component objects that are of a DIP nature (as defined above).

**See Also**

`InBGA`

`InEdgeConnector`

`InLCC`

`InPGA`

`InSIP`

`InSMD`

`InSMTComponent`

`InSOIC`

`InThruComponent`

### InEdgeConnector Attribute Check

**Description**

Returns child objects of components that are of an Edge Connector nature. All of the pads of such a component are of a surface mount nature, and these are located on both sides of the PCB. All of the pads reside in either the same row or the same column, the number of pads on either side of the PCB is less than three times the number of pads on the opposite side of the PCB, and the total number of pads is at least 4.

Note: The parent objects of these component objects are not also returned.

**Syntax**

`InEdgeConnector : Boolean`

**Examples**

`InEdgeConnector = True`

Returns all child objects of all Component objects that are of an Edge Connector nature (as defined above).

`Not InEdgeConnector`

`InEdgeConnector = False`

Returns all objects except child objects of Component objects that are of an Edge Connector nature (as defined above).

**See Also**

`InBGA`

`InDIP`

`InLCC`

`InPGA`

`InSIP`
InFromTo Membership Check

Description
Returns a From-To object whose Net, From Pad, and To Pad properties all match those specified by the Query.

Syntax
InFromTo('Net_Name (From_Pad : To_Pad)') : Boolean

Net_Name, From_Pad, and To_Pad are all Strings; the only single quote characters (’) used are those depicted above (immediately within the outermost pair of brackets).

Each of the From_Pad and To_Pad strings specifies the full Name that the corresponding pad has to match; each of these strings consists of an initial substring which identifies the Name property of the pad’s parent Component object, followed by a ‘-’ character, followed by another substring which identifies the Name property of the pad itself.

Example
InFromTo('GND (J4-2 : U3-4)')
InFromTo('GND (J4-2 : U3-4)') = True

Returns a From-To object having a Net property of ‘GND’, a From Pad property of ’J4-2’, and a To Pad property of ’U3-4’. (As such, the From Pad has a Name property of ’2’, and its parent Component object has a Name property of ’J4’; similarly, the To Pad has a Name property of ’4’, and its parent Component object has a Name property of ’U3’.)

InFromToClass Membership Check

Description
Returns all From-To objects that are members of the specified From-To Class.

The user can define From-To Classes as desired, and each From-To object within the file can either be a member, or not, of each From-To Class which is so created.

The ’All From-Tos’ From-To Class exists in every PCB file by default; it includes all From-To objects in the file. It is not possible to change which From-To objects are members of that From-To Class, but the user has full control over which From-To objects are members of any other From-To Classes (which are created and named by the user).

Syntax
InFromToClass(Name : String) : Boolean

The String specifies the Name property of the From-To Class.

Examples
InFromToClass('All From-Tos')
InFromToClass('All From-Tos') = True

Returns all From-To objects. (All From-To objects are members of the (default) From-To Class of ’All From-Tos’.)

InFromToClass('Bus Terminators')
InFromToClass('Bus Terminators') = True

Returns all From-To objects that are members of the user-defined From-To Class with a Name property of ’Bus Terminators’. (If that From-To Class has not been created, or if no From-To objects within the file are members of it, then no objects are returned.)

InLayerClass Membership Check

Description
Returns all objects having a Layer property that matches a layer that is a member of the specified Layer Class.
The user can define Layer Classes as desired, and each enabled layer can either be a member, or not, of each Layer Class which is so created.

The following four Layer Classes exist in every PCB file by default:

'All Layers': This includes both outside Signal layers, all internal Signal layers which have been enabled by the user, all Internal Plane layers which have been enabled by the user, all Mechanical layers which have been enabled by the user, both Overlay layers, both Paste Mask layers, both Solder Mask layers, the Drill Guide layer, the Keep-Out layer, the Drill Drawing layer, and the Multi-Layer layer.

'Component Layers': This includes both outside Signal layers.

'Electrical Layers': This includes both outside Signal layers, all internal Signal layers which have been enabled by the user, and all Internal Plane layers which have been enabled by the user.

'Signal Layers': This includes both outside Signal layers, and all internal Signal layers which have been enabled by the user.

It is not possible to change which layers are members of each of those four Layer Classes, but the user has full control over which layers are members of any other Layer Classes (which are created and named by the user).

**Syntax**

\[
\text{InLayerClass(Name : String) : Boolean}
\]

The String specifies the Name property of the Layer Class

**Examples**

\[
\text{InLayerClass('Component Layers')}
\]

\[
\text{InLayerClass('Component Layers')} = \text{True}
\]

Returns all objects that have a Layer property of 'Top Layer' or 'Bottom Layer'. (Those layers are members of the (default) Layer Class of 'Component Layers'.)

\[
\text{InLayerClass('My Top Side')}
\]

\[
\text{InLayerClass('My Top Side')} = \text{True}
\]

Returns all objects that have a Layer property that matches a layer that is a member of the user-defined Layer Class with a Name property of 'My Top Side'. (If that Layer Class has not been created, or if none of the currently enabled layers within the file are members of it, then no objects are returned.)

As an example of which layers could be members of that Layer Class, the user may have paired the Mechanical 3 and Mechanical 4 layers, with the former (Mechanical) layer being used with top side components, and the latter (Mechanical) layer being used with bottom side components; hence the 'My Top Side' Layer Class could consist of the Top (Signal), Top Overlay, Top Paste Mask, Top Solder Mask, and Mechanical 3 layers (and hence there could well be another Layer Class with a Name property of 'My Bottom Side', and consisting of the Bottom (Signal), Bottom Overlay, Bottom Paste Mask, Bottom Solder Mask, and Mechanical 4 layers).

**InLCC Attribute Check**

**Description**

Returns child objects of components that are of a Surface Mount LCC nature. Such a component contains 2 rows of pads which each contain at least 4 pads, and 2 columns of pads which each contain at least 4 pads, and all of the pads are of a surface mount nature.

Note: The parent objects of these component objects are not returned.

**Syntax**

\[
\text{InLCC : Boolean}
\]

**Examples**

\[
\text{InLCC}
\]

\[
\text{InLCC} = \text{True}
\]

Returns all child objects of all Component objects that are of a Surface Mount LCC nature.

\[
\text{Not InLCC}
\]

\[
\text{InLCC} = \text{False}
\]

Returns all objects except child objects of Component objects that are of a Surface Mount LCC nature.

**See Also**
InBGA
InDIP
InEdgeConnector
InPGA
InSIP
InSMSIP
InSMTComponent
InSOIC
InThruComponent

**InNamedPolygon Membership Check**

**Description**
Returns all objects that are part of the named polygon as specified by the Query.

**Syntax**

\[
\text{InNamedPolygon('PolygonName')} : \text{Boolean}
\]

The PolygonName string is the Name property of an actual polygon object.

**Example**

\[
\text{InNamedPolygon('Custom')} \\
\text{InNamedPolygon('Custom')} = \text{True}
\]

Returns all objects that belong within the Polygon having a Name property of 'Custom'.

\[
\text{Not InNamedPolygon('Custom')} \\
\text{InNamedPolygon('Custom')} = \text{False}
\]

Returns all objects that don't belong to the Polygon object having a Name property of 'Custom'.

**InNet Membership Check**

**Description**
Returns each Net object having a Name property matching that specified by the Query; all of the child objects of each such Net object are also returned.

**Syntax**

\[
\text{InNet(Name : String)} : \text{Boolean} \\
\text{InNet(Name : String , Name : String)} : \text{Boolean} \\
\text{InNet(Name : String , Name : String , Name : String)} : \text{Boolean} \\
\ldots
\]

The '?' and '*' characters are "wild card" in nature; '?' replaces any single character, while '*' replaces any number of any characters.

**Examples**

\[
\text{InNet('GND')} \\
\text{InNet('GND')} = \text{True}
\]

Returns a Net object that has a Name property of 'GND', along with all objects which have a Net property of 'GND' (which are child objects of that Net object).

\[
\text{InNet('GND','+5V')} \\
\text{InNet('GND','+5V')} = \text{True}
\]

Returns Net objects that have a Name property of 'GND' or '+5V', along with all objects which have a Net property of 'GND' or '+5V' (which are child objects of those Net objects).

\[
\text{InNet('D?')} \\
\text{InNet('D?')} = \text{True}
\]
Returns all Net objects that have a Name property whose associated string starts with 'D' and has one following character, along with all objects which have a Net property whose associated string starts with 'D' and has one following character (which are child objects of the returned Net objects); examples of compliant Net names are 'D0', 'D1', etc.

**InNetClass Membership Check**

**Description**

Returns all Net objects that are members of the specified Net Class.

The user can define Net Classes as desired, and each Net within the file can either be a member, or not, of each Net Class which is so created.

The 'All Nets' Net Class exists in every PCB file by default; it includes all Nets in the file. It is not possible to change which Nets are members of that Net Class, but the user has full control over which Nets are members of any other Net Classes (which are created and named by the user).

**Syntax**

\[
\text{InNetClass}(\text{Name} : \text{String}) : \text{Boolean}
\]

The String specifies the Name property of the Net Class.

**Examples**

\[
\begin{align*}
\text{InNetClass('All Nets')} \\
\text{InNetClass('All Nets')} = \text{True}
\end{align*}
\]

Returns all Net objects. (All Nets are members of the (default) Net Class of 'All Nets'.)

\[
\begin{align*}
\text{InNetClass('Data Bus Nets')} \\
\text{InNetClass('Data Bus Nets')} = \text{True}
\end{align*}
\]

Returns all Net objects that are members of the user-defined Net Class with a Name property of 'Data Bus Nets'. (If that Net Class has not been created, or if no Nets within the file are members of it, then no objects are returned.)

**InPadClass Membership Check**

**Description**

Returns all Pad objects that are members of the specified Pad Class.

The user can define Pad Classes as desired, and each pad within the file can either be a member, or not, of each Pad Class which is so created.

The 'All Pads' Pad Class exists in every PCB file by default; it includes all pads in the file. It is not possible to change which pads are members of that Pad Class, but the user has full control over which pads are members of any other Pad Classes (which are created and named by the user).

**Syntax**

\[
\text{InPadClass}(\text{Name} : \text{String}) : \text{Boolean}
\]

The String specifies the Name property of the Pad Class.

**Examples**

\[
\begin{align*}
\text{InPadClass('All Pads')} \\
\text{InPadClass('All Pads')} = \text{True}
\end{align*}
\]

Returns all Pad objects. (All pads are members of the (default) Pad Class of 'All Pads'.)

\[
\begin{align*}
\text{InPadClass('Top Side Probe Pads')} \\
\text{InPadClass('Top Side Probe Pads')} = \text{True}
\end{align*}
\]

Returns all Pad objects that are members of the user-defined Pad Class with a Name property of 'Top Side Probe Pads'. (If that Pad Class has not been created, or if no pads within the file are members of it, then no objects are returned.)

**InPGA Attribute Check**

**Description**
Returns child objects of components that are of a PGA nature. Such a component contains at least 6 rows of pads and at least 6 columns of pads, the (absolute) difference between the number of rows of pads and the number of columns of pads is less than 6, the total pad count is at least half of the product of the number of rows of pads and the number of columns of pads, and at least one (and typically all) of the pads are of a through-hole nature.

Note: The parent objects of these component objects are not also returned.

**Syntax**

```
InPGA : Boolean
```

**Examples**

```
InPGA
InPGA = True
```

Returns all child objects of all Component objects that are of a PGA nature.

```
InPGA = False
```

Returns all objects except child objects of Component objects that are of a PGA nature.

**See Also**

InBGA
InDIP
InEdgeConnector
InLCC
InSIP
InSMSIP
InSMTComponent
InSOIC
InThruComponent

---

**InPoly Attribute Check**

**Description**

Returns child objects (arcs and tracks) of any polygon object on any Signal layer or on the Multi-Layer layer.

Note: The parent objects of these polygon objects are not also returned.

Note: This is an alias for InPolygon (Attribute Check).

**Syntax**

```
InPoly : Boolean
```

**Examples**

```
InPoly
InPoly = True
```

Returns Arc and Track objects which are child objects of any Polygon object on any Signal layer or on the Multi-Layer layer.

```
Not InPoly
InPoly = False
```

Returns all objects except Arc and Track objects which are child objects of any Polygon object on any Signal layer or on the Multi-Layer layer.

---

**InPolygon Attribute Check**

**Description**

Returns child objects (arcs and tracks) of any polygon object on any Signal layer or on the Multi-Layer layer.

Note: The parent objects of these polygon objects are not also returned.

Note: This is an alias for InPoly (Attribute Check).
Syntax

**InPolygon : Boolean**

**Examples**

InPolygon

InPolygon = True

Returns Arc and Track objects which are child objects of any Polygon object on any Signal layer or on the Multi-Layer layer.

Not InPolygon

InPolygon = False

Returns all objects except Arc and Track objects which are child objects of any Polygon object on any Signal layer or on the Multi-Layer layer.

### InPolygonClass Membership Check

**Description**

Returns all Polygon objects that are members of the specified Polygon Class.

The user can define Polygon Classes, and each Polygon object within the file can either be a member, or not, of each created Polygon Class.

The 'All Polygons' Polygon Class exists in every PCB document by default and includes all polygons in this PCB document. It is not possible to change which Polygon objects are members of that All Polygons Class, but the user has full control over which Polygon objects are members of other Polygon Classes.

**Syntax**

InPolygonClass(Name : String) : Boolean

The String specifies the Name property of the Polygon Class.

**Examples**

InPolygonClass('All Polygons')

InPolygonClass('All Polygons') = True

Returns all polygon objects of the (default) 'All Polygons' polygon class.

InPolygonClass('CustomPolygons')

InPolygonClass('CustomPolygons') = True

Returns all polygon objects that are members of the user-defined Polygon Class with a Name property of 'CustomPolygons'. (If that CustomPolygons Class has not been created, or if no polygons objects within the PCB document are members of it, then no objects are returned.)

### InRegion Membership Check

**Description**

Returns objects that reside within the region specified by the Query.

Note: The parent objects of group objects (Components, Coordinates, Dimensions, Nets, and Polygons) are not also returned.
All coordinates are referenced to the absolute origin (and not to the user origin or component insertion point), and at present, all coordinates always use units of mils. (1 mil = 0.001 inch)

Note: This is an alias for InRegionAbsolute (Membership Check).

**Syntax**

InRegion(X1 : Distance_String , Y1 : Distance_String , X2 : Distance_String , Y2 : Distance_String) : Boolean

Each instance of Distance_String is a numerical string that specifies a coordinate in the PCB (Library) file. The left and right boundaries of the region are specified by the X1 and X2 parameters, while the lower and upper boundaries of the region are specified by the Y1 and Y2 parameters.

**Example**

InRegion(12700,4700,13300,5000)
InRegion(12700,4700,13300,5000) = True

Returns objects residing within the rectangular region bounded by X = 12700 and X = 13300, and by Y = 4700 and Y = 5000; those coordinates are referenced to the absolute origin.

**InRegionAbsolute Membership Check**

**Description**

Returns objects that reside within the region specified by the Query.

Note: The parent objects of group objects (Components, Coordinates, Dimensions, Nets, and Polygons) are not also returned.

All coordinates are referenced to the absolute origin (and not to the user origin or component insertion point), and at present, all coordinates always use units of mils. (1 mil = 0.001 inch)

Note: This is an alias for InRegion (Membership Check).

**Syntax**

InRegionAbsolute(X1 : Distance_String , Y1 : Distance_String , X2 : Distance_String , Y2 : Distance_String) : Boolean

Each instance of Distance_String is a numerical string that specifies a coordinate in the PCB (Library) file. The left and right boundaries of the region are specified by the X1 and X2 parameters, while the lower and upper boundaries of the region are specified by the Y1 and Y2 parameters.

**Example**

InRegionAbsolute(12700,4700,13300,5000)
InRegionAbsolute(12700,4700,13300,5000) = True

Returns objects residing within the rectangular region bounded by X = 12700 and X = 13300, and by Y = 4700 and Y = 5000; those coordinates are referenced to the absolute origin.

**InRegionRelative Membership Check**

**Description**

Returns objects that reside within the region specified by the Query.

Note: The parent objects of group objects (Components, Coordinates, Dimensions, Nets, and Polygons) are not also returned.

All coordinates are referenced to the user origin (PCB files) or component insertion point (PCB Library files) (rather than to the absolute origin), and at present, all coordinates always use units of mils. (1 mil = 0.001 inch)

**Syntax**

InRegionRelative(X1 : Distance_String , Y1 : Distance_String , X2 : Distance_String , Y2 : Distance_String) : Boolean

Each instance of Distance_String is a numerical string that specifies a coordinate in the PCB (Library) file. The left and right boundaries of the region are specified by the X1 and X2 parameters, while the lower and upper boundaries of the region are specified by the Y1 and Y2 parameters.

**Example**

InRegionRelative(-700,-400,1300,1600)
InRegionRelative(-700,-400,1300,1600) = True

Returns objects residing within the rectangular region bounded by X = -700 and X = 1300, and by Y = -400 and Y = 1600; those coordinates are referenced to the user origin (or component insertion point).

**InSIP Attribute Check**

**Description**

Returns child objects of components that are of a SIP nature. Such a component contains at least four pads, and all of its pads are located in either the same row or the same column. Additionally, at least one (and typically all) of these pads are of a through-hole nature.

Note: The parent objects of these component objects are not also returned.
Syntax

InSIP : Boolean

Examples

InSIP
InSIP = True

Returns all child objects of all Component objects that are of a SIP nature (as defined above).

Not InSIP
InSIP = False

Returns all objects except child objects of Component objects that are of a SIP nature (as defined above).

See Also

InBGA
InDIP
InEdgeConnector
InLCC
InPGA
InSMSIP
InSMTComponent
InSOIC
InThruComponent

InSMSIP Attribute Check

Description

Returns child objects of components that are of a Surface Mount SIP nature. Such a component contains at least four pads, and all of its pads are located in either the same row or the same column. Additionally, all of these pads are of a surface mount nature.

Note: The parent objects of these component objects are not also returned.

Syntax

InSMSIP : Boolean

Examples

InSMSIP
InSMSIP = True

Returns all child objects of all Component objects that are of a Surface Mount SIP nature (as defined above).

Not InSMSIP
InSMSIP = False

Returns all objects except child objects of Component objects that are of a Surface Mount SIP nature (as defined above).

See Also

InBGA
InDIP
InEdgeConnector
InLCC
InPGA
InSIP
InSMTComponent
InSOIC
InThruComponent
InSMTComponent Attribute Check

Description
Returns child objects of components that are of a SMT nature. All of the pads of such a component are of a surface mount nature.
Note: The parent objects of these component objects are not also returned.

Syntax
InSMTComponent : Boolean

Examples
InSMTComponent
InSMTComponent = True

Returns all child objects of all Component objects that are of a SMT nature (as defined above).

Not InSMTComponent
InSMTComponent = False

Returns all objects except child objects of Component objects that are of a SMT nature (as defined above).

See Also
InBGA
InDIP
InEdgeConnector
InLCC
InPGA
InSIP
InSMSIP
InSOIC
InThruComponent

InSOIC Attribute Check

Description
Returns child objects of components that are of a Surface Mount DIP (SOIC) nature. All of the pads of such a component are of a surface mount nature, and the total pad count is equal to the product of the number of rows of pads and the number of columns of pads. Such a component also has either exactly two rows of pads and at least three columns of pads, or else has exactly two columns of pads and at least three rows of pads.
Note: The parent objects of these component objects are not also returned.

Syntax
InSOIC : Boolean

Examples
InSOIC
InSOIC = True

Returns all child objects of all Component objects that are of a Surface Mount DIP (SOIC) nature (as defined above).

Not InSOIC
InSOIC = False

Returns all objects except child objects of Component objects that are of a Surface Mount DIP (SOIC) nature (as defined above).

See Also
InBGA
InDIP
InEdgeConnector
InLCC
InPGA
InSIP
InSMSIP
InSMTComponent
InThruComponent

**InThruComponent Attribute Check**

**Description**
Returns child objects of components that are of a Thru-Hole nature. At least one (and typically all) of the pads of such a component are of a through-hole nature.

**Note:** The parent objects of these component objects are not also returned.

**Syntax**
InThruComponent : Boolean

**Examples**
InThruComponent
InThruComponent = True
Returns all child objects of all Component objects that are of a Thru-Hole nature (as defined above).

Not InThruComponent
InThruComponent = False
Returns all objects except child objects of Component objects that are of a Thru-Hole nature (as defined above).

**See Also**
InBGA
InDIP
InEdgeConnector
InLCC
InPGA
InSIP
InSMSIP
InSMTComponent
InSOIC

**InvertedTextBorder**

**Description**
Returns text objects having an InvertedTextBorder property that complies with the Query.

**Syntax**
InvertedTextBorder : : Boolean_String
Boolean_String must be either 'True' or 'False'.

**Examples**
InvertedTextBorder = 'True'
Returns all text objects that have the InvertedTextBorder setting set to true.
InvertedTextBorder = 'False'
Returns all the text objects that have the InvertedTextBorder setting set to false as well as other objects that don't have the InvertedTextBorder property.
IsAngularDimension Object Type Check

Description
Returns Angular Dimension objects.

Syntax
IsAngularDimension : Boolean

Examples
IsAngularDimension
IsAngularDimension = True
Returns all Angular Dimension objects.
Not IsAngularDimension
IsAngularDimension = False
Returns all objects except Angular Dimension objects.

IsArc Object Type Check (PCB)

Description
Returns Arc objects only.

Syntax
IsArc : Boolean

Examples
IsArc
IsArc = True
Returns all Arc objects.
Not IsArc
IsArc = False
Returns all objects except Arc objects.

IsBaselineDimension Object Type Check

Description
Returns Baseline Dimension objects.

Syntax
IsBaselineDimension : Boolean

Examples
IsBaselineDimension
IsBaselineDimension = True
Returns all Baseline Dimension objects.
Not IsBaselineDimension
IsBaselineDimension = False
Returns all objects except Baseline Dimension objects.

IsBGA Attribute Check

Description
Returns components that are of a BGA nature. Such a component contains at least 10 rows of pads and at least 10 columns of pads, the (absolute) difference between the number of rows of pads and the number of columns of pads is less than 6, the total
pad count is at least half of the product of the number of rows of pads and the number of columns of pads, and all of the pads are of a surface mount nature.

Note: The child objects of these component objects are not also returned.

Syntax

\[
\text{IsBGA} : \text{Boolean}
\]

Examples

\[
\text{IsBGA} = \text{True}
\]

Returns all Component objects that are of a BGA nature (as defined above).

\[
\text{Not IsBGA}
\]

\[
\text{IsBGA} = \text{False}
\]

Returns all objects except Component objects that are of a BGA nature (as defined above).

See Also

\[
\text{IsDIP}
\]

\[
\text{IsEdgeConnector}
\]

\[
\text{IsLCC}
\]

\[
\text{IsPGA}
\]

\[
\text{IsSIP}
\]

\[
\text{IsSMSIP}
\]

\[
\text{IsSMTComponent}
\]

\[
\text{IsSOIC}
\]

\[
\text{IsThruComponent}
\]

IsBoard Object Type Check

Description

In its affirmative (True) state, returns none of the objects whose properties can be displayed in the 'List' Workspace Panel, and displays just the region occupied by the PCB.

In its negated (False) state, returns all of the objects whose properties can be displayed in the 'List' Workspace Panel, while still displaying the region occupied by the PCB.

Syntax

\[
\text{IsBoard} : \text{Boolean}
\]

Examples

\[
\text{IsBoard} = \text{True}
\]

Returns none of the objects whose properties can be displayed in the 'List' Workspace Panel, and displays just the region occupied by the PCB.

\[
\text{Not IsBoard}
\]

\[
\text{IsBoard} = \text{False}
\]

Returns all of the objects whose properties can be displayed in the 'List' Workspace Panel, while still displaying the region occupied by the PCB.

IsBoardOutline Object Type Check

Description

In its affirmative (True) state, returns none of the objects whose properties can be displayed in the 'List' Workspace Panel, and displays just the region occupied by the PCB.
In its negated (False) state, returns all of the objects whose properties can be displayed in the 'List' Workspace Panel, while still displaying the region occupied by the PCB.

**Syntax**

```plaintext
IsBoardOutline : Boolean
```

**Examples**

```plaintext
IsBoardOutline
IsBoardOutline = True
```

Returns none of the objects whose properties can be displayed in the 'List' Workspace Panel, and displays just the region occupied by the PCB.

```plaintext
Not IsBoardOutline
IsBoardOutline = False
```

Returns all of the objects whose properties can be displayed in the 'List' Workspace Panel, while still displaying the region occupied by the PCB.

### IsBold Field

**Description**

Returns all Coordinate, Dimension, and Text objects having a True Type Font and as well as that IsBold property that complies with the Query.

**Note:** The child objects of these Coordinate and Dimension objects are not also returned.

**Syntax**

```plaintext
IsBold : Boolean_String
```

**Boolean_String** must be either 'True' or 'False'.

**Examples**

```plaintext
IsBold = 'True'
```

Returns all Coordinate, Original Dimension, and Text objects that are bold.

```plaintext
IsBold <> 'True'
```

Returns all Coordinate, Dimension, and Text objects that are not bolded.

### IsCenterDimension Object Type Check

**Description**

Returns Center Dimension objects.

**Syntax**

```plaintext
IsCenterDimension : Boolean
```

**Examples**

```plaintext
IsCenterDimension
IsCenterDimension = True
```

Returns all Center Dimension objects.

```plaintext
Not IsCenterDimension
IsCenterDimension = False
```

Returns all objects except Center Dimension objects.

### IsClass Object Type Check

**Description**

Returns Class objects.
IsClass : Boolean

Examples
IsClass
IsClass = True
Returns all Class objects.
Not IsClass
IsClass = False
Returns all objects except Class objects.

IsComment Object Type Check (PCB)

Description
Returns Text objects having a 'Comment' String Type property; these are child objects of Component objects, and display the (Component) Comment property of their parent objects.

Syntax
IsComment : Boolean

Examples
IsComment
IsComment = True
Returns all Text objects having a 'Comment' String Type property.
Not IsComment
IsComment = False
Returns all objects except Text objects having a 'Comment' String Type property.

IsComponent Object Type Check

Description
Returns Component objects.

Syntax
IsComponent : Boolean

Examples
IsComponent
IsComponent = True
Returns all Component objects.
Not IsComponent
IsComponent = False
Returns all objects except Component objects.

IsComponentArc Object Type Check

Description
Returns Arc objects which are child objects of Component objects.

Syntax
IsComponentArc : Boolean

Examples
IsComponentArc
IsComponentArc = True
Returns all Arc objects which are child objects of Component objects.

Not IsComponentArc
IsComponentArc = False
Returns all objects except Arc objects which are child objects of Component objects.

**IsComponentFill Object Type Check**

**Description**
Returns Fill objects which are child objects of Component objects.

**Syntax**
IsComponentFill : Boolean

**Examples**
IsComponentFill
IsComponentFill = True
Returns all Fill objects which are child objects of Component objects.
Not IsComponentFill
IsComponentFill = False
Returns all objects except Fill objects which are child objects of Component objects.

**IsComponentPad Object Type Check**

**Description**
Returns Pad objects which are child objects of Component objects.

**Syntax**
IsComponentPad : Boolean

**Examples**
IsComponentPad
IsComponentPad = True
Returns all Pad objects which are child objects of Component objects.
Not IsComponentPad
IsComponentPad = False
Returns all objects except Pad objects which are child objects of Component objects.

**IsComponentPrimitive Attribute Check**

**Description**
Returns objects of a Primitive type (i.e. arcs, fills, pads, strings, tracks, and vias) which are child objects of some Component object.

Note: The parent objects of these component objects are not also returned.

**Syntax**
IsComponentPrimitive : Boolean

**Examples**
IsComponentPrimitive
IsComponentPrimitive = True
Returns all Arc, Fill, Pad, Text, Track, and Via objects which are child objects of any Component object.
Not IsComponentPrimitive
IsComponentPrimitive = False
Returns all objects except Arc, Fill, Pad, Text, Track, and Via objects which are child objects of any Component object.
IsComponentSelected Attribute Check

Description
Returns objects of a Primitive type (i.e. arcs, fills, pads, strings, tracks, and vias) which are child objects of a Component object that is currently in a selected state.

Note: The parent objects of these component objects are not also returned.

Syntax
IsComponentSelected : Boolean

Examples
IsComponentSelected
IsComponentSelected = True

Returns all Arc, Fill, Pad, Text, Track, and Via objects which are child objects of any Component object that is currently in a selected state.

Not IsComponentSelected
IsComponentSelected = False

Returns all objects except Arc, Fill, Pad, Text, Track, and Via objects which are child objects of any Component object that is currently in a selected state.

IsComponentText Object Type Check

Description
Returns Text objects which are child objects of Component objects.

Syntax
IsComponentText : Boolean

Examples
IsComponentText
IsComponentText = True

Returns all Text objects which are child objects of Component objects.

Not IsComponentText
IsComponentText = False

Returns all objects except Text objects which are child objects of Component objects.

IsComponentTrack Object Type Check

Description
Returns Track objects which are child objects of Component objects.

Syntax
IsComponentTrack : Boolean

Examples
IsComponentTrack
IsComponentTrack = True

Returns all Track objects which are child objects of Component objects.

Not IsComponentTrack
IsComponentTrack = False

Returns all objects except Track objects which are child objects of Component objects.
IsComponentVia Object Type Check

Description
Returns Via objects which are child objects of Component objects.

Syntax
IsComponentVia : Boolean

Examples
IsComponentVia
IsComponentVia = True
Returns all Via objects which are child objects of Component objects.
Not IsComponentVia
IsComponentVia = False
Returns all objects except Via objects which are child objects of Component objects.

IsConnection Object Type Check

Description
Returns the "connection" details, which consists of what is displayed on the 'Connections and From Tos' (System) layer.
Note: These "connection" details do not have their properties displayed in the 'List' Workspace Panel.

Syntax
IsConnection: Boolean

Examples
IsConnection
IsConnection = True
Returns the "connection" details, which consists of what is displayed on the 'Connections and From Tos' (System) layer.
IsConnection
IsConnection = False
Returns all objects; the "connection" details are also displayed, because they are (un-listed) child objects of Net objects, which are returned by this Query.
Not IsConnection And Not IsNet
(IsConnection = False) And (IsNet = False)
Returns all objects except for Net objects; the "connection" details are not displayed, because they are (un-listed) child objects of Net objects, and no Net objects are returned by this Query.

IsCoordinate Object Type Check

Description
Returns Coordinate objects.

Syntax
IsCoordinate : Boolean

Examples
IsCoordinate
IsCoordinate = True
Returns all Coordinate objects.
Not IsCoordinate
IsCoordinate = False
Returns all objects except Coordinate objects.
IsCopperRegion Object Type Check

Description
Returns region objects.

Syntax
IsCopperRegion : Boolean

Examples
IsCopperRegion
IsCopperRegion = True
Returns all region objects.
Not IsCopperRegion
IsCopperRegion = False
Returns all objects except Region objects.

IsCutoutRegion Object Type Check

Description
Returns cutout region objects.

Syntax
IsCutoutRegion : Boolean

Examples
IsCutoutRegion
IsCutoutRegion = True
Returns all region objects.
Not IsCutoutRegion
IsCutoutRegion = False
Returns all objects except Cutout Region objects.

IsDatumDimension Object Type Check

Description
Returns Datum Dimension objects.

Syntax
IsDatumDimension : Boolean

Examples
IsDatumDimension
IsDatumDimension = True
Returns all Datum Dimension objects.
Not IsDatumDimension
IsDatumDimension = False
Returns all objects except Datum Dimension objects.

IsDesignator Object Type Check (PCB)

Description
Returns Text objects having a 'Designator' String Type property; these are child objects of Component objects, and display the Name (/ Designator) property of their parent objects.

Syntax
IsDesignator : Boolean

Examples
IsDesignator
IsDesignator = True
Returns all Text objects having a 'Designator' String Type property.
Not IsDesignator
IsDesignator = False
Returns all objects except Text objects having a 'Designator' String Type property.

IsDifferentialPair Object Type Check

Description
Returns differential pair objects.
Syntax
IsDifferentialPair : Boolean
Examples
IsDifferentialPair
IsDifferentialPair = True
Returns all Differential Pair objects.
Not IsDifferentialPair
IsDifferentialPair = False
Returns all objects except differential pair objects.

IsDimension Object Type Check

Description
Returns all types of Dimension objects.
Syntax
IsDimension : Boolean
Examples
IsDimension
IsDimension = True
Returns all types of Dimension objects, thus returning all Angular Dimension, Baseline Dimension, Center Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Original Dimension, Radial Diameter Dimension, and Radial Dimension objects.
Not IsDimension
IsDimension = False
Returns all objects except Dimension objects, thus returning all objects except Angular Dimension, Baseline Dimension, Center Dimension, Datum Dimension, Leader Dimension, Linear Diameter Dimension, Linear Dimension, Original Dimension, Radial Diameter Dimension, and Radial Dimension objects.

IsDIP Attribute Check

Description
Returns components that are of a DIP nature. At least one (and typically all) of the pads of such a component are of a through-hole nature, and the total pad count is equal to the product of the number of rows of pads and the number of columns of pads. Such a component also has either exactly two rows of pads and at least three columns of pads, or else has exactly two columns of pads and at least three rows of pads.
Note: The child objects of these component objects are not also returned.
Syntax

IsDIP : Boolean

Examples

IsDIP
IsDIP = True

Returns all Component objects that are of a DIP nature.

Not IsDIP
IsDIP = False

Returns all objects except Component objects that are of a DIP nature.

See Also

IsBGA
IsEdgeConnector
IsLCC
IsPGA
IsSIP
IsSMSIP
IsSMTComponent
IsSOIC
IsThruComponent

IsEdgeConnector Attribute Check

Description

Returns components that are of an Edge Connector nature. All of the pads of such a component are of a surface mount nature, and these are located on both sides of the PCB. All of the pads reside in either the same row or the same column, the number of pads on either side of the PCB is less than three times the number of pads on the opposite side of the PCB, and the total number of pads is at least 4.

Note: The child objects of these component objects are not returned.

Syntax

IsEdgeConnector : Boolean

Examples

IsEdgeConnector
IsEdgeConnector = True

Returns all Component objects that are of an Edge Connector nature (as defined above).

Not IsEdgeConnector
IsEdgeConnector = False

Returns all objects except Component objects that are of an Edge Connector nature (as defined above).

See Also

IsBGA
IsDIP
IsLCC
IsPGA
IsSIP
IsSMSIP
IsSMTComponent
IsSOIC
IsThruComponent
**IsElectrical Attribute Check**

**Description**
Returns objects of a Primitive type (i.e. arcs, fills, pads, tracks, and vias) which have a Net property, i.e. those objects whose Net property is *not* 'No Net'.

Only objects on the Signal layers or Multi-Layer layer can have a Net property, and Text objects do not have a Net property (regardless of their Layer property).

**Syntax**
IsElectrical : Boolean

**Examples**
IsElectrical = True

Returns Arc, Fill, Pad, Track, and Via objects which have a Net property.

Not IsElectrical
IsElectrical = False

Returns all objects *except* Arc, Fill, Pad, Track, and Via objects which do *not* have a Net property of 'No Net'.

**IsEmbeddedBoard Object Type Check**

**Description**
Returns embedded board array objects.

**Syntax**
IsEmbeddedBoard : Boolean

**Examples**
IsEmbeddedBoard
IsEmbeddedBoard = True

Returns all embedded board array objects.

Not IsEmbeddedBoard
IsEmbeddedBoard = False

Returns all objects *except* Embedded Board array objects.

**IsFill Object Type Check**

**Description**
Returns Fill objects.

**Syntax**
IsFill : Boolean

**Examples**
IsFill
IsFill = True

Returns all Fill objects.

Not IsFill
IsFill = False

Returns all objects *except* Fill objects.

**IsFromTo Object Type Check**

**Description**
Returns From To objects.

**Syntax**

IsFromTo : Boolean

**Examples**

IsFromTo
IsFromTo = True
Returns all From To objects.

Not IsFromTo
IsFromTo = False
Returns all objects except From To objects.

### IsGraphicalComponent Attribute Check

**Description**

Returns Component objects whose Component Type property is 'Graphical'; such components are neither listed in any Bill of Materials files produced from the PCB file, nor synchronised to any components contained within the Schematic file(s) linked to the PCB file.

Note: The child objects of these component objects are not also returned.

**Syntax**

IsGraphicalComponent : Boolean

**Examples**

IsGraphicalComponent
IsGraphicalComponent = True
Returns Component objects that have a 'Graphical' Component Type property.

Not IsGraphicalComponent
IsGraphicalComponent = False
Returns all objects except Component objects that have a 'Graphical' Component Type property.

**See Also**

IsMechanicalComponent
IsNetTieComponent
IsNoBOMComponent
IsStandardComponent

### IsHorizontal Attribute Check

**Description**

Returns horizontal tracks, i.e. those tracks whose Y1 and Y2 properties are identical.

**Syntax**

IsHorizontal : Boolean

**Examples**

IsHorizontal
IsHorizontal = True
Returns all Track objects that have identical Y1 and Y2 properties.

Not IsHorizontal
IsHorizontal = False
Returns all objects except Track objects that have identical Y1 and Y2 properties.

**See Also**
IsOblique
IsObliqueNegative
IsObliquePositive
IsVertical

**IsItalic Field**

**Description**
Returns all Coordinate, Dimension, and Text objects having a True Type Font and as well as that IsItalic property that complies with the Query.

Note: The child objects of these Coordinate and Dimension objects are **not** also returned.

**Syntax**
IsItalic : Boolean

Boolean_String must be either 'True' or 'False'.

**Examples**
IsItalic = 'True'
Returns all Coordinate, Original Dimension, and Text objects that are italicised.

IsItalic <> 'True'
Not (IsItalic = 'True')
Returns all Coordinate, Dimension, and Text objects that are not italicised.

**IsKeepOut Attribute Check**

**Description**
Returns objects whose Layer property is the Keep-Out layer, and arc, fill and track objects whose Keepout property is True.

**Syntax**
IsKeepOut : Boolean

**Examples**
IsKeepOut
IsKeepOut = True
Returns all objects on the Keep-Out layer, and Arc, Fill and Track objects with a True Keepout property.

Not IsKeepOut
IsKeepOut = False
Returns all objects except objects on the Keep-Out layer, and Arc, Fill and Track objects with a True Keepout property.

**IsLCC Attribute Check**

**Description**
Returns components that are of a Surface Mount LCC nature. Such a component contains 2 rows of pads which each contain at least 4 pads, and 2 columns of pads which each contain at least 4 pads, and all of the pads are of a surface mount nature.

Note: The child objects of these component objects are not also returned.

**Syntax**
IsLCC : Boolean

**Examples**
IsLCC
IsLCC = True
Returns all Component objects that are of a Surface Mount LCC nature (as defined above).

Not IsLCC
IsLCC = False
Returns all objects except Component objects that are of a Surface Mount LCC nature (as defined above).

See Also
IsBGA
IsDIP
IsEdgeConnector
IsPGA
IsSIP
IsSMSIP
IsSMTComponent
IsSOIC
IsThruComponent

IsLeaderDimension Object Type Check

Description
Returns Leader Dimension objects.

Syntax
IsLeaderDimension : Boolean

Examples
IsLeaderDimension
IsLeaderDimension = True
Returns all Leader Dimension objects.
Not IsLeaderDimension
IsLeaderDimension = False
Returns all objects except Leader Dimension objects.

IsLinearDiameterDimension Object Type Check

Description
Returns Linear Diameter Dimension objects.

Syntax
IsLinearDiameterDimension : Boolean

Examples
IsLinearDiameterDimension
IsLinearDiameterDimension = True
Returns all Linear Diameter Dimension objects.
Not IsLinearDiameterDimension
IsLinearDiameterDimension = False
Returns all objects except Linear Diameter Dimension objects.

IsLinearDimension Object Type Check

Description
Returns Linear Dimension objects.

Syntax
IsLinearDimension : Boolean
**Examples**

IsLinearDimension

IsLinearDimension = True

Returns all Linear Dimension objects.

Not IsLinearDimension

IsLinearDimension = False

Returns all objects except Linear Dimension objects.

---

**IsMechanicalComponent Attribute Check**

**Description**

Returns Component objects whose Component Type property is 'Mechanical'; such components are listed in Bill of Materials files produced from the PCB file. If such components exist, they are also synchronised to corresponding components contained within the Schematic file(s) linked to the PCB file.

Note: The child objects of these component objects are not also returned.

**Syntax**

IsMechanicalComponent : Boolean

**Examples**

IsMechanicalComponent

IsMechanicalComponent = True

Returns Component objects that have a 'Mechanical' Component Type property.

Not IsMechanicalComponent

IsMechanicalComponent = False

Returns all objects except Component objects that have a 'Mechanical' Component Type property.

**See Also**

IsGraphicalComponent

IsNetTieComponent

IsNoBOMComponent

IsStandardComponent

---

**IsNet Object Type Check**

**Description**

Returns Net objects.

**Syntax**

IsNet : Boolean

**Examples**

IsNet

IsNet = True

Returns all Net objects.

Not IsNet

IsNet = False

Returns all objects except Net objects.

---

**IsNetSelected Attribute Check**

**Description**
Returns objects having a Net property whose corresponding Net object is currently in a selected state; the types of objects returned are arcs, fills, pads, tracks, vias, and polygons.
The selected state of each Net object can be changed from the List Workspace Panel.
Note: The selected Net objects are not also returned, and nor are the child objects of any polygons which are returned.

**Syntax**

`IsNetSelected : Boolean`

**Examples**

`IsNetSelected`

`IsNetSelected = True`

Returns all Arc, Fill, Pad, Track, Via, and Polygon objects that have a Net property whose corresponding Net object is currently in a selected state.

`Not IsNetSelected`

`IsNetSelected = False`

Returns all objects except Arc, Fill, Pad, Track, Via, and Polygon objects that have a Net property whose corresponding Net object is currently in a selected state.

**IsNetTieComponent Attribute Check**

**Description**

Returns Component objects whose Component Type property is 'Net Tie' or 'Net Tie (In BOM)'. Such components are used to permit different nets to be electrically connected to one another in a controlled manner, and without generating any Design Rule Violations. Both of these types of components are synchronised to corresponding components contained within the Schematic file(s) linked to the PCB file, but while components having a ‘Net Tie (In BOM)’ Component Type property are also listed in Bill of Materials files produced from the PCB file, components having a ‘Net Tie’ Component Type property are not.

Note: The child objects of these component objects are not also returned.

**Syntax**

`IsNetTieComponent : Boolean`

**Examples**

`IsNetTieComponent`

`IsNetTieComponent = True`

Returns Component objects that have a ‘Net Tie’ or ‘Net Tie (In BOM)’ Component Type property.

`Not IsNetTieComponent`

`IsNetTieComponent = False`

Returns all objects except Component objects that have a ‘Net Tie’ or ‘Net Tie (In BOM)’ Component Type property.

**See Also**

- `IsGraphicalComponent`
- `IsMechanicalComponent`
- `IsNoBOMComponent`
- `IsStandardComponent`

**IsNoBOMComponent Attribute Check**

**Description**

Returns Component objects whose Component Type property is ‘Standard (No BOM)’; such components are synchronised to corresponding components contained within the Schematic file(s) linked to the PCB file, but are not listed in Bill of Materials files produced from the PCB file.

Note: The child objects of these component objects are not also returned.

**Syntax**

`IsNoBOMComponent : Boolean`
Examples
IsNoBOMComponent
IsNoBOMComponent = True
Returns Component objects that have a 'Standard (No BOM)' Component Type property.
Not IsNoBOMComponent
IsNoBOMComponent = False
Returns all objects except Component objects that have a 'Standard (No BOM)' Component Type property.
See Also
IsGraphicalComponent
IsMechanicalComponent
IsNetTieComponent
IsStandardComponent

IsOblique Attribute Check

Description
Returns tracks of a diagonal nature if these are on either a rising 45 degrees angle or a falling 45 degrees angle, i.e. those tracks for which the absolute value of the difference between their Y1 and Y2 properties is equal to the absolute value of the difference between their X1 and X2 properties.
Syntax
IsOblique : Boolean
Examples
IsOblique
IsOblique = True
Returns all Track objects on a rising 45 degrees angle or a falling 45 degrees angle.
Not IsOblique
IsOblique = False
Returns all objects except Track objects on a rising 45 degrees angle or a falling 45 degrees angle.
See Also
IsHorizontal
IsObliqueNegative
IsObliquePositive
IsVertical

IsObliqueNegative Attribute Check

Description
Returns tracks of a diagonal nature if these are on a falling 45 degrees angle, i.e. those tracks for which the difference between their Y1 and Y2 properties is opposite in sign to, but otherwise equal, to the difference between their X1 and X2 properties.
Syntax
IsObliqueNegative : Boolean
Examples
IsObliqueNegative
IsObliqueNegative = True
Returns all Track objects on a falling 45 degrees angle.
Not IsObliqueNegative
IsObliqueNegative = False
Returns all objects except Track objects on a falling 45 degrees angle.

See Also
IsHorizontal
IsOblique
IsObliquePositive
IsVertical

IsObliquePositive Attribute Check

Description
Returns tracks of a diagonal nature if these are on a rising 45 degrees angle, i.e. those tracks for which the difference between their Y1 and Y2 properties is equal (including in sign) to the difference between their X1 and X2 properties.

Syntax
IsObliquePositive : Boolean

Examples
IsObliquePositive
IsObliquePositive = True

Returns all Track objects on a rising 45 degrees angle.
Not IsObliquePositive
IsObliquePositive = False

Returns all objects except Track objects on a rising 45 degrees angle.

See Also
IsHorizontal
IsOblique
IsObliqueNegative
IsVertical

IsOriginalDimension Object Type Check

Description
Returns Original Dimension objects.

Syntax
IsOriginalDimension : Boolean

Examples
IsOriginalDimension
IsOriginalDimension = True

Returns all Original Dimension objects.
Not IsOriginalDimension
IsOriginalDimension = False

Returns all objects except Original Dimension objects.

IsPad Object Type Check

Description
Returns Pad objects.

Syntax
IsPad : Boolean

Examples
IsPad
IsPad = True
Returns all Pad objects.
Not IsPad
IsPad = False
Returns all objects except Pad objects.

**IsPGA Attribute Check**

**Description**
Returns components that are of a PGA nature. Such a component contains at least 6 rows of pads and at least 6 columns of pads, the (absolute) difference between the number of rows of pads and the number of columns of pads is less than 6, the total pad count is at least half of the product of the number of rows of pads and the number of columns of pads, and at least one (and typically all) of the pads are of a through-hole nature.

Note: The child objects of these component objects are not also returned.

**Syntax**
IsPGA : Boolean

**Examples**
IsPGA
IsPGA = True
Returns all Component objects that are of a PGA nature (as defined above).
Not IsPGA
IsPGA = False
Returns all objects except Component objects that are of a PGA nature (as defined above).

**See Also**
IsBGA
IsDIP
IsEdgeConnector
IsLCC
IsSiP
IsSMTComponent
IsSOIC
IsThruComponent

**IsPoly Object Type Check**

**Description**
Returns Polygon objects.

Note: This is an alias for IsPolygon (Object Type Check).

**Syntax**
IsPoly : Boolean

**Examples**
IsPoly
IsPoly = True
Returns all Polygon objects.
Not IsPoly
IsPoly = False
Returns all objects except Polygon objects.

**IsPolygon Object Type Check (PCB)**

**Description**  
Returns Polygon objects.

*Note:* This is an alias for IsPoly (Object Type Check).

**Syntax**  
IsPolygon : Boolean

**Examples**  
IsPolygon
IsPolygon = True
Returns all Polygon objects.
Not IsPolygon
IsPolygon = False
Returns all objects except Polygon objects.

**IsRadialDiameterDimension Object Type Check**

**Description**  
Returns Radial Diameter Dimension objects.

**Syntax**  
IsRadialDiameterDimension : Boolean

**Examples**  
IsRadialDiameterDimension
IsRadialDiameterDimension = True
Returns all Radial Diameter Dimension objects.
Not IsRadialDiameterDimension
IsRadialDiameterDimension = False
Returns all objects except Radial Diameter Dimension objects.

**IsRadialDimension Object Type Check**

**Description**  
Returns Radial Dimension objects.

**Syntax**  
IsRadialDimension : Boolean

**Examples**  
IsRadialDimension
IsRadialDimension = True
Returns all Radial Dimension objects.
Not IsRadialDimension
IsRadialDimension = False
Returns all objects except Radial Dimension objects.
IsRectangularPad Attribute Check

Description
Returns pads on the Top, Bottom, and Keep-Out layers having a 'Rectangle' Shape property, and pads on the Multi-Layer layer having a 'Rectangle' Shape property on at least one external layer.

Syntax
IsRectangularPad : Boolean

Examples
IsRectangularPad
IsRectangularPad = True
Returns pads on the Top, Bottom, and Keep-Out layers having a 'Rectangle' Shape property, and pads on the Multi-Layer layer having a 'Rectangle' Shape property on at least one external layer.
Not IsRectangularPad
IsRectangularPad = False
Returns all objects except pads on the Top, Bottom, and Keep-Out layers having a 'Rectangle' Shape property, and pads on the Multi-Layer layer having a 'Rectangle' Shape property on at least one external layer.

See Also
IsRoundPad
IsSquarePad

IsRegion Object Type Check

Description
Returns region objects.

Syntax
IsRegion : Boolean

Examples
IsRegion
IsRegion = True
Returns all region objects.
Not IsRegion
IsRegion = False
Returns all objects except region objects.

IsRoom Object Type Check

Description
Returns Confinement Constraint Rule (/ Room) objects.
Note: This is an alias for IsRule_ConfinementConstraint (Object Type Check).

Syntax
IsRoom : Boolean

Examples
IsRoom
IsRoom = True
Returns all Confinement Constraint Rule (/ Room) objects.
Not IsRoom
IsRoom = False
Returns all objects except Confinement Constraint Rule (/ Room) objects.
IsRoundPad Attribute Check

**Description**
Returns pads on the Top, Bottom, and Keep-Out layers having a 'Round' Shape property, and pads on the Multi-Layer layer having a 'Round' Shape property on at least one external layer.

**Syntax**
IsRoundPad : Boolean

**Examples**
IsRoundPad
IsRoundPad = True
Returns pads on the Top, Bottom, and Keep-Out layers having a 'Round' Shape property, and pads on the Multi-Layer layer having a 'Round' Shape property on at least one external layer.

Not IsRoundPad
IsRoundPad = False

Returns all objects except pads on the Top, Bottom, and Keep-Out layers having a 'Round' Shape property, and pads on the Multi-Layer layer having a 'Round' Shape property on at least one external layer.

**See Also**
IsRectangularPad
IsSquarePad

IsRule Object Type Check

**Description**
Returns all types of (Design) Rule objects.

**Syntax**
IsRule : Boolean

**Examples**
IsRule
IsRule = True


Not IsRule
IsRule = False

Rule, SMD To Plane Rule, Solder Mask Expansion Rule, Supply Nets Rule, Test Point Style Rule, Test Point Usage Rule, Unconnected Pin Rule, Undershoot Falling Edge Rule, Undershoot Rising Edge Rule, and Vias Under SMD Rule objects.

**IsRule_AcuteAngle Object Type Check**

**Description**

Returns Acute Angle Rule objects.

**Syntax**

IsRule_AcuteAngle : Boolean

**Examples**

IsRule_AcuteAngle  
IsRule_AcuteAngle = True  
Returns all Acute Angle Rule objects.  
Not IsRule_AcuteAngle  
IsRule_AcuteAngle = False  
Returns all objects except Acute Angle Rule objects.

**IsRule_BrokenNets Object Type Check**

**Description**

Returns Broken Nets Rule objects.

**Syntax**

IsRule_BrokenNets : Boolean

**Examples**

IsRule_BrokenNets  
IsRule_BrokenNets = True  
Returns all Broken Nets Rule objects.  
Not IsRule_BrokenNets  
IsRule_BrokenNets = False  
Returns all objects except Broken Nets Rule objects.

**IsRule_Clearance Object Type Check**

**Description**

Returns Clearance Rule objects.

**Syntax**

IsRule_Clearance : Boolean

**Examples**

IsRule_Clearance  
IsRule_Clearance = True  
Returns all Clearance Rule objects.  
Not IsRule_Clearance  
IsRule_Clearance = False  
Returns all objects except Clearance Rule objects.

**IsRule_ComponentClearance Object Type Check**

**Description**
Returns Component Clearance Rule objects.

**Syntax**

IsRule_ComponentClearance : Boolean

**Examples**

IsRule_ComponentClearance

IsRule_ComponentClearance = True

Returns all Component Clearance Rule objects.

Not IsRule_ComponentClearance

IsRule_ComponentClearance = False

Returns all objects except Component Clearance Rule objects.

---

**IsRule_ComponentRotations Object Type Check**

**Description**

Returns Component Rotations Rule objects.

**Syntax**

IsRule_ComponentRotations : Boolean

**Examples**

IsRule_ComponentRotations

IsRule_ComponentRotations = True

Returns all Component Rotations Rule objects.

Not IsRule_ComponentRotations

IsRule_ComponentRotations = False

Returns all objects except Component Rotations Rule objects.

---

**IsRule_ConfinementConstraint Object Type Check**

**Description**

Returns Confinement Constraint Rule (/ Room) objects.

Note: This is an alias for IsRoom (Object Type Check).

**Syntax**

IsRule_ConfinementConstraint : Boolean

**Examples**

IsRule_ConfinementConstraint

IsRule_ConfinementConstraint = True

Returns all Confinement Constraint Rule (/ Room) objects.

Not IsRule_ConfinementConstraint

IsRule_ConfinementConstraint = False

Returns all objects except Confinement Constraint Rule (/ Room) objects.

---

**IsRule_DaisyChainStubLength Object Type Check**

**Description**

Returns Daisy Chain Stub Length Rule objects.

**Syntax**

IsRule_DaisyChainStubLength : Boolean

**Examples**
IsRule_DaisyChainStubLength
IsRule_DaisyChainStubLength = True
Returns all Daisy Chain Stub Length Rule objects.
Not IsRule_DaisyChainStubLength
IsRule_DaisyChainStubLength = False
Returns all objects except Daisy Chain Stub Length Rule objects.

IsRule_DifferentialPairsRouting Object Type Check

Description
Returns Differential Pairs Rule objects.

Syntax
IsRule_DifferentialPairsRouting : Boolean

Examples
IsRule_DifferentialPairsRouting
IsRule_DifferentialPairsRouting = True
Returns all Differential Pairs Routing Rule objects.
Not IsRule_DifferentialPairsRouting
IsRule_DifferentialPairsRouting = False
Returns all objects except Differential Pairs Routing Rules objects.

IsRule_FanoutControl Object Type Check

Description
Returns Fanout Control Rule objects.

Syntax
IsRule_FanoutControl : Boolean

Examples
IsRule_FanoutControl
IsRule_FanoutControl = True
Returns all Fanout Control Rule objects.
Not IsRule_FanoutControl
IsRule_FanoutControl = False
Returns all objects except Fanout Control Rule objects.

IsRule_FlightTime_FallingEdge Object Type Check

Description
Returns Flight Time Falling Edge Rule objects.

Syntax
IsRule_FlightTime_FallingEdge : Boolean

Examples
IsRule_FlightTime_FallingEdge
IsRule_FlightTime_FallingEdge = True
Returns all Flight Time Falling Edge Rule objects.
Not IsRule_FlightTime_FallingEdge
IsRule_FlightTime_FallingEdge = False
IsRule_FlightTime_RisingEdge Object Type Check

Description
Returns Flight Time Rising Edge Rule objects.

Syntax
IsRule_FlightTime_RisingEdge : Boolean

Examples
IsRule_FlightTime_RisingEdge
IsRule_FlightTime_RisingEdge = True
Returns all Flight Time Rising Edge Rule objects.
Not IsRule_FlightTime_RisingEdge
IsRule_FlightTime_RisingEdge = False
Returns all objects except Flight Time Rising Edge Rule objects.

IsRule_LayerPair Object Type Check

Description
Returns Layer Pair Rule objects.

Syntax
IsRule_LayerPair : Boolean

Examples
IsRule_LayerPair
IsRule_LayerPair = True
Returns all Layer Pair Rule objects.
Not IsRule_LayerPair
IsRule_LayerPair = False
Returns all objects except Layer Pair Rule objects.

IsRule_LayerStack Object Type Check

Description
Provided to return Layer Stack Rule objects. However, as a corresponding type of Design Rule has yet to be implemented, it currently returns no objects in its affirmative (True) state, and all objects in its negated (False) state.

Syntax
IsRule_LayerStack : Boolean

Examples
IsRule_LayerStack
IsRule_LayerStack = True
Returns no objects.
Not IsRule_LayerStack
IsRule_LayerStack = False
Returns all objects.

IsRule_MatchedLengths Object Type Check

Description
Returns Matched Lengths Rule objects.

**Syntax**

```
IsRule_MatchedLengths : Boolean
```

**Examples**

```
IsRule_MatchedLengths
IsRule_MatchedLengths = True
```

Returns all Matched Lengths Rule objects.

```
Not IsRule_MatchedLengths
IsRule_MatchedLengths = False
```

Returns all objects except Matched Lengths Rule objects.

---

**IsRule_MaximumViaCount Object Type Check**

**Description**

Returns Maximum Via Count Rule objects.

**Syntax**

```
IsRule_MaximumViaCount : Boolean
```

**Examples**

```
IsRule_MaximumViaCount
IsRule_MaximumViaCount = True
```

Returns all Maximum Via Count Rule objects.

```
Not IsRule_MaximumViaCount
IsRule_MaximumViaCount = False
```

Returns all objects except Maximum Via Count Rule objects.

---

**IsRule_MaxMinHeightConstraint Object Type Check**

**Description**

Returns Max Min Height Rule objects.

**Syntax**

```
IsRule_MaxMinHeightConstraint : Boolean
```

**Examples**

```
IsRule_MaxMinHeightConstraint
IsRule_MaxMinHeightConstraint = True
```

Returns all Max Min Height Rule objects.

```
Not IsRule_MaxMinHeightConstraint
IsRule_MaxMinHeightConstraint = False
```

Returns all objects except Max Min Height Rule objects.

---

**IsRule_MaxMinHoleSize Object Type Check**

**Description**

Returns Max Min Hole Size Rule objects.

**Syntax**

```
IsRule_MaxMinHoleSize : Boolean
```

**Examples**

```
IsRule_MaxMinHoleSize
```
IsRule_MaxMinHoleSize = True
Returns all Max Min Hole Size Rule objects.
Not IsRule_MaxMinHoleSize
IsRule_MaxMinHoleSize = False
Returns all objects except Max Min Hole Size Rule objects.

IsRule_MaxMinImpedance Object Type Check

Description
Returns Max Min Impedance Rule objects.
Syntax
IsRule_MaxMinImpedance : Boolean
Examples
IsRule_MaxMinImpedance
IsRule_MaxMinImpedance = True
Returns all Max Min Impedance Rule objects.
Not IsRule_MaxMinImpedance
IsRule_MaxMinImpedance = False
Returns all objects except Max Min Impedance Rule objects.

IsRule_MaxMinLength Object Type Check

Description
Returns Max Min Length Rule objects.
Syntax
IsRule_MaxMinLength : Boolean
Examples
IsRule_MaxMinLength
IsRule_MaxMinLength = True
Returns all Max Min Length Rule objects.
Not IsRule_MaxMinLength
IsRule_MaxMinLength = False
Returns all objects except Max Min Length Rule objects.

IsRule_MaxMinWidth Object Type Check

Description
Returns Max Min Width Rule objects.
Syntax
IsRule_MaxMinWidth : Boolean
Examples
IsRule_MaxMinWidth
IsRule_MaxMinWidth = True
Returns all Max Min Width Rule objects.
Not IsRule_MaxMinWidth
IsRule_MaxMinWidth = False
Returns all objects except Max Min Width Rule objects.
IsRule_MaxSlope_FallingEdge Object Type Check

Description
Returns Max Slope Falling Edge Rule objects.

Syntax
IsRule_MaxSlope_FallingEdge : Boolean

Examples
IsRule_MaxSlope_FallingEdge
IsRule_MaxSlope_FallingEdge = True
Returns all Max Slope Falling Edge Rule objects.
Not IsRule_MaxSlope_FallingEdge
IsRule_MaxSlope_FallingEdge = False
Returns all objects except Max Slope Falling Edge Rule objects.

IsRule_MaxSlope_RisingEdge Object Type Check

Description
Returns Max Slope Rising Edge Rule objects.

Syntax
IsRule_MaxSlope_RisingEdge : Boolean

Examples
IsRule_MaxSlope_RisingEdge
IsRule_MaxSlope_RisingEdge = True
Returns all Max Slope Rising Edge Rule objects.
Not IsRule_MaxSlope_RisingEdge
IsRule_MaxSlope_RisingEdge = False
Returns all objects except Max Slope Rising Edge Rule objects.

IsRule_MinimumAnnularRing Object Type Check

Description
Returns Minimum Annular Ring Rule objects.

Syntax
IsRule_MinimumAnnularRing : Boolean

Examples
IsRule_MinimumAnnularRing
IsRule_MinimumAnnularRing = True
Returns all Minimum Annular Ring Rule objects.
Not IsRule_MinimumAnnularRing
IsRule_MinimumAnnularRing = False
Returns all objects except Minimum Annular Ring Rule objects.

IsRule_NetsToIgnore Object Type Check

Description
Returns Nets To Ignore Rule objects.

Syntax
IsRule_NetsToIgnore : Boolean
Examples
IsRule_NetsToIgnore
IsRule_NetsToIgnore = True
Returns all Nets To Ignore Rule objects.
Not IsRule_NetsToIgnore
IsRule_NetsToIgnore = False
Returns all objects except Nets To Ignore Rule objects.

IsRule_Overshoot_FallingEdge Object Type Check

Description
Returns Overshoot Falling Edge Rule objects.
Syntax
IsRule_Overshoot_FallingEdge : Boolean
Examples
IsRule_Overshoot_FallingEdge
IsRule_Overshoot_FallingEdge = True
Returns all Overshoot Falling Edge Rule objects.
Not IsRule_Overshoot_FallingEdge
IsRule_Overshoot_FallingEdge = False
Returns all objects except Overshoot Falling Edge Rule objects.

IsRule_Overshoot_RisingEdge Object Type Check

Description
Returns Overshoot Rising Edge Rule objects.
Syntax
IsRule_Overshoot_RisingEdge : Boolean
Examples
IsRule_Overshoot_RisingEdge
IsRule_Overshoot_RisingEdge = True
Returns all Overshoot Rising Edge Rule objects.
Not IsRule_Overshoot_RisingEdge
IsRule_Overshoot_RisingEdge = False
Returns all objects except Overshoot Rising Edge Rule objects.

IsRule_ParallelSegment Object Type Check

Description
Returns Parallel Segment Rule objects.
Syntax
IsRule_ParallelSegment : Boolean
Examples
IsRule_ParallelSegment
IsRule_ParallelSegment = True
Returns all Parallel Segment Rule objects.
Not IsRule_ParallelSegment
Returns all Parallel Segment Rule objects.
IsRule_ParallelSegment = False
Returns all objects except Parallel Segment Rule objects.

**IsRule_PasteMaskExpansion Object Type Check**

**Description**
Returns Paste Mask Expansion Rule objects.

**Syntax**
IsRule_PasteMaskExpansion : Boolean

**Examples**
IsRule_PasteMaskExpansion
IsRule_PasteMaskExpansion = True
Returns all Paste Mask Expansion Rule objects.
Not IsRule_PasteMaskExpansion
IsRule_PasteMaskExpansion = False
Returns all objects except Paste Mask Expansion Rule objects.

**IsRule_PermittedLayers Object Type Check**

**Description**
Returns Permitted Layers Rule objects.

**Syntax**
IsRule_PermittedLayers : Boolean

**Examples**
IsRule_PermittedLayers
IsRule_PermittedLayers = True
Returns all Permitted Layers Rule objects.
Not IsRule_PermittedLayers
IsRule_PermittedLayers = False
Returns all objects except Permitted Layers Rule objects.

**IsRule_PolygonConnectStyle Object Type Check**

**Description**
Returns Polygon Connect Style Rule objects.

**Syntax**
IsRule_PolygonConnectStyle : Boolean

**Examples**
IsRule_PolygonConnectStyle
IsRule_PolygonConnectStyle = True
Returns all Polygon Connect Style Rule objects.
Not IsRule_PolygonConnectStyle
IsRule_PolygonConnectStyle = False
Returns all objects except Polygon Connect Style Rule objects.

**IsRule_PowerPlaneClearance Object Type Check**

**Description**
Returns Power Plane Clearance Rule objects.

**Syntax**

IsRule_PowerPlaneClearance : Boolean

**Examples**

IsRule_PowerPlaneClearance

IsRule_PowerPlaneClearance = True

Returns all Power Plane Clearance Rule objects.

Not IsRule_PowerPlaneClearance

IsRule_PowerPlaneClearance = False

Returns all objects except Power Plane Clearance Rule objects.

**IsRule_PowerPlaneConnectStyle Object Type Check**

**Description**

Returns Power Plane Connect Style Rule objects.

**Syntax**

IsRule_PowerPlaneConnectStyle : Boolean

**Examples**

IsRule_PowerPlaneConnectStyle

IsRule_PowerPlaneConnectStyle = True

Returns all Power Plane Connect Style Rule objects.

Not IsRule_PowerPlaneConnectStyle

IsRule_PowerPlaneConnectStyle = False

Returns all objects except Power Plane Connect Style Rule objects.

**IsRule_RoutingCornerStyle Object Type Check**

**Description**

Returns Routing Corner Style Rule objects.

**Syntax**

IsRule_RoutingCornerStyle : Boolean

**Examples**

IsRule_RoutingCornerStyle

IsRule_RoutingCornerStyle = True

Returns all Routing Corner Style Rule objects.

Not IsRule_RoutingCornerStyle

IsRule_RoutingCornerStyle = False

Returns all objects except Routing Corner Style Rule objects.

**IsRule_RoutingLayers Object Type Check**

**Description**

Returns Routing Layers Rule objects.

**Syntax**

IsRule_RoutingLayers : Boolean

**Examples**

IsRule_RoutingLayers
IsRule_RoutingLayers = True
Returns all Routing Layers Rule objects.
Not IsRule_RoutingLayers
IsRule_RoutingLayers = False
Returns all objects except Routing Layers Rule objects.

IsRule_RoutingPriority Object Type Check

Description
Returns Routing Priority Rule objects.
Syntax
IsRule_RoutingPriority : Boolean
Examples
IsRule_RoutingPriority
IsRule_RoutingPriority = True
Returns all Routing Priority Rule objects.
Not IsRule_RoutingPriority
IsRule_RoutingPriority = False
Returns all objects except Routing Priority Rule objects.

IsRule_RoutingTopology Object Type Check

Description
Returns Routing Topology Rule objects.
Syntax
IsRule_RoutingTopology : Boolean
Examples
IsRule_RoutingTopology
IsRule_RoutingTopology = True
Returns all Routing Topology Rule objects.
Not IsRule_RoutingTopology
IsRule_RoutingTopology = False
Returns all objects except Routing Topology Rule objects.

IsRule_RoutingViaStyle Object Type Check

Description
Returns Routing Via Style Rule objects.
Syntax
IsRule_RoutingViaStyle : Boolean
Examples
IsRule_RoutingViaStyle
IsRule_RoutingViaStyle = True
Returns all Routing Via Style Rule objects.
Not IsRule_RoutingViaStyle
IsRule_RoutingViaStyle = False
Returns all objects except Routing Via Style Rule objects.
IsRule_ShortCircuit Object Type Check

Description
Returns Short Circuit Rule objects.

Syntax
IsRule_ShortCircuit : Boolean

Examples
IsRule_ShortCircuit
IsRule_ShortCircuit = True
Returns all Short Circuit Rule objects.
Not IsRule_ShortCircuit
IsRule_ShortCircuit = False
Returns all objects except Short Circuit Rule objects.

IsRule_SignalBaseValue Object Type Check

Description
Returns Signal Base Value Rule objects.

Syntax
IsRule_SignalBaseValue : Boolean

Examples
IsRule_SignalBaseValue
IsRule_SignalBaseValue = True
Returns all Signal Base Value Rule objects.
Not IsRule_SignalBaseValue
IsRule_SignalBaseValue = False
Returns all objects except Signal Base Value Rule objects.

IsRule_SignalStimulus Object Type Check

Description
Returns Signal Stimulus Rule objects.

Syntax
IsRule_SignalStimulus : Boolean

Examples
IsRule_SignalStimulus
IsRule_SignalStimulus = True
Returns all Signal Stimulus Rule objects.
Not IsRule_SignalStimulus
IsRule_SignalStimulus = False
Returns all objects except Signal Stimulus Rule objects.

IsRule_SignalTopValue Object Type Check

Description
Returns Signal Top Value Rule objects.

Syntax
IsRule_SignalTopValue : Boolean
Examples
IsRule_SignalTopValue
IsRule_SignalTopValue = True
Returns all Signal Top Value Rule objects.
Not IsRule_SignalTopValue
IsRule_SignalTopValue = False
Returns all objects except Signal Top Value Rule objects.

IsRule_SMDNeckDown Object Type Check

Description
Returns SMD Neck Down Rule objects.

Syntax
IsRule_SMDNeckDown : Boolean

Examples
IsRule_SMDNeckDown
IsRule_SMDNeckDown = True
Returns all SMD Neck Down Rule objects.
Not IsRule_SMDNeckDown
IsRule_SMDNeckDown = False
Returns all objects except SMD Neck Down Rule objects.

IsRule_SMDToCorner Object Type Check

Description
Returns SMD To Corner Rule objects.

Syntax
IsRule_SMDToCorner : Boolean

Examples
IsRule_SMDToCorner
IsRule_SMDToCorner = True
Returns all SMD To Corner Rule objects.
Not IsRule_SMDToCorner
IsRule_SMDToCorner = False
Returns all objects except SMD To Corner Rule objects.

IsRule_SMDToPlane Object Type Check

Description
Returns SMD To Plane Rule objects.

Syntax
IsRule_SMDToPlane : Boolean

Examples
IsRule_SMDToPlane
IsRule_SMDToPlane = True
Returns all SMD To Plane Rule objects.
Not IsRule_SMDToPlane
IsRule_SMDToPlane = False
Returns all objects except SMD To Plane Rule objects.

IsRule_SolderMaskExpansion Object Type Check

Description
Returns Solder Mask Expansion Rule objects.

Syntax
IsRule_SolderMaskExpansion : Boolean

Examples
IsRule_SolderMaskExpansion = True
Returns all Solder Mask Expansion Rule objects.
Not IsRule_SolderMaskExpansion
IsRule_SolderMaskExpansion = False
Returns all objects except Solder Mask Expansion Rule objects.

IsRule_SupplyNets Object Type Check

Description
Returns Supply Nets Rule objects.

Syntax
IsRule_SupplyNets : Boolean

Examples
IsRule_SupplyNets
IsRule_SupplyNets = True
Returns all Supply Nets Rule objects.
Not IsRule_SupplyNets
IsRule_SupplyNets = False
Returns all objects except Supply Nets Rule objects.

IsRule_TestPointStyle Object Type Check

Description
Returns Test Point Style Rule objects.

Syntax
IsRule_TestPointStyle : Boolean

Examples
IsRule_TestPointStyle
IsRule_TestPointStyle = True
Returns all Test Point Style Rule objects.
Not IsRule_TestPointStyle
IsRule_TestPointStyle = False
Returns all objects except Test Point Style Rule objects.

IsRule_TestPointUsage Object Type Check

Description
Returns Test Point Usage Rule objects.

Syntax
IsRule_TestPointUsage : Boolean

Examples
IsRule_TestPointUsage
IsRule_TestPointUsage = True
Returns all Test Point Usage Rule objects.
Not IsRule_TestPointUsage
IsRule_TestPointUsage = False
Returns all objects except Test Point Usage Rule objects.

IsRule_UnconnectedPin Object Type Check

Description
Returns Unconnected Pin Rule objects.

Syntax
IsRule_UnconnectedPin : Boolean

Examples
IsRule_UnconnectedPin
IsRule_UnconnectedPin = True
Returns all Unconnected Pin Rule objects.
Not IsRule_UnconnectedPin
IsRule_UnconnectedPin = False
Returns all objects except Unconnected Pin Rule objects.

IsRule_Undershoot_FallingEdge Object Type Check

Description
Returns Undershoot Falling Edge Rule objects.

Syntax
IsRule_Undershoot_FallingEdge : Boolean

Examples
IsRule_Undershoot_FallingEdge
IsRule_Undershoot_FallingEdge = True
Returns all Undershoot Falling Edge Rule objects.
Not IsRule_Undershoot_FallingEdge
IsRule_Undershoot_FallingEdge = False
Returns all objects except Undershoot Falling Edge Rule objects.

IsRule_Undershoot_RisingEdge Object Type Check

Description
Returns Undershoot Rising Edge Rule objects.

Syntax
IsRule_Undershoot_RisingEdge : Boolean

Examples
IsRule_Undershoot_RisingEdge
IsRule_Undershoot_RisingEdge = True
Returns all Undershoot Rising Edge Rule objects.
Not IsRule_Undershoot_RisingEdge
IsRule_Undershoot_RisingEdge = False
Returns all objects except Undershoot Rising Edge Rule objects.

IsRule_ViasUnderSMD Object Type Check

Description
Returns Vias Under SMD Rule objects.

Syntax
IsRule_ViasUnderSMD : Boolean

Examples
IsRule_ViasUnderSMD
IsRule_ViasUnderSMD = True
Returns all Vias Under SMD Rule objects.
Not IsRule_ViasUnderSMD
IsRule_ViasUnderSMD = False
Returns all objects except Vias Under SMD Rule objects.

IsSelected Attribute Check (PCB)

Description
Returns objects that are currently in a selected state.
This permits the user to control specifically which objects are returned, as the user is always able to control (from the List Workspace Panel) which objects are currently selected.

Syntax
IsSelected : Boolean

Examples
IsSelected
IsSelected = True
Returns all objects that are currently in a selected state.
Not IsSelected
IsSelected = False
Returns all objects except those that are currently in a selected state.

IsSIP Attribute Check

Description
Returns components that are of a SIP nature. Such a component contains at least four pads, and all of its pads are located in either the same row or the same column. Additionally, at least one (and typically all) of these pads are of a through-hole nature.
Note: The child objects of these component objects are not also returned.

Syntax
IsSIP : Boolean

Examples
IsSIP
IsSIP = True
Returns all Component objects that are of a SIP nature (as defined above).
Not IsSIP
IsSIP = False
Returns all objects except Component objects that are of a SIP nature (as defined above).

See Also
IsBGA
IsDIP
IsEdgeConnector
IsLCC
IsPGA
IsSMSIP
IsSMTComponent
IsSOIC
IsThruComponent

IsSMSIP Attribute Check

Description
Returns components that are of a Surface Mount SIP nature. Such a component contains at least four pads, and all of its pads
are located in either the same row or the same column. Additionally, all of these pads are of a surface mount nature.
Note: The child objects of these component objects are not also returned.

Syntax
IsSMSIP : Boolean

Examples
IsSMSIP
IsSMSIP = True
Returns all Component objects that are of a Surface Mount SIP nature (as defined above).
Not IsSMSIP
IsSMSIP = False
Returns all objects except Component objects that are of a Surface Mount SIP nature (as defined above).

See Also
IsBGA
IsDIP
IsEdgeConnector
IsLCC
IsPGA
IsSIP
IsSMTComponent
IsSOIC
IsThruComponent

IsSMTComponent Attribute Check

Description
Returns components that are of a SMT nature. All of the pads of such a component are of a surface mount nature.
Note: The child objects of these component objects are not also returned.

Syntax
IsSMTComponent : Boolean
Examples
IsSMTComponent
IsSMTComponent = True
Returns all Component objects that are of a SMT nature (as defined above).
Not IsSMTComponent
IsSMTComponent = False
Returns all objects except Component objects that are of a SMT nature (as defined above).

See Also
IsBGA
IsDIP
IsEdgeConnector
IsLCC
IsPGA
IsSiP
IsSMSIP
IsSOIC
IsThruComponent

IsSMTPin Attribute Check

Description
Returns all pads that are not on the Multi-Layer layer.

Syntax
IsSMTPin : Boolean

Examples
IsSMTPin
IsSMTPin = True
Returns all pads that are not on the Multi-Layer layer.
Not IsSMTPin
IsSMTPin = False
Returns all objects except pads that are not on the Multi-Layer layer.

See Also
IsThruPin

IsSOIC Attribute Check

Description
Returns components that are of a Surface Mount DIP (SOIC) nature. All of the pads of such a component are of a surface mount nature, and the total pad count is equal to the product of the number of rows of pads and the number of columns of pads. Such a component also has either exactly two rows of pads and at least three columns of pads, or else has exactly two columns of pads and at least two rows of pads.

Note: The child objects of these component objects are not also returned.

Syntax
IsSOIC : Boolean

Examples
IsSOIC
IsSOIC = True
Returns all Component objects that are of a Surface Mount DIP (SOIC) nature (as defined above).

Not IsSOIC
IsSOIC = False

Returns all objects except Component objects that are of a Surface Mount DIP (SOIC) nature (as defined above).

See Also
IsBGA
IsDIP
IsEdgeConnector
IsLCC
IsPGA
IsSiP
IsSMSIP
IsSMTComponent
IsThruComponent

**IsSquarePad Attribute Check**

**Description**
Returns pads on the Top, Bottom, and Keep-Out layers having a 'Rectangle' Shape property, and identical X-Size and Y-Size properties, and pads on the Multi-Layer layer having a 'Rectangle' Shape property, and identical X-Size and Y-Size properties, on at least one external layer.

**Syntax**
IsSquarePad : Boolean

**Examples**
IsSquarePad
IsSquarePad = True

Returns pads on the Top, Bottom, and Keep-Out layers having a 'Rectangle' Shape property, and identical X-Size and Y-Size properties, and pads on the Multi-Layer layer having a 'Rectangle' Shape property, and identical X-Size and Y-Size properties, on at least one external layer.

Not IsSquarePad
IsSquarePad = False

Returns all objects except pads on the Top, Bottom, and Keep-Out layers having a 'Rectangle' Shape property, and identical X-Size and Y-Size properties, and pads on the Multi-Layer layer having a 'Rectangle' Shape property, and identical X-Size and Y-Size properties, on at least one external layer.

See Also
IsRectangularPad
IsRoundPad

**IsStandardComponent Attribute Check**

**Description**
Returns Component objects whose Component Type property is 'Standard'; such components are listed in Bill of Materials files produced from the PCB file, and are also synchronised to corresponding components contained within the Schematic file(s) linked to the PCB file.

Note: The child objects of these component objects are not also returned.

**Syntax**
IsStandardComponent : Boolean

**Examples**
IsStandardComponent
IsStandardComponent = True
Returns Component objects that have a 'Standard' Component Type property.
Not IsStandardComponent
IsStandardComponent = False
Returns all objects except Component objects that have a 'Standard' Component Type property.

See Also
IsGraphicalComponent
IsMechanicalComponent
IsNetTieComponent
IsNoBOMComponent

IsSubnetJumper Attribute Check

Description
Returns subnet jumpers as specified by the query statement.

Syntax
IsSubnetJumper : Boolean

Examples
IsSubnetJumper
IsSubnetJumper = True
Returns subnet jumpers objects (usually track objects).
Not IsSubnetJumper
IsSubnetJumper = False
Returns all objects except subnet jumpers.

IsText Object Type Check

Description
Returns Text objects.

Syntax
IsText : Boolean

Examples
IsText
IsText = True
Returns all Text objects.
Not IsText
IsText = False
Returns all objects except Text objects.

IsTextInverted Field

Description
Returns objects having an InvertedTextBorder property that complies with the Query.

Syntax
InvertedTextBorder : Boolean

Examples
InvertedTextBorder
InvertedTextBorder = True
Returns all PCB components that have the Part Swapping setting set to true.
InvertedTextBorder = False
Returns all the PCB components that have the Part Swapping setting set to false.

**IsThruComponent Attribute Check**

**Description**
Returns components that are of a Thru-Hole nature. At least one (and typically all) of the pads of such a component are of a through-hole nature.

**Note:** The child objects of these component objects are not also returned.

**Syntax**
IsThruComponent : Boolean

**Examples**
IsThruComponent
IsThruComponent = True
Returns all Component objects that are of a Thru-Hole nature (as defined above).
Not IsThruComponent
IsThruComponent = False
Returns all objects except Component objects that are of a Thru-Hole nature (as defined above).

**See Also**
IsBGA
IsDIP
IsEdgeConnector
IsLCC
IsPGA
IsSIP
IsSMSIP
IsSMTComponent
IsSOIC

**IsThruPin Attribute Check**

**Description**
Returns all pads that are on the Multi-Layer layer.

**Syntax**
IsThruPin : Boolean

**Examples**
IsThruPin
IsThruPin = True
Returns all Pad objects that are on the Multi-Layer layer.
Not IsThruPin
IsThruPin = False
Returns all objects except Pad objects that are on the Multi-Layer layer.

**See Also**
IsSMTPin
IsTrack Object Type Check

**Description**
Returns Track objects.

**Syntax**
IsTrack : Boolean

**Examples**
IsTrack = True
Returns all Track objects.
Not IsTrack
IsTrack = False
Returns all objects except Track objects.

IsUserRouted Attribute Check

**Description**
Returns objects which were placed by the user, which excludes objects which were added to the file by Auto Route procedures.

**Syntax**
IsUserRouted : Boolean

**Examples**
IsUserRouted = True
Returns all objects which were placed by the user.
Not IsUserRouted
IsUserRouted = False
Returns all objects except objects which were placed by the user.

IsVertical Attribute Check

**Description**
Returns vertical tracks, i.e. those tracks whose X1 and X2 properties are identical.

**Syntax**
IsVertical : Boolean

**Examples**
IsVertical = True
Returns all Track objects that have identical X1 and X2 properties.
Not IsVertical
IsVertical = False
Returns all objects except Track objects that have identical X1 and X2 properties.

**See Also**
IsHorizontal
IsOblique
IsObliqueNegative
IsObliquePositive
IsVia Object Type Check

**Description**
Returns Via objects.

**Syntax**
IsVia : Boolean

**Examples**
IsVia
IsVia = True
Returns all Via objects.
Not IsVia
IsVia = False
Returns all objects except Via objects.

IsViolation Object Type Check

**Description**
Returns Violation objects.

**Syntax**
IsViolation : Boolean

**Examples**
IsViolation
IsViolation = True
Returns all Violation objects.
Not IsViolation
IsViolation = False
Returns all objects except Violation objects.

IsWire Attribute Check (PCB)

**Description**
Returns arcs and tracks on the Signal layers and Multi-Layer layer.

**Syntax**
IsWire : Boolean

**Examples**
IsWire
IsWire = True
Returns Arc and Track objects on the Signal layers and Multi-Layer layer.
Not IsWire
IsWire = False
Returns all objects except Arc and Track objects on the Signal layers and Multi-Layer layer.

Keepout Field

**Description**
Returns all Arc, Fill, and Track objects having a Keepout property that complies with the Query.
Note: The Keepout property is only defined for Arc, Fill, and Track objects.

**Syntax**
Keepout : Boolean

Examples
Keepout
Keepout = True
Returns all Arc, Fill, and Track objects that have a True Keepout property.

Not Keepout
Keepout = False
Returns all objects except Arc, Fill, and Track objects that have a True Keepout property. (Only Arc, Fill, and Track objects have a Keepout property, so all remaining types of objects do not have a True Keepout property, and are thus also returned by this Query.)

Kind Field

Description
Returns all objects having an Object Kind property that complies with the Query.
Note: This is an alias for ObjectKind (Field).

Syntax
Kind = ObjectKind_String
Kind <> ObjectKind_String

ObjectKind_String must be one of the strings from the following list:

Note: The single quote characters ('') shown at the start and end of each ObjectKind_String are both mandatory.

Examples
Kind = 'Class'
Returns all Class objects.
Kind <> 'Via'
Returns all objects except for Via objects.

L Field

Description
Returns all objects having a Layer property that complies with the Query.
Note: This is an alias for Layer (Field).

Syntax
L = Layer_String
L <> Layer_String

Layer_String must be one of the strings from the following list:
Layer Field

Description

Returns all objects having a Layer property that complies with the Query.

Note: This is an alias for L (Field).

Syntax

Layer = Layer_String

Layer <> Layer_String

Layer_String must be one of the strings from the following list:

'TopLayer', 'MidLayer1', 'MidLayer2', 'MidLayer3', 'MidLayer4', 'MidLayer5', 'MidLayer6', 'MidLayer7', 'MidLayer8', 'MidLayer9', 'MidLayer10', 'MidLayer11', 'MidLayer12', 'MidLayer13', 'MidLayer14', 'MidLayer15', 'MidLayer16', 'MidLayer17', 'MidLayer18', 'MidLayer19', 'MidLayer20', 'MidLayer21', 'MidLayer22', 'MidLayer23', 'MidLayer24', 'MidLayer25', 'MidLayer26', 'MidLayer27', 'MidLayer28', 'MidLayer29', 'MidLayer30', 'BottomLayer', 'TopOverlay', 'BottomOverlay', 'TopPaste', 'BottomPaste', 'TopSolder', 'BottomSolder', 'InternalPlane1', 'InternalPlane2', 'InternalPlane3', 'InternalPlane4', 'InternalPlane5', 'InternalPlane6', 'InternalPlane7', 'InternalPlane8', 'InternalPlane9', 'InternalPlane10', 'InternalPlane11', 'InternalPlane12', 'InternalPlane13', 'InternalPlane14', 'InternalPlane15', 'InternalPlane16', 'DrillGuide', 'KeepOutLayer', 'Mechanical1', 'Mechanical2', 'Mechanical3', 'Mechanical4', 'Mechanical5', 'Mechanical6', 'Mechanical7', 'Mechanical8', 'Mechanical9', 'Mechanical10', 'Mechanical11', 'Mechanical12', 'Mechanical13', 'Mechanical14', 'Mechanical15', 'Mechanical16', 'DrillDrawing', 'MultiLayer', '-'
**LeaderDimensionDot Field**

**Description**
Returns all Leader Dimension objects having a Dot property that complies with the Query.

Note: The child objects of these Leader Dimension objects are **not** also returned.

Note: The LeaderDimensionDot property is only defined for Leader Dimension objects.

**Syntax**
LeaderDimensionDot : Boolean_String

Boolean_String must be either 'True' or 'False'.

**Examples**
- **LeaderDimensionDot = 'True'**
  Returns all Leader Dimension objects that have a 'True' Dot property.
- **LeaderDimensionDot = 'False'**
  Returns all Leader Dimension objects that have a 'False' Dot property.
- **LeaderDimensionDot <> 'True'**
  Not (LeaderDimensionDot = 'True')
  Returns all objects except Leader Dimension objects that have a 'True' Dot property.
- **LeaderDimensionDot <> 'False'**
  Not (LeaderDimensionDot = 'False')
  Returns all objects except Leader Dimension objects that have a 'False' Dot property.

**LeaderDimensionDotSize Field**

**Description**
Returns all Leader Dimension objects having a Dot Size property that complies with the Query.

Note: The child objects of these Leader Dimension objects are **not** also returned.

Note: The LeaderDimensionDotSize property is only defined for Leader Dimension objects.

**Syntax**
LeaderDimensionDotSize : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
- **AsMils(LeaderDimensionDotSize) : Number** (This specifies a distance with a unit of 1 mil.)
- **AsMM(LeaderDimensionDotSize) : Number** (This specifies a distance with a unit of 1 mm.)

**Examples**
- **LeaderDimensionDotSize = 40**
  Returns all Leader Dimension objects that have a Dot Size property which is 40 current measurement units.
- **LeaderDimensionDotSize >= 1.143**
  Returns all Leader Dimension objects that have a Dot Size property which is greater than or equal to 1.143 current measurement units.
- **AsMils(LeaderDimensionDotSize) Between 30 And 50**
  Returns all Leader Dimension objects that have a Dot Size property which is greater than or equal to 30 mil and less than or equal to 50 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
- **AsMM(LeaderDimensionDotSize) < 0.889**
  Returns all Leader Dimension objects that have a Dot Size property which is less than 0.889 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
LeaderDimensionShape Field

Description
Returns all Leader Dimension objects having a Leader Shape property that complies with the Query.
Note: The child objects of these Leader Dimension objects are not also returned.
Note: The LeaderDimensionShape property is only defined for Leader Dimension objects.

Syntax
LeaderDimensionShape = LeaderShape_String
LeaderDimensionShape <> LeaderShape_String
LeaderShape_String must be one of the strings from the following list:
'None', 'Round', 'Square'
Note: The single quote characters (') shown at the start and end of each LeaderShape_String are both mandatory.

Examples
LeaderDimensionShape = 'Round'
Returns all Leader Dimension objects that have a 'Round' Leader Shape property.
LeaderDimensionShape <> 'Square'
Returns all objects except Leader Dimension objects that have a 'Square' Leader Shape property. (Only Leader Dimension objects have a Leader Shape property, so all remaining types of objects do not have a 'Square' LeaderDimensionShape property, and are thus also returned by this Query.)

Locked Field (PCB)

Description
Returns all objects having a Locked property that complies with the Query.

Syntax
Locked : Boolean

Examples
Locked
Locked = True
Returns all objects that have a True Locked property.
Not Locked
Locked = False
Returns all objects except those that have a True Locked property.

LockPrimitives Field

Description
Returns all Component, Coordinate, Dimension, and Polygon objects having a Lock Primitives property that complies with the Query.
Note: The child objects of these Component, Coordinate, Dimension, and Polygon objects are not also returned.
Note: The LockPrimitives property is only defined for Component, Coordinate, Dimension, and Polygon objects.

Syntax
LockPrimitives : Boolean_String
Boolean_String must be either 'True' or 'False'.

Examples
LockPrimitives = 'True'
Returns all Component, Coordinate, Dimension, and Polygon objects that have a 'True' Lock Primitives property.
LockPrimitives = 'False'
Returns all Component, Coordinate, Dimension, and Polygon objects that have a 'False' Lock Primitives property.
LockPrimitives <> 'True'
Not (LockPrimitives = 'True')

Returns all objects except Component, Coordinate, Dimension, and Polygon objects that have a 'True' Lock Primitives property.
LockPrimitives <> 'False'
Not (LockPrimitives = 'False')

Returns all objects except Component, Coordinate, Dimension, and Polygon objects that have a 'False' Lock Primitives property.

**ManHat Attribute Check**

**Description**
Returns tracks whose Manhattan Length complies with the Query.
The Manhattan Length of a track is determined by adding the absolute value of the difference between its X1 and X2 properties to the absolute value of the difference between its Y1 and Y2 properties. (The terminology is derived from Manhattan Island, in New York City, where the streets are laid out in a rectangular pattern.)
At present, all distances always use units of mils. (1 mil = 0.001 inch)

**Syntax**
ManHat : Distance_String

**Examples**
ManHat Between 500 And 2000
Returns tracks whose Manhattan Length is between 500 mil and 2000 mil.
ManHat >= 700
Returns tracks whose Manhattan Length is greater than or equal to 700 mil.

**Mirror Field**

**Description**
Returns all Text objects having a Mirror property that complies with the Query.
Note: The Mirror property is only defined for Text objects.

**Syntax**
Mirror : Boolean_String
Boolean_String must be either 'True' or 'False'.

**Examples**
Mirror = 'True'
Returns all Text objects that have a 'True' Mirror property.
Mirror = 'False'
Returns all Text objects that have a 'False' Mirror property.
Mirror <> 'True'
Not (Mirror = 'True')

Returns all objects except Text objects that have a 'True' Mirror property.
Mirror <> 'False'
Not (Mirror = 'False')

Returns all objects except Text objects that have a 'False' Mirror property.

**N Field**

**Description**
Returns Arc, Fill, Pad, Track, Via, From To, and Polygon objects whose Net property complies with the Query.

Note: The N property is only defined for Arc, Fill, Pad, Track, Via, From To, and Polygon objects.

Note: The associated Net object (whose Name property matches the Net specified, and which is a parent object of the returned objects,) is not also returned.

Note: This is an alias for Net (Field).

Syntax
N : String

Examples
N = 'GND'
Returns all Arc, Fill, Pad, Track, Via, From To, and Polygon objects that have a Net property of 'GND'.
N Like 'G*'
Returns all Arc, Fill, Pad, Track, Via, From To, and Polygon objects that have a Net property whose associated string commences with 'G'.

Name Field (PCB)

Description
Returns Class, Component, Design Rule, Net, Pad, and Violation objects whose Name property complies with the Query.

Note: The Name property is only defined for Class, Component, Design Rule, Net, Pad, and Violation objects.

Note: In the case of Pad objects, a full Name property has to be specified; this consists of an initial substring which specifies the Name property of the pad's parent component, followed by a '-' character, followed by another substring which specifies the Name property of the pad itself. Pads which are of a "free" nature are identified by an initial substring of 'Free-' (followed by the substring that specifies the pad's own Name property).

Syntax
Name : String

Examples
Name = 'All Components'
Returns all Class, Component, Design Rule, Net, Pad, and Violation objects that have a Name property of 'All Components'.
Name = 'U4-1'
Returns all Class, Component, Design Rule, Net, Pad, and Violation objects that have a Name property of 'U4-1'.
Name Like 'R7*'
Returns all Class, Component, Design Rule, Net, Pad, and Violation objects that have a Name property whose associated string commences with 'R7'.

NegativeNet Field

Description
Returns Arc, Fill, Pad, Track, Via, From To, and Polygon objects that are associated with differential pairs whose NegativeNet property complies with the Query.

Note: The Net property is only defined for Arc, Fill, Pad, Track, Via, From To, and Polygon objects.

Note: The associated Net object (whose Name property matches the Net specified, and which is a parent object of the returned objects,) is not also returned.

Syntax
NegativeNet : Boolean

Boolean must be either 'True' or 'False' (enclosed with single inverted commas).

Examples
NegativeNet = 'True'
Returns all objects that have a 'True' NegativeNet property.
NegativeNet = 'False'
Returns all objects that have a 'False' NegativeNet property.

NegativeNet <> 'True'
Not (NegativeNet = 'True')
Returns all objects except objects that have a 'True' NegativeNet property.

NegativeNet <> 'False'
Not (NegativeNet = 'False')
Returns all objects except objects that have a 'False' NegativeNet property.

Net Field

Description
Returns Arc, Fill, Pad, Track, Via, From To, and Polygon objects whose Net property complies with the Query.

Note: The Net property is only defined for Arc, Fill, Pad, Track, Via, From To, and Polygon objects.

Note: The associated Net object (whose Name property matches the Net specified, and which is a parent object of the returned objects, ) is not also returned.

Note: This is an alias for N (Field).

Syntax
Net : String

Examples
Net = 'GND'
Returns all Arc, Fill, Pad, Track, Via, From To, and Polygon objects that have a Net property of 'GND'.
Net Like 'G*'
Returns all Arc, Fill, Pad, Track, Via, From To, and Polygon objects that have a Net property whose associated string commences with 'G'.

NetLength Attribute Check

Description
Returns Net objects when the total length of arcs and tracks that have the same Net property for each of those complies with the Query; the associated child objects of the complying Net objects are also returned.

The length of each track is the (Pythagorean) distance between its end points, and is thus equal to SQRT(SQR(X1 - X2) + SQR(Y1 - Y2)), where X1 and Y1 are the coordinates of one end of the track, while X2 and Y2 are the coordinates of its other end.

The length of each arc is the product of its Radius and the difference between its End Angle and Start Angle, when that difference is measured in units of radians. (180 degrees = pi radians.)

At present, all distances always use mils unit. (1 mil = 0.001 inch)

Syntax
NetLength : Distance_String

Examples
NetLength Between 7000 And 15000
Returns Net objects (and their associated child objects) when the total length of the arcs and tracks associated with each Net is between 7000 mil and 15000 mil.
NetLength <= 12000
Returns Net objects (and their associated child objects) when the total length of the arcs and tracks associated with each Net is less than or equal to 12000 mil.
**NetNodes Field**

**Description**
Returns Arc, Fill, Pad, Track, Via, From To, and Polygon objects whose NetNodes property complies with the Query.

*Note:* The Net property is only defined for Arc, Fill, Pad, Track, Via, From To, and Polygon objects.

**Syntax**

\[
\text{NetNodes : Number}
\]

**Examples**

\[
\text{NetNodes > 2}
\]

Returns all Arc, Fill, Pad, Track, Via, From To, and Polygon objects that have at least two net nodes.

**NetPinCount Attribute Check**

**Description**
Returns Net objects when the total number of pads that have the same Net property for each of those complies with the Query; the associated child objects of the complying Net objects are also returned.

**Syntax**

\[
\text{NetPinCount : Number}
\]

**Examples**

\[
\text{NetPinCount Between 12 And 30}
\]

Returns Net objects (and their associated child objects) when the total number of the pads associated with each Net is between 12 and 30.

\[
\text{NetPinCount > 10}
\]

Returns Net objects (and their associated child objects) when the total number of the pads associated with each Net is greater than 10.

**NetViaCount Attribute Check**

**Description**
Returns Net objects when the total number of vias that have the same Net property for each of those complies with the Query; the associated child objects of the complying Net objects are also returned.

**Syntax**

\[
\text{NetViaCount : Number}
\]

**Examples**

\[
\text{NetViaCount Between 8 And 20}
\]

Returns Net objects (and their associated child objects) when the total number of the vias associated with each Net is between 8 and 20.

\[
\text{NetViaCount < 6}
\]

Returns Net objects (and their associated child objects) when the total number of the vias associated with each Net is less than 6.

**ObjectKind Field (PCB)**

**Description**
Returns all objects having an Object Kind property that complies with the Query.

*Note:* This is an alias for Kind (Field).

**Syntax**

\[
\text{ObjectKind = ObjectKind_String}
\]

\[
\text{ObjectKind <> ObjectKind_String}
\]
ObjectKind_String must be one of the strings from the following list:

'Acute Angle Rule', 'Angular Dimension', 'Arc', 'Baseline Dimension', 'Broken Nets Rule', 'Center Dimension', 'Class',
'Clearance Rule', 'Component', 'Component Clearance Rule', 'Component Rotations Rule', 'Confinement Constraint Rule',
'Coordinate', 'Daisy Chain Stub Length Rule', 'Datum Dimension', 'Fanout Control Rule', 'Fill', 'Flight Time Falling Edge Rule',
'Flight Time Rising Edge Rule', 'From To', 'Layer Pair Rule', 'Leader Dimension', 'Linear Diameter Dimension', 'Linear Dimension',
'Matched Lengths Rule', 'Max Min Height Rule', 'Max Min Hole Size Rule', 'Max Min Impedance Rule', 'Max Min Length Rule',
'Max Min Width Rule', 'Max Slope Falling Edge Rule', 'Max Slope Rising Edge Rule', 'Maximum Via Count Rule',
'Minimum Annular Ring Rule', 'Net', 'Nets To Ignore Rule', 'Original Dimension', 'Overshoot Falling Edge Rule',
'Overshoot Rising Edge Rule', 'Pad', 'Parallel Segment Rule', 'Paste Mask Expansion Rule', 'Permitted Layers Rule', 'Poly', 'Polygon Connect Style Rule',
'Power Plane Clearance Rule', 'Power Plane Connect Style Rule', 'Radial Diameter Dimension', 'Radial Dimension',
'Short Circuit Rule', 'Signal Base Value Rule', 'Signal Stimulus Rule', 'Signal Top Value Rule', 'SMD Neck Down Rule',
'SMD To Corner Rule', 'SMD To Plane Rule', 'Solder Mask Expansion Rule', 'Supply Nets Rule', 'Test Point Style Rule',
'Test Point Usage Rule', 'Text', 'Track', 'Unconnected Pin Rule', 'Undershoot Falling Edge Rule', 'Undershoot Rising Edge Rule',
'Via', 'Vias Under SMD Rule', 'Violation'

Note: The single quote characters (') shown at the start and end of each ObjectKind_String are both mandatory.

**Examples**

ObjectKind = 'Class'

Returns all Class objects.

ObjectKind <> 'Via'

Returns all objects except Via objects.

---

**OnBottom Layer Check**

**Description**

Returns all objects on the Bottom (Signal), Bottom Overlay, Bottom Paste Mask, and Bottom Solder Mask layers; the objects
returned include components placed on the bottom side of the PCB, as those objects have a Bottom (Signal) Layer property.

**Syntax**

OnBottom : Boolean

**Examples**

OnBottom

OnBottom = True

Returns all objects on the Bottom (Signal), Bottom Overlay, Bottom Paste Mask, and Bottom Solder Mask layers.

Not OnBottom

OnBottom = False

Returns all objects except objects on the Bottom (Signal), Bottom Overlay, Bottom Paste Mask, and Bottom Solder Mask layers.

---

**OnBottomLayer Layer Check**

**Description**

Returns all objects on the Bottom (Signal) layer; the objects returned include components placed on the bottom side of the PCB,
as those objects have a Bottom (Signal) Layer property.

**Syntax**

OnBottomLayer : Boolean

**Examples**

OnBottomLayer

OnBottomLayer = True

Returns all objects on the Bottom (Signal) layer.

Not OnBottomLayer
OnBottomLayer = False
Returns all objects except objects on the Bottom (Signal) layer.

**OnBottomPaste Layer Check**

**Description**
Returns all objects on the Bottom Paste Mask layer and all pads on the Bottom (Signal) layer; the latter objects are also returned because they also render to the Bottom Paste Mask layer.

**Syntax**
OnBottomPaste : Boolean

**Examples**
OnBottomPaste
OnBottomPaste = True
Returns all objects on the Bottom Paste Mask layer and all pads on the Bottom (Signal) layer.
Not OnBottomPaste
OnBottomPaste = False
Returns all objects except objects on the Bottom Paste Mask layer and pads on the Bottom (Signal) layer.

**OnBottomSilkscreen Layer Check**

**Description**
Returns all objects on the Bottom Overlay layer.

**Syntax**
OnBottomSilkscreen : Boolean

**Examples**
OnBottomSilkscreen
OnBottomSilkscreen = True
Returns all objects on the Bottom Overlay layer.
Not OnBottomSilkscreen
OnBottomSilkscreen = False
Returns all objects except objects on the Bottom Overlay layer.

**OnBottomSolderMask Layer Check**

**Description**
Returns all objects on the Bottom Solder Mask layer and all pads and vias on the Bottom (Signal) and Multi-Layer layers; the latter objects are also returned because they also render to the Bottom Solder Mask layer.

**Syntax**
OnBottomSolderMask : Boolean

**Examples**
OnBottomSolderMask
OnBottomSolderMask = True
Returns all objects on the Bottom Solder Mask layer and all pads and vias on the Bottom (Signal) and Multi-Layer layers.
Not OnBottomSolderMask
OnBottomSolderMask = False
Returns all objects except objects on the Bottom Solder Mask layer and pads and vias on the Bottom (Signal) and Multi-Layer layers.
**OnCopper Layer Check**

**Description**
Returns all objects on the Top (Signal), Bottom (Signal), and Multi-Layer layers, and all objects on any internal Signal and Internal Plane layers which are currently enabled. The objects returned include all components, as those objects either have a Top (Signal) or Bottom (Signal) Layer property.

**Syntax**
```plaintext```
OnCopper : Boolean
```

**Examples**
```plaintext```
OnCopper
OnCopper = True
Returns all objects on the Top (Signal), Bottom (Signal), and Multi-Layer layers, and on any internal Signal and Internal Plane layers which are currently enabled.
Not OnCopper
OnCopper = False
Returns all objects except objects on the Top (Signal), Bottom (Signal), and Multi-Layer layers, and on any internal Signal and Internal Plane layers which are currently enabled.
```

**OnGrid Membership Check**

**Description**
Returns all objects whose X1 property and Y1 property complies with the Query; that occurs when both of those coordinates are exactly integral multiples of the distances specified.

At present, all coordinates are referenced to the absolute origin (and not to the user origin or component insertion point), and all coordinates always use units of mils. (1 mil = 0.001 inch)

**Syntax**
```plaintext```
OnGrid(X1 : Distance_String , Y1 : Distance_String) : Boolean
```

Each instance of Distance_String is a numerical string that specifies a distance in the PCB (Library) file. To be returned, an object's X1 property has to be an integral multiple of the first distance specified, and its Y1 property has to be an integral multiple of the second distance specified.

**Examples**
```plaintext```
OnGrid(25,25)
OnGrid(25,25) = True
Returns all objects whose X1 property is an integral multiple of 25mil and whose Y1 property is an integral multiple of 25mil.
OnGrid(5,100)
OnGrid(5,100) = True
Returns all objects whose X1 property is an integral multiple of 5mil and whose Y1 property is an integral multiple of 100mil.
```

**OnInverted Layer Check**

**Description**
Returns all objects on the Top Paste Mask, Bottom Paste Mask, Top Solder Mask, and Bottom Solder Mask layers, and all objects on any Internal Plane layers which are currently enabled. These layers are all of an "Inverted" or "Reverse" nature, because the presence of any object on these layers results in a corresponding opening in a solder paste stencil, or in the solder mask film, or a corresponding absence of copper (in the cases of Paste Mask, Solder Mask, and Internal Plane layers respectively).

**Syntax**
```plaintext```
OnInverted : Boolean
```

**Examples**
```plaintext```
OnInverted
OnInverted = True
Returns all objects on the Top Paste Mask, Bottom Paste Mask, Top Solder Mask, and Bottom Solder Mask layers, and on any Internal Plane layers which are currently enabled.

Not OnInverted
OnInverted = False
Returns all objects except objects on the Top Paste Mask, Bottom Paste Mask, Top Solder Mask, and Bottom Solder Mask layers, and on any Internal Plane layers which are currently enabled.

OnLayer Membership Check

Description
Returns all objects having a Layer property that complies with the Query.

Syntax
OnLayer(Layer : Layer_String) : Boolean

Layer_String must be one of the strings from the following list:
'TopLayer', 'MidLayer1', 'MidLayer2', 'MidLayer3', 'MidLayer4', 'MidLayer5', 'MidLayer6', 'MidLayer7', 'MidLayer8', 'MidLayer9', 'MidLayer10', 'MidLayer11', 'MidLayer12', 'MidLayer13', 'MidLayer14', 'MidLayer15', 'MidLayer16', 'MidLayer17', 'MidLayer18', 'MidLayer19', 'MidLayer20', 'MidLayer21', 'MidLayer22', 'MidLayer23', 'MidLayer24', 'MidLayer25', 'MidLayer26', 'MidLayer27', 'MidLayer28', 'MidLayer29', 'MidLayer30', 'BottomLayer', 'TopOverlay', 'BottomOverlay', 'TopPaste', 'BottomPaste', 'TopSolder', 'BottomSolder', 'InternalPlane1', 'InternalPlane2', 'InternalPlane3', 'InternalPlane4', 'InternalPlane5', 'InternalPlane6', 'InternalPlane7', 'InternalPlane8', 'InternalPlane9', 'InternalPlane10', 'InternalPlane11', 'InternalPlane12', 'InternalPlane13', 'InternalPlane14', 'InternalPlane15', 'InternalPlane16', 'DrillGuide', 'KeepOutLayer', 'Mechanical1', 'Mechanical2', 'Mechanical3', 'Mechanical4', 'Mechanical5', 'Mechanical6', 'Mechanical7', 'Mechanical8', 'Mechanical9', 'Mechanical10', 'Mechanical11', 'Mechanical12', 'Mechanical13', 'Mechanical14', 'Mechanical15', 'Mechanical16', 'DrillDrawing', 'MultiLayer'

Note: The single quote characters ('') shown at the start and end of each Layer_String are both mandatory.

Examples
OnLayer('DrillGuide')
OnLayer('DrillGuide') = True
Returns all objects on the Drill Guide layer.

Not OnLayer('TopPaste') && Not OnLayer('BottomPaste')
OnLayer('TopPaste') = False && OnLayer('BottomPaste') = False
Returns all objects except for those on the Top Paste and Bottom Paste layers.

See Also
Layer Field

OnMechanical Layer Check

Description
Returns all objects on any Mechanical layers which are currently enabled.

Syntax
OnMechanical : Boolean

Examples
OnMechanical
OnMechanical = True
Returns all objects on any Mechanical layers which are currently enabled.

Not OnMechanical
OnMechanical = False
Returns all objects except objects on any Mechanical layers which are currently enabled.
OnMid Layer Check

Description
Returns all objects on any internal Signal layers which are currently enabled.

Syntax
OnMid : Boolean

Examples
OnMid
OnMid = True
Returns all objects on any internal Signal layers which are currently enabled.
Not OnMid
OnMid = False
Returns all objects except objects on any internal Signal layers which are currently enabled.

OnMultiLayer Layer Check

Description
Returns all objects on the Multi-Layer layer.

Syntax
OnMultiLayer : Boolean

Examples
OnMultiLayer
OnMultiLayer = True
Returns all objects on the Multi-Layer layer.
Not OnMultiLayer
OnMultiLayer = False
Returns all objects except objects on the Multi-Layer layer.

OnOutside Layer Check

Description
Returns all objects on the Top (Signal) and Bottom (Signal) layers. The objects returned include all components, as those objects either have a Top (Signal) or Bottom (Signal) Layer property.

Syntax
OnOutside : Boolean

Examples
OnOutside
OnOutside = True
Returns all objects on the Top (Signal) and Bottom (Signal) layers.
Not OnOutside
OnOutside = False
Returns all objects except objects on the Top (Signal) and Bottom (Signal) layers.

OnPaste Layer Check

Description
Returns all objects on the Top Paste Mask and Bottom Paste Mask layers and all pads on the Top (Signal) and Bottom (Signal) layers; the latter objects are also returned because they also render to the Paste Mask layers.
Syntax
OnPaste : Boolean

Examples
OnPaste
OnPaste = True
Returns all objects on the Top Paste Mask and Bottom Paste Mask layers and all pads on the Top (Signal) and Bottom (Signal) layers.
Not OnPaste
OnPaste = False
Returns all objects except objects on the Top Paste Mask and Bottom Paste Mask layers and pads on the Top (Signal) and Bottom (Signal) layers.

OnPlane Layer Check

Description
Returns all objects on any Internal Plane layers which are currently enabled.

Syntax
OnPlane : Boolean

Examples
OnPlane
OnPlane = True
Returns all objects on any Internal Plane layers which are currently enabled.
Not OnPlane
OnPlane = False
Returns all objects except objects on any Internal Plane layers which are currently enabled.

OnSignal Layer Check

Description
Returns all objects on the Top (Signal), Bottom (Signal), and Multi-Layer layers, and all objects on any internal Signal layers which are currently enabled. The objects returned include all components, as those objects either have a Top (Signal) or Bottom (Signal) Layer property

Syntax
OnSignal : Boolean

Examples
OnSignal
OnSignal = True
Returns all objects on the Top (Signal), Bottom (Signal), and Multi-Layer layers, and all objects on any internal Signal layers which are currently enabled.
Not OnSignal
OnSignal = False
Returns all objects except objects on the Top (Signal), Bottom (Signal), and Multi-Layer layers, and on any internal Signal layers which are currently enabled.

OnSilkscreen Layer Check

Description
Returns all objects on the Top Overlay and Bottom Overlay layers.

Syntax
OnSilkscreen : Boolean

**Examples**

OnSilkscreen
OnSilkscreen = True
Returns all objects on the Top Overlay and Bottom Overlay layers.

Not OnSilkscreen
OnSilkscreen = False
Returns all objects except objects on the Top Overlay and Bottom Overlay layers.

---

**OnSolderMask Layer Check**

**Description**
Returns all objects on the Top Solder Mask and Bottom Solder Mask layers and all pads and vias on the Top (Signal), Bottom (Signal), and Multi-Layer layers; the latter objects are also returned because they also render to the Solder Mask layers.

**Syntax**
OnSolderMask : Boolean

**Examples**

OnSolderMask
OnSolderMask = True
Returns all objects on the Top Solder Mask and Bottom Solder Mask layers and all pads and vias on the Top (Signal), Bottom (Signal), and Multi-Layer layers.

Not OnSolderMask
OnSolderMask = False
Returns all objects except objects on the Top Solder Mask and Bottom Solder Mask layers and pads and vias on the Top (Signal), Bottom (Signal), and Multi-Layer layers.

---

**OnTop Layer Check**

**Description**
Returns all objects on the Top (Signal), Top Overlay, Top Paste Mask, and Top Solder Mask layers; the objects returned include components placed on the top side of the PCB, as those objects have a Top (Signal) Layer property.

**Syntax**
OnTop : Boolean

**Examples**

OnTop
OnTop = True
Returns all objects on the Top (Signal), Top Overlay, Top Paste Mask, and Top Solder Mask layers.

Not OnTop
OnTop = False
Returns all objects except objects on the Top (Signal), Top Overlay, Top Paste Mask, and Top Solder Mask layers.

---

**OnTopLayer Layer Check**

**Description**
Returns all objects on the Top (Signal) layer; the objects returned include components placed on the top side of the PCB, as those objects have a Top (Signal) Layer property.

**Syntax**
OnTopLayer : Boolean
Examples
OnTopLayer
OnTopLayer = True
Returns all objects on the Top (Signal) layer.
Not OnTopLayer
OnTopLayer = False
Returns all objects except objects on the Top (Signal) layer.

OnTopPaste Layer Check

Description
Returns all objects on the Top Paste Mask layer and all pads on the Top (Signal) layer; the latter objects are also returned because they also render to the Top Paste Mask layer.

Syntax
OnTopPaste : Boolean

Examples
OnTopPaste
OnTopPaste = True
Returns all objects on the Top Paste Mask layer and all pads on the Top (Signal) layer.
Not OnTopPaste
OnTopPaste = False
Returns all objects except objects on the Top Paste Mask layer and pads on the Top (Signal) layer.

OnTopSilkscreen Layer Check

Description
Returns all objects on the Top Overlay layer.

Syntax
OnTopSilkscreen : Boolean

Examples
OnTopSilkscreen
OnTopSilkscreen = True
Returns all objects on the Top Overlay layer.
Not OnTopSilkscreen
OnTopSilkscreen = False
Returns all objects except objects on the Top Overlay layer.

OnTopSolderMask Layer Check

Description
Returns all objects on the Top Solder Mask layer and all pads and vias on the Top (Signal) and Multi-Layer layers; the latter objects are also returned because they also render to the Top Solder Mask layer.

Syntax
OnTopSolderMask : Boolean

Examples
OnTopSolderMask
OnTopSolderMask = True
Returns all objects on the Top Solder Mask layer and all pads and vias on the Top (Signal) and Multi-Layer layers.
Not OnTopSolderMask
OnTopSolderMask = False
Returns all objects except objects on the Top Solder Mask layer and pads and vias on the Top (Signal) and Multi-Layer layers.

**OverallHeight Field**

**Description**
Returns components that have component bodies and whose overall height property that complies with the OverallHeight Query. An overall height is the distance from the board to the topside of the component body. The standoff height is the distance from the board to the underside of the component body.

**Syntax**
OverallHeight : Number
If the PCB has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the PCB has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(OverallHeight) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(OverallHeight) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**
OverallHeight = 100
Returns all components that have a OverallHeight property which is equal to 100 current measurement units.
OverallHeight >= 4.064
Returns all components that have a OverallHeight property which is greater than or equal to 4.064 current measurement units.
AsMils(OverallHeight) > 80
Returns all components that have a OverallHeight property which is greater than 80 mil.
AsMM(OverallHeight) Between 1.524 And 3.81
Returns all components that have a OverallHeight property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm.
OverallHeight <> 50
Returns all components that have a OverallHeight property which is not equal to 50 current measurement units.
AsMils(OverallHeight) <= 120
Returns all components that have a OverallHeight property which is less than or equal to 120 mil.
AsMM(OverallHeight) < 1.778
Returns all components that have a OverallHeight property which is less than 1.778 mm.
PadElectricalType Field

Description
Returns all Pad objects having an Electrical Type property that complies with the Query.

Note: The PadElectricalType property is only defined for Pad objects.

Syntax
PadElectricalType = ElectricalType_String
PadElectricalType <> ElectricalType_String

ElectricalType_String must be one of the strings from the following list:
'Load', 'Source', 'Terminator'

Note: The single quote characters ('') shown at the start and end of each ElectricalType_String are both mandatory.

Examples
PadElectricalType = 'Load'
Returns Pad objects that have a 'Load' Electrical Type property.
PadElectricalType <> 'Source'
Returns all objects except Pad objects that have a 'Source' Electrical Type property. (Only Pad objects have an Electrical Type property, so all remaining types of objects do not have a 'Source' PadElectricalType property, and are thus also returned by this Query.)

PadIsPlated Field

Description
Returns all Pad objects having a Plated property that complies with the Query.

Note: The PadIsPlated property is only defined for Pad objects.

Syntax
PadIsPlated : Boolean_String

Boolean_String must be either 'True' or 'False'.

Examples
PadIsPlated = 'True'
Returns all Pad objects that have a 'True' Plated property.
PadIsPlated = 'False'
Returns all Pad objects that have a 'False' Plated property.
PadIsPlated <> 'True'
Not (PadIsPlated = 'True')
Returns all objects except Pad objects that have a 'True' Plated property.
PadIsPlated <> 'False'
Not (PadIsPlated = 'False')
Returns all objects except Pad objects that have a 'False' Plated property.

PadShape_AllLayers Field

Description
Returns pad objects whose Pad Shape (All Layers) property complies with the Query.

When a pad has a 'Simple' Padstack Mode property, the Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.
However, this property is **not** defined for any pad which has a 'Top-Middle-Bottom' or 'Full Stack' Padstack Mode property instead. (Such pads can only be on the Multi-Layer layer.) It is possible for such pads to have different Shape properties on different (Signal) layers, so it is not meaningful to define such a property for those pads. (Even when such a pad does have the same Shape property on all of the (Signal) layers, this property is still not defined.)

Note: The PadShape_AllLayers property is only defined for Pad objects with a 'Simple' Padstack Mode property.

**Syntax**

```
PadShape_AllLayers = PadShape_String
PadShape_AllLayers <> PadShape_String
```

PadShape_String must be one of the strings from the following list:

'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters ('') shown at the start and end of each PadShape_String are both mandatory.

**Examples**

```
PadShape_AllLayers = 'Rectangular'
Returns all Pad objects that have a 'Rectangular' Pad Shape (All Layers) property.

PadShape_AllLayers <> 'Octagonal'
Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (All Layers) property. (Only Pad objects with a 'Simple' Padstack Mode property have a Pad Shape (All Layers) property, so all remaining types of objects do not have an 'Octagonal' PadShape_AllLayers property, and are thus also returned by this Query.)

PadShape_AllLayers = 'Round' || PadShape_AllLayers = 'Rectangular'
IsPad && PadShape_AllLayers <> 'Octagonal'
Returns all Pad objects that do not have an 'Octagonal' Pad Shape (All Layers) property.
```

**PadShape_BottomLayer Field**

**Description**

Returns pad objects whose Pad Shape (Bottom Layer) property complies with the Query. That property specifies a pad's Shape property on the Bottom Signal layer when the pad has a 'Top-Middle-Bottom' or 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Bottom Layer) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

Note: The PadShape_BottomLayer property is only defined for Pad objects.

**Syntax**

```
PadShape_BottomLayer = PadShape_String
PadShape_BottomLayer <> PadShape_String
```

PadShape_String must be one of the strings from the following list:

'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters ('') shown at the start and end of each PadShape_String are both mandatory.

**Examples**

```
PadShape_BottomLayer = 'Rectangular'
Returns all Pad objects that have a 'Rectangular' Pad Shape (Bottom Layer) property.

PadShape_BottomLayer <> 'Octagonal'
Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Bottom Layer) property. (Only Pad objects have a Pad Shape (Bottom Layer) property, so all remaining types of objects do not have an 'Octagonal' PadShape_BottomLayer property, and are thus also returned by this Query.)

PadShape_BottomLayer = 'Round' || PadShape_BottomLayer = 'Rectangular'
IsPad && PadShape_BottomLayer <> 'Octagonal'
```
Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Bottom Layer) property.

### PadShape_MidLayer1 Field

**Description**

Returns pad objects whose Pad Shape (Mid Layer 1) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 1 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 1) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 1) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadShape_MidLayer1 property is only defined for Pad objects.

**Syntax**

```plaintext
PadShape_MidLayer1 = PadShape_String
PadShape_MidLayer1 <> PadShape_String
```

PadShape_String must be one of the strings from the following list:

- 'Octagonal'
- 'Rectangular'
- 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

**Examples**

```plaintext
PadShape_MidLayer1 = 'Rectangular'

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 1) property.
```

```plaintext
PadShape_MidLayer1 <> 'Octagonal'

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 1) property. (Only Pad objects have a Pad Shape (Mid Layer 1) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer1 property, and are thus also returned by this Query.)
```

```plaintext
PadShape_MidLayer1 = 'Round' || PadShape_MidLayer1 = 'Rectangular'
IsPad && PadShape_MidLayer1 <> 'Octagonal'

Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 1) property.
```

### PadShape_MidLayer10 Field

**Description**

Returns pad objects whose Pad Shape (Mid Layer 10) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 10 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 10) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 10) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadShape_MidLayer10 property is only defined for Pad objects.

**Syntax**

PadShape_MidLayer10 = PadShape_String
PadShape_MidLayer10 <> PadShape_String

PadShape_String must be one of the strings from the following list:

'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

**Examples**

PadShape_MidLayer10 = 'Rectangular'

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 10) property.

PadShape_MidLayer10 <> 'Octagonal'

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 10) property. (Only Pad objects have a Pad Shape (Mid Layer 10) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer10 property, and are thus also returned by this Query.)

PadShape_MidLayer10 = 'Round' || PadShape_MidLayer10 = 'Rectangular'

IsPad && PadShape_MidLayer10 <> 'Octagonal'

Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 10) property.

---

**PadShape_MidLayer11 Field**

**Description**

Returns pad objects whose Pad Shape (Mid Layer 11) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 11 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 11) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 11) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadShape_MidLayer11 property is only defined for Pad objects.

**Syntax**

PadShape_MidLayer11 = PadShape_String
PadShape_MidLayer11 <> PadShape_String

PadShape_String must be one of the strings from the following list:

'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

**Examples**

PadShape_MidLayer11 = 'Rectangular'

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 11) property.

PadShape_MidLayer11 <> 'Octagonal'

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 11) property. (Only Pad objects have a Pad Shape (Mid Layer 11) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer11 property, and are thus also returned by this Query.)

PadShape_MidLayer11 = 'Round' || PadShape_MidLayer11 = 'Rectangular'

IsPad && PadShape_MidLayer11 <> 'Octagonal'
Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 11) property.

**PadShape_MidLayer12 Field**

**Description**

Returns pad objects whose Pad Shape (Mid Layer 12) property complies with the Query.

That property specifies a pad’s Shape property on the Mid-Layer 12 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 12) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad’s Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 12) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadShape_MidLayer12 property is only defined for Pad objects.

**Syntax**

PadShape_MidLayer12 = PadShape_String

PadShape_MidLayer12 <> PadShape_String

PadShape_String must be one of the strings from the following list: 'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

**Examples**

PadShape_MidLayer12 = 'Rectangular'

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 12) property.

PadShape_MidLayer12 <> 'Octagonal'

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 12) property. (Only Pad objects have a Pad Shape (Mid Layer 12) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer12 property, and are thus also returned by this Query.)

PadShape_MidLayer12 = 'Round' || PadShape_MidLayer12 = 'Rectangular'

IsPad && PadShape_MidLayer12 <> 'Octagonal'

Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 12) property.

**PadShape_MidLayer13 Field**

**Description**

Returns pad objects whose Pad Shape (Mid Layer 13) property complies with the Query.

That property specifies a pad’s Shape property on the Mid-Layer 13 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 13) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad’s Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 13) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadShape_MidLayer13 property is only defined for Pad objects.

**Syntax**

PadShape_MidLayer13 = PadShape_String

PadShape_MidLayer13 <> PadShape_String

PadShape_String must be one of the strings from the following list:

'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

**Examples**

PadShape_MidLayer13 = 'Rectangular'

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 13) property.

PadShape_MidLayer13 <> 'Octagonal'

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 13) property. (Only Pad objects have a Pad Shape (Mid Layer 13) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer13 property, and are thus also returned by this Query.)

PadShape_MidLayer13 = 'Round' || PadShape_MidLayer13 = 'Rectangular'

IsPad && PadShape_MidLayer13 <> 'Octagonal'

Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 13) property.

---

**PadShape_MidLayer14 Field**

**Description**

Returns pad objects whose Pad Shape (Mid Layer 14) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 14 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 14) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 14) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadShape_MidLayer14 property is only defined for Pad objects.

**Syntax**

PadShape_MidLayer14 = PadShape_String

PadShape_MidLayer14 <> PadShape_String

PadShape_String must be one of the strings from the following list:

'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

**Examples**

PadShape_MidLayer14 = 'Rectangular'

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 14) property.

PadShape_MidLayer14 <> 'Octagonal'

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 14) property. (Only Pad objects have a Pad Shape (Mid Layer 14) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer14 property, and are thus also returned by this Query.)

PadShape_MidLayer14 = 'Round' || PadShape_MidLayer14 = 'Rectangular'

IsPad && PadShape_MidLayer14 <> 'Octagonal'
Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 14) property.

**PadShape_MidLayer15 Field**

**Description**

Returns pad objects whose Pad Shape (Mid Layer 15) property complies with the Query. That property specifies a pad's Shape property on the Mid-Layer 15 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 15) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 15) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

**Note:** The PadShape_MidLayer15 property is only defined for Pad objects.

**Syntax**

```
PadShape_MidLayer15 = PadShape_String
PadShape_MidLayer15 <> PadShape_String
```

PadShape_String must be one of the strings from the following list:

'Octagonal', 'Rectangular', 'Round'

**Note:** The single quote characters ('') shown at the start and end of each PadShape_String are both mandatory.

**Examples**

```
PadShape_MidLayer15 = 'Rectangular'
```

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 15) property.

```
PadShape_MidLayer15 <> 'Octagonal'
```

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 15) property. (Only Pad objects have a Pad Shape (Mid Layer 15) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer15 property, and are thus also returned by this Query.)

```
PadShape_MidLayer15 = 'Round' || PadShape_MidLayer15 = 'Rectangular'
IsPad && PadShape_MidLayer15 <> 'Octagonal'
```

Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 15) property.

**PadShape_MidLayer16 Field**

**Description**

Returns pad objects whose Pad Shape (Mid Layer 16) property complies with the Query. That property specifies a pad's Shape property on the Mid-Layer 16 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 16) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 16) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadShape_MidLayer16 property is only defined for Pad objects.

**Syntax**

PadShape_MidLayer16 = PadShape_String  
PadShape_MidLayer16 <> PadShape_String

PadShape_String must be one of the strings from the following list: 'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

**Examples**

PadShape_MidLayer16 = 'Rectangular'  
Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 16) property.

PadShape_MidLayer16 <> 'Octagonal'  
Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 16) property. (Only Pad objects have a Pad Shape (Mid Layer 16) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer16 property, and are thus also returned by this Query.)

PadShape_MidLayer16 = 'Round' || PadShape_MidLayer16 = 'Rectangular'  
IsPad && PadShape_MidLayer16 <> 'Octagonal'

Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 16) property.

**PadShape_MidLayer17 Field**

**Description**

Returns pad objects whose Pad Shape (Mid Layer 17) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 17 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 17) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 17) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadShape_MidLayer17 property is only defined for Pad objects.

**Syntax**

PadShape_MidLayer17 = PadShape_String  
PadShape_MidLayer17 <> PadShape_String

PadShape_String must be one of the strings from the following list: 'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

**Examples**

PadShape_MidLayer17 = 'Rectangular'  
Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 17) property.

PadShape_MidLayer17 <> 'Octagonal'  
Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 17) property. (Only Pad objects have a Pad Shape (Mid Layer 17) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer17 property, and are thus also returned by this Query.)

PadShape_MidLayer17 = 'Round' || PadShape_MidLayer17 = 'Rectangular'  
IsPad && PadShape_MidLayer17 <> 'Octagonal'
Returns all Pad objects that do **not** have an 'Octagonal' Pad Shape (Mid Layer 17) property.

**PadShape_MidLayer18 Field**

**Description**
Returns pad objects whose Pad Shape (Mid Layer 18) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 18 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 18) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 18) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

**Note:** The PadShape_MidLayer18 property is only defined for Pad objects.

**Syntax**

PadShape_MidLayer18 = PadShape_String

PadShape_MidLayer18 <> PadShape_String

PadShape_String must be one of the strings from the following list:

'Octagonal', 'Rectangular', 'Round'

**Note:** The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

**Examples**

PadShape_MidLayer18 = 'Rectangular'

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 18) property.

PadShape_MidLayer18 <> 'Octagonal'

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 18) property. (Only Pad objects have a Pad Shape (Mid Layer 18) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer18 property, and are thus also returned by this Query.)

PadShape_MidLayer18 = 'Round' || PadShape_MidLayer18 = 'Rectangular'

IsPad && PadShape_MidLayer18 <> 'Octagonal'

Returns all Pad objects that do **not** have an 'Octagonal' Pad Shape (Mid Layer 18) property.

**PadShape_MidLayer19 Field**

**Description**
Returns pad objects whose Pad Shape (Mid Layer 19) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 19 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 19) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 19) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadShape_MidLayer19 property is only defined for Pad objects.

Syntax

PadShape_MidLayer19 = PadShape_String
PadShape_MidLayer19 <> PadShape_String

PadShape_String must be one of the strings from the following list:
'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

Examples

PadShape_MidLayer19 = 'Rectangular'

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 19) property.

PadShape_MidLayer19 <> 'Octagonal'

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 19) property. (Only Pad objects have a Pad Shape (Mid Layer 19) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer19 property, and are thus also returned by this Query.)

PadShape_MidLayer19 = 'Round' || PadShape_MidLayer19 = 'Rectangular'
IsPad && PadShape_MidLayer19 <> 'Octagonal'

Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 19) property.

PadShape_MidLayer2 Field

Description

Returns pad objects whose Pad Shape (Mid Layer 2) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 2 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 2) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 2) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadShape_MidLayer2 property is only defined for Pad objects.

Syntax

PadShape_MidLayer2 = PadShape_String
PadShape_MidLayer2 <> PadShape_String

PadShape_String must be one of the strings from the following list:
'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

Examples

PadShape_MidLayer2 = 'Rectangular'

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 2) property.

PadShape_MidLayer2 <> 'Octagonal'

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 2) property. (Only Pad objects have a Pad Shape (Mid Layer 2) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer2 property, and are thus also returned by this Query.)

PadShape_MidLayer2 = 'Round' || PadShape_MidLayer2 = 'Rectangular'
IsPad && PadShape_MidLayer2 <> 'Octagonal'
Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 2) property.

**PadShape_MidLayer20 Field**

**Description**
Returns pad objects whose Pad Shape (Mid Layer 20) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 20 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 20) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 20) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadShape_MidLayer20 property is only defined for Pad objects.

**Syntax**

PadShape_MidLayer20 = PadShape_String
PadShape_MidLayer20 <> PadShape_String

PadShape_String must be one of the strings from the following list:
'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

**Examples**

PadShape_MidLayer20 = 'Rectangular'
Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 20) property.

PadShape_MidLayer20 <> 'Octagonal'
Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 20) property. (Only Pad objects have a Pad Shape (Mid Layer 20) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer20 property, and are thus also returned by this Query.)

PadShape_MidLayer20 = 'Round' || PadShape_MidLayer20 = 'Rectangular'
IsPad && PadShape_MidLayer20 <> 'Octagonal'

Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 20) property.

**PadShape_MidLayer21 Field**

**Description**
Returns pad objects whose Pad Shape (Mid Layer 21) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 21 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 21) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 21) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadShape_MidLayer21 property is only defined for Pad objects.

Syntax

PadShape_MidLayer21 = PadShape_String
PadShape_MidLayer21 <> PadShape_String

PadShape_String must be one of the strings from the following list:
'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters ('') shown at the start and end of each PadShape_String are both mandatory.

Examples

PadShape_MidLayer21 = 'Rectangular'
Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 21) property.

PadShape_MidLayer21 <> 'Octagonal'
Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 21) property. (Only Pad objects have a Pad Shape (Mid Layer 21) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer21 property, and are thus also returned by this Query.)

PadShape_MidLayer21 = 'Round' || PadShape_MidLayer21 = 'Rectangular'
IsPad && PadShape_MidLayer21 <> 'Octagonal'
Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 21) property.

PadShape_MidLayer22 Field

Description

Returns pad objects whose Pad Shape (Mid Layer 22) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 22 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 22) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 22) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadShape_MidLayer22 property is only defined for Pad objects.

Syntax

PadShape_MidLayer22 = PadShape_String
PadShape_MidLayer22 <> PadShape_String

PadShape_String must be one of the strings from the following list:
'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters ('') shown at the start and end of each PadShape_String are both mandatory.

Examples

PadShape_MidLayer22 = 'Rectangular'
Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 22) property.

PadShape_MidLayer22 <> 'Octagonal'
Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 22) property. (Only Pad objects have a Pad Shape (Mid Layer 22) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer22 property, and are thus also returned by this Query.)

PadShape_MidLayer22 = 'Round' || PadShape_MidLayer22 = 'Rectangular'
IsPad && PadShape_MidLayer22 <> 'Octagonal'
Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 22) property.

**PadShape_MidLayer23 Field**

**Description**
Returns pad objects whose Pad Shape (Mid Layer 23) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 23 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 23) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 23) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadShape_MidLayer23 property is only defined for Pad objects.

**Syntax**

PadShape_MidLayer23 = PadShape_String
PadShape_MidLayer23 <> PadShape_String

PadShape_String must be one of the strings from the following list:

'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

**Examples**

PadShape_MidLayer23 = 'Rectangular'

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 23) property.

PadShape_MidLayer23 <> 'Octagonal'

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 23) property. (Only Pad objects have a Pad Shape (Mid Layer 23) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer23 property, and are thus also returned by this Query.)

PadShape_MidLayer23 = 'Round' || PadShape_MidLayer23 = 'Rectangular'

IsPad && PadShape_MidLayer23 <> 'Octagonal'

Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 23) property.

**PadShape_MidLayer24 Field**

**Description**
Returns pad objects whose Pad Shape (Mid Layer 24) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 24 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 24) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 24) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadShape_MidLayer24 property is only defined for Pad objects.

Syntax

PadShape_MidLayer24 = PadShape_String

PadShape_MidLayer24 <> PadShape_String

PadShape_String must be one of the strings from the following list:
"Octagonal", "Rectangular", "Round"

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

Examples

PadShape_MidLayer24 = 'Rectangular'

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 24) property.

PadShape_MidLayer24 <> 'Octagonal'

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 24) property. (Only Pad objects have a Pad Shape (Mid Layer 24) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer24 property, and are thus also returned by this Query.)

PadShape_MidLayer24 = 'Round' || PadShape_MidLayer24 = 'Rectangular'

IsPad && PadShape_MidLayer24 <> 'Octagonal'

Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 24) property.

PadShape_MidLayer25 Field

Description

Returns pad objects whose Pad Shape (Mid Layer 25) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 25 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 25) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 25) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadShape_MidLayer25 property is only defined for Pad objects.

Syntax

PadShape_MidLayer25 = PadShape_String

PadShape_MidLayer25 <> PadShape_String

PadShape_String must be one of the strings from the following list:
"Octagonal", "Rectangular", "Round"

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

Examples

PadShape_MidLayer25 = 'Rectangular'

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 25) property.

PadShape_MidLayer25 <> 'Octagonal'

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 25) property. (Only Pad objects have a Pad Shape (Mid Layer 25) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer25 property, and are thus also returned by this Query.)

PadShape_MidLayer25 = 'Round' || PadShape_MidLayer25 = 'Rectangular'

IsPad && PadShape_MidLayer25 <> 'Octagonal'
Returns all Pad objects that do **not** have an 'Octagonal' Pad Shape (Mid Layer 25) property.

**PadShape_MidLayer26 Field**

**Description**
Returns pad objects whose Pad Shape (Mid Layer 26) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 26 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 26) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 26) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadShape_MidLayer26 property is only defined for Pad objects.

**Syntax**

```
PadShape_MidLayer26 = PadShape_String
PadShape_MidLayer26 <> PadShape_String
```

PadShape_String must be one of the strings from the following list:

'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

**Examples**

```
PadShape_MidLayer26 = 'Rectangular'
```

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 26) property.

```
PadShape_MidLayer26 <> 'Octagonal'
```

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 26) property. (Only Pad objects have a Pad Shape (Mid Layer 26) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer26 property, and are thus also returned by this Query.)

```
PadShape_MidLayer26 = 'Round' || PadShape_MidLayer26 = 'Rectangular'
IsPad && PadShape_MidLayer26 <> 'Octagonal'
```

Returns all Pad objects that do **not** have an 'Octagonal' Pad Shape (Mid Layer 26) property.

**PadShape_MidLayer27 Field**

**Description**
Returns pad objects whose Pad Shape (Mid Layer 27) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 27 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 27) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 27) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadShape_MidLayer27 property is only defined for Pad objects.

Syntax
PadShape_MidLayer27 = PadShape_String
PadShape_MidLayer27 <> PadShape_String

PadShape_String must be one of the strings from the following list:
'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

Examples
PadShape_MidLayer27 = 'Rectangular'
Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 27) property.

PadShape_MidLayer27 <> 'Octagonal'
Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 27) property. (Only Pad objects have a Pad Shape (Mid Layer 27) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer27 property, and are thus also returned by this Query.)

PadShape_MidLayer27 = 'Round' || PadShape_MidLayer27 = 'Rectangular'
IsPad && PadShape_MidLayer27 <> 'Octagonal'
Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 27) property.

PadShape_MidLayer28 Field

Description
Returns pad objects whose Pad Shape (Mid Layer 28) property complies with the Query.
That property specifies a pad's Shape property on the Mid-Layer 28 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 28) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 28) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadShape_MidLayer28 property is only defined for Pad objects.

Syntax
PadShape_MidLayer28 = PadShape_String
PadShape_MidLayer28 <> PadShape_String

PadShape_String must be one of the strings from the following list:
'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

Examples
PadShape_MidLayer28 = 'Rectangular'
Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 28) property.

PadShape_MidLayer28 <> 'Octagonal'
Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 28) property. (Only Pad objects have a Pad Shape (Mid Layer 28) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer28 property, and are thus also returned by this Query.)

PadShape_MidLayer28 = 'Round' || PadShape_MidLayer28 = 'Rectangular'
IsPad && PadShape_MidLayer28 <> 'Octagonal'
Returns all Pad objects that do **not** have an 'Octagonal' Pad Shape (Mid Layer 28) property.

**PadShape_MidLayer29 Field**

**Description**
Returns pad objects whose Pad Shape (Mid Layer 29) property complies with the Query.
That property specifies a pad's Shape property on the Mid-Layer 29 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 29) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 29) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadShape_MidLayer29 property is only defined for Pad objects.

**Syntax**

PadShape_MidLayer29 = PadShape_String
PadShape_MidLayer29 <> PadShape_String

PadShape_String must be one of the strings from the following list:
'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters ('') shown at the start and end of each PadShape_String are both mandatory.

**Examples**

PadShape_MidLayer29 = 'Rectangular'
Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 29) property.

PadShape_MidLayer29 <> 'Octagonal'
Returns all objects **except** Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 29) property. (Only Pad objects have a Pad Shape (Mid Layer 29) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer29 property, and are thus also returned by this Query.)

PadShape_MidLayer29 = 'Round' || PadShape_MidLayer29 = 'Rectangular'
IsPad && PadShape_MidLayer29 <> 'Octagonal'
Returns all Pad objects that do **not** have an 'Octagonal' Pad Shape (Mid Layer 29) property.

**PadShape_MidLayer3 Field**

**Description**
Returns pad objects whose Pad Shape (Mid Layer 3) property complies with the Query.
That property specifies a pad's Shape property on the Mid-Layer 3 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 3) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 3) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadShape_MidLayer3 property is only defined for Pad objects.

Syntax

PadShape_MidLayer3 = PadShape_String
PadShape_MidLayer3 <> PadShape_String

PadShape_String must be one of the strings from the following list:
'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

Examples

PadShape_MidLayer3 = 'Rectangular'

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 3) property.

PadShape_MidLayer3 <> 'Octagonal'

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 3) property. (Only Pad objects have a Pad Shape (Mid Layer 3) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer3 property, and are thus also returned by this Query.)

PadShape_MidLayer3 = 'Round' || PadShape_MidLayer3 = 'Rectangular'

IsPad && PadShape_MidLayer3 <> 'Octagonal'

Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 3) property.

PadShape_MidLayer30 Field

Description

Returns pad objects whose Pad Shape (Mid Layer 30) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 30 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 30) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 30) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadShape_MidLayer30 property is only defined for Pad objects.

Syntax

PadShape_MidLayer30 = PadShape_String
PadShape_MidLayer30 <> PadShape_String

PadShape_String must be one of the strings from the following list:
'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

Examples

PadShape_MidLayer30 = 'Rectangular'

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 30) property.

PadShape_MidLayer30 <> 'Octagonal'

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 30) property. (Only Pad objects have a Pad Shape (Mid Layer 30) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer30 property, and are thus also returned by this Query.)

PadShape_MidLayer30 = 'Round' || PadShape_MidLayer30 = 'Rectangular'

IsPad && PadShape_MidLayer30 <> 'Octagonal'
Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 30) property.

**PadShape_MidLayer4 Field**

**Description**
Returns pad objects whose Pad Shape (Mid Layer 4) property complies with the Query.
That property specifies a pad's Shape property on the Mid-Layer 4 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 4) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 4) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadShape_MidLayer4 property is only defined for Pad objects.

**Syntax**

\[
\text{PadShape}\_\text{MidLayer4} = \text{PadShape}\_\text{String} \\
\text{PadShape}\_\text{MidLayer4} \not= \text{PadShape}\_\text{String}
\]

\text{PadShape}\_\text{String} must be one of the strings from the following list:
- 'Octagonal'
- 'Rectangular'
- 'Round'

Note: The single quote characters (') shown at the start and end of each \text{PadShape}\_\text{String} are both mandatory.

**Examples**

\[
\text{PadShape}\_\text{MidLayer4} = 'Rectangular'
\]

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 4) property.
\[
\text{PadShape}\_\text{MidLayer4} = 'Octagonal'
\]

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 4) property. (Only Pad objects have a Pad Shape (Mid Layer 4) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer4 property, and are thus also returned by this Query.)
\[
\text{PadShape}\_\text{MidLayer4} = 'Round' \text{ || } \text{PadShape}\_\text{MidLayer4} = 'Rectangular' \\
\text{IsPad} \&\& \text{PadShape}\_\text{MidLayer4} \not= 'Octagonal'
\]

Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 4) property.

**PadShape_MidLayer5 Field**

**Description**
Returns pad objects whose Pad Shape (Mid Layer 5) property complies with the Query.
That property specifies a pad's Shape property on the Mid-Layer 5 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 5) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 5) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadShape_MidLayer5 property is only defined for Pad objects.

**Syntax**

```plaintext
PadShape_MidLayer5 = PadShape_String
PadShape_MidLayer5 <> PadShape_String
```

PadShape_String must be one of the strings from the following list:
'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

**Examples**

```plaintext
PadShape_MidLayer5 = 'Rectangular'
```

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 5) property.

```plaintext
PadShape_MidLayer5 <> 'Octagonal'
```

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 5) property. (Only Pad objects have a Pad Shape (Mid Layer 5) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer5 property, and are thus also returned by this Query.)

```plaintext
PadShape_MidLayer5 = 'Round' || PadShape_MidLayer5 = 'Rectangular'
IsPad && PadShape_MidLayer5 <> 'Octagonal'
```

Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 5) property.

---

**PadShape_MidLayer6 Field**

**Description**

Returns pad objects whose Pad Shape (Mid Layer 6) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 6 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 6) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 6) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadShape_MidLayer6 property is only defined for Pad objects.

**Syntax**

```plaintext
PadShape_MidLayer6 = PadShape_String
PadShape_MidLayer6 <> PadShape_String
```

PadShape_String must be one of the strings from the following list:
'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

**Examples**

```plaintext
PadShape_MidLayer6 = 'Rectangular'
```

Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 6) property.

```plaintext
PadShape_MidLayer6 <> 'Octagonal'
```

Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 6) property. (Only Pad objects have a Pad Shape (Mid Layer 6) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer6 property, and are thus also returned by this Query.)

```plaintext
PadShape_MidLayer6 = 'Round' || PadShape_MidLayer6 = 'Rectangular'
IsPad && PadShape_MidLayer6 <> 'Octagonal'
```
Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 6) property.

PadShape_MidLayer7 Field

Description
Returns pad objects whose Pad Shape (Mid Layer 7) property complies with the Query.
That property specifies a pad's Shape property on the Mid-Layer 7 layer when the pad has a 'Full Stack' Padstack Mode
property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is
assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 7) property is assessed as being equal to its
Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in
the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any
of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 7) property is assessed as being
equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is
not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property;
pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadShape_MidLayer7 property is only defined for Pad objects.

Syntax
PadShape_MidLayer7 = PadShape_String
PadShape_MidLayer7 <> PadShape_String

PadShape_String must be one of the strings from the following list:
'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

Examples
PadShape_MidLayer7 = 'Rectangular'
Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 7) property.

PadShape_MidLayer7 <> 'Octagonal'
Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 7) property. (Only Pad objects have a
Pad Shape (Mid Layer 7) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer7 property,
and are thus also returned by this Query.)

PadShape_MidLayer7 = 'Round' || PadShape_MidLayer7 = 'Rectangular'
IsPad && PadShape_MidLayer7 <> 'Octagonal'

Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 7) property.
PadShape_MidLayer8 Field

Description
Returns pad objects whose Pad Shape (Mid Layer 8) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 8 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 8) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 8) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadShape_MidLayer8 property is only defined for Pad objects.

Syntax
PadShape_MidLayer8 = PadShape_String
PadShape_MidLayer8 <> PadShape_String

PadShape_String must be one of the strings from the following list:
'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

Examples
PadShape_MidLayer8 = 'Rectangular'
Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 8) property.

PadShape_MidLayer8 <> 'Octagonal'
Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 8) property. (Only Pad objects have a Pad Shape (Mid Layer 8) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer8 property, and are thus also returned by this Query.)

PadShape_MidLayer8 = 'Round' || PadShape_MidLayer8 = 'Rectangular' &&Pad && PadShape_MidLayer8 <> 'Octagonal'
Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 8) property.

PadShape_MidLayer9 Field

Description
Returns pad objects whose Pad Shape (Mid Layer 9) property complies with the Query.

That property specifies a pad's Shape property on the Mid-Layer 9 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Mid Layer 9) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Shape (Mid Layer 9) property is assessed as being equal to its Pad Shape (Mid Layer 1) property, which specifies its Shape property on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadShape_MidLayer9 property is only defined for Pad objects.
Syntax
PadShape_MidLayer9 = PadShape_String
PadShape_MidLayer9 <> PadShape_String

PadShape_String must be one of the strings from the following list:
'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

Examples
PadShape_MidLayer9 = 'Rectangular'
Returns all Pad objects that have a 'Rectangular' Pad Shape (Mid Layer 9) property.

PadShape_MidLayer9 <> 'Octagonal'
Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Mid Layer 9) property. (Only Pad objects have a Pad Shape (Mid Layer 9) property, so all remaining types of objects do not have an 'Octagonal' PadShape_MidLayer9 property, and are thus also returned by this Query.)

PadShape_MidLayer9 = 'Round' || PadShape_MidLayer9 = 'Rectangular'
IsPad && PadShape_MidLayer9 <> 'Octagonal'
Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Mid Layer 9) property.

PadShape_TopLayer Field

Description
Returns pad objects whose Pad Shape (Top Layer) property complies with the Query.

That property specifies a pad's Shape property on the Top Signal layer when the pad has a 'Top-Middle-Bottom' or 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Shape (Top Layer) property is assessed as being equal to its Pad Shape (All Layers) property. (The Pad Shape (All Layers) property specifies the pad's Shape property on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Shape property on the same layer as the pad in the case of pads on any of the other layers.)

Note: The PadShape_TopLayer property is only defined for Pad objects.

Syntax
PadShape_TopLayer = PadShape_String
PadShape_TopLayer <> PadShape_String

PadShape_String must be one of the strings from the following list:
'Octagonal', 'Rectangular', 'Round'

Note: The single quote characters (') shown at the start and end of each PadShape_String are both mandatory.

Examples
PadShape_TopLayer = 'Rectangular'
Returns all Pad objects that have a 'Rectangular' Pad Shape (Top Layer) property.

PadShape_TopLayer <> 'Octagonal'
Returns all objects except Pad objects that have an 'Octagonal' Pad Shape (Top Layer) property. (Only Pad objects have a Pad Shape (Top Layer) property, so all remaining types of objects do not have an 'Octagonal' PadShape_TopLayer property, and are thus also returned by this Query.)

PadShape_TopLayer = 'Round' || PadShape_TopLayer = 'Rectangular'
IsPad && PadShape_TopLayer <> 'Octagonal'
Returns all Pad objects that do not have an 'Octagonal' Pad Shape (Top Layer) property.
PadStackMode Field

Description
Returns all Pad objects having a Pad Stack Mode property that complies with the Query.
Note: The PadStackMode property is only defined for Pad objects.

Syntax
PadStackMode = PadStackMode_String
PadStackMode <> PadStackMode_String

PadStackMode_String must be one of the strings from the following list:
'Full Stack', 'Simple', 'Top-Middle-Bottom'

Note: The single quote characters (') shown at the start and end of each PadStackMode_String are both mandatory.

Examples
PadStackMode = 'Simple'
Returns all Pad objects that have a 'Simple' Pad Stack Mode property.

PadStackMode <> 'Full Stack'
Returns all objects except Pad objects that have a 'Full Stack' Pad Stack Mode property. (Only Pad objects have a Pad Stack Mode property, so all remaining types of objects do not have a 'Full Stack' PadStackMode property, and are thus also returned by this Query.)

PadStackMode = 'Simple' || PadStackMode = 'Top-Middle-Bottom'
IsPad && PadStackMode <> 'Full Stack'
Returns all Pad objects that do not have a 'Full Stack' Pad Stack Mode property.

PadStackSettings Field

Description
Returns all Pad objects having a Pad Stack Settings property that complies with the Query.
Note: The PadStackSettings property is only defined for Pad objects.

Syntax

Examples
PadXSize_AllLayers Field

Description
Returns pad objects whose Pad X Size (All Layers) property complies with the Query.

When a pad has a 'Simple' Padstack Mode property, the Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.

However, this property is not defined for any pad which has a 'Top-Middle-Bottom' or 'Full Stack' Padstack Mode property instead. (Such pads can only be on the Multi-Layer layer.) It is possible for such pads to have different X-Size distances on different (Signal) layers, so it is not meaningful to define such a property for those pads. (Even when such a pad does have the same X-Size distances on all of the (Signal) layers, this property is still not defined.)

Note: The PadXSize_AllLayers property is only defined for Pad objects with a 'Simple' Padstack Mode property.

Syntax
PadXSize_AllLayers : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_AllLayers) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_AllLayers) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_AllLayers = 100
Returns all Pad objects that have a Pad X Size (All Layers) property which is equal to 100 current measurement units.

PadXSize_AllLayers >= 4.064
Returns all Pad objects that have a Pad X Size (All Layers) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadXSize_AllLayers) > 80
Returns all Pad objects that have a Pad X Size (All Layers) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadXSize_AllLayers) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (All Layers) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadXSize_AllLayers <> 50
Returns all Pad objects that have a Pad X Size (All Layers) property which is not equal to 50 current measurement units.

AsMils(PadXSize_AllLayers) <= 120
Returns all Pad objects that have a Pad X Size (All Layers) property which is less than or equal to 120 mil.

AsMM(PadXSize_AllLayers) < 1.778
Returns all Pad objects that have a Pad X Size (All Layers) property which is less than 1.778 mm.

PadXSize_BottomLayer Field

Description
Returns pad objects whose Pad X Size (Bottom Layer) property complies with the Query.

That property specifies a pad's X-Size distance on the Bottom Signal layer when the pad has a 'Top-Middle-Bottom' or 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Bottom Layer) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

Note: The PadXSize_BottomLayer property is only defined for Pad objects.

**Syntax**

PadXSize_BottomLayer : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

- AsMils(PadXSize_BottomLayer) : Number {This specifies a distance with a unit of 1 mil.}
- AsMM(PadXSize_BottomLayer) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**

- PadXSize_BottomLayer = 100
  Returns all Pad objects that have a Pad X Size (Bottom Layer) property which is equal to 100 current measurement units.
- PadXSize_BottomLayer >= 4.064
  Returns all Pad objects that have a Pad X Size (Bottom Layer) property which is greater than or equal to 4.064 current measurement units.
- AsMils(PadXSize_BottomLayer) > 80
  Returns all Pad objects that have a Pad X Size (Bottom Layer) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
- AsMM(PadXSize_BottomLayer) Between 1.524 And 3.81
  Returns all Pad objects that have a Pad X Size (Bottom Layer) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
- PadXSize_BottomLayer <> 50
  Returns all Pad objects that have a Pad X Size (Bottom Layer) property which is not equal to 50 current measurement units.
- AsMils(PadXSize_BottomLayer) <= 120
  Returns all Pad objects that have a Pad X Size (Bottom Layer) property which is less than or equal to 120 mil.
- AsMM(PadXSize_BottomLayer) < 1.778
  Returns all Pad objects that have a Pad X Size (Bottom Layer) property which is less than 1.778 mm.
**PadXSize_MidLayer1 Field**

**Description**

Returns pad objects whose Pad X Size (Mid Layer 1) property complies with the Query. That property specifies a pad's X-Size distance on the MidLayer1 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 1) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.) For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 1) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer1 property is only defined for Pad objects.

**Syntax**

`PadXSize_MidLayer1 : Number`

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

- `AsMils(PadXSize_MidLayer1) : Number` {This specifies a distance with a unit of 1 mil.}
- `AsMM(PadXSize_MidLayer1) : Number` {This specifies a distance with a unit of 1 mm.}

**Examples**

- `PadXSize_MidLayer1 = 100`
  Returns all Pad objects that have a Pad X Size (Mid Layer 1) property which is equal to 100 current measurement units.

- `PadXSize_MidLayer1 >= 4.064`
  Returns all Pad objects that have a Pad X Size (Mid Layer 1) property which is greater than or equal to 4.064 current measurement units.

- `AsMils(PadXSize_MidLayer1) > 80`
  Returns all Pad objects that have a Pad X Size (Mid Layer 1) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

- `AsMM(PadXSize_MidLayer1) Between 1.524 And 3.81`
  Returns all Pad objects that have a Pad X Size (Mid Layer 1) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

- `PadXSize_MidLayer1 <= 50`
  Returns all Pad objects that have a Pad X Size (Mid Layer 1) property which is not equal to 50 current measurement units.

- `AsMils(PadXSize_MidLayer1) <= 120`
  Returns all Pad objects that have a Pad X Size (Mid Layer 1) property which is less than or equal to 120 mil.

- `AsMM(PadXSize_MidLayer1) < 1.778`
  Returns all Pad objects that have a Pad X Size (Mid Layer 1) property which is less than 1.778 mm.

---

**PadXSize_MidLayer10 Field**

**Description**

Returns pad objects whose Pad X Size (Mid Layer 10) property complies with the Query.
That property specifies a pad's X-Size distance on the MidLayer10 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 10) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 10) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer10 property is only defined for Pad objects.

Syntax

PadXSize_MidLayer10 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(PadXSize_MidLayer10) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer10) : Number {This specifies a distance with a unit of 1 mm.}

Examples

PadXSize_MidLayer10 = 100

Returns all Pad objects that have a Pad X Size (Mid Layer 10) property which is equal to 100 current measurement units.

PadXSize_MidLayer10 >= 4.064

Returns all Pad objects that have a Pad X Size (Mid Layer 10) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadXSize_MidLayer10) > 80

Returns all Pad objects that have a Pad X Size (Mid Layer 10) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadXSize_MidLayer10) Between 1.524 And 3.81

Returns all Pad objects that have a Pad X Size (Mid Layer 10) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadXSize_MidLayer10 <> 50

Returns all Pad objects that have a Pad X Size (Mid Layer 10) property which is not equal to 50 current measurement units.

AsMils(PadXSize_MidLayer10) <= 120

Returns all Pad objects that have a Pad X Size (Mid Layer 10) property which is less than or equal to 120 mil.

AsMM(PadXSize_MidLayer10) < 1.778

Returns all Pad objects that have a Pad X Size (Mid Layer 10) property which is less than 1.778 mm.

PadXSize_MidLayer11 Field

Description

Returns pad objects whose Pad X Size (Mid Layer 11) property complies with the Query.

That property specifies a pad's X-Size distance on the MidLayer11 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 11) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers
in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 11) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer11 property is only defined for Pad objects.

**Syntax**

PadXSize_MidLayer11 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(PadXSize_MidLayer11) : Number {This specifies a distance with a unit of 1 mil.}

AsMM(PadXSize_MidLayer11) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**

PadXSize_MidLayer11 = 100

Returns all Pad objects that have a Pad X Size (Mid Layer 11) property which is equal to 100 current measurement units.

PadXSize_MidLayer11 >= 4.064

Returns all Pad objects that have a Pad X Size (Mid Layer 11) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadXSize_MidLayer11) > 80

Returns all Pad objects that have a Pad X Size (Mid Layer 11) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadXSize_MidLayer11) Between 1.524 And 3.81

Returns all Pad objects that have a Pad X Size (Mid Layer 11) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadXSize_MidLayer11 <> 50

Returns all Pad objects that have a Pad X Size (Mid Layer 11) property which is not equal to 50 current measurement units.

AsMils(PadXSize_MidLayer11) <= 120

Returns all Pad objects that have a Pad X Size (Mid Layer 11) property which is less than or equal to 120 mil.

AsMM(PadXSize_MidLayer11) < 1.778

Returns all Pad objects that have a Pad X Size (Mid Layer 11) property which is less than 1.778 mm.
PadXSize_MidLayer12 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 12) property complies with the Query.

That property specifies a pad's X-Size distance on the MidLayer12 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 12) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 12) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer12 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer12 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(PadXSize_MidLayer12) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer12) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer12 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 12) property which is equal to 100 current measurement units.

PadXSize_MidLayer12 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 12) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadXSize_MidLayer12) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 12) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadXSize_MidLayer12) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 12) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadXSize_MidLayer12 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 12) property which is not equal to 50 current measurement units.

AsMils(PadXSize_MidLayer12) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 12) property which is less than or equal to 120 mil.

AsMM(PadXSize_MidLayer12) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 12) property which is less than 1.778 mm.
PadXSize_MidLayer13 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 13) property complies with the Query.
That property specifies a pad's X-Size distance on the MidLayer13 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 13) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 13) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadXSize_MidLayer13 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer13 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer13) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer13) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer13 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 13) property which is equal to 100 current measurement units.
PadXSize_MidLayer13 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 13) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadXSize_MidLayer13) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 13) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer13) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 13) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer13 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 13) property which is not equal to 50 current measurement units.
AsMils(PadXSize_MidLayer13) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 13) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer13) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 13) property which is less than 1.778 mm.
PadXSize_MidLayer14 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 14) property complies with the Query.
That property specifies a pad's X-Size distance on the MidLayer14 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 14) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 14) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadXSize_MidLayer14 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer14 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer14) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer14) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer14 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 14) property which is equal to 100 current measurement units.
PadXSize_MidLayer14 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 14) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadXSize_MidLayer14) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 14) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer14) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 14) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer14 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 14) property which is not equal to 50 current measurement units.
AsMils(PadXSize_MidLayer14) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 14) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer14) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 14) property which is less than 1.778 mm.
PadXSize_MidLayer15 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 15) property complies with the Query.
That property specifies a pad’s X-Size distance on the MidLayer15 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 15) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad’s X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad’s X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 15) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadXSize_MidLayer15 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer15 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer15) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer15) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer15 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 15) property which is equal to 100 current measurement units.
PadXSize_MidLayer15 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 15) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadXSize_MidLayer15) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 15) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer15) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 15) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer15 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 15) property which is not equal to 50 current measurement units.
AsMils(PadXSize_MidLayer15) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 15) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer15) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 15) property which is less than 1.778 mm.
PadXSize_MidLayer16 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 16) property complies with the Query.
That property specifies a pad's X-Size distance on the MidLayer16 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 16) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 16) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadXSize_MidLayer16 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer16 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer16) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer16) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer16 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 16) property which is equal to 100 current measurement units.
PadXSize_MidLayer16 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 16) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadXSize_MidLayer16) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 16) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer16) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 16) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer16 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 16) property which is not equal to 50 current measurement units.
AsMils(PadXSize_MidLayer16) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 16) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer16) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 16) property which is less than 1.778 mm.
PadXSize_MidLayer17 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 17) property complies with the Query.
That property specifies a pad's X-Size distance on the MidLayer17 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 17) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 17) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer17 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer17 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer17) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer17) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer17 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 17) property which is equal to 100 current measurement units.
PadXSize_MidLayer17 => 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 17) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadXSize_MidLayer17) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 17) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer17) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 17) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer17 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 17) property which is not equal to 50 current measurement units.
AsMils(PadXSize_MidLayer17) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 17) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer17) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 17) property which is less than 1.778 mm.
PadXSize_MidLayer18 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 18) property complies with the Query.

That property specifies a pad's X-Size distance on the MidLayer18 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 18) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 18) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer18 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer18 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer18) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer18) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer18 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 18) property which is equal to 100 current measurement units.

PadXSize_MidLayer18 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 18) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadXSize_MidLayer18) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 18) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadXSize_MidLayer18) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 18) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadXSize_MidLayer18 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 18) property which is not equal to 50 current measurement units.

AsMils(PadXSize_MidLayer18) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 18) property which is less than or equal to 120 mil.

AsMM(PadXSize_MidLayer18) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 18) property which is less than 1.778 mm.

PadXSize_MidLayer19 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 19) property complies with the Query.
That property specifies a pad's X-Size distance on the MidLayer19 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 19) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 19) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer19 property is only defined for Pad objects.

**Syntax**

```plaintext
PadXSize_MidLayer19 : Number
```

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

```plaintext
AsMils(PadXSize_MidLayer19) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer19) : Number {This specifies a distance with a unit of 1 mm.}
```

**Examples**

```plaintext
PadXSize_MidLayer19 = 100
```

Returns all Pad objects that have a Pad X Size (Mid Layer 19) property which is equal to 100 current measurement units.

```plaintext
PadXSize_MidLayer19 >= 4.064
```

Returns all Pad objects that have a Pad X Size (Mid Layer 19) property which is greater than or equal to 4.064 current measurement units.

```plaintext
AsMils(PadXSize_MidLayer19) > 80
```

Returns all Pad objects that have a Pad X Size (Mid Layer 19) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

```plaintext
AsMM(PadXSize_MidLayer19) Between 1.524 And 3.81
```

Returns all Pad objects that have a Pad X Size (Mid Layer 19) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

```plaintext
PadXSize_MidLayer19 <> 50
```

Returns all Pad objects that have a Pad X Size (Mid Layer 19) property which is **not** equal to 50 current measurement units.

```plaintext
AsMils(PadXSize_MidLayer19) <= 120
```

Returns all Pad objects that have a Pad X Size (Mid Layer 19) property which is less than or equal to 120 mil.

```plaintext
AsMM(PadXSize_MidLayer19) < 1.778
```

Returns all Pad objects that have a Pad X Size (Mid Layer 19) property which is less than 1.778 mm.
PadXSize_MidLayer2 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 2) property complies with the Query.

That property specifies a pad's X-Size distance on the MidLayer2 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 2) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 2) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer2 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer2 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer2) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer2) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer2 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 2) property which is equal to 100 current measurement units.
PadXSize_MidLayer2 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 2) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadXSize_MidLayer2) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 2) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer2) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 2) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer2 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 2) property which is not equal to 50 current measurement units.
AsMils(PadXSize_MidLayer2) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 2) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer2) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 2) property which is less than 1.778 mm.
PadXSize_MidLayer20 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 20) property complies with the Query.
That property specifies a pad's X-Size distance on the MidLayer20 layer when the pad has a 'Full Stack' Padstack Mode
property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is
assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 20) property is assessed as being equal to its
Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers
in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on
any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 20) property is assessed as being
equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is
not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property;
pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadXSize_MidLayer20 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer20 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001
inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001
meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer20) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer20) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer20 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 20) property which is equal to 100 current measurement units.
PadXSize_MidLayer20 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 20) property which is greater than or equal to 4.064 current
measurement units.
AsMils(PadXSize_MidLayer20) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 20) property which is greater than 80 mil. (The current value of the
target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer20) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 20) property which is greater than or equal to 1.524 mm and less
than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer20 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 20) property which is not equal to 50 current measurement units.
AsMils(PadXSize_MidLayer20) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 20) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer20) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 20) property which is less than 1.778 mm.
PadXSize_MidLayer21 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 21) property complies with the Query.
That property specifies a pad's X-Size distance on the MidLayer21 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 21) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 21) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer21 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer21 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer21) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer21) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer21 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 21) property which is equal to 100 current measurement units.
PadXSize_MidLayer21 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 21) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadXSize_MidLayer21) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 21) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer21) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 21) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer21 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 21) property which is not equal to 50 current measurement units.
AsMils(PadXSize_MidLayer21) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 21) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer21) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 21) property which is less than 1.778 mm.
PadXSize_MidLayer22 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 22) property complies with the Query.
That property specifies a pad's X-Size distance on the MidLayer22 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 22) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 22) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadXSize_MidLayer22 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer22 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer22) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer22) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer22 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 22) property which is equal to 100 current measurement units.
PadXSize_MidLayer22 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 22) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadXSize_MidLayer22) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 22) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer22) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 22) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer22 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 22) property which is not equal to 50 current measurement units.
AsMils(PadXSize_MidLayer22) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 22) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer22) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 22) property which is less than 1.778 mm.
PadXSize_MidLayer23 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 23) property complies with the Query.
That property specifies a pad’s X-Size distance on the MidLayer23 layer when the pad has a ‘Full Stack’ Padstack Mode property (and by implication, a ‘Multi-Layer’ Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a ‘Simple’ Padstack Mode property, its Pad X Size (Mid Layer 23) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad’s X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad’s X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)
For any pad having a ‘Top-Middle-Bottom’ Padstack Mode property, its Pad X Size (Mid Layer 23) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or “Middle”) Signal layers. (It is not possible for a pad to have a ‘Top-Middle-Bottom’ Padstack Mode property unless it also has a ‘Multi-Layer’ Layer property; pads on all remaining layers can only have a ‘Simple’ Padstack Mode property.)
Note: The PadXSize_MidLayer23 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer23 : Number
If the target file has an ‘Imperial’ Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a ‘Metric’ Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer23) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer23) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer23 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 23) property which is equal to 100 current measurement units.
PadXSize_MidLayer23 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 23) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadXSize_MidLayer23) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 23) property which is greater than 80 mil. (The current value of the target file’s Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer23) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 23) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file’s Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer23 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 23) property which is not equal to 50 current measurement units.
AsMils(PadXSize_MidLayer23) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 23) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer23) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 23) property which is less than 1.778 mm.
PadXSize_MidLayer24 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 24) property complies with the Query. That property specifies a pad's X-Size distance on the MidLayer24 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 24) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 24) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer24 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer24 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer24) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer24) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer24 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 24) property which is equal to 100 current measurement units.
PadXSize_MidLayer24 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 24) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadXSize_MidLayer24) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 24) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer24) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 24) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer24 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 24) property which is not equal to 50 current measurement units.
AsMils(PadXSize_MidLayer24) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 24) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer24) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 24) property which is less than 1.778 mm.
PadXSize_MidLayer25 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 25) property complies with the Query.

That property specifies a pad's X-Size distance on the MidLayer25 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 25) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 25) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer25 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer25 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer25) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer25) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer25 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 25) property which is equal to 100 current measurement units.
PadXSize_MidLayer25 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 25) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadXSize_MidLayer25) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 25) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer25) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 25) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer25 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 25) property which is not equal to 50 current measurement units.
AsMils(PadXSize_MidLayer25) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 25) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer25) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 25) property which is less than 1.778 mm.
PadXSize_MidLayer26 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 26) property complies with the Query.
That property specifies a pad's X-Size distance on the MidLayer26 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 26) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 26) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadXSize_MidLayer26 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer26 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer26) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer26) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer26 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 26) property which is equal to 100 current measurement units.
PadXSize_MidLayer26 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 26) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadXSize_MidLayer26) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 26) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer26) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 26) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer26 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 26) property which is not equal to 50 current measurement units.
AsMils(PadXSize_MidLayer26) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 26) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer26) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 26) property which is less than 1.778 mm.
PadXSize_MidLayer27 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 27) property complies with the Query.
That property specifies a pad's X-Size distance on the MidLayer27 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 27) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 27) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer27 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer27 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer27) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer27) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer27 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 27) property which is equal to 100 current measurement units.

PadXSize_MidLayer27 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 27) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadXSize_MidLayer27) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 27) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadXSize_MidLayer27) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 27) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadXSize_MidLayer27 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 27) property which is not equal to 50 current measurement units.

AsMils(PadXSize_MidLayer27) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 27) property which is less than or equal to 120 mil.

AsMM(PadXSize_MidLayer27) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 27) property which is less than 1.778 mm.
PadXSize_MidLayer28 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 28) property complies with the Query.
That property specifies a pad’s X-Size distance on the MidLayer28 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 28) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad’s X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad’s X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 28) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer28 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer28 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer28) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer28) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer28 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 28) property which is equal to 100 current measurement units.
PadXSize_MidLayer28 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 28) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadXSize_MidLayer28) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 28) property which is greater than 80 mil. (The current value of the target file’s Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer28) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 28) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file’s Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer28 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 28) property which is not equal to 50 current measurement units.
AsMils(PadXSize_MidLayer28) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 28) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer28) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 28) property which is less than 1.778 mm.
PadXSize_MidLayer29 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 29) property complies with the Query. That property specifies a pad's X-Size distance on the MidLayer29 layer when the pad has a 'Full Stack' Padstack Mode property and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 29) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 29) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer29 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer29 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer29) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer29) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer29 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 29) property which is equal to 100 current measurement units.

PadXSize_MidLayer29 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 29) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadXSize_MidLayer29) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 29) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadXSize_MidLayer29) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 29) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadXSize_MidLayer29 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 29) property which is not equal to 50 current measurement units.

AsMils(PadXSize_MidLayer29) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 29) property which is less than or equal to 120 mil.

AsMM(PadXSize_MidLayer29) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 29) property which is less than 1.778 mm.
PadXSize_MidLayer3 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 3) property complies with the Query.
That property specifies a pad's X-Size distance on the MidLayer3 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 3) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 3) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer3 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer3 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer3) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer3) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer3 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 3) property which is equal to 100 current measurement units.

PadXSize_MidLayer3 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 3) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadXSize_MidLayer3) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 3) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadXSize_MidLayer3) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 3) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadXSize_MidLayer3 <= 50
Returns all Pad objects that have a Pad X Size (Mid Layer 3) property which is not equal to 50 current measurement units.

AsMils(PadXSize_MidLayer3) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 3) property which is less than or equal to 120 mil.

AsMM(PadXSize_MidLayer3) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 3) property which is less than 1.778 mm.
PadXSize_MidLayer30 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 30) property complies with the Query.

That property specifies a pad's X-Size distance on the MidLayer30 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 30) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 30) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer30 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer30 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer30) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer30) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer30 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 30) property which is equal to 100 current measurement units.

PadXSize_MidLayer30 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 30) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadXSize_MidLayer30) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 30) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadXSize_MidLayer30) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 30) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadXSize_MidLayer30 <> 50
Returns all Pad objects that have a Pad X Size (Mid Layer 30) property which is not equal to 50 current measurement units.

AsMils(PadXSize_MidLayer30) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 30) property which is less than or equal to 120 mil.

AsMM(PadXSize_MidLayer30) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 30) property which is less than 1.778 mm.
PadXSize_MidLayer4 Field

Description

Returns pad objects whose Pad X Size (Mid Layer 4) property complies with the Query.

That property specifies a pad's X-Size distance on the MidLayer4 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 4) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 4) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer4 property is only defined for Pad objects.

Syntax

PadXSize_MidLayer4 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(PadXSize_MidLayer4) : Number {This specifies a distance with a unit of 1 mil.}

AsMM(PadXSize_MidLayer4) : Number {This specifies a distance with a unit of 1 mm.}

Examples

PadXSize_MidLayer4 = 100

Returns all Pad objects that have a Pad X Size (Mid Layer 4) property which is equal to 100 current measurement units.

PadXSize_MidLayer4 >= 4.064

Returns all Pad objects that have a Pad X Size (Mid Layer 4) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadXSize_MidLayer4) > 80

Returns all Pad objects that have a Pad X Size (Mid Layer 4) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadXSize_MidLayer4) Between 1.524 And 3.81

Returns all Pad objects that have a Pad X Size (Mid Layer 4) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadXSize_MidLayer4 <= 50

Returns all Pad objects that have a Pad X Size (Mid Layer 4) property which is not equal to 50 current measurement units.

AsMils(PadXSize_MidLayer4) <= 120

Returns all Pad objects that have a Pad X Size (Mid Layer 4) property which is less than or equal to 120 mil.

AsMM(PadXSize_MidLayer4) < 1.778

Returns all Pad objects that have a Pad X Size (Mid Layer 4) property which is less than 1.778 mm.
PadXSize_MidLayer5 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 5) property complies with the Query.
That property specifies a pad's X-Size distance on the MidLayer5 layer when the pad has a 'Full Stack' Padstack Mode property
(and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed
for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 5) property is assessed as being equal to its
Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers
in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on
any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 5) property is assessed as being
equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is
not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property;
pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadXSize_MidLayer5 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer5 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001
inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001
meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer5) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer5) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer5 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 5) property which is equal to 100 current measurement units.
PadXSize_MidLayer5 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 5) property which is greater than or equal to 4.064 current
measurement units.
AsMils(PadXSize_MidLayer5) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 5) property which is greater than 80 mil. (The current value of the
target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer5) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 5) property which is greater than or equal to 1.524 mm and less than
or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer5 <= 50
Returns all Pad objects that have a Pad X Size (Mid Layer 5) property which is not equal to 50 current measurement units.
AsMils(PadXSize_MidLayer5) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 5) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer5) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 5) property which is less than 1.778 mm.
PadXSize_MidLayer6 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 6) property complies with the Query.
That property specifies a pad's X-Size distance on the MidLayer6 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 6) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 6) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadXSize_MidLayer6 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer6 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer6) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer6) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer6 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 6) property which is equal to 100 current measurement units.
PadXSize_MidLayer6 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 6) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadXSize_MidLayer6) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 6) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer6) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 6) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer6 <= 50
Returns all Pad objects that have a Pad X Size (Mid Layer 6) property which is not equal to 50 current measurement units.
AsMils(PadXSize_MidLayer6) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 6) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer6) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 6) property which is less than 1.778 mm.
PadXSize_MidLayer7 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 7) property complies with the Query. That property specifies a pad's X-Size distance on the MidLayer7 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 7) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 7) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer7 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer7 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer7) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer7) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer7 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 7) property which is equal to 100 current measurement units.
PadXSize_MidLayer7 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 7) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadXSize_MidLayer7) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 7) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer7) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 7) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer7 <= 50
Returns all Pad objects that have a Pad X Size (Mid Layer 7) property which is equal to 50 current measurement units.
AsMils(PadXSize_MidLayer7) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 7) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer7) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 7) property which is less than 1.778 mm.
PadXSize_MidLayer8 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 8) property complies with the Query. That property specifies a pad's X-Size distance on the MidLayer8 layer when the pad has a 'Full Stack' Padstack Mode property and by implication, a 'Multi-Layer' Layer property. However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 8) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 8) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadXSize_MidLayer8 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer8 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer8) : Number (This specifies a distance with a unit of 1 mil.)
AsMM(PadXSize_MidLayer8) : Number (This specifies a distance with a unit of 1 mm.)

Examples
PadXSize_MidLayer8 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 8) property which is equal to 100 current measurement units.
PadXSize_MidLayer8 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 8) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadXSize_MidLayer8) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 8) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer8) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 8) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer8 <= 50
Returns all Pad objects that have a Pad X Size (Mid Layer 8) property which is not equal to 50 current measurement units.
AsMils(PadXSize_MidLayer8) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 8) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer8) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 8) property which is less than 1.778 mm.
PadXSize_MidLayer9 Field

Description
Returns pad objects whose Pad X Size (Mid Layer 9) property complies with the Query.
That property specifies a pad's X-Size distance on the MidLayer9 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Mid Layer 9) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad X Size (Mid Layer 9) property is assessed as being equal to its Pad X Size (Mid Layer 1) property, which specifies its X-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadXSize_MidLayer9 property is only defined for Pad objects.

Syntax
PadXSize_MidLayer9 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadXSize_MidLayer9) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_MidLayer9) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadXSize_MidLayer9 = 100
Returns all Pad objects that have a Pad X Size (Mid Layer 9) property which is equal to 100 current measurement units.
PadXSize_MidLayer9 >= 4.064
Returns all Pad objects that have a Pad X Size (Mid Layer 9) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadXSize_MidLayer9) > 80
Returns all Pad objects that have a Pad X Size (Mid Layer 9) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadXSize_MidLayer9) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Mid Layer 9) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadXSize_MidLayer9 <= 50
Returns all Pad objects that have a Pad X Size (Mid Layer 9) property which is not equal to 50 current measurement units.
AsMils(PadXSize_MidLayer9) <= 120
Returns all Pad objects that have a Pad X Size (Mid Layer 9) property which is less than or equal to 120 mil.
AsMM(PadXSize_MidLayer9) < 1.778
Returns all Pad objects that have a Pad X Size (Mid Layer 9) property which is less than 1.778 mm.


**PadXSize_TopLayer Field**

**Description**

Returns pad objects whose Pad X Size (Top Layer) property complies with the Query.

That property specifies a pad's X-size distance on the Top Signal layer when the pad has a 'Top-Middle-Bottom' or 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad X Size (Top Layer) property is assessed as being equal to its Pad X Size (All Layers) property. (The Pad X Size (All Layers) property specifies the pad's X-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's X-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

Note: The PadXSize_TopLayer property is only defined for Pad objects.

**Syntax**

```
PadXSize_TopLayer : Number
```

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

```
AsMils(PadXSize_TopLayer) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadXSize_TopLayer) : Number {This specifies a distance with a unit of 1 mm.}
```

**Examples**

```
PadXSize_TopLayer = 100
Returns all Pad objects that have a Pad X Size (Top Layer) property which is equal to 100 current measurement units.
```

```
PadXSize_TopLayer >= 4.064
Returns all Pad objects that have a Pad X Size (Top Layer) property which is greater than or equal to 4.064 current measurement units.
```

```
AsMils(PadXSize_TopLayer) > 80
Returns all Pad objects that have a Pad X Size (Top Layer) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
```

```
AsMM(PadXSize_TopLayer) Between 1.524 And 3.81
Returns all Pad objects that have a Pad X Size (Top Layer) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
```

```
PadXSize_TopLayer <> 50
Returns all Pad objects that have a Pad X Size (Top Layer) property which is not equal to 50 current measurement units.
```

```
AsMils(PadXSize_TopLayer) <= 120
Returns all Pad objects that have a Pad X Size (Top Layer) property which is less than or equal to 120 mil.
```

```
AsMM(PadXSize_TopLayer) < 1.778
Returns all Pad objects that have a Pad X Size (Top Layer) property which is less than 1.778 mm.
```
PadYSize_AllLayers Field

Description

Returns pad objects whose Pad Y Size (All Layers) property complies with the Query.

When a pad has a 'Simple' Padstack Mode property, the Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.

However, this property is not defined for any pad which has a 'Top-Middle-Bottom' or 'Full Stack' Padstack Mode property instead. (Such pads can only be on the Multi-Layer layer.) It is possible for such pads to have different Y-Size distances on different (Signal) layers, so it is not meaningful to define such a property for those pads. (Even when such a pad does have the same Y-Size distances on all of the (Signal) layers, this property is still not defined.)

Note: The PadYSize_AllLayers property is only defined for Pad objects with a 'Simple' Padstack Mode property.

Syntax

PadYSize_AllLayers : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(PadYSize_AllLayers) : Number {This specifies a distance with a unit of 1 mil.}

AsMM(PadYSize_AllLayers) : Number {This specifies a distance with a unit of 1 mm.}

Examples

PadYSize_AllLayers = 100

Returns all Pad objects that have a Pad Y Size (All Layers) property which is equal to 100 current measurement units.

PadYSize_AllLayers >= 4.064

Returns all Pad objects that have a Pad Y Size (All Layers) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadYSize_AllLayers) > 80

Returns all Pad objects that have a Pad Y Size (All Layers) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadYSize_AllLayers) Between 1.524 And 3.81

Returns all Pad objects that have a Pad Y Size (All Layers) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadYSize_AllLayers <> 50

Returns all Pad objects that have a Pad Y Size (All Layers) property which is not equal to 50 current measurement units.

AsMils(PadYSize_AllLayers) <= 120

Returns all Pad objects that have a Pad Y Size (All Layers) property which is less than or equal to 120 mil.

AsMM(PadYSize_AllLayers) < 1.778

Returns all Pad objects that have a Pad Y Size (All Layers) property which is less than 1.778 mm.
PadYSize_BottomLayer Field

Description
Returns pad objects whose Pad Y Size (Bottom Layer) property complies with the Query. 
That property specifies a pad's Y-Size distance on the Bottom Signal layer when the pad has a 'Top-Middle-Bottom' or 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Bottom Layer) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

Note: The PadYSize_BottomLayer property is only defined for Pad objects.

Syntax
PadYSize_BottomLayer : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_BottomLayer) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_BottomLayer) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_BottomLayer = 100
Returns all Pad objects that have a Pad Y Size (Bottom Layer) property which is equal to 100 current measurement units.

PadYSize_BottomLayer >= 4.064
Returns all Pad objects that have a Pad Y Size (Bottom Layer) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadYSize_BottomLayer) > 80
Returns all Pad objects that have a Pad Y Size (Bottom Layer) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadYSize_BottomLayer) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Bottom Layer) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadYSize_BottomLayer <> 50
Returns all Pad objects that have a Pad Y Size (Bottom Layer) property which is not equal to 50 current measurement units.

AsMils(PadYSize_BottomLayer) <= 120
Returns all Pad objects that have a Pad Y Size (Bottom Layer) property which is less than or equal to 120 mil.

AsMM(PadYSize_BottomLayer) < 1.778
Returns all Pad objects that have a Pad Y Size (Bottom Layer) property which is less than 1.778 mm.
PadYSize_MidLayer1 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 1) property complies with the Query. That property specifies a pad's Y-Size distance on the MidLayer1 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 1) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 1) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer1 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer1 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer1) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer1) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer1 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 1) property which is equal to 100 current measurement units.
PadYSize_MidLayer1 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 1) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadYSize_MidLayer1) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 1) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer1) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 1) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer1 <= 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 1) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer1) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 1) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer1) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 1) property which is less than 1.778 mm.
PadYSize_MidLayer10 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 10) property complies with the Query.
That property specifies a pad's Y-Size distance on the MidLayer10 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 10) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 10) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadYSize_MidLayer10 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer10 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer10) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer10) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer10 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 10) property which is equal to 100 current measurement units.
PadYSize_MidLayer10 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 10) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadYSize_MidLayer10) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 10) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer10) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 10) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer10 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 10) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer10) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 10) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer10) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 10) property which is less than 1.778 mm.
PadYSize_MidLayer11 Field

**Description**

Returns pad objects whose Pad Y Size (Mid Layer 11) property complies with the Query.

That property specifies a pad's Y-Size distance on the MidLayer11 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 11) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 11) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer11 property is only defined for Pad objects.

**Syntax**

PadYSize_MidLayer11 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

- AsMils(PadYSize_MidLayer11) : Number (This specifies a distance with a unit of 1 mil.)
- AsMM(PadYSize_MidLayer11) : Number (This specifies a distance with a unit of 1 mm.)

**Examples**

- PadYSize_MidLayer11 = 100
  Returns all Pad objects that have a Pad Y Size (Mid Layer 11) property which is equal to 100 current measurement units.
- PadYSize_MidLayer11 >= 4.064
  Returns all Pad objects that have a Pad Y Size (Mid Layer 11) property which is greater than or equal to 4.064 current measurement units.
- AsMils(PadYSize_MidLayer11) > 80
  Returns all Pad objects that have a Pad Y Size (Mid Layer 11) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
- AsMM(PadYSize_MidLayer11) Between 1.524 And 3.81
  Returns all Pad objects that have a Pad Y Size (Mid Layer 11) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
- PadYSize_MidLayer11 <= 50
  Returns all Pad objects that have a Pad Y Size (Mid Layer 11) property which is not equal to 50 current measurement units.
- AsMils(PadYSize_MidLayer11) <= 120
  Returns all Pad objects that have a Pad Y Size (Mid Layer 11) property which is less than or equal to 120 mil.
- AsMM(PadYSize_MidLayer11) < 1.778
  Returns all Pad objects that have a Pad Y Size (Mid Layer 11) property which is less than 1.778 mm.
PadYSize_MidLayer12 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 12) property complies with the Query.

That property specifies a pad’s Y-Size distance on the MidLayer12 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 12) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad’s Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad’s Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 12) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer12 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer12 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer12) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer12) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer12 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 12) property which is equal to 100 current measurement units.

PadYSize_MidLayer12 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 12) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadYSize_MidLayer12) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 12) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadYSize_MidLayer12) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 12) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadYSize_MidLayer12 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 12) property which is not equal to 50 current measurement units.

AsMils(PadYSize_MidLayer12) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 12) property which is less than or equal to 120 mil.

AsMM(PadYSize_MidLayer12) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 12) property which is less than 1.778 mm.
PadYSize_MidLayer13 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 13) property complies with the Query.

That property specifies a pad’s Y-Size distance on the MidLayer13 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 13) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad’s Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad’s Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 13) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer13 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer13 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer13) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer13) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer13 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 13) property which is equal to 100 current measurement units.
PadYSize_MidLayer13 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 13) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadYSize_MidLayer13) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 13) property which is greater than 80 mil. (The current value of the target file’s Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer13) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 13) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file’s Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer13 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 13) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer13) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 13) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer13) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 13) property which is less than 1.778 mm.
PadYSize_MidLayer14 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 14) property complies with the Query.

That property specifies a pad's Y-Size distance on the MidLayer14 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 14) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 14) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer14 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer14 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer14) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer14) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer14 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 14) property which is equal to 100 current measurement units.
PadYSize_MidLayer14 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 14) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadYSize_MidLayer14) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 14) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer14) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 14) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer14 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 14) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer14) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 14) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer14) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 14) property which is less than 1.778 mm.
PadYSize_MidLayer15 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 15) property complies with the Query.
That property specifies a pad's Y-Size distance on the MidLayer15 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 15) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 15) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadYSize_MidLayer15 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer15 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer15) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer15) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer15 = 100 Returns all Pad objects that have a Pad Y Size (Mid Layer 15) property which is equal to 100 current measurement units.
PadYSize_MidLayer15 >= 4.064 Returns all Pad objects that have a Pad Y Size (Mid Layer 15) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadYSize_MidLayer15) > 80 Returns all Pad objects that have a Pad Y Size (Mid Layer 15) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer15) Between 1.524 And 3.81 Returns all Pad objects that have a Pad Y Size (Mid Layer 15) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer15 <> 50 Returns all Pad objects that have a Pad Y Size (Mid Layer 15) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer15) <= 120 Returns all Pad objects that have a Pad Y Size (Mid Layer 15) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer15) < 1.778 Returns all Pad objects that have a Pad Y Size (Mid Layer 15) property which is less than 1.778 mm.
**PadYSize_MidLayer16 Field**

**Description**

Returns pad objects whose Pad Y Size (Mid Layer 16) property complies with the Query.

That property specifies a pad's Y-Size distance on the MidLayer16 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 16) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 16) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer16 property is only defined for Pad objects.

**Syntax**

PadYSize_MidLayer16 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(PadYSize_MidLayer16) : Number {This specifies a distance with a unit of 1 mil.}

AsMM(PadYSize_MidLayer16) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**

PadYSize_MidLayer16 = 100

Returns all Pad objects that have a Pad Y Size (Mid Layer 16) property which is equal to 100 current measurement units.

PadYSize_MidLayer16 >= 4.064

Returns all Pad objects that have a Pad Y Size (Mid Layer 16) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadYSize_MidLayer16) > 80

Returns all Pad objects that have a Pad Y Size (Mid Layer 16) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadYSize_MidLayer16) Between 1.524 And 3.81

Returns all Pad objects that have a Pad Y Size (Mid Layer 16) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadYSize_MidLayer16 <> 50

Returns all Pad objects that have a Pad Y Size (Mid Layer 16) property which is not equal to 50 current measurement units.

AsMils(PadYSize_MidLayer16) <= 120

Returns all Pad objects that have a Pad Y Size (Mid Layer 16) property which is less than or equal to 120 mil.

AsMM(PadYSize_MidLayer16) < 1.778

Returns all Pad objects that have a Pad Y Size (Mid Layer 16) property which is less than 1.778 mm.
PadYSize_MidLayer17 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 17) property complies with the Query.
That property specifies a pad's Y-Size distance on the MidLayer17 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 17) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 17) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadYSize_MidLayer17 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer17 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer17) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer17) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer17 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 17) property which is equal to 100 current measurement units.
PadYSize_MidLayer17 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 17) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadYSize_MidLayer17) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 17) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer17) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 17) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer17 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 17) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer17) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 17) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer17) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 17) property which is less than 1.778 mm.
PadYSize_MidLayer18 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 18) property complies with the Query.
That property specifies a pad's Y-Size distance on the MidLayer18 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 18) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 18) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer18 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer18 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer18) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer18) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer18 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 18) property which is equal to 100 current measurement units.
PadYSize_MidLayer18 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 18) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadYSize_MidLayer18) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 18) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer18) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 18) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer18 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 18) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer18) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 18) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer18) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 18) property which is less than 1.778 mm.
PadYSize_MidLayer19 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 19) property complies with the Query.
That property specifies a pad's Y-Size distance on the MidLayer19 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 19) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 19) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer19 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer19 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer19) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer19) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer19 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 19) property which is equal to 100 current measurement units.

PadYSize_MidLayer19 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 19) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadYSize_MidLayer19) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 19) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadYSize_MidLayer19) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 19) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadYSize_MidLayer19 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 19) property which is not equal to 50 current measurement units.

AsMils(PadYSize_MidLayer19) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 19) property which is less than or equal to 120 mil.

AsMM(PadYSize_MidLayer19) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 19) property which is less than 1.778 mm.
PadYSize_MidLayer2 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 2) property complies with the Query. That property specifies a pad’s Y-Size distance on the MidLayer2 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 2) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad’s Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 2) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer2 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer2 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer2) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer2) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer2 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 2) property which is equal to 100 current measurement units.
PadYSize_MidLayer2 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 2) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadYSize_MidLayer2) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 2) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer2) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 2) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer2 <= 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 2) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer2) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 2) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer2) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 2) property which is less than 1.778 mm.
PadYSize_MidLayer20 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 20) property complies with the Query.
That property specifies a pad's Y-Size distance on the MidLayer20 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 20) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 20) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadYSize_MidLayer20 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer20 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer20) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer20) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer20 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 20) property which is equal to 100 current measurement units.
PadYSize_MidLayer20 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 20) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadYSize_MidLayer20) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 20) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer20) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 20) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer20 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 20) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer20) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 20) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer20) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 20) property which is less than 1.778 mm.
PadYSize_MidLayer21 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 21) property complies with the Query.
That property specifies a pad's Y-Size distance on the MidLayer21 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 21) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 21) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadYSize_MidLayer21 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer21 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer21) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer21) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer21 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 21) property which is equal to 100 current measurement units.
PadYSize_MidLayer21 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 21) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadYSize_MidLayer21) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 21) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer21) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 21) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer21 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 21) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer21) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 21) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer21) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 21) property which is less than 1.778 mm.
PadYSize_MidLayer22 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 22) property complies with the Query.
That property specifies a pad's Y-Size distance on the MidLayer22 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 22) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad’s Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 22) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer22 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer22 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer22) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer22) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer22 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 22) property which is equal to 100 current measurement units.
PadYSize_MidLayer22 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 22) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadYSize_MidLayer22) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 22) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer22) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 22) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer22 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 22) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer22) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 22) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer22) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 22) property which is less than 1.778 mm.
PadYSize_MidLayer23 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 23) property complies with the Query.
That property specifies a pad's Y-Size distance on the MidLayer23 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 23) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 23) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer23 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer23 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer23) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer23) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer23 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 23) property which is equal to 100 current measurement units.
PadYSize_MidLayer23 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 23) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadYSize_MidLayer23) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 23) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer23) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 23) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer23 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 23) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer23) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 23) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer23) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 23) property which is less than 1.778 mm.
PadYSize_MidLayer24 Field

Description

Returns pad objects whose Pad Y Size (Mid Layer 24) property complies with the Query. That property specifies a pad's Y-Size distance on the MidLayer24 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 24) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 24) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer24 property is only defined for Pad objects.

Syntax

PadYSize_MidLayer24 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

- AsMils(PadYSize_MidLayer24) : Number {This specifies a distance with a unit of 1 mil.}
- AsMM(PadYSize_MidLayer24) : Number {This specifies a distance with a unit of 1 mm.}

Examples

PadYSize_MidLayer24 = 100

Returns all Pad objects that have a Pad Y Size (Mid Layer 24) property which is equal to 100 current measurement units.

PadYSize_MidLayer24 >= 4.064

Returns all Pad objects that have a Pad Y Size (Mid Layer 24) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadYSize_MidLayer24) > 80

Returns all Pad objects that have a Pad Y Size (Mid Layer 24) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadYSize_MidLayer24) Between 1.524 And 3.81

Returns all Pad objects that have a Pad Y Size (Mid Layer 24) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadYSize_MidLayer24 <> 50

Returns all Pad objects that have a Pad Y Size (Mid Layer 24) property which is not equal to 50 current measurement units.

AsMils(PadYSize_MidLayer24) <= 120

Returns all Pad objects that have a Pad Y Size (Mid Layer 24) property which is less than or equal to 120 mil.

AsMM(PadYSize_MidLayer24) < 1.778

Returns all Pad objects that have a Pad Y Size (Mid Layer 24) property which is less than 1.778 mm.
PadYSize_MidLayer25 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 25) property complies with the Query.

That property specifies a pad’s Y-Size distance on the MidLayer25 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 25) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad’s Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad’s Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 25) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer25 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer25 : Number

If the target file has an ‘Imperial’ Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a ‘Metric’ Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer25) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer25) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer25 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 25) property which is equal to 100 current measurement units.

PadYSize_MidLayer25 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 25) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadYSize_MidLayer25) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 25) property which is greater than 80 mil. (The current value of the target file’s Measurement Unit property has no impact upon this outcome.)

AsMM(PadYSize_MidLayer25) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 25) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file’s Measurement Unit property has no impact upon this outcome.)

PadYSize_MidLayer25 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 25) property which is not equal to 50 current measurement units.

AsMils(PadYSize_MidLayer25) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 25) property which is less than or equal to 120 mil.

AsMM(PadYSize_MidLayer25) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 25) property which is less than 1.778 mm.
PadYSize_MidLayer26 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 26) property complies with the Query.
That property specifies a pad’s Y-Size distance on the MidLayer26 layer when the pad has a ‘Full Stack’ Padstack Mode property (and by implication, a ‘Multi-Layer’ Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a ‘Simple’ Padstack Mode property, its Pad Y Size (Mid Layer 26) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad’s Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad’s Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a ‘Top-Middle-Bottom’ Padstack Mode property, its Pad Y Size (Mid Layer 26) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or “Middle”) Signal layers. (It is not possible for a pad to have a ‘Top-Middle-Bottom’ Padstack Mode property unless it also has a ‘Multi-Layer’ Layer property; pads on all remaining layers can only have a ‘Simple’ Padstack Mode property.)

Note: The PadYSize_MidLayer26 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer26 : Number
If the target file has an ‘Imperial’ Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a ‘Metric’ Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer26) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer26) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer26 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 26) property which is equal to 100 current measurement units.
PadYSize_MidLayer26 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 26) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadYSize_MidLayer26) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 26) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer26) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 26) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer26 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 26) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer26) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 26) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer26) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 26) property which is less than 1.778 mm.
PadYSize_MidLayer27 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 27) property complies with the Query.
That property specifies a pad's Y-Size distance on the MidLayer27 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 27) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 27) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer27 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer27 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer27) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer27) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer27 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 27) property which is equal to 100 current measurement units.

PadYSize_MidLayer27 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 27) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadYSize_MidLayer27) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 27) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadYSize_MidLayer27) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 27) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadYSize_MidLayer27 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 27) property which is not equal to 50 current measurement units.

AsMils(PadYSize_MidLayer27) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 27) property which is less than or equal to 120 mil.

AsMM(PadYSize_MidLayer27) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 27) property which is less than 1.778 mm.
PadYSize_MidLayer28 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 28) property complies with the Query.
That property specifies a pad's Y-Size distance on the MidLayer28 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 28) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 28) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer28 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer28 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer28) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer28) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer28 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 28) property which is equal to 100 current measurement units.
PadYSize_MidLayer28 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 28) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadYSize_MidLayer28) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 28) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer28) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 28) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer28 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 28) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer28) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 28) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer28) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 28) property which is less than 1.778 mm.
PadYSize_MidLayer29 Field

Description

Returns pad objects whose Pad Y Size (Mid Layer 29) property complies with the Query.

That property specifies a pad's Y-Size distance on the MidLayer29 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 29) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 29) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer29 property is only defined for Pad objects.

Syntax

PadYSize_MidLayer29 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(PadYSize_MidLayer29) : Number {This specifies a distance with a unit of 1 mil.}

AsMM(PadYSize_MidLayer29) : Number {This specifies a distance with a unit of 1 mm.}

Examples

PadYSize_MidLayer29 = 100

Returns all Pad objects that have a Pad Y Size (Mid Layer 29) property which is equal to 100 current measurement units.

PadYSize_MidLayer29 >= 4.064

Returns all Pad objects that have a Pad Y Size (Mid Layer 29) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadYSize_MidLayer29) > 80

Returns all Pad objects that have a Pad Y Size (Mid Layer 29) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadYSize_MidLayer29) Between 1.524 And 3.81

Returns all Pad objects that have a Pad Y Size (Mid Layer 29) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadYSize_MidLayer29 <> 50

Returns all Pad objects that have a Pad Y Size (Mid Layer 29) property which is not equal to 50 current measurement units.

AsMils(PadYSize_MidLayer29) <= 120

Returns all Pad objects that have a Pad Y Size (Mid Layer 29) property which is less than or equal to 120 mil.

AsMM(PadYSize_MidLayer29) < 1.778

Returns all Pad objects that have a Pad Y Size (Mid Layer 29) property which is less than 1.778 mm.
PadYSize_MidLayer3 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 3) property complies with the Query.
That property specifies a pad's Y-Size distance on the MidLayer3 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 3) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 3) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer3 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer3 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer3) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer3) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer3 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 3) property which is equal to 100 current measurement units.
PadYSize_MidLayer3 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 3) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadYSize_MidLayer3) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 3) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer3) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 3) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer3 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 3) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer3) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 3) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer3) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 3) property which is less than 1.778 mm.
PadYSize_MidLayer30 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 30) property complies with the Query.
That property specifies a pad's Y-Size distance on the MidLayer30 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 30) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 30) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadYSize_MidLayer30 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer30 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer30) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer30) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer30 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 30) property which is equal to 100 current measurement units.
PadYSize_MidLayer30 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 30) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadYSize_MidLayer30) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 30) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer30) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 30) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer30 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 30) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer30) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 30) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer30) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 30) property which is less than 1.778 mm.
PadYSize_MidLayer4 Field

**Description**

Returns pad objects whose Pad Y Size (Mid Layer 4) property complies with the Query.

That property specifies a pad's Y-Size distance on the MidLayer4 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 4) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 4) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer4 property is only defined for Pad objects.

**Syntax**

PaddingSize_MidLayer4 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(PaddingSize_MidLayer4) : Number {This specifies a distance with a unit of 1 mil.}

AsMM(PaddingSize_MidLayer4) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**

PadYSize_MidLayer4 = 100

Returns all Pad objects that have a Pad Y Size (Mid Layer 4) property which is equal to 100 current measurement units.

PadYSize_MidLayer4 >= 4.064

Returns all Pad objects that have a Pad Y Size (Mid Layer 4) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadYSize_MidLayer4) > 80

Returns all Pad objects that have a Pad Y Size (Mid Layer 4) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadYSize_MidLayer4) Between 1.524 And 3.81

Returns all Pad objects that have a Pad Y Size (Mid Layer 4) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadYSize_MidLayer4 <= 50

Returns all Pad objects that have a Pad Y Size (Mid Layer 4) property which is equal to 50 current measurement units.

AsMils(PadYSize_MidLayer4) <= 120

Returns all Pad objects that have a Pad Y Size (Mid Layer 4) property which is less than or equal to 120 mil.

AsMM(PadYSize_MidLayer4) < 1.778

Returns all Pad objects that have a Pad Y Size (Mid Layer 4) property which is less than 1.778 mm.
PadYSize_MidLayer5 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 5) property complies with the Query.
That property specifies a pad's Y-Size distance on the MidLayer5 layer when the pad has a 'Full Stack' Padstack Mode property
(and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed
for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 5) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers
in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on
any of the other layers.)
For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 5) property is assessed as being equal
to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is
not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property;
pads on all remaining layers can only have a 'Simple' Padstack Mode property.)
Note: The PadYSize_MidLayer5 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer5 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001
inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001
meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer5) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer5) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer5 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 5) property which is equal to 100 current measurement units.
PadYSize_MidLayer5 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 5) property which is greater than or equal to 4.064 current
measurement units.
AsMils(PadYSize_MidLayer5) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 5) property which is greater than 80 mil. (The current value of the
target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer5) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 5) property which is greater than or equal to 1.524 mm and less than
or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer5 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 5) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer5) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 5) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer5) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 5) property which is less than 1.778 mm.
PadYSize_MidLayer6 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 6) property complies with the Query.

That property specifies a pad's Y-Size distance on the MidLayer6 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 6) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 6) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer6 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer6 : Number  
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer6) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer6) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer6 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 6) property which is equal to 100 current measurement units.
PadYSize_MidLayer6 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 6) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadYSize_MidLayer6) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 6) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer6) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 6) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer6 <= 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 6) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer6) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 6) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer6) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 6) property which is less than 1.778 mm.
PadYSize_MidLayer7 Field

Description
Returns pad objects whose Pad Y Size (Mid Layer 7) property complies with the Query.

That property specifies a pad's Y-Size distance on the MidLayer7 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 7) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 7) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer7 property is only defined for Pad objects.

Syntax
PadYSize_MidLayer7 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_MidLayer7) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_MidLayer7) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_MidLayer7 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 7) property which is equal to 100 current measurement units.
PadYSize_MidLayer7 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 7) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadYSize_MidLayer7) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 7) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_MidLayer7) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 7) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_MidLayer7 <= 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 7) property which is not equal to 50 current measurement units.
AsMils(PadYSize_MidLayer7) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 7) property which is less than or equal to 120 mil.
AsMM(PadYSize_MidLayer7) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 7) property which is less than 1.778 mm.
**PadYSize_MidLayer8 Field**

**Description**
Returns pad objects whose Pad Y Size (Mid Layer 8) property complies with the Query.

That property specifies a pad's Y-Size distance on the MidLayer8 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 8) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 8) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer8 property is only defined for Pad objects.

**Syntax**

PadYSize_MidLayer8 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

- **AsMils** (PadYSize_MidLayer8) : Number {This specifies a distance with a unit of 1 mil.}
- **AsMM** (PadYSize_MidLayer8) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**

PadYSize_MidLayer8 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 8) property which is equal to 100 current measurement units.

PadYSize_MidLayer8 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 8) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadYSize_MidLayer8) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 8) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadYSize_MidLayer8) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 8) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadYSize_MidLayer8 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 8) property which is not equal to 50 current measurement units.

AsMils(PadYSize_MidLayer8) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 8) property which is less than or equal to 120 mil.

AsMM(PadYSize_MidLayer8) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 8) property which is less than 1.778 mm.
**PadYSize_MidLayer9 Field**

**Description**
Returns pad objects whose Pad Y Size (Mid Layer 9) property complies with the Query.

That property specifies a pad's Y-Size distance on the MidLayer9 layer when the pad has a 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.

For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Mid Layer 9) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

For any pad having a 'Top-Middle-Bottom' Padstack Mode property, its Pad Y Size (Mid Layer 9) property is assessed as being equal to its Pad Y Size (Mid Layer 1) property, which specifies its Y-Size distance on all internal (or "Middle") Signal layers. (It is not possible for a pad to have a 'Top-Middle-Bottom' Padstack Mode property unless it also has a 'Multi-Layer' Layer property; pads on all remaining layers can only have a 'Simple' Padstack Mode property.)

Note: The PadYSize_MidLayer9 property is only defined for Pad objects.

**Syntax**

PadYSize_MidLayer9 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(PadYSize_MidLayer9) : Number {This specifies a distance with a unit of 1 mil.}

AsMM(PadYSize_MidLayer9) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**

PadYSize_MidLayer9 = 100
Returns all Pad objects that have a Pad Y Size (Mid Layer 9) property which is equal to 100 current measurement units.

PadYSize_MidLayer9 >= 4.064
Returns all Pad objects that have a Pad Y Size (Mid Layer 9) property which is greater than or equal to 4.064 current measurement units.

AsMils(PadYSize_MidLayer9) > 80
Returns all Pad objects that have a Pad Y Size (Mid Layer 9) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PadYSize_MidLayer9) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Mid Layer 9) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

PadYSize_MidLayer9 <> 50
Returns all Pad objects that have a Pad Y Size (Mid Layer 9) property which is not equal to 50 current measurement units.

AsMils(PadYSize_MidLayer9) <= 120
Returns all Pad objects that have a Pad Y Size (Mid Layer 9) property which is less than or equal to 120 mil.

AsMM(PadYSize_MidLayer9) < 1.778
Returns all Pad objects that have a Pad Y Size (Mid Layer 9) property which is less than 1.778 mm.
PadYSize_TopLayer Field

Description
Returns pad objects whose Pad Y Size (Top Layer) property complies with the Query.
That property specifies a pad's Y-size distance on the Top Signal layer when the pad has a 'Top-Middle-Bottom' or 'Full Stack' Padstack Mode property (and by implication, a 'Multi-Layer' Layer property). However, for any Queries that use this Keyword, that property is assessed for all pads, regardless of their Padstack Mode and Layer properties.
For any pad having a 'Simple' Padstack Mode property, its Pad Y Size (Top Layer) property is assessed as being equal to its Pad Y Size (All Layers) property. (The Pad Y Size (All Layers) property specifies the pad's Y-Size distance on all Signal layers in the case of pads on the Multi-Layer layer, or the pad's Y-Size distance on the same layer as the pad in the case of pads on any of the other layers.)

Note: The PadYSize_TopLayer property is only defined for Pad objects.

Syntax
PadYSize_TopLayer : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(PadYSize_TopLayer) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PadYSize_TopLayer) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PadYSize_TopLayer = 100
Returns all Pad objects that have a Pad Y Size (Top Layer) property which is equal to 100 current measurement units.
PadYSize_TopLayer >= 4.064
Returns all Pad objects that have a Pad Y Size (Top Layer) property which is greater than or equal to 4.064 current measurement units.
AsMils(PadYSize_TopLayer) > 80
Returns all Pad objects that have a Pad Y Size (Top Layer) property which is greater than 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PadYSize_TopLayer) Between 1.524 And 3.81
Returns all Pad objects that have a Pad Y Size (Top Layer) property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
PadYSize_TopLayer <> 50
Returns all Pad objects that have a Pad Y Size (Top Layer) property which is not equal to 50 current measurement units.
AsMils(PadYSize_TopLayer) <= 120
Returns all Pad objects that have a Pad Y Size (Top Layer) property which is less than or equal to 120 mil.
AsMM(PadYSize_TopLayer) < 1.778
Returns all Pad objects that have a Pad Y Size (Top Layer) property which is less than 1.778 mm.

PasteMaskExpansion Field

Description
Returns pads whose Paste Mask Override property is True, and whose Paste Mask Expansion property complies with the Query.
Each pad's Paste Mask Expansion distance is determined by (Paste Mask Expansion) Design Rules when its Paste Mask Override property is False. However, when a pad's Paste Mask Override property is True, its Paste Mask Expansion distance is then determined by its Paste Mask Expansion property instead (which property is otherwise of a disabled nature).
Regardless of how a pad's Paste Mask Expansion distance is determined, that distance can be defined to be negative, or zero, or positive. As such, it is possible for each pad's Paste Mask Expansion property to be negative, zero, or positive in magnitude.

Note: The PasteMaskExpansion property is only defined for Pad objects (whose Paste Mask Override property is True).

**Syntax**

PasteMaskExpansion : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(PasteMaskExpansion) : Number {This specifies a distance with a unit of 1 mil.}

AsMM(PasteMaskExpansion) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**

PasteMaskExpansion = 4

Returns all Pad objects that have a True Paste Mask Override property and a Paste Mask Expansion property which is 4 current measurement units.

PasteMaskExpansion >= 0.0508

Returns all Pad objects that have a True Paste Mask Override property and a Paste Mask Expansion property which is greater than or equal to 0.0508 current measurement units.

AsMils(PasteMaskExpansion) Between 3 And 5

Returns all Pad objects that have a True Paste Mask Override property and a Paste Mask Expansion property which is greater than or equal to 3 mil and less than or equal to 5 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PasteMaskExpansion) < 0.1524

Returns all Pad objects that have a True Paste Mask Override property and a Paste Mask Expansion property which is less than 0.1524 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

**See Also**

PasteMaskOverride Field

**PasteMaskOverride Field**

**Description**

Returns pad objects whose Paste Mask Override property complies with the Query.

Each pad's Paste Mask Override property determines whether its Paste Mask Expansion distance is determined by (Paste Mask Expansion) Design Rules or by its Paste Mask Expansion property; 'False' and 'True' properties select each of those options respectively.

Note: The PasteMaskOverride property is only defined for Pad objects.

**Syntax**

PasteMaskOverride : Boolean_String

Boolean_String must be either 'True' or 'False'.

**Examples**

PasteMaskOverride = 'True'

Returns all Pad objects that have a 'True' Paste Mask Override property.

PasteMaskOverride = 'False'

Returns all Pad objects that have a 'False' Paste Mask Override property.

PasteMaskOverride <> 'True'

Not (PasteMaskOverride = 'True')

Returns all objects except Pad objects that have a 'True' Paste Mask Override property.
PasteMaskOverride <> 'False'

Not (PasteMaskOverride = 'False')

Returns all objects except Pad objects that have a 'False' Paste Mask Override property.

**PolygonArcApproximation Field**

**Description**
Returns polygons whose Arc Approximation property complies with the Query.

**Note:** The child objects of these Polygon objects are not also returned.

**Note:** The PolygonArcApproximation property is only defined for Polygon objects.

**Syntax**

```
PolygonArcApproximation : Number
```

If the PCB has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the PCB has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

- `AsMils(PolygonArcApproximation) : Number {This specifies a distance with a unit of 1 mil.}`
- `AsMM(PolygonArcApproximation) : Number {This specifies a distance with a unit of 1 mm.}`

**Examples**

- `PolygonArcApproximation = 0.0127`
  
  Returns all Polygon objects that have a Arc Approximation property which is 0.0127 current measurement units.

- `PolygonArcApproximation >= 1.27`
  
  Returns all Polygon objects that have a Arc Approximation property which is greater than or equal to 1.27 current measurement units.

- `AsMils(PolygonArcApproximation) Between 0.01 And 0.02`
  
  Returns all Polygon objects that have a Arc Approximation property which is greater than or equal to 0.01 mil and less than or equal to 0.02 mil. (The current value of the PCB's Measurement Unit property has no impact upon this outcome.)

- `AsMM(PolygonArcApproximation) < 0.0127`
  
  Returns all Polygon objects that have a Arc Approximation property which is less than 1.016 mm. (The current value of the PCB's Measurement Unit property has no impact upon this outcome.)

**PolygonGridSize Field**

**Description**
Returns polygons whose Grid Size property complies with the Query.

**Note:** The child objects of these Polygon objects are not also returned.

**Note:** The PolygonGridSize property is only defined for Polygon objects.

**Syntax**

```
PolygonGridSize : Number
```

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

- `AsMils(PolygonGridSize) : Number {This specifies a distance with a unit of 1 mil.}`
- `AsMM(PolygonGridSize) : Number {This specifies a distance with a unit of 1 mm.}`

**Examples**

- `PolygonGridSize = 20`
Returns all Polygon objects that have a Grid Size property which is 20 current measurement units.

```
PolygonGridSize >= 1.27
```

Returns all Polygon objects that have a Grid Size property which is greater than or equal to 1.27 current measurement units.

```
AsMils(PolygonGridSize) Between 10 And 30
```

Returns all Polygon objects that have a Grid Size property which is greater than or equal to 10 mil and less than or equal to 30 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

```
AsMM(PolygonGridSize) < 1.016
```

Returns all Polygon objects that have a Grid Size property which is less than 1.016 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

### PolygonHatchingStyle Field

**Description**

Returns polygons whose Hatching Style property complies with the Query.

Note: The child objects of these Polygon objects are not also returned.

Note: The PolygonHatchingStyle property is only defined for Polygon objects.

**Syntax**

```
PolygonHatchingStyle = PolygonHatchingStyle_String
PolygonHatchingStyle <> PolygonHatchingStyle_String
```

PolygonHatchingStyle_String must be one of the strings from the following list:

'45-Degree Hatch', '90-Degree Hatch', 'Horizontal Hatch', 'No Hatching', 'Vertical Hatch'

Note: The single quote characters (') shown at the start and end of each PolygonHatchingStyle_String are both mandatory.

**Examples**

```
PolygonHatchingStyle = 'Horizontal Hatch'
```

Returns all polygons that have a 'Horizontal Hatch' Hatching Style property.

```
PolygonHatchingStyle <> '45-Degree Hatch'
```

Returns all objects except polygon objects that have a '45-Degree Hatch' Hatching Style property. (Only polygon objects have a Hatching Style property, so all remaining types of objects do not have a '45-Degree Hatch' PolygonHatchingStyle property, and are thus also returned by this Query.)

```
IsPolygon && PolygonHatchingStyle <> '45-Degree Hatch'
```

Returns all polygon objects that do not have a '45-Degree Hatch' Hatching Style property.
PolygonIgnoreViolations Field

Description
Returns all polygon objects having a PolygonIgnoreViolations property that complies with the Query. This field needs to be used in conjunction with the IsPolygon statement.

Syntax

```
PolygonIgnoreViolations = Boolean
PolygonIgnoreViolations <> Boolean
```

Examples

```
IsPolygon and (PolygonIgnoreViolations = False)
Returns all polygon objects that violate the PCB design rules. Those polygons that have the Ignore On-Line Violations enabled are not returned.

IsPolygon and (PolygonIgnoreViolations <> False)
Returns all polygon objects that have the Ignore On-Line Violations property enabled except those polygons that violate the PCB design rules.
```

PolygonIslandAreaThreshold Field

Description
Returns polygons whose Island Area Threshold property complies with the Query.

Note: The child objects of these Polygon objects are not also returned.

Note: The PolygonIslandAreaThreshold property is only defined for Polygon objects.

Syntax

```
PolygonIslandAreaThreshold : Number
```

If the PCB has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the PCB has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

```
AsMils(PolygonIslandAreaThreshold ) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PolygonIslandAreaThreshold ) : Number {This specifies a distance with a unit of 1 mm.}
```

Examples

```
PolygonIslandAreaThreshold = 2
Returns all Polygon objects that have an Island area property which is 2 current square measurement units.

PolygonIslandAreaThreshold >= 2
Returns all Polygon objects that have an Island area property which is greater than or equal to 2 current square measurement units.

AsMils(PolygonIslandAreaThreshold ) Between 3000 And 4000
Returns all Polygon objects that have an Island area property which is greater than or equal to 3000 mil and less than or equal to 4000 square mils. (The current value of the PCB's Measurement Unit property has no impact upon this outcome.)

AsMM(PolygonIslandAreaThreshold ) < 1.016
Returns all Polygon objects that have an Island area property which is less than 1.016 square mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
```

PolygonMinimumPrimitiveSize Field

Description
Returns polygons whose Minimum Primitive Size property complies with the Query.

Note: The child objects of these Polygon objects are not also returned.
Note: The Polygon Minimum Primitive Size property is only defined for Polygon objects.

**Syntax**

PolygonMinimumPrimitiveSize : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(PolygonMinimumPrimitiveSize) : Number {This specifies a distance with a unit of 1 mil.}

AsMM(PolygonMinimumPrimitiveSize) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**

PolygonMinimumPrimitiveSize = 4

Returns all Polygon objects that have a Minimum Primitive Size property which is 4 current measurement units.

PolygonMinimumPrimitiveSize >= 0.0508

Returns all Polygon objects that have a Minimum Primitive Size property which is greater than or equal to 0.0508 current measurement units.

AsMils(PolygonMinimumPrimitiveSize) Between 3 And 5

Returns all Polygon objects that have a Minimum Primitive Size property which is greater than or equal to 3 mil and less than or equal to 5 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(PolygonMinimumPrimitiveSize) < 0.1524

Returns all Polygon objects that have a Minimum Primitive Size property which is less than 0.1524 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

**PolygonNeckWidthThreshold Field**

**Description**

Returns polygons whose Neck Width Threshold property complies with the Query.

Note: The child objects of these Polygon objects are not also returned.

Note: The PolygonNeckWidthThreshold property is only defined for Polygon objects.

**Syntax**

PolygonNeckWidthThreshold : Number

If the PCB has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the PCB has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(PolygonNeckWidthThreshold) : Number {This specifies a distance with a unit of 1 mil.}

AsMM(PolygonNeckWidthThreshold) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**

PolygonNeckWidthThreshold= 5

Returns all Polygon objects that have a neck width threshold property which is 5 current measurement units.

PolygonNeckWidthThreshold>= 5

Returns all Polygon objects that have a neck width threshold property which is greater than or equal to 5 current measurement units.

AsMils(PolygonNeckWidthThreshold) Between 5 And 10

Returns all Polygon objects that have a neck width threshold property which is greater than or equal to 5 mil and less than or equal to 10 square mils. (The current value of the PCB's Measurement Unit property has no impact upon this outcome.)

AsMM(PolygonNeckWidthThreshold) < 0.127
Returns all Polygon objects that have a neck width threshold property which is less than 0.127 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

**PolygonPourOverSameNet Field**

**Description**
Returns Polygon objects whose Pour Over Same Net property complies with the Query.

Note: The child objects of these Polygon objects are not also returned.

Note: The PolygonPourOverSameNet property is only defined for Polygon objects.

**Syntax**
PolygonPourOverSameNet : Boolean_String

Boolean_String must be either 'True' or 'False'.

**Examples**
PolygonPourOverSameNet = 'True'
Returns all Polygon objects that have a 'True' Pour Over Same Net property.
PolygonPourOverSameNet = 'False'
Returns all Polygon objects that have a 'False' Pour Over Same Net property.
PolygonPourOverSameNet <> 'True'
Not (PolygonPourOverSameNet = 'True')
Returns all objects except Polygon objects that have a 'True' Pour Over Same Net property.
PolygonPourOverSameNet <> 'False'
Not (PolygonPourOverSameNet = 'False')
Returns all objects except Polygon objects that have a 'False' Pour Over Same Net property.

**PolygonRemoveDeadCopper Field**

**Description**
Returns polygons whose Remove Dead Copper property complies with the Query.

Note: The child objects of these Polygon objects are not also returned.

Note: The PolygonRemoveDeadCopper property is only defined for Polygon objects.

**Syntax**
PolygonRemoveDeadCopper : Boolean_String

Boolean_String must be either 'True' or 'False'.

**Examples**
PolygonRemoveDeadCopper = 'True'
Returns all Polygon objects that have a 'True' Remove Dead Copper property.
PolygonRemoveDeadCopper = 'False'
Returns all Polygon objects that have a 'False' Remove Dead Copper property.
PolygonRemoveDeadCopper <> 'True'
Not (PolygonRemoveDeadCopper = 'True')
Returns all objects except Polygon objects that have a 'True' Remove Dead Copper property.
PolygonRemoveDeadCopper <> 'False'
Not (PolygonRemoveDeadCopper = 'False')
Returns all objects except Polygon objects that have a 'False' Remove Dead Copper property.
**PolygonRemoveIslandsByArea Field**

**Description**
Returns Polygon objects whose Remove Islands By Area property complies with the Query.

*Note:* The child objects of these Polygon objects are not also returned.

*Note:* The PolygonPourOverSameNet property is only defined for Polygon objects.

**Syntax**

```plaintext
PolygonRemoveIslandsByArea : Boolean_String
```

**Boolean_String** must be either 'True' or 'False'.

**Examples**

```plaintext
PolygonRemoveIslandsByArea= 'True'
Returns all Polygon objects that have a 'True' PolygonRemoveIslandsByArea property.
```

```plaintext
PolygonRemoveIslandsByArea= 'False'
Returns all Polygon objects that have a 'False' PolygonRemoveIslandsByArea property.
```

```plaintext
PolygonRemoveIslandsByArea<> 'True'
Not (PolygonRemoveIslandsByArea= 'True')
Returns all objects except Polygon objects that have a 'True' PolygonRemoveIslandsByArea property.
```

```plaintext
PolygonRemoveIslandsByArea<> 'False'
Not (PolygonRemoveIslandsByArea= 'False')
Returns all objects except Polygon objects that have a 'False' PolygonRemoveIslandsByArea property.
```

---

**PolygonRemoveNarrowNecks Field**

**Description**
Returns Polygon objects whose Remove Narrow Necks property complies with the Query.

*Note:* The child objects of these Polygon objects are not also returned.

*Note:* The PolygonRemoveNarrowNecks property is only defined for Polygon objects.

**Syntax**

```plaintext
PolygonRemoveNarrowNecks : Boolean_String
```

**Boolean_String** must be either 'True' or 'False'.

**Examples**

```plaintext
PolygonRemoveNarrowNecks = 'True'
Returns all Polygon objects that have a 'True' PolygonRemoveNarrowNecks property.
```

```plaintext
PolygonRemoveNarrowNecks = 'False'
Returns all Polygon objects that have a 'False' PolygonRemoveNarrowNecks property.
```

```plaintext
PolygonRemoveNarrowNecks <> 'True'
Not (PolygonRemoveNarrowNecks = 'True')
Returns all objects except Polygon objects that have a 'True' PolygonRemoveNarrowNecks property.
```

```plaintext
PolygonRemoveNarrowNecks <> 'False'
Not (PolygonRemoveNarrowNecks = 'False')
Returns all objects except Polygon objects that have a 'False' PolygonRemoveNarrowNecks property.
```
PolygonSurroundPads Field

Description
Returns polygons whose Surround Pads With property complies with the Query.
Note: The child objects of these Polygon objects are not also returned.
Note: The PolygonSurroundPads property is only defined for Polygon objects.

Syntax
PolygonSurroundPads = PolygonSurroundPads_String
PolygonSurroundPads <> PolygonSurroundPads_String

PolygonSurroundPads_String must be one of the strings from the following list:
'Arcs', 'Octagons'
Note: The single quote characters (') shown at the start and end of each PolygonSurroundPads_String are both mandatory.

Examples
PolygonSurroundPads = 'Arcs'
Returns all polygon objects that have an 'Arcs' Surround Pads With property.
PolygonSurroundPads <> 'Octagons'
Returns all objects except polygon objects that have an 'Octagons' Surround Pads With property. (Only polygon objects have a Surround Pads With property, so all remaining types of objects do not have an 'Octagons' PolygonSurroundPads property, and are thus also returned by this Query.)
IsPolygon && PolygonSurroundPads <> 'Octagons'
Returns all polygon objects that do not have an 'Octagons' Surround Pads With property.

PolygonTrackWidth Field

Description
Returns polygons whose Track Width property complies with the Query.
Note: The child objects of these Polygon objects are not also returned.
Note: The PolygonTrackWidth property is only defined for Polygon objects.

Syntax
PolygonTrackWidth : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(PolygonTrackWidth) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(PolygonTrackWidth) : Number {This specifies a distance with a unit of 1 mm.}

Examples
PolygonTrackWidth = 20
Returns all Polygon objects that have a (Polygon) Track Width property which is 20 current measurement units.
PolygonTrackWidth >= 1.27
Returns all Polygon objects that have a (Polygon) Track Width property which is greater than or equal to 1.27 current measurement units.
AsMils(PolygonTrackWidth) Between 10 And 30
Returns all Polygon objects that have a (Polygon) Track Width property which is greater than or equal to 10 mil and less than or equal to 30 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(PolygonTrackWidth) < 1.016
Returns all Polygon objects that have a (Polygon) Track Width property which is less than 1.016 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

**PositiveNet Field**

**Description**
Returns Arc, Fill, Pad, Track, Via, From To, and Polygon objects that are associated with differential pairs whose PositiveNet property complies with the Query.

Note: The Net property is only defined for Arc, Fill, Pad, Track, Via, From To, and Polygon objects.
Note: The associated Net object (whose Name property matches the Net specified, and which is a parent object of the returned objects,) is not also returned.

**Syntax**

PositiveNet : Boolean_String

Boolean_String must be either 'True' or 'False'.

**Examples**

PositiveNet = 'True'

Returns all objects that have a 'True' PositiveNet property.

PositiveNet = 'False'

Returns all objects that have a 'False' PositiveNet property.

PositiveNet <> 'True'

Not (PositiveNet = 'True')

Returns all objects except objects that have a 'True' PositiveNet property.

PositiveNet <> 'False'

Not (PositiveNet = 'False')

Returns all objects except objects that have a 'False' PositiveNet property.

**RadialDimensionAngularStep Field**

**Description**
Returns all Radial Diameter Dimension and Radial Dimension objects having an Angular Step property that complies with the Query.

Note: The child objects of these Radial Diameter Dimension and Radial Dimension objects are not also returned.

Note: The RadialDimensionAngularStep property is only defined for Radial Diameter Dimension and Radial Dimension objects.

**Syntax**

RadialDimensionAngularStep : Number

The number returned specifies the Angular Step in units of degrees.

**Example**

RadialDimensionAngularStep = 45

Returns all Radial Diameter Dimension and Radial Dimension objects that have an Angular Step property of 45 degrees.
Rotation Field

Description
Returns all Fill, Pad, Text, and Component objects whose Rotation property complies with the Query.
Note: The Rotation property is only defined for Fill, Pad, Text, and Component objects.

Syntax
Rotation : Number

The number returned specifies the Rotation in units of degrees.

Examples
Rotation Between 45 And 135
Returns all Fill, Pad, Text, and Component objects that have a Rotation property which is greater than or equal to 45 degrees and less than or equal to 135 degrees.
Rotation >= 90
Returns all Fill, Pad, Text, and Component objects that have a Rotation property which is greater than or equal to 90 degrees.
Rotation < 45
Returns all Fill, Pad, Text, and Component objects that have a Rotation property which is less than 45 degrees. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
Rotation Between 315 And 360 || Rotation Between 0 And 45
Returns all Fill, Pad, Text, and Component objects that have a Rotation property which is between 315 degrees and 45 degrees.

RowCount Attribute Check

Description
Returns Component objects whose count of rows of pads complies with the Query; all of their child objects are also returned.

Syntax
RowCount : Number

Examples
RowCount = 8
Returns all Components that have exactly 8 rows of pads; all of their child objects are also returned.
RowCount Between 6 And 10
Returns all Components that have between 6 and 10 rows of pads; all of their child objects are also returned.
RowCount <=4
Returns all Components that have up to 4 rows of pads; all of their child objects are also returned. This Query also returns all objects which are not child objects of any Component object. (Such objects do not have an explicit RowCount property, so that is assessed as being zero, which is less than 4.)

See Also
BigColumnCount
BigRowCount
ColumnCount

RuleAttributes Field

Description
Returns all Design Rule and Violation objects having a Rule Attributes property that complies with the Query.
Note: The RuleAttributes property is only defined for Design Rule and Violation objects.

Syntax
RuleAttributes : String

Examples
RuleAttributes = 'No Attributes'
Returns all Design Rule and Violation objects that have a Rule Attributes property of 'No Attributes'.
RuleAttributes = 'Clearance = 10mil'
Returns all Design Rule and Violation objects that have a Rule Attributes property of 'Clearance = 10mil'.

RuleCategory Field

Description
Returns all Design Rule objects having a Rule Category property that complies with the Query.
Note: The RuleCategory property is only defined for Design Rule objects.

Syntax
RuleCategory = RuleCategory_String
RuleCategory <> RuleCategory_String

RuleCategory_String must be one of the strings from the following list:
'Electrical', 'High Speed', 'Manufacturing', 'Mask', 'Placement', 'Plane', 'Routing', 'Signal Integrity', 'SMT', 'Testpoint'
Note: The single quote characters (') shown at the start and end of each RuleCategory_String are both mandatory.

Examples
RuleCategory = 'High Speed'
Returns all Design Rule objects that have a 'High Speed' Rule Category property.
RuleCategory <> 'Signal Integrity'
Returns all objects except Design Rule objects that have a 'Signal Integrity' Rule Category property. (Only Design Rule objects have a Rule Category property, so all remaining types of objects do not have a 'Signal Integrity' RuleCategory property, and are thus also returned by this Query.)
IsRule && RuleCategory <> 'Signal Integrity'
Returns all Design Rule objects that do not have a 'Signal Integrity' Rule Category property.

RuleComment Field

Description
Returns all Design Rule objects having a Rule Comment property that complies with the Query.
Note: The RuleComment property is only defined for Design Rule objects.

Syntax
RuleComment : String

Examples
RuleComment = 'All fiducials masked on Paste Mask layers'
Returns all Design Rule objects that have a Rule Comment property of 'All fiducials masked on Paste Mask layers'.
RuleComment = ''
Returns all objects except Design Rule objects that have a Rule Comment property that is not ''. (Only Design Rule objects have a Rule Comment property, so all remaining types of objects are assessed as having a RuleComment property of '', and are thus also returned by this Query.)
RuleEnabled Field

Description
Returns all Design Rule objects having a Rule Enabled property that complies with the Query.
Note: The RuleEnabled property is only defined for Design Rule objects.

Syntax
RuleEnabled : Boolean_String
Boolean_String must be either 'True' or 'False'.

Examples
RuleEnabled = 'True'
Returns all Design Rule objects that have a 'True' Rule Enabled property.
RuleEnabled = 'False'
Returns all Design Rule objects that have a 'False' Rule Enabled property.
RuleEnabled <> 'True'
Not (RuleEnabled = 'True')
Returns all objects except Design Rule objects that have a 'True' Rule Enabled property.
RuleEnabled <> 'False'
Not (RuleEnabled = 'False')
Returns all objects except Design Rule objects that have a 'False' Rule Enabled property.

RulePriority Field

Description
Returns all Design Rule objects having a Rule Priority property that complies with the Query.
Note: The RulePriority property is only defined for Design Rule objects.

Syntax
RulePriority : Numeric_String

The RulePriority keyword can always be used in conjunction with a Numeric_String; it can alternatively be used in conjunction with a Number, but only when all non-Design Rule objects (which do not have a Rule Priority property) have been totally excluded by one or more preceding tokens within the Query. Examples of thus-compliant tokens are as follows:
RulePriority <> ''
RulePriority > ''
ObjectKind Like '* Rule'
IsRule

The usage of such tokens is demonstrated in the examples that follow.

Examples
RulePriority = '1'
RulePriority <> '' && RulePriority = 1
Returns all Design Rule objects that have a Rule Priority property of '1'.
RulePriority > '' && RulePriority Between 2 And 9
Returns all Design Rule objects that have a Rule Priority property which is greater than or equal to 2 and less than or equal to 9.
ObjectKind Like '* Rule' && RulePriority >= 4
Returns all Design Rule objects that have a Rule Priority property which is greater than or equal to 4.
IsRule && RulePriority < 10
Returns all Design Rule objects that have a Rule Priority property which is less than 10.
RulePriority <> '1'
Returns all objects except Design Rule objects that have a Rule Priority property of 1. (Only Design Rule objects have a Rule Priority property, so all remaining types of objects do not have a RulePriority property of 1, and are thus also returned by this Query.)

IsRule && RulePriority <> 1
Returns all Design Rule objects that do not have a Rule Priority property of 1.

**RuleScope1 Field**

**Description**
Returns all Design Rule and Violation objects having a Scope 1 property that complies with the Query.

**Note:** The RuleScope1 property is only defined for Design Rule and Violation objects.

**Syntax**
RuleScope1 : String

**Examples**
RuleScope1 = 'All'
Returns all Design Rule and Violation objects that have a Scope 1 property of **All**. (The Scope 1 property is itself a Query, and as that is a string, it needs to be surrounded by single quote characters (') when it is specified within another Query.)

RuleScope1 = 'InNet(''D7'')'
Returns all Design Rule and Violation objects that have a Scope 1 property of **InNet('D7')**. (The name of the net (D7) is surrounded by single quote characters within the Query specifying the Scope 1 property, and thus needs to be surrounded by **two** single quote characters when that Query is specified within another Query.)

RuleScope1 Like 'InNet(''???'')'
Returns all Design Rule and Violation objects having a Scope 1 property of **InNet('???)**; the string of the associated net needs to consist of exactly three characters, with examples of compliant Scope 1 properties being **InNet('GND')** and **InNet('VCC')**.

**RuleScope2 Field**

**Description**
Returns all Design Rule and Violation objects having a Scope 2 property that complies with the Query.

**Note:** The RuleScope2 property is only defined for Design Rule and Violation objects.

**Syntax**
RuleScope2 : String

**Examples**
RuleScope2 = 'All'
Returns all Design Rule and Violation objects that have a Scope 2 property of **All**. (The Scope 2 property is itself a Query, and as that is a string, it needs to be surrounded by single quote characters (') when it is specified within another Query.)

RuleScope2 = '-'
Returns all Design Rule and Violation objects that have no Scope 2 property. (A few types of Design Rules have a Scope 1 and Scope 2 property, but most types of Design Rules have just a Scope 1 property, and thus a Scope 2 property cannot be defined for such Design Rules.)

RuleScope2 = 'InComponent(''P1'')'
Returns all Design Rule and Violation objects that have a Scope 2 property of **InComponent('P1')**. (The name of the component (P1) is surrounded by single quote characters within the Query specifying the Scope 1 property, and thus needs to be surrounded by **two** single quote characters when that Query is specified within another Query.)

RuleScope2 Like 'InComponent(''??*'')'
Returns all Design Rule and Violation objects that have a Scope 2 property of `InComponent('??*')`; the string of the associated component needs to consist of at least two characters, with examples of compliant Scope 2 properties being `InComponent('C3')` and `InComponent('RP1')`.

**RuleType Field**

**Description**
Returns all Design Rule objects having a Rule Type property that complies with the Query.

Note: The RuleType property is only defined for Design Rule objects.

**Syntax**

```
RuleType = RuleType_String
RuleType <> RuleType_String
```

*RuleType_String* must be one of the strings from the following list:

- 'Acute Angle'
- 'Clearance'
- 'Component Clearance'
- 'Component Orientations'
- 'Daisy Chain Stub Length'
- 'Fanout Control'
- 'Flight Time - Falling Edge'
- 'Flight Time - Rising Edge'
- 'Height'
- 'Hole Size'
- 'Impedance'
- 'Layer Pairs'
- 'Length'
- 'Matched Net Lengths'
- 'Maximum Via Count'
- 'Minimum Annular Ring'
- 'Nets To Ignore'
- 'Overshoot - Falling Edge'
- 'Overshoot - Rising Edge'
- 'Parallel Segment'
- 'Paste Mask Expansion'
- 'Permitted Layers'
- 'Polygon Connect Style'
- 'Power Plane Clearance'
- 'Power Plane Connect Style'
- 'Room Definition'
- 'Routing Corners'
- 'Routing Layers'
- 'Routing Priority'
- 'Routing Topology'
- 'Routing Via Style'
- 'Short-Circuit'
- 'Signal Base Value'
- 'Signal Stimulus'
- 'Signal Top Value'
- 'Slope - Falling Edge'
- 'Slope - Rising Edge'
- 'SMD Neck-Down'
- 'SMD To Corner'
- 'SMD To Plane'
- 'Solder Mask Expansion'
- 'Supply Nets'
- 'Testpoint Style'
- 'Testpoint Usage'
- 'Un-Connected Pin'
- 'Un-Routed Net'
- 'Undershoot - Falling Edge'
- 'Undershoot - Rising Edge'
- 'Vias Under SMD'
- 'Width'

Note: The single quote characters ('') shown at the start and end of each RuleType_String are both mandatory.

**Examples**

```
RuleType = 'Hole Size'
```

Returns all Design Rule objects that have a 'Hole Size' Rule Type property.

```
RuleType <> 'Paste Mask Expansion'
```

Returns all objects except Design Rule objects that have a 'Paste Mask Expansion' Rule Type property. (Only Design Rule objects have a Rule Type property, so all remaining types of objects do not have a 'Paste Mask Expansion' RuleType property, and are thus also returned by this Query.)

```
IsRule && RuleType <> 'Paste Mask Expansion'
```

Returns all Design Rule objects that do not have a 'Paste Mask Expansion' Rule Type property.

**ShowComment Field**

**Description**
Returns components whose Show Comment property complies with the Query.

Note: The child objects of these Component objects are not also returned.

Note: The ShowComment property is only defined for Component objects.

**Syntax**

```
ShowComment : Boolean_String
```

*Boolean_String* must be either 'True' or 'False'.

**Examples**

```
ShowComment = 'True'
```

Returns all Component objects that have a 'True' Show Comment property.

```
ShowComment = 'False'
```

Returns all Component objects that have a 'False' Show Comment property.

```
ShowComment <> 'True'
```

Not (ShowComment = 'True')
Returns all objects except Component objects that have a 'True' Show Comment property.
ShowComment <> 'False'
Not (ShowComment = 'False')

Returns all objects except Component objects that have a 'False' Show Comment property.

**ShowName Field (PCB)**

**Description**
Returns Component objects whose Show Name property complies with the Query.

Note: The child objects of these Component objects are **not** also returned.

Note: The ShowName property is only defined for Component objects.

**Syntax**
ShowName : Boolean_String

Boolean_String must be either 'True' or 'False'.

**Examples**

ShowName = 'True'
Returns all Component objects that have a 'True' Show Name property.

ShowName = 'False'
Returns all Component objects that have a 'False' Show Name property.

ShowName <> 'True'
Not (ShowName = 'True')
Returns all objects except Component objects that have a 'True' Show Name property.

ShowName <> 'False'
Not (ShowName = 'False')
Returns all objects except Component objects that have a 'False' Show Name property.

**SlotHoleLength Field**

**Description**
Returns all pad objects having a SlotHoleLength property that complies with the Query.

**Syntax**

SlotHoleLength = MeasurementValue
SlotHoleLength <> MeasurementValue

MeasurementValue.

If the PCB document has an 'Imperial' Measurement Unit property, the MeasurementValue specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the PCB document has a 'Metric' Measurement Unit property, the MeasurementValue specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(SolderMaskExpansion) : Number (This specifies a distance with a unit of 1 mil.)
AsMM(SolderMaskExpansion) : Number (This specifies a distance with a unit of 1 mm.)

**Examples**

SlotHoleLength = 50
Returns all Pad objects that have a SlotHoleLength property which is 50 current measurement units.

SlotHoleLength >= 50
Returns all Pad objects that have a SlotHoleLength property which greater than or equal to 50 current measurement units.

AsMils(SlotHoleLength) Between 20 And 50
Returns all Pad objects that have a SlotHoleLength property which is greater than or equal to 20 mil and less than or equal to 50 mil units.
\[ \text{AsMM(SlotHoleLength)} < 50 \]

Returns all Pad objects that have a SlotHoleLength property which is less than 50 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

### SolderMaskExpansion Field

**Description**

Returns pads and vias whose Solder Mask Override property is True, and whose Solder Mask Expansion property complies with the Query.

Each pad's and via's Solder Mask Expansion distance is determined by (Solder Mask Expansion) Design Rules when its Solder Mask Override property is False. However, when a pad's or via's Solder Mask Override property is True, its Solder Mask Expansion distance is then determined by its Solder Mask Expansion property instead (which property is otherwise of a disabled nature).

Regardless of how a pad's or via's Solder Mask Expansion distance is determined, that distance can be defined to be negative, or zero, or positive. As such, it is possible for each pad's and via's Solder Mask Expansion property to be negative, zero, or positive in magnitude.

Note: The SolderMaskExpansion property is only defined for Pad and Via objects (whose Solder Mask Override property is True).

**Syntax**

\[ \text{SolderMaskExpansion} : \text{Number} \]

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

\[ \text{AsMils(SolderMaskExpansion)} : \text{Number} \text{ (This specifies a distance with a unit of 1 mil.)} \]

\[ \text{AsMM(SolderMaskExpansion)} : \text{Number} \text{ (This specifies a distance with a unit of 1 mm.)} \]

**Examples**

\[ \text{SolderMaskExpansion} = 4 \]

Returns all Pad and Via objects that have a True Solder Mask Override property and a Solder Mask Expansion property which is 4 current measurement units.

\[ \text{SolderMaskExpansion} \geq 0.0508 \]

Returns all Pad and Via objects that have a True Solder Mask Override property and a Solder Mask Expansion property which is greater than or equal to 0.0508 current measurement units.

\[ \text{AsMils(SolderMaskExpansion)} \text{ Between 3 And 5} \]

Returns all Pad and Via objects that have a True Solder Mask Override property and a Solder Mask Expansion property which is greater than or equal to 3 mil and less than or equal to 5 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

\[ \text{AsMM(SolderMaskExpansion)} < 0.1524 \]

Returns all Pad and Via objects that have a True Solder Mask Override property and a Solder Mask Expansion property which is less than 0.1524 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

**See Also**

SolderMaskOverride Field
**SolderMaskOverride Field**

**Description**

Returns pad and via objects whose Solder Mask Override property complies with the Query.

Each pad's and via's Solder Mask Override property determines whether its Solder Mask Expansion distance is determined by (Solder Mask Expansion) Design Rules or by its Solder Mask Expansion property; 'False' and 'True' properties select each of those options respectively.

Note: The SolderMaskOverride property is only defined for Pad and Via objects.

**Syntax**

SolderMaskOverride : Boolean_String

Boolean_String must be either 'True' or 'False'.

**Examples**

SolderMaskOverride = 'True'

Returns all Pad and Via objects that have a 'True' Solder Mask Override property.

SolderMaskOverride = 'False'

Returns all Pad and Via objects that have a 'False' Solder Mask Override property.

SolderMaskOverride <> 'True'

Not (SolderMaskOverride = 'True')

Returns all objects except Pad and Via objects that have a 'True' Solder Mask Override property.

SolderMaskOverride <> 'False'

Not (SolderMaskOverride = 'False')

Returns all objects except Pad and Via objects that have a 'False' Solder Mask Override property.

---

**SolderMaskTentingBottom Field**

**Description**

Returns pad and via objects whose Solder Mask Tenting - Bottom property complies with the Query.

Each pad's and via's Solder Mask Tenting - Bottom property determines whether an image of it is rendered on the Bottom Solder Mask layer or whether it is "tented" on that layer (meaning that an image of it is not rendered on that layer); False and True properties select each of those options respectively.

Note: The SolderMaskTentingBottom property is only defined for Pad and Via objects.

**Syntax**

SolderMaskTentingBottom : Boolean

**Examples**

SolderMaskTentingBottom

SolderMaskTentingBottom = True

Returns all Pad and Via objects that have a True Solder Mask Tenting - Bottom property.

Not SolderMaskTentingBottom

SolderMaskTentingBottom = False

Returns all objects except Pad and Via objects that have a True Solder Mask Tenting - Bottom property.
SolderMaskTentingTop Field

Description
Returns pad and via objects whose Solder Mask Tenting - Top property complies with the Query. Each pad's and via's Solder Mask Tenting - Top property determines whether an image of it is rendered on the Top Solder Mask layer or whether it is "tented" on that layer (meaning that an image of it is not rendered on that layer); False and True properties select each of those options respectively.

Note: The SolderMaskTentingTop property is only defined for Pad and Via objects.

Syntax
SolderMaskTentingTop : Boolean

Examples
SolderMaskTentingTop
SolderMaskTentingTop = True
Returns all Pad and Via objects that have a True Solder Mask Tenting - Top property.

SolderMaskTentingTop = False
Returns all objects except Pad and Via objects that have a True Solder Mask Tenting - Top property.

StandoffHeight Field

Description
Returns all component objects having a StandoffHeight property that complies with the Query. Note the standoff height is the distance from the board to the underside of the component body.

Syntax
StandOffHeight: Number

If the PCB has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the PCB has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:
AsMils(StandOffHeight): Number {This specifies a distance with a unit of 1 mil.}
AsMM(StandOffHeight): Number {This specifies a distance with a unit of 1 mm.}

Examples
StandOffHeight = 100
Returns all components that have a StandoffHeight property which is equal to 100 current measurement units.
StandOffHeight >= 4.064
Returns all components that have a StandoffHeight property which is greater than or equal to 4.064 current measurement units.
AsMils(StandOffHeight) > 80
Returns all components that have a StandoffHeight property which is greater than 80 mil.
AsMM(StandOffHeight) Between 1.524 And 3.81
Returns all components that have a StandoffHeight property which is greater than or equal to 1.524 mm and less than or equal to 3.81 mm.
StandOffHeight <= 50
Returns all components that have a StandoffHeight property which is not equal to 50 current measurement units.
AsMils(StandOffHeight) <= 120
Returns all components that have a StandoffHeight property which is less than or equal to 120 mil.
AsMM(StandOffHeight) < 1.778
Returns all components that have a `StandOffHeight` property which is less than 1.778 mm.

**StartLayer Field**

**Description**

Returns all Via objects having a Start Layer property that complies with the Query.

Note: The StartLayer property is only defined for Via objects.

**Syntax**

```
StartLayer = Layer_String
StartLayer <> Layer_String
```

*Layer_String* must be one of the strings from the following list:

- 'Top Layer'
- 'MidLayer1'
- 'MidLayer2'
- 'MidLayer3'
- 'MidLayer4'
- 'MidLayer5'
- 'MidLayer6'
- 'MidLayer7'
- 'MidLayer8'
- 'MidLayer9'
- 'MidLayer10'
- 'MidLayer11'
- 'MidLayer12'
- 'MidLayer13'
- 'MidLayer14'
- 'MidLayer15'
- 'MidLayer16'
- 'MidLayer17'
- 'MidLayer18'
- 'MidLayer19'
- 'MidLayer20'
- 'MidLayer21'
- 'MidLayer22'
- 'MidLayer23'
- 'MidLayer24'
- 'MidLayer25'
- 'MidLayer26'
- 'MidLayer27'
- 'MidLayer28'
- 'MidLayer29'
- 'MidLayer30'
- 'Bottom Layer'
- 'InternalPlane1'
- 'InternalPlane2'
- 'InternalPlane3'
- 'InternalPlane4'
- 'InternalPlane5'
- 'InternalPlane6'
- 'InternalPlane7'
- 'InternalPlane8'
- 'InternalPlane9'
- 'InternalPlane10'
- 'InternalPlane11'
- 'InternalPlane12'
- 'InternalPlane13'
- 'InternalPlane14'
- 'InternalPlane15'
- 'InternalPlane16'

Note: The single quote characters (') shown at the start and end of each *Layer_String* are both mandatory.

**Examples**

- **Example 1:**
  ```
  StartLayer = 'Top Layer'
  ```
  Returns all Via objects that have a 'Top Layer' Start Layer property.

- **Example 2:**
  ```
  StartLayer <> 'MidLayer1'
  ```
  Returns all objects except Via objects that have a 'MidLayer1' Start Layer property. (Only Via objects have a Start Layer property, so all remaining types of objects do not have a 'MidLayer1' StartLayer property, and are thus also returned by this Query.)

- **Example 3:**
  ```
  IsVia && StartLayer <> 'MidLayer1'
  ```
  Returns all Via objects that do **not** have a 'MidLayer1' Start Layer property.

**See Also**

**StopLayer Field**

**Description**

Returns all Via objects having a Stop Layer property that complies with the Query.

Note: The StopLayer property is only defined for Via objects.

**Syntax**

```
StopLayer = Layer_String
StopLayer <> Layer_String
```

*Layer_String* must be one of the strings from the following list:

- 'Top Layer'
- 'MidLayer1'
- 'MidLayer2'
- 'MidLayer3'
- 'MidLayer4'
- 'MidLayer5'
- 'MidLayer6'
- 'MidLayer7'
- 'MidLayer8'
- 'MidLayer9'
- 'MidLayer10'
- 'MidLayer11'
- 'MidLayer12'
- 'MidLayer13'
- 'MidLayer14'
- 'MidLayer15'
- 'MidLayer16'
- 'MidLayer17'
- 'MidLayer18'
- 'MidLayer19'
- 'MidLayer20'
- 'MidLayer21'
- 'MidLayer22'
- 'MidLayer23'
- 'MidLayer24'
- 'MidLayer25'
- 'MidLayer26'
- 'MidLayer27'
- 'MidLayer28'
- 'MidLayer29'
- 'MidLayer30'
- 'Bottom Layer'
- 'InternalPlane1'
- 'InternalPlane2'
- 'InternalPlane3'
- 'InternalPlane4'
- 'InternalPlane5'
- 'InternalPlane6'
- 'InternalPlane7'
- 'InternalPlane8'
- 'InternalPlane9'
- 'InternalPlane10'
- 'InternalPlane11'
- 'InternalPlane12'
- 'InternalPlane13'
- 'InternalPlane14'
- 'InternalPlane15'
- 'InternalPlane16'

Note: The single quote characters (') shown at the start and end of each *Layer_String* are both mandatory.

**Examples**

- **Example 1:**
  ```
  StopLayer = 'Bottom Layer'
  ```
Returns all Via objects that have a 'Bottom Layer' Stop Layer property.
StopLayer <> 'InternalPlane1'
Returns all objects except Via objects that have an 'InternalPlane1' Stop Layer property. (Only Via objects have a Stop Layer property, so all remaining types of objects do not have an 'InternalPlane1' StopLayer property, and are thus also returned by this Query.)
IsVia && StopLayer <> 'InternalPlane1'
Returns all Via objects that do not have an 'InternalPlane1' Stop Layer property.

See Also
StartLayer Field

StringText Field (PCB)

Description
Returns Text and Leader Dimension objects whose String property complies with the Query.
Note: The child objects of these Leader Dimension objects are not also returned.
Note: The StringText property is only defined for Text and Leader Dimension objects.

Syntax
StringText : String

Example
StringText = '100nF'
Returns all Text and Leader Dimension objects that have a String property of '100nF'.

StringType Field

Description
Returns all Text objects having a String Type property that complies with the Query.
Note: The StringType property is only defined for Text objects.

Syntax
StringType = StringType_String
StringType <> StringType_String
StringType_String must be one of the strings from the following list:
'Comment', 'Designator', 'Free'
Note: The single quote characters ('') shown at the start and end of each StringType_String are both mandatory.

Examples
StringType = 'Designator'
Returns all Text objects that have a 'Designator' String Type property.
StringType <> 'Free'
Returns all objects except Text objects that have a 'Free' String Type property. (Only Text objects have a String Type property, so all remaining types of objects do not have a 'Free' StringType property, and are thus also returned by this Query.)
StringType = 'Comment' || StringType = 'Designator'
IsString && StringType <> 'Free'
Returns all Text objects that do not have a 'Free' String Type property.
**StrokeFont Field**

**Description**

Returns all Text objects having a StrokeFont property that complies with the Query.

*Note: The StringType property is only defined for Text objects.*

**Syntax**

```
StringType = StringType_String
StringType <> StringType_String
```

*StringType_String must be one of the strings from the following list:
'Comment', 'Designator', 'Free'*

*Note: The single quote characters (') shown at the start and end of each StringType_String are both mandatory.*

**Examples**

```
StringType = 'Designator'
Returns all Text objects that have a 'Designator' String Type property.
```

```
StringType <> 'Free'
Returns all objects except Text objects that have a 'Free' String Type property. (Only Text objects have a String Type property, so all remaining types of objects do not have a 'Free' StringType property, and are thus also returned by this Query.)
```

```
StringType = 'Comment' || StringType = 'Designator'
IsString && StringType <> 'Free'
Returns all Text objects that do not have a 'Free' String Type property.
```

**TestpointBottom Field**

**Description**

Returns pad and via objects whose Testpoint - Bottom property complies with the Query.

*Note: The TestpointBottom property is only defined for Pad and Via objects.*

**Syntax**

```
TestpointBottom : Boolean
```

**Examples**

```
TestpointBottom
TestpointBottom = True
```

*Returns all Pad and Via objects that have a True Testpoint - Bottom property.*

```
Not TestpointBottom
TestpointBottom = False
```

*Returns all objects except Pad and Via objects that have a True Testpoint - Bottom property.*

**TestpointTop Field**

**Description**

Returns pad and via objects whose Testpoint - Top property complies with the Query.

*Note: The TestpointTop property is only defined for Pad and Via objects.*

**Syntax**

```
TestpointTop : Boolean
```

**Examples**

```
TestpointTop
TestpointTop = True
```

*Returns all Pad and Via objects that have a True Testpoint - Top property.*
Not TestpointTop
TestpointTop = False

Returns all objects except Pad and Via objects that have a True Testpoint - Top property.

**TextHeight Field**

**Description**

Returns Text, Coordinate, and Original Dimension objects whose Text Height property complies with the Query.

Note: The TextHeight property is only defined for Text, Coordinate, and Original Dimension objects.

**Syntax**

```
TextHeight : Number
```

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

```
AsMils(TextHeight) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(TextHeight) : Number {This specifies a distance with a unit of 1 mm.}
```

**Examples**

```
TextHeight = 60
Returns all Text, Coordinate, and Original Dimension objects that have a Text Height property which is 60 current measurement units.

TextHeight >= 1.27
Returns all Text, Coordinate, and Original Dimension objects that have a Text Height property which is greater than or equal to 1.27 current measurement units.

AsMils(TextHeight) Between 40 And 80
Returns all Text, Coordinate, and Original Dimension objects that have a Text Height property which is greater than or equal to 40 mil and less than or equal to 80 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(TextHeight) < 1.778
Returns all Text, Coordinate, and Original Dimension objects that have a Text Height property which is less than 1.778 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
```

**TextWidth Field**

**Description**

Returns Text, Coordinate, and Original Dimension objects whose Text Width property complies with the Query.

Note: The TextWidth property is only defined for Text, Coordinate, and Original Dimension objects.

**Syntax**

```
TextWidth : Number
```

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

```
AsMils(TextWidth) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(TextWidth) : Number {This specifies a distance with a unit of 1 mm.}
```

**Examples**

```
TextWidth = 6
```
Returns all Text, Coordinate, and Original Dimension objects that have a Text Width property which is 6 current measurement units.

\[ \text{TextWidth} \geq 0.127 \]

Returns all Text, Coordinate, and Original Dimension objects that have a Text Width property which is greater than or equal to 0.127 current measurement units.

\[ \text{AsMils(TextWidth)} \text{ Between 4 And 8} \]

Returns all Text, Coordinate, and Original Dimension objects that have a Text Width property which is greater than or equal to 4 mil and less than or equal to 8 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

\[ \text{AsMM(TextWidth)} < 0.1778 \]

Returns all Text, Coordinate, and Original Dimension objects that have a Text Width property which is less than 0.1778 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

**TouchesRoom Membership Check**

**Description**

Returns objects that either totally or partially reside within the region occupied by any Room having a Name property that is specified by the Query.

**Note:** The parent objects of group objects (Components, Coordinates, Dimensions, Nets, and Polygons) are not also returned.

**Syntax**

\[ \text{TouchesRoom(Name : String)} : \text{Boolean} \]

\[ \text{TouchesRoom(Name : String , Name : String)} : \text{Boolean} \]

\[ \text{TouchesRoom(Name : String , Name : String , Name : String)} : \text{Boolean} \]

**Examples**

\[ \text{TouchesRoom('Power')} \]

\[ \text{TouchesRoom('Power')} = \text{True} \]

Returns objects that either totally or partially reside within the region occupied by any Room having a Name property of 'Power'.

\[ \text{TouchesRoom('COUT1','COUT2')} \]

\[ \text{TouchesRoom('COUT1','COUT2')} = \text{True} \]

Returns objects that either totally or partially reside within the regions occupied by any Rooms having a Name property of 'COUT1' or 'COUT2'.

\[ \text{TouchesRoom('Monitor','Auxilary','Effects')} \]

\[ \text{TouchesRoom('Monitor','Auxilary','Effects')} = \text{True} \]

Returns objects that either totally or partially reside within the regions occupied by any Rooms having a Name property of 'Monitor' or 'Auxilary' or 'Effects'.

---

Version (v1.7) Apr 21, 2008
TraceCopperHeight Attribute Check

Description
Returns arcs, fills, and tracks on Signal layers whose Copper Thickness property complies with the Query.
Each currently enabled Signal layer has its own Copper Thickness property, which can be viewed and edited from a child ('Edit Layer') dialog box invoked from the 'Layer Stack Manager' dialog box.
Note: The objects returned do not include any of the arcs, fills, and tracks on the MultiLayer layer. (Because different Signal layers can have different Copper Thickness properties, a Copper Thickness property for the MultiLayer layer is not always meaningful.)

At present, all distances always use units of mils. (1 mil = 0.001 inch)
Note: This is an alias for TraceHeight (Attribute Check).

Syntax
TraceCopperHeight : Distance_String

Examples
TraceCopperHeight = 1.4
Returns all Arc, Fill, and Track objects on all Signal layers that have a Copper Thickness property which is equal to 1.4 mil.
TraceCopperHeight Between 1.1 And 1.5
Returns all Arc, Fill, and Track objects on all Signal layers that have a Copper Thickness property which is greater than or equal to 1.1 mil and less than or equal to 1.5 mil.
TraceCopperHeight >= 1.2
Returns all Arc, Fill, and Track objects on all Signal layers that have a Copper Thickness property which is greater than or equal to 1.2 mil.
TraceCopperHeight < 1.3
Returns all objects except Arc, Fill, and Track objects on Signal layers that have a Copper Thickness property which is greater than or equal to 1.3 mil. (Only Arc, Fill, and Track objects on Signal layers have a TraceCopperHeight property, so all remaining types of objects are assessed as having a TraceCopperHeight property of 0, and are thus also returned by this Query.)
TraceCopperHeight = 2.8 || (IsArc Or IsFill Or IsTrack && OnMultiLayer)
Returns all Arc, Fill, and Track objects on all Signal layers that have a Copper Thickness property which is equal to 2.8 mil, and all Arc, Fill, and Track objects on the MultiLayer layer.

TraceHeight Impedance

Description
Specifies the height of the copper on a Signal layer within a multi-layer PCB. That distance has an influence upon the characteristic impedance between a track on that layer and the copper on an (adjacent) Internal Plane layer (or the copper on (adjacent) surrounding Internal Plane layers), or the width of such a track when a particular characteristic impedance is required.

Syntax
TraceHeight : Number
The distance is specified in units of mils. (1 mil = 0.001 inch)

Examples
(87/SQRT(Er+1.41))*LN(5.98*TraceToPlaneDistance/(0.8*TraceWidth + TraceHeight))
The default Query for specifying the characteristic impedance between a track on the Signal layer and the copper on the (adjacent) Internal Plane layer, when the Microstrip option is selected.
(60/SQRT(Er))*LN((1.9*PlaneToPlaneDistance)/(0.8*TraceWidth + TraceHeight))
The default Query for specifying the characteristic impedance between a track on the intermediate Signal layer and the copper on the two surrounding Internal Plane layers, when the Stripline option is selected.
TraceHeight Attribute Check

Description
Returns arcs, fills, and tracks on Signal layers whose Copper Thickness property complies with the Query.

Each currently enabled Signal layer has its own Copper Thickness property, which can be viewed and edited from a child ('Edit Layer') dialog box invoked from the 'Layer Stack Manager' dialog box.

Note: The objects returned do not include any of the arcs, fills, and tracks on the MultiLayer layer. Because different Signal layers can have different Copper Thickness properties, a Copper Thickness property for the MultiLayer layer is not always meaningful.

At present, all distances always use units of mils. (1 mil = 0.001 inch)

Note: This is an alias for TraceCopperHeight (Attribute Check).

Syntax
TraceHeight : Distance_String

Examples
TraceHeight = 1.4
Returns all Arc, Fill, and Track objects on all Signal layers that have a Copper Thickness property which is equal to 1.4 mil.

TraceHeight Between 1.1 And 1.5
Returns all Arc, Fill, and Track objects on all Signal layers that have a Copper Thickness property which is greater than or equal to 1.1 mil and less than or equal to 1.5 mil.

TraceHeight >= 1.2
Returns all Arc, Fill, and Track objects on all Signal layers that have a Copper Thickness property which is greater than or equal to 1.2 mil.

TraceHeight < 1.3
Returns all objects except Arc, Fill, and Track objects on Signal layers that have a Copper Thickness property which is greater than or equal to 1.3 mil. (Only Arc, Fill, and Track objects on Signal layers have a TraceHeight property, so all remaining types of objects are assessed as having a TraceHeight property of 0, and are thus also returned by this Query.)

TraceHeight = 2.8 || (IsArc Or IsFill Or IsTrack && OnMultiLayer)
Returns all Arc, Fill, and Track objects on all Signal layers that have a Copper Thickness property which is equal to 2.8 mil, and all Arc, Fill, and Track objects on the MultiLayer layer.

TraceLength Attribute Check

Description
Returns arc and track objects on Signal layers whose length complies with the Query.

The length of each track is the (Pythagorean) distance between its end points, and is thus equal to SQRT(SQR(X1 - X2) + SQR(Y1 - Y2)), where X1 and Y1 are the coordinates of one end of the track, while X2 and Y2 are the coordinates of its other end.

The length of each arc is the product of its Radius and the difference between its End Angle and Start Angle, when that difference is measured in units of radians. (180 degrees = pi radians.)

At present, all distances always use mils unit. (1 mil = 0.001 inch)

Syntax
TraceLength : Distance_String

Examples
TraceLength Between 400 And 1800
Returns arcs and tracks having a length which is between 400 mil and 1800 mil.

TraceLength > 300
Returns arcs and tracks having a length which is greater than 300 mil.
TraceToPlaneDistance Impedance

Description
Specifies the copper-to-copper distance between a track on a Signal layer and an (adjacent) Internal Plane layer within a multi-layer PCB. That distance has an influence upon the characteristic impedance between that track and the copper on that Internal Plane layer, or the width of that track when a particular characteristic impedance is required.

Syntax
TraceToPlaneDistance : Distance_String

The distance is specified in units of mils. (1 mil = 0.001 inch)

Examples

(87/SQRT(Er+1.41))*LN(5.98*TraceToPlaneDistance/(0.8*TraceWidth + TraceHeight))

The default Query for specifying the characteristic impedance between a track on the Signal layer and the copper on the (adjacent) Internal Plane layer, when the Microstrip option is selected.

((5.98*TraceToPlaneDistance/EXP(CharacteristicImpedance*SQR(Er+1.41)/87))- TraceHeight)*1.25

The default Query for specifying the required width of a track on the Signal layer, when the Microstrip option is selected.

TraceWidth Impedance

Description
Specifies the width of a track on a Signal layer within a multi-layer PCB. That distance has an influence upon the characteristic impedance between that track and the copper on an (adjacent) Internal Plane layer (or the copper on (adjacent) surrounding Internal Plane layers).

Syntax
TraceWidth : Number

The distance is specified in units of mils. (1 mil = 0.001 inch)

Examples

(87/SQRT(Er+1.41))*LN(5.98*TraceToPlaneDistance/(0.8*TraceWidth + TraceHeight))

The default Query for specifying the characteristic impedance between a track on the Signal layer and the copper on the (adjacent) Internal Plane layer, when the Microstrip option is selected.

(60/SQRT(Er))*LN((1.9*PlaneToPlaneDistance)/(0.8*TraceWidth + TraceHeight))

The default Query for specifying the characteristic impedance between a track on the intermediate Signal layer and the copper on the two surrounding Internal Plane layers, when the Stripline option is selected.

UseTrueTypeFonts Field

Description
Returns Text, Coordinate, and Original Dimension objects whose UseTrueTypeFonts property complies with the Query.

Note: The UseTrueTypeFonts property is only defined for Text, Coordinate, and Dimension objects.

Syntax
UseTrueTypeFonts : Boolean_String

Boolean_String must be either 'True' or 'False'.

Examples

UseTrueTypeFonts = 'True'

Returns all Coordinate, Original Dimension, and Text objects that are using the True Type fonts.

UseTrueTypeFonts <> 'True'

Not (UseTrueTypeFonts = 'True')

Returns all Coordinate, Dimension, and Text objects that are not using true type fonts.
**ViaDiameter Field**

**Description**
Returns vias whose Via Diameter property complies with the Query.

Note: The ViaDiameter property is only defined for Via objects.

Note: This is an alias for ViaSize (Field).

**Syntax**

`ViaDiameter : Number`

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

- `AsMils(ViaDiameter) : Number` {This specifies a distance with a unit of 1 mil.}
- `AsMM(ViaDiameter) : Number` {This specifies a distance with a unit of 1 mm.}

**Examples**

- `ViaDiameter = 40`
  Returns all Via objects that have a Via Diameter property which is 40 current measurement units.

- `ViaDiameter >= 0.508`
  Returns all Via objects that have a Via Diameter property which is greater than or equal to 0.508 current measurement units.

- `AsMils(ViaDiameter) Between 30 And 50`
  Returns all Via objects that have a Via Diameter property which is greater than or equal to 30 mil and less than or equal to 50 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

- `AsMM(ViaDiameter) < 1.524`
  Returns all Via objects that have a Via Diameter property which is less than 1.524 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

**ViaSize Field**

**Description**
Returns vias whose Via Diameter property complies with the Query.

Note: The ViaSize property is only defined for Via objects.

Note: This is an alias for ViaDiameter (Field).

**Syntax**

`ViaSize : Number`

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

- `AsMils(ViaSize) : Number` {This specifies a distance with a unit of 1 mil.}
- `AsMM(ViaSize) : Number` {This specifies a distance with a unit of 1 mm.}

**Examples**

- `ViaSize = 40`
  Returns all Via objects that have a Via Diameter property which is 40 current measurement units.

- `ViaSize >= 0.508`
  Returns all Via objects that have a Via Diameter property which is greater than or equal to 0.508 current measurement units.

- `AsMils(ViaSize) Between 30 And 50`
Returns all Via objects that have a Via Diameter property which is greater than or equal to 30 mil and less than or equal to 50 mil. (The current value of the target file’s Measurement Unit property has no impact upon this outcome.)

\[ AsMM(ViaSize) < 1.524 \]

Returns all Via objects that have a Via Diameter property which is less than 1.524 mm. (The current value of the target file’s Measurement Unit property has no impact upon this outcome.)

ViolatingPrimitive1 Field

Description
Returns all Violation objects having a Primitive 1 property that complies with the Query.

Note: The ViolatingPrimitive1 property is only defined for Violation objects.

Syntax

ViolatingPrimitive1 : String

Example

ViolatingPrimitive1 = 'Net GND'

Returns all Violation objects that have a Primitive 1 property of 'Net GND'.

ViolatingPrimitive2 Field

Description
Returns all Violation objects having a Primitive 2 property that complies with the Query.

Note: The ViolatingPrimitive2 property is only defined for Violation objects.

Syntax

ViolatingPrimitive2 : String

Example

ViolatingPrimitive2 = 'Net +5V'

Returns all Violation objects that have a Primitive 2 property of 'Net +5V'.

ViolationDescription Field

Description
Returns all Violation objects having a Violation Description property that complies with the Query.

Note: The ViolationDescription property is only defined for Violation objects.

Syntax

ViolationDescription : String

Example

ViolationDescription = 'Actual Width = 8mil'

Returns all Violation objects that have a Violation Description property of 'Actual Width = 8mil'.

W Field

Description
Returns arcs and tracks whose Width property complies with the Query.

Note: The W property is only defined for Arc and Track objects.

Note: This is an alias for Width (Field).

Syntax

W : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(W) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(W) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**

W = 20

Returns all Arc and Track objects that have a Width property which is 20 current measurement units.

W >= 0.381

Returns all Arc and Track objects that have a Width property which is greater than or equal to 0.381 current measurement units.

AsMils(W) Between 10 And 30

Returns all Arc and Track objects that have a Width property which is greater than or equal to 10 mil and less than or equal to 30 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(W) < 0.635

Returns all Arc and Track objects that have a Width property which is less than 0.635 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

### Width Field (PCB)

**Description**

Returns arcs and tracks whose Width property complies with the Query.

Note: The Width property is only defined for Arc and Track objects.

Note: This is an alias for W (Field).

**Syntax**

Width : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(Width) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(Width) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**

Width = 20

Returns all Arc and Track objects that have a Width property which is 20 current measurement units.

Width >= 0.381

Returns all Arc and Track objects that have a Width property which is greater than or equal to 0.381 current measurement units.

AsMils(Width) Between 10 And 30

Returns all Arc and Track objects that have a Width property which is greater than or equal to 10 mil and less than or equal to 30 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(Width) < 0.635

Returns all Arc and Track objects that have a Width property which is less than 0.635 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
**WithinRoom Membership Check**

**Description**
Returns Arc, Fill, Text, and Track objects that **totally** reside within the region occupied by any Room having a Name property that is specified by the Query, and Pad and Via objects whose centre points reside within the region occupied by any Room having a Name property that is specified by the Query.

Note: The parent objects of group objects (Components, Coordinates, Dimensions, Nets, and Polygons) are not also returned.

**Syntax**

WithinRoom(Name : String) : Boolean
WithinRoom(Name : String , Name : String) : Boolean
WithinRoom(Name : String , Name : String , Name : String) : Boolean

**Examples**

WithinRoom('Power')
WithinRoom('Power') = True
Returns Arc, Fill, Text, and Track objects that **totally** reside within the region occupied by any Room having a Name property of 'Power', and Pad and Via objects whose centre points reside within the region occupied by any Room having a Name property of 'Power'.

WithinRoom('COUT1','COUT2')
WithinRoom('COUT1','COUT2') = True
Returns Arc, Fill, Text, and Track objects that **totally** reside within the region occupied by any Room having a Name property of 'COUT1' or 'COUT2', and Pad and Via objects whose centre points reside within the region occupied by any Room having a Name property of 'COUT1' or 'COUT2'.

WithinRoom('Monitor','Auxilary','Effects')
WithinRoom('Monitor','Auxilary','Effects') = True
Returns Arc, Fill, Text, and Track objects that **totally** reside within the region occupied by any Room having a Name property of 'Monitor' or 'Auxilary' or 'Effects', and Pad and Via objects whose centre points reside within the region occupied by any Room having a Name property of 'Monitor' or 'Auxilary' or 'Effects'.

**X1 Field**

**Description**
Returns objects whose X1 property complies with the Query.

Note: The X1 property is only defined for Arc, Fill, Pad, Text, Track, Via, Component, Coordinate, and Original Dimension objects.

**Syntax**

X1 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(X1) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(X1) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**

X1 = 1200
Returns all objects that have a X1 property which is 1200 current measurement units.

X1 >= 38.1
Returns all objects that have a X1 property which is greater than or equal to 38.1 current measurement units.

AsMils(X1) Between 1000 And 1400
Returns all objects that have a X1 property which is greater than or equal to 1000 mil and less than or equal to 1400 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(X1) < 20.32

Returns all objects that have a X1 property which is less than 20.32 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

**X2 Field**

**Description**

Returns fills, tracks, and dimensions whose X2 property complies with the Query.

Note: The X2 property is only defined for Fill, Track, and Original Dimension objects.

**Syntax**

X2 : Number

If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).

If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).

A specific option for the distance unit can be specified as follows:

AsMils(X2) : Number {This specifies a distance with a unit of 1 mil.}

AsMM(X2) : Number {This specifies a distance with a unit of 1 mm.}

**Examples**

X2 = 1200

Returns all Fill, Track, and Original Dimension objects that have a X2 property which is 1200 current measurement units.

X2 >= 38.1

Returns all Fill, Track, and Original Dimension objects that have a X2 property which is greater than or equal to 38.1 current measurement units.

AsMils(X2) Between 1000 And 1400

Returns all Fill, Track, and Original Dimension objects that have a X2 property which is greater than or equal to 1000 mil and less than or equal to 1400 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

AsMM(X2) < 20.32

Returns all Fill, Track, and Original Dimension objects that have a X2 property which is less than 20.32 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
Y1 Field

Description
Returns objects whose Y1 property complies with the Query.
Note: The Y1 property is only defined for Arc, Fill, Pad, Text, Track, Via, Component, Coordinate, and Original Dimension objects.

Syntax
Y1 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(Y1) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(Y1) : Number {This specifies a distance with a unit of 1 mm.}

Examples
Y1 = 1200
Returns all objects that have a Y1 property which is 1200 current measurement units.
Y1 >= 38.1
Returns all objects that have a Y1 property which is greater than or equal to 38.1 current measurement units.
AsMils(Y1) Between 1000 And 1400
Returns all objects that have a Y1 property which is greater than or equal to 1000 mil and less than or equal to 1400 mil. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)
AsMM(Y1) < 20.32
Returns all objects that have a Y1 property which is less than 20.32 mm. (The current value of the target file's Measurement Unit property has no impact upon this outcome.)

Y2 Field

Description
Returns fills, tracks, and dimensions whose Y2 property complies with the Query.
Note: The Y2 property is only defined for Fill, Track, and Original Dimension objects.

Syntax
Y2 : Number
If the target file has an 'Imperial' Measurement Unit property, the Number specifies a distance with a unit of 1 mil (1 mil = 0.001 inch).
If the target file has a 'Metric' Measurement Unit property, the Number specifies a distance with a unit of 1 mm (1 mm = 0.001 meter).
A specific option for the distance unit can be specified as follows:
AsMils(Y2) : Number {This specifies a distance with a unit of 1 mil.}
AsMM(Y2) : Number {This specifies a distance with a unit of 1 mm.}

Examples
Y2 = 1200
Returns all Fill, Track, and Original Dimension objects that have a Y2 property which is 1200 current measurement units.
Y2 >= 38.1
Returns all Fill, Track, and Original Dimension objects that have a Y2 property which is greater than or equal to 38.1 current measurement units.
AsMils(Y2) Between 1000 And 1400
Returns all Fill, Track, and Original Dimension objects that have a Y2 property which is greater than or equal to 1000 mil and less than or equal to 1400 mil. (The current value of the target file’s Measurement Unit property has no impact upon this outcome.)

AsMM(Y2) < 20.32
Returns all Fill, Track, and Original Dimension objects that have a Y2 property which is less than 20.32 mm. (The current value of the target file’s Measurement Unit property has no impact upon this outcome.)

**Z0 Impedance**

**Description**
Specifies the characteristic impedance required between a track on a Signal layer and the copper on an (adjacent) Internal Plane layer (or the copper on (adjacent) surrounding Internal Plane layers) within a multi-layer PCB. That characteristic impedance has an influence upon what width that track must subsequently have.

Note: This is an alias for CharacteristicImpedance (Impedance).

**Syntax**
Z0 : Number

The characteristic impedance is specified in units of ohms.

**Examples**

\[((5.98*TraceToPlaneDistance/EXP(Z0*SQRT(Er+1.41)/87))- TraceHeight)*1.25\]

A modified form of the default Query (in which Z0 has replaced CharacteristicImpedance) for specifying the required width of a track on the Signal layer, when the Microstrip option is selected.

\[((1.9*PlaneToPlaneDistance/EXP(Z0/(60/SQRT(Er))))- TraceHeight)*1.25\]

A modified form of the default Query (in which Z0 has replaced CharacteristicImpedance) for specifying the required width of a track on the intermediate Signal layer, when the Stripline option is selected.

**System Functions**

**ABS Arithmetic Function**

**Description**
This function returns a (real) number whose value is equal to the absolute value of the parameter's numeric value. (If that numeric value is positive, the number returned has the same value; otherwise the number returned is equal to the parameter's numeric value multiplied by -1.)

**Syntax**
ABS(X : Number / Numeric_String) : Number

**Examples**
ABS(2.3)
Returns 2.3.

ABS(-7.2)
Returns 7.2.

**See Also**
SIGN Arithmetic Function
ZERO Arithmetic Function

**ACOS Trigonometric Function**

**Description**
This function returns a (real) number whose value is equal to the angle of the inverse cosine of the number specified by the 
parameter's numeric value; that angle is in units of radians (and not degrees).

Syntax

ACOS(X : Number / Numeric_String) : Number

Examples

ACOS(0)
Returns ~ 1.5708.
ACOS(0.5)
Returns ~ 1.0472.
ACOS(1)
Returns 0.

ASIN Trigonometric Function

Description

This function returns a (real) number whose value is equal to the angle of the inverse sine of the number specified by the 
parameter's numeric value; that angle is in units of radians (and not degrees).

Syntax

ASIN(X : Number / Numeric_String) : Number

Examples

ASIN(0)
Returns 0.
ASIN(0.5)
Returns ~ 0.5236.
ASIN(1)
Returns ~ 1.5708.

ATAN Trigonometric Function

Description

This function returns a (real) number whose value is equal to the angle of the inverse tangent of the number specified by the 
parameter's numeric value; that angle is in units of radians (and not degrees).

Syntax

ATAN(X : Number / Numeric_String) : Number

Examples

ATAN(0)
Returns 0.
ATAN(1)
Returns ~ 0.7854.
ATAN(2.5)
Returns ~ 1.1903.
AVG Aggregate Function

Description
This function returns a (real) number whose value is equal to the average value of all of the parameters provided. Each parameter must be either a number or a numeric string, and any number of parameters can be provided (as long as at least one parameter is provided).

Syntax
AVG(N1 : Number / Numeric_String) : Number
AVG(N1 , N2 : Number / Numeric_String) : Number
AVG(N1 , N2 , N3 : Number / Numeric_String) : Number

Examples
AVG(2)
Returns 2.
AVG(7,2,3)
Returns 4.

CEIL Arithmetic Function

Description
This function returns an integral number, whose value depends upon the nature of the parameter's numeric value. If that numeric value is an exact integer, the number returned has exactly the same value; otherwise the number returned is the smallest integer that is larger in value than the parameter's numeric value.

Syntax
CEIL(X : Number / Numeric_String) : Number

Examples
CEIL(5.0)
Returns 5.
CEIL(5.33)
Returns 6.
CEIL(-5.0)
Returns -5.
CEIL(-5.33)
Returns -5.

See Also
FLOOR Arithmetic Function

Copy System Function

Description
This function returns a substring of a string. It is possible to specify how many characters will be within the substring returned, and the location within the string which will be the first character of the substring returned.

Syntax
Copy(S : String , Index : Integer , Count : Integer) : String

S is a string-type expression. Index and Count are integer-type expressions. Copy returns a substring containing Count characters starting at S[Index].
If Index is larger than the length of S, Copy returns an empty string.
If Count specifies more characters than are available, only the characters from S[Index] to the end of S are returned.
Examples

Copy('Cats and dogs',1,1)
Returns 'C'. (Return 1 character, from the 1st location within the string.)
Copy('Cats and dogs',2,1)
Returns 'a'. (Return 1 character, from the 2nd location within the string.)
Copy('Cats and dogs',13,1)
Returns 's'. (Return 1 character, from the 13th location within the string.)
Copy('Cats and dogs',14,1)
Returns " (an empty string). (Return 1 character, from the 14th location within the string; however, there are only 13 characters within the string, so an empty string is returned.)
Copy('Cats and dogs',1,2)
Returns 'Ca'. (Return 2 characters, starting from the 1st location within the string.)
Copy('Cats and dogs',2,2)
Returns 'at'. (Return 2 characters, starting from the 2nd location within the string.)
Copy('Cats and dogs',12,2)
Returns 'gs'. (Return 2 characters, starting from the 12th location within the string.)
Copy('Cats and dogs',13,2)
Returns 's'. (Return 2 characters, starting from the 13th location within the string; however, there are only 13 characters within the string, so only 1 character gets returned.)
Copy('Cats and dogs',15,2)
Returns " (an empty string). (Return 2 characters, starting from the 15th location within the string; however, there are only 13 characters within the string, so an empty string gets returned.)
Copy('Cats and dogs',1,6)
Returns 'Cats a'. (Return 6 characters, starting from the 1st location within the string.)
Copy('Cats and dogs',5,9)
Returns ' and dogs'. (Return 9 characters, starting from the 5th location within the string.)
Copy('Cats and dogs',8,5)
Returns 'd dog'. (Return 5 characters, starting from the 8th location within the string.)
Copy('Cats and dogs',8,6)
Returns 'd dogs'. (Return 6 characters, starting from the 8th location within the string.)
Copy('Cats and dogs',8,7)
Returns 'd dogs'. (Return 7 characters, starting from the 8th location within the string; however, there are only 13 characters within the string, so only 6 characters get returned.)
COS Trigonometric Function

Description
This function returns a (real) number whose value is equal to the cosine of the angle specified by the parameter's numeric value; that angle is in units of radians (and not degrees).

Syntax
COS(X : Number / Numeric_String) : Number

Examples
COS(0)
Returns 1.
COS(0.8)
Returns ~ 0.6967.
COS(1.57)
Returns ~ 0.0008.

COSH Trigonometric Function

Description
This function returns a (real) number whose value is equal to the hyperbolic cosine of the parameter's numeric value.

Syntax
COSH(X : Number / Numeric_String) : Number

Examples
COSH(TextHeight / TextWidth) < 70.5
Returns objects for which the hyperbolic cosine of the ratio of their TextHeight distance to their TextWidth distance is less than 70.5. (That is equivalent to returning objects for which the ratio of their TextHeight distance to their TextWidth distance is less than ~ 4.9487.)
(TextHeight / TextWidth) >= COSH(2.3)
Returns objects for which the ratio of their TextHeight distance to their TextWidth distance is equal to or greater than the hyperbolic cosine of 2.3 (which is ~ 5.0372).

COTAN Trigonometric Function

Description
This function returns a (real) number whose value is equal to the cotangent of the angle specified by the parameter's numeric value; that angle is in units of radians (and not degrees).

Syntax
COTAN(X : Number / Numeric_String) : Number

Examples
COTAN(0.1)
Returns ~ 9.9666.
COTAN(0.8)
Returns ~ 0.9712.
COTAN(1.57)
Returns ~ 0.0008.

EXP Exponential/Logarithmic Function

Description
This function returns a (real) number whose value is equal to $e$ raised to the power of the parameter's numeric value. (The number $e$ is the base of natural logarithms, and is approximately equal to 2.7183.)

**Syntax**

`EXP(X : Number / Numeric_String) : Number`

**Examples**

- `EXP(0)` returns 1.
- `EXP(1)` returns ~2.7183.

### FLOOR Arithmetic Function

**Description**

This function returns an integral number, whose value depends upon the nature of the parameter's numeric value. If that numeric value is an exact integer, the number returned has exactly the same value; otherwise the number returned is the largest integer that is smaller in value than the parameter's numeric value.

**Syntax**

`FLOOR(X : Number / Numeric_String) : Number`

**Examples**

- `FLOOR(5.0)` returns 5.
- `FLOOR(5.33)` returns 5.
- `FLOOR(-5.0)` returns -5.
- `FLOOR(-5.33)` returns -6.

**See Also**

CEIL Arithmetic Function

### FRAC Arithmetic Function

**Description**

This function returns a (real) number, whose value depends upon the nature of the parameter's numeric value.

If that numeric value is positive but is not an exact integer, the number returned is the difference between the parameter's numeric value and the largest integer that is smaller in value than the parameter's numeric value.

If that numeric value is negative but is not an exact integer, the number returned is the difference between the parameter's numeric value and the smallest integer that is larger in value than the parameter's numeric value.

If the parameter's numeric value is an exact integer, the number returned is 0.

**Syntax**

`FRAC(X : Number / Numeric_String) : Number`

**Examples**

- `FRAC(5.0)` returns 0.
- `FRAC(5.33)` returns 0.33.
- `FRAC(-5.0)` returns 0.
- `FRAC(-5.33)` returns 0.
**FRAC(-5.33)**

Returns -0.33.

**See Also**

INT Arithmetic Function / TRUNC Arithmetic Function

**IIF Arithmetic Function**

**Description**

This is a special purpose construction which contains three queries. The value of the first query (between the opening bracket and the first comma) is evaluated. If that is True, the second query (between the commas) is returned; otherwise, the third query (between the second comma and closing bracket) is returned.

This Keyword is similar in concept to "If ... Then ... Else ..." constructions provided with many computer programming languages.

**Syntax**

\[
\text{IIF}(L: \text{Boolean Query}, A: \text{Query}, B: \text{Query})
\]

L, A, and B are all strings which each specify some Query; L needs to be a Boolean Query (i.e. of a nature that returns either a True or False result).

IIF(L,A,B) has an identical outcome to the Query comprised of:

\[
((L = \text{True}) \text{ And } A) \text{ Or } ((L = \text{False}) \text{ And } B)
\]

As such, it is especially useful when L is of a complex nature, as it only needs to be specified once.

**Examples**

IsArc And IIF(ArcStopAngle - ArcStartAngle >= 0, ArcStopAngle - ArcStartAngle Between 30 And 60, ArcStartAngle - ArcStopAngle Between 300 And 330)

Returns all arcs that have an arc length which is greater than or equal to 30 degrees and less than or equal to 60 degrees. (The arc length property of an arc is an angle which is determined by subtracting its Start Angle property from its Stop Angle property. A complicating consideration is that it is possible for an arc's Stop Angle property to be smaller than its Start Angle property, but this Query correctly evaluates the true arc length property for all possible arc objects.)

IsComponent And IIF(Layer = 'Top Layer', Rotation <> 0, Rotation <> 180)

Returns all components that are on the top side of the PCB whose Rotation property is not 0 degrees, and all components that are on the bottom side of the PCB whose Rotation property is not 180 degrees. (Component objects can only reside on the Top signal layer or Bottom signal layer, so any component which is not on the Top signal layer must then be on the Bottom signal layer instead.)

**INT Arithmetic Function**

**Description**

This function returns an integral number, whose value depends upon the nature of the parameter's numeric value.

If that numeric value is positive but is not an exact integer, the number returned is the largest integer that is smaller in value than the parameter's numeric value.

If that numeric value is negative but is not an exact integer, the number returned is the smallest integer that is larger in value than the parameter's numeric value.

If that numeric value is an exact integer, the number returned has exactly the same value.

Note: This is an alias for TRUNC (Arithmetic Function).

**Syntax**

\[
\text{INT}(X: \text{Number / Numeric}\_\text{String}): \text{Number}
\]

**Examples**

INT(5.0)

Returns 5.

INT(5.33)

Returns 5.
INT(-5.0)  
Returns -5.  
INT(-5.33)  
Returns -5.  

See Also  
FRAC Arithmetic Function

**Length System Function**

**Description**  
This function returns the number of characters contained within a string.

**Syntax**  
Length(S : String) : Integer  
S is the string whose character count is to be returned.

**Examples**  
Length(Name) Between 4 And 7  
Returns objects that have a Name property which contains between 4 and 7 characters.  
Length('Cat')  
Returns 3.  
Length('Darryll')  
Returns 7.  
Length('Cats and dogs')  
Returns 13.  
Length('')  
Returns 0.

**LG Exponential/Logarithmic Function**

**Description**  
This function returns a (real) number whose value is equal to the base 10 logarithm of the parameter's numeric value.

**Syntax**  
LG(X : Number / Numeric_String) : Number

**Examples**  
LG(1)  
Returns 0.  
LG(10)  
Returns 1.  
LG(25)  
Returns ~ 1.3979.
**LN Exponential/Logarithmic Function**

**Description**
This function returns a (real) number whose value is equal to the natural (or base e) logarithm of the parameter's numeric value. (The number e is ~ 2.7183.)

**Syntax**

\[
\text{LN}(X : \text{Number} / \text{Numeric\_String}) : \text{Number}
\]

**Examples**

- \(\text{LN}(1)\)
  - Returns 0.
- \(\text{LN}(2)\)
  - Returns ~ 0.6931.
- \(\text{LN}(10)\)
  - Returns ~ 2.3026.

**LOG Exponential/Logarithmic Function**

**Description**
This function returns a (real) number whose value is equal to the base 2 logarithm of the parameter's numeric value.

**Syntax**

\[
\text{LOG}(X : \text{Number} / \text{Numeric\_String}) : \text{Number}
\]

**Examples**

- \(\text{LOG}(1)\)
  - Returns 0.
- \(\text{LOG}(2)\)
  - Returns 1.
- \(\text{LOG}(8)\)
  - Returns 3.
- \(\text{LOG}(10)\)
  - Returns ~ 3.3219.

**MAX Aggregate Function**

**Description**
This function returns a (real) number whose value is equal to the maximum value of all of the parameters provided. Each parameter must be either a number or a numeric string, and any number of parameters can be provided (as long as at least one parameter is provided).

**Syntax**

\[
\text{MAX}(N1 : \text{Number} / \text{Numeric\_String}) : \text{Number}
\]

**Examples**

- \(\text{MAX}(2)\)
  - Returns 2.
- \(\text{MAX}(7,2,3)\)
  - Returns 7.
MIN Aggregate Function

Description
This function returns a (real) number whose value is equal to the minimum value of all of the parameters provided. Each parameter must be either a number or a numeric string, and any number of parameters can be provided (as long as at least one parameter is provided).

Syntax
MIN(N1 : Number / Numeric_String) : Number
MIN(N1 , N2 : Number / Numeric_String) : Number
MIN(N1 , N2 , N3 : Number / Numeric_String) : Number

Examples
MIN(2)
Returns 2.
MIN(7,2,3)
Returns 2.

PI Arithmetic Function

Description
This function returns a (real) number whose value is approximately equal to pi (which is ~ 3.1416, and is the ratio of a circle’s circumference to its diameter).

It has been provided because all of the trigonometric functions which have also been provided either return an angle in units of radians, or use a parameter of an angle in units of radians, and there could be times when users would prefer to refer to angles in units of degrees instead.

Syntax
PI : Number

Examples
SIN(PI / 2)
Returns 1.
COS(PI / 3)
Returns 0.5.
TAN(PI / 4)
Returns 1.

Pos System Function

Description
This function returns the index value of the first character in a specified substring that occurs in a given string.

Syntax
Pos(Substr : String , S : String) : Integer

Pos searches for a substring, Substr, in a string, S. Substr and S are string-type expressions.
Pos searches for Substr within S and returns an integer value that is the index of the first character of Substr within S. Pos is case-sensitive. If Substr is not found, Pos returns zero.

Examples
Pos('T',Name) = 1
Returns all objects that have a Name property whose associated string first contains 'T' at its first position; 'T1' and 'Time Table' are examples of such strings. ('T' is located twice within the string 'Time Table', but the first location of this is at its first position.)
Pos('A',Name) = 2
Returns all objects that have a Name property whose associated string first contains 'A' at its second position; 'RA1' and 'CABLE CHART' are examples of such strings.

```
Pos('B', Name) = 0
```

Returns all objects that have a Name property whose associated string does not contain 'B' at any position; 'D1' and 'Time Table' are examples of such strings. (The string of 'Time Table' contains 'b' at its eighth position, but Pos is case-sensitive.)

```
Pos('D', Name) > 0
```

Returns all objects that have a Name property whose associated string contains 'D' at at least one position and at any location(s); 'Data Chart', 'U1D', and 'Install either D1 and D2 or D3 and D4, but never install all of these.' are examples of such strings.

```
Pos('Con', Name) = 1
```

Returns all objects that have a Name property whose associated string first contains 'Con' at its first, second, and third positions; 'Concatenate' is an example of a such string. (The first location of 'Con' within 'Concatenate' is the first three characters of this; the first character within 'Con' is 'C', and its location is the first character of 'Concatenate'.)

```
Pos('ate', Name) = 5
```

Returns all objects that have a Name property whose associated string first contains 'ate' at its fifth, sixth, and seventh positions; 'Concatenate' is an example of a such string. (The first location of 'ate' within 'Concatenate' is the fifth through to seventh characters of this; the first character within 'ate' is 'a', and its location is the fifth character of 'Concatenate'.)

```
Pos('ate', Name) = 0
```

Returns all objects that have a Name property whose associated string does not contain 'ate' anywhere; 'C1' and 'TIME:' are examples of such strings.

## POWER Arithmetic Function

### Description
This function returns a (real) number whose value is equal to the first parameter's numeric value raised to the power of the second parameter's numeric value.

### Syntax
```
POWER(X : Number / Numeric_String , Y : Number / Numeric_String) : Number
```

### Examples
```
POWER(3,2)
```
Returns 9.
```
POWER(32,0.2)
```
Returns 2.
```
POWER(16,-0.5)
```
Returns 0.25.

## Pred System Function

### Description
This function returns a (real) number whose value is equal to parameter's numeric value less 1.0.

### Syntax
```
Pred(X : Number / Numeric_String) : Number
```

### Example
```
Pred(4.1)
```
Returns 3.1.

### See Also
Succ System Function
PROD Aggregate Function

Description
This function returns a (real) number whose value is equal to the product of all of the parameters provided. Each parameter must be either a number or a numeric string, and any number of parameters can be provided (as long as at least one parameter is provided).

Syntax
PROD(N1 : Number / Numeric_String) : Number
PROD(N1 , N2 : Number / Numeric_String) : Number
PROD(N1 , N2 , N3 : Number / Numeric_String) : Number

Examples
PROD(2)
Returns 2.
PROD(7,2,3)
Returns 42.

Random System Function

Description
Returns a random number between 0 and X-1.

Syntax
Random(Number) : Number

Example
Random(10)
Returns a value between 0 and 9.

ROUND Arithmetic Function

Description
This function returns an integral number, whose value depends upon the nature of the parameter's numeric value.
If that numeric value is an exact integer, the number returned has exactly the same value.
If that numeric value is not an exact integer, the number returned is the integer which is closest in value to the parameter's numeric value. Positive numbers which are exactly midway between adjacent integers are rounded "upwards", while negative numbers which are exactly midway between adjacent integers are rounded "downwards" (see examples below).

Syntax
ROUND(X : Number / Numeric_String) : Number

Examples
ROUND(5.0)
Returns 5.
ROUND(5.33)
Returns 5.
ROUND(5.5)
Returns 6. (Rounded "upwards".)
ROUND(5.66)
Returns 6.
ROUND(-5.0)
Returns -5.
ROUND(-5.33)
Returns -5.

ROUND(-5.5)

Returns -6. (Rounded "downwards").

ROUND(-5.66)

Returns -6.

See Also
INT Arithmetic Function

**SIGN Arithmetic Function**

**Description**

This function returns sign value of x (=1 if x>0; =0 if x=0; =-1 if x <0). That is, it returns an integral number, whose value depends upon the nature of the parameter's numeric value.

If that numeric value is positive, the number returned is 1.

If that numeric value is negative, the number returned is -1.

If that numeric value is exactly zero, the number returned is 0.

**Syntax**

\[
\text{SIGN}(X : \text{Number / Numeric_String}) : \text{Number}
\]

**Examples**

\[
\text{SIGN}(5.33) \quad \text{Returns} \quad 1.
\]

\[
\text{SIGN}(-5.33) \quad \text{Returns} \quad -1.
\]

\[
\text{SIGN}(0) \quad \text{Returns} \quad 0.
\]

See Also
ABS Arithmetic Function
ZERO Arithmetic Function

**SIN Trigonometric Function**

**Description**

This function returns a (real) number whose value is equal to the sine of the angle specified by the parameter's numeric value; that angle is in units of radians (and not degrees).

**Syntax**

\[
\text{SIN}(X : \text{Number / Numeric_String}) : \text{Number}
\]

**Examples**

\[
\text{SIN}(0) \quad \text{Returns} \quad 0.
\]

\[
\text{SIN}(0.8) \quad \text{Returns} \quad \sim 0.7174.
\]

\[
\text{SIN}(1.57) \quad \text{Returns} \quad \sim 1.0.
\]
**SINH Trigonometric Function**

**Description**
This function returns a (real) number whose value is equal to the hyperbolic sine of the parameter's numeric value.

**Syntax**
\[ \text{SINH}(X : \text{Number / Numeric\_String}) : \text{Number} \]

**Examples**
\[ \text{SINH}(\text{TextWidth} / \text{TextHeight}) < 0.17 \]
Returns objects for which the hyperbolic sine of the ratio of their TextWidth distance to their TextHeight distance is less than 0.17. (That is equivalent to returning objects for which the ratio of their TextWidth distance to their TextHeight distance is less than ~ 0.1692.)

\[ (\text{TextWidth} / \text{TextHeight}) \geq \text{SINH}(0.2) \]
Returns objects for which the ratio of their TextWidth distance to their TextHeight distance is equal to or greater than the hyperbolic sine of 0.2 (which is ~ 0.2013).

**SQR Arithmetic Function**

**Description**
This function returns a (real) number whose value is equal to the square of the parameter's numeric value.

**Syntax**
\[ \text{SQR}(X : \text{Number / Numeric\_String}) : \text{Number} \]

**Examples**
\[ \text{SQR}(2) \]
Returns 4.

\[ \text{SQR}(14.1) \]
Returns 198.81.

\[ \text{SQR}(-1.5) \]
Returns 2.25.

**SQRT Arithmetic Function**

**Description**
This function returns a (real) number whose value is equal to the (positive) square root of the parameter's numeric value.

**Syntax**
\[ \text{SQRT}(X : \text{Number / Numeric\_String}) : \text{Number} \]

**Examples**
\[ \text{SQRT}(4) \]
Returns 2.

\[ \text{SQRT}(10) \]
Returns ~ 3.1623.
Succ System Function

**Description**
This function returns a (real) number whose value is equal to parameter's numeric value plus 1.0.

**Syntax**
Succ(X : Number / Numeric_String) : Number

**Example**
Succ(2.1)
Returns 3.1.

**See Also**
Pred System Function

SUM Aggregate Function

**Description**
This function returns a (real) number whose value is equal to the sum of all of the parameters provided. Each parameter must be either a number or a numeric string, and any number of parameters can be provided (as long as at least one parameter is provided).

**Syntax**
SUM(N1 : Number / Numeric_String) : Number
SUM(N1 , N2 : Number / Numeric_String) : Number
SUM(N1 , N2 , N3 : Number / Numeric_String) : Number

**Examples**
SUM(2)
Returns 2.
SUM(7,2,3)
Returns 12.

TAN Trigonometric Function

**Description**
This function returns a (real) number whose value is equal to the tangent of the angle specified by the parameter's numeric value; that angle is in units of radians (and not degrees).

**Syntax**
TAN(X : Number / Numeric_String) : Number

**Examples**
TAN(0)
Returns 0.
TAN(0.8)
Returns ~ 1.0296.
TAN(1.57)
Returns ~ 1255.7656.

TANH Trigonometric Function

**Description**
This function returns a (real) number whose value is equal to the hyperbolic tangent of the parameter's numeric value.
**Syntax**

TANH(X : Number / Numeric_String) : Number

**Examples**

TANH(TextWidth / TextHeight) < 0.17

Returns objects for which the hyperbolic tangent of the ratio of their TextWidth distance to their TextHeight distance is less than 0.17. (That is equivalent to returning objects for which the ratio of their TextWidth distance to their TextHeight distance is less than ~ 0.1717.)

(TextWidth / TextHeight) >= TANH(0.2)

Returns objects for which the ratio of their TextWidth distance to their TextHeight distance is equal to or greater than the hyperbolic tangent of 0.2 (which is ~ 0.1974).

---

**TRUNC Arithmetic Function**

**Description**

This function returns integer part of x (same as INT(x)) that is, it returns an integral number, whose value depends upon the nature of the parameter's numeric value.

If that numeric value is positive but is not an exact integer, the number returned is the largest integer that is smaller in value than the parameter's numeric value.

If that numeric value is negative but is not an exact integer, the number returned is the smallest integer that is larger in value than the parameter's numeric value.

If that numeric value is an exact integer, the number returned has exactly the same value.

**Note:** This is an alias for INT (Arithmetic Function).

**Syntax**

TRUNC(X : Number / Numeric_String) : Number

**Examples**

TRUNC(5.0)

Returns 5.

TRUNC(5.33)

Returns 5.

TRUNC(-5.0)

Returns -5.

TRUNC(-5.33)

Returns -5.

**See Also**

FRAC Arithmetic Function

---

**ZERO Arithmetic Function**

**Description**

This function returns 0, if x=0 else returns 1, that is, it returns an integral number whose value is either 0 or 1, depending upon the parameter's numeric value. If that numeric value is zero, the number returned is 0; otherwise the number returned is 1.

**Syntax**

ZERO(X : Number / Numeric_String) : Number

**Examples**

ZERO(5.33)

Returns 1.

ZERO(-5.33)

Returns 1.
ZERO (0)

Returns 0.

See Also

ABS Arithmetic Function
SIGN Arithmetic Function
### Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version No.</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-Dec-2004</td>
<td>1.0</td>
<td>New product release</td>
</tr>
<tr>
<td>20-Sep-2005</td>
<td>1.1</td>
<td>Spaces removed from layer names in Layer and OnLayer keyword descriptions</td>
</tr>
<tr>
<td>9 Nov 2005</td>
<td>1.2</td>
<td>Updated for Altium Designer 6</td>
</tr>
<tr>
<td>2 Mar 2006</td>
<td>1.3</td>
<td>Revised Query Keywords</td>
</tr>
<tr>
<td>31 Mar 2006</td>
<td>1.4</td>
<td>Fixed a few PCB entries and formatting issues.</td>
</tr>
<tr>
<td>1-Dec-2006</td>
<td>1.5</td>
<td>Updated for Altium Designer 6.6 - new keywords.</td>
</tr>
<tr>
<td>18-Feb-2008</td>
<td>1.6</td>
<td>Updated for Altium Designer 6.9 - new keywords.</td>
</tr>
<tr>
<td>21-Apr-2008</td>
<td>1.7</td>
<td>Updated Page Size to A4</td>
</tr>
<tr>
<td>02-Aug-2011</td>
<td>-</td>
<td>Updated template.</td>
</tr>
</tbody>
</table>

Software, hardware, documentation and related materials:

Copyright © 2011 Altium Limited.

All rights reserved. You are permitted to print this document provided that (1) the use of such is for personal use only and will not be copied or posted on any network computer or broadcast in any media, and (2) no modifications of the document is made. Unauthorized duplication, in whole or part, of this document by any means, mechanical or electronic, including translation into another language, except for brief excerpts in published reviews, is prohibited without the express written permission of Altium Limited. Unauthorized duplication of this work may also be prohibited by local statute. Violators may be subject to both criminal and civil penalties, including fines and/or imprisonment.

Altium, Altium Designer, Board Insight, DXP, Innovation Station, LiveDesign, NanoBoard, NanoTalk, OpenBus, P-CAD, SimCode, Situs, TASKING, and Topological Autorouting and their respective logos are trademarks or registered trademarks of Altium Limited or its subsidiaries. All other registered or unregistered trademarks referenced herein are the property of their respective owners and no trademark rights to the same are claimed.