

Sorter tool

On this page

- `$sorter.sort(collection, fieldName)`
- `$sorter.sort(collection, fieldName, isCaseInsensitive)`
- `$sorter.sort(collection)`
- `$sorter.sort(collection, fieldNameCollection, direction)`
- `$sorter.sort(collection, fieldName, direction, isCaseInsensitive)`
- `$sorter.sortByFirstNumber(collection, fieldName)`
- `$sorter.sortByFirstNumber(collection)`
- `$sorter.sortByLocale(collection, String)`
- `$sorter.sortByLocale(collection, String, String)`
- `$sorter.humanSort(collection, fieldName)`
- `$sorter.humanSort(collection, fieldName, isCaseInsensitive)`
- `$sorter.humanSort(collection)`
- `$sorter.humanSort(collection, fieldNameCollection, direction)`
- `$sorter.humanSort(collection, fieldName, direction, isCaseInsensitive)`
- `$sorter.containmentTreeSort(collection)`
- `$sorter.tagSort(collection, stereotypeName, tagName)`
- `$sorter.tagSort($collection, $stereotypeName, $tagName, $forceNumber)`
- `$sorter.tagSort($collection, $stereotype, $tagName)`
- `$sorter.tagSort($collection, $stereotype, $tagName, $forceNumber)`

\$sorter is used to sort a collection of report templates.

 **Information**
The following functions perform case-insensitive sorting by default:

- `$sorter.sort(collection, fieldName)`
- `$sorter.sort(collection)`

\$sorter.sort(collection, fieldName)

`$sorter.sort(collection, fieldNameCollection, direction)`

The sort function for report templates. The context name of this class is "sorter". Use \$sorter to access public functions of this class through templates.

`$sorter.humanSort(collection, direction)`

	• <code>\$sorter.humanSort(collection)</code> • <code>\$sorter.humanSort(collection, fieldNameCollection, direction)</code>	Name	Type	Description
Parameter(s)		collection	java.util.Collection	A collection to be sorted
		fieldName	java.lang.String	A fieldName to be sorted and the sort direction.
Return		-	java.util.Collection	A sorted collection.

For example:

```
#foreach ($rel in $sorter.sort($package, "name"))
$rel.name
#end
```

- \$package is the collection to be sorted.
- "name:desc" is separated by ":" in two parts:
 - The first part is to identify fieldName to be sorted.
 - The second part is the option to identify the sorting direction. Sometimes, the direction is not identified. It is ascending by default.

\$sorter.sort(collection, fieldName, isCaseInsensitive)

The sort function for report templates. This function allows case-sensitive sorting.

	Name	Type	Description
Parameter(s)	collection	java.util.Collection	A collection to be sorted.
	fieldName	java.lang.String	A fieldName to be sorted and the sort direction.
	isCaseInsensitive	boolean	True if case-insensitive sorting.
Return	-	java.util.Collection	A sorted collection.

\$sorter.sort(collection)

This is the sort function for report templates. The context name of this class is "sorter". Use \$sorter to access public functions of this class through the templates.

	Name	Type	Description
Parameter(s)	collection	java.util.Collection	A collection to be sorted.
Return	-	java.util.Collection	A sorted collection.

For example:

```
#foreach ($rel in $sorter.sort($package))
$rel.name
#end
```

- \$package is a collection to be sorted.



The collection will be sorted by name, in ascending order, by default.

\$sorter.sort(collection, fieldNameCollection, direction)

This sort function allows specifying more than one field name to be sorted. The priority of fields is ranked by the sequence in the fieldNameCollection parameter.

	Name	Type	Description
Parameter(s)	collection	java.util.Collection	A collection to be sorted.
	fieldNameCollection	java.util.Collection	A collection of field names, sorted by the priority of field name order in the collection.
	direction	java.lang.String	Sorting direction. It is either "asc" or "desc". If this field is not entered, "asc" is used by default.
Return	-	java.util.Collection	A sorted collection.

For example:

```
#foreach ($p in $sorter.sort($Package, ["version", "name"], "desc"))
$p.name
#end
```

\$sorter.sort(collection, fieldName, direction, isCaseInsensitive)

This sort function allows specifying more than one field name to be sorted. The priority of fields is ranked by the sequence in the fieldNameCollection parameter. This function allows case-sensitive sorting.

	Name	Type	Description
Parameter(s)	collection	java.util.Collection	A collection to be sorted.
	fieldName	java.lang.String	A fieldName to be sorted and the sort direction.
	direction	java.lang.String	Sorting direction. It is either "asc" or "desc". If this field is not entered, "asc" is used by default.
	isCaseInsensitive	boolean	True if case-insensitive sorting.
Return	-	java.util.Collection	A sorted collection.

\$sorter.sortByFirstNumber(collection, fieldName)

The sortByFirstNumber function is for report templates. The context name of this class is "sorter". Use \$sorter to access public functions of this class through templates.

	Name	Type	Description
Parameter(s)	collection	java.util.Collection	A collection to be sorted.
	fieldName	java.lang.String	A fieldName to be sorted and the sorting direction.
Return	-	java.util.Collection	A sorted collection.

For example:

```
#foreach ($rel in $sorter.sortByFirstNumber($package, "name:desc"))
$rel.name
#end
```

- \$package is a collection to be sorted by **FirstNumber**.
- "name:desc" is separated by ":" in two parts:
 - The first part is to identify fieldName to be sorted.
 - The second part is the option to identify the sorting direction. Sometimes, the direction is not identified. It is ascending by default.

\$sorter.sortByFirstNumber(collection)

The sortByFirstNumber function for report templates. The context name of this class is "sorter". Use \$sorter to access public functions of this class through templates.

	Name	Type	Description
Parameter(s)	collection	java.util.Collection	A collection to be sorted.
Return	-	java.util.Collection	A sorted collection.

For example:

```
#foreach ($rel in $sorter.sortByFirstNumber($package))
$rel.name
#end
```

- \$package is a collection to be sorted by FirstNumber.

\$sorter.sortByLocale(collection, String)

This is a function to sort for report templates. The context name of this class is "sorter". Use \$sorter to access public functions of this class through templates. To sort a given collection by a particular language, identify the country code to specify the language.

	Name	Type	Description
Parameter(s)	collection	java.util.Collection	A collection to be sorted.
	countryCode	java.lang.String	The country code to specify a language to perform sorting.
Return	-	java.util.Collection	A sorted collection.

For example:

```
#foreach ($p in $sorter.sortByLocale($package, "DE"))
$p.name
#end
```

- \$package is a collection to be sorted by country code.
- "DE" is the country code for GERMANY (ISO country code).



This method sorts a collection by "name" attribute of each element by default.

\$sorter.sortByLocale(collection, String, String)

This is a function to sort for report templates. The context name of this class is "sorter". Use \$sorter to access public functions of this class through templates. To sort a given collection by a particular language, specify the language by identifying the country code and field name.

	Name	Type	Description
Parameter(s)	collection	java.util.Collection	A collection to be sorted.
	fieldName	java.lang.String	A fieldName to be sorted.
	countryCode	java.lang.String	The country code to specify a language to perform sorting.
Return	-	java.util.Collection	A sorted collection.

For example:

```
#foreach ($p in $sorter.sortByLocale($package, "name", "DE"))
$p.name
#end
```

- \$package is the collection to be sorted.
- "name" is the field name to be sorted.
- "DE" is the country code for GERMANY (ISO country code).

\$sorter.humanSort(collection, fieldName)

This is a special function to sort text in a human-like order. It splits text into numeric and non-numeric chunks and sorts them in numerical order. For example, "foo10" is ordered after "foo2".

	Name	Type	Description
Parameter(s)	collection	java.util.Collection	A collection to be sorted.
	fieldName	java.lang.String	A fieldName to be sorted and the sorting direction.
Return	-	java.util.Collection	A sorted collection.

For example:

```
#foreach ($rel in $sorter.humanSort($package, "name:desc"))
$rel.name
#end
```

- \$package is the collection to be sorted.
- "name:desc" is separated by ":" in two parts:
 - The first part is to identify fieldName to be sorted.

- The second part is the option to identify the sorting scheme. Sometimes, the order is not identified. It is ascending by default.

\$sorter.humanSort(collection, fieldName, isCaseInsensitive)

This is a special function to sort text in a human-like order. It splits text into numeric and non-numeric chunks and sorts them in numerical order. For example, "foo10" is ordered after "foo2". This function allows case-sensitive sorting.

	Name	Type	Description
Parameter(s)	collection	java.util.Collection	A collection to be sorted.
	fieldName	java.lang.String	A fieldName to be sorted and the sorting direction.
	isCaseInsensitive	boolean	True if case-insensitive sorting.
Return	-	java.util.Collection	A sorted collection.

\$sorter.humanSort(collection)

This is a special function to sort text in a human-like order. It splits text into numeric and non-numeric chunks and sorts them in numerical order. For example, "foo10" is ordered after "foo2".

	Name	Type	Description
Parameter(s)	collection	java.util.Collection	A collection to be sorted.
Return	-	java.util.Collection	A sorted collection.

For example:

```
#foreach ($rel in $sorter.humanSort($package))
$rel.name
#end
```

- \$package is the collection to be sorted.

\$sorter.humanSort(collection, fieldNameCollection, direction)

This humanSort function allows specifying more than one field name to be sorted. The priority of fields is ranked by the sequence in the fieldNameCollection parameter.

	Name	Type	Description
Parameter(s)	collection	java.util.Collection	A collection to be sorted.
	fieldNameCollection	java.util.Collection	A collection of field names, sorted by the priority of field name order in the collection.
	direction	java.lang.String	Sorting direction. It is either "asc" or "desc". If this field is not entered, "asc" is used by default.
Return	-	java.util.Collection	A sorted collection.

For example:

```
#foreach ($p in $sorter.humanSort($Package, ["version", "name"], "desc"))
$p.name
#end
```

\$sorter.humanSort(collection, fieldName, direction, isCaseInsensitive)

This humanSort function allows specifying more than one field name to be sorted. The priority of fields is ranked by the sequence in the fieldNameCollection parameter. This function allows case-sensitive sorting.

	Name	Type	Description
Parameter(s)	collection	java.util.Collection	A collection to be sorted.

	fieldName	java.lang.String	A fieldName to be sorted and the sorting direction.
	direction	java.lang.String	Sorting direction. It is either "asc" or "desc". If this field is not entered, "asc" is used by default.
	isCaseInsensitive	boolean	True if case-insensitive sorting.
Return	-	java.util.Collection	A sorted collection.

\$sorter.containmentTreeSort(collection)

This function is for sorting elements that match the Containment tree order.

	Name	Type	Description
Parameter(s)	collection	java.util.Collection	A collection to be sorted.
Return	-	java.util.Collection	A sorted collection.

\$sorter.tagSort(collection, stereotypeName, tagName)

This function is for sorting a collection item by specifying an applied stereotype name and tag name.

	Name	Type	Description
Parameter(s)	collection	java.util.Collection	A collection to be sorted.
	stereotypeName	java.lang.String	An applied stereotype name.
	tagName	java.lang.String	A tag name of the applied stereotype and the sorting direction.
Return	-	java.util.Collection	A sorted collection.

For example:

```
#foreach($p in $sorter.tagSort($Package, "MyStereotype", "tagA:desc"))
$report.getStereotypePropertyString($p, "Mystereotype", "tagA") :: $p.name
#end
```

- \$package is the collection to be sorted.
- "name:desc" is separated by ":" in two parts:
 - The first part is to identify tagName to be sorted.
 - The second part is the option to identify the sorting scheme. Sometimes, the order is not identified. It is ascending by default.

\$sorter.tagSort(\$collection, \$stereotypeName, \$tagName, \$forceNumber)

This function is for sorting a collection item by specifying an applied stereotype name and tag name and forcing a tag value to convert to a numeric before sorting.

	Name	Type	Description
Parameter(s)	collection	java.util.Collection	A collection to be sorted.
	stereotypeName	java.lang.String	An applied stereotype name.
	tagName	java.lang.String	A tag name of the applied stereotype and the sorting direction.
	forceNumber	boolean	True if a tag value is converted to a numeric before sorting.
Return	-	java.util.Collection	A sorted collection.

For example:

```
#foreach($p in $sorter.tagSort($Package, "MyStereotype", "tagA:desc", true))
$report.getStereotypePropertyString($p, "Mystereotype", "tagA") :: $p.name
#end
```

- \$package is the collection to be sorted.

- "name:desc" is separated by ":" in two parts:
 - The first part is to identify tagName to be sorted.
 - The second part is the option to identify the sorting scheme. Sometimes, the order is not identified. It is ascending by default.
- If this parameter is not defined in the function, the forceNumber default value of **false** will be applied.

\$sorter.tagSort(\$collection, \$stereotype, \$tagName)

This function is for sorting a collection item by specifying an applied stereotype object and tag name.

	Name	Type	Description
Parameter(s)	collection	java.util.Collection	A collection to be sorted.
	stereotype	com.nomagic.uml2.ext.magicdraw.mdprofiles.Stereotype	An applied stereotype object.
	tagName	java.lang.String	A tag name of the applied stereotype and the sorting direction.
Return	-	java.util.Collection	A sorted collection.

For example:

```
#foreach($p in $sorter.tagSort($Package, $stereotype, "tagA:desc"))
$report.getStereotypePropertyString($p, "Mystereotype", "tagA") :: $p.name
#end
```

- \$package is the collection to be sorted.
- "name:desc" is separated by ":" in two parts:
 - The first part is to identify tagName to be sorted.
 - The second part is the option to identify the sorting scheme. Sometimes, the order is not identified. It is ascending by default.

\$sorter.tagSort(\$collection, \$stereotype, \$tagName, \$forceNumber)

This function is for sorting a collection item by specifying an applied stereotype object and tag name and forcing a tag value to convert to a numeric before sorting.

	Name	Type	Description
Parameter(s)	collection	java.util.Collection	A collection to be sorted.
	stereotype	com.nomagic.uml2.ext.magicdraw.mdprofiles.Stereotype	An applied stereotype object.
	tagName	java.lang.String	A tag name of the applied stereotype and the sorting direction.
	forceNumber	boolean	True if a tag value is converted to a numeric before sorting.
Return	-	java.util.Collection	A sorted collection.

For example:

```
#foreach($p in $sorter.tagSort($Package, $stereotype, "tagA:desc", true))
$report.getStereotypePropertyString($p, "Mystereotype", "tagA") :: $p.name
#end
```

- \$package is the collection to be sorted.
- "name:desc" is separated by ":" in two parts:
 - The first part is to identify tagName to be sorted.
 - The second part is the option to identify the sorting scheme. Sometimes, the order is not identified. It is ascending by default.
- If this parameter is not defined in the function, the forceNumber default value of **false** will be applied.