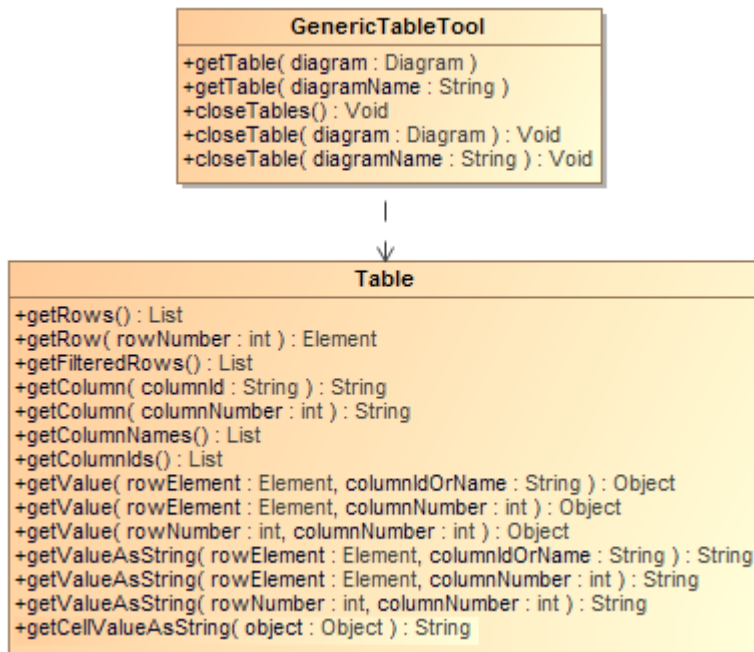


# Generic table tool API

## On this page

- Getting generic table data
  - Getting generic table instances from diagram elements
  - Getting generic table instances from diagram names
- Closing tables
  - Closing all diagram tables
  - Closing a specific table diagram
  - Closing a table diagram with a specific diagram name
- Getting row elements
  - Getting all row elements
  - Getting all filtered row elements
  - Getting row elements in specific row numbers
- Getting column names
  - Getting column names from column ID
  - Getting column names from column numbers
  - Getting all column names
  - Getting all column IDs
- Getting cell values
  - Getting values from row element and column ID
  - Getting values from row element and column name
  - Getting values from row element and column number
  - Getting values from row and column numbers
  - Getting values from row element and column IDs as string
  - Getting values from row element and column name as string
  - Getting values from row element and column number as string
  - Getting values from row and column numbers as string
  - Getting cell value from returned object as string
- Getting visible column and cell values
  - Getting all visible column IDs
  - Getting visible column names from column numbers
  - Getting visible values from row elements and column number
  - Getting visible values from row number and column number
  - Getting visible values from row element and column number as string
  - Getting visible values from row and column numbers as string



GenericTableTool diagram.

Class GenericTableTool:

Methods:

- +getTable( diagram : Diagram ) : Table

- +getTable( diagramName : String ) : Table
- +closeTables() : Void
- +closeTable(diagram : Diagram) : Void
- +closeTable(diagramName : String) : Void

Class Table:

Methods:

- +getRows() : List<Element>
- +getRow( rowNumber : int ) : Element
- +getFilteredRows() : List<Element>
- +getColumn( columnName : String ) : String
- +getColumn( columnNumber : int ) : String
- +getColumnNames() : List<String>
- +getColumnIds() : List<String>
- +getValue( rowElement : Element, columnName : String ) : Object
- +getValue( rowElement : Element, columnNumber : int ) : Object
- +getValue( rowNumber : int, columnNumber : int ) : Object
- +getValueAsString( rowElement : Element, columnName : String ) : String
- +getValueAsString( rowElement : Element, columnNumber : int ) : String
- +getValueAsString( rowNumber : int, columnNumber : int ) : String
- +getVisibleColumnIds() : List<String>
- +getVisibleColumn( columnNumber : int ) : String
- +getVisibleValue( rowElement : Element, columnNumber : int ) : Object
- +getVisibleValue( rowNumber : int, columnNumber : int ) : Object
- +getVisibleValueAsString( rowElement : Element, columnNumber : int ) : String
- +getVisibleValueAsString( rowNumber : int, columnNumber : int ) : String

## Getting generic table data

A Generic Table is a special diagram. You can retrieve it by using the variable \$Diagram. Thus, every method must accept the diagram instance or diagram name.

Use the following method to get the Generic Table instance. We use Generic Table instances to retrieve table information such as row elements and column names.

### Getting generic table instances from diagram elements

To get a Generic Table instance from a specified diagram element, use the following code.

**getTable( diagram : Diagram ) : Table**

```
#set($table = $generic.getTable($diagram))
```

Where the parameter is:

- diagram – the diagram element.

Return the instance of a Generic Table.

Example code:

```
#foreach($diagram in $project.getDiagrams("Generic Table"))
#set($table = $generic.getTable($diagram))
#end
```

### Getting generic table instances from diagram names

To get a Generic Table instance from a specific diagram name, use the following code.

**getTable( diagramName : String ) : Table**

```
#set($table = $generic.getTable($diagram))
```

Where the parameter is:

- diagram – the diagram name.

Return the instance of a Generic Table.

Example code:

```
#set($diagram = "diagram name")
#set($table = $generic.getTable($diagram))
```

## Closing tables

Use the following method to close a table diagram.

### Closing all diagram tables

**CloseTables() : Void**

```
$generic.closeTables()
```

### Closing a specific table diagram

**closeTable(diagram: Diagram) : Void**

```
$generic.closeTable($diagram)
```

Where the parameter is:

- diagram – a table diagram you want to close.

### Closing a table diagram with a specific diagram name

**closeTable(diagramName: String) : Void**

```
$generic.closeTable($diagramName)
```

Where the parameter is:

- diagramName – the name of a table diagram you want to close.

## Getting row elements

The Generic Table consists of a series of row elements and column names. Use the following methods to retrieve a list of row elements.

### Getting all row elements

**getRows() : List<Element>**

```
$table.getRows()
```

The returned value is a list of Elements. The following shows an example of how to print all row element names.

```
#foreach($row in $table.rows)
$row.name
#end
```

### Getting all filtered row elements

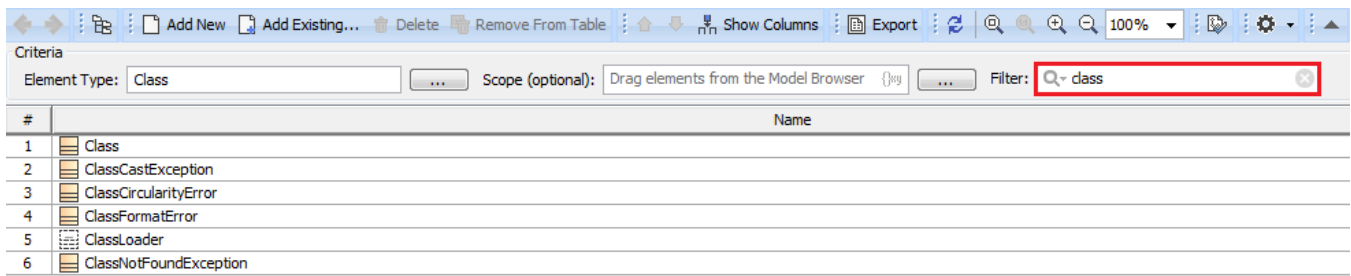
**getFilteredRows() : List<Element>**

```
$table.getFilteredRows()
```

The returned value is a list of Elements that have been filtered by the Filter field.

```
#foreach($row in $table.getFilteredRows())
$row.name
#end
```

The figure below shows the rows returned after applying the filter.



The screenshot shows a software interface with a search bar containing the text 'class'. Below the search bar is a table with two columns: '#' and 'Name'. The table contains six rows of class names:

#	Name
1	Class
2	ClassCastException
3	ClassCircularityError
4	ClassFormatError
5	ClassLoader
6	ClassNotFoundException

Rows returned after filtering.

## Getting row elements in specific row numbers

**getRow( rowNumber : int ) : Element**

```
$table.getRow($rowNumber)
```

Where the parameter is:

- rowNumber – the row number starts with 0.

The returned value is an Element.

## Getting column names

Column names are referred by column IDs. A column is a MagicDraw QProperty ID that is not readable or accessible by the user. We need to create other methods that provide better usability than using QProperty ID. Use the following methods to retrieve column names.



The methods in this section retrieves column names from the list returned by Show Columns menu item from generic table diagram. The methods in the preceding section, however, retrieves column names from the diagram itself and not from the list returned by Show Columns menu item.

## Getting column names from column ID

**getColumn( columnId : String ) : String**

```
$table.getColumn($columnId)
```

Where the parameter is:

- columnId – the ID of a column. Using column IDs has a benefit of consistency between different languages, for example, French and English. The column ID can be retrieved by the method getColumnIds().

The returned value is the name of a column.

## Getting column names from column numbers

**getColumn(columnNumber : int) : String**

```
$table.getColumn($columnNumber)
```

Where the parameter is:

- columnNumber – a column number starts with 1. The column number 0 is the row number.

The returned value is the name of a column.

## Getting all column names

**getColumnNames() : List<String>**

```
$table.getColumnNames()
```

The returned value is a list of column names.

Example code:

Prints all column names.

```
#foreach($colname in $table.getColumnNames())
$colname
#end
```

## Getting all column IDs

**getColumnIds() : List<String>**

```
$table.getColumnIds()
```

The returned value is a list of column IDs.

Example code:

Print all columns IDs

```
#foreach($colid in $table.getColumnIds())
$table.getColumn($colid)
#end
```

## Getting cell values

A cell value is the value in a row element with a column name. To get a cell value, you need to provide both the row element and column. A cell value is an Element or row number. Use the following methods to retrieve a cell value.

## Getting values from row element and column ID

**getValue( rowElement : Element, columnIdOrName : String ) : Object**

```
$table.getValue($rowElement, $columnId)
```

Where the parameters are:

- rowElement – a row element.
- columnId – a column ID.

The returned value is the value in a cell. If the cell contains an element, the value is the Element.

Example code:

Prints the row, column, and value.

```
#foreach($diagram in $project.getDiagrams("Generic Table"))
#set($table = $generic.getTable($diagram))
#foreach($row in $table.getRows())
<h1>$row.name</h1>
#foreach($col in $table.getColumnIds())
$table.getColumn($col) : $table.getValue($row, $col)
#end
#end
#end
```

## Getting values from row element and column name

**getValue( rowElement : Element, columnName : String ) : Object**

```
$table.getValue($rowElement, $columnName)
```

Where the parameters are:

- rowElement – a row element.
- columnName – a column name.

The returned value is the value in a cell. If the cell contains an element, the value is the Element.

## Getting values from row element and column number

**getValue( rowElement : Element, columnNumber : int ) : Object**

```
$table.getValue($rowElement, $columnNumber)
```

Where the parameters are:

- rowElement – a row element.
- columnNumber – a column number starts with 1. The column number 0 is the row number.

The returned value is the value in a cell. If the cell contains an element, the value is the Element.

## Getting values from row and column numbers

**getValue( rowNumber : int, columnNumber : int ) : Object**

```
$table.getValue($rowNumber, $columnNumber)
```

Where the parameters are:

- rowNumber – a row number starts with 0.
- columnNumber – a column number starts with 1. The column number 0 is the row number.

The returned value is the value in a cell. If the cell contains an element, the value is the Element.

## Getting values from row element and column IDs as string

**getValueAsString( rowElement : Element, columnName : String ) : String**

```
$table.getValueAsString($rowElement, $columnId)
```

Where the parameters are:

- rowElement – a row element.
- columnId – a column ID.

The returned value is the value in a cell and is converted into String. The String value is created by MagicDraw text representation API.

## Getting values from row element and column name as string

**getValueAsString( rowElement : Element, columnName : String ) : String**

```
$table.getValueAsString($rowElement, $columnName)
```

Where the parameters are:

- rowElement – a row element
- columnName – a column name

The returned value is the value in a cell and is converted into String. The String value is created by MagicDraw text representation API.

## Getting values from row element and column number as string

**getValueAsString( rowElement : Element, columnNumber : int ) : String**

```
$table.getValueAsString($rowElement, $columnNumber)
```

Where the parameters are:

- rowElement – a row element.
- columnName – a column number starts with 1. The column number 0 is the row number.

The returned value is the value in a cell and is converted into String. The String value is created by MagicDraw text representation API.

## Getting values from row and column numbers as string

**getValueAsString( rowNumber : int, columnNumber : int ) : String**

```
$table.getValueAsString($rowNumber, $columnNumber)
```

Where the parameters are:

- rowNumber – a row number starts with 0.
- columnNumber – a column number starts with 1. The column number 0 is the row number.

The returned value is the value in a cell and is converted into String. The String value is created by MagicDraw text representation API.

## Getting cell value from returned object as string

**getCellValueAsString( object : Object ) : String**

```
$table.getCellValueAsString($object)
```

Where the parameter is:

- object – an object returned from the getValue function.

The returned value is the object in cell value converted into String.

For example:

```
#set($result = $table.getValue($row, $id))
#ife($list.isArray($result))
    #foreach($r in $result)
        - $table.getCellValueAsString($r)
    #end
#else
    $table.getCellValueAsString($result)
#end
```

## Getting visible column and cell values

A visible cell value is the column which is visible on a diagram. Unlike the methods described in section 23.1.5 Getting Cell Values, the following methods will retrieve only the value which is visible on a diagram.

### Getting all visible column IDs

**getVisibleColumnIds() : List<String>**

```
$table.getVisibleColumnIds()
```

The returned value is a list of visible column IDs.

### Getting visible column names from column numbers

### **getVisibleColumn( columnNumber : int ) : String**

```
$table.getVisibleColumn($columnNumber)
```

Where the parameter is:

- columnNumber – a column number starts with 1. The column number 0 is the row number.

The returned value is the name of a visible column.

### **Getting visible values from row elements and column number**

#### **getVisibleValue( rowElement : Element, columnNumber : int ) : Object**

```
$table.getVisibleValue($rowElement, $columnNumber)
```

Where the parameters are:

- rowElement – a row element.
- columnNumber – a column number starts with 1. The column number 0 is the row number.

The returned value is the value in a visible cell. If the cell contains an element, the value is the Element.

### **Getting visible values from row number and column number**

#### **getVisibleValue( rowNumber : int, columnNumber : int ) : Object**

```
$table.getVisibleValue($rowNumber, $columnNumber)
```

Where the parameters are:

- rowNumber – a row number starts with 0.
- columnNumber – a column number starts with 1. The column number 0 is the row number.

The returned value is the value in a visible cell. If the cell contains an element, the value is the Element.

### **Getting visible values from row element and column number as string**

#### **getVisibleValueAsString( rowElement : Element, columnNumber : int ) : String**

```
$table.getVisibleValueAsString($rowElement, $columnNumber)
```

Where the parameters are:

- rowElement – a row element.
- columnNumber – a column number starts with 1. The column number 0 is the row number.

The returned value is the value in a visible cell and is converted into String. The String value is created by MagicDraw text representation API.

### **Getting visible values from row and column numbers as string**

#### **getVisibleValueAsString( rowNumber : int, columnNumber : int ) : String**

```
$table.getVisibleValueAsString($rowNumber, $columnNumber)
```

Where the parameters are:

- rowNumber – a row number starts with 0.
- columnNumber – a column number starts with 1. The column number 0 is the row number.

The returned value is the value in a visible cell and is converted into String. The String value is created by MagicDraw text representation API.